



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

TREBALL FI DE GRAU

Grau en Enginyeria Electrònica Industrial i Automàtica

**DESENVOLUPAMENT D'UN CLIENT OPC PER A LA
MONITORITZACIÓ D'UN SISTEMA DE SENSORS SENSE FILS**



Memòria, Pressupost i Annexos

Autor: Albert Rovira i Pedragosa
Director: Javier Gámiz i Caro
Co-Director: Moisès Graells i Sobre
Convocatòria: Juny 2017

Resum

En el present treball es tracten els passos seguits per a la lectura de la temperatura proporcionada per unes sondes sense fils, mostrant com ha de ser la configuració de les comunicacions de tots els elements. Les sondes es connecten a un receptor per Zigbee i aquest al seu torn és servidor OPC. Amb un programa SCADA es monitoritzen les dades i es crea un registre, utilitzant formats el més universals possible, en aquest cas .csv i .xml.

L'objectiu és crear un sistema SCADA el més robust i funcional possible per al laboratori de química utilitzant els aparells de que es disposa.

Per aconseguir-ho es fa una valoració dels diferents sistemes SCADA que es poden implementar i es decideix quina opció és la més adequada, explicant-ne la configuració. També es tracta la creació d'un codi en llenguatge Python que implementa el protocol típic d'Internet de les coses (MQTT) i permet que des d'un Smartphone o qualsevol dispositiu intel·ligent es tingui accés a les dades de la sonda en temps real.

Resumen

En el presente trabajo se tratan los pasos seguidos en la lectura de la temperatura proporcionada por unas sondas sin hilos, mostrando como tiene que ser la configuración de las comunicaciones entre todos los elementos. Las sondas se conectan a un receptor por Zigbee y este a su vez es servidor OPC. Con un programa SCADA se monitorizan los datos y se crea un registro, utilizando formatos lo más universales posible, en esta ocasión .csv i .xml.

El objetivo es crear un sistema SCADA lo más robusto i funcional posible para el laboratorio de química utilizando los instrumentos de que se dispone.

Para conseguirlo se hace una valoración de los distintos sistemas SCADA que se pueden implementar i se decide cual es la opción más adecuada, explicando su configuración. También se trata la creación de un código en lenguaje Python que implementa el protocolo típico de Internet de las cosas (MQTT) i permite que des de un Smartphone o cualquier dispositivo inteligente se pueda acceder a los datos de la sonda en tiempo real.

Abstract

This project is about the steps followed to read the temperature coming from a wireless sensor, showing in detail de configuration used in all the communications between elements. The sensors are connected to a receiver using Zigbee and the receiver is an OPC server. The monitorization is kept by an SCADA which also creates a historical file, using formats the more universal the better, in this case .csv and .xml.

The objective is to create an SCADA system the most robust and functional possible, using the instruments available in the chemistry laboratory.

To achieve this, an evaluation of the different systems is made, deciding which is the best option and explaining the configuration used. Furthermore, the creation of a Python script, used to publish the information in the typical Internet of Things protocol (MQTT), is explained. This script allows any smart device, such as a Smartphone, to receive the information from the sensors in real time.

Agraïments

Agrair a en Moisès Graells la proposta del projecte i el material prestat. A en Javier Gámiz, per l'atenció i l'ajuda constant i ràpida sense la qual hauria estat impossible. Al personal del servei tècnic de Wonderware, doncs el seu consell en moments clau va ser molt important i més tenint en compte que no sóc client. A Newport, per proporcionar de manera gratuïta la llicència de l'OPC Server. I, finalment, a la meva família, per l'ajuda permanent al llarg de la carrera.



Índex

RESUM	
RESUMEN	I
ABSTRACT	II
AGRAÏMENTS	III
1. PREFACI	1
1.1. Origen del treball	1
1.2. Motivació	1
2. INTRODUCCIÓ	3
2.1. Objectius del treball.....	3
2.2. Abast del treball.....	3
3. MODEL OSI I ESTÀNDARDS DE COMUNICACIÓ	5
3.1. Model OSI	5
3.2. Estàndard de Comunicació OPC	6
3.2.1. OPC Classic	7
3.2.2. OPC UA	7
3.3. Estàndard de Comunicació MQTT	7
3.4. Protocol Zigbee	8
4. MODELS DE COMUNICACIÓ	9
4.1. Client-Servidor	9
4.2. Publicador-Subscriptor	9
5. EL SISTEMA FÍSIC	11
5.1. Sensor de temperatura.....	11
5.1.1. Configuració Sonda	12
5.2. Receptor.....	15
5.2.1. Configuració del receptor	16
5.2.2. Configuració del servidor OPC	19
6. INTERFÍCIE SCADA	22
6.1. Configuració de l'SCADA.....	24

6.1.1.	Configuració SCADA Intouch	25
6.1.2.	Configuració Rapid SCADA	29
6.1.3.	Eines de Diagnosi	34
6.2.	Estructura i Interfície final SCADA	35
7.	IOT DEL SISTEMA	38
7.1.	Funcionament	40
8.	ANÀLISI DE L'IMPACTE AMBIENTAL	44
9.	PLANIFICACIÓ	45
	CONCLUSIONS	47
	PRESSUPOST I ANÀLISI ECONÒMICA	49
	BIBLIOGRAFIA	51
	ANNEX	53

1. Prefaci

1.1. Origen del treball

El treball va ser una proposta d'en Moisès Graells. Tenia a la seva disposició un conjunt receptor i sondes que no estava utilitzant perquè no tenia un programa o un sistema que li dones l'opció d'obtenir les dades de manera remota. S'havien fet propostes i intents d'obtenir les dades de temperatura de la sondes però el sistema no era robust i quan hi havia un petit error o imprevist no funcionava.

1.2. Motivació

Sempre m'ha agradat aplicar els conceptes teòrics a la realitat. En la proposta d'en Moisès vaig veure l'oportunitat de portar a la pràctica conceptes teòrics apresos al llarg de la carrera. Una altre fet que em va portar a fer aquest treball és que mai he estat defensor de grans treballs o estudis que no tenen aplicació pràctica.

A més a més sempre m'he sentit molt agraït a la universitat i si puc dedicar temps i esforços a millorar-ne els sistemes això m'omple.

2. Introducció

2.1. Objectius del treball

L'objectiu principal és obtenir de forma remota i automatitzada les dades provinents d'unes sondes de temperatura que serviran per controlar i supervisar un determinat procés. A més a més, també es pretén donar als estudiants d'enginyeria química un primer contacte amb sistemes SCADA, molt utilitzats en la indústria. Intentant que els mateixos estudiant hagin de fer una petita part de la configuració perquè tractin una mica amb comunicacions.

2.2. Abast del treball

Cal configurar totes les comunicacions i entendre el seu funcionament. Fer una valoració de desenvolupadors SCADA i triar el que més prestacions aportí per al sistema donat, que ha de permetre l'opció de crear un històric en un format accessible per als estudiants. Paral·lelament s'ha de dissenyar un sistema d'IoT (Internet of Things), que serveixi per introduir el projectista, els estudiants i perquè no, algun professor, al món que s'obre i fer-se una idea de les possibilitats que ofereix.

3. Model OSI i Estàndards de comunicació

3.1. Model OSI

El model OSI (Open System Interconnection) neix de la necessitat d'aplicar uns estàndards de comunicació. Als anys 80 diferents empreses tenien els seus propis sistemes de comunicació propietaris i la comunicació només existia en sistemes de la mateixa empresa. Si s'havia d'ampliar la comunicació únicament ho podia fer una empresa i hi havia dificultats d'interconnexió entre diferents sistemes. És en aquest context que la ISO (International Organization for Standardization) decideix crear un model de referència basada en nivells per a la comunicació entre ordinadors. Aquest model es divideix en 7 nivells i tot i que no s'ha aplicat mai en la seva totalitat és molt útil per dividir la comunicació en capes que ajuden a la comprensió, disseny i manteniment.

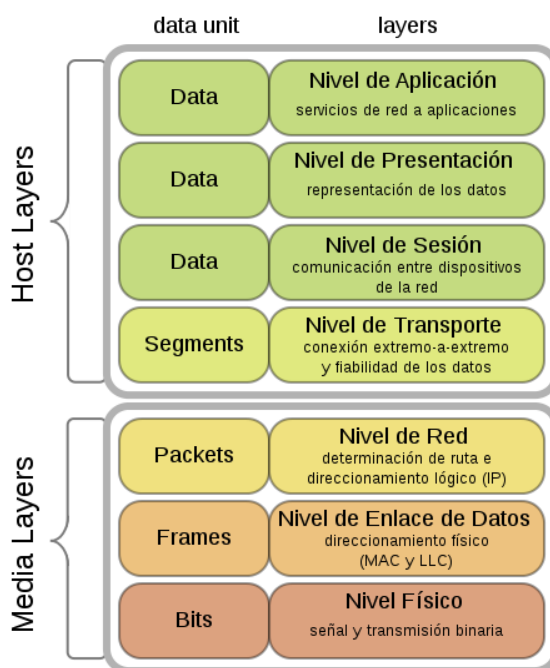


Figura 3.1.1. Model OSI (Font: Wikipedia)

Seguidament anem a donar quatre pinzellades de cada nivell:

- Nivell d'aplicació: Nivell situat a l'extrem superior. Es defineixen els protocols i les interfícies que utilitzarà l'usuari.

- Nivell de presentació: S'encarrega de la representació de la informació. Treballa en temes de com s'organitza la semàntica i la sintaxis. També és on es xifren o comprimeixen les dades
- Nivell de sessió: S'ocupa d'establir, mantenir i acabar una sessió entre dos ordinadors.
- Nivell de transport: Té cura del flux de dades. Divideix les dades en paquets.
- Nivell de xarxa: Encarregat d'encaminar el missatge entre un node origen i un destí. Defineix la ruta entre una o més xarxes.
- Nivell Enllaç: S'estableix una línia de comunicació lliure d'errors. Es defineix el direccionament físic, l'accés al medi i la detecció d'errors.
- Nivell Físic: Nivell inferior del model. Es defineixen les característiques físiques del medi de transmissió. Fibra òptica, parell trenat, connectors, velocitat de transmissió...

3.2. Estàndard de Comunicació OPC

OPC és un estàndard de comunicació entre dispositius de control obert (no propietari), situat al nivell d'aplicació del model OSI. Permet la interoperabilitat i l'intercanvi d'informació independentment de la plataforma. Es pot utilitzar en un arquitectura client-servidor i servidor-servidor, donant accés a dades en temps real, alarmes, històric i més.



Figura 3.2.1. Logotip Fundació OPC (Font: OPC Foundation)

Es va originar al 1996, provinent del protocol OLE (Object Linking and Embedding) propietat de Microsoft. De fet les sigles OPC signifiquen: OLE for Process Control.

El protocol està regulat per la OPC Foundation, un consorci amb participació d'un gran nombre d'empreses, unes 450.

OPC té diferents versions, no necessàriament compatibles entre elles, que es poden agrupar en dues grans famílies. Aquestes són OPC Classic i OPC Unified Architecture (OPC UA).

3.2.1. OPC Classic

Consta de tres subgrups. Tenen la característica d'estar basats en la tecnologia Microsoft COM/DCOM (Distributed Component Object Model). Els principals problemes que presenten és que només es poden implementar en un sistema Microsoft i que no són una estructura única. Els subgrups són:

- OPC Data Access (DA): Possiblement el més estès. Destinat a l'intercanvi de dades en temps real. Cada dada inclou valor, data de la mesura i qualitat.
- OPC Alarms & Events (OPC EA): Intercanvi d'alarmes i esdeveniments.
- OPC Historical Data Access (OPC HDA): Servidor OPC que emmagatzema dades en una base de dades i defineix com els clients poden accedir-hi.

3.2.2. OPC UA

Aposta actual d'OPC Foundation. Engloba totes les funcionalitats de l'OPC Classic i és compatible amb tots els sistemes operatius, Windows, Linux i Android.

3.3. Estàndard de Comunicació MQTT

Estàndard de comunicació màquina-màquina (M2M) obert. Engloba els nivells sessió, presentació i aplicació del model OSI. Es tracta d'un protocol de comunicació de tipus publicació-subscripció molt lleuger (consumeix pocs recursos) ideal per aplicacions d'internet de les coses (IoT), ordinadors de placa i mòbils intel·ligents.

Fou creat per Andrew James Stanford-Clark, investigador d'IBM, i Arlen Nipper, d'ARCOM (ara Eurotech) al 1999.



Figura 3.3.2. Logotip MQTT (Font: mqtt.org)

3.4. Protocol Zigbee

Zigbee és un protocol sense fils que va des del nivell físic fins al nivell d'aplicació del model OSI. En els nivells físic i de control d'accés al medi està inclòs a l'estàndard IEEE 802.15.4 que regula les bandes de 2,4 GHz, 900 MHz i 868 MHz. Destaca pel seu baix consum d'energia i la distància d'abast.

El protocol es va concebre al 1998 i es va estandarditzar al 2003. La ZigBee Alliance, integrada per importants empreses del món tecnològic és l'encarregada de mantenir el protocol.



Figura 3.4.1. Logotip Zigbee (Font: ZigBee Alliance)

4. Models de comunicació

A efectes pràctics per aquest treball és important entendre dos tipus de models de comunicació.

4.1. Client-Servidor

En aquest model hi ha un ordinador que té la informació (servidor) i un altre que sol·licita la informació (client). Perquè tingui lloc l'intercanvi el client ha de demanar iniciar una sessió de comunicació amb el servidor. Exemples d'aplicacions que segueixen aquest model ho són els Emails i la World Wide Web. La següent figura n'il·lustra el funcionament.

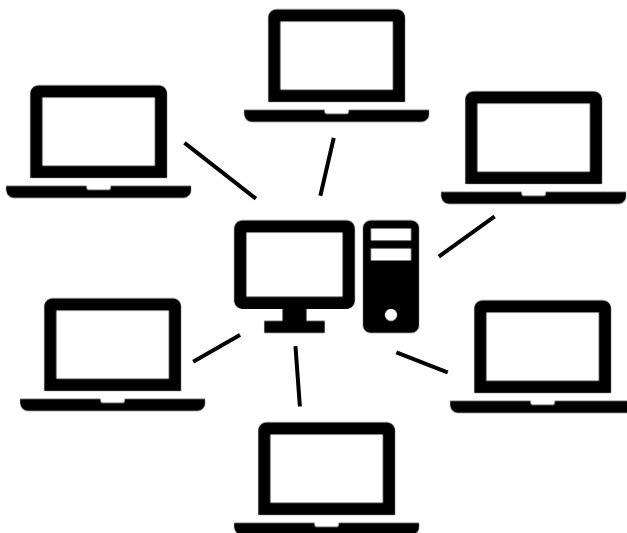


Figura 4.1.1. Model Client-Servidor

4.2. Publicador-Subscriptor

És un sistema de dades on hi ha un publicador que penja informació a un Broker. Aquesta Broker és accessible a multitud de subscriptors i s'ha d'entendre com un simple gestor de la informació. Les dades s'agrupen per Topics i un aparell es pot subscriure a un Topic. Els aparells subscriptors a uns Topics van rebent les informacions que s'hi penjen. Si cap aparell està subscript a un Topic, les dades que si publiquen es perden. Aquest sistema queda representat a la figura següent.

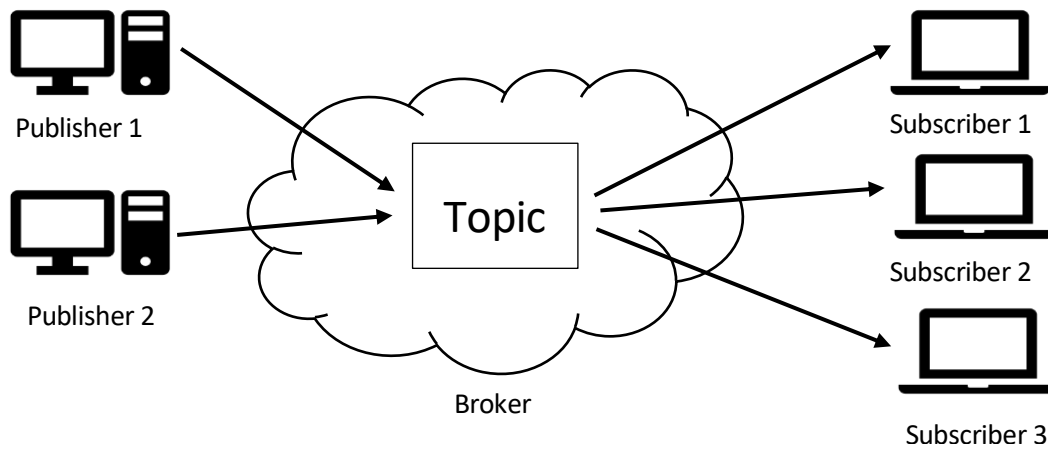


Figura 4.2.1. Model Publicador-Subscriptor

5. El sistema físic

5.1. Sensor de temperatura

El sensor de temperatura és del fabricant estatunidenc Omega (actualment Newport), model UWRTD-2-NEMA. Es pot configurar per ser utilitzat amb diferents tipus de sondes, sempre RTD (resistència variable amb la temperatura) Pt-100. En el cas d'estudi s'utilitza una sonda Pt-100 amb coeficient $\alpha=0,00385 \Omega/K$.

Les característiques més importants de la sonda són:

- Rang: -200 a 850 °C.
- Exactitud: $\pm 0,5$ °C.
- Resolució: 1 °C.
- Temperatura Operació: -10 a 70 °C.
- Distància operació RF:
 - o Exterior: fins a 120m
 - o Interior: fins a 40m



Figura 5.1. Imatge del sensor de temperatura UWRTD-2-NEMA (font: Newport)

El sensor és pot configurar amb un software que la companyia distribueix de manera lliure i que corre sobre un sistema operatiu Windows. El programa porta per nom "End Device Configuration Wizard". Ens podem connectar amb el sensor a través d'un cable USB Mini-B. Els paràmetres que es poden

modificar del sensor són els relatius a la comunicació per radiofreqüència (ZigBee), és a dir, l'adreça de transmissió, el canal de RF, l'identificador de la xarxa i l'adreça del receptor. També és pot modificar el tipus de sensor perquè coincideixi amb l'instal·lat. S'ha d'especificar si el sensor és de 2 o 3 fils, el coeficient α i el nombre de mostres que es volen prendre per unitat de temps.

En relació a aquest últim paràmetre és important configurar-lo bé perquè d'ell dependrà la vida de la bateria. Seguidament s'adjunta una taula amb la relació que tenen el número de mostres i la bateria.

For Model UWTC-2, UWRTD-2, UWRH-2

Transmit Time	Estimated Battery Life
1 Sample/2 Seconds	6 days
1 Sample/3 Seconds	9 days
1 Sample/5 Seconds	15 days
1 Sample/10 Seconds	30 days
1 Sample/15 Seconds	45 days
1 Sample/30 Seconds	90 days
1 Sample/45 Seconds	135 days
1 Sample/60 Seconds	180 days

Taula 5.1.1. Relació entre el temps de mostreig i la durada de la bateria (font: Newport)

5.1.1. Configuració Sonda

Com ja s'ha dit, és necessari tenir instal·lat un programa d'Omega anomenat "End Device Configuration Wizard" i un cable USB a USB Mini-B.

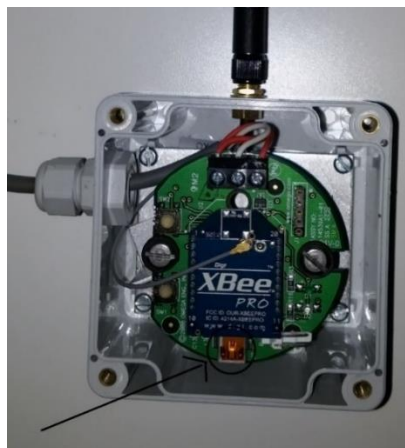


Figura 5.2. Imatge de l'interior del sensor de temperatura UWRTD-2-NEMA

Es connecta el sensor amb l'ordinador a través del port assenyalat a la figura 5.2. Amb el sensor connectat a l'ordinador obrim el software. La pàgina d'inici és la que es mostra a la figura 5.3.

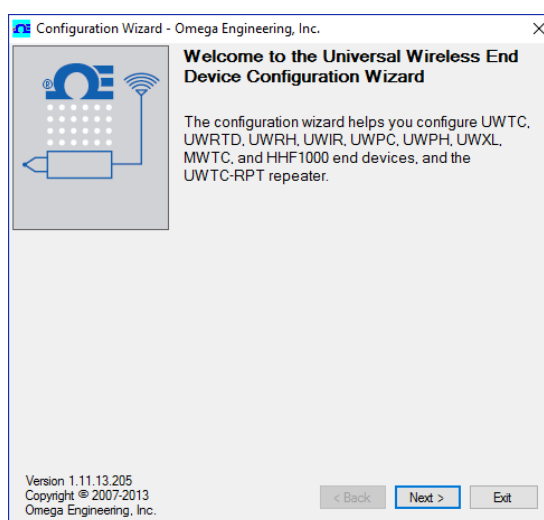


Figura 5.3 Pàgina d'inici de l'assistent

Es van seguint les indicacions que dona l'assistent i s'arriba a la finestra de la figura 5.4. Quan s'és en aquesta finestra cal fer una seqüència amb els polsadors del sensor. Aquesta és: polsar el SW1, mantenint-lo polsat, polsar l'SW2, deixar l'SW2 i deixar l'SW1.

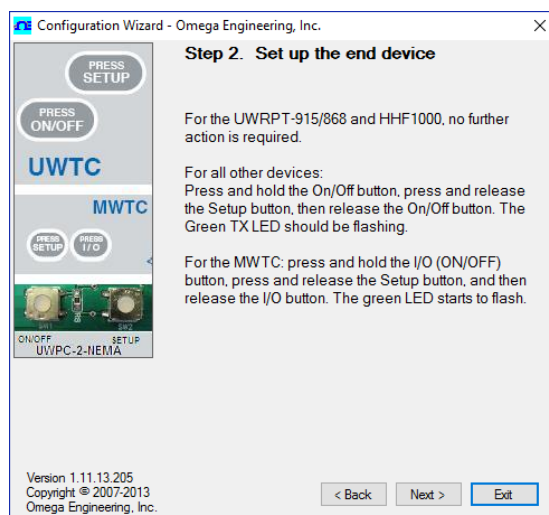


Figura 5.4. Pàgina on fer la seqüència indicada.

Seguidament es fa clic a següent, següent i s'arriba a la pàgina on es pot modificar la configuració del sensor. Aquesta pàgina es mostra a la figura 5.5.

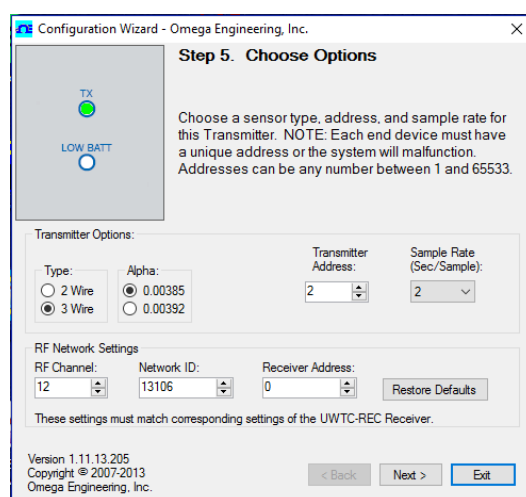


Figura 5.5. Pàgina amb la configuració del sensor.

En aquesta pàgina és on hi ha la configuració del sensor. Els paràmetres més importants a modificar per l'usuari són l'adreça de transmissió i el període de mostreig. Una vegada configurat el sensor es fa clic a següent. Així s'envia la configuració a la sonda que quan s'hagi configurat s'apagarà. Es polsa una vegada el SW1 (polsador on/off) i ja es pot seguir utilitzant.

5.2. Receptor

Dispositiu també del fabricant Omega, model UWTC-REC3. Concentra els senyals procedents de fins a 32 sensors comunicats per Zigbee. Té un port de comunicació Ethernet i soporta els protocols TCP/IP, ARP, ICMP, DHCP, DNS, HTTP i Telnet. A part també té un servidor de pàgina Web que permet configurar-lo i veure Applets Java amb dades en temps real. Newport va crear un software que permet configurar el dispositiu. Aquest software s'anomena iConnect. No obstant, la mateixa configuració és pot fer amb un explorador web.

Omega (Newport) distribueix un servidor OPC per al seu receptor. És un factor molt important, ja que significa una porta a un llenguatge estandaritzat i admès per tots o casi tots els fabricants. Utilitzar aquest software ens permet treballar amb hardware i software de diferents empreses i fer accessible i poder tractar les dades amb moltíssimes aplicacions i SCADAs. Si no existís aquest Servidor les temperatures de les sondes només serien accessible per als programes propis d'OMEGA.

El programa del que parlem es diu Newport OPC Server i posen a accés lliure una versió demostració de dues hores. Com que ens interessa tenir un sistema el màxim de potent sense limitacions ens contacta la companyia, amb seu als E.E.U.U. demanant si, donat el cas que la UPC es una institució acadèmica i sense finalitat productiva, ens podrien facilitar una llicència sense límits de temps. Per sorpresa la resposta és que sí, de manera que s'aconsegueix que el servidor OPC no estigui limitat a dues hores.

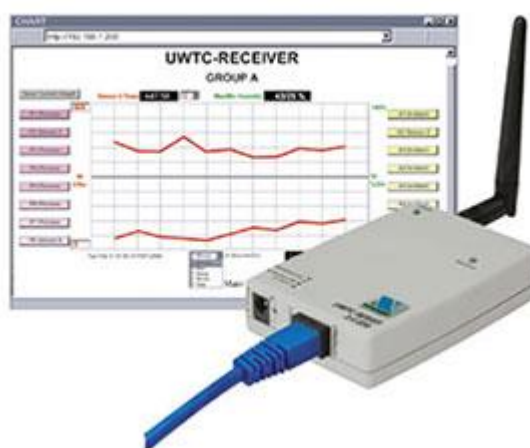


Figura 5.7. Imatge del receptor UWTC-REC3 (font: Newport)

5.2.1. Configuració del receptor

El receptor d'Omega UWTC-REC3 suporta varis protocols de xarxa, com ara TCP/IP. També té una interfície gràfica que permet configurar-lo. De fet, té un servidor web que permet la lectura de dades i gràfics. No obstant, està dissenyat per treballar amb una versió de JAVA anterior a les actuals i és molt difícil aconseguir llegir les dades.

Per configurar el receptor el que cal fer és obrir un explorador web i amb el receptor connectat a la xarxa on tinguem el nostre ordinador personal escriure la direcció IP del receptor on normalment escrivim la direcció web. Aquesta direcció IP és fixa i sempre serà la mateixa. En cas de perdre aquesta direcció i no recordar-la es pot "resetejar" de manera que el receptor torni a una direcció IP coneguda.

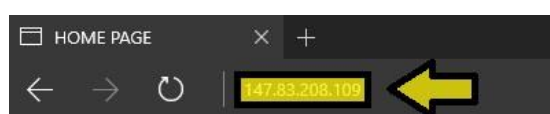


Figura 5.8. Com accedir al servidor Web del receptor

En el cas de l'exemple la direcció IP és 147.83.208.109. Una vegada fet això apareixerà la pàgina de la figura 5.9.

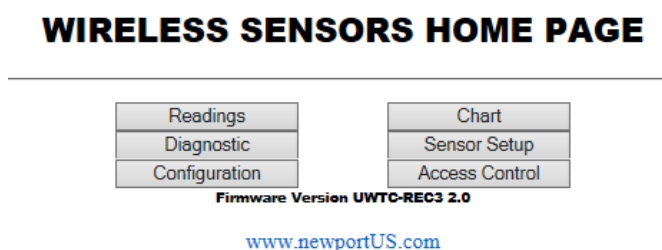


Figura 5.9. Servidor web del receptor

En aquesta pàgina una de les opcions que ens interessa és la de configuració (Figura 5.10). Aquí es pot donar un nom al receptor, configurar la comunicació TCP i la potència de la comunicació de radiofreqüència. Una altre opció que també es pot configurar és l'Access Control (Figura 5.11). Per entrar a aquesta pestanya serà necessària una contrasenya. Aquesta és Tacer7.

CONFIGURATION

Name

Temperature

TCP Connections Port

PID 13106

Channel 12

Transmission Power

Please refer to your local regulations for the allowed maximum transmission power

Radio Version 2.0

[Main Menu](#)

Figura 5.10. Pàgina de configuració

ACCESS CONTROL

Login Password

Admin Password

Web Server

Host Name

MAC Address

IP Address

Gateway Address

Subnet Mask

[Main Menu](#)

Figura 5.11. Pàgina d'Access Control

Una altre opció a tenir en compte a l'hora de configurar la comunicació és com ens connectem amb el receptor i si hem de canviar algun aspecte de la configuració de xarxa del nostre ordinador personal.

Si el receptor està comunicat a la www(world wide web) no cal que ens preocupem, però si ens connectem al receptor de manera local amb un cable Ethernet hem de canviar la configuració del protocol IP. Això ho hem de fer anant a la configuració de xarxa i canviar la configuració de les propietats d'Ethernet.

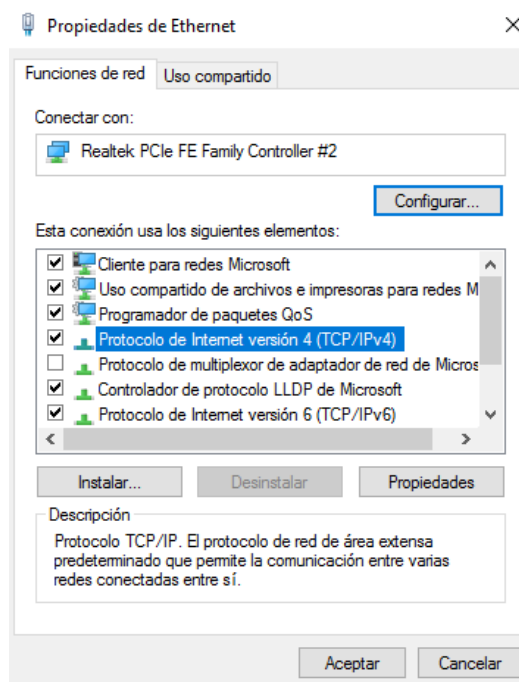


Figura 5.12. Finestra amb les propietats d'Ethernet

Fem clic a les propietats del protocol d'internet a la versió 4 i s'escull una IP fixa d'un rang proper a la del receptor. Si volem podem utilitzar una màscara de subxarxa, és recomanable.

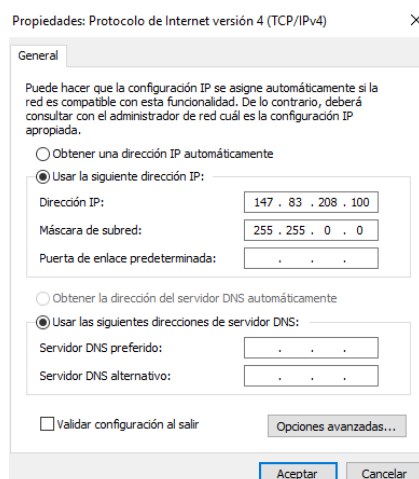


Figura 5.13. Finestra amb la configuració del protocol IP

5.2.2. Configuració del servidor OPC

Després de descarregar el servidor de la pàgina web de Newport iniciem l'executable "OPC Configuration", a la carpeta de Newport. En un inici, la configuració està en blanc, com es mostra a la figura.

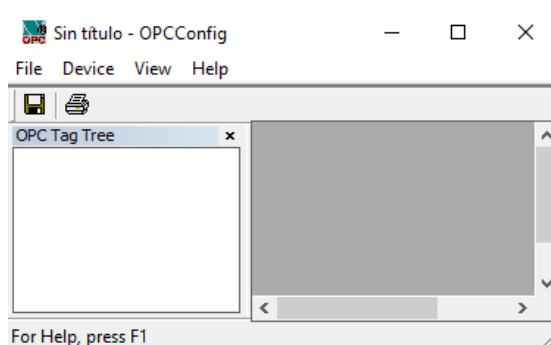


Figura 5.14. Finestra inicial de la configuració

En aquesta figura cal fer clic a Device -> Add -> Communication. Llavors apareix la pestanya que es mostra a la figura.

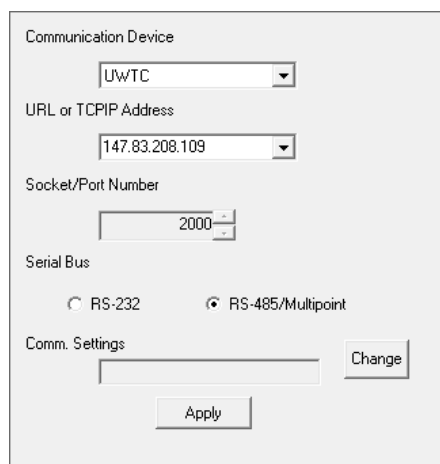


Figura 5.15. Finestra de configuració del receptor

Aquí cal indicar el tipus de receptor (en aquest cas UWTC) i la direcció IP. El port queda fix i la configuració sèrie (dues últimes opcions) no afecten, ja que ens comunicarem per Ethernet amb el protocol TCP/IP.

Ara cal afegir els dispositius de mesura. Per fer-ho fem clic a Device -> Add -> Measurement. Aquí la finestra que apareix és la següent.

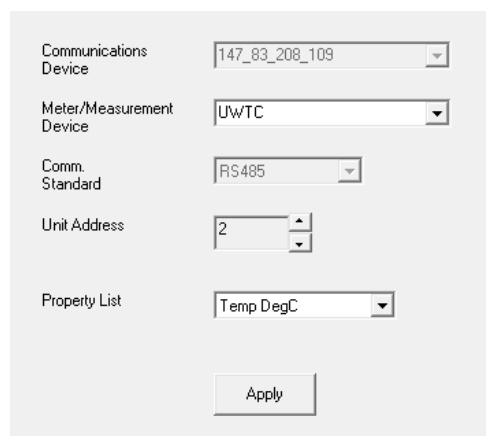


Figura 5.16. Finestra configuració del sensor

Cal seleccionar el tipus de dispositiu, UWTC, la direcció, en aquest cas 2 i les unitats de la temperatura. Tot seguit cal reiniciar l'executable amb la configuració i el sistema queda com a la figura.

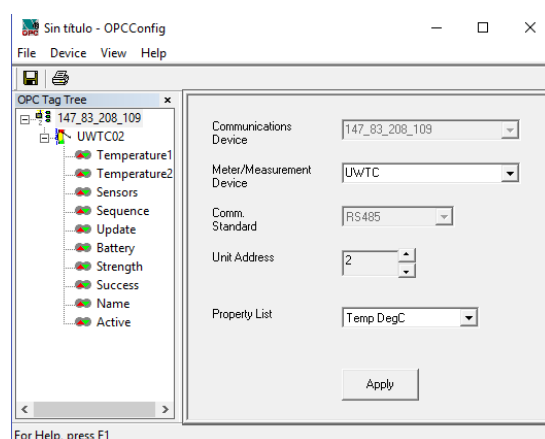


Figura 5.17. Finestra configuració final

Aquí veiem els tags que té aquest sensor. La forma d'accedir a aquests tags és la unió de la direcció IP, canviant els punts per guions baixos, el nom del dispositiu de mesura i el nom del tag. Totes les informacions unides amb un punt. Per exemple, el nom de l'item Temperatura 1 serà 147_83_208_109.UWTC02.Temperature1.

Una vegada feta la configuració del servidor OPC és interessant veure que realment funciona. Això ho podem verificar connectant un client OPC i llegint els tags. Existeixen alguns clients gratuïts per fer proves. Un d'ells és el client de Matrikon, que es pot descarregar de la seva pàgina web després de

registrar-se. El nom del client és MatrikonOPC Explorer. Una bona opció és descarregar-se també el Matrikon OPC Simulation Server. Un servidor que ens pot servir per provar les aplicacions clients.

Una vegada tenim el software instal·lat cal que ens mostri els servidors OPC que tenim instal·lats. Per fer-ho cliquem dues vegades sobre el dibuix de la pantalla amb el nom localhost. Després ens podem connectar al servidor OPC i crear un grup amb els tags que volem veure. És una bona pràctica per confirmar que tota la configuració funciona i llegim els tags. Si no llegim una temperatura coherent caldrà revisar la comunicació per veure on ens esta fallant. El client de Matrikon esta molt aconseguit i es capaç de connectar-se a quasi bé tots els servidor OPC, independentment de la versió o de la configuració que hi hagi al nostre sistema de comunicació COM/DCOM. A la figura es mostra un exemple de comunicació al servidor OPC llegint tots els tags disponibles per comprovar que la configuració és bona. Com veiem apareix una aplicació que descompte les dues hores de la versió demo.

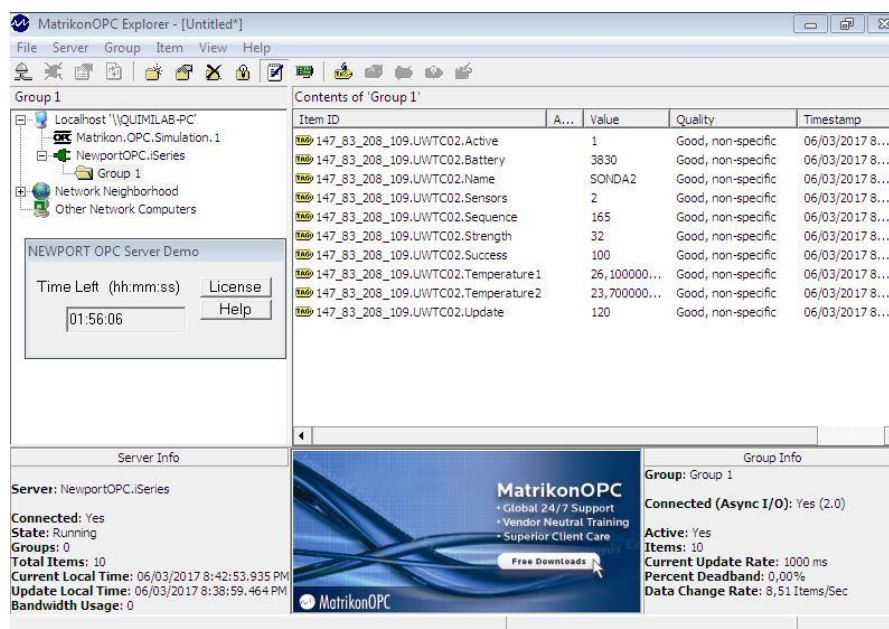


Figura 5.18. Configuració Matrikon OPC Explorer

6. Interfície SCADA

Un SCADA (Supervisory Control and Data Acquisition) és un sistema que permet, com el seu nom diu, l'adquisició de dades, la supervisió i el control. Es tracta d'un sistema que permet a l'usuari saber en tot moment quin és l'estat d'un determinat procés, ja siguin variables físiques (temperatura, humitat, pressió, viscositat...) o accions (comprimint, escalfant, refredant..) que es duen a terme en un procés automàtic. És un sistema escalable i pot servir per controlar petits processos o fàbriques i empreses senceres.



Figura 6.1. Exemple d'SCADA (Font: Industrial Control System Integration)

És tracta doncs d'una font d'informació en temps real que permet conèixer a fons un procés, controlar-lo i treballar per millorar-lo. Aquest factors aplicats a un àmbit de producció s'acaben traduïnt en més eficiència, uniformitat i qualitat.

Al mercat actual hi ha una gran varietat d'empreses que tenen el seu propi software de desenvolupament d'SCADA. Gairebé totes les grans firmes destinades a l'automatització de processos tenen el seu. Escollir entre un software o un altre no canvia gaire el resultat final, canviarà sobretot la interfície, els dibuixos, els gràfics i les formes de presentar les dades, però en quant a funcionalitats no hi ha grans diferències.

No obstant, com en el món de la informàtica en general si que es pot diferenciar entre dos grans grups. Els sistemes SCADA registrats i de pagament, dels quals s'acostuma a distribuir una versió gratuïta amb importants limitacions o els sistemes SCADA lliures i gratuïts.

Com a norma general a la indústria hi ha tendència a escollir SCADAs de pagament, doncs porten associats una infraestructura de suport dedicada, ràpida, eficient i amb “garanties” de continuïtat en el temps. A més a més acostumen a donar uns gràfics bons.



Figura 6.2. Exemple d'SCADA SIMATIC WinCC de Siemens

Per contra, els sistemes lliure ofereixen una gran facilitat de distribució i aprenentatge. Acostumen a tenir uns gràfics bastant pitjors que els sistemes de pagament, però en quant a funcionalitat no queden molt lluny.



Figura 6.3. Exemple d'SCADA neoSCADA (lliure)

En un inici en aquest treball s'havia previst treballar amb l'SCADA Intouch, de Wonderware. És tracta del software amb el que es treballa en diverses assignatures del grau en Electrònica Industrial i Automàtica i, per tant molts estudiants i professors de l'EEBE el coneixen. Això facilita el manteniment

i la resolució de problemes en cas de que apareguin. No obstant, en un inici es creia que al tractar-se d'una projecte per l'escola es podria disposar de un sistema "full", no limitat per una llicència demostració. No és així, ja que la UPC no disposa de cap llicència que no estigui limitada. Veient això i que el software tampoc ofería una solució òptima per al projecte es va decidir passar a una versió lliure.

Una vegada situats en el marc de la utilització d'un software lliure s'obre un ventall bastant ampli d'opcions. Primer de tot cal considerar sobre quin sistema operatiu es vol treballar, si aquest és Windows o Linux. En aquest cas és necessari treballar sobre Windows perquè el servidor OPC només funciona sobre aquesta superfície. Un cop delimitat el sistema operatiu hi ha softwares que queden descartats de facto. Fent una recerca s'arriba a un conjunt de 4 possibles sistemes SCADA. Aquest són Indigo SCADA, neoSCADA, PROMOTIC i Rapid SCADA.

- Indigo SCADA: Software amb moltes funcionalitats, possiblement dels que més n'ofereix. No obstant, es descarta perquè és un software per usuaris avançats amb una configuració difícil.
- neoSCADA: De nou un SCADA amb moltes funcionalitats i a més a més amb una interfície gràfica maca i agradable. Es descarta, com en el cas anterior per ser difícil de configurar i a més proporcionar poca documentació de suport.
- PROMOTIC: Software amb una versió lliure i una de pagament. La versió lliure que ofereixen és molt extensa i els gràfics són agradables, no obstant té algunes mancances en quan a la creació d'històrics. Aquests problemes es solucionen amb la versió de pagament.
- Rapid SCADA: SCADA amb bastanta documentació de suport, audiovisuals i text. Fàcil de trobar, clara i concisa. A més a més, té un fòrum amb preguntes d'usuaris respostes per usuaris i desenvolupadors. L'estructura és marcada i l'usuari n'entén fàcil el funcionament. Tanmateix, el desenvolupament de la interfície gràfica té unes prestacions molt baixes. No té cap llibreria amb objectes ja dissenyats i les animacions són pràcticament nul·les. Tot i així es premia la funcionalitat i facilitat per sobre de l'estètica (que amb una mica de treball tampoc és tant dolenta) i s'escull aquest software per fer el disseny de l'SCADA.

6.1. Configuració de l'SCADA

Com que en un principi la intenció era fer l'SCADA amb Intouch de Wonderware i es té tota la documentació i procediments de com s'ha de descriure la configuració amb Intouch, s'explicarà tant la configuració amb Inotuch com amb Rapid SCADA. D'aquesta manera també es poden veure els pros i contres d'un i altre.

6.1.1. Configuració SCADA Intouch

Wonderware té al mercat el sistema Wonderware System Platform. Es tracta d'un sistema molt potent i dels més utilitzats, aproximadament 1 de cada 3 sistemes SCADA estan fets amb Wonderware (segons diu l'empresa). En aquest punt tractarem la versió del 2014. Actualment ja han llançat la versió del 2017, versió que es mostra millorada i amb apostes valentes i interessants.

Wonderware System Platform consta de les següents aplicacions:

- Application server
- Intouch
- Historian
- Historian Client
- Device Integration
- Information Server

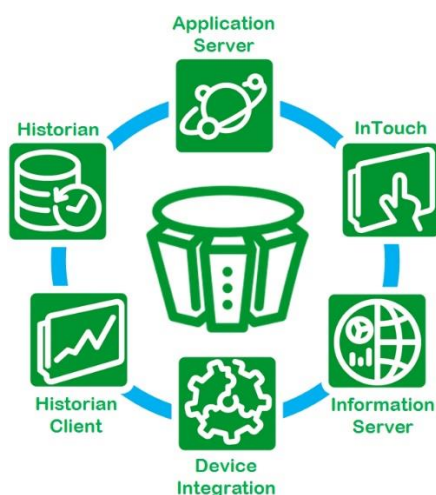


Figura 6.1.1. Aplicacions de la Wonderware System Platform

En aquest projecte les aplicacions que s'haurien d'utilitzar són Intouch i Device Integration. Intouch és on es desenvolupa la part gràfica del projecte i la gestió de l'històric. Device Integration (en el nostre cas OI Gateway) és la part del sistema que es comunica amb l'exterior, en aquest cas amb el servidor OPC de Newport. Es tracta del programa amb més opcions de connexió del mercat, capaç de comunicar-se amb pràcticament qualsevol dispositiu intel·ligent.

Per explicar la configuració es seguirà el flux de les dades, des de l'obtenció fins al registre en una Base de Dades. Es començarà per mirar la configuració necessària d'entrada al sistema. Per entrar dades al sistema cal configurar el portal Device Integration. S'utilitza l'aplicació OI Gateway, que és la que ens

farà de client del servidor OPC i farà de servidor per Intouch. Aquesta aplicació es configura des de la SMC. La visió general que es té s'ofereix a la figura següent

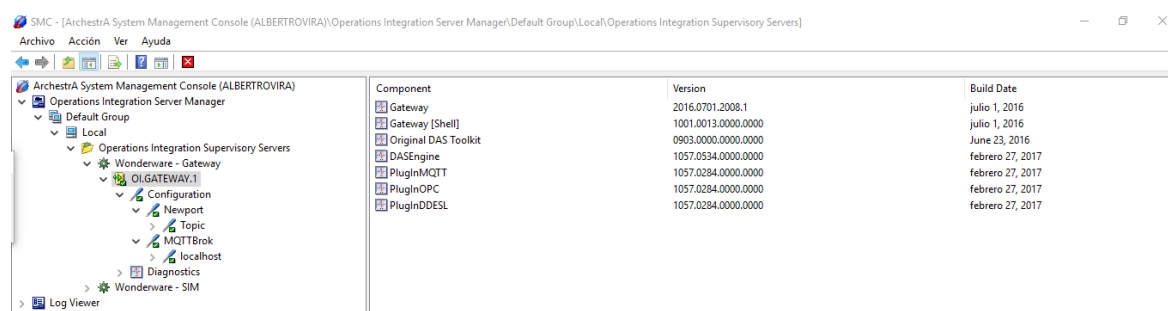


Figura 6.1.2. Visió general SMC

Aquí es pot veure la configuració de la Gateway. És necessari configurar dos elements: el Servidor Newport i l'MQTT broker. Per crear la connexió amb el servidor Newport cal fer clic amb el botó dret sobre Configuration i seleccionar l'opció Add OPC connection. Apareix a la pantalla:

Newport Parameters

Server Node: localhost

Server Name: NewportOPC.iSeries

Reconnect Attempts: 3

Reconnect Period: 30000 ms

Poke Retries: 0

Activate Server Out of Proc

Allow Optional Data Type Suffix In Item Name

Use Synchronous Reads and Writes

Figura 6.1.3. Configuració Newport OPC

Aquí els principals paràmetres a configurar són el node i el nom del servidor. Com es pot veure es poden seleccionar d'una llista que apareix al fer clic als punts suspensius.

Ara cal seleccionar els tags que es vol llegir d'aquest servidor i agrupar-los en un carpeta (Topic). Per fer això, al fer clic amb el botó dret sobre la connexió creada hem de seleccionar crear grup. Podem afegir els items (tags) de forma manual si en sabem el nom, o automàticament fent un "Browse".

Finalment cal configurar el Broker MQTT i seleccionar els items a publicar a la configuració del client OPC.



Figura 6.1.4. Configuració Broker MQTT

Quan la IO Gateway està configurat ja es pot passar a la configuració de l'SCADA. A la nostra aplicació cal crear un accés (comunