



UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Escola d'Enginyeria de Barcelona Est

TREBALL FI DE GRAU

Grau en Enginyeria Electrònica Industrial i Automàtica

**POSADA EN MARXA DE COMUNICACIONS OPC UA PER A
ROBOT COL·LABORATIU**



Memòria i Annexos

Autor: Pol Blancafort Figueras
Director: Pedro Ponsa Asensio
Convocatòria: Gener 2023

Resum

En l'actualitat es disposa de diferents mètodes per comunicar el creixent nombre de dispositius a la indústria i als centres de producció. Alguns d'aquests mètodes es van desenvolupar fa gairebé mig segle i comencen a tenir importants mancances. En els últims anys, s'han fet intents d'establir nous protocols de comunicació desenvolupats d'acord a les necessitats de la indústria 4.0: una comunicació fiable, ràpida i segura entre robots, màquines, sensors i persones.

OPC UA és un dels nous protocols que ja ha agafat més força al sector de l'automatització, degut en part a que compta amb avançades mesures de seguretat i un ampli suport de dispositius de diferents marques. El focus d'aquest treball acadèmic és documentar el desplegament d'aquest estàndard de comunicació entre el robot UR3e (Universal Robots) i una sèrie de programes informàtics.

Un programari extern instal·lat al robot habilita la possibilitat de fer aplicacions robòtiques que treballin amb la comunicació OPC UA. Al llarg d'aquest treball, s'investiga de quines maneres es pot utilitzar la xarxa OPC UA per intercanviar informació d'interès, controlar el robot i millorar el cicle de producció. Paral·lelament es realitzarà una guia detallada per tal de donar instruccions a futurs estudiants i investigadors sobre els primers passos de la comunicació OPC UA i el seu ús en robots col·laboratius.

Resumen

En la actualidad se dispone de diferentes métodos para comunicar el creciente número de dispositivos en la industria i en los centros de producción. Algunos de estos métodos se idearon a finales del siglo XX y empiezan a mostrar signos de desfase. Por este motivo, en los últimos años ha crecido el interés en nuevos protocolos de comunicación diseñados con las necesidades de la industria 4.0 en mente: una comunicación fiable, rápida y segura entre todo el ecosistema de dispositivos, ya sean máquinas, sensores o personas.

OPC UA es uno de los protocolos relativamente nuevo que más fuerza está cogiendo en el sector de la automatización, debido en parte a su apuesta por la seguridad y un amplio soporte para multitud de dispositivos y marcas. El foco de este trabajo es documentar el despliegue de esta tecnología de comunicación entre el robot UR3e (Universal Robots) i una serie de programas informáticos.

Un programa externo instalado en el robot habilita el estándar de comunicación OPC UA y permite realizar aplicaciones robóticas que interactúen con esta tecnología. A lo largo de este trabajo se investiga las diferentes maneras de usar OPC UA para el intercambio de información, el control del robot y para mejorar el ciclo de producción. Al mismo tiempo, se realizará una guía detallada para proporcionar unas pautas iniciales a investigadores y estudiantes que quieran configurar la tecnología OPC UA y usarla en robots colaborativos.

Abstract

Nowadays, we have several methods to communicate the growing number of devices in the industry and manufacturing centres. Some of these options were developed more than four decades ago and are now showing signs of age. In recent years, the popularity of new communications protocols architected with the ideas of Industry 4.0 has increased steadily alongside the need to have a reliable, fast, and secure network for sensors, machines, robots, and people.

One of these relatively new options is the standard OPC UA, which offers excellent support from devices by being platform-independent with a strong focus on security. These attributes are helping OPC UA become one of the top choices for scenarios where there are multiple devices sharing data.

The aim of this research is to document the set-up of such technology between the robot UR3e (Universal Robots) and some PC programs. At the same time, the information in this project will be available in a concise step-by-step guide that will serve as a starting point for future students and researchers wanting to use OPC UA with collaborative robots.



Agraïments

Aquest treball va néixer gràcies al seu Director Pedro Ponsa, sense el qual no hagués estat possible. Les seves indicacions, aportacions i l'ajuda constant al llarg de tot al projecte han estat essencials per donar forma al treball i aconseguir els objectius proposats.

En segon lloc vull agrair a la Universitat Politècnica de Catalunya l'aposta per tecnologies innovadores com el robot UR3e i el protocol de comunicacions OPC UA i la compra de recursos sense els quals no s'hagués pogut realitzar la part pràctica d'investigació d'aquest treball.



Glossari

UR3e *model de robot articulat de la marca Universal Robots*

OPC UA *acrònim del protocol de comunicació OPC Unified Architecture*

Programari *(software, en anglès) conjunt dels programes informàtics, procediments i documentació que fan alguna tasca en un ordinador/màquina*

Teach Pendant *tauleta tàctil, consola de programació del robot UR3e*

UR *Universal Robots*

PolyScope *Programari dels robots UR, entorn on es realitza la configuració, programació i el control del robot*

URCap *plataforma d'UR que permet a distribuïdors i fabricants afegir accessoris als robots de la marca per ampliar les seves funcions*

IP *Internet Protocol*

ISA *Integració de Sistemes Automàtics, assignatura universitària*

RIVC *Robòtica Industrial i Visió per Computador, assignatura universitària*

SICIEIA *Sistemes d'Informació i Comunicació Industrial, assignatura universitària*

Cobot *terme utilitzat per referir-se als robots col·laboratius, és a dir, aquells que estan dissenyats per a treballar conjuntament amb persones i interactuar directament amb elles*

Servidor *element principal d'una xarxa de comunicacions que té el rol d'iniciar la comunicació, aportar dades a la xarxa, oferir paràmetres de configuració etc.*

Client *element secundari d'una xarxa de comunicacions que es connecta al Servidor i pot llegir i/o proporcionar dades a la xarxa*

Discovery *funcionalitat present en el Clients de les xarxes OPC UA que els hi permet descobrir tots els elements que participen a la xarxa OPC UA*

Scripts *metodologia de programació dels robots UR basada en el llenguatge Python que es presenta com una alternativa més avançada a la programació estàndard disponible dins del PolyScope*

Tallafoc *(de l'anglès Firewall) Sistema que hom col·loca entre una xarxa d'àrea local i una xarxa pública com internet, per tal d'assegurar que totes les comunicacions entre la xarxa local i la pública es realitzen segons les polítiques de seguretat de l'organització propietària de la xarxa local*

PopUp *traducció en anglès del terme 'finestres emergents'*

Índex

RESUM	I
RESUMEN	II
ABSTRACT	III
AGRAÏMENTS	V
GLOSSARI	VII
1. PREFACI	1
1.1. Motivació del treball	1
1.2. Requeriments previs	2
1.3. Objectius del treball	2
1.4. Abast i estructuració del contingut	3
2. PROTOCOL OPC UA	5
2.1. Orígens de l'estàndard OPC	5
2.2. Introducció a OPC Unified Architecture	7
2.2.1. Visió general de OPC UA	7
2.2.2. Estructura del protocol	8
2.3. Transport i accés a les dades	8
2.4. Seguretat	12
2.4.1. Model de seguretat de OPC UA	12
2.4.2. Posada en marxa d'una sessió OPC UA	14
2.4.3. Codi de bones pràctiques	16
3. ROBOT UR3E	19
3.1. Introducció al model UR3e	20
3.1.1. Universal Robots	20
3.1.2. Característiques tècniques del robot	21
3.2. Comunicacions compatibles amb el robot	23
3.2.1. MODBUS TCP	23
3.2.2. PROFINET	25
3.2.3. ETHERNET/IP	26
3.2.4. MQTT	27
3.2.5. OPC UA	28
3.3. URCap OPC UA	28

4.	CONFIGURACIÓ I ÚS DE L'ENTORN OPC UA	29
4.1.	Requeriments previs.....	30
4.2.	Configuració del Servidor	30
4.3.	Variables OPC UA.....	32
4.4.	Programació específica de OPC UA.....	34
4.5.	Clients.....	35
4.5.1.	UAExpert	35
4.5.2.	OPC Expert	39
4.5.3.	Prosys UA Monitor	45
4.6.	Tests de seguretat.....	50
5.	APLICACIONS ROBÒTIQUES AMB OPC UA	57
5.1.	Bucle simple OPC UA	57
5.2.	Programa amb interacció de l'operari	61
5.3.	Assemblatge amb funcions OPC UA.....	66
	ANÀLISI DE L'IMPACTE AMBIENTAL	78
	CONCLUSIONS	79
	ANÀLISI ECONÒMICA	81
	BIBLIOGRAFIA	83
	ANNEX A	87
A1.	Captures de pantalla dels escenaris dels test de seguretat.....	87
A2.	Especificacions robot UR3e	99
A3.	Programes complets de les aplicacions robòtiques.....	100

1. Prefaci

En els últims anys la indústria 4.0 ha esdevingut un terme clau del sector tecnològic. Es parla d'una revolució en la qual persones i màquines treballen de manera conjunta per millorar la producció, reduir costos i augmentar la competitivitat de les empreses. Actualment, ja existeixen moltes marques que disposen de centres moderns en els quals es treballa amb totes les innovacions de la Indústria 4.0: robots, *IoT* (Internet de les coses), 5G, *BigData*, Intel·ligència Artificial entre d'altres.

Una de les claus d'aquest nou paradigma de treball és la comunicació eficaç entre el creixent nombre de màquines i dispositius a les plantes de producció. Tots els dispositius comparteixen dades en temps real sobre el seu estat, possibles errors, mesures dels múltiples sensors etc. Avui en dia, disposem de diversos protocols de comunicació establerts ja en la indústria, els quals tenen els seus pros i contres però tots busquen oferir una comunicació fiable, robusta, ràpida i segura.

1.1. Motivació del treball

Tant el director d'aquest treball, Pedro Ponsa, com jo, el seu autor, compartim un punt de vista similar sobre el valor que haurien de tenir les pràctiques dels estudis en enginyeries: la universitat té el deure de formar als estudiants amb aquelles tecnologies que estan en ús en la indústria per afavorir que, arribats a l'entorn professional, els estudiants puguem desenvolupar tasques en un ampli rang d'escenaris i dispositius.

Les xarxes de comunicació són un àmbit amb un rol essencial a la indústria i són part del temari d'assignatures com ISA i SICIEIA. El temps disponible en les pràctiques d'aquestes assignatures és limitat, com també ho és el material a disposició dels alumnes. Per aquest motiu, cal focalitzar els esforços i els recursos econòmics disponibles a estudiar modes de comunicació que realment tinguin un pes en l'àmbit de l'automatització i que tinguin garantit el seu ús en el futur.

OPC UA és un exponent perfecte d'aquesta voluntat de treballar amb eines actuals. En l'actualitat aquest protocol ja s'utilitza en escenaris industrials reals, a banda de que professionals del sector asseguruen que seguirà sent competitiu en el futur [1]. No obstant, degut a la seva relativa modernitat, cal més documentació del seu ús en àmbits acadèmics. En l'àmbit d'actuació de la Universitat Politècnica de Catalunya, s'han realitzat alguns Treballs de Fi d'Estudis que investiguen OPC UA en diversos escenaris (simulació [2], logística [3], emergència climàtica [4] etc.). Ara bé, en relació a l'àmbit d'automatització i robòtica hi han pocs documents. I més concretament, OPC UA treballant amb el

robot d'UR suposa una completa innovació i no hi ha pràcticament antecedents. Aquest és el punt de partida del TFE.

1.2. Requeriments previs

Al llarg d'aquest treball es faran ús de dispositius i programaris que requereixen certa experiència en informàtica. Més concretament, s'ha treballat amb xarxes de comunicació on participen Servidors i Clients en l'enviament de paquets, trames i variables de diferents tipus. Aquesta terminologia és comuna en l'àmbit de SICIEIA, assignatura que juntament amb ISA, dona una bona base teòrica per encarar aquest projecte.

En referència als coneixements de robòtica i programació, ha estat essencial realitzar el curs didàctic disponible online a l'acadèmia de UR, el qual explica detalladament les característiques del robot i dona instruccions sobre la seva programació i el seu funcionament normal. El llenguatge de programació del robot es visual i intuïtiu, però internament es basa en Python (llenguatge de programació d'alt nivell) i la tauleta de control del robot treballa amb Linux (sistema operatiu de codi obert). Coneixements en aquests sectors no són fonamentals però han arribat a facilitar certes tasques.

Adicionalment han estat útils els coneixements proporcionats per l'assignatura RIVC.

1.3. Objectius del treball

El primer objectiu clau d'aquest projecte acadèmic és detallar la configuració del protocol de comunicacions OPC UA en el cas específic del robot UR3e.

El segon objectiu del treball serà realitzar el disseny i la programació d'una aplicació d'assemblatge que englobi tots els conceptes apresos fins al moment i exemplifiqui les possibilitats de control i monitoratge del protocol en aplicacions robòtiques.

El conjunt d'aquesta informació culminarà en la realització d'una guia on es trobaran instruccions resumides de la configuració i els possibles usos de l'entorn OPC UA en el robot UR3e. Aquest és, doncs, el tercer objectiu del meu treball de fi d'estudis: proporcionar a futurs estudiants i investigadors els coneixements necessaris per configurar i utilitzar una xarxa OPC UA amb un model de robot de la marca Universal Robots.

1.4. Abast i estructuració del contingut

El director d'aquest treball ja va destacar durant la fase conceptual que l'abast del mateix vindria condicionat en tot moment pel temps disponible. Degut a que es compta amb molt poca informació prèvia referent al projecte, no era realista posar terminis a les primeres etapes del projecte. Per exemple, era molt difícil saber (en un principi) quan es trigaria a tenir un servidor OPC UA actiu i funcionant. Per aquest motiu es va acordar dotar al projecte d'un elevat grau de flexibilitat pel que fa al contingut estudiat. Es comptava, doncs, amb uns quatre mesos aproximadament per intentar aconseguir els tres objectius claus proposats. El treball ha anat evolucionant cada cop que s'ha aconseguit alguna fita important, moment en el qual es realitzaven les reestructuracions pertinents tenint en compte el temps disponible restant.

Aquest enfocament flexible i àgil ha permès invertir el temps suficient en cada una de les seccions del projecte i anar afegint de noves quan es creia convenient, en funció del temps invertit respecte el total disponible. En certs moments va ser necessari descartar tests o seccions completes d'acord amb aquestes pautes. A les conclusions d'aquest projecte apareixen propostes de possibles continuacions del treball, que fan èmfasis a seccions que només es cobreixen de manera general en aquest escrit però que es podrien ampliar en futurs TFE.

El treball està dividit en quatre grans blocs: el protocol OPC UA, el robot UR3e, configuració de la xarxa OPC UA i un quart apartat on s'exposen els programes del robot realitzats. El contingut de cada secció es detalla a continuació:

1. La primera part del projecte inclou una extensa i detallada cerca d'informació sobre l'estàndard OPC UA, els seus orígens i el seu impacte en la indústria. Aquesta cerca d'informació s'ha anat completant al llarg dels quatre mesos disponibles.
2. El segon bloc detalla les característiques i les especificacions tècniques del robot UR3e, l'empresa que el fabrica i el conjunt de protocols de comunicació que suporta.
3. El tercer bloc actua com a nexa dels dos primers punts. En aquesta secció es detallarà com habilitar i configurar OPC UA en el cas particular del robot UR3e. Aquesta fase del projecte equivaldria al treball de camp, on es realitzen proves físicament al laboratori i es documenten els resultats. Es començarà per entendre el programari extern que s'ha instal·lat al robot que permet l'ús del protocol OPC UA, així com les seves funcionalitats i els diferents paràmetres disponibles per l'usuari.

4. En el bloc final del projecte s'exposen els programes que s'han anat realitzant per provar els diferents paràmetres de la comunicació OPC UA i investigar les possibilitats d'aquesta tecnologia. En l'últim programa presentat, que correspon a un assemblatge, es posen en pràctica tots els coneixements i les descobertes fetes al llarg del treball en un cas pràctic real.

2. Protocol OPC UA

2.1. Orígens de l'estàndard OPC

D'ençà els anys 90, l'ús d'ordinadors i sistemes d'automatització basats en programaris (*software*, en anglès) va augmentar de manera ràpida i consistent d'acord amb les necessitats productives de la societat. La popularitat i el creixent suport de Windows (desenvolupat pel gegant Microsoft) com a sistema operatiu propicia el seu ús per tasques de visualització i control. Així doncs, en els centres de producció es busca poder treballar amb aquest sistema operatiu juntament amb tots els dispositius ja instal·lats. Aquest enfocament, però, topa amb una problemàtica important: la convivència en un mateix espai d'aparells de diferents fabricants, diversos tipus de busos de comunicació i, a sobre de tot, protocols de comunicació diferents per cada component.

Els grans proveïdors de programaris de HMI (Interfície Home-Màquina) i d'SCADA (Supervisió, Control i Adquisició de Dades) s'enfronten al problema d'haver de donar suport a Windows de tots els diferents dispositius del mercat i a més a més, anticipar-se a les noves generacions de dispositius que ja s'estaven dissenyant a les seus dels fabricants del sector.

En aquest context, l'any 1995 [5] es crea un grup de treball per donar resposta al creixent problema de compatibilitat el qual és impulsat per les companyies Fisher-Rosemount, Rockwell Software, Opto 22 i Intellution. El seu objectiu és clar: definir un estàndard *Plug&Play* (terme per definir un sistema de fàcil connexió i ús) per als controladors (o *drivers*) dels dispositius, permeten així un accés estandaritzat a les dades d'automatització en sistemes Windows

El resultat va ser el llançament de l'especificació OPC Data Access l'agost del 1996 [5]. La nomenclatura OPC deriva inicialment de OLE (sistema de Microsoft per annexar i integrar objectes) per Control de Processos. El manteniment i la millora d'aquest estàndard es va posar en mans de la organització sense ànim de lucre OPC Foundation [6], que neix arrel del grup de treball. Actualment, OPC Foundation segueix en actiu i la seva nomenclatura ha evolucionat cap a **Open Platform Communications** (Plataforma de comunicacions oberta) per remarcar la importància de la seva missió: administrar una organització global que vetlla per crear estàndards de transferència de dades per el sector de l'automatització. Destaquen un important focus en donar suport a diferents marques i distribuïdors i proporcionar una interconnexió segura i robusta.

La primera versió de OPC Data Acces [7] (DA) ja s'enfocava cap a aquesta direcció i en només un any des de la seva publicació diversos fabricants de software i hardware van adoptar aquest protocol. La capacitat d'aquest protocol per llegir, sobreescriure i monitorar variables en temps real dels diferents

processos van convèncer a les grans marques del sector. D'aquesta manera, es confirmava, per una banda, la necessitat d'estàndards d'aquesta mena, però al mateix temps es feia evident que calia seguir avançant cap a una formalització, certificació i validació a gran escala del protocol.

Les següents versions de l'estàndard OPC segueixen aquestes pautes i al mateix temps incorporen millores d'acord als comentaris i suggeriments rebuts de la primera generació d'usuaris.

L'any 1999 es posa a l'abast dels usuaris OPC Alarms & Events [8] (OPC AE). Aquesta especificació de segona generació incorpora les capacitats de OPC DA i introdueix el suport per rebre notificacions d'eventualitats i alarmes. Els diferents dispositius poden informar als operaris de l'inici/final d'un procés o el canvi inesperat d'algun paràmetre dins del funcionament normal. A diferència de la primera generació, no cal una instrucció explícita per rebre aquestes notificacions, sinó que els diferents dispositius les van enviant a mesures que tenen lloc. Es compta amb sistemes de filtratge per limitar el nombre i la freqüència de les notificacions.

Ja al segle XXI, a l'any 2001 s'ha de destacar la publicació de dos especificacions: OPC Historical Data Access i OPC Security. La primera d'elles, OPC HDA [9] (Historical Data Acces) proporciona accés a dades emmagatzemades. Els usuaris d'aquesta especificació tenen accés a dades del passat i poden determinar quina informació visualitzar en un rang de temps determinat. El protocol incorpora també els elements necessaris per introduir, substituir i eliminar dades de la base de dades històrica. La segona especificació, OPC Security [10], habilita la possibilitat de que l'administrador de l'entorn OPC afegixi millores de seguretat i limiti l'accés a dades confidencials, per tal de protegir-se de modificacions del sistema no autoritzades. També dona unes pautes de com treballar en un entorn OPC segur. Aquesta especificació permet múltiples nivells de seguretat que coexisteixen amb les altres aplicacions OPC.

En els següents anys, OPC Foundation introdueix noves especificacions que incorporen millores iteratives i graduals. Cal remarcar la versió OPC XML-DA [11], publicada el 2003. Aquesta especificació dona suport per tecnologies web (HTTP, SOAP) i certes capacitats d'accés a internet. Al final, aquesta especificació no va tenir tant èxit com s'havia previst degut en part a un funcionament limitat pel seu disseny i un elevat consum de recursos. Es va acabar aplicant en sistemes integrats i plataformes que no treballaven amb Microsoft.

L'any 2003 va ser també l'any en que es va idear la creació d'un nou estàndard OPC, el qual havia de recollir tota l'experiència aconseguida fins al moment i incorporar les millors parts de cada publicació prèvia. Naixia així **OPC Unified Architecture**.

2.2. Introducció a OPC Unified Architecture

La ràpida adopció de les especificacions OPC clàssiques (DA, HDA) va tenir una conseqüència inesperada per la OPC Foundation: OPC s'utilitzava fins i tot en àrees per les quals no havia estat dissenyat ni degudament testejat. Els fabricants es trobaven per altra banda que volien utilitzar l'estàndard OPC per certes tasques però no era factible per les limitacions relacionades amb COM i DCOM (ambdós són els sistemes en els quals es basa l'enviament i accés de dades en OPC).

L'especificació que dona títol al meu treball, OPC Unified Architecture [12] (OPC UA) neix arrel del desig de crear un veritable substitut de l'especificació original basada en COM (OPC DA) sense perdre cap atribut ni sacrificar rendiment. Així doncs, havia de mantenir els pilars centrals que havien fet tant popular la versió original: fiabilitat i robustesa, escalabilitat, seguretat, alt rendiment i, per sobre de tot, independència de la plataforma emprada.

2.2.1. Visió general de OPC UA

OPC UA compta amb un grau molt més elevat de flexibilitat i té moltes més característiques que totes les especificacions OPC originals juntes. Al mateix temps, incorpora tots els conceptes exitosos de les versions prèvies, soluciona els principals problemes detectats en les existents especificacions i afegeix estandardització per molts escenaris d'ús addicionals.

Per tal de permetre una transició relativament fàcil de l'OPC clàssic a OPC UA, OPC Foundation dissenya la nova especificació per a que contingui molts dels atributs coneguts de la versió original, simplement que en alguns d'ells actualitza la nomenclatura. Per aquest motiu, la publicació de la versió 1.0 [5] al Juliol de 2006 es considera més aviat una evolució i no pas una revolució.

Es tardaria un any més per poder trobar els primers productes al mercat basats en OPC UA . També al 2007, OPC Foundation introdueix el Programa de Certificació i els Laboratoris de Tests [13] per garantir el suport i el rendiment adequats en una àmplia oferta de dispositius.

L'any 2010 apareixen els primers dispositius OPC UA integrats que incorporen la versió 1.01 de l'estàndard. En aquest context s'inicia l'adopció a gran escala del nou estàndard OPC UA.

2.2.2. Estructura del protocol

Molts usuaris dins del sector de l'automatització van apostar per el nou protocol. Aquesta decisió es va recolzar, en part, a l'avançat funcionament de OPC UA. Aquesta especificació permet crear una xarxa de comunicació per enviar, rebre i accedir ràpidament a diversos tipus de dades dins un context segur i fiable. El disseny i l'arquitectura del protocol [14] van ser claus per aconseguir aquestes fites.

OPC UA es va idear amb una estructura basada en capes, cada una orientada a donar resposta a un dels objectius clau de la nova especificació:

- **Equivalència funcional:** es dona suport a les versions antigues i es tradueixen els diferents paràmetres d'OPC Clàssic a OPC UA. Basant-se en l'èxit de OPC clàssic, OPC UA millora i supera les prestacions de les primeres versions. OPC UA és funcionalment equivalent a OPC clàssic però amb moltes més funcions.
- **Independència de la plataforma utilitzada:** la nova especificació s'enfoca a múltiples escenaris d'ús, des d'un microcontrolador integrat a una infraestructura al núvol, independentment del fabricant o la marca. OPC UA proporciona la infraestructura necessària per treballar amb dispositius de tota mena funcionant amb Windows/Apple OSX/Android/Linux.
- **Seguretat:** encriptació, autenticació i auditoria de la informació retransmesa.
- **Ampliable:** disposa de l'habilitat d'afegir noves característiques sense afectar les aplicacions existents. Aquesta capa proporciona la capacitat a OPC UA de ser 'actualitzable' en el futur donant suport a nous protocols de transport, algorismes de seguretat, sistemes de xifrat etc.
- **Complet modelatge de la informació:** es disposa de la capacitat per modelar i traduir en dades útils sistemes simples i al mateix temps estructures complexes.

Totes aquestes capes participen activament de forma conjunta en una aplicació estàndard de OPC UA. Ara bé, cal destacar dues de les capes per la seva vital importància en el funcionament de OPC Unified Architecture: transport de les dades i seguretat.

2.3. Transport i accés a les dades

Les dades són la clau del sistema de comunicació OPC UA [14]. Ser capaç d'accedir a elles, llegir-les, modificar-les i reenviar-les, és la funció per la qual va néixer OPC. L'última especificació, OPC UA, permet visualitzar-les en temps real i veure els valors que tenien en el passat.

L'origen d'aquestes dades poden ser sensors, PLCs, microcontroladors, robots o els propis operaris de les plantes de producció. En un escenari industrial on els múltiples sensors i dispositius envien

constantment noves lectures i valors del seu estat, cal primer de tot modelar el sistema complet. S'assegura així que les diferents dades tenen el format corresponent per poder ser enviades/llegides.

L'estàndard OPC UA defineix unes normes i unes pautes a seguir per parametritzar el conjunt de dades de la planta i transformar-les en informació tangible i usable. Aquest marc de treball conforma un dels pilars fonamentals de la comunicació OPC UA. El protocol proporciona diferents models base que poden ser aplicats en escenaris industrials comuns. Al mateix temps, dona la possibilitat de que altres organitzacions o empreses agafin aquests models base i els modifiquin per adaptar-los a escenaris específics. D'aquesta manera s'aconsegueix una xarxa capaç d'entendre i traduir les diferents dades provinents dels diferents participants, tot en una infraestructura unificada.

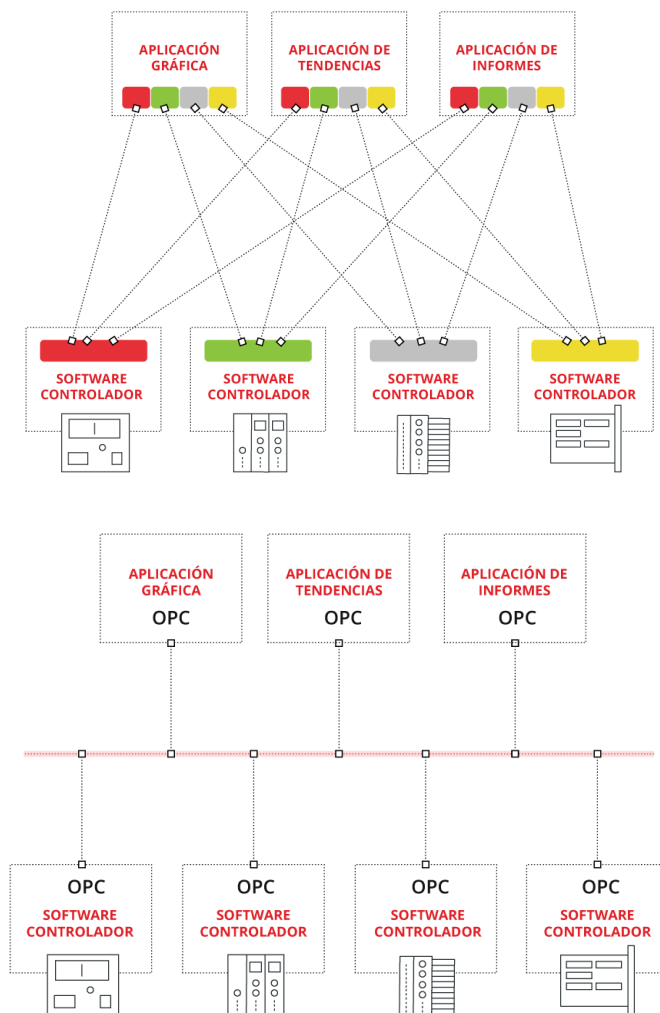


Figura 2.1. Xarxa funcionant sense i amb OPC (Font: incibe-cert)

Un cop es disposa de les dades degudament parametritzades, el següent pas correspon al transport de les mateixes a través de la xarxa OPC UA. Pel que fa al transport, OPC UA defineix dos mecanismes: UA TCP i SOAP/HTTP.

- **UA TCP:** protocol binari d'alt rendiment per comunicacions intranet. Aquest sistema permet tenir una comunicació ràpida i simple dins la xarxa. Se'n pot destacar la capacitat de reaccionar a partir d'errors que hagin tingut lloc durant el transport i recuperar-los quan calgui.
- **SOAP/HTTP:** aquest protocol dona accés a estàndards d'Internet àmpliament acceptats com HTTP. És una comunicació simple i compatible amb els elements de seguretat dels ordinadors.

Aquests mecanismes de transport es fan servir amb l'objectiu final d'establir una connexió entre un Servidor OPC UA i un Client a nivell de xarxa. El plantejament client-servidor és la base de la comunicació OPC UA, com ja ho era en les especificacions clàssiques.

El Servidor és l'encarregat d'encapsular la informació i el seu origen per després fer-la accessible a la xarxa. També recau sobre el Servidor la tasca del modelatge de les dades. Per la seva banda, el Client es connecta a un Servidor OPC UA i pot accedir llavors a les dades disponibles. És molt important destacar que les aplicacions que consumeixen i proporcionen dades poden ser alhora tant Client com Servidor. Aquesta afirmació s'ha corroborat al llarg del treball.

En el moment de connectar-se per primera vegada i en la finalització de la sessió, té lloc un intercanvi de missatges i acusaments de rebuda per garantir que es conforma la connexió desitjada i que aquesta es fa de manera segura.

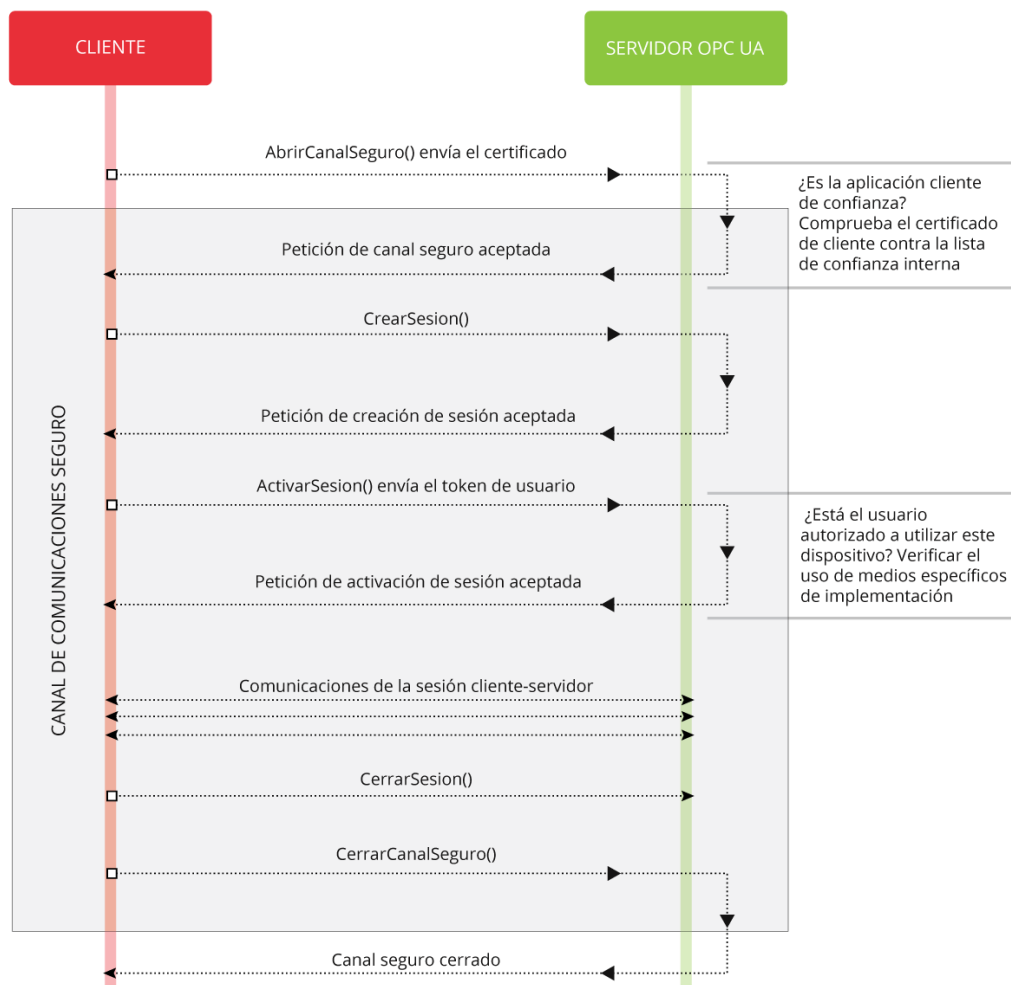


Figura 2.2. Flux de comunicació d’una xarxa OPC (Font: incibe-cert)

El flux normal de comunicació [15] d’una aplicació OPC UA incorpora certes eines per facilitar, per exemple, la detecció de dispositius dins la xarxa. Per exemple, en el cas dels Clients, compten amb una funció anomenada “Discovery” que permet localitzar Servidors dins la xarxa OPC UA sense requerir contacte previ. Una altra eina vital pel funcionament estàndard és les opcions de subscripció del Client: aquest es pot ‘subscriure’ a variables per fer el seu seguiment i visualitzar sempre el valor en temps real. D’altra banda es poden destacar les eines que faciliten la visualització de forma gràfica de les dades, que permeten la visualització de valors històrics.

La publicació de OPC UA ha vingut sempre acompanyada de la creació de programari per ordinadors per part d’empreses especialitzades. Aquestes empreses posen a l’abast dels usuaris de l’especificació OPC UA diversos Clients els quals es diferencien justament per l’èmfasi que fa cada un en les eines anteriorment comentades i l’aspecte visual. Aquests programaris poden actuar, en moltes ocasions, tant de Servidor com Clients, tot depèn del tipus d’aplicació que es vulgui aconseguir. En l’apartat 4.5

es detallaran els Clients que s'han utilitzat durant la fase d'investigació d'aquest treball i se'n destacaran els trets diferencials.

2.4. Seguretat

Anteriorment ja s'ha destacat la importància vital que té la seguretat com a atribut clau de OPC UA. En l'actualitat, les inversions en seguretat han esdevingut un requisit indispensable per moltes empreses independentment del sector on treballin. El sector de l'automatització no és cap excepció i des de fa dècades aposta de manera clara per incorporar sistemes de seguretat en tots els elements que participen a la cadena de producció. Això involucra des dels exportadors de matèries primes fins als distribuïdors del producte acabat, passant per tots els intermediaris del procés.

El focus d'aquest treball es centra en la xarxes de comunicació en escenaris on participen múltiples dispositius, com pot ser una planta de producció. Les indústries han d'estudiar atentament una sèrie de variables en el moment de triar la infraestructura que connectarà i comunicarà tots aquests dispositius. La seguretat és una d'aquestes consideracions, i s'imposa com a una variable essencial i de summa importància en el moment de triar una tecnologia envers una altra.

En aquest apartat es detalla els diferents paràmetres de seguretat que OPC Foundation va decidir incorporar en la seva última versió de OPC. D'altra banda, s'explicaran els atributs particulars que són configurables per l'usuari en el robot UR3e per tal d'entendre quin rol pot tenir la seguretat en el desplegament de OPC UA en un robot col·laboratiu.

2.4.1. Model de seguretat de OPC UA

El disseny i l'arquitectura del protocol OPC UA es van formalitzar des d'un principi per proporcionar una xarxa de comunicació completament segura. El primer pas cap a aquesta fita va ser dissenyar OPC UA per poder funcionar amb *firewalls* (o tallafocs) actius. Els tallafocs són la primera línia de defensa en ordinadors i altres dispositius per mantenir la seguretat d'una xarxa privada. Aquests programaris es deixen funcionant en segon pla i s'aconsella que estiguin sempre actius. Permetre a OPC UA poder funcionar conjuntament amb tallafocs va ser clau en la seva popularitat.

La seguretat de OPC UA es va abordar [14] amb una estructura basada en capes, on cada una d'elles disposa de responsabilitats específiques. Aquesta especialització va permetre, per una banda, un major grau de flexibilitat alhora d'implementar OPC Unified Architecture en escenaris amb diferents requisits

de seguretat. D'altra banda, garantia que tots els elements del flux de comunicació quedaven degudament coberts. Es va definir tres capes:

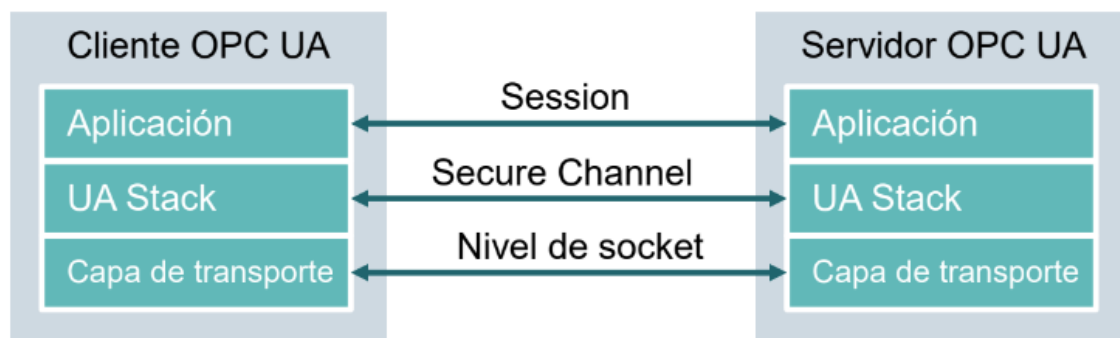


Figura 2.3. Capes de seguretat de OPC UA (Font: SIEMENS)

- **Capa d'Aplicació:** aquesta capa es fa servir per retransmetre informació de la planta, paràmetres, instruccions i altres dades en temps real dels diferents dispositius. En aquesta capa superior té lloc l'autenticació dels usuaris, ja sigui a través d'un usuari/contrasenya o dels certificats. L'autorització (o denegació) de la sessió també es situa en aquest estadi.
- **Capa de Comunicació:** aquesta capa és l'encarregada de mantenir la integritat dels missatges mitjançant signatures i altrament xifrar la informació sensible dels missatges retransmesos. Per altra banda s'introdueix el concepte d'autenticació d'aplicació per permetre a les aplicacions identificar-ne d'altres.
- **Capa de Transport:** la capa inferior compta amb la tasca d'enviar i rebre la informació segura a través de connexions *Socket* (un *socket* és un punt final d'un d'una comunicació bidireccional entre dos programes que s'executen dins una xarxa).

Aquestes tres capes treballen de manera conjunta per garantir la seguretat de tota la infraestructura OPC UA. A banda de les tres capes, intervenen altres paràmetres de seguretat en una aplicació estàndard OPC UA. Cada fabricant determina quins d'aquests paràmetres són directament editables per l'usuari. A continuació es llisten els set atributs [12] que estan presents en aquest protocol per disseny:

- **Transport:** es defineixen una sèrie de protocols de transport predeterminats, com OPC-binari (se'n destaca la rapidesa) o JSON (el qual treballa amb Sockets i és altament compatible).

- **Xifrat de la sessió:** es missatges es xifren en diferents nivells per retransmetre'ls de forma segura.
- **Signatura de missatges:** mitjançant la qual el recipient pot verificar l'origen i la integritat dels missatges rebuts.
- **Numeració dels paquets:** cada cop que s'envia un paquet d'informació, s'incrementa un comptador. El destinatari només acceptarà aquells missatges entrants que disposin d'un número major que el emmagatzemat. Quan s'accepta un missatge, el nombre emmagatzemat s'actualitza per evitar duplicats o intents d'atacs.
- **Autenticació:** totes les aplicacions OPC UA s'han d'acceptar mútuament, pas previ a formalitzar la comunicació. Cada Client i Servidor estan identificats amb certificats.
- **Control d'usuaris:** les aplicacions poden demanar l'autenticació dels usuaris mitjançant un usuari i una contrasenya, i addicionalment poden restringir l'accés a certes dades a només usuaris específics.
- **Auditoria:** cada acció dins una sessió OPC UA queda degudament registrada i se'n pot rastrejar l'autor en tot moment.

2.4.2. Posada en marxa d'una sessió OPC UA

Per tal de configurar un entorn OPC UA segur, cal incorporar les mesures de seguretat que s'han citat anteriorment. Aquesta configuració es realitza en el Servidor, i per formalitzar la comunicació el Client o Clients hauran de disposar d'exactament els mateixos paràmetres de seguretat que el Servidor.

De manera generalitzada, tots els fabricants de dispositius OPC UA inclouen la possibilitat de, com a mínim, editar els següents paràmetres [16] des de la pantalla de configuració del Servidor:

- Modes de seguretat pels missatges: permet seleccionar entre Cap / Signar / Signar i encriptar
- Polítiques de seguretat: les opcions disponibles són Cap / Basic128Rsa15 / Basic256 / Basic256Sha256
- Autenticació d'usuaris: es pot triar entre Anònim / Usuari i contrasenya

Destacar que aquestes opcions poden variar en funció del fabricant i de la iteració OPC UA utilitzada. D'altra banda, donat que els algorismes de seguretat s'actualitzen constantment per donar resposta a noves vulnerabilitats que es puguin detectar, pot ser que les opcions dins de la categoria Polítiques de seguretat es modifiquin amb el temps. Pel que fa a la creació dels usuaris, aquesta es realitza molts cops directament a la pantalla de configuració del Servidor, on s'assignen les credencials.

La figura 2.4 detalla les diferents opcions i explica el seu impacte en el xifrat o en l'autenticació d'usuaris.

Opció	Descripció
Security Policy	None: no se utilitza ningú tip de seguretat en el Secure Channel. Basic128Rsa15: juego de algoritmos de cifrado. Basic256: juego de algoritmos de cifrado ampliados.
Message Security Mode	None: los mensajes no se guardan en la copia de seguridad. Sign: los mensajes se firman. Sign&Encrypt: los mensajes se firman y cifran.
User Authentication	Anonymous: no se requiere autenticación de usuario. User Password: la autenticación de usuario se realiza a través de un nombre de usuario y una contraseña. Certificate: la autenticación de usuario se realiza a través de un certificado.

Figura 2.4. Mecanismes de seguretat en OPC UA (Font: SIEMENS)

Cal destacar que, tot i afegir sistemes de xifrat a la comunicació OPC UA, cada paquet de dades enviat manté una capçalera sense xifrar que incorpora dades sobre l'adreça IP i el port del dispositiu que ha enviat les dades. És una decisió que es va prendre en el moment de dissenyar l'estructura de seguretat del protocol i que es confirmarà en l'apartat 4.6 – Tests de seguretat.

Un cop s'han seleccionat els paràmetres de seguretat idèntics en el Servidor i el Client, s'inicia el procés de connexió. Aquest procés segueix una estructura pautada que continua en la línia de garantir un entorn segur.

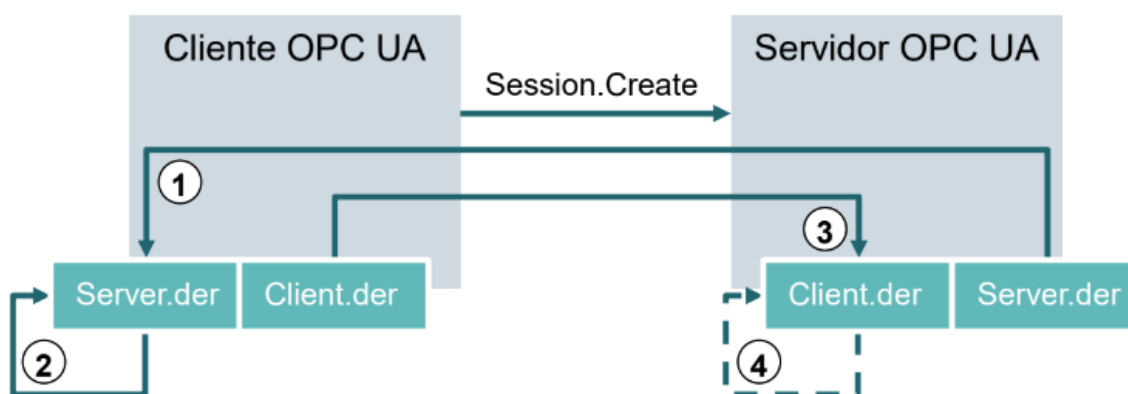


Figura 2.5. Flux de la primera etapa d'una comunicació OPC UA (Font: SIEMENS)

1. En el moment d'establir la connexió amb el Servidor, el Client rep el certificat del Servidor.

2. El programa que fa de Client pot decidir que fe amb el certificat: rebutjar-lo en cas que no es conegui la identitat, o acceptar-lo.
3. En el mateix procés, el Client envia alhora el seu certificat al Servidor. En un primer moment, el Servidor **rebutjarà** el certificat i l'emmagatzemarà a la carpeta de certificats denegats.
4. A continuació, cal que un administrador accepti manualment el certificat del Client en el Servidor. Només quan s'han acceptat els certificats per part del Client i del Servidor, s'oficialitza la comunicació.

2.4.3. Codi de bones pràctiques

El 18 de Juny del 2018, OPC Foundation publica un paper [17] on detalla les bones pràctiques i la importància de treballar amb els mètodes de seguretat que proporciona l'estàndard OPC UA. Aquest paper es publica pocs mesos després de la formació del grup de seguretat [18] de la OPC Foundation, en el qual recau la tasca d'analitzar, documentar i constatar que efectivament l'estàndard OPC UA és segur. Aquest grup ha publicat anualment des de llavors diversos informes i estudis on analitzen i detallen els diferents nivells de seguretat del protocol, tenint en compte les actualitzacions que ha rebut l'estàndard al llarg dels anys.

Aquesta recurrent auditoria interna denota el pes que té el concepte de seguretat per els membres de OPC Foundation i els seus clients. És un dels pilars de la comunicació OPC UA i sempre es tracta com a un element essencial i prioritari. No obstant, citant la informació disponible en el paper, es destaca que tot i ser un protocol segur per disseny, són els propis usuaris els que han d'implementar i utilitzar els paràmetres de seguretat disponibles si volen tenir un entorn realment segur.

En el seu afany de facilitar la configuració i l'entesa dels diferents modes de seguretat, han publicat també guies on recomanen als usuaris l'ús de determinades característiques OPC UA essencials per un funcionament segur.

Així doncs, citant la informació disponible a l'escrit [17], un entorn que treballi amb OPC UA hauria d'incloure els següents paràmetres de seguretat:

- Mode de Seguretat: D'acord amb l'article, el mode de seguretat OPC UA "hauria de ser 'Signar' o 'Signar-i-encriptar'. Així s'assegura que es força autenticació a nivell de l'aplicació. S'ha d'utilitzar el mode 'Signar-i-encriptar' si es vol protegir la integritat i confidencialitat de les dades."
- Selecció de algorismes criptogràfics: "Com a mínim la política de seguretat 'Basic256Sha256' hauria de ser seleccionada [...] en cas que tant servidor com client suportin aquesta política.

No es recomana l'ús de polítiques de seguretat més febles o que estiguin antiquades, com per exemple 'SHA-1'."

- Autenticació d'usuari: l'escrit adverteix que l'identificador 'Anonymous' hauria de ser utilitzat "només per accedir recursos del servidor OPC UA que no siguin crítics, ja que no proporciona cap protecció (s'anul·la la possibilitat de rastrejar qui ha modificat les dades o la configuració quan es selecciona aquest identificador genèric). D'altra banda, un atacant podria utilitzar aquest identificador per llegir o escriure dades sense autorització si no es limiten els drets d'aquest usuari anònim."
- Certificat i emmagatzematge de clau privada: "Mai s'han de guardar claus privades o el corresponent certificat en un sistema d'arxius no encriptat. Es recomana utilitzar les funcions dedicades de cada sistema operatiu per guardar correctament els certificats i restringir-ne l'accés."
- Ús de certificats: Segons indica l'escrit, no s'haurien d'acceptar mai connexions que no proporcionin certificats de confiança. El document també subratlla que "els certificats d'auto-signa no s'haurien de donar per vàlids automàticament (és a dir, sense una verificació addicional). Tots els certificats han de ser o bé d'auto-signa o signats per una CA (Autoritat Certificada) autoritzada."
- Administrar i mantenir certificats: "Es recomana fer ús de les llistes de confiança i la possibilitat de revocar certificats per administrar els certificats i tenir-los al dia. Només els usuaris amb permisos i processos de confiança haurien de poder modificar aquestes llistes."

3. Robot UR3e

Juntament amb el protocol OPC UA, el model de robot col·laboratiu UR3e de l'empresa Universal Robots és el segon pilar sobre el qual es desenvolupa aquest treball acadèmic.

Hi ha dos motius clars que expliquen el perquè de la tria d'aquest model en particular. En primer lloc, es va apostar per el robot UR3e ja que es va confirmar la possibilitat d'habilitar les funcionalitats OPC UA mitjançant un programari extern. És a dir, una empresa externa comercialitza un programari específic per models de UR que habilita la comunicació OPC UA de manera relativament fàcil.

El segon motiu va ser, principalment, per un tema de disponibilitat. El laboratori A5.4 del campus d'Enginyeria Barcelona Est [19] disposa, a dia d'avui, de quatre models diferents de robots industrials. Ara bé, només un d'ells forma part de la categoria de cobots (robots col·laboratius), justament el UR3e. Aquest treball es va idear per investigar les funcionalitats OPC UA en robots col·laboratius, i doncs, el UR3e era l'únic cobot disponible en el laboratori de robòtica.



Figura 3.1. Robots disponibles al laboratori A5.4 de l'EEBE (Font: pròpia)

En cap cas es va veure aquesta tria com una limitació, més aviat al contrari. Dins el mercat de robots col·laboratius, Universal Robots és un dels màxims exponents d'aquesta tecnologia i distribueix models arreu del planeta. Compta amb varies comunitats de suport online i una oficina directament a

Barcelona, propera a la universitat. Tot plegat feia evident la tria d'aquest model en particular per documentar el desplegament del protocol OPC UA dins el rang de temps disponible.

En els propers subapartats es detallen les característiques principals del robot, s'expliquen les possibles aplicacions que permet i finalment es detalla el programari extern que habilita OPC UA en el *cobot*.

3.1. Introducció al model UR3e

3.1.1. Universal Robots

Universal Robots [20] (UR) és el fabricant i distribuïdor del model UR3e [21]. Aquesta empresa danesa amb múltiples oficines arreu del món, es va fundar l'any 2005 amb l'objectiu de fer la robòtica més accessible per a petites i mitjanes empreses. El 2008 [22] ven el primer robot col·laboratiu, el model UE5. Disposava d'una programació 3D fàcil d'utilitzar però al mateix temps sofisticada, també comptava amb una interfície d'usuari intuïtiva facilitant la configuració i una posada en marxa ràpida. El llançament d'aquest model va tenir un impacte important en el sector de la robòtica i va posicionar Universal Robots al mapa.

Quatre anys més tard presenten el UR10, amb més capacitat de càrrega i rang que el primer model. El 2015 arribaria l'UR3, el més petit dels robots de UR però alhora el més econòmic i accessible per empreses de diferents dimensions.

L'any 2018, amb tota l'experiència recollida durant els anteriors deu anys, UR presenta la sèrie "e". Aquesta "e" representa, citant la informació de la seva web [20], evolució/empoderament/'easy to use' (facilitat d'ús).

Entre les millores d'aquesta nova sèrie es troben una millora de la precisió, de l'experiència d'usuari amb un taulell de control redissenyat i nou programari intern del robot per facilitar la programació. El nou entorn on s'interactua amb el robot va rebre el nom de PolyScope. Ara bé, la novetat més important va ser, sens dubte, la incorporació d'un sensor de parell/força a l'extrem del braç. Aquest permetia al robot detectar xocs i objectes inesperats en l'itinerari augmentant considerablement la seguretat. Aquesta millora recolzava l'idea de la marca per comercialitzar robots que poguessin treballar conjuntament amb persones.

Actualment, Universal Robots compta amb cinc models diferents: UR3e, UR5e, UR10e, UR16e i el recentment presentat UR20. Les diferències principals corresponen en aspectes tals com dimensions, capacitat de càrrega i abast. UR segueix donant suport als models antics.

Part de l'èxit de l'empresa danesa recau en el seu ecosistema de productes que amplien les possibilitats dels robots UR. Anomenat UR+ [23], aquest ecosistema ofereix accés a més de quatre-cents kits, components, pinces, programari i accessoris de seguretat certificats que s'integren perfectament amb els cobots de la marca. Alguns d'aquests components del ventall UR+ requereixen la instal·lació de programaris externs en el sistema operatiu del robot. Els distribuïdors, integradors i fabricants poden proporcionar accés a aquests programaris mitjançant la plataforma URCaps [24], la qual engloba productes certificats que es poden comprar per separat, tant programaris com productes físics. Aquests accessoris amplien les capacitats dels robots i habiliten funcions que no es troben en la versió bàsica del robot. L'accés directe a URCaps dins l'entorn del robot es troba dins l'apartat 'Instal·lació'.

3.1.2. Característiques tècniques del robot

El model seleccionat per desenvolupar aquest treball d'investigació ha estat el UR3e [21]. Correspon al braç robòtic més petit dels que comercialitza UR. Disposa d'un disseny compacte que facilita el seu ús en escenaris amb espai reduït. Pesa 11.2kg i disposa d'una capacitat per carregar objectes de fins a 3kg. El seu braç abasta com a màxim 500mm.

Aquest robot s'inclou dins la categoria de robots de sis graus de llibertat, és a dir que compta amb mobilitat en sis punts diferents: base, ombro, colze i tres possibles rotacions de canells. Cada una d'aquestes sis articulacions disposa de 360° de moviment lliure en els dos sentits, i addicionalment, l'element terminal col·locat a l'última articulació pot rotar indefinidament.

El seu disseny compacte el fa idoni per tasques d'assemblatge simples i també per a escenaris automatitzats en bancs de treball. El seu sistema de seguretat ha estat certificat per TÜV [25], l'Associació alemanya d'Inspecció Tècnica, i permet al personal de planta treballar conjuntament amb aquest robot.

El seu procés d'instal·lació és simple i relativament ràpid. El robot inclou un programari intuïtiu i compta amb nombrosos cursos gratuïts per aprendre a programar-lo. Aquestes característiques li atorguen un elevat grau de flexibilitat, donat que pot desenvolupar moltes feines diferents, ja sigui un "pick&place" (programes basats en recollir coses d'un lloc i col·locar-les en un altre), assemblatge de peces, soldadura etc.

Depenent de l'element terminal instal·lat, se li pot donar al robot la capacitat de realitzar tasques de poliment, encolatge, cargolat i muntatge de peces. En el robot del laboratori es troba instal·lada una pinça elèctrica de dos dits, la qual és el resultat del TFE d'un estudiant de la EBBE [26]. Aquesta pinça elèctrica va estar ideada per realitzar tasques d'assemblatge i "pick&place". Donat que la pinça permet

realitzar diverses tasques robòtiques de manera satisfactòria i es va confirmar el seu bon funcionament, s'aprofitarà per la part final d'aquest treball.



Figura 3.2. Robot UR3e del laboratori amb la pinça instal·lada (Font: Autor)

El robot incorpora un controlador que s'instal·la a part i es connecta amb el robot a través d'un cable. En el controlador es troben disponibles les connexions d'entrades i sortides per a que el robot pugui interactuar amb sensors, llums, cintes i altres màquines. Finalment cal destacar que el model disponible al laboratori compta amb una tauleta de control tàctil, coneguda com a "Teach Pendant" que actua com a pont entre l'operari i el robot, i permet, entre moltes altres funcions, controlar-lo i programar-lo.

Cal destacar que a nivell de connexions, cablejat i muntatge no es tocarà cap part del maquinari instal·lat. És a dir, es farà ús de la configuració present al laboratori. Els canvis tindran lloc en l'apartat de programari, que és el focus d'aquest treball.

En els annexes d'aquest treball s'adjunten les especificacions tècniques completes del robot i del controlador proporcionades per UR [27].

3.2. Comunicacions compatibles amb el robot

Els robots UR estan pensats per realitzar un ventall de tasques repetitives en cicle i de manera local. És possible, però, habilitar opcions de comunicació que enviïn dades a un dispositiu extern per habilitar funcions de control del robot i monitoratge de variables.

Universal Robots dona suports a certs protocols de comunicació. Quan un model d'UR surt de fàbrica té habilitat el suport per Modbus, Profinet i Ethernet/IP. Per introduir nous protocols d'intercanvi d'informació com MQTT o OPC UA cal fer ús dels URCaps. Les diferents comunicacions tenen diferències notables pel que fa als usos, modes de seguretat i en el seu funcionament. En els següents apartats es detallen de manera general les característiques diferencials de cada una.

3.2.1. MODBUS TCP

Universal Robots instal·la de forma predeterminada el protocol Modbus TCP en els seus robots. Aquest tipus de comunicació permet al robot comunicar-se amb altres robots, PLCs, HMI i altres dispositius. Es basa en la tecnologia Ethernet TCP/IP i funciona amb una distribució Servidor-Client. Amb el protocol Modbus TCP [28] el servidor esdevé un element passiu que es dedica a escoltar fins a rebre alguna comanda del client. Aquest últim consumeix més recursos ja que és el que sol·licita informació i inicia les comunicacions. Amb aquest protocol els robots poden ser tant servidors com clients.

La configuració d'aquest protocol es realitza directament dins la interfície del robot en un apartat específic a la categoria "Instalation", dins de 'Fieldbus'.



Figura 3.3. Configuració de MODBUS dins de PolyScope (Font: UR)

Un cop s'ha configurat Modbus i les entrades/sortides corresponents, la interfície de programació PolyScope genera noves funcions accessibles per l'usuari que interactuen directament amb el programa del robot. Amb aquestes noves línies de codi es pot aconseguir que el robot romangui en espera fins que una certa entrada tingui el valor desitjat, que en determinats punts del programa es configuren en OFF/ON les sortides Modbus o assignar a variables locals del programa el valor que tingui alguna sortida/entrada.

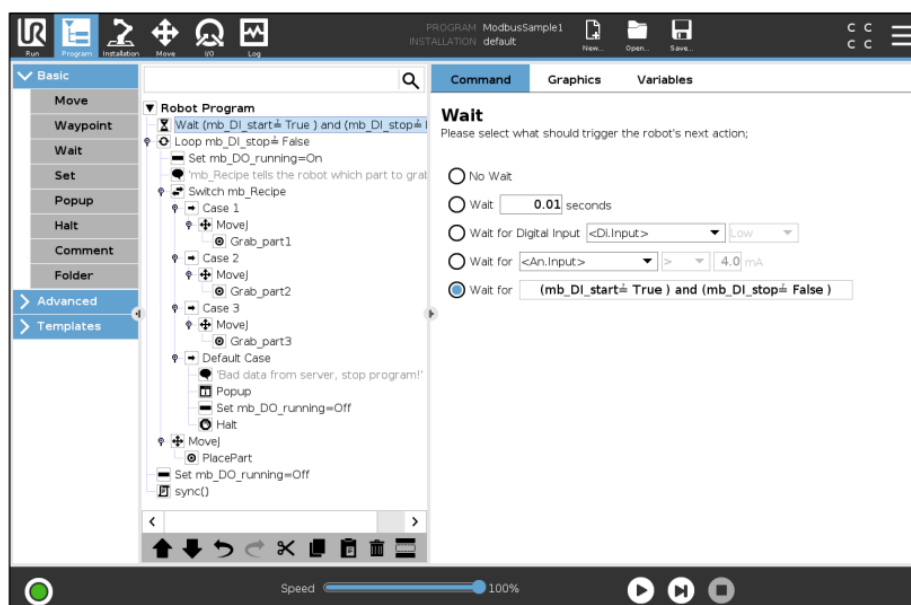


Figura 3.4. Programa funcionant amb comandes MODBUS (Font: UR)

Modbus TCP ofereix doncs les possibilitats de comunicar-se amb elements externs i controlar el programa del robot sempre i quan la transmissió d'informació es realitzi amb dispositius que suportin aquest protocol i es trobin a la mateixa xarxa.

Cal destacar per una banda que el protocol Modbus TCP està més encarat a comunicar maquinari a nivell de planta. Per altra banda, en referència a la seguretat, té dos grans debilitats [29] comparat amb altres alternatives del mercat: la seva arquitectura no incorpora xifrat de les comunicacions, habilitant així que terceres persones puguin descobrir el número d'elements de la xarxa i els seus identificadors. La segona debilitat s'aprofita de la primera, ja que un cop es saben els clients/servidors i les identificacions corresponents, poden tenir lloc casos de suplantacions d'identitat. En altres paraules, un element extern es pot fer passar per un dels elements de la xarxa i proporcionar valors erronis i fins i tot donar ordres.

Amb aquestes vulnerabilitats en ment, a finals de l'any 2018 es presenta MODBUS/TCP Security [30], una alternativa a l'especificació original que es centra en millorar les característiques de seguretat del protocol. Aquesta nova versió introdueix importants canvis referents a la capa de transport (amb aquesta nova iteració l'intercanvi d'informació té lloc dins un canal xifrat), s'imposa l'ús de certificats per confirmar la identitat dels elements de la xarxa i, finalment, es substitueix els termes màster/esclau per els seus anàlegs informàtics, client/servidor.

Tot i les importants novetats, com ha passat sempre amb la introducció de nous protocols en la indústria, el seu desplegament ha estat més aviat lent i falta veure quants anys han de passar per confirmar si aquesta "nova" especificació s'ha establert oficialment en el sector. UR ha decidit no implementar aquesta última versió del protocol MODBUS/TCP, així doncs, la versió disponible al robot es considera no segura [29].

3.2.2. PROFINET

Profinet és l'estàndard obert d'Ethernet Industrial presentat el 2003 per PROFIBUS International [31]. Ha esdevingut un dels estàndards de comunicació més utilitzats en xarxes d'automatització. Aquest protocol està ideat per controlar l'intercanvi de dades entre màquines i dispositius a nivell de camp. Se'n pot destacar sobretot la rapidesa de les interaccions.

La implementació dins l'ecosistema de robots UR s'enfoca a l'intercanvi de dades entre el robot i PLCs fent ús de programes com TIA Portal (Siemens). Dins l'entorn PolyScope podem habilitar [32] PROFINET des de l'apartat 'Instal·lació'. Un cop realitzades les configuracions pertinents, similar a MODBUS, es proporciona a l'usuari una sèrie de comandes noves, o 'scripts', que es poden afegir a la programació tradicional.

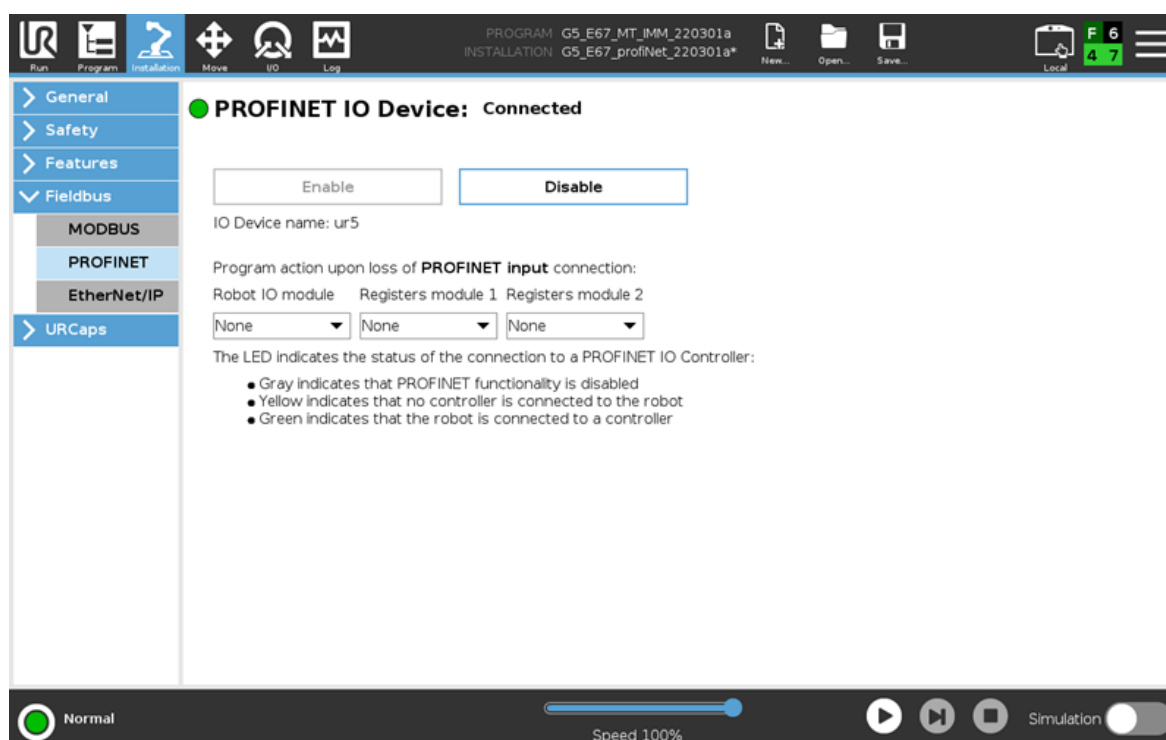


Figura 3.5. Configuració de PROFINET directament des de PolyScope (Font: UR)

PROFINET funcionant en un robot UR pot, per exemple, llegir els valors dels dispositius connectats al PLC configurat i assignar aquest valor a variables del programa. De la mateixa manera, pot assignar un cert valor a una sortida del PLC, tot sense sortir de la programació del PolyScope.

En termes de seguretat [33], les versions més recents de PROFINET inclouen l'ús de certificats que assegurin l'autenticació dels diferents participants. Per altra banda es dona suport a l'autenticació dels missatges mitjançant criptografia. Aquests atributs juntament amb unes pautes de seguretat publicades per PROFIBUS International [34], conformen les defenses més aviat bàsiques de l'estàndard. Tot i els intents per posar-se al dia en matèria de seguretat, no és un dels protocols més segurs.

3.2.3. ETHERNET/IP

Ethernet/IP forma part de CIP (Protocol Industrial Comú), el qual és un protocol de comunicació per transferir dades d'automatització entre dos dispositius. La implementació del protocol CIP sobre Ethernet rep el nom de Ethernet/IP. Similar a PROFINET, Ethernet/IP busca flexibilitat i rapidesa alhora d'interconnectar tots els dispositius Ethernet. Utilitza la tecnologia TCP/IP i anomena Escàner al Client

i Adaptador al Servidor. El seu ús principal és moure dades entre dispositius E/S (entrades/sortides) i PLCs a nivell de camp.

La implementació en l'entorn PolyScope de UR és molt semblant a PROFINET, oferint unes comandes extres que permeten llegir/escriure els valors dels dispositius connectats a la xarxa [35]. La seguretat de l'estàndard Ethernet/IP [36] es basa en l'ús de certificats per garantir la identitat dels equips que intervenen en la comunicació. Altrament, s'assegura l'autenticació dels paquets enviats mitjançant eines que confirmen que aquests no han estat modificats durant el transport. Opcionalment es pot introduir xifrat a les comunicacions, però aquesta capa extra de seguretat pot comportar retards addicionals que no són acceptables en totes les aplicacions. Es recomana l'ús de connexions VPN (Xarxes virtuals privades) per dotar de més seguretat la xarxa.

Destacar que Ethernet/IP és la xarxa de comunicacions física present en els dispositius del laboratori A4.5 i serà sobre aquesta xarxa on s'implementarà OPC UA.

3.2.4. MQTT

Universal Robots dona suport a la funcionalitat MQTT. Mitjançant un programari extern [37] de pagament instal·lat a la secció URCaps, es pot habilitar les funcions MQTT. Aquestes permeten un intercanvi de missatges (dades) basat en publicació/subscripció a elements externs. Se'n pot destacar l'escalabilitat i la flexibilitat de les seves xarxes. Amb aquesta comunicació es permet implementar noves solucions de la Indústria 4.0 de manera fàcil (*IoT*, núvol, *BigData* etc.)

És una comunicació molt eficient i amb un impacte relativament baix tant per la xarxa d'internet com per els processadors dels dispositius que intervenen. La comunicació és bidireccional, és a dir, que aquesta pot ser iniciada per qualsevol element de la xarxa MQTT. El model seguit per MQTT és el següent: els clients envien les dades a un servidor central. Mitjançant la publicació, aquest servidor distribueix les dades exclusivament als clients que ho hagin demanat quan es detecti un canvi en el valor de les dades, minimitzant així els recursos utilitzats.

Amb el programari extern, es pot xifrar les dades enviades i la versió professional disposa d'extenses opcions de personalització. Permet introduir autenticació amb usuari/contrasenya. Com en els anteriors casos, el programari habilita unes línies de codi específiques per treballar amb MQTT en la programació del PolyScope.

És una de les opcions de comunicació més modernes i amb més suport de cara al futur. No obstant, s'enfoca més en el sector IoT i la seva tecnologia es limita a transmetre/rebre informació. És una eina molt potent per tenir un rol de suport.

3.2.5. OPC UA

El model UR3e compta amb suport del protocol OPC UA mitjançant un programari extern de pagament desenvolupat per Rocketfarm. La implementació de l'estàndard en els robot UR compta amb la majoria d'eines bàsiques de OPC UA, àmplies opcions de seguretat i compatibilitat amb múltiples clients de Windows.

Aquesta és la comunicació al voltant de la qual es desenvolupa el treball.

3.3. URCap OPC UA

Els robots d'Universal Robots no suporten OPC UA de manera nativa. A diferència de MODBUS o PROFINET, cal comprar i instal·lar un programari extern per habilitar aquesta comunicació. A dia d'avui, l'única empresa que comercialitza aquest programari és Rocketfarm [38], una empresa noruega que desenvolupa programaris específics pels robots UR. El cost oficial d'aquest programari és de 1490€ per una llicència indefinida. Rocketfarm et proporciona una guia bàsica [39] per dur a terme la instal·lació i un resum de la posada en marxa. Aquesta ha estat l'única ajuda per a la configuració de l'entorn OPC UA.

La instal·lació del programari es va fer seguint les instruccions proporcionades i les llicències que Rocketfarm va enviar al campus EEBE. Aquest treball acadèmic va començar just després d'haver-se instal·lat el URCap OPC UA. Aquest programari està basat en la versió OPC UA de codi obert open62541.

Cal destacar que durant la realització d'aquest treball d'investigació, Rocketfarm ha publicat una tercera versió del URCap OPC UA, anomenada OPC UA VDMA [40]. La versió 3.0 conté totes les parts de la segona però amplia algunes funcionalitats donant suport a 'OPC VDMA 40010 Robotics'. A grans trets, aquesta especificació publica de manera automàtica sempre i quan estigui el servidor encès, una sèrie de paràmetres tècnics del robot:

- Fabricant, model, categoria del robot, número de sèrie i velocitat.
- Posició i velocitat de cada articulació.
- Model del motor de cada articulació, número de sèrie del motor i temperatura del mateix.
- Informació sobre el programari actual del robot i la seva versió.
- Nom del programa i la seva direcció en la memòria del robot.
- Estat dels botons de parada d'emergència.

4. Configuració i ús de l'entorn OPC UA

En aquest apartat del treball s'exposen el seguit de proves i tests realitzats amb la finalitat d'entendre el funcionament de la comunicació OPC UA. Paral·lelament s'ha documentat en detall cada pas realitzat per tal de tenir unes pautes escrites sobre la posada en marxa d'aquest protocol en robots col·laboratius.

Com a 'entorn OPC UA' s'entén tota aquella infraestructura necessària per fer funcionar la tecnologia OPC UA: el robot UR3e amb el programari URCap, els Clients instal·lats al PC i la infraestructura Ethernet/IP que connecta físicament els dos dispositius mitjançant cables amb connectors RJ45. En aquest entorn conviuen altres dispositius com PLCs, passarel·les de comunicacions (Ewon IoT) i altres robots. En el cas del laboratori A5.4 es treballa amb una connexió en estrella on al centre hi trobem un *switch* de Cisco, el qual actua de pont entre tots els dispositius.

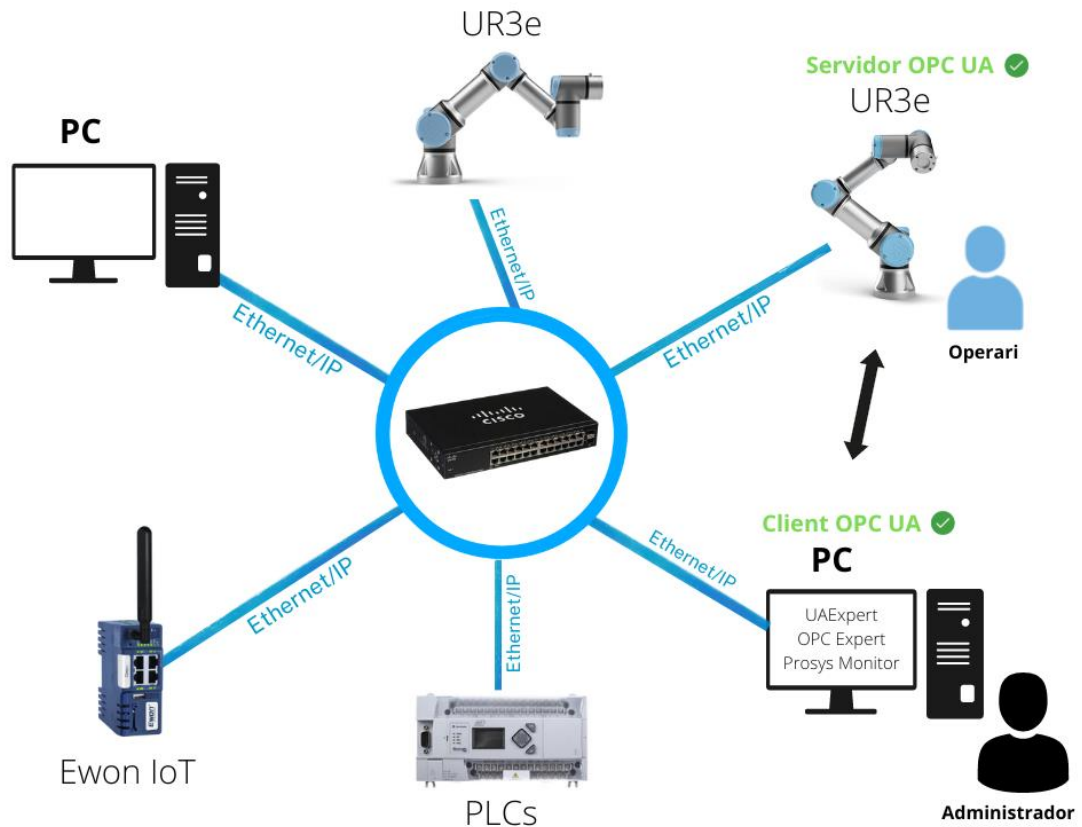


Figura 4.1. Esquema de la xarxa de comunicacions del laboratori (Font: Autor)

La metodologia seguida per entendre els diferents paràmetres modificables i el seu impacte en el funcionament de l'entorn OPC UA ha estat una metodologia prova-error. Donat el temps disponible i

els coneixements tècnics limitats a l'inici del projecte, es va creure adient invertir el temps suficient en cada element configurable i provar varies combinacions, encara que els resultats no sempre fossin satisfactoris. Aquestes proves inclourien tests en format d'aplicacions robòtiques reals per acabar de confirmar les possibilitats d'ús de cada atribut de la comunicació OPC UA. La investigació aniria acompanyada d'una detallada documentació de les proves realitzades, l'escenari específic testejat i les conclusions extretes.

Com a últim punt de la introducció, destacar que la majoria de punts recollits en aquest apartat estan redactats en format de guia. D'aquesta manera es segueix un ordre cronològic dels passos, facilitant el seu enteniment, i s'adequa el format a un dels objectius del treball, que és justament la realització d'una guia.

4.1. Requeriments previs

Per la configuració inicial es farà ús, per una banda, del robot UR3e i el mòdul URCaps OPC UA. D'altra banda s'utilitzarà un programari de Windows anomenat UA Expert que permet testejar i fer projectes dins entorns OPC de manera gratuïta.

El robot UR3e disposa d'una adreça IP i té connexió Ethernet amb el "router" (encaminador) del laboratori. Es pot confirmar que, al PolyScope, dins del menú a l'extrem dret de la pantalla, Ajustes > Sistema > Red, la xarxa està connectada i s'està utilitzant la IP "130.130.130.51".

Depenent de l'ordinador de sobretaula del laboratori que utilitzem, pot ser necessari connectar a la corrent el PLC Compact Logix del banc de treball, ja que aquest element habilita la connexió a la xarxa de l'ordinador. L'ordinador utilitzat per la majoria de les proves és el que disposa de l'adreça IP 130.130.130.11. Es conclou, doncs, que tan robot com ordinador han d'estar a la mateixa xarxa LAN.

Per acabar, remarcar que els motors no cal que estiguin encesos per a fer funcionar el Servidor, tot i que si es vol moure el braç robòtic sí que cal realitzar la configuració inicial en el PolyScope.

4.2. Configuració del Servidor

En l'apartat '2.3 – Transport de dades' ja s'ha destacat que tots els participants d'una xarxa OPC UA poden funcionar tant de Servidor com de Client. No obstant, ha d'existir almenys un Servidor en el qual

es connectin el o els Clients. En tots els casos detallats en aquest treball acadèmic, el rol del Servidor s'assigna al robot UR3e, és a dir, que es configura el Servidor directament des del robot.

El rol del Servidor serà iniciar/finalitzar la comunicació, permetre la configuració dels paràmetres de seguretat i finalment, l'edició de les variables que s'enviaran dins la xarxa OPC UA.

Un cop s'ha engegat el robot i ha passat la fase inicial, cal activar el Servidor. Per fer-ho hem d'anar a Instalación > URCaps > OPC UA. Aquí podem trobar la configuració d'aquesta versió d'OPC que ha implementat Rocketfarm. En pantalla podem veure quatre categories: Client / Server / Daemon / License.

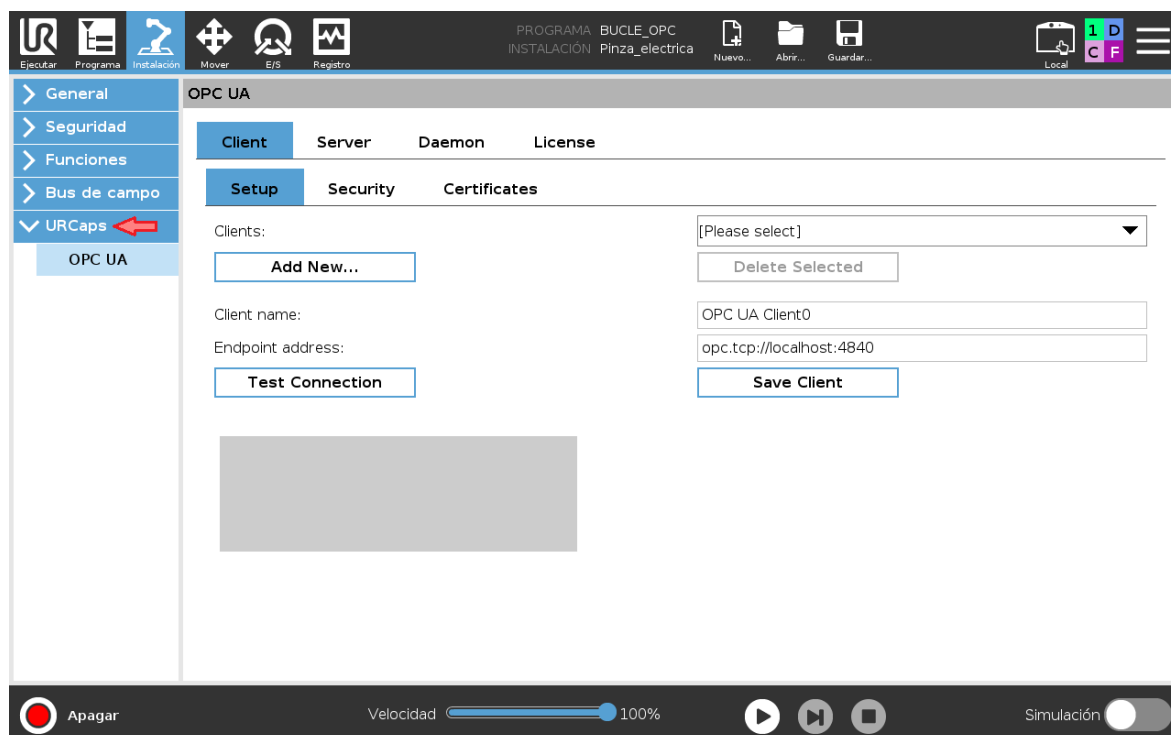


Figura 4.2. Accés al URCap OPC UA (Font: Autor)

Per aquest apartat ens interessa accedir a Server. El primer subapartat dins de Servidor s'anomena "Endpoints" i es permet iniciar/parar el servidor i editar-ne el port. El port és un dels mètodes que té aquest protocol de comunicació per confirmar que ens connectem al servidor que toca. El port per defecte és 4840.

Cal clicar sobre Start Server per a iniciar-lo (si Start no està disponible, primer clicar sobre Stop). Un indicador verd s'encén per informar que el servidor està funcionant.

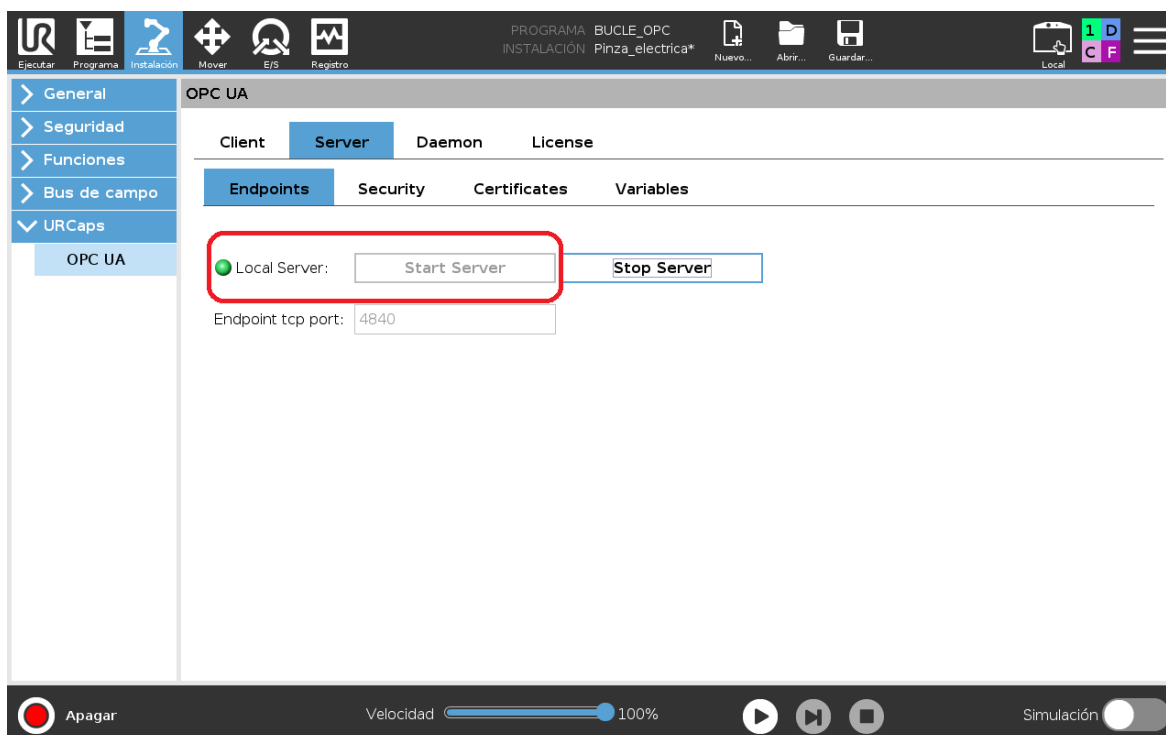


Figura 4.3. Pantalla per encendre/apagar el Servidor OPC UA (Font: Autor)

El següent subapartat dins de Server és “Security”. Aquest apartat ens deixa triar quin és el tipus de seguretat que volem per el nostre entorn OPC UA. Sempre haurem de triar un tipus de seguretat i un mètode d'autenticació. En escenaris acadèmics i per tal de simplificar els passos de la configuració es pot seleccionar Security Mode > None i User Auth > Anonymous. Aquesta configuració, com es veurà més endavant, no proporciona cap protecció i es recomana únicament per a un context docent.

Opcionalment es pot confirmar que en l'apartat Daemon l'indicador de OPC UA està verd (actiu).

4.3. Variables OPC UA

Les variables són els elements principals que s'envien a través de la xarxa OPC UA. En aquest apartat s'explica en detall la seva creació i configuració per una sèrie d'escenaris pràctics, els quals haurien de poder generalitzar-se per permetre a qualsevol lector d'aquest treball entendre com crear les variables necessàries per la seva aplicació OPC UA específica.

El programari OPC UA present en el robot posa a la nostra disposició l'opció de crear variables des de zero, les quals seran les que es podran escriure/llegir dins la xarxa. Per trobar la finestra on crear i editar les variables cal anar a Instalation > URCaps > OPC UA > Server. A la barra superior ens apareixerà la categoria de 'Variables'.

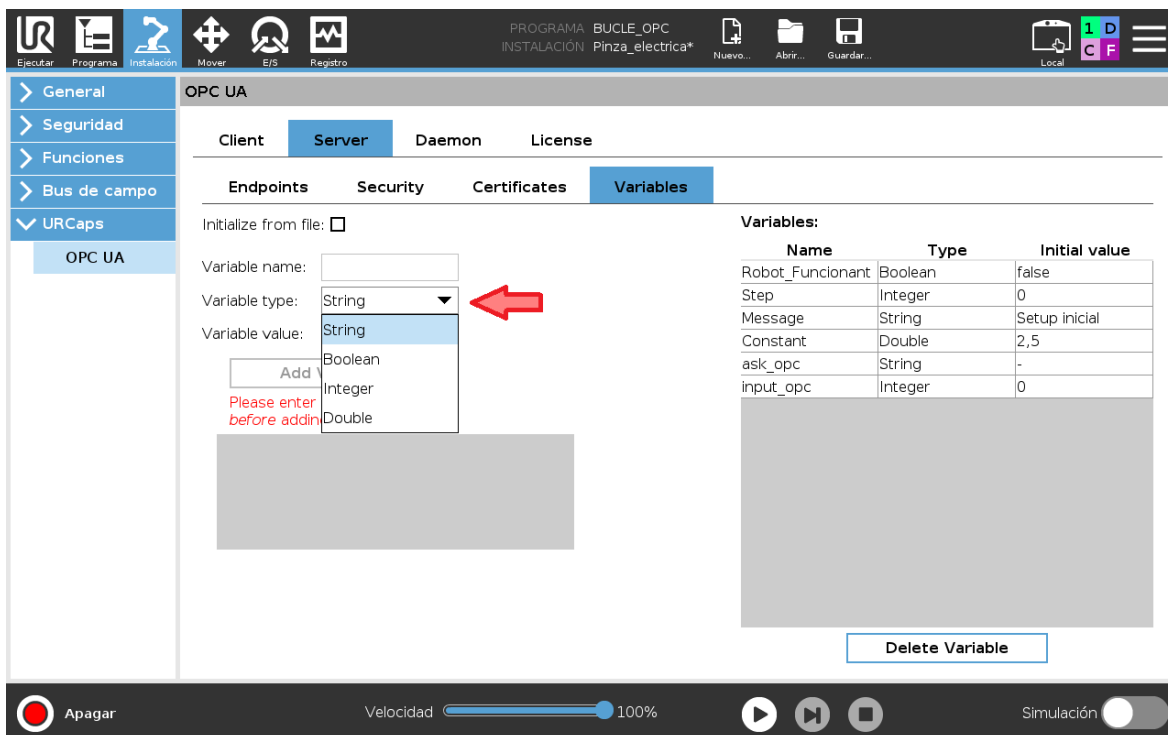


Figura 4.4. Pantalla de creaci3 de variables OPC UA (Font: Autor)

Aquesta p3gina ens permet crear variables, editar-ne el nom, el valor inicial i el tipus. Disposem de quatre tipus diferents (amb la versi3 2.0 del programari URCap OPC UA):

- **String:** ens permet imposar com a valor tant lletres com car3cters num3rics. Aquest tipus 3s 3til quan interressi enviar missatges o informaci3 amb molts car3cters.
- **Boolean:** variable de tipus Vertader/Fals. Les variables booleanes indiquen normalment dos estats (funcionant, rep3s, enc3s, apagat, etc.).
- **Integer:** el valor d'aquesta variable ser3 sempre un n3mero enter. Ideal per aquells casos on la variable no inclogui decimals.
- **Double:** permet posar un decimal en n3meros positius i negatius. Cal seleccionar aquest tipus si la variable t3 decimals, per exemple en temporitzadors o lectures que requereixin precisi3.

La creaci3 i l'assignaci3 del valor inicial d'aquestes variables es pot realitzar 3nicament si el Servidor est3 apagat.

3s important destacar que aquestes variables s3n complement3ries a les variables locals que l'usuari pugui tenir pel programa principal del robot. Per tant, dins la comunicaci3 OPC UA treballarem 3nicament amb les variables creades dins la pestanya Servidor de URCaps. Aquesta 3s una de les limitacions presents en la implementaci3 del protocol OPC UA feta per Rocketfarm.

4.4. Programació específica de OPC UA

L'URCap de Rocketfarm que s'ha afegit al robot també ens proporciona dos nous blocs de codi per utilitzar dins la programació gràfica del cobot UR. Els trobarem a Programa > URCaps. Aquests dos blocs de codi representen la manera més fàcil i intuïtiva d'introduir funcions OPC UA en un programa estàndard del robot. Addicionalment, es pot interactuar amb la xarxa OPC UA mitjançant la programació amb 'scripts', però en la totalitat d'aquest treball es treballa amb la programació per blocs.

Les funcions de l'OPC UA URCap es divideixen en OPC UA Client i OPC UA Servidor. Tant les funcions de client com de servidor es poden utilitzar simultàniament en el mateix programa:

- **OPC UA Client:** les funcions de Client s'utilitzen per connectar-nos a qualsevol servidor OPC UA remot i intercanviar dades amb l'entorn OPC. Si cliquem sobre aquest bloc, s'afegeix al programa principal una línia de codi que disposa de quatre funcionalitats:
 - Connect (ens permet iniciar la comunicació amb un Servidor remot, sent el robot el client)
 - Read (ens permet assignar el valor d'una variable del Servidor remot a una de les variables locals del programa)
 - Write (serveix per assignar el valor d'una de les variables locals a una variable del servidor remot)
 - Disconnect (ens desconnecta del servidor remot)
- **OPC UA Server:** Per si volem treballar amb la configuració on el robot és el Servidor i es vol proporcionar dades a clients OPC UA remots. Si premem sobre aquesta opció s'afegeix una línia de codi amb dos accions:
 - Read (ens permet assignar el valor d'una variable de l'entorn OPC a una de les variables locals del programa)
 - Write (serveix per assignar el valor d'una de les variables locals a una variable del servidor remot)

Les dues accions dins de Server, tenen una funció addicional que permet editar com actua el programa en cas d'error per si no s'aconsegueix llegir/escriure. Podem no fer res, reintentar o finalitzar el programa.

S'ha de destacar una idea important que apareixerà múltiples vegades al llarg del treball: aquests blocs de programació 'Read/Write' són els encarregats de relacionar les variables OPC amb les variables locals del programa. En altres paraules, la limitació més gran del URCap OPC UA és que el robot no pot modificar directament el valor de les variables OPC, sinó que ha de fer ús dels blocs anteriorment

mencionats per sobreesciure o llegir el valor de les variables OPC. Aquesta metodologia s'explica en més detall a l'apartat '5 – Aplicacions robòtiques amb OPC UA'.

L'impacte de la introducció d'aquests nous blocs de programació és que un projecte que treballi amb OPC UA serà més llarg que no pas un que no requereixi d'aquest tipus de comunicació.

4.5. Clients

El Servidor és només un dels dos elements que componen una xarxa OPC UA. El rol del Client és per una banda, connectar-se al Servidor i visualitzar les dades que s'estiguin retransmetent. En situacions puntuals, el Client pot actuar de Servidor i editar el valor de certes variables mitjançant l'escriptura. En casos normals, però, el client es limita a monitoritzar el seguit d'informació que surt del Servidor.

Existeixen diferents programaris d'ordinador que actuen com a Clients de xarxes OPC UA. Algunes de les opcions són de pagament, mentre que altres són gratuïtes o bé tenen un límit de temps. Altres diferències notables recauen en aspectes gràfics (interfície de l'usuari, opcions de personalització etc.). Un altre element diferencial pot ser la incorporació d'eines addicionals que amplien les funcionalitats de cada Client.

En aquest treball acadèmic, s'ha atorgat, principalment, un rol de monitoratge al programari que actua com a Client. A internet es poden trobar diferents programes per monitorar xarxes OPC UA. Al llarg d'aquest projecte s'han utilitzat tres Clients: UA Expert, OPC Expert i Prosys UA Monitor. En aquest apartat es detallarà les característiques de cada un, la seva configuració i alguns dels seus usos.

4.5.1. UAExpert

UAExpert és un dels productes que ofereix l'empresa alemanya Unified Automation [41]. Aquesta companyia es centra en desenvolupar programaris al voltant de OPC i OPC UA. UAExpert és una de les seves propostes més populars, en part per què compta amb un gran suport dels desenvolupadors, fòrums per la comunitat i és gratuït.

UAExpert és un Client OPC UA desenvolupat amb llenguatge C++ dissenyat per escenaris genèrics on es vulgui testejar la comunicació OPC UA. Es pot descarregar des de la seva web [42] i compta amb suport de pràcticament totes les especificacions de OPC UA.

Posada en marxa de UAExpert:

Un cop hem instal·lat i obert el programa, per formalitzar la comunicació cal afegir un Servidor clicant el botó dret del ratolí sobre 'Server' en l'apartat 'Project'. Evidentment, cal que haguem engegat prèviament el Servidor en el robot UR3e.

Se'ns obrirà una nova pestanya on podem posar un nom del Servidor. Després cal anar a la pestanya 'Advanced' i a 'Endpoint Url' posarem **opc.tcp://130.130.130.51:4840**. Aquesta adreça correspon a l'adreça IP del robot més el port del servidor que hem iniciat en l'apartat anterior.

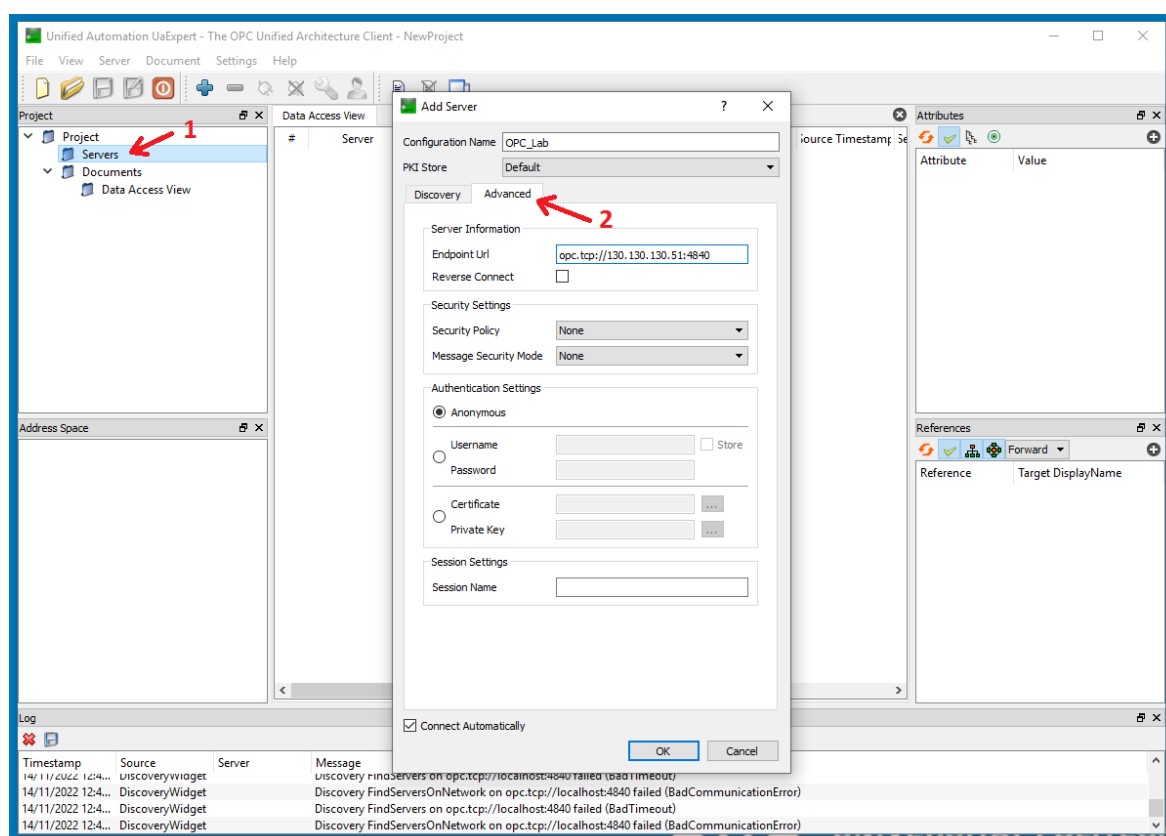


Figura 4.5. Configuració del Servidor en UAExpert (Font: Autor)

En aquesta pestanya també hem de seleccionar la mateixa configuració de seguretat que haguem posat al Servidor. En un escenari acadèmic es pot seleccionar Security Policy > None i Authentication Settings > Anonymous. El següent pas és seleccionar l'opció "Connect Automatically" i llavors OK.

Després se'ns obrirà una finestra on ens informa que el certificat del servidor no és de confiança. Aquest missatge és esperat i només cal seleccionar a la part inferior la casella "Accept the server certificate for this session" i després el botó "Trust Server Certificate". Si ens surt en verd com a la foto de la dreta, podem prémer a Continuar.

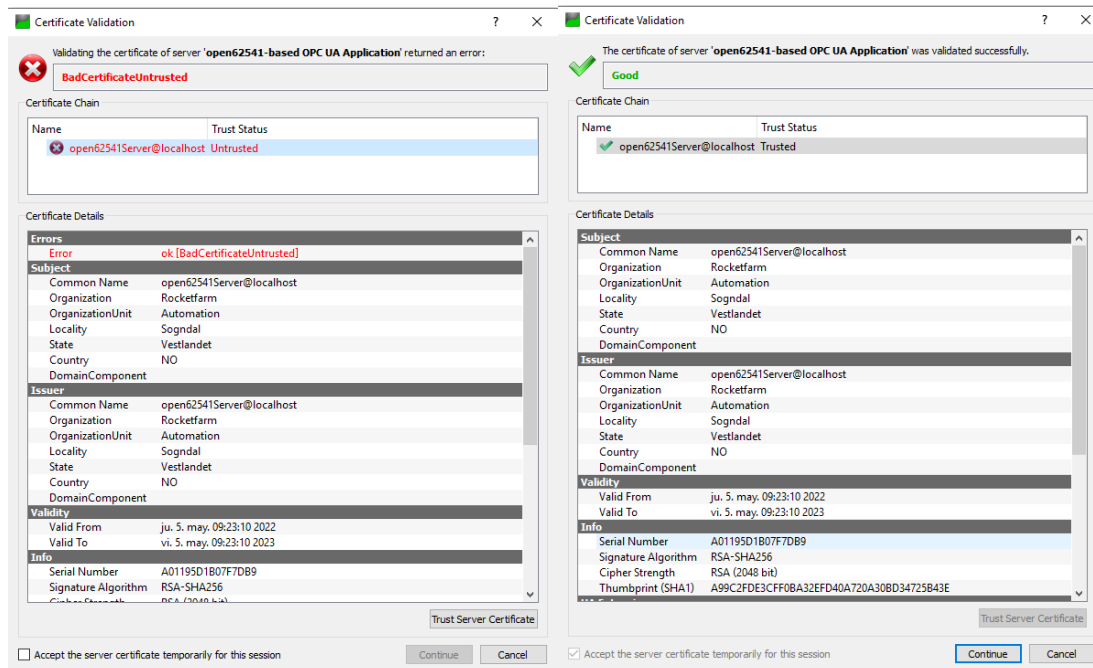


Figura 4.6. L'abans i el després de la validació dels certificats (Font: Autor)

Arribats en aquest punt, podem buscar a la part inferior, on es troba el Message Log, el següent missatge: 'Connection state of server OPC_Lab changed to Connected'.

Si apareix, significa que la connexió ha estat satisfactòria i tenim connexió amb el robot (Servidor).

Ha finalitzat la configuració inicial del Client UAExpert. El següent pas és personalitzar l'espai central del programa per tal de veure les variables de la xarxa OPC UA que calgui. Tots els Clients OPC UA disposen d'una zona anomenada 'Discovery' [43] on es recullen tots els paràmetres que s'estan enviant des del Servidor. En el cas de UA Expert, la funcionalitat 'Discovery' es troba a l'esquerra, en una pestanya anomenada 'Adress Space'. Si es clica sobre 'Objects' podem veure un seguit de variables que corresponen a les que s'hagin configurat prèviament en el Servidor. Altrament es pot veure una variable de nom 'Server' que inclou dades tècniques sobre el Servidor en el qual s'està connectant el Client.

Hauem d'arrossegar cada variable a la zona central ('Data Access View') per tal de veure el nom de cada variable i el seu valor, entre altres dades d'interès. Aquesta metodologia d'arrossegar les variables dins de 'Discovery' fins a la zona central es comuna en molts dels Clients OPC UA.

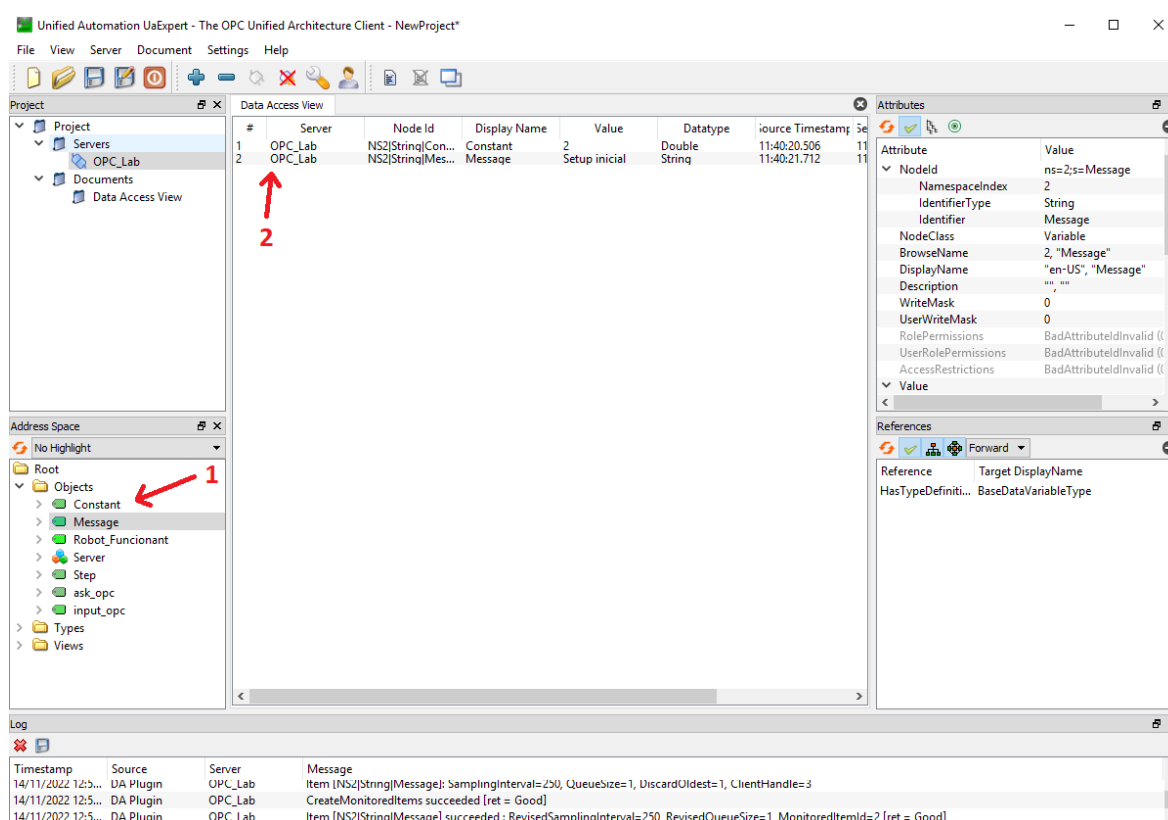


Figura 4.7. Pantalla principal de UAExpert (Font: Autor)

Es pot personalitzar la visualització dels atributs de les variables afegint o eliminant columnes amb el botó dret del ratolí. Addicionalment, la zona dreta ens amplia la informació de cada variable quan seleccionem una en concret.

L'atribut 'Value' de cada variable s'actualitza en temps real d'acord amb el valor que tingui la variable a cada moment en l'entorn OPC UA. Cal destacar que UAExpert permet sobre escriure aquest valor fent doble clic i introduint el nou valor desitjat en l'atribut 'Value'. En el moment que modifiquem el valor, el Client agafa momentàniament el rol de Servidor per editar el valor d'una variable OPC UA. Aquesta possibilitat de sobre escriure valors és similar en tots els clients OPC UA.

Les funcions de monitoratge i modificació dels valors de les variables OPC UA són les utilitats principals del programari UAExpert. Destacar que específicament aquest programa compta amb una interfície gràfica simple i intuïtiva i la seva configuració és senzilla. Tot i que disposa de funcions per crear gràfics, no s'ha pogut aconseguir aquesta visualització almenys amb el programa UAExpert i el robot UR3e.

4.5.2. OPC Expert

Durant el procés de recerca d'informació sobre el conjunt de programes d'ordinador que donen suport a aplicacions OPC UA, s'han trobat alternatives amb funcions extres que són interessants alhora de treballar amb aquesta comunicació.

Un d'aquests programes és OPC Expert [44], una programari que permet veure i manipular de manera fàcil informació OPC. Té una compatibilitat total amb l'estàndard OPC UA i s'ha de destacar que incorpora un programari adicional per treballar directament amb Excel i les dades OPC. Aquesta sintonia entre OPC Expert i el programa Excel obre un ventall de possibilitats molt interessant pel que fa a gràfics i mètodes de visualització de les dades. Aquesta funcionalitat només s'ofereix en aquest Client.

L'ús de OPC Expert és gratuït però està limitat a sessions de quatre hores de durada. Per la seva banda, la funcionalitat Excel està limitada a una hora. Passat aquest temps s'ha de tancar Excel i tornar-lo a obrir, obtenint així una hora més. Cal assegurar-se que no hi ha cap procés actiu ni de Excel ni de OPC Expert a l'administrador de tasques. Es pot repetir aquest pas tants cops com es vulgui per allargar l'ús d'aquest programa.

En el marc de treball d'aquest projecte universitari aquestes limitacions no ens afecten, però per aplicacions de caràcter professional s'hauria de contemplar la compra de la llicència del programa. Aquest es pot descarregar directament des de la seva web (requerint un registre previ). La pantalla principal del programa és similar a UAExpert: a la banda esquerra hi trobem l'accés al Servidor OPC i a la xarxa general, i a la pestanya de la dreta podem arrossegat les diferents variables que vulguem visualitzar. També compta amb un 'Message Log' (historial de missatges i errors).

Posada en marxa de OPC Expert:

La seva configuració és parcialment diferent a UAExpert. OPC Expert compta amb una funcionalitat de descobriment que permet al programari cercar en la xarxa local totes les aplicacions OPC i OPC UA. Que es trobi l'aplicació OPC UA és un requisit necessari per tal de connectar-nos. Així doncs, el primer pas serà especificar l'adreça IP on cercar el servidor OPC UA. Per fer-ho, cal fer clic dret a l'apartat 'Entire Network' i afegir l'adreça IP del robot (130.130.130.51) en l'opció 'Add computer'.

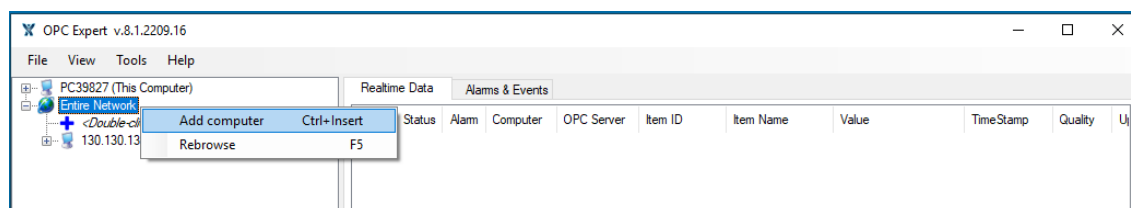


Figura 4.8. Cerca del Servidor a OPC Expert (Font: Autor)

El següent pas és especificar el mode de connexió al Servidor concret. Un cop aparegui el Servidor del robot sota l'adreça IP, cal fer clic dret i en l'opció 'Connect' seleccionar 'Default' per connectar-nos al Servidor si aquest no té opcions de seguretat configurades. Si volem personalitzar les característiques de seguretat, caldrà llavors seleccionar 'Custom' per tal d'introduir els paràmetres específics de l'entorn OPC UA (autenticació, signatura o xifratge). Reiterar que només es formalitzarà la connexió en el cas que tant Client com Servidor coincideixin en tots els atributs de seguretat.

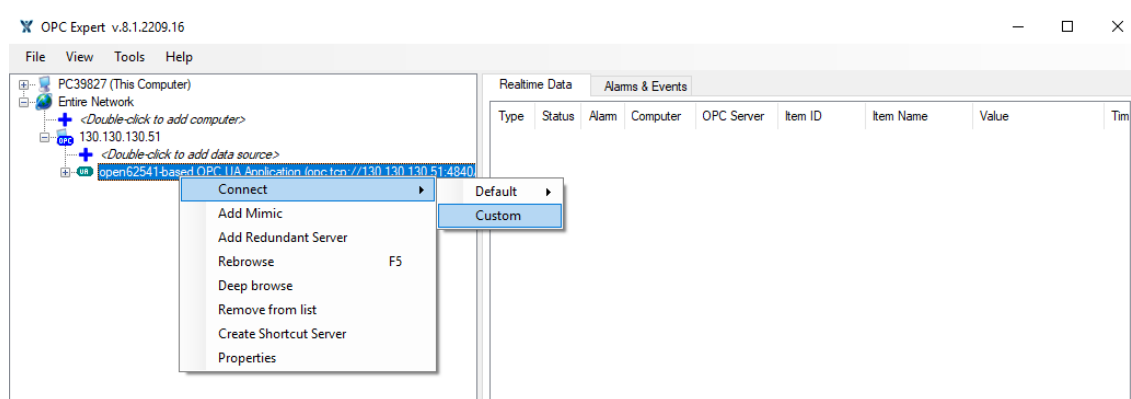


Figura 4.9. Configuració del mode de connexió amb el Servidor en OPC Expert (Font: Autor)

Amb la connexió realitzada, ens apareix a l'esquerra la funcionalitat 'Discovery' per tal de poder visualitzar les diferents variables OPC del servidor. Es pot fer el seguiment en temps real de les variables arrossegant-les a la zona central, similar a UAExpert. Allà es poden visualitzar altres dades addicionals referents a les variables. A la part inferior es troba l'historial d'avisos i errors.

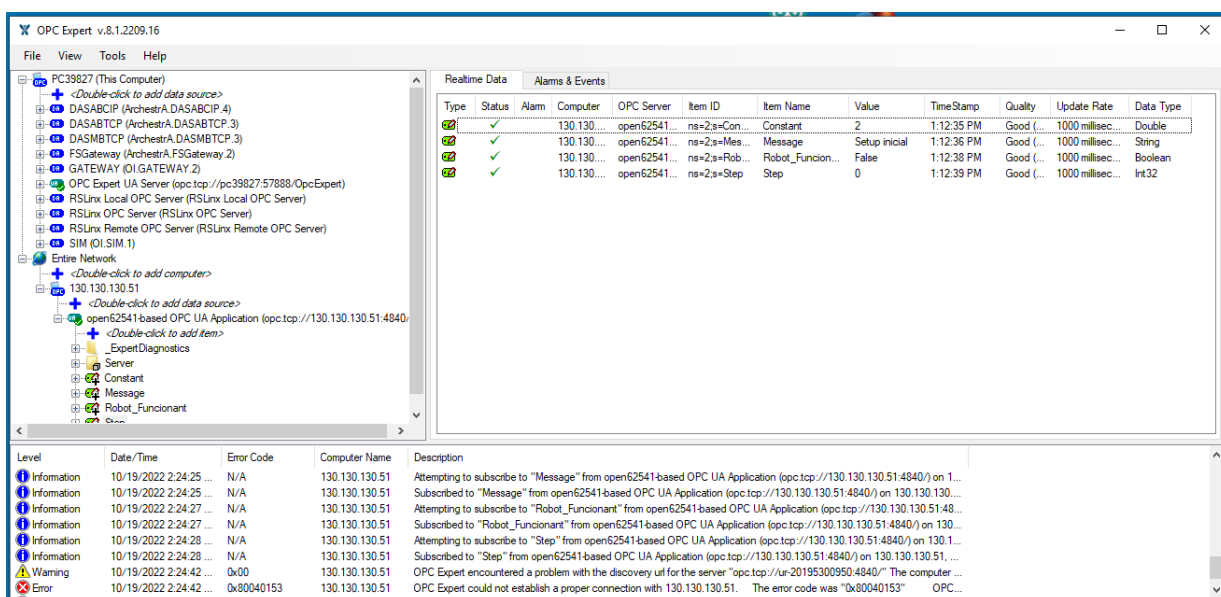


Figura 4.10. Pantalla principal URCap OPC Expert (Font: Autor)

Una funcionalitat a destacar d'aquest Client és la possibilitat de realitzar gràfics simples de manera immediata. Tan sols cal fer clic dret sobre la variable que es vulgui visualitzar en format gràfic i seleccionar Trend. Se'ns obre una pestanya on es mostrarà un històric del valor que té la variable actualment i en el passat (comença a registrar des del moment que obrim la pestanya Trend). Aquesta funcionalitat està pensada per mostrar variables Booleanes i Enteres, ja que l'eix vertical es visualitzen només valors numèrics. Es pot fer diferents nivells de zoom, personalitzar el color, veure el valor actual/passat de la variable i ens mostra si hi ha hagut un tall en la comunicació OPC UA.

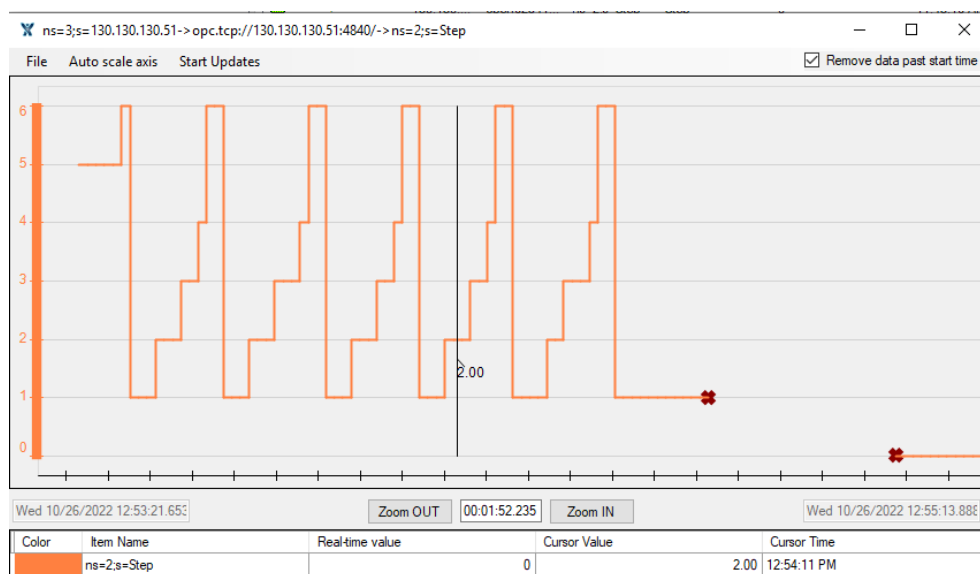


Figura 4.11. Visualització de gràfics a OPC Expert (Font: Autor)

Si activem la funcionalitat de Excel (Tools > Options > General > Enable Excel features), ens apareixerà el proper cop que obrim Excel un missatge que demana acceptar la instal·lació del Add-in. Un cop s'ha instal·lat, Excel es reinicia i ja apareixerà a barra superior una nova pestanya anomenada OPC.

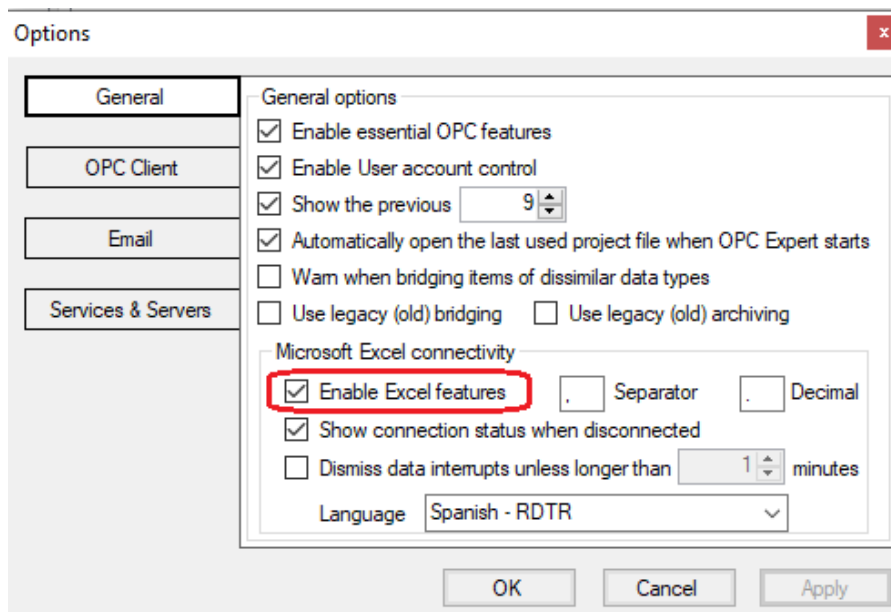


Figura 4.12. Opció per habilitar la funcionalitat Excel (Font: Autor)

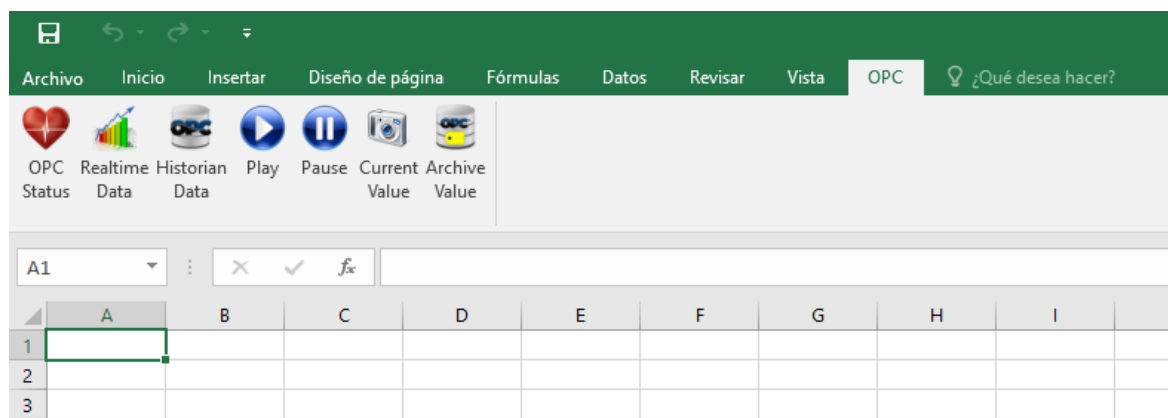


Figura 4.13. Accés directe a les funcionalitats OPC a Excel (Font: Autor)

Aquesta és una funció molt interessant que diferencia OPC Expert de la competència i que obre les portes a crear fulls de càlcul completament personalitzats per visualitzar i editar el conjunt de dades de la comunicació OPC UA.

Més en detall, a la pestanya OPC d'Excel apareixen set icones. A continuació s'explica la funció de cada un:

- OPC Status: col·loca en 6 files informació bàsica sobre l'estat del Servidor (estat de la connexió, possibilitat de rebre dades en temps real/històric, puls cada 1 segon)
- Realtime Data: s'obre una pestanya que permet configurar quina o quines variables volem visualitzar en temps real.
- Historian Data: apareix una nova pestanya que permet incorporar un històric de dades en un determinat rang de temps.
- Play: inicia la visualització en temps real.
- Pause: atura la visualització en temps real.
- Current Value: grava en una casella el valor actual d'una variable, però no l'actualitza en temps real.
- Archive Value: serveix per recuperar el valor d'una variable en un moment determinat del passat.

Ús de les funcions d'Excel de OPC Expert:

OPC Expert permet a Excel rebre dades OPC històriques i en temps real [45]. OPC Expert utilitza un programari addicional i un servidor RTD (Dades en Temps Real) per gestionar la informació. La manera de treballar amb Excel és mitjançant fórmules preestablertes. Aquestes es poden escriure directament a una casella del full de càlcul.

Per visualitzar el valor d'una de les variables en temps real, la instrucció més bàsica és la següent:
`=RDTR("expertd";"130.130.130.51";"opc.tcp://130.130.130.51:4840/";"ns=2;s=Step";500)`

Els elements més importants d'aquesta fórmula són els darrers: 'ns' indica el node, s el nom de la variable i 500 és el temps d'actualització.

`=RDTR("expertd";"130.130.130.51";"opc.tcp://130.130.130.51:4840/";"ns=2;s=Step";"Timestamp";1000)`

Afegint l'element 'Timestamp' ens col·loca en la casella el temps exacte (HH:MM:SS) en el qual s'ha rebut el valor.

Podem variar lleugerament l'expressió anterior per permetre guardar valors de cicles passats:

```
=RDTR("expertd";;"130.130.130.51";"opc.tcp://130.130.130.51:4840/";"ns=2;s=Step";"Cycle(1)";1000)
```

```
=RDTR("expertd";;"130.130.130.51";"opc.tcp://130.130.130.51:4840/";"ns=2;s=Step";"Cycle(2)";1000)
```

El valor més recent es col·loca en la casella del cicle 1. Passat 1000ms, es guarda en cicle 2 i s'actualitza cicle 1 amb el valor més recent. Passat 1 segon, el valor cicle 2 es guarda en cicle 3 i el del cicle 1 a cicle 2. És una seqüència que es realitza en bucle i es permet visualitzar quin valor tenia una certa variable en els cicles passats (la versió gratuïta permet treballar amb fins a 15 cicles). Seleccionant els 15 cicles podem introduir un gràfic de línies i visualitzar per exemple en quin pas del programa es troba el robot en cada cicle.

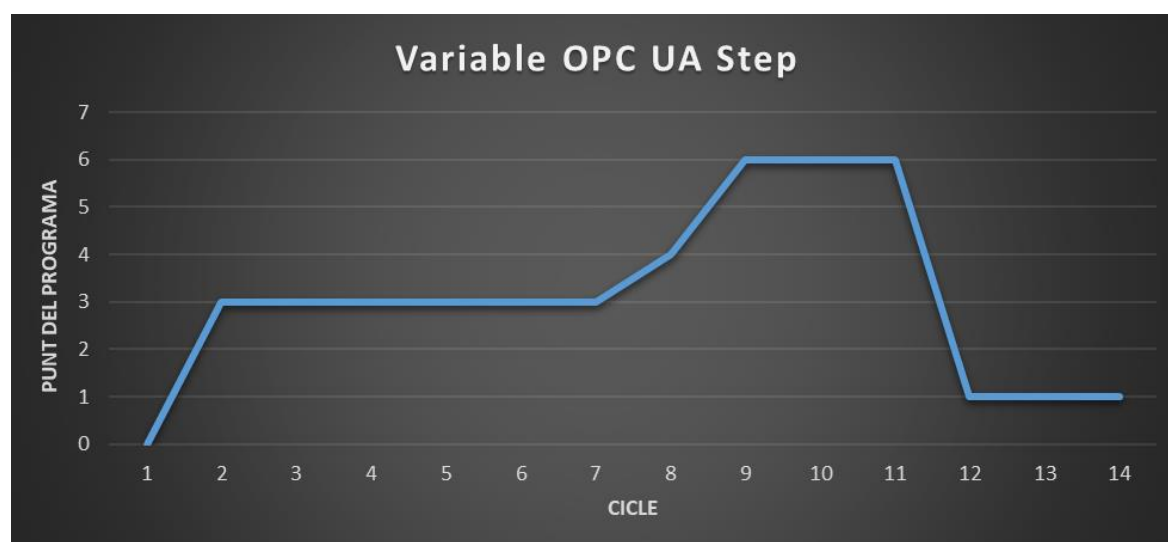


Figura 4.14. Gràfic d'una variable OPC UA fet amb Excel (Font: Autor)

L'anterior gràfic s'actualitza en temps real, a la freqüència d'actualització que s'hagi configurat al Servidor i al OPC Expert.

Adicionalment, el Add-in de Excel ens dona la possibilitat d'interactuar directament amb el programa OPC Expert per modificar el valor de les variables mostrades. És a dir, podem canviar el valor d'una variable OPC directament des de Excel. Això permet, per exemple, controlar el robot amb un 'if' condicional que comprovi el valor de certa variable i realitzi uns moviments o uns altres depenent del valor que s'imposi al programa Excel.

Per aconseguir-ho és tant fàcil com afegir al final de la fórmula de visualització en temps real la casella on es troba el nou valor.

=RDTR("expertd";;"130.130.130.51";"opc.tcp://130.130.51:4840/";"ns=2;s=Constant";;500;D15)

A la casella D15 podem introduir un valor numèric que s'imposarà directament sobre la variable OPC (Constant en el cas anterior). És a dir, en el moment que posem 1 a D15 i es prem Enter, Constant passa a tenir un valor de 1. OPC Expert s'encarrega de la comunicació amb el Servidor.

Finalment, ajuntant totes les possibilitats que s'han detallat en un sol full de càlcul, podem obtenir un quadre de visualització de la informació OPC UA com aquest:

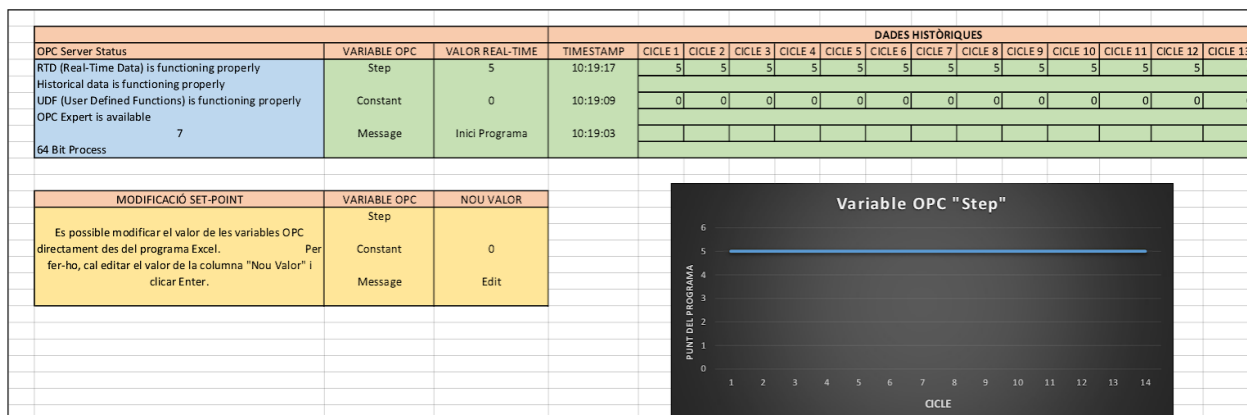


Figura 4.15. Muntatge final amb OPC Expert i Excel (Font: Autor)

4.5.3. Prosys UA Monitor

Prosys UA Monitor [46] és un dels programaris que desenvolupa i ven l'empresa finlandesa Prosys OPC. Aquesta companyia es dedica a la comercialització de programes d'ordinador que treballen dins l'àmbit de OPC i OPC UA. Tots els seus productes són de pagament però compten amb versions de prova limitades a sessions de dues hores.

La característica diferencial de Prosys UA Monitor és el seu focus en la visualització de les dades. No és el mateix voler visualitzar una variable booleana que un 'string'. Aquest Client compta amb un complet ventall d'opcions intel·ligents de visualització per diferents tipus de variables. El grau de personalització que ofereix és molt elevat. Amb tot plegat, es pot aconseguir crear una pantalla de monitoratge adaptada completament a l'aplicació OPC UA específica.

A diferència de les altres dues alternatives, UAExpert i OPC Expert, les dades no es mostren en simples files i columnes de text sinó que permet personalitzar la mida i el color de la lletra, la dimensió de cada casella, la posició en la quadrícula etc.

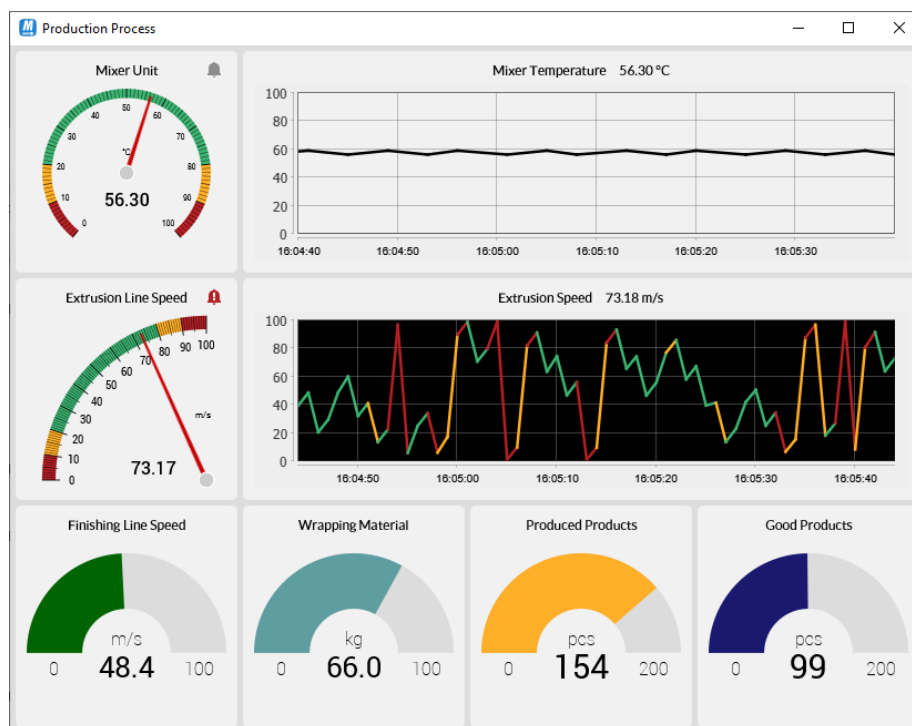


Figura 4.16. Exemple de la pantalla final feta amb Prosys UA Monitor (Font: ProsysOPC)

Un altra característica que diferencia aquest programari dels seus competidors és la creació simplificada i intuïtiva d'alarmes i avisos. El programa permet configurar en pocs clics un sistema d'alarmes que salta quan una o més variables arribin a un determinat valor. Totes les alarmes i missatges d'avís es recullen en una pestanya unificada.

Posada en marxa de Prosys UA Monitor:

La configuració de Prosys UA Monitor repeteix la metodologia seguida en els altres clients: en obrir el programa, el primer pas és anar a la finestra 'Source Servers' i escriure les dades del Servidor (prèviament activat en el robot UR3e). El programa ens deixa testejar la comunicació per confirmar que el Servidor es troba a l'abast. La figura 4.18 especifica les dades que cal posar per connectar-nos al Servidor del UR3e. Els paràmetres de seguretat són idèntics en els dos punts. Seguint les pautes de OPC UA, el programa demanarà acceptar el certificat del Servidor abans de connectar-nos. Un cop s'accepti, Client i Servidor estaran connectats.

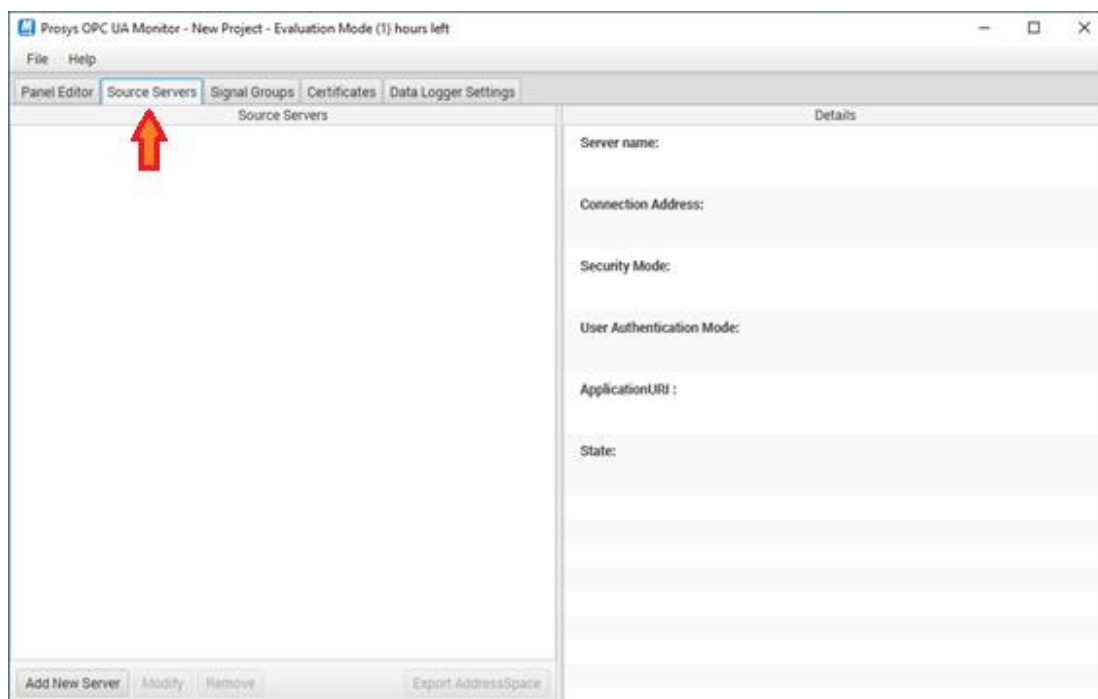


Figura 4.17. Accés a la configuració del Servidor a Prosys UA Monitor (Font: Autor)

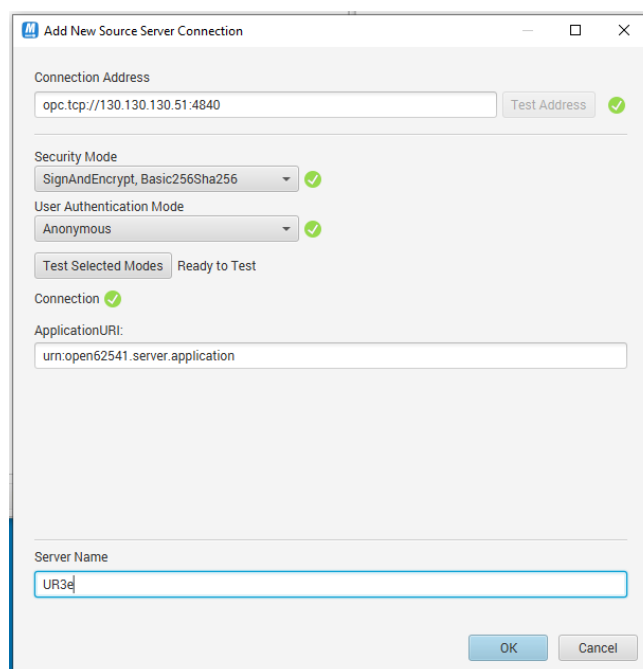


Figura 4.18. Configuració del Servidor a Prosys UA Monitor (Font: Autor)

El següent pas és importar aquelles variables que voldrem veure en pantalla. Cal anar ara a la pestanya 'Signal Groups', a l'apartat 'Address Space' seleccionar el Servidor desitjat i llavors tindrem accés a la funció 'Discovery', la qual ens permet cercar les variables OPC UA dins de 'Objects'. Com ja s'ha fet amb els altres Clients, cal arrossegar cada variable una a una a l'espai central per poder-la visualitzar.

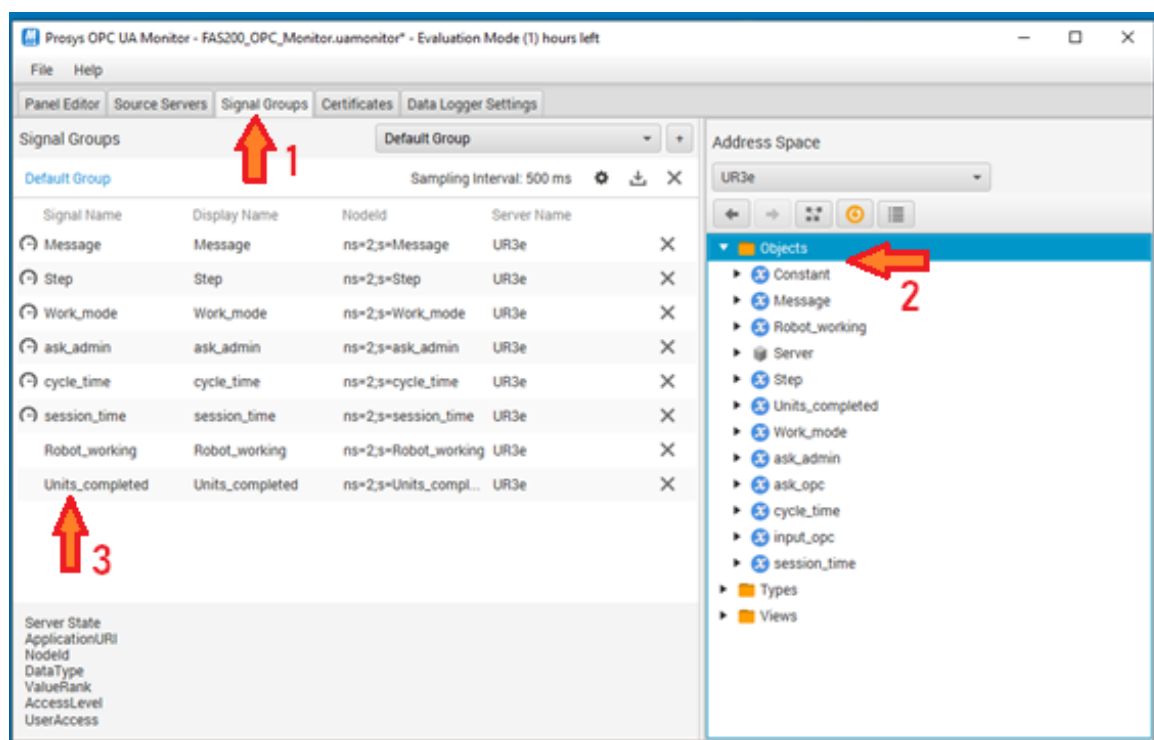


Figura 4.19. Configuració de les variables a visualitzar (Font: Autor)

Un cop haguem arrossegat totes les variables que es desitgi monitoritzar, cal clicar a 'Panel Editor' per tal de configurar la pantalla de monitoratge. En aquesta pantalla podem anar afegint caselles que incorporaran les funcions gràfiques per mostrar les variables. Cada casella té la seva finestra de configuració on seleccionem la variable a mostrar i des d'on podem personalitzar com es visualitzarà (color, mida de la lletra, unitats, atributs específics etc.). També a la finestra de personalització serà des d'on podem afegir i configurar alarmes i avisos per cada variable en particular. En referència a aquest tema, el programa permet afegir un llindar màxim o mínim que farà saltar un avís/alarma en cas que el valor de la variable OPC superi el llindar configurat. Addicionalment, es pot personalitzar el missatge que es mostra quan s'activi l'avís.

Finalment destacar que totes les variables tenen una casella que es pot clicar per habilitar les opcions de escriptura en aquella variable. Aquesta opció habilita l'escenari en el qual el Client pot editar i sobreesciure el valor de les variables OPC UA.

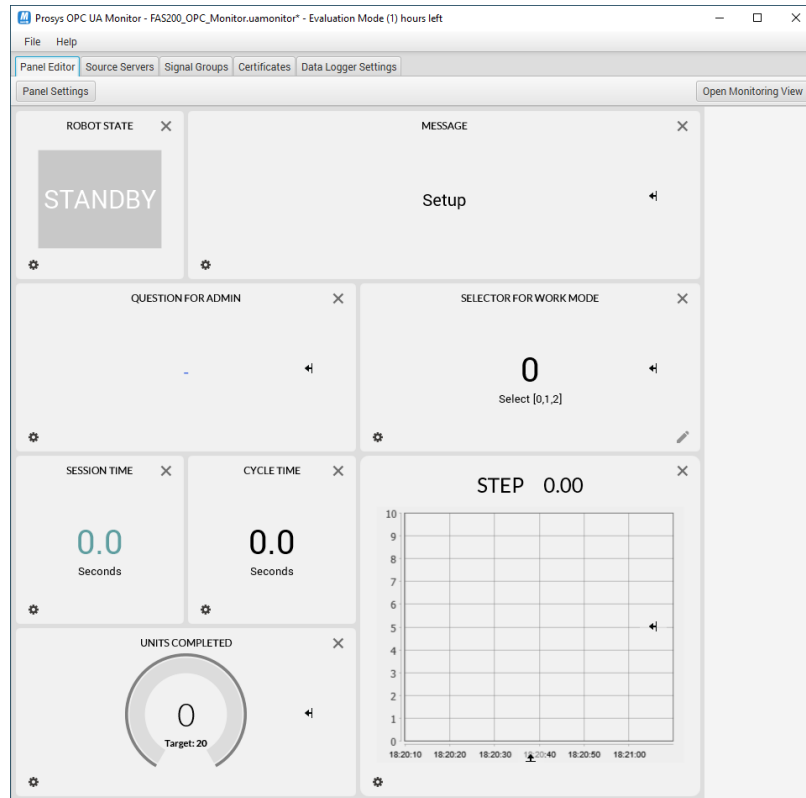


Figura 4.20. Panell d'edició de les variables OPC UA (Font: Autor)

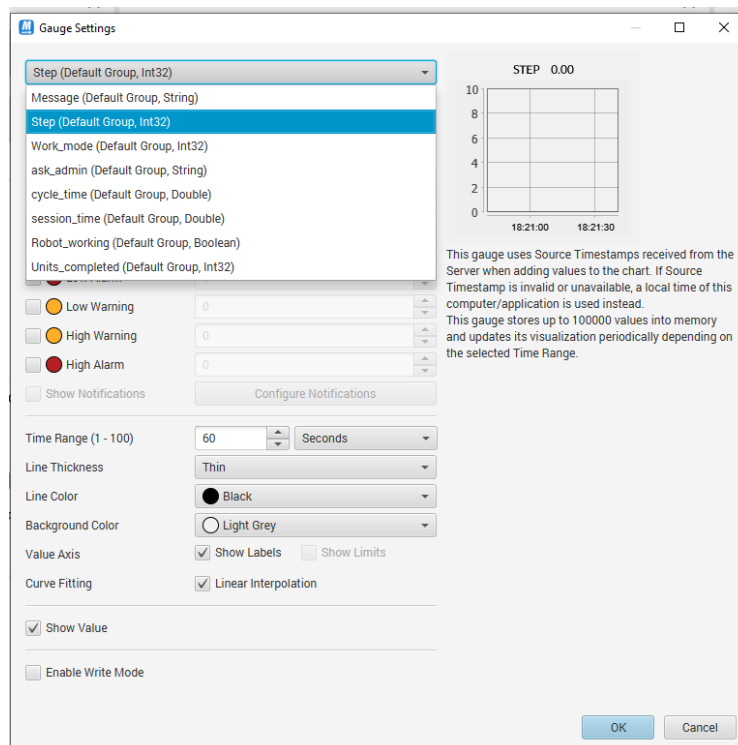


Figura 4.21. Edició específica d'una casella del panell (Font: Autor)

Després de personalitzar diverses caselles i configurar la visualització de vuit variables, cal clicar a l'opció 'Open Monitoring View' per veure la interfície gràfica final. Amb pocs minuts es pot aconseguir una pantalla de monitoratge similar a aquesta:

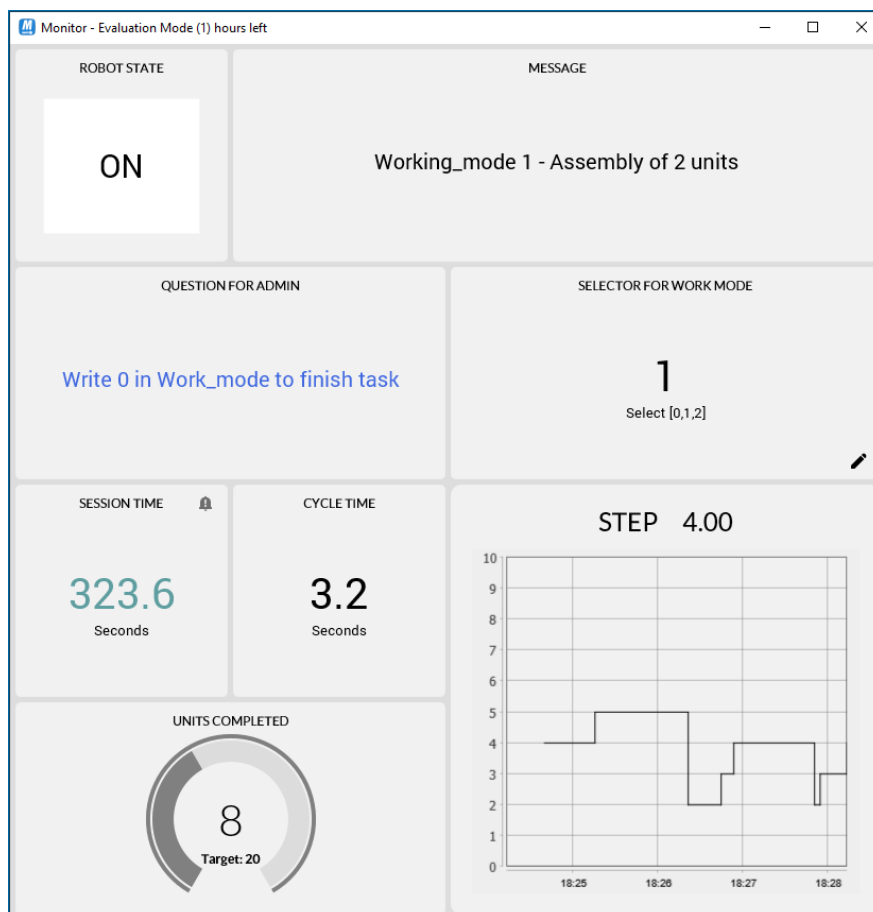


Figura 4.22. Pantalla final feta amb Prosys UA Monitor (Font: Autor)

4.6. Tests de seguretat

L'apartat 2 d'aquest treball inclou un recull detallat de les característiques de seguretat que incorporen la majoria de les aplicacions OPC UA. La teoria afirma que és un protocol segur per disseny i que, seguint el codi de bones pràctiques, es pot treballar en un entorn OPC UA protegit davant d'atacs externs.

L'encriptació i la signatura dels diferents elements enviats es fa de manera automàtica i pràcticament invisible per a l'usuari final. Per aquest motiu pot ser difícil confirmar que efectivament s'estigui treballant en un entorn protegit. Per tal de corroborar aquestes afirmacions i alhora entendre l'impacte

de les diferents combinacions de seguretat presents en el URCap OPC UA, s'han realitzat una sèrie de tests. S'han intentat reproduir seguint una metodologia comuna en tots ells, replicant un escenari gairebé idèntic en tots els casos.

Per la realització dels experiments s'ha utilitzat el programari Wireshark [47]. Aquest és un programari lliure i de codi obert que permet analitzar les diferents trames que s'envien per determinats protocols. Abarca la xarxa local en la qual estigui connectat l'ordinador i proporciona diferents eines per estudiar els paquets enviats. El seu ús és gratuït. Aquest programari s'ha combinat amb el Client OPC Expert, el qual disposa d'opcions per personalitzar el mode de seguretat empleat i compta amb una finestra on resumeix el nivell de protecció present en l'entorn OPC UA.

Pel que fa a les proves, s'han estudiat 5 possibles casos, els quals són el resultat de combinar els següents paràmetres de seguretat presents en el URCap OPC UA:

- Modes de seguretat pels missatges: permet seleccionar entre Cap / Signar / Signar i encriptar
- Polítiques de seguretat: les opcions disponibles són Cap / Basic128Rsa15 / Basic256 / Basic256Sha256
- Autenticació d'usuari: es pot triar entre Anònim / Usuari i contrasenya

Cas 1: No incorpora cap mètode de seguretat. Es selecciona Anònim per l'autenticació i Cap en modes de seguretat.

Cas 2: Introduïm autenticació mitjançant usuari i contrasenya. No es xifren ni es signen els missatges.

Cas 3: Mantenim usuari i contrasenya i s'utilitza signatura (Basic256Sha256). No hi ha xifrat.

Cas 4: Correspon a seleccionar autenticació amb usuari i contrasenya i addicionalment es firmen i es xifren els missatges amb la política Basic128Rsa15.

Cas 5: aquest cas incorpora tots els mètodes de seguretat presents i les polítiques més segures. S'ha seleccionat usuari i contrasenya, signatura dels missatges i xifrat mitjançant Basic256Sha256.

La metodologia seguida per testejar cada cas ha estat la següent: el servidor es configura amb les combinacions de seguretat especificades. Es replica la configuració de seguretat en el Client per permetre la connexió. Seguidament s'engega el Wireshark, el qual comença a mostrar la totalitat de les xarxes que estan en ús en l'ordinador. Es mou el robot amb un programa determinat que interactua amb variables OPC UA. El test finalitza quan s'hagin realitzat cinc captures de pantalla: opcions de seguretat que detecta el Client, capçalera del missatge en cas de fer lectura/escriptura d'una variable OPC UA, cos del missatge en editar una variable String i una de tipus Double.

Aquesta metodologia es segueix per cada cas per poder obtenir uns resultats comparables entre ells. Aquests s'han recollit en format de taula (Taula 1). Les captures de pantalla resultants dels diferents casos es troben a l'Annex A. Cal destacar que en l'annex apareixen dos casos més, sumant un total de 7. Ara bé, dos dels casos són idèntics al cas 3 (signatura de missatges) i només canvia el mode de signatura seleccionat. No es va observar cap canvi notable i per aquest motiu s'ha prescindit d'aquests casos a la taula.

La selecció de les diferents combinacions de seguretat s'ha realitzat a l'apartat 'Security' dins de 'Server'. En cada cas testejat, es seleccionava una combinació idèntica entre Servidor i Client. A la següent figura es poden observar totes les opcions de seguretat disponibles:

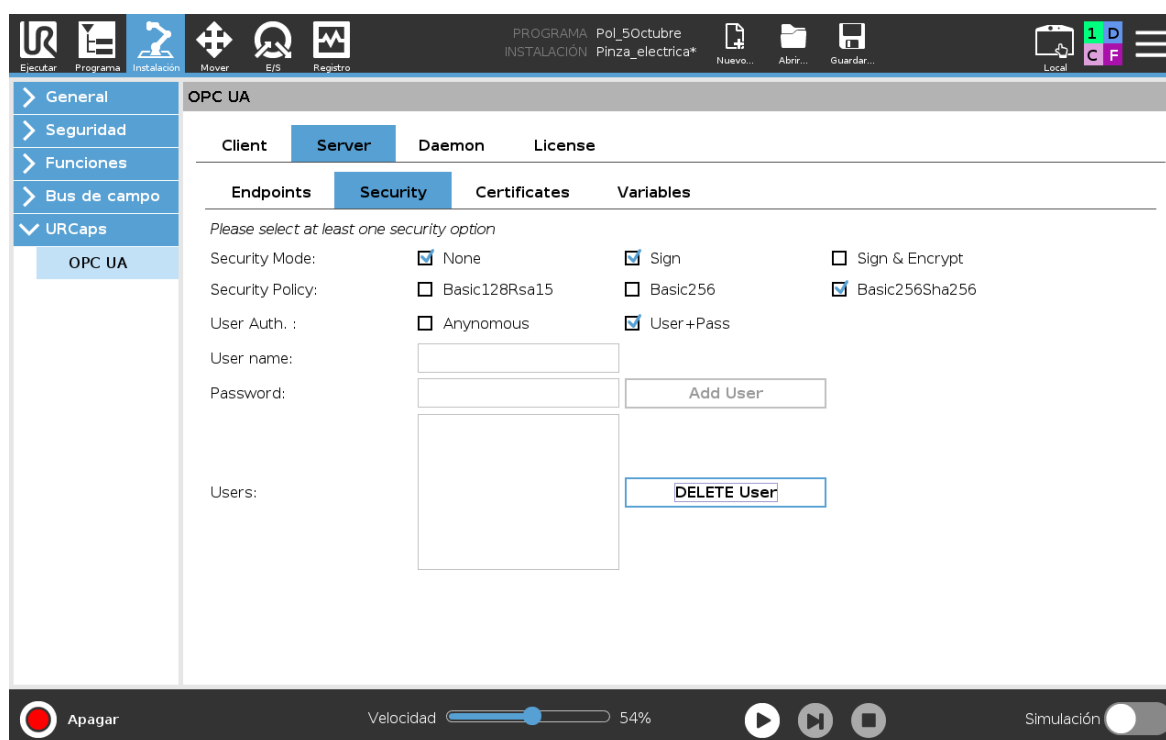


Figura 4.23. Una de les possibles combinacions de seguretat de OPC UA (Font: Autor)

Conclusions dels test de seguretat

El cas més segur de tots correspon al cas 5. En aquest cas el nivell de seguretat és de 3, el màxim. En aquest cas es signen els missatges per garantir la seva integritat, es protegeix l'accés a les dades mitjançant usuari i contrasenya i es xifren els paquets enviats per dificultar la tasca de veure quines comandes OPC UA s'estan realitzant. Cal destacar que aquest cas és l'única combinació que es recomana en el codi de bones pràctiques citat al principi del treball. OPC Foundation assegura que les altres combinacions poden ser vulnerables ja que utilitzen algorismes de xifrat antiquats. Així doncs,

en qualsevol aplicació real o que no sigui en l'àmbit acadèmic s'hauria de partir de, com a mínim, aquesta configuració de seguretat.

A l'altre extrem trobem el cas 1. En aquest cas la nostra comunicació és molt vulnerable i només cal que els atacants tinguin accés a la xarxa local per poder veure dades del Servidor i del Client, si les comandes realitzades són de lectura o escriptura i, encara més important, quina variable s'ha modificat i amb quin valor. El nivell de seguretat d'aquest cas correspon a 1, el mínim. Només es recomana l'ús d'aquesta combinació de seguretat en ambients acadèmics amb supervisió del professorat.

Entre mig trobem els altres casos. Tot i comptar amb més mesures de seguretat que el cas 1, es consideren igualment no segurs al no incorporar suficients mètodes de protecció o funcionar amb algorismes obsolets.

El paràmetre que té més impacte en la visualització d'informació crítica és el xifrat. En els casos que s'activa aquesta protecció, els missatges es xifren i només es visible la capçalera de la trama. Aquest és un comportament esperat que correspon amb la informació detallada en l'apartat 2 sobre la seguretat de OPC UA. Aquesta capçalera conté informació com l'adreça IP del Client i del Servidor, així com els seus ports. Ara bé, aquesta és l'única informació disponible que tindria un possible atacant, ja que el cos del missatge es troba totalment encriptat.

L'autenticació amb usuari i contrasenya té un impacte directe per l'operari del robot i l'administrador de planta, ja que ambdós han de conèixer les credencials per tal d'accedir a les dades i/o al control del robot.

La signatura dels missatges és un atribut que no s'ha pogut testejar al complet, ja que només intervé en cas que un atacant hagi pogut modificar el paquet enviat durant el seu transport. No ha estat possible replicar aquest escenari amb els meus coneixements actuals.

A la següent pàgina es troba la taula comparativa dels 5 casos. Es detalla la combinació de seguretat seleccionada, l'autenticació requerida en el moment de connectar Servidor-Client, la capçalera del missatge i un seguit d'informació important de l'entorn OPC UA: adreça IP, port, dades de seguretat i atributs de la variable editada (nom, tipus i valor).

En color **verd** es subratlla l'única combinació realment segura, el cas 5. Correspon a un nivell de seguretat 3 i només aquest cas incorpora totes les recomanacions oficials de OPC Foundation detallades en l'apartat '2.4.5 – Codi de bones pràctiques'. En color **groc** es marca aquells casos que tenen un nivell de seguretat 2, però que no es recomanen donat que compten amb mesures de protecció obsoletes. En **vermell**, essent un nivell de seguretat 1, els casos que només es contemplan per escenaris acadèmics degut al seu baix nivell de protecció.

Taula 1. Comparativa dels modes de seguretat disponibles.

Cas	1. Anònim	2. Usuari i Contrasenya	3. Usuari i Contrasenya + Signatura	4. Usuari i Contrasenya + Signatura + Encrypt Basic128Rsa15	5. Usuari i Contrasenya + Signatura + Encrypt Basic256Sha256
Connexió Servidor-Client	No es demana cap credencial per iniciar la comunicació.	Per establir la comunicació, demana usuari i contrasenya.	Per establir la comunicació, demana usuari i contrasenya.	Per establir la comunicació, demana usuari i contrasenya.	Per establir la comunicació, demana usuari i contrasenya.
Assumpte Missatge	'UA Secure Conversation Message: Read/Write Request'	'UA Secure Conversation Message: Read/Write Request'	'UA Secure Conversation Message: Read/Write Request'	'UA Secure Conversation Message: ServiceId 0'	'UA Secure Conversation Message: ServiceId 0'
Dades de l'entorn OPC	Adreça IP del Servidor i del Client, Port del Servidor i del Client, direcció física (MAC). Marca de temps.	Adreça IP del Servidor i del Client, Port del Servidor i del Client, direcció física (MAC). Marca de temps.	Adreça IP del Servidor i del Client, Port del Servidor i del Client, direcció física (MAC). Marca de temps.	Adreça IP del Servidor i del Client, Port del Servidor i del Client, direcció física (MAC). Marca de temps.	Adreça IP del Servidor i del Client, Port del Servidor i del Client, direcció física (MAC). Marca de temps.
Dades visibles de la trama	Dimensió i tipus del missatge, amplis detalls de seguretat, màscara de xifrat, comanda de lectura/escriptura, NodeId, Nom/Tipus/Valor de la variable.	Dimensió i tipus del missatge, amplis detalls de seguretat, màscara de xifrat, comanda de lectura/escriptura, NodeId, Nom/Tipus/Valor de la variable.	Dimensió i tipus del missatge, amplis detalls de seguretat, màscara de xifrat, comanda de lectura/escriptura, NodeId, Nom/Tipus/Valor de la variable.	S'indica la llargada i tipus del missatge, detalls de seguretat limitats.	S'indica la llargada i tipus del missatge, detalls de seguretat limitats.

5. Aplicacions robòtiques amb OPC UA

Amb les idees presentades fins aquest punt, s'haurien de tenir els coneixements necessaris per habilitar un Servidor al robot UR3e, configurar algunes variables i connectar alguns dels Clients citats.

Ara bé, si les variables OPC no varien, estariem observant sempre una pantalla on les dades tenen constantment el mateix valor. Cal, llavors, configurar el programa que realitza el robot en bucle per a que interactuï amb les variables OPC UA i actualitzi el valor quan calgui.

En aquest apartat es farà ús de les comandes 'Read/Write' introduïdes a l'apartat '4.4 – Programació específica de OPC UA' per tal de modificar tres programes estàndard del robot UR3e per a que treballin amb la xarxa OPC UA. Es posaran en pràctica, doncs, totes les idees i coneixements que s'han explicat en els apartats anteriors.

Per aconseguir-ho, es van haver d'idear escenaris en els quals fos interessant obtenir dades del funcionament del robot. Es va pensar en situacions anàlogues a les que ens podem trobar en plantes de producció i cadenes de muntatge actuals on treballen persones i robots, així com la manera de replicar-les al laboratori en una escala menor. Cada escenari ajudava a entendre i confirmar el funcionament dels diferents paràmetres de la comunicació OPC UA, per aquest motiu el primer escenari té una complexitat baixa, ja que correspon als primers mesos d'investigació quan encara no s'havien assimilat alguns dels atributs. El segon escenari augmenta la complexitat i afegeix funcionalitats addicionals. El tercer cas, realitzat a la fase final del projecte, engloba tots els coneixements apresos fins aleshores i incorpora parts dels dos primers.

5.1. Bucle simple OPC UA

En un entorn industrial real, el robot col·laboratiu UR3e tindria definida la seva tasca: un programa que repeteix tantes vegades com calgui per realitzar un procés industrial. En aquest cas la comunicació OPC UA pot agafar el rol d'element de suport, és a dir, proporcionar informació addicional a un segon operari sobre l'estat del procés. Amb aquest objectiu en ment, es parteix d'un programa simple on el robot realitza una sèrie de moviments i fa ús de l'element terminal del qual disposa: una pinça elèctrica. El programa es realitza en bucle fins que s'indica el contrari. En aquest programa s'introduiran estratègicament les comandes OPC UA per tal que aquest pugui comunicar-se amb un Client OPC UA.

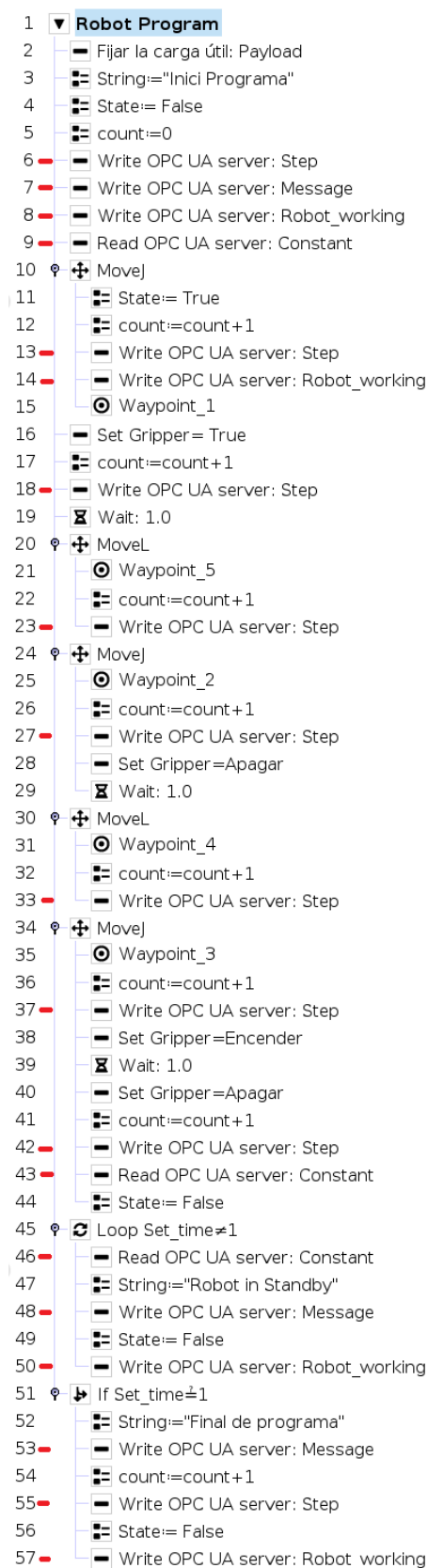


Figura 5.1. Programa Bucle OPC complet (Font: Autor)

La figura 5.1 mostra el programa complet del primer escenari pràctic funcionant amb OPC UA.

El robot realitza un total de cinc moviments i en el quart tanca la pinça elèctrica. Al finalitzar l'últim moviment, el programa entra en un bucle indefinit en cas que una certa variable tingui un valor diferent a 1. En el moment que la variable s'iguali a 1, acabem el programa i tornem al principi. El programa es repetirà per sempre.

Aquest programa podria funcionar perfectament sense la comunicació OPC UA. Però, com és el focus d'aquest treball, s'han afegit 19 línies (marcades amb vermell) que corresponen a interaccions amb la xarxa OPC UA. Aquestes comandes poden ser de lectura o escriptura.

Pel que fa a les variables, cal diferenciar entre dos tipus: les locals i les de OPC. Les locals són String, State, Count i Set_time. Aquestes interactuaran de manera directa amb les variables OPC que s'han creat en el Servidor seguint les pautes de l'apartat 4.3: 'Message', 'Robot_Working', 'Step' i 'Constant'.

Aquest simple bucle va servir per corroborar que si volem editar el valor d'una variable OPC, no es pot modificar directament des del programa, ja que aquest no té accés a les variables de la xarxa OPC UA. El que s'ha de fer és utilitzar les comandes OPC UA 'Read/Write' que poden replicar el valor d'una variable local en una OPC i a l'inrevés.

És important que s'entengui aquest concepte, per aquest motiu detallem un cas particular a continuació:

Les línies 3-4-5 inicialitzen les variables locals String, State, Count amb un determinat valor. Les següents línies 6-7-8 imposen el valor d'aquestes tres variables a 'Message', 'Robot_Funcionant' i 'Constant' respectivament. És a dir, mitjançant la comanda 'Write', s'aconsegueix que la

variable 'Message' tingui el mateix terme que String. El mateix passa amb les altres dues variables. El valor de les variables OPC es pot veure des del client (UAExpert / OPC Expert / Prosys UA Monitor) que reflexa aquest canvi en temps real. En un primer moment es va pensar que seria possible fer directament des del programa que, per exemple, la variable OPC 'Message' fos igual a un cert valor. Doncs es va confirmar que no era possible per aquesta via i que calia un pas entremig.

Amb la comanda 'Read' passa un fet similar: imaginem que volem fer un condicional ('if') que comprovi el valor de la variable OPC Constant, i depenent d'aquest valor, el programa faci una cosa o una altra. Doncs es va detectar, de nou, que el programa no té accés a les variables OPC directament, sinó que hem de fer ús d'una comanda addicional 'Read'. El cas de la línia 46 exemplifica aquesta anomalia: afegim una comanda que llegirà el valor que tingui en aquell moment la variable OPC 'Constant' i imposarà aquest valor sobre una variable local, en aquest cas 'Set_time'. Ara ja podem fer, a la línia 51, un condicional que comprovi el valor de 'Set_time', que és equivalent a Constant.

Aquesta situació es dona en totes les variables OPC, independentment del tipus. Així doncs, es conclou que per cada variable OPC UA que es vulgui utilitzar, hem d'introduir una equivalent en el programa. El resultat és que totes les variables OPC UA estan emparellades amb una de local, i mitjançant les comandes 'Read/Write' les relacionem.

Per aquest motiu es va observar des del principi que el fet de treballar amb la xarxa OPC UA causava, irremeiablement, un augment de la longitud del programa degut a les noves línies. No obstant, les noves comandes no tenen un impacte en el temps d'execució ja que s'efectuen en qüestió de milisegons. És a dir, en aquest estadi del projecte es va formular la hipòtesis que la introducció de les noves comandes OPC UA causava un augment de la llargada del programa però no tenia el mateix impacte en el temps d'execució, que no es veia afectat. A mesura que va avançar el projecte, es va descobrir un mètode per minimitzar el número necessari de comandes OPC UA, minimitzant el seu impacte.

Aquest primer programa va servir d'altra banda per comprovar que a través del Client es podia editar en temps real el valor d'una variable OPC UA. El programa entra en bucle quan Set_time no val 1. Per defecte aquesta variable val 0, és a dir, que sempre entrarà al bucle. En cas que des del Client variem el valor de 'Constant', emparellada amb Set_time, es llegirà aquest canvi que permetrà sortir del bucle i finalitzar el programa. Amb aquesta prova es va confirmar que el Client pot actuar momentàniament com a Servidor i que es poden realitzar aplicacions robòtiques on el programa depengui de la resposta del Client.

En les següents quatre figures es poden veure alguns dels moviments que realitza el robot i les dades que es visualitzen des del Client (UA Expert) en cada pas.

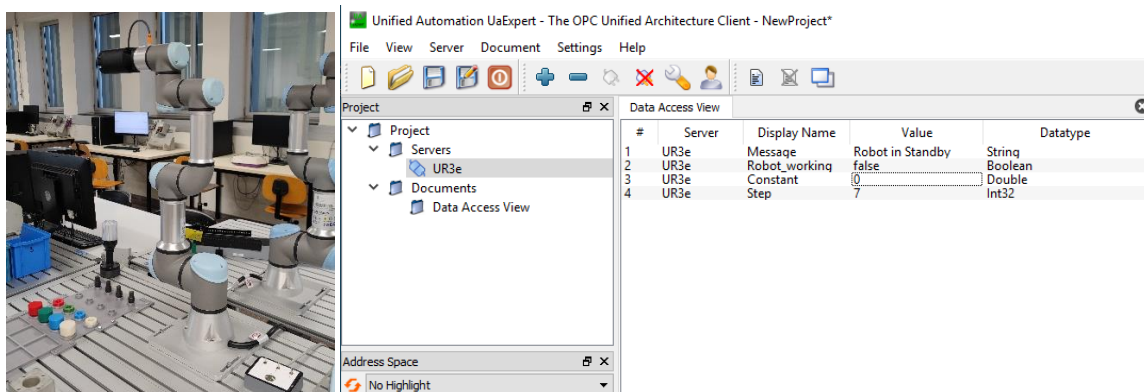


Figura 5.2. Robot en repòs i pantalla del Client (Font: autor)

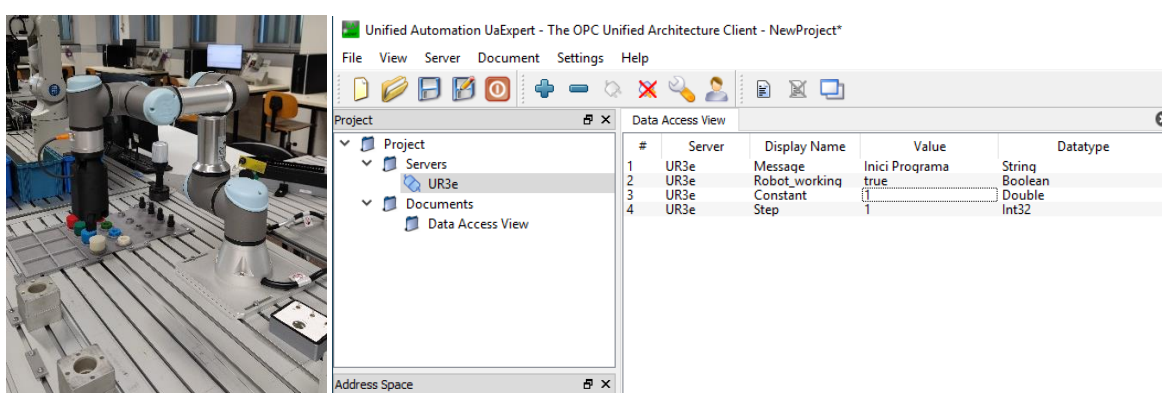


Figura 5.3. Robot en el moviment núm. 1 i pantalla del Client (Font: autor)

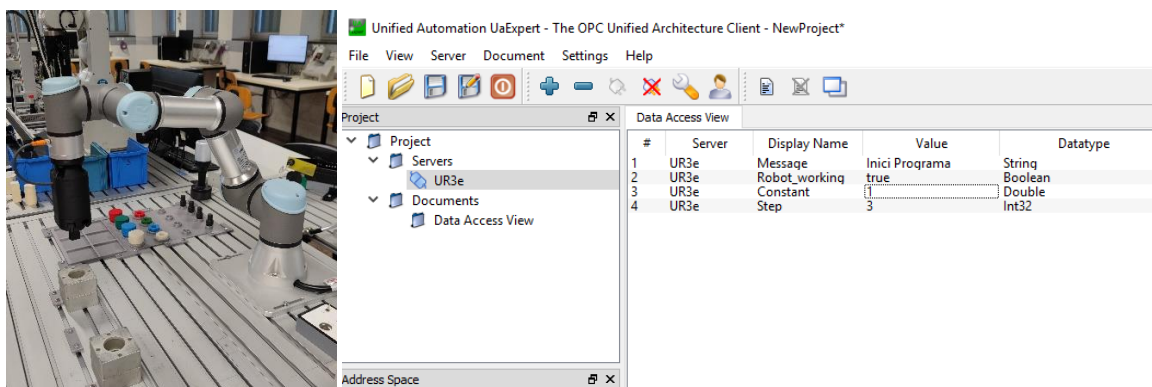


Figura 5.4. Robot en el moviment núm. 3 i pantalla del Client (Font: autor)

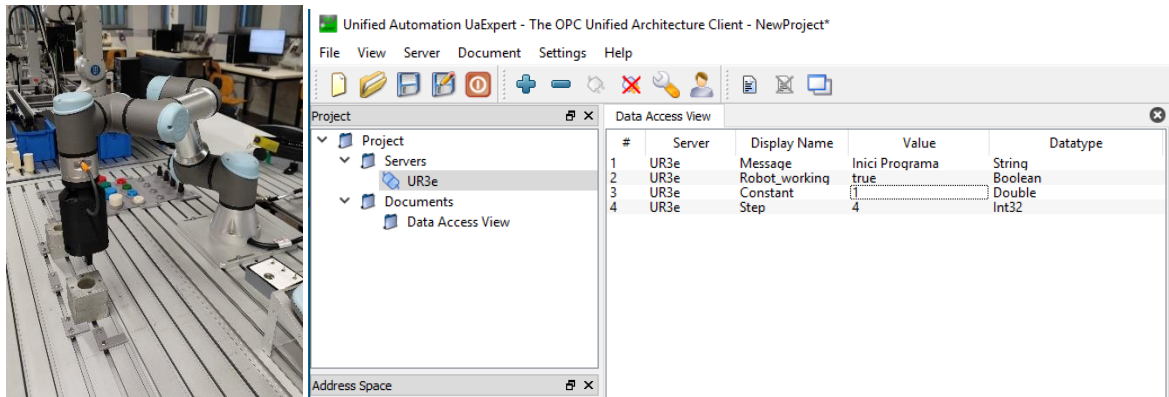


Figura 5.5. Robot en el moviment núm. 4 i pantalla del Client (Font: autor)

5.2. Programa amb interacció de l'operari

Els robots col·laboratius com el UR3e estan pensats per treballar de prop amb persones. Compten amb sistemes de seguretat i funcions que permeten “avisar” a l'operari que ha passat alguna incidència.

Fins ara s'han introduït les funcions d'OPC UA com a elements de suport (monitoratge). Tanmateix, es pot donar a OPC UA un rol més important i permetre que intercedeixi directament en el control del robot.

La segona aplicació que es va idear buscava la participació activa dels operaris en el funcionament del robot. Mitjançant el conjunt d'eines que el UR3e OPC UA posa a la nostra disposició, s'investigaria de quines maneres el programa pot demanar la participació directa d'un operari per concretar els següents passos del robot.

Es va decidir utilitzar el sistema d'avisos (finestres emergents, *PopUp* en anglès) de l'entorn PolyScope. Amb aquesta funcionalitat podem mostrar missatges en pantalla a mode d'avís/error o en format pregunta esperant una resposta per part del operari que es trobi amb el robot. La resposta es pot assignar llavors a una variable i actuar en funció del valor que tingui. Les respostes admeses poden ser: si/no, un número enter/decimal o bé una cadena de caràcters.

Per introduir una finestra emergent al programa es pot utilitzar la instrucció 'Popup' dins la categoria “Basic”. Es pot escriure un text i canviar la icona que l'acompanya. Les finestres es mostren amb dos botons: “OK” (per confirmar i seguir amb el programa) i “Parar el programa” (per si volem aturar-ho tot).

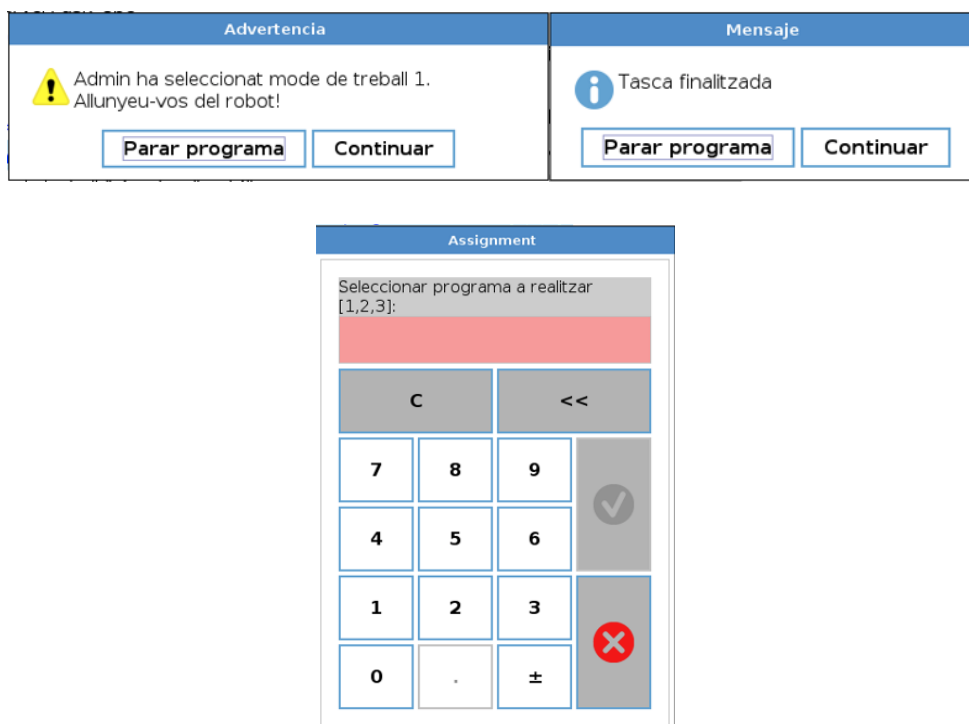


Figura 5.6. Exemples de finestres emergents utilitzades per al programa OPC_Feedback (Font: autor)

D'altra banda podem mostrar un missatge en pantalla amb una comanda 'Assignment', la qual es troba a la categoria 'Advanced'. En general aquesta instrucció s'utilitza per assignar i editar nous valors a variables locals, però si seleccionem 'Operador' en el desplegable 'Fuente' es pot observar que apareixen opcions que sol·liciten realimentació de l'operador. En aquesta pàgina podem seleccionar quin tipus de resposta s'espera, sobre quina variable es gravarà la resposta que es doni i finalment el missatge (la pregunta mostrada).

Aquestes són les eines que UR posa a la nostra disposició des del TeachPendant. I com ja s'ha vist en els anteriors apartats, també es compta amb les eines addicionals pròpies de l'entorn OPC UA. Un segon operari (que es trobi utilitzant el programa UAExpert o un altre Client) pot donar realimentació mitjançant l'opció d'editar variables directament com a Client. El segon programa treballa en aquesta via, demanant la intervenció tant de l'operari que es troba amb el robot com del que es troba a l'ordinador.

El programa resultant, que es mostra a la figura 5.8, està ideat per funcionar amb dos operaris: el que es troba al costat del robot i l'administrador que supervisa el procés des de l'ordinador. El programa té dues parts ben diferenciades: 'Before Start' i el 'Robot Program'. Pel seu funcionament són necessàries 7 variables:

- Message: variable OPC de tipus string. Ens permetrà mostrar l'estat del robot a l'administrador.
- ask_opc: variable OPC de tipus string que mostra una pregunta a l'administrador.
- input_opc: variable OPC de tipus Integer que registra la resposta de l'administrador.
- String: variable local de tipus string que edita el valor de Message.
- String2: variable local de tipus string que edita el valor de ask_opc.
- client_output: variable local de tipus Integer que replica el valor de input_opc.
- input_local: variable local de tipus Integer que permet registrar la resposta del operari.

De forma similar al primer programa, tenim 3 variables locals relacionades directament amb 3 variables OPC, és a dir, que estan emparellades. En cada parella, el valor que tingui una variable es reflexa a la seva companya. Disposem d'una quarta variable local a mode de suport necessària per guardar la resposta de l'operari.

El programa s'ha plantejat amb dos modes de treballs. Al entrar en qualsevol dels modes, apareix una finestra al TeachPendant per tal d'informar a l'operari dels següents passos del robot. A continuació es troba el diagrama de flux del programa OPC_Feedback:

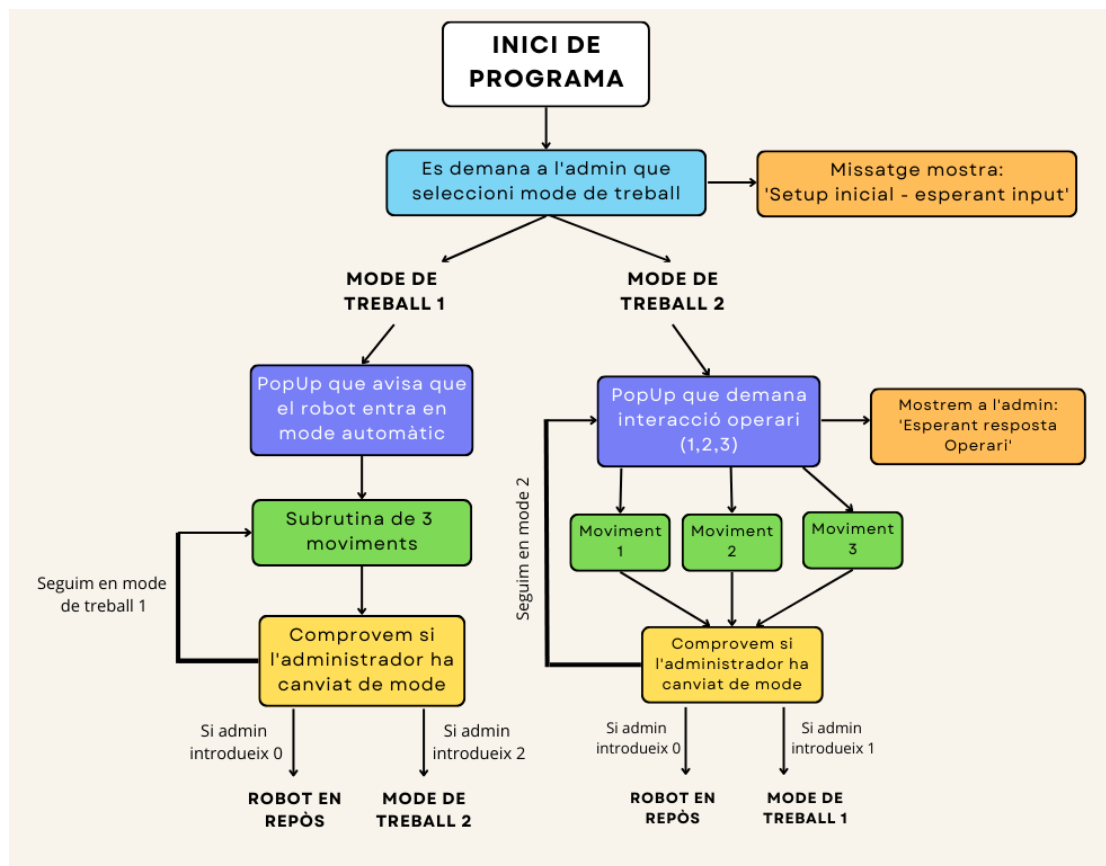


Figura 5.7. Diagrama de flux del programa OPC_Feedback (Font: autor)

En la primera secció 'BeforeStart' trobem la inicialització de la càrrega del robot (pinça buida), movem el robot a la posició inicial i assignem el valor dels missatges que es mostraran a l'Administrador mitjançant el programa UAExpert (o un dels altres Clients). A la línia 5 s'assigna un valor a String i a la 6 es sobreescriu aquest valor sobre la variable Message. A la línia 7 s'imposa una nova frase a String2 i la següent instrucció grava aquest valor sobre ask_opc. El resultat és que durant aquesta part inicial del programa es mostren a l'administrador els missatges: "Setup Inicial: esperant input" i se li fa la pregunta "Editar input_opc amb el mode de treball desitjat [1,2]".

Aquesta secció acaba amb un bucle que espera indefinidament a que la variable client_output (que té el mateix valor que input_opc) no valgui zero, que és el seu valor inicial. En altres paraules, el programa romandrà en espera en aquest bucle fins que client_output agafi un valor diferent a zero.

El següent bloc, 'Robot Program', és el cos del programa en si. La programació està feta per actuar davant 3 casos, depenent de si l'administrador imposa 0/1/2 a la variable 'input_opc'.

Si selecciona 1, s'ha plantejat el programa per a que el robot treballi de forma autònoma, sense intervenció de l'operari. S'avisava amb un 'PopUp' (línia 13) que el robot es mourà per tal d'informar a l'operari del moviment autònom. El robot no es començarà a moure fins que l'operari tanqui la finestra emergent. A la línia 14 assignem una nova frase a String que es mostrarà a l'administrador. En aquest mode autònom, el robot realitza la subrutina MT1 (Mode de treball 1) que passa per tres punts en bucle. A l'inici de cada bucle, llegeix la variable 'input_opc' per comprovar si l'administrador ha canviat de mode (línia 17).

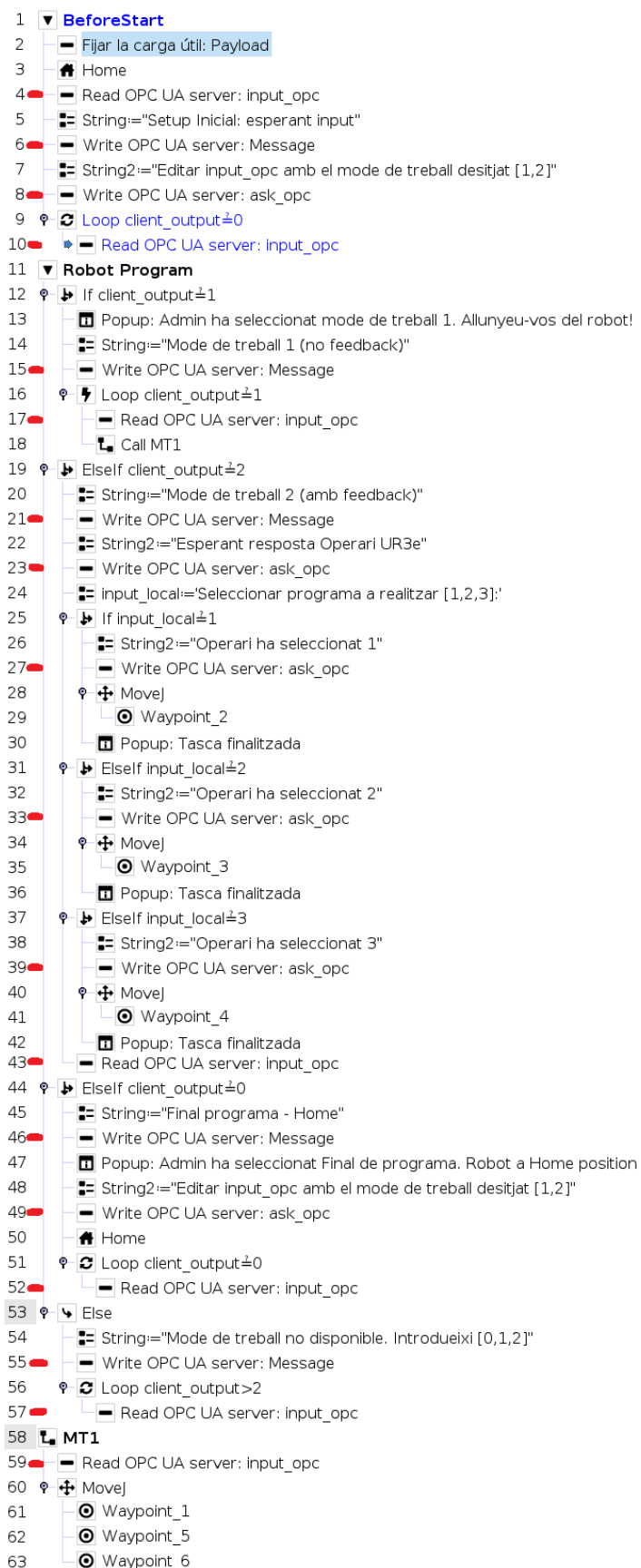


Figura 5.8. Programa OPC_Feedback complet (Font: Autor)

En cas que l'administrador imposi el valor 2 a la variable 'input_opc', es complirà el condicional de la línia 19 i el robot entra en un mode de treball on si que requereix l'operari. La idea d'aquest mode de treball és que operari i robot treballin junts en la confecció d'una peça hipotètica. A les línies 20 i 24 es comunica tant a l'operari com a l'administrador que el robot entra a treballar en el mode 2. Específicament, se li pregunta a l'operari quin moviment es vol realitzar (1,2,3). La idea és que l'operari ordeni el robot a quin punt anar depenent del que necessiti pel muntatge de la peça. El moviment 1 porta el robot a sobre de la peça, el moviment 2 el mou a un altra punt i, finalment, el moviment 3 envia al robot a un punt elevat que permet l'operari agafar la peça finalitzada.

Cada cop que el robot completa el moviment a un dels punts, apareix una finestra emergent que comunica a l'operari que el robot ha acabat la tasca. Posteriorment se li torna a preguntar quin moviment ha de fer.

En qualsevol moment l'administrador pot decidir finalitzar el cicle i parar el robot a la posició inicial, introduint un zero a la variable corresponent. De la mateixa manera pot decidir torna al mode de treball 2 introduint un dos a 'input_opc'.

En cas que introdueixi un número diferent a 1/2/3 se li comunica que no està contemplat a la programació (línia 54).

Aquest programa va servir per consolidar els coneixements en l'ús de les variables OPC UA i les del programa. De nou, es va confirmar que el programa amb funcions OPC UA resultava sent més llarg amb motiu de la introducció de les comandes de lectura/escriptura OPC UA (marcades en vermell). També va servir per descobrir les múltiples possibilitats del sistema de finestres emergents com a element per avisar als operaris de camp sobre els següents moviments que faria el robot. En escenaris on es treballa amb robots col·laboratius, és molt important que els operaris que es trobin prop dels robots siguin conscients del mode de treball que està realitzant i que el robot no es mogui sobtadament sense el seu coneixement. Com es va comprovar amb la realització d'aquest segon programa, OPC UA pot utilitzar-se per fer saltar finestres emergents que avisin als operaris d'informació important.

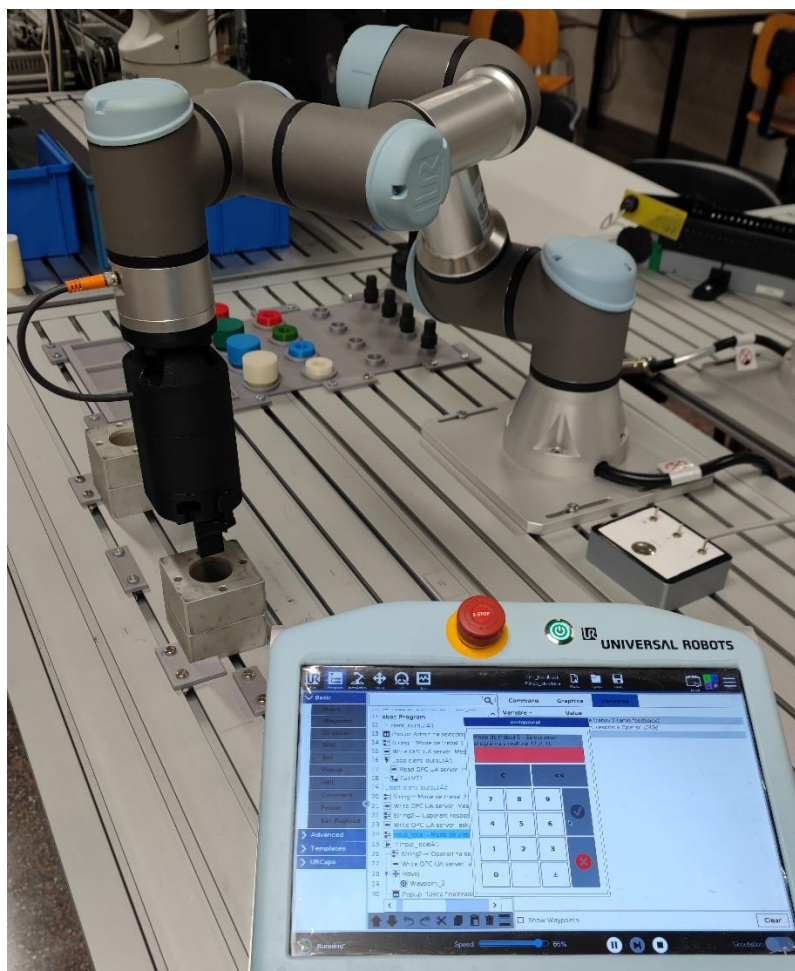


Figura 5.9. Robot esperant la resposta de l'operari mitjançant una finestra emergent. (Font: Autor)

Finalment, aquest programa va acabar de confirmar les possibilitats de dotar al Client amb un rol més encarat al control del robot. En casos que interressi, es pot assignar al Client com a Administrador, i preparar el programa del robot per acceptar diferents modes de treball. En una època on el subministrament de les peces i les matèries primes no està del tot garantit, és interessant treballar sota el concepte de 'Fabricació flexible', on adaptem el funcionament del robot a les necessitats puntuals de cada jornada laboral.

5.3. Assemblatge amb funcions OPC UA

La realització d'aquest tercer escenari representa l'etapa final del treball. En aquest punt, es disposava d'un bon nivell de coneixements del robot UR3e i de la comunicació OPC UA. Les múltiples proves i programes realitzats prèviament havien confirmat el potencial de treballar conjuntament amb el robot

UR3e i la tecnologia OPC UA, però també havien servit per descobrir algunes de les limitacions de la solució OPC UA que havia implementat Rocketfarm.

Per aquest estadi final del projecte, es va creure convenient realitzar un programa en el robot que servís de nexa de tots els coneixements assimilats i englobés els diversos paràmetres que han anat sorgint al llarg del treball. Es va proposar agafar un cas pràctic real: l'assemblatge que realitza el sistema FAS200 [48], present al mateix laboratori. Aquest sistema d'assemblatge flexible realitza el muntatge d'una peça que incorpora una base, un coixinet, un eix i una tapa, els quals s'han de muntar en un ordre concret. L'objectiu d'aquest tercer programa és replicar la funció de les estacions que conformen el sistema FAS200 amb el robot UR3e, i paral·lelament, interactuar amb la xarxa OPC UA. Les estacions realitzen el muntatge d'una peça que es compon per quatre elements. A la següent figura es poden observar els diferents passos necessaris per obtenir el muntatge final .

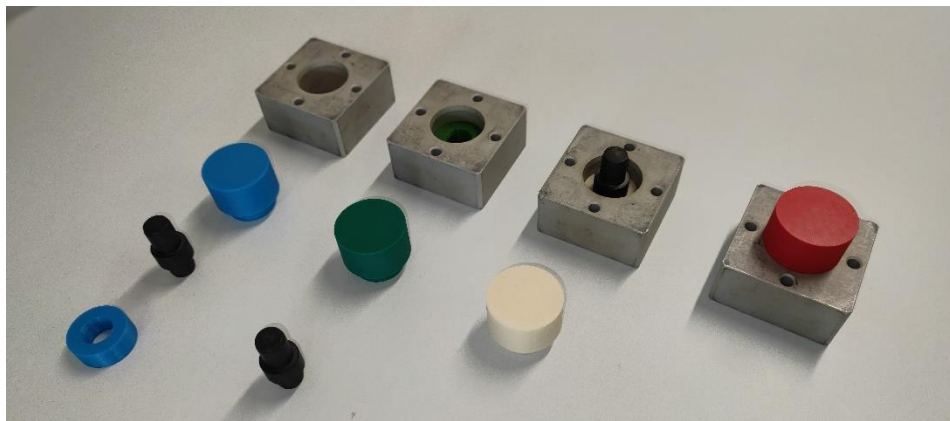


Figura 5.10. Procés d'assemblatge FAS200. Esquerra: peça inicial sense muntar, dreta: peça finalitzada. (Font: Autor)



Figura 5.11. Sistema d'assemblatge FAS200 al laboratori A5.4 (Font: Autor)

Per tal d'entendre aquest programa, degut a la seva complexitat i al nivell de dificultat superior envers als dos primers que s'han explicat, cal anar punt a punt.

En primer lloc es resumirà el plantejament del programa i el rol que té el robot i el personal. En aquest tercer cas, es treballa sota dues consignes: manteniment predictiu i fabricació flexible.

- **Manteniment predictiu:** un dels objectius secundaris d'aquest programa és proporcionar les eines necessàries per a que un administrador pugui saber quan temps porta el robot funcionant, quan ha durat el cicle i si cal un manteniment del robot. Aquestes dades són essencials per garantir un correcte funcionament del robot i l'eficiència de l'aplicació robòtica.
- **Fabricació flexible:** similar al cas del segon programa, per aquest tercer escenari es busca un grau de flexibilitat per tal d'adaptar el funcionament del robot a les característiques concretes de cada jornada. Donat que indústries a tot al món estan fent front a problemàtiques relacionades amb el subministrament de components, s'ha ideat aquest tercer cas per contemplar dos casos, un on es tenen disponibles els materials per realitzar dues peces cada cicle, i un segon cas on hipotèticament es té accés a més material i es pot doblar la producció fins a quatre peces per cicle.

El fet de treballar sota aquestes consignes fa que aquest tercer programa sigui força similar a escenaris reals que es poden trobar en les indústries actuals.

El rol del robot en aquest tercer escenari és clar: realitzar l'assemblatge de la peça final sempre i quan disposi del material necessari. S'ha plantejat aquest escenari per a que el robot tingui tres estats: en repòs, realitzant l'assemblatge de 2 unitats o realitzant l'assemblatge de 4 unitats, en funció del mode seleccionat. El rol de l'administrador, a qui entenem com la persona que es troba a la sala de control monitoritzant tota la planta, serà justament decidir en quin mode de treball es troba el robot. L'administrador hauria de tenir els coneixements i la informació necessària per saber si el robot ha d'estar en repòs, realitzant 2 unitats per cicle o si han entrat noves existències i pot passar a muntar 4 unitats per cicle. També recau en ell la feina de monitoritzar el procés i totes les dades que proporcioni el robot. Finalment, l'operari de planta serà el responsable d'encendre i apagar el robot i addicionalment, serà l'encarregat de confirmar mitjançant el *TeachPendant* que el robot disposa de les peces necessàries per iniciar un nou cicle, tant per 2 com per 4 unitats.

En aquest escenari, la planta de producció treballa amb una xarxa OPC UA. El servidor es configurarà l'única combinació de seguretat recomanada (és a dir, amb autenticació, signatura i xifrat en nivell Basic256Sha256). Per el funcionament del programa calen 8 variables OPC:

- **Robot_Working:** variable booleana Vertader/False que indica si el robot està en repòs o treballant. La seva parella local s'anomena 'working'.
- **Step:** variable entera que emmagatzema el pas en el que es troba el programa. La seva parella local s'anomena 'pas'.
- **Units_completed:** variable entera que compta quantes unitats s'han acabat. La seva parella local s'anomena 'units'.
- **Working_mode:** variable entera que conté el valor (0,1,2) del mode de treball a realitzar. La seva parella local s'anomena 'control'.
- **ask_admin:** variable de tipus string que serveix per demanar una acció a l'administrador. La seva parella local s'anomena 'string_admin'.
- **Message:** variable de tipus string que s'utilitza per mostrar una frase a l'administrador. La seva parella local s'anomena 'string'.
- **cycle_time:** variable de tipus Double que compta els segons que dura el cicle de muntatge. La seva parella local s'anomena 'temps_cicle'.
- **session_time:** variable de tipus Double que compta els segons que porta encès el robot, des de que s'inicia el programa. La seva parella local s'anomena 'temps_on'.

El programa fa servir addicionalment una llum d'estat. Aquesta llum LED disposa de diferents combinacions de colors i es troba com a sortida digital del robot. Abarca quatre sortides digitals: DO 0, DO 1, DO 2 i DO 3. Depenent de quines estiguin en estat alt o baix, es veu un color o un altre.

Els codis de colors, d'acord amb el document del INSHT [49] (Instituto Nacional de Seguridad e Higiene en el Trabajo), s'ha assignat de la següent manera:

- Blanc: indica robot en repòs. (Enceses les sortides digital 1,2 i 3)
- Verd: el robot es troba e funcionament i s'està movent. (S'activa DO2)
- Blau: es requereix intervenció de l'operari. (S'activa DO3)
- Groc: indica avís/alarma. (S'activa DO1)
- Vermell: falla, avaria, situació d'emergència PE. (S'activa DO 0)

El funcionament del programa és el següent: inicialment, configurem el valor inicial d'algunes de les variables (per exemple: Units_completed, Work_mode i temporitzadors es posen a zero). Confirmem que la llum es troba apagada i la pinça oberta. En començar el programa s'inicia el temporitzador 'temps_on'. Es confirma que el robot es trobi a la posició inicial i es presenta a l'administrador l'opció de triar mode de treball 1 o 2. El robot entrarà sempre en estat de repòs degut a que la variable control s'inicialitza a zero, fet que fa entrar el programa en el primer bucle. En aquest bucle s'encén la llum d'estat en blanc, indicant repòs. El bucle correspon al pas 2. L'únic que fa el robot en aquest bucle és llegir el valor de work_mode per saber si l'administrador ha donat una nova ordre.

En cas que es seleccioni des del Client work_mode 1, es crida una subrutina que construeix dos peces completes. Si es selecciona work_mode 2, el robot fa l'assemblatge de quatre unitats completes.

La figura 5.12 mostra el codi complet amb les subrutines minimitzades mentre que la 5.13 hi apareix una subrutina detallada. Hi han algunes funcions que cal destacar dins d'aquest subprogrames. Quan aquestes s'inicien, assignem Verdader a la variable 'Robot_working'. Seguidament, encenem el llum d'estat en blau i demanem a l'operari que confirmi que el robot té totes les peces per muntar 2 o 4 unitats, depenent del mode de treball. El pas en que s'està esperant la resposta de l'operari correspon al pas 3. Se li comunica a l'administrador que el robot està en espera mitjançant 'string'. En cas que l'operari confirmi tancant la finestra emergent, el robot inicia el moviment (pas = 4) i posa la llum en verd. En el moment que es comença a moure, també s'inicia el comptador de cicle, el qual finalitza quan el robot acaba el cicle i està a la posició inicial. El comportament de la segona subrutina és idèntic, però es fa el muntatge de 4 unitats finals.

Abans d'acabar la subrutina, es llegeix la variable 'Work_mode' per comprovar si durant l'execució l'administrador ha decidit canviar de mode. S'ha decidit que encara que l'administrador modifiqui el mode de treball durant el cicle, el robot sempre acabarà la tasca que es troba realitzant abans de canviar de mode.

El programa finalitza amb un fil de programació paral·lel (*thread*) al programa principal (línies 357-363). Aquestes línies s'executen al mateix temps que el programa i són les comandes OPC UA que sobreescriven el valor de les variables locals sobre les OPC. En els altres dos programes explicats anteriorment (Bucle i OPC_Feedback) es desconeixia aquesta funció present a la programació d'UR i calia introduir una nova comanda d'escriptura OPC UA en cada punt que es vulgues relacionar una variable local amb la seva parella OPC. En aquest estadi del projecte es va poder comprovar que la introducció d'aquest fil paral·lel al final del programa aconsegueix el mateix objectiu però redueix molt l'impacte en la llargada del programa. Aquesta millora és signe de la maduresa dels coneixements que es tenien a l'etapa final del treball, els quals van permetre optimitzar i fer més eficient el codi final.

D'ara endavant, doncs, es recomana sempre la utilització de fils (*threads*) en programes OPC UA per dur a terme les comandes 'Write' ja que disminueixen considerablement el número de les noves línies de codi necessàries per treballar amb OPC UA.

En el cas de la lectura, 'Read', es va confirmar que no és possible realitzar-la dins un fil paral·lel i cal seguir posant les comandes en el precís moment que calgui comprovar quin valor té una variable OPC.

```

! FAS200_OPC.yml
1   Program
2   Init Variables
3   Robot Program
4   pas:=0
5   temps_cicle: Restablecer
6   Write OPC UA server: cycle_time
7   temps_on: Iniciar
8   Set DO[2]=Apagar
9   Set DO[3]=Apagar
10  Set Gripper=Apagar
11  Read OPC UA server: Work_mode
12  string:="Setup inicial"
13  MoveJ
14  pas:=1
15  Pos_Ini
16  string_admin:="Select the desired working mode [1,2]"
17  Loop control±0
18  Set DO[0]=Apagar
19  Set DO[1]=Encender
20  Set DO[2]=Encender
21  Set DO[3]=Encender
22  string:="Mode 0 - Robot in standby"
23  pas:=2
24  Read OPC UA server: Work_mode
25  Loop control±1
26  string:="Working_mode 1 - Assembly of 2 units"
27  string_admin:="Write 0 in Work_mode to finish task"
28  Call SubProgram_2A
29  Read OPC UA server: Work_mode
30  Loop control±2
31  string:="Working_mode 2 - Assembly of 4 units"
32  string_admin:="Write 0 in Work_mode to finish task"
33  Call SubProgram_2A
34  temps_cicle: Iniciar
35  Call SubProgram_2B
36  Read OPC UA server: Work_mode
37 > SubProgram_2A...
207 > SubProgram_2B...
356 Thread_1
357 Write OPC UA server: Robot_working
358 Write OPC UA server: Step
359 Write OPC UA server: Units_completed
360 Write OPC UA server: cycle_time
361 Write OPC UA server: session_time
362 Write OPC UA server: ask_admin
363 Write OPC UA server: Message

```

Figura 5.12. Extracte del programa FAS200_OPC (Font: Autor)

```
37 SubProgram_2A
38   Read OPC UA server: Work_mode
39   working:= True
40   string:="Waiting for operator to refill the required pieces..."
41   pas:=3
42   Set DO[0]=Apagar
43   Set DO[1]=Apagar
44   Set DO[2]=Apagar
45   Set DO[3]=Encender
46   If control±1
47     | Popup: Robot will engage movement in Mode 1. Confirm that it has the required modules available:
48   Else
49     | Popup: Robot will engage movement in Mode 2. Confirm that it has the required modules available:
50   Set DO[1]=Apagar
51   Read OPC UA server: Work_mode
52   If control±1
53     | string:="Working_mode 1 - Assembly of 2 units"
54   ElseIf control±2
55     | string:="Working_mode 2 - Assembly of 4 units"
56   Else
57     | Read OPC UA server: Work_mode
58     temps_cicle: Restablecer
59     temps_cicle: Iniciar
60     Set DO[0]=Apagar
61     Set DO[1]=Apagar
62     Set DO[3]=Apagar
63     Set DO[2]=Encender
64     MoveJ
65     | pas:=4
66     | Pos_Ini
67     | Set Gripper=Encender
68     | Wait: 1.0
69     MoveJ
70     | PrePick_1a
200   MoveJ
201   | Pos_Ini
202   | units:=units+2
203   | Set DO[2]=Apagar
204   | working:= False
205   | temps_cicle: Detener
206   | working:= False
```

Figura 5.13. Extracte de la subrutina del programa FAS200_OPC (Font: Autor)

Monitoratge de les variables OPC UA durant l'assemblatge:

El ventall de variables que interactuen amb el programa es poden visualitzar mitjançant un o més clients. Una bona combinació és l'ús simultani de Prosys UA Monitor amb UA Expert. El primer serveix per a tenir una finestra de visualització més gràfica i UA Expert destaca per la informació addicional que proporciona tant de les variables com del Servidor. En ambdós pantalles les variables s'actualitzen en temps real.

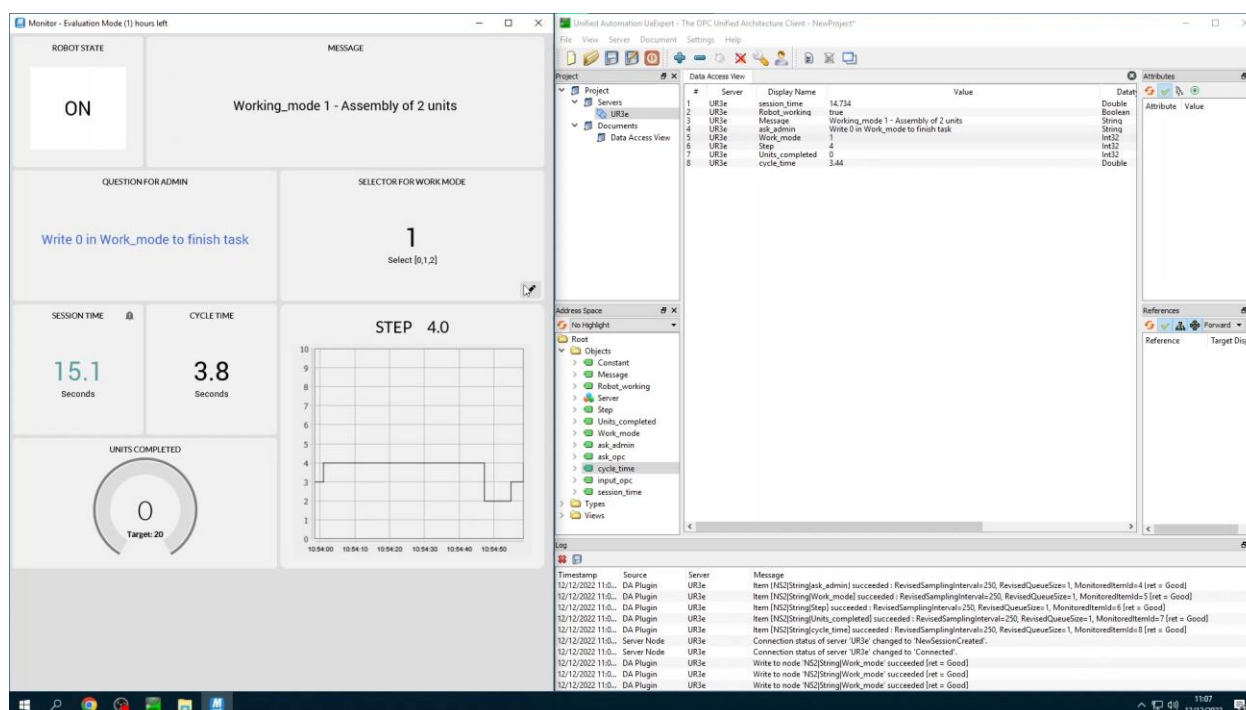


Figura 5.14. Monitoratge del programa Assemblatge FAS200 amb OPC UA (Font: Autor)

S'ha configurat Prosys UA Monitor per avisar a l'administrador quan el robot porti encès un cert temps. Quan aquest el temps de la sessió supera un determinat valor configurable, saltarà una alarma informant que cal realitzar manteniment.

El gràfic present a la finestra de Prosys UA Monitor indica el temps que ha estat el robot en cada pas. Es pot detectar si el robot porta molta estona esperant la intervenció de l'operari o altres paràmetres d'eficiència.

L'administrador pot modificar el mode de treball o bé des de UA Expert o directament des de la pantalla de Prosys UA Monitor. Aquest comportament es totalment configurable, és a dir, que es pot dissenyar la pantalla de Prosys per a que cap variable sigui modificable des del Client.

A continuació hi ha un recull d'imatges del robot treballant en el programa FAS200_OPC. La imatge 1 correspon a l'estat inicial del robot, en repòs i sense cap mode de treball seleccionat. La fotografia número 2 pertany al cas en que s'ha seleccionat mode de treball 1 i el robot inicia el muntatge de dues peces (llum d'estat en color verd). La imatge 3 correspon al moment on el robot acaba el muntatge de dues unitats i demana interacció a l'operari per confirmar que pot seguir amb la tasca (llum d'estat en color blau i finestra emergent al *TeachPendant*). Finalment, la imatge 4 és del punt final on el robot es troba en repòs després d'haver fet l'assemblatge de quatre unitats (llum d'estat en blanc).

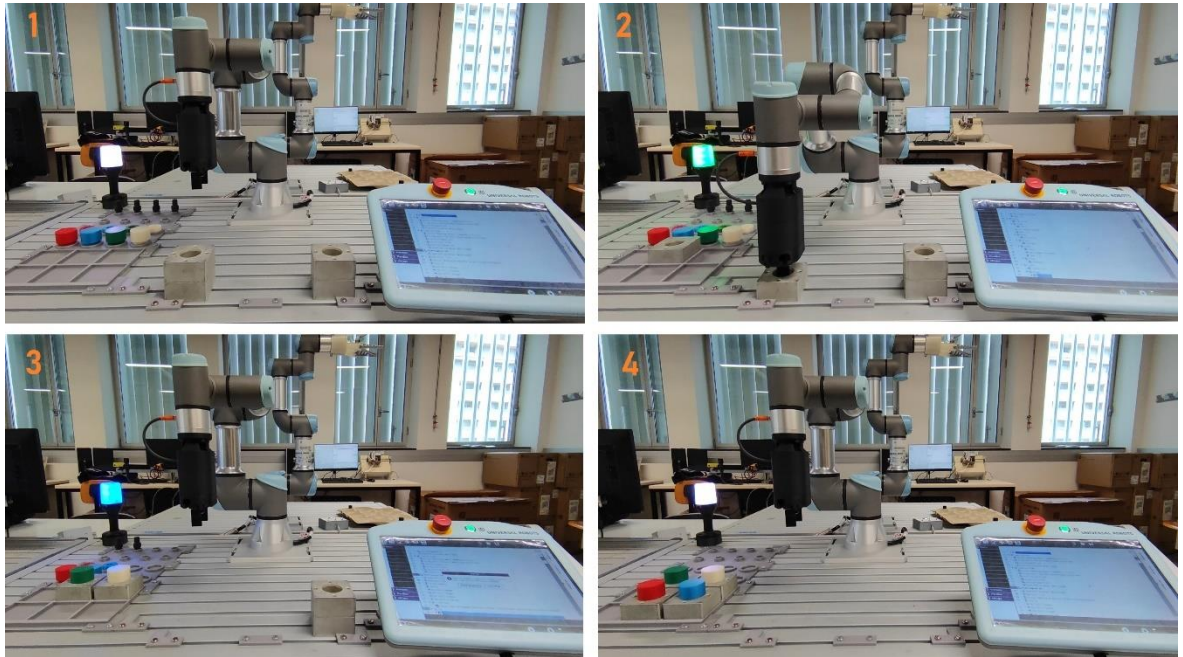


Figura 5.15. Recull d'imatges del robot fent el programa FAS200_OPC (Font: Autor)

L'ordinador que apareix a la part del darrere de les imatges és l'ordinador que disposava dels Clients i des d'on es realitzava la monitoratge. Es disposava d'una pantalla igual que en la figura 5.6 funcionat amb Prosys UA Monitor i UA Expert. És interessant visualitzar que es mostrava per pantalla en els quatre casos de la figura 5.7:

- Cas 1: mostrem en pantalla que el robot es troba en repòs i que el mode seleccionat és zero. S'indica a l'administrador que pot triar el mode en el selector corresponent. El comptador d'unitats completades està a zero. S'indica en quin pas es troba el robot (pas 2 correspon a l'estat en repòs).
- Cas 2: el procés d'assemblatge s'ha iniciat i el robot es troba en estat 'ON'. S'indica a l'administrador que el mode de treball és 1 i que pot finalitzar el muntatge introduint un zero en el camp adequat. El comptador d'unitats completades està a zero. S'indica pas 4.

- Cas 3: s'han completat dues unitats, el cycle ha durat 60.6 segons i s'indica a l'administrador que en aquest moment el robot està esperant la confirmació de l'usuari per a seguir amb la tasca. (Quan el robot espera una intervenció de l'usuari correspon al pas 3).
- Cas 4: El comptador d'unitats indica 6 unitats completades. El robot ha treballat en mode dos i el cycle complet ha durat 130.3 segons. Porta 227 segons encès i es troba en repòs, és a dir, mode de treball 0.

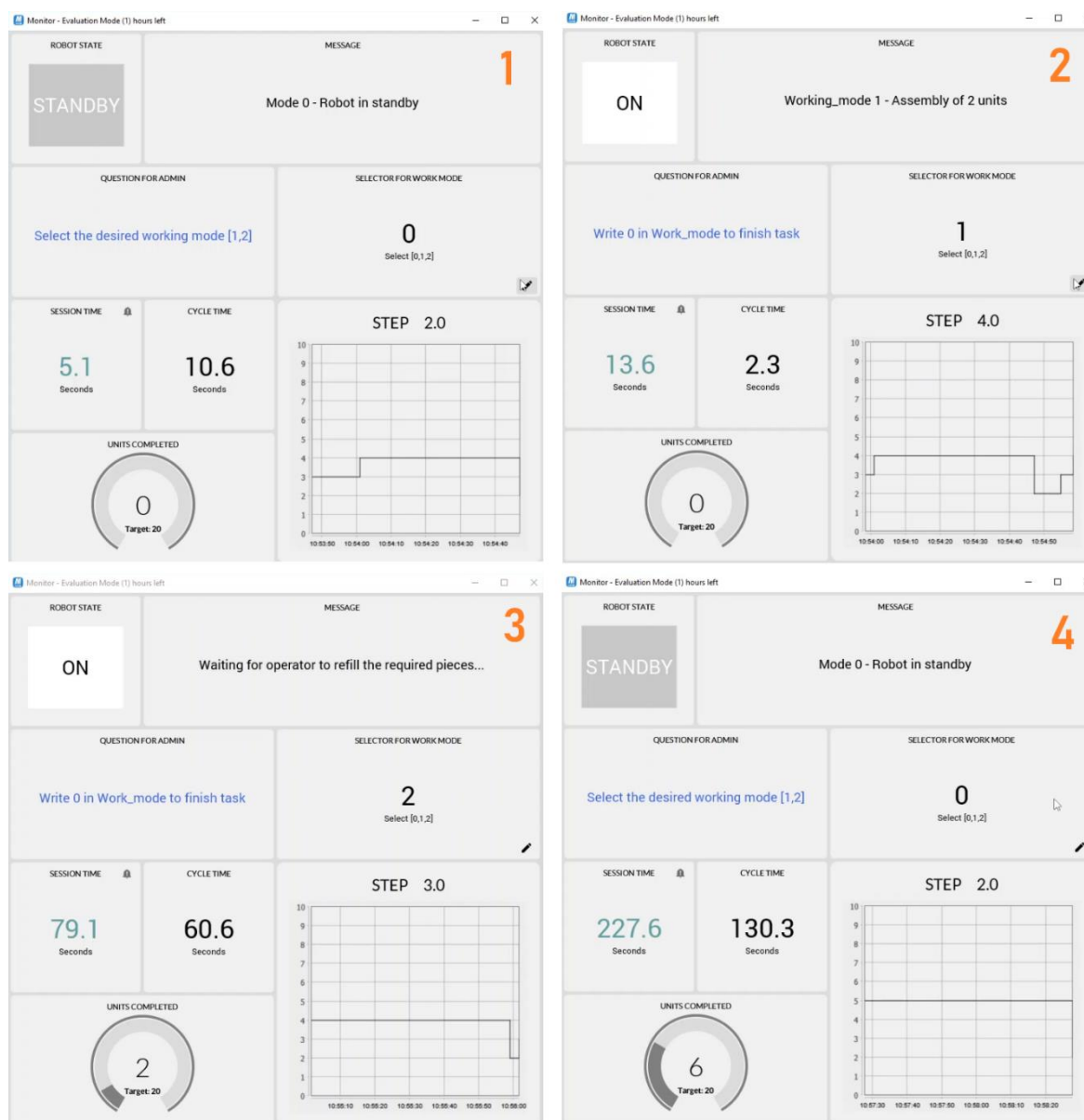


Figura 5.16. Monitorització dels quatre casos de la figura 5.9 (Font: Autor)

El programa també té configurades una sèrie d'alarmes i missatges d'avís que l'administrador pot personalitzar per rebre un missatge quan el robot superi un cert valor de temps a la variable

'Session_Time'. A la figura 5.9 es pot veure la pantalla de notificacions que mostra un missatge d'exemple quan el robot supera 500 segons i 600 segons. Aquest sistema es pot fer servir, com ja s'ha comentat, per coordinar el manteniment del robot amb el departament corresponent. Cada usuari pot personalitzar el nombre d'alarmes i avisos depenent de l'aplicació i l'escenari on es treballi.

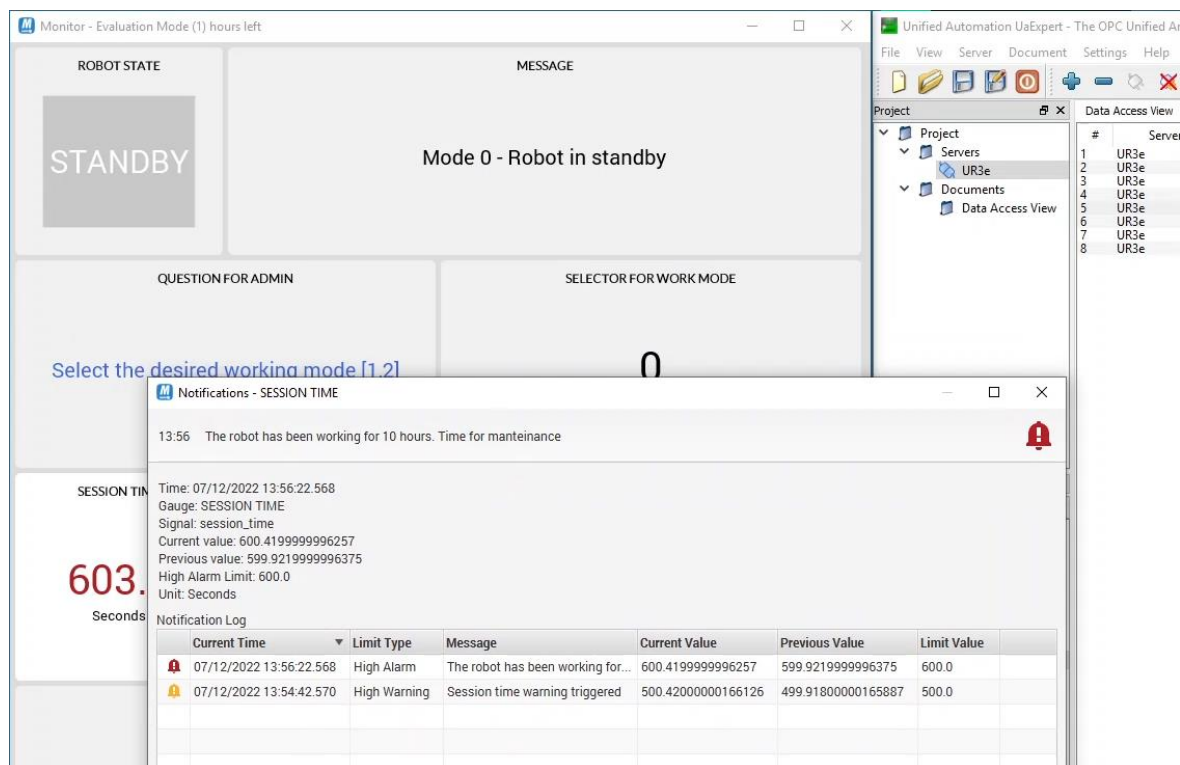


Figura 5.17. Sistema d'alarmes i avisos de Prosys UA Monitor (Font: Autor)

Amb la realització d'aquest tercer i últim programa es va poder confirmar el gran potencial que té OPC UA pel que fa a tasques de monitoratge. La combinació de dos Clients gratuïts i fàcils de configurar van permetre tenir les pantalles mostrades en les figures anteriors en qüestió de dies. Cal destacar també que, degut al elevat grau de personalització i flexibilitat de UA Expert i especialment Prosys UA Monitor, aquest tercer programa està preparat per si en el futur calgués monitoritzar noves variables provinents del robot o altres dispositius (sensors, cintes etc.).

Els exemples de monitoratge presents en aquests apartats representen una petita fracció de totes les possibilitats que ofereix OPC UA en escenaris on es vulgui monitorar informació d'interès. En un futur TFE on es posi el focus en aquests casos de monitoratge es podrien investigar molt més a fons totes les possibilitats.

Anàlisi de l'impacte ambiental

Donat que l'abast del meu treball es centra en la posada en marxa d'un programari en un robot, l'impacte ambiental d'aquest és una variable difícil de quantificar.

No obstant, es pot destacar que el robot UR3e compta amb unes certificacions que poden tenir un impacte ambiental positiu. Per exemple es pot destacar que el robot compleix la directiva *RoHS* 2011/65/EU [50] i compta amb la certificació corresponent [51] per garantir que no conté certes substàncies perilloses tant per les persones com pel medi ambient. Aquestes substàncies inclouen mercuri, cadmi i plom, entre altres.

Adicionalment, els models d'Universal Robots formen part de la legislació referent a WEEE [52] (Residus d'Aparells Elèctrics i Electrònics). Aquesta directiva treballa en l'àmbit Europeu per tal de minimitzar la creació de residus electrònics i per promoure l'ús eficient dels recursos. També dicta unes pautes per el correcte reciclatge d'aquests residus. Al manual d'usuari [51], UR insta als clients a reciclar els robots i els altres components en acabar la seva vida útil d'acord amb la legislació aplicable a cada país.

Finalment destacar l'impacte positiu que pot arribar a tenir la implementació del protocol OPC UA en plantes de producció actuals. Com s'ha pogut comprovar, es pot dissenyar un sistema que retransmeti dades d'interès com el temps de funcionament i el temps en repòs. Aquestes variables, que s'engloben sota la consigna de manteniment predictiu, poden servir per donar eines als enginyers per adaptar la producció i apagar els robots quan no hagin de treballar, reduint l'impacte energètic. Un bon manteniment assegura que els robots i els altres dispositius puguin funcionar correctament tota la seva vida útil, evitant així la compra de nous models i la generació de residus.

OPC UA pot ser una eina interessant per tenir més dades a la nostra disposició i poder prendre decisions que s'encaminin a reduir la petjada ambiental.

Conclusions

Al llarg d'aquest treball de fi d'estudis s'ha exposat amb tot detall la posada en marxa d'una xarxa OPC UA en un robot col·laboratiu. El primer objectiu del treball, doncs, s'ha complert satisfactòriament: la xarxa OPC UA és totalment funcional i el desplegament d'aquesta tecnologia ha quedat degudament documentat en els capítols d'aquesta memòria.

Per tal d'estudiar i fer-se una idea del possible impacte que podia tenir la combinació del protocol OPC UA i el robot UR3e, s'han realitzat una sèrie de programes en el robot que exemplifiquen diferents escenaris actuals on es podria implementar aquesta tecnologia. El tercer programa, en el qual es realitza un assemblatge d'una peça, actua com a recull de tots els elements presentats al llarg del treball i confirma que s'han assimilat els coneixements i les aptituds bàsiques necessàries per dissenyar aplicacions robòtiques que interactuïn amb la tecnologia OPC UA.

Però entendre i saber aplicar els paràmetres de la xarxa OPC UA era només un dels objectius del treball. Amb la creació de la 'Guia ràpida per la configuració d'una xarxa OPC UA en robots UR' (disponible com a Annex B) es posa a l'abast de la comunitat acadèmica i científica un manual amb informació detallada sobre la configuració i l'ús d'aquesta tecnologia en robots de la marca Universal Robots. Tot i estar enfocada en aquest escenari específic, la estandardització del protocol hauria de permetre generalitzar la majoria de conceptes i paràmetres per tal de poder ser utilitzats en altres escenaris OPC UA. Aquesta és una de les forteses d'aquesta tecnologia que s'ha confirmat per exemple en l'apartat de configuració dels Clients: tot i estar desenvolupats per diferents empreses, els tres programaris estudiats compten amb una configuració molt similar i, en la majoria de casos, els paràmetres tenien els mateixos noms. Addicionalment es pot destacar que part de la informació i tests realitzats en aquest projecte han quedat integrats en una pràctica docent de l'assignatura ISA.

OPC UA és realment una tecnologia amb molt potencial. En poques setmanes, es va tenir un Servidor OPC UA actiu i l'entesa dels diferents atributs de la tecnologia va ser relativament fàcil. És un protocol modern que compta amb suport dels seus desenvolupadors per preparar-lo de cara al futur. El focus en seguretat i en l'àmplia compatibilitat de dispositius sense importar el fabricant fa que sigui un protocol molt interessant per comunicar robots i altres dispositius amb les sales de control.

No obstant, degut en part al temps disponible i a la poca documentació que es tenia disponible per l'escenari concret del treball (URCap OPC UA en el robot UR3e), tinc la sensació que només hem observat la punta del iceberg. El protocol OPC UA és només una eina, però recau en els enginyers la tasca de pensar aplicacions que apliquin tot el seu potencial. Els tres programes exposats en el treball són bons exemples de les característiques generals de la tecnologia i deixen entreveure l'impacte que pot tenir.

Per acabar l'apartat de les conclusions, es fan tres propostes sobre aspectes que, havent tingut més temps i més recursos, s'haguessin pogut investigar. Cal destacar, però, que donat la seva complexitat tècnica podrien investigar-se per separat en futurs treballs de fi d'estudis:

- Funcionalitats específiques de la nova versió del URCap VDMA de l'empresa Rocketfarm (monitoratge de variables del robot com posició/velocitat/acceleració de les articulacions, temperatura etc.).
- L'ús de scripts avançats per completar les possibilitats de la xarxa OPC UA (i interactuar amb altres programaris d'ordinador com per exemple Python o RoboDK).
- Una aplicació robòtica que coordini els dos robots UR3e que es disposa ara al laboratori juntament amb la tecnologia OPC UA.

Anàlisi Econòmica

Per a realitzar les diverses investigacions exposades al llarg del treball, s'han fet ús de programaris i maquinari específic que va suposar un cost en el moment de la seva compra per la Universitat Politècnica de Catalunya.

En aquest apartat s'engloben tots aquells elements que es poden incloure dins l'anàlisi econòmica d'aquest treball de fi d'estudis.

El cost del programari que habilita la comunicació OPC UA en el robot UR va costar 800€ en la data de la seva compra. El robot UR3e, per la seva banda, va requerir una inversió de 18.000€ (sense IVA). Aquest és el cost dels dos elements protagonistes del projecte.

Cal destacar que la pinça elèctrica, que és el resultat del TFE del company Antoni Ferriol [26], té un cost de producció de 373,32€ tal com es va calcular en l'apartat corresponent del seu treball. S'ha considerat aquest valor per el càlcul del cost total.

Adicionalment a les despeses mencionades, en la propera taula s'afegeix el cost pertinent a elements auxiliars que també s'han utilitzat:

Taula 2. Llista de materials i costos associats.

Unitats	Concepte	Preu Unitari	Preu Total
1	Robot UR3e	18.000,00€	18.000,00€
1	URCap OPC UA (Rocketfarm)	800,00€	800,00€
1	Pinça elèctrica de dos dits	373,32€	373,32€
1	Taula d'alumini	1200,00€	1200,00€
8	Set de peces impreses en 3D	4,00€	32,00€
Subtotal			20.405,32€
I.V.A (21%)			4.285,12€
Cost total			24.690,44€

A continuació es detallen els costos dels recursos humans que ha comportat la realització de tot el projecte. Per la variable del preu/hora s'ha utilitzat un valor de referència de la remuneració actual d'un enginyer de robòtica (18€/hora aproximadament).

Taula 3. Despeses recursos humans.

Concepte	Hores	Preu/hora	Preu total
Investigació	180	18€	3.240,00€
Posada en marxa de la tecnologia OPC UA	372	18€	6.696,00€
Documentació	55	18€	990,00€
Total hores	607		10.926,00€

Finalment, el cost total del projecte resultant de sumar els valors del cost de materials i dels recursos humans és de **35.616,44€**.

Bibliografia

- [1] “The Future is Already Here – OPC Connect.” <https://opconnect.opcfoundation.org/2015/06/the-future-is-already-here/> (accessed Jan. 03, 2023).
- [2] “Visualización de datos de un simulador OPC UA.” <https://upcommons.upc.edu/handle/2117/112435> (accessed Jan. 03, 2023).
- [3] “Design and implementation of a self-response tool for logistic platforms.” <https://upcommons.upc.edu/handle/2117/192430> (accessed Jan. 03, 2023).
- [4] “Modelatge de l'emergència climàtica amb NECADA.” <https://upcommons.upc.edu/handle/2117/343215> (accessed Jan. 03, 2023).
- [5] “History - OPC Foundation.” <https://opcfoundation.org/about/opc-foundation/history/> (accessed Dec. 19, 2022).
- [6] “Home Page - OPC Foundation.” <https://opcfoundation.org/> (accessed Dec. 22, 2022).
- [7] “Classic - OPC Foundation.” <https://opcfoundation.org/developer-tools/specifications-classic/data-access/> (accessed Dec. 19, 2022).
- [8] “Classic - OPC Foundation.” <https://opcfoundation.org/developer-tools/specifications-classic/alarms-and-events/> (accessed Dec. 19, 2022).
- [9] “Classic - OPC Foundation.” <https://opcfoundation.org/developer-tools/specifications-classic/historical-data-access/> (accessed Dec. 19, 2022).
- [10] “OPC Security Specification Protects Sensitive Information.” <https://www.opcti.com/OPC-Security-specs.aspx> (accessed Dec. 19, 2022).
- [11] “Classic - OPC Foundation.” <https://opcfoundation.org/developer-tools/specifications-classic/xml-data-access/> (accessed Dec. 19, 2022).
- [12] “Unified Architecture - OPC Foundation.” <https://opcfoundation.org/about/opc-technologies/opc-ua/> (accessed Dec. 19, 2022).
- [13] “How to Certify - OPC Foundation.” <https://opcfoundation.org/certification/how-to-certify/> (accessed Dec. 22, 2022).
- [14] W. Mahnke, S. H. Leitner, and M. Damm, “OPC unified architecture,” *OPC Unified Archit.*, pp. 1–339, 2009, doi: 10.1007/978-3-540-68899-0/COVER.
- [15] “Estandarización y seguridad en el protocolo OPC UA | INCIBE-CERT.” <https://www.incibe-cert.es/blog/estandarizacion-y-seguridad-el-protocolo-opc-ua> (accessed Dec. 22, 2022).
- [16] S. A. Cooperates, T. I. A. Portal, O. P. C. Ua, and O. P. C. Scout, “Documentación didáctica / para

cursos de formació,” 2019.

- [17] “Practical Security Recommendations for building OPC UA Applications Whitepaper Security Working Group,” 2018.
- [18] “Practical Security Guidelines for Building OPC UA Applications – OPC Connect.” <https://opcconnect.opcfoundation.org/2018/06/practical-security-guidelines-for-building-opc-ua-applications/> (accessed Dec. 19, 2022).
- [19] “Escola d’Enginyeria de Barcelona Est. EEBE — UPC. Universitat Politècnica de Catalunya.” https://eebe.upc.edu/ca?set_language=ca (accessed Dec. 19, 2022).
- [20] “Robots Colaborativos | Universal Robots.” <https://www.universal-robots.com/es/> (accessed Dec. 19, 2022).
- [21] “Brazo Robótico UR3e | Universal Robots.” <https://www.universal-robots.com/es/productos/robot-ur3/> (accessed Dec. 19, 2022).
- [22] “Acerca de Universal Robots [Nuestra historia].” <https://www.universal-robots.com/es/acerca-de-universal-robots/nuestra-historia/> (accessed Dec. 19, 2022).
- [23] “Modelos de Robots Industriales | Universal Robots.” <https://www.universal-robots.com/es/plus/productos/> (accessed Dec. 19, 2022).
- [24] “Universal Robots - Launch of URCaps: The new platform for UR accessories and peripherals.” <https://www.universal-robots.com/es/acerca-de-universal-robots/noticias/launch-of-urcaps-the-new-platform-for-ur-accessories-and-peripherals/> (accessed Dec. 19, 2022).
- [25] “TÜV Rheinland - Página principal | ES | TÜV Rheinland.” <https://www.tuv.com/spain/es/> (accessed Dec. 22, 2022).
- [26] “Disseny i prototipat d’una pinça elèctrica per robot col·laboratiu.” <https://upcommons.upc.edu/handle/2117/374650> (accessed Dec. 22, 2022).
- [27] “UR3e Technical Specifications,” no. July, p. 2021, 2021.
- [28] “Using the Modbus TCP Client Interface of the UR Robot Original instructions (EN),” 2009.
- [29] “Desmontando Modbus | INCIBE-CERT.” <https://www.incibe-cert.es/blog/desmontando-modbus> (accessed Dec. 19, 2022).
- [30] “Evolving towards secure Modbus | INCIBE-CERT.” <https://www.incibe-cert.es/en/blog/evolving-towards-secure-modbus> (accessed Dec. 19, 2022).
- [31] “profibus.com - www.profibus.com.” <https://www.profibus.com/> (accessed Dec. 22, 2022).
- [32] “Profinet Guide - 20596.” <https://www.universal-robots.com/articles/ur/interface-communication/profinet-how-to-guide-e-series/> (accessed Dec. 19, 2022).
- [33] “Características y seguridad en PROFINET | INCIBE-CERT.” <https://www.incibe->

- cert.es/blog/caracteristicas-y-seguridad-profinet (accessed Dec. 19, 2022).
- [34] “Download - www.profibus.com.” <https://www.profibus.com/download/profinet-security-guideline> (accessed Dec. 27, 2022).
- [35] “Ethernet IP guide - 18712 - collaborative robots support.” <https://www.universal-robots.com/articles/ur/interface-communication/ethernet-ip-guide/> (accessed Dec. 19, 2022).
- [36] “Protocolo EtherNet/IP: analizando sus comunicaciones y medidas de seguridad | INCIBE-CERT.” <https://www.incibe-cert.es/blog/protocolo-ethernetip-analizando-sus-comunicaciones-y-medidas-seguridad> (accessed Dec. 22, 2022).
- [37] “UR+ | MQTT Connector Professional.” <https://www.universal-robots.com/plus/products/4each/mqtt-connector-professional/> (accessed Dec. 19, 2022).
- [38] “Rocketfarm | UR+ Software Experts | Palletizing | OPC UA.” <https://rocketfarm.no/> (accessed Dec. 22, 2022).
- [39] ROCKETFARM, “OPC UA URCap User Manual,” Sogndal, 2020. [Online]. Available: <https://rocketfarm.no/software-products/opc-ua-udma/>.
- [40] “OPC UA VDMA | URCap | Industry 4.0 with UR | Rocketfarm.” <https://rocketfarm.no/software-products/opc-ua-udma/> (accessed Dec. 22, 2022).
- [41] “UaExpert ‘UA Reference Client’ - Unified Automation.” <https://www.unified-automation.com/products/development-tools/uaexpert.html> (accessed Dec. 27, 2022).
- [42] “OPC UA Clients - Unified Automation.” <https://www.unified-automation.com/downloads/opc-ua-clients.html> (accessed Dec. 27, 2022).
- [43] “How OPC UA Clients Discover Servers (Part 2).” <https://www.automation.com/en-us/articles/2015-2/how-opc-ua-clients-discover-servers-part-2> (accessed Dec. 27, 2022).
- [44] “OPC Expert Software | Download Now | Free OPC Client.” <https://opcexpert.com/> (accessed Dec. 27, 2022).
- [45] “OPC to Excel | Free Excel Add-In | OPC Expert.” <https://opcexpert.com/opc-to-excel/> (accessed Dec. 27, 2022).
- [46] “Prosys OPC UA Monitor - Prosys OPC.” <https://www.prosysopc.com/products/opc-ua-monitor/> (accessed Dec. 27, 2022).
- [47] “Wireshark · Go Deep.” <https://www.wireshark.org/> (accessed Jan. 03, 2023).
- [48] “FAS-200 - Sistema de ensamblaje flexible.” <https://www.smctraining.com/es/webpage/indexpage/235> (accessed Dec. 19, 2022).
- [49] J. S. Pereda, “NTP 1098: Equipo eléctrico de máquinas: colores y marcados de los órganos de accionamiento Elaborado por,” *Inst. Nac. Segur. e Hig. en el Trab.*, pp. 1–4, 2017, [Online]. Available:

<http://www.insht.es/InshtWeb/Contenidos/Documentacion/NTP/NTP/Ficheros/1090a1100/n tp-1098.pdf>.

- [50] “RoHS Directive.” https://environment.ec.europa.eu/topics/waste-and-recycling/rohs-directive_en (accessed Dec. 23, 2022).
- [51] “Universal Robots e-Series User Manual UR3e Original instructions (en).”
- [52] “Waste from Electrical and Electronic Equipment (WEEE).” https://environment.ec.europa.eu/topics/waste-and-recycling/waste-electrical-and-electronic-equipment-weee_en (accessed Dec. 27, 2022).

Annex A

En el conjunt d'aquests annexos es troben recollits per una banda els resultats dels test de seguretat així com les especificacions completes del robot UR3e. D'altra banda, s'inclou el codi complet de les tres aplicacions robòtiques. A diferència del codi mostrat en apartats anteriors, que estava editat per facilitar la seva visualització, apareix en aquest annex la totalitat del codi sense limitacions de longitud.

A1. Captures de pantalla dels escenaris dels test de seguretat

- Escenari 1: Anònim + None

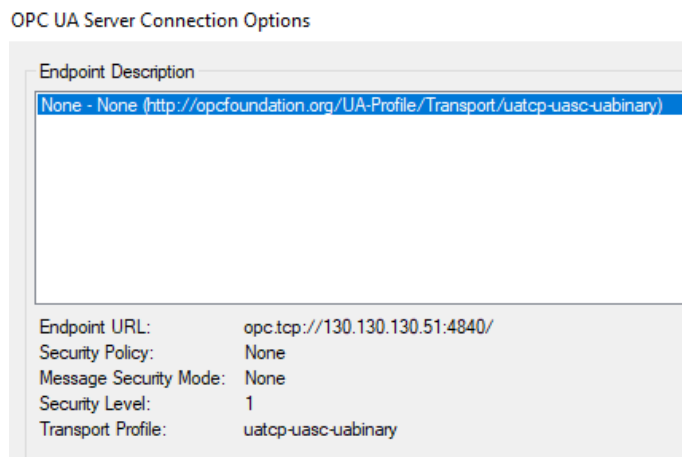


Figura A1. Detalls de seguretat del cas 1 extrets de OPC Expert (Font: Autor)

57	8.720274	130.130.130.11	130.130.130.51	OpcUa	162 UA Secure Conversation Message: ReadRequest
58	8.720500	130.130.130.51	130.130.130.11	OpcUa	120 UA Secure Conversation Message: ReadResponse
270	30.108693	130.130.130.11	130.130.130.51	OpcUa	165 UA Secure Conversation Message: WriteRequest
271	30.108954	130.130.130.51	130.130.130.11	OpcUa	118 UA Secure Conversation Message: WriteResponse

Figura A2. Capçalera de les trames OPC UA del cas 1 extretes amb Wireshark (Font: Autor)

Wireshark - Packet 270 - Ethernet 8

```

> Frame 270: 165 bytes on wire (1320 bits), 165 bytes captured (1320 bits) on interface \Device\NPF_{DA724C76-9CA6-40C2-A270-2648F34AD71A}, id 0
> Ethernet II, Src: ASSMANNE_17:ee:5d (3c:49:37:17:ee:5d), Dst: MscVertr_20:23:fa (00:30:d6:20:23:fa)
> Internet Protocol Version 4, Src: 130.130.130.11, Dst: 130.130.130.51
> Transmission Control Protocol, Src Port: 62997, Dst Port: 4840, Seq: 1355, Ack: 985, Len: 111
▼ OpcUa Binary Protocol
  Message Type: MSG
  Chunk Type: F
  Message Size: 111
  SecureChannelId: 3
  Security Token Id: 3
  Security Sequence Number: 26
  Security RequestId: 26
  ▼ OpcUa Service : Encodeable Object
    > TypeId : ExpandedNodeId
    ▼ WriteRequest
      ▼ RequestHeader: RequestHeader
        ▼ AuthenticationToken: NodeId
          ... 0100 = EncodingMask: GUID (0x4)
          Namespace Index: 1
          Identifier Guid: 5f27b736-2415-bc16-38d3-5db5b02be2ce
          Timestamp: Nov 10, 2022 13:41:49.797251400 Hora estándar romance
          RequestHandle: 11
          > Return Diagnostics: 0x00000000
          AuditEntryId: [OpcUa Null String]
          TimeoutHint: 0
          ▼ AdditionalHeader: ExtensionObject
            > TypeId: ExpandedNodeId
            > EncodingMask: 0x00
          ▼ NodesToWrite: Array of WriteValue
            ArraySize: 1
            ▼ [0]: WriteValue
              ▼ NodeId: NodeId
                ... 0011 = EncodingMask: String (0x3)
                Namespace Index: 2
                Identifier String: Constant
                AttributeId: Value (0x0000000d)
                IndexRange: [OpcUa Null String]
                ▼ Value: DataValue
                  > EncodingMask: 0x01, has value
                  ▼ Value: Variant
                    Variant Type: Double (0x0b)
                    Double: 1

```

Wireshark - Packet 76 - Ethernet 8

```

> Frame 76: 170 bytes on wire (1360 bits), 170 bytes captured (1360 bits) on interface \Device\NPF_{DA724C76-9CA6-40C2-A270-2648F34AD71A}, id 0
> Ethernet II, Src: ASSMANNE_17:ee:5d (3c:49:37:17:ee:5d), Dst: MscVertr_20:23:fa (00:30:d6:20:23:fa)
> Internet Protocol Version 4, Src: 130.130.130.11, Dst: 130.130.130.51
> Transmission Control Protocol, Src Port: 62997, Dst Port: 4840, Seq: 389, Ack: 436, Len: 116
▼ OpcUa Binary Protocol
  Message Type: MSG
  Chunk Type: F
  Message Size: 116
  SecureChannelId: 3
  Security Token Id: 3
  Security Sequence Number: 55
  Security RequestId: 55
  ▼ OpcUa Service : Encodeable Object
    > TypeId : ExpandedNodeId
    ▼ WriteRequest
      ▼ RequestHeader: RequestHeader
        ▼ AuthenticationToken: NodeId
          ... 0100 = EncodingMask: GUID (0x4)
          Namespace Index: 1
          Identifier Guid: 5f27b736-2415-bc16-38d3-5db5b02be2ce
          Timestamp: Nov 10, 2022 13:43:02.619212700 Hora estándar romance
          RequestHandle: 13
          > Return Diagnostics: 0x00000000
          AuditEntryId: [OpcUa Null String]
          TimeoutHint: 0
          ▼ AdditionalHeader: ExtensionObject
            > TypeId: ExpandedNodeId
            > EncodingMask: 0x00
          ▼ NodesToWrite: Array of WriteValue
            ArraySize: 1
            ▼ [0]: WriteValue
              ▼ NodeId: NodeId
                ... 0011 = EncodingMask: String (0x3)
                Namespace Index: 2
                Identifier String: Message
                AttributeId: Value (0x0000000d)
                IndexRange: [OpcUa Null String]
                ▼ Value: DataValue
                  > EncodingMask: 0x01, has value
                  ▼ Value: Variant
                    Variant Type: String (0x0c)
                    String: Escenari 1

```

Figura A3. Detalls del paquet enviat del cas 1 aconseguits via Wireshark (Font: Autor)

○ Escenari 2: User&Password + None

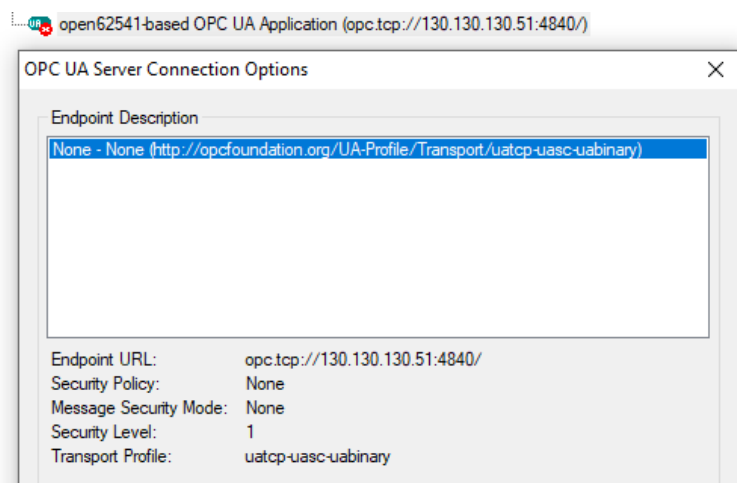


Figura A4. Detalls de seguretat del cas 2 extrets de OPC Expert (Font: Autor)

754	92.512513	130.130.130.11	130.130.130.51	OpcUa	140 UA Secure Conversation Message: PublishRequest
769	94.511866	130.130.130.51	130.130.130.11	OpcUa	204 UA Secure Conversation Message: PublishResponse
146	15.520442	130.130.130.11	130.130.130.51	OpcUa	165 UA Secure Conversation Message: WriteRequest
147	15.520788	130.130.130.51	130.130.130.11	OpcUa	118 UA Secure Conversation Message: WriteResponse

Figura A5. Capçalera de les trames OPC UA del cas 2 extretes amb Wireshark (Font: Autor)

Wireshark - Packet 769 - Ethernet 8

```

> Frame 769: 204 bytes on wire (1632 bits), 204 bytes captured (1632 bits) on interface \Device\NPF_{DA724C76-9CA6-40C2-A270-264BF34AD71A}, id 0
> Ethernet II, Src: MscVertr_20:23:fa (00:30:d6:20:23:fa), Dst: ASSMANNE_17:ee:5d (3c:49:37:17:ee:5d)
> Internet Protocol Version 4, Src: 130.130.130.51, Dst: 130.130.130.11
> Transmission Control Protocol, Src Port: 4840, Dst Port: 51541, Seq: 2955, Ack: 3472, Len: 150
▼ OpcUa Binary Protocol
  Message Type: MSG
  Chunk Type: F
  Message Size: 150
  SecureChannelId: 1
  Security Token Id: 1
  Security Sequence Number: 46
  Security RequestId: 46
  OpcUa Service : Encodeable Object
  ▼ TypeId : ExpandedNodeId
    NodeId EncodingMask: Four byte encoded Numeric (0x01)
    NodeId Namespace Index: 0
    NodeId Identifier Numeric: PublishResponse (829)
  ▼ PublishResponse
    ResponseHeader: ResponseHeader
    Timestamp: Nov 10, 2022 11:11:07.725358000 Hora estándar romance
    RequestHandle: 11
    ServiceResult: 0x00000000 [Good]
    > ServiceDiagnostics: DiagnosticInfo
    > StringTable: Array of String
    ▼ AdditionalHeader: ExtensionObject
      > TypeId: ExpandedNodeId
      > EncodingMask: 0x00
    SubscriptionId: 2
    > AvailableSequenceNumbers: Array of UInt32
    MoreNotifications: False
  ▼ NotificationMessage: NotificationMessage
    SequenceNumber: 10
    PublishTime: Nov 10, 2022 11:11:07.725357000 Hora estándar romance
    ▼ NotificationData: Array of ExtensionObject
      ArraySize: 1
      ▼ [0]: ExtensionObject
        > TypeId: ExpandedNodeId
        > EncodingMask: 0x01, has binary body
        ▼ DataChangeNotification: DataChangeNotification
          MonitoredItems: Array of MonitoredItemNotification
          ArraySize: 1
          ▼ [0]: MonitoredItemNotification
            ClientHandle: 17
            Value: DataValue
            > EncodingMask: 0x0d, has value, has source timestamp, has server timestamp
            ▼ Value: Variant
              Variant Type: String (0x0c)
              String: Inici Programa
              SourceTimestamp: Nov 10, 2022 11:11:06.886925000 Hora estándar romance
              ServerTimestamp: Nov 10, 2022 11:11:06.886925000 Hora estándar romance

```


 Details de seguretat

Wireshark - Packet 146 - Ethernet 8

```

> Frame 146: 165 bytes on wire (1320 bits), 165 bytes captured (1320 bits) on interface \Device\NPF_{DA724C76-9CA6-40C2-A270-264BF34AD71A}, id 0
> Ethernet II, Src: ASSMANNE_17:ee:5d (3c:49:37:17:ee:5d), Dst: MscVertr_20:23:fa (00:30:d6:20:23:fa)
> Internet Protocol Version 4, Src: 130.130.130.11, Dst: 130.130.130.51
> Transmission Control Protocol, Src Port: 51541, Dst Port: 4840, Seq: 522, Ack: 403, Len: 111
▼ OpcUa Binary Protocol
  Message Type: MSG
  Chunk Type: F
  Message Size: 111
  SecureChannelId: 1
  Security Token Id: 9
  Security Sequence Number: 329
  Security RequestId: 329
  OpcUa Service : Encodeable Object
  > TypeId : ExpandedNodeId
  ▼ WriteRequest
    RequestHeader: RequestHeader
    > AuthenticationToken: NodeId
    Timestamp: Nov 10, 2022 12:43:23.632997900 Hora estándar romance
    RequestHandle: 15
    > Return Diagnostics: 0x00000000
    AuditEntryId: [OpcUa Null String]
    TimeoutHint: 0
    ▼ AdditionalHeader: ExtensionObject
      > TypeId: ExpandedNodeId
      > EncodingMask: 0x00
    ▼ NodesToWrite: Array of WriteValue
      ArraySize: 1
      ▼ [0]: WriteValue
        NodeId: NodeId
        ... 0011 = EncodingMask: String (0x3)
        Namespace Index: 2
        Identifier String: Constant
        AttributeId: Value (0x0000000d)
        IndexRange: [OpcUa Null String]
        Value: DataValue
        > EncodingMask: 0x01, has value
        Value: Variant
        Variant Type: Double (0x0b)
        Double: 1

```

Figura A6. Detalls del paquet enviat del cas 2 aconseguits via Wireshark (Font: Autor)

○ Escenari 3: User&Password + Sign (Basic128Rsa15)

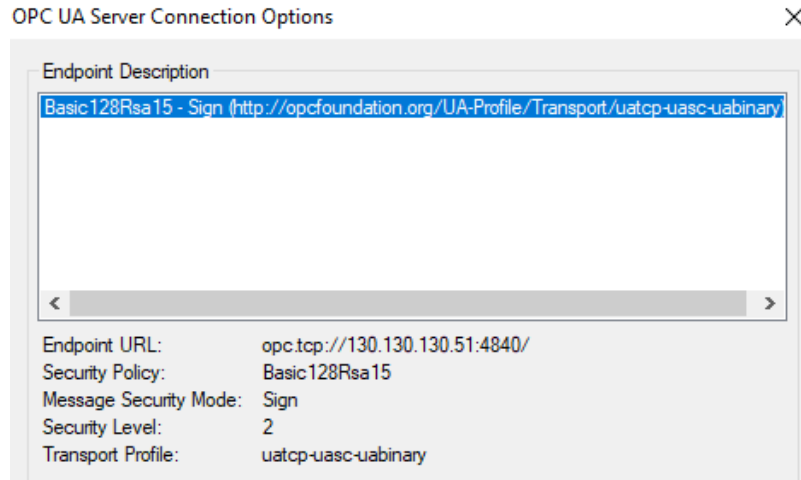
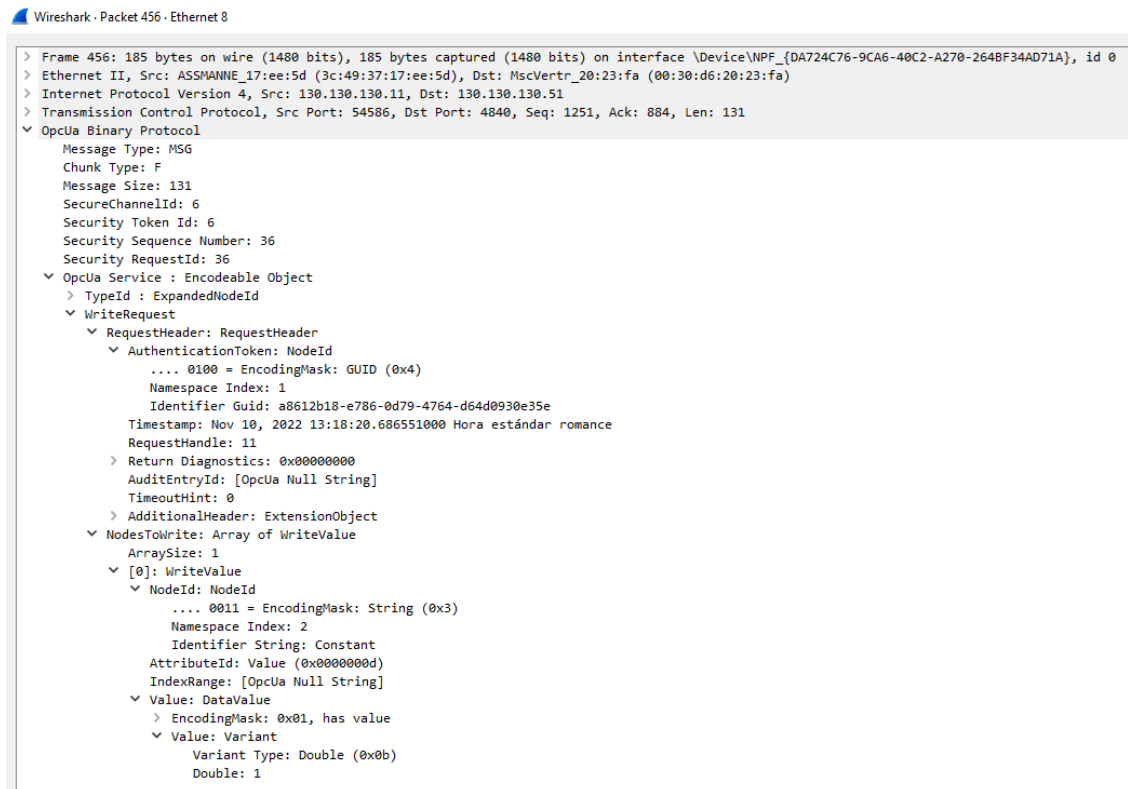


Figura A7. Detalls de seguretat del cas 3 extrets de OPC Expert (Font: Autor)

194	17.837630	130.130.130.11	130.130.130.51	OpcUa	182 UA Secure Conversation Message: ReadRequest
195	17.837907	130.130.130.51	130.130.130.11	OpcUa	140 UA Secure Conversation Message: ReadResponse
456	43.707071	130.130.130.11	130.130.130.51	OpcUa	185 UA Secure Conversation Message: WriteRequest
457	43.707354	130.130.130.51	130.130.130.11	OpcUa	138 UA Secure Conversation Message: WriteResponse

Figura A8. Capçalera de les trames OPC UA del cas 3 extretes amb Wireshark (Font: Autor)



Wireshark · Packet 245 · Ethernet 8

```

> Frame 245: 227 bytes on wire (1816 bits), 227 bytes captured (1816 bits) on interface \Device\NPF_{DA724C76-9CA6-40C2-A270-264BF34AD71A}, id 0
> Ethernet II, Src: MscVertr_20:23:fa (00:30:d6:20:23:fa), Dst: ASSMANNE_17:ee:5d (3c:49:37:17:ee:5d)
> Internet Protocol Version 4, Src: 130.130.130.51, Dst: 130.130.130.11
> Transmission Control Protocol, Src Port: 4840, Dst Port: 54586, Seq: 944, Ack: 959, Len: 173
▼ OpcUa Binary Protocol
  Message Type: MSG
  Chunk Type: F
  Message Size: 173
  SecureChannelId: 6
  Security Token Id: 7
  Security Sequence Number: 81
  Security RequestId: 79
▼ OpcUa Service : Encodeable Object
  > TypeId : ExpandedNodeId
  ▼ PublishResponse
    > ResponseHeader: ResponseHeader
    SubscriptionId: 2
    > AvailableSequenceNumbers: Array of UInt32
    MoreNotifications: False
    ▼ NotificationMessage: NotificationMessage
      SequenceNumber: 8
      PublishTime: Nov 10, 2022 12:09:19.900200000 Hora estándar romance
      ▼ NotificationData: Array of ExtensionObject
        ArraySize: 1
        ▼ [0]: ExtensionObject
          > TypeId: ExpandedNodeId
          > EncodingMask: 0x01, has binary body
          ▼ DataChangeNotification: DataChangeNotification
            ▼ MonitoredItems: Array of MonitoredItemNotification
              ArraySize: 1
              ▼ [0]: MonitoredItemNotification
                ClientHandle: 5
                ▼ Value: DataValue
                  > EncodingMask: 0x0d, has value, has source timestamp, has server timestamp
                  ▼ Value: Variant
                    Variant Type: String (0x0c)
                    String: Final de programa
                    SourceTimestamp: Nov 10, 2022 12:09:19.580831000 Hora estándar romance
                    ServerTimestamp: Nov 10, 2022 12:09:19.580831000 Hora estándar romance
                > DiagnosticInfos: Array of DiagnosticInfo
          > Results: Array of StatusCode
          > DiagnosticInfos: Array of DiagnosticInfo
  
```

Figura A9. Detalls del paquet enviat del cas 3 aconseguits via Wireshark (Font: Autor)

○ Escenari 4: User&Password + Sign (Basic256)

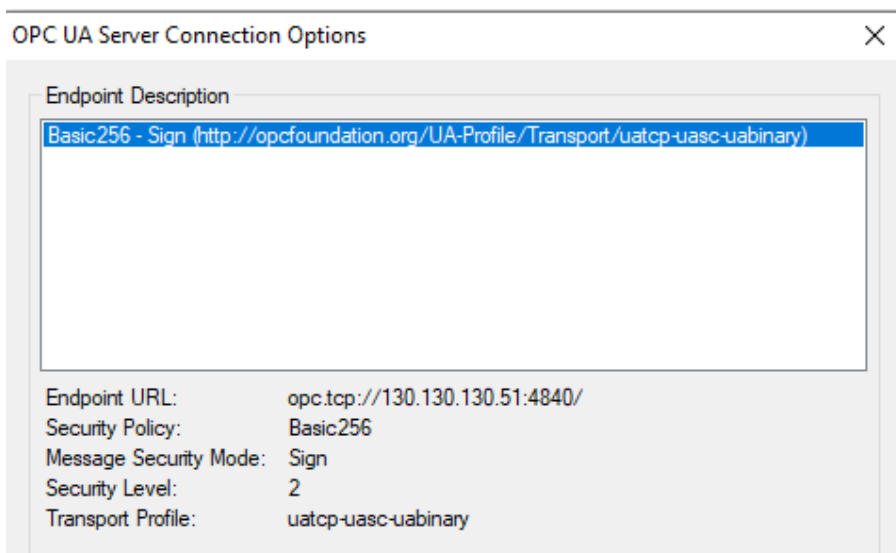


Figura A10. Detalls de seguretat del cas 4 extrets de OPC Expert (Font: Autor)

101	6.332395	130.130.130.11	130.130.130.51	OpcUa	182	UA Secure Conversation Message: ReadRequest
102	6.332661	130.130.130.51	130.130.130.11	OpcUa	140	UA Secure Conversation Message: ReadResponse



421	41.870606	130.130.130.11	130.130.130.51	OpcUa	190 UA Secure Conversation Message: WriteRequest
422	41.871015	130.130.130.51	130.130.130.11	OpcUa	138 UA Secure Conversation Message: WriteResponse

Figura A11. Capçalera de les trames OPC UA del cas 4 extretes amb Wireshark (Font: Autor)

```

Wireshark · Packet 421 · Ethernet 8
> Frame 421: 190 bytes on wire (1520 bits), 190 bytes captured (1520 bits) on interface \Device\NPF_{DA724C76-9CA6-40C2-A270-2648F34AD71A}, id 0
> Ethernet II, Src: ASSMANNE_17:ee:5d (3c:49:37:17:ee:5d), Dst: MscVertr_20:23:fa (00:30:d6:20:23:fa)
> Internet Protocol Version 4, Src: 130.130.130.11, Dst: 130.130.130.51
> Transmission Control Protocol, Src Port: 51329, Dst Port: 4840, Seq: 1920, Ack: 1875, Len: 136
▼ OpcUa Binary Protocol
  Message Type: MSG
  Chunk Type: F
  Message Size: 136
  SecureChannelId: 3
  Security Token Id: 3
  Security Sequence Number: 47
  Security RequestId: 47
  ▼ OpcUa Service : Encodeable Object
    > TypeId : ExpandedNodeId
    ▼ WriteRequest
      ▼ RequestHeader: RequestHeader
        ▼ AuthenticationToken: NodeId
          ... 0100 = EncodingMask: GUID (0x4)
          Namespace Index: 1
          Identifier Guid: 62e54702-7002-ef84-c87c-d78d85f86f86
          Timestamp: Nov 10, 2022 13:27:46.966960600 Hora estándar romance
          RequestHandle: 13
        > Return Diagnostics: 0x00000000
        AuditEntryId: [OpcUa Null String]
        TimeoutHint: 0
      ▼ AdditionalHeader: ExtensionObject
        > TypeId: ExpandedNodeId
        > EncodingMask: 0x00
      ▼ NodesToWrite: Array of WriteValue
        ArraySize: 1
        ▼ [0]: WriteValue
          ▼ NodeId: NodeId
            ... 0011 = EncodingMask: String (0x3)
            Namespace Index: 2
            Identifier String: Message
            AttributeId: Value (0x0000000d)
            IndexRange: [OpcUa Null String]
          ▼ Value: DataValue
            > EncodingMask: 0x01, has value
            ▼ Value: Variant
              Variant Type: String (0x0c)
              String: Escenari 4
    
```

Figura A12. Detalls del paquet enviat del cas 4 aconseguits via Wireshark (Font: Autor)

```

Wireshark · Packet 366 · Ethernet 8
> Frame 366: 185 bytes on wire (1480 bits), 185 bytes captured (1480 bits) on interface \Device\NPF_{DA724C76-9CA6-40C2-A270-264BF34AD71A}, id 0
> Ethernet II, Src: ASSMANNE_17:ee:5d (3c:49:37:17:ee:5d), Dst: MscVertr_20:23:fa (00:30:d6:20:23:fa)
> Internet Protocol Version 4, Src: 130.130.130.11, Dst: 130.130.130.51
> Transmission Control Protocol, Src Port: 51329, Dst Port: 4840, Seq: 1427, Ack: 1459, Len: 131
▼ OpcUa Binary Protocol
  Message Type: MSG
  Chunk Type: F
  Message Size: 131
  SecureChannelId: 3
  Security Token Id: 3
  Security Sequence Number: 43
  Security RequestId: 43
▼ OpcUa Service : Encodeable Object
  > TypeId : ExpandedNodeId
  ▼ WriteRequest
    ▼ RequestHeader: RequestHeader
      ▼ AuthenticationToken: NodeId
        ... 0100 = EncodingMask: GUID (0x4)
        Namespace Index: 1
        Identifier Guid: 62e54702-7002-ef84-c87c-d78d85f86f86
        Timestamp: Nov 10, 2022 13:27:40.199417800 Hora estándar romance
        RequestHandle: 12
      > ReturnDiagnostics: 0x00000000
      AuditEntryId: [OpcUa Null String]
      TimeoutHint: 0
    ▼ AdditionalHeader: ExtensionObject
      > TypeId: ExpandedNodeId
      > EncodingMask: 0x00
    ▼ NodesToWrite: Array of WriteValue
      ArraySize: 1
      ▼ [0]: WriteValue
        ▼ NodeId: NodeId
          ... 0011 = EncodingMask: String (0x3)
          Namespace Index: 2
          Identifier String: Constant
          AttributeId: Value (0x0000000d)
          IndexRange: [OpcUa Null String]
        ▼ Value: DataValue
          > EncodingMask: 0x01, has value
          ▼ Value: Variant
            Variant Type: Double (0x0b)
            Double: 0

```

Figura A13. Detalls del paquet enviat del cas 4 aconseguits via Wireshark (Font: Autor)

○ **Escenari 5: User&Password + Sign (Basic256Sha256)**

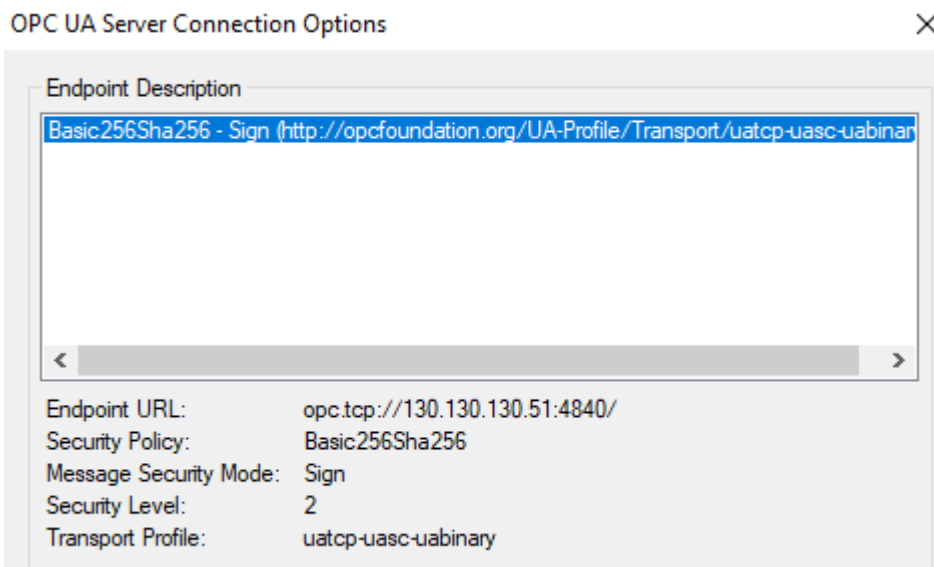


Figura A14. Detalls de seguretat del cas 5 extrets de OPC Expert (Font: Autor)

159	23.626271	130.130.130.11	130.130.130.51	OpcUa	194 UA Secure Conversation Message: ReadRequest
160	23.626539	130.130.130.51	130.130.130.11	OpcUa	152 UA Secure Conversation Message: ReadResponse
593	75.646542	130.130.130.11	130.130.130.51	OpcUa	197 UA Secure Conversation Message: WriteRequest
594	75.646828	130.130.130.51	130.130.130.11	OpcUa	150 UA Secure Conversation Message: WriteResponse

Figura A15. Capçalera de les trames OPC UA del cas 5 extretes amb Wireshark (Font: Autor)

```

Wireshark · Packet 593 · Ethernet 8
> Frame 593: 197 bytes on wire (1576 bits), 197 bytes captured (1576 bits) on interface \Device\NPF_{DA724C76-9CA6-40C2-A270-2648F34AD71A}, id 0
> Ethernet II, Src: ASSMANNE_17:ee:5d (3c:49:37:17:ee:5d), Dst: MscVertr_20:23:fa (00:30:d6:20:23:fa)
> Internet Protocol Version 4, Src: 130.130.130.11, Dst: 130.130.130.51
> Transmission Control Protocol, Src Port: 50057, Dst Port: 4840, Seq: 4090, Ack: 3697, Len: 143
▼ OpcUa Binary Protocol
  Message Type: MSG
  Chunk Type: F
  Message Size: 143
  SecureChannelId: 3
  Security Token Id: 5
  Security Sequence Number: 90
  Security RequestId: 90
  ▼ OpcUa Service : Encodeable Object
    > TypeId : ExpandedNodeId
    ▼ WriteRequest
      ▼ RequestHeader: RequestHeader
        ▼ AuthenticationToken: NodeId
          ... 0100 = EncodingMask: GUID (0x4)
          Namespace Index: 1
          Identifier Guid: f01271e7-5544-1f26-7497-2912f07f97c9
          Timestamp: Nov 10, 2022 13:36:56.325354800 Hora estándar romance
          RequestHandle: 11
        > Return Diagnostics: 0x00000000
        AuditEntryId: [OpcUa Null String]
        TimeoutHint: 0
      ▼ AdditionalHeader: ExtensionObject
        > TypeId: ExpandedNodeId
        > EncodingMask: 0x00
      ▼ NodesToWrite: Array of WriteValue
        ArraySize: 1
        ▼ [0]: WriteValue
          ▼ NodeId: NodeId
            ... 0011 = EncodingMask: String (0x3)
            Namespace Index: 2
            Identifier String: Constant
            AttributeId: Value (0x0000000d)
            IndexRange: [OpcUa Null String]
          ▼ Value: DataValue
            > EncodingMask: 0x01, has value
            ▼ Value: Variant
              Variant Type: Double (0x0b)
              Double: 1
    
```

Figura A16. Detalls del paquet enviat del cas 5 aconseguits via Wireshark (Font: Autor)

```

Wireshark - Packet 59 - Ethernet 8
> Frame 59: 202 bytes on wire (1616 bits), 202 bytes captured (1616 bits) on interface \Device\NPF_{DA724C76-9CA6-40C2-A270-264BF34AD71A}, id 0
> Ethernet II, Src: ASSMANNE_17:ee:5d (3c:49:37:17:ee:5d), Dst: MscVertr_20:23:fa (00:30:d6:20:23:fa)
> Internet Protocol Version 4, Src: 130.130.130.11, Dst: 130.130.130.51
> Transmission Control Protocol, Src Port: 50057, Dst Port: 4840, Seq: 281, Ack: 197, Len: 148
▼ OpcUa Binary Protocol
  Message Type: MSG
  Chunk Type: F
  Message Size: 148
  SecureChannelId: 3
  Security Token Id: 5
  Security Sequence Number: 131
  Security RequestId: 131
  ▼ OpcUa Service : Encodeable Object
    > TypeId : ExpandedNodeId
    ▼ WriteRequest
      ▼ RequestHeader: RequestHeader
        AuthenticationToken: NodeId
          ... 0100 = EncodingMask: GUID (0x4)
          Namespace Index: 1
          Identifier Guid: f01271e7-5544-1f26-7497-2912f07f97c9
          Timestamp: Nov 10, 2022 13:38:50.078597300 Hora estándar romance
          RequestHandle: 14
          > Return Diagnostics: 0x00000000
          AuditEntryId: [OpcUa Null String]
          TimeoutHint: 0
          ▼ AdditionalHeader: ExtensionObject
            > TypeId: ExpandedNodeId
            > EncodingMask: 0x00
          ▼ NodesToWrite: Array of WriteValue
            ArraySize: 1
            ▼ [0]: WriteValue
              NodeId: NodeId
                ... 0011 = EncodingMask: String (0x3)
                Namespace Index: 2
                Identifier String: Message
                AttributeId: Value (0x0000000d)
                IndexRange: [OpcUa Null String]
                Value: DataValue
                  > EncodingMask: 0x01, has value
                  Value: Variant
                    Variant Type: String (0x0c)
                    String: Escenari 5

```

Figura A17. Detalls del paquet enviat del cas 5 aconseguits via Wireshark (Font: Autor)

○ **Escenari 6: User&Password + Sign&Encrypt (Basic256)**

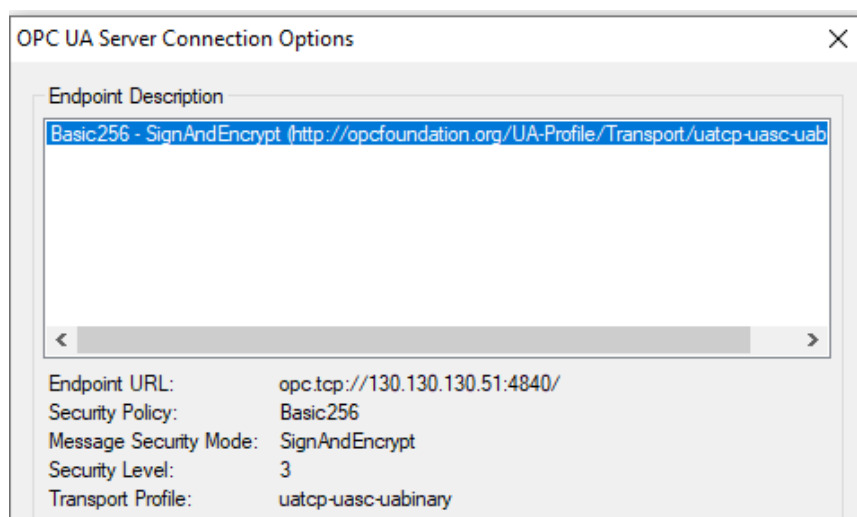


Figura A18. Detalls de seguretat del cas 6 extrets de OPC Expert (Font: Autor)

111	8.625393	130.130.130.11	130.130.130.51	OpcUa	198 UA Secure Conversation Message: ServiceId 0
112	8.625720	130.130.130.51	130.130.130.11	OpcUa	150 UA Secure Conversation Message: ServiceId 0

Figura A19. Capçalera de les trames OPC UA del cas 6 extretes amb Wireshark (Font: Autor)

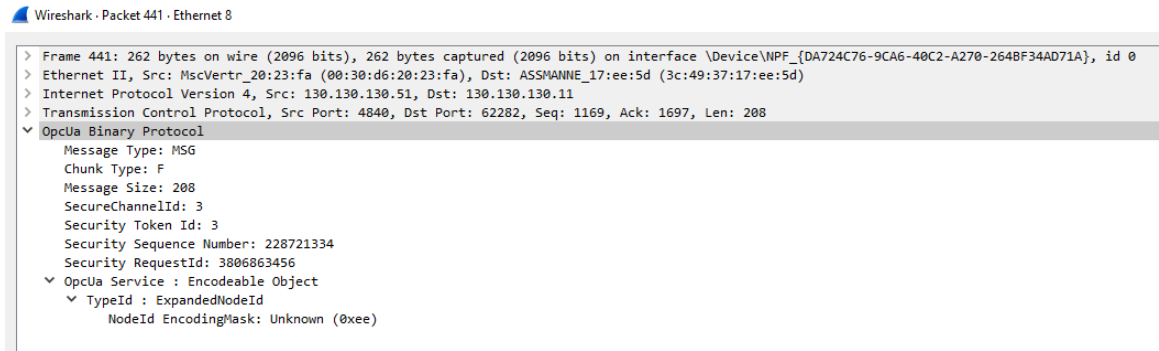


Figura A20. Detalls del paquet enviat del cas 6 aconseguits via Wireshark (Font: Autor)

○ **Escenari 7: User&Password + Sign&Encrypt (Basic256Sha256)**

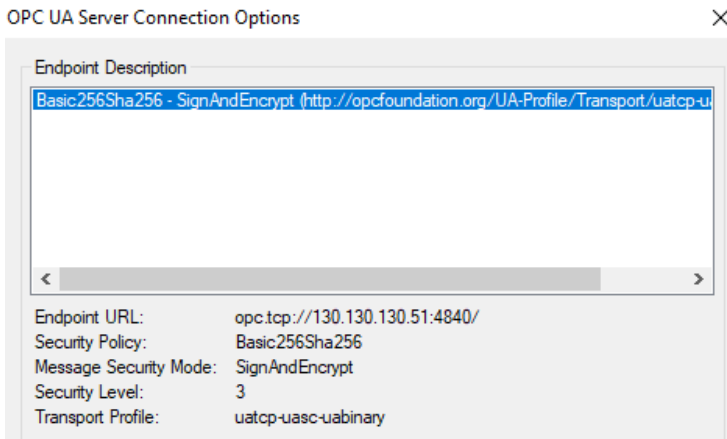


Figura A21. Detalls de seguretat del cas 7 extrets de OPC Expert (Font: Autor)

153	15.395543	130.130.130.51	130.130.130.11	OpCua	166 UA Secure Conversation Message: ServiceId 0
189	20.410676	130.130.130.11	130.130.130.51	OpCua	198 UA Secure Conversation Message: ServiceId 0

Figura A22. Capçalera de les trames OPC UA del cas 7 extretes amb Wireshark (Font: Autor)

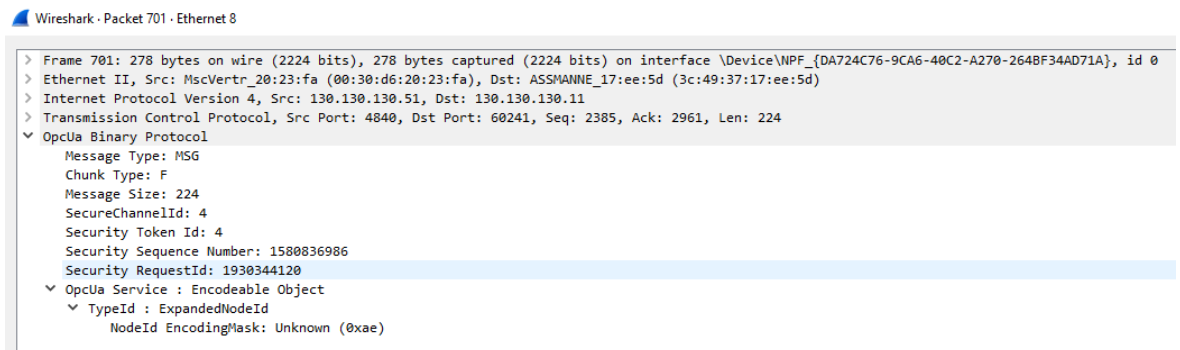


Figura A23. Detalls del paquet enviat del cas 7 aconseguits via Wireshark (Font: Autor)


A2. Especificacions robot UR3e

UR3e Technical Specifications

With a 3 kg payload capacity and 500 mm reach, the compact form factor of the UR3e makes it a fit for tight workspaces.

Today, more than 50,000 UR collaborative industrial robots have been delivered to customers across industries and around the world. UR3e is one of four e-Series cobots, each with a different payload and reach combination. e-Series brings incredible flexibility and unparalleled ease of use to your application.

UR3e		
Specification		
Payload	3 kg (6.6 lbs)	
Reach	500 mm (19.7 in)	
Degrees of freedom	6 rotating joints	
Programming	12 inch touchscreen with polyscope graphical user interface	
Performance		
Power, Consumption, Maximum Average	300 W	
Power, Consumption, Typical with moderate settings (approximate)	100 W	
Safety	17 configurable safety functions	
Certifications	EN ISO 13849-1, PLd Category 3, and EN ISO 18218-1	
Force Sensing, Tool Flange	Force, x-y-z	Torque, x-y-z
Range	30.0 N	10.0 Nm
Precision	2.0 N	0.1 Nm
Accuracy	3.5 N	0.1 Nm
Movement		
Pose Repeatability per ISO 9283	± 0.03 mm	
Axis movement	Working range	Maximum speed
Base	± 360°	± 180°/s
Shoulder	± 360°	± 180°/s
Elbow	± 360°	± 180°/s
Wrist 1	± 360°	± 360°/s
Wrist 2	± 360°	± 360°/s
Wrist 3	Infinite	± 360°/s
Typical TCP speed	1 m/s (39.4 in/s)	
Features		
IP classification	IP54	
ISO 14644-1 Class Cleanroom	5	
Noise	Less than 60 dB(A)	
Robot mounting	Any orientation	
I/O ports		
Digital in	2	
Digital out	2	
Analog in	2	
Tool I/O Power Supply Voltage	12/24 V	
Tool I/O Power Supply	600 mA	
Physical		
Footprint	Ø 128 mm	
Materials	Aluminium, Plastic, Steel	
Tool (end-effector) connector type	M8 M8 8-pin	
Cable length robot arm	6 m (236 in) cable included. 12 m (472 in) and high-flex options available.	
Weight including cable	11.2 kg (17.7 lbs)	
Operating temperature range	0-50°C	
Humidity	90%RH (non-condensing)	



Control Box	
Features	
IP classification	IP44
ISO 14644-1 Class Cleanroom	6
Operating temperature range	0-50°C
Humidity	90%RH (non-condensing)
I/O ports	
Digital in	16
Digital out	16
Analog in	2
Analog out	2
Quadrature Digital Inputs	4
I/O Power Supply	
	24V 2A
Communication	
500 Hz Control frequency	
Modbus TCP	
PROFINET	
Ethernet/IP	
USB 2.0, USB 3.0	
Power source	
	100-240VAC, 47-440Hz
Physical	
Control box size (W x H x D)	460 mm x 449 mm x 254 mm (18.2 in x 17.6 in x 10 in)
Weight	12 kg (26.5 lbs)
Materials	Powder Coated Steel

The control box is also available in an OEM version.

Teach Pendant	
Features	
IP classification	IP54
Humidity	90%RH (non-condensing)
Display resolution	1280 x 800 pixels
Physical	
Materials	Plastic, PP
Weight	1.6 kg (3.5 lbs) including 1m of TP cable
Cable length	4.5 m (177.17 in)

The teach pendant is also available in a 3PE option.

Contact

Universal Robots A/S
Energivej 25
5260 Odense
Denmark
+45 89 93 89 89
sales@universal-robots.com
universal-robots.com

Figura A24: Especificacions tècniques completes del robot UR3e.

A3. Programes complets de les aplicacions robòtiques

- **BUCLE_OPC** (Bucle simple OPC UA)

```
1 Program
2   Robot Program
3   Fijar la carga útil: Payload
4   String:="Inici Programa"
5   State:= False
6   count:=0
7   Write OPC UA server: Step
8   Write OPC UA server: Message
9   Write OPC UA server: Robot_working
10  Read OPC UA server: Constant
11  MoveJ
12    State:= True
13    count:=count+1
14    Write OPC UA server: Step
15    Write OPC UA server: Robot_working
16    Waypoint_1
17  Set Gripper= True
18  count:=count+1
19  Write OPC UA server: Step
20  Wait: 1.0
21  MoveL
22    Waypoint_5
23    count:=count+1
24    Write OPC UA server: Step
```

Figura A25. Captura del programa BUCLE_OPC (Font: Autor)

```
25     MoveJ
26         Waypoint_2
27         count:=count+1
28     Write OPC UA server: Step
29     Set Gripper=Apagar
30     Wait: 1.0
31     MoveL
32         Waypoint_4
33         count:=count+1
34     Write OPC UA server: Step
35     MoveJ
36         Waypoint_3
37         count:=count+1
38     Write OPC UA server: Step
39     Set Gripper=Encender
40     Wait: 1.0
41     Set Gripper=Apagar
42     count:=count+1
43     Write OPC UA server: Step
44     Read OPC UA server: Constant
45     State:= False
46     Loop Set_time#1
47         Read OPC UA server: Constant
48         String:="Robot in Standby"
49         Write OPC UA server: Message
50         State:= False
51         Write OPC UA server: Robot_working
52     If Set_time#1
53         String:="Final de programa"
54         Write OPC UA server: Message
55         count:=count+1
56         Write OPC UA server: Step
57         State:= False
58         Write OPC UA server: Robot_working
```

Figura A26. Captura del programa BUCLE_OPC (Font: Autor)

- OPC_Feedback (Interacció operari)

```

1 Program
2 BeforeStart
3   Fijar la carga útil: Payload
4   Home
5   Read OPC UA server: input_opc
6   String:="Setup Inicial: esperant input"
7   Write OPC UA server: Message
8   String2:="Editar input_opc amb el mode de treball desitjat [1,2]"
9   Write OPC UA server: ask_opc
10  Loop client_output=0
11    Read OPC UA server: input_opc
12 Robot Program
13  If client_output=1
14    Popup: Admin ha seleccionat mode de treball 1. Allunyeu-vos del robot!
15    String:="Mode de treball 1 (no feedback)"
16    Write OPC UA server: Message
17    Loop client_output=1
18      Read OPC UA server: input_opc
19      Call MT1
20  ElseIf client_output=2
21    String:="Mode de treball 2 (amb feedback)"
22    Write OPC UA server: Message
23    String2:="Esperant resposta Operari UR3e"
24    Write OPC UA server: ask_opc
25    input_local='Mode de treball 2 - Seleccionar programa a realitzar [1,2,3]:'
26    If input_local=1
27      String2:="Operari ha seleccionat 1"
28      Write OPC UA server: ask_opc
29      MoveJ
30      Waypoint_2
31      Popup: Tasca finalitzada
32    ElseIf input_local=2
33      String2:="Operari ha seleccionat 2"
34      Write OPC UA server: ask_opc
35      MoveJ
36      Waypoint_3
37      Popup: Tasca finalitzada
38    ElseIf input_local=3
39      String2:="Operari ha seleccionat 3"
40      Write OPC UA server: ask_opc
41      MoveJ
42      Waypoint_4
43      Popup: Tasca finalitzada
44      Read OPC UA server: input_opc
45    ElseIf client_output=0
46      String:="Final programa - Home"
47      Write OPC UA server: Message
48      Popup: Admin ha seleccionat Final de programa. Robot a Home position
49      String2:="Editar input_opc amb el mode de treball desitjat [1,2]"
50      Write OPC UA server: ask_opc
51      Home
52      Loop client_output=0
53        Read OPC UA server: input_opc
54    Else
55      String:="Mode de treball no disponible. Introdueixi [0,1,2]"
56      Write OPC UA server: Message
57      Loop client_output>2
58        Read OPC UA server: input_opc
59  MT1
60    Read OPC UA server: input_opc
61    MoveJ
62    Waypoint_1
63    Waypoint_5
64    Waypoint_6

```

Figura A27. Captura del programa OPC_Feedback (Font: Autor)

○ FAS200_OPC (Assemblatge)

```
1 Program
2   Init Variables
3   Robot Program
4   pas:=0
5   temps_cicle: Restablecer
6   Write OPC UA server: cycle_time
7   temps_on: Iniciar
8   Set DO[2]=Apagar
9   Set DO[3]=Apagar
10  Set Gripper=Apagar
11  Read OPC UA server: Work_mode
12  string="Setup inicial"
13  MoveJ
14  pas:=1
15  Pos_Ini
16  string_admin:="Select the desired working mode [1,2]"
17  Loop control±0
18  Set DO[0]=Apagar
19  Set DO[1]=Encender
20  Set DO[2]=Encender
21  Set DO[3]=Encender
22  string="Mode 0 - Robot in standby"
23  pas:=2
24  Read OPC UA server: Work_mode
25  Loop control±1
26  string="Working_mode 1 - Assembly of 2 units"
27  string_admin:="Write 0 in Work_mode to finish task"
28  Call SubProgram_2A
29  Read OPC UA server: Work_mode
30  Loop control±2
31  string="Working_mode 2 - Assembly of 4 units"
32  string_admin:="Write 0 in Work_mode to finish task"
33  Call SubProgram_2A
34  temps_cicle: Iniciar
35  Call SubProgram_2B
36  Read OPC UA server: Work_mode
37  SubProgram_2A
38  Read OPC UA server: Work_mode
39  working:= True
40  string="Waiting for operator to refill the required pieces..."
41  pas:=3
42  Set DO[0]=Apagar
43  Set DO[1]=Apagar
44  Set DO[2]=Apagar
45  Set DO[3]=Encender
46  If control±1
47  Popup: Robot will engage movement in Mode 1. Confirm that it has the required modules available:
48  Else
49  Popup: Robot will engage movement in Mode 2. Confirm that it has the required modules available:
50  Set DO[1]=Apagar
51  Read OPC UA server: Work_mode
52  If control±1
53  string="Working_mode 1 - Assembly of 2 units"
54  ElseIf control±2
55  string="Working_mode 2 - Assembly of 4 units"
56  Else
57  Read OPC UA server: Work_mode
58  temps_cicle: Restablecer
59  temps_cicle: Iniciar
60  Set DO[0]=Apagar
61  Set DO[1]=Apagar
62  Set DO[3]=Apagar
63  Set DO[2]=Encender
```

Figura A28. Captura del programa FAS200_OPC (Font: Autor)

```
64     MoveJ
65     pas:=4
66     Pos_Ini
67     Set Gripper=Encender
68     Wait: 1.0
69     MoveJ
70     PrePick_1a
71     MoveL
72     Pick_1
73     Set Gripper=Apagar
74     Wait: 1.5
75     MoveL
76     PrePick_1a
77     MoveJ
78     PrePlace_1
79     MoveL
80     Place_1
81     Set Gripper=Encender
82     Wait: 1.5
83     MoveL
84     PrePlace_1
85     MoveJ
86     PrePick_1b
87     MoveL
88     Waypoint_2
89     Set Gripper=Apagar
90     Wait: 1.5
91     MoveL
92     PrePick_1b
93     MoveJ
94     PrePlace_2
95     MoveL
96     Place_2
97     Set Gripper=Encender
98     Wait: 1.5
99     MoveJ
100    PrePlace_2
101    MoveJ
102    Set Gripper=Apagar
103    PrePick_R2
104    MoveL
105    Pick_R2
106    Set Gripper=Encender
107    Wait: 1.0
108    MoveL
109    PrePick_R2
110    MoveJ
111    PrePlace_R2
112    MoveL
113    Place_R2
114    Set Gripper=Apagar
115    Wait: 1.0
116    MoveL
117    PrePlace_R2
118    MoveJ
119    pas:=4
120    PrePick_R1
121    MoveL
122    Pick_R1
123    Set Gripper=Encender
124    Wait: 1.0
125    MoveL
126    PrePick_R1
127    MoveJ
128    PrePlace_R1
129    MoveL
130    Place_R1
131    Set Gripper=Apagar
132    Wait: 1.0
```

Figura A29. Captura del programa FAS200_OPC (Font: Autor)

```
132 Wait: 1.0
133 MoveL
134   PrePlace_R1
135 MoveJ
136   PrePick_E2
137 MoveL
138   Pick_E2
139 Set Gripper=Encender
140 Wait: 1.0
141 MoveL
142   PrePick_E2
143 MoveJ
144   PrePlace_E2
145 MoveL
146   Place_E2
147 Set Gripper=Apagar
148 Wait: 1.0
149 MoveL
150   PrePlace_E2
151 MoveJ
152   PrePick_E1
153 MoveL
154   Pick_E1
155 Set Gripper=Encender
156 Wait: 1.0
157 MoveL
158   PrePick_E1
159 MoveJ
160   PrePlace_E1
161 MoveL
162   Place_E1
163 Set Gripper=Apagar
164 Wait: 1.0
165 MoveL
166   PrePlace_E1
167 MoveJ
168   pas:=4
169   PrePick_T1
170 MoveL
171   Pick_T1
172 Set Gripper=Encender
173 Wait: 1.0
174 MoveL
175   PrePick_T1
176 MoveJ
177   PrePlace_T1
178 MoveL
179   Place_T1
180 Set Gripper=Apagar
181 Wait: 1.0
182 MoveL
183   PrePlace_T1
184 MoveJ
185   PrePick_T2
186 MoveL
187   Pick_T2
188 Set Gripper=Encender
189 Wait: 1.0
190 MoveL
191   PrePick_T2
192 MoveJ
193   PrePlace_T2
194 MoveL
195   Place_T2
196 Set Gripper=Apagar
197 Wait: 1.0
198 MoveL
199   PrePlace_T2
200 MoveJ
201   Pos_Ini
202   units:=units+2
203 Set DO[2]=Apagar
204 working:= False
205 temps_cicle:= Detener
206 working:= False
```

Figura A30. Captura del programa FAS200_OPC (Font: Autor)

```

207 SubProgram_2B
208   working= True
209   Set DO[1]=Apagar
210   Set DO[0]=Apagar
211   Set DO[2]=Encender
212   string="Working_mode 2 - Assembly of 4 units"
213   working= True
214   Set DO[2]=Encender
215   pas:=5
216   MoveJ
217     Set Gripper=Encender
218     PrePick_2
219     MoveL
220       Pick_3
221     Set Gripper=Apagar
222     Wait: 1.5
223     MoveL
224       PrePick_2
225     MoveJ
226       PrePlace_3
227     MoveL
228       Place_3
229     Set Gripper=Encender
230     Wait: 1.5
231     MoveL
232       PrePlace_3
233   MoveJ
234     PrePick_2
235     MoveL
236       Pick_4
237     Set Gripper=Apagar
238     Wait: 1.5
239     MoveL
240       PrePick_2
241     MoveJ
242       PrePlace_4
243     MoveL
244       Place_4
245     Set Gripper=Encender
246     Wait: 1.5
247     MoveL
248       PrePlace_4
249   MoveJ
250     Set Gripper=Apagar
251     PrePick_R4
252     MoveL
253       Place_R4
254     Set Gripper=Encender
255     Wait: 1.5
256     MoveL
257       PrePick_R4
258     MoveL
259       PrePlace_4a
260     MoveL
261       Waypoint_1
262     Set Gripper=Apagar
263     Wait: 1.5
264     MoveL
265       PrePlace_4a
266   MoveJ
267     pas:=5
268     PrePick_R3
269     MoveL
270       Pick_R3
271     Set Gripper=Encender
272     Wait: 1.5
273     MoveL
274       PrePick_R3
275     MoveL
276       PrePlace_R3b
277     MoveL
278       Place_R3
279     Set Gripper=Apagar
280     Wait: 1.5
281     MoveL
282       PrePlace_R3b

```

Figura A31. Captura del programa FAS200_OPC (Font: Autor)

```
283 MoveJ
284 PrePick_E4
285 MoveL
286 Pick_E4
287 Set Gripper=Encender
288 Wait: 1.5
289 MoveL
290 PrePick_E4
291 MoveJ
292 PrePlace4b
293 MoveL
294 PlaceE4
295 Set Gripper=Apagar
296 Wait: 1.5
297 MoveL
298 PrePlace4b
299 Set Gripper=Apagar
300 MoveJ
301 pas:=5
302 PrePick_E3
303 MoveL
304 Pick_E3
305 Set Gripper=Encender
306 Wait: 1.5
307 MoveL
308 PrePick_E3
309 MoveJ
310 PrePlace_E3b
311 MoveL
312 Place_E3
313 Set Gripper=Apagar
314 Wait: 1.5
315 MoveL
316 PrePlace_E3b
317 MoveJ
318 PrePick_T4
319 MoveL
320 Waypoint_4
321 Set Gripper=Encender
322 Wait: 1.5
323 MoveL
324 PrePick_T4
325 MoveL
326 PrePlaceT4
327 MoveJ
328 Place_T4
329 Set Gripper=Apagar
330 Wait: 1.5
331 MoveL
332 PrePlaceT4
333 MoveJ
334 PrePick_T3
335 MoveL
336 Pick_T3
337 Set Gripper=Encender
338 Wait: 1.0
339 MoveL
340 PrePick_T3
341 MoveL
342 PrePlace_T3
343 MoveJ
344 Waypoint_5
345 Set Gripper=Apagar
346 Wait: 1.0
347 MoveL
348 PrePlace_T3
349 MoveJ
350 Pos_Ini
351 Set DO[2]=Apagar
352 working:= False
353 temps_cicle: Detener
354 units:=units+2
355 working:= False
356 Thread_1
357 Write OPC UA server: Robot_working
358 Write OPC UA server: Step
359 Write OPC UA server: Units_completed
360 Write OPC UA server: cycle_time
361 Write OPC UA server: session_time
362 Write OPC UA server: ask_admin
363 Write OPC UA server: Message
```

Figura A32. Captura del programa FAS200_OPC (Font: Autor)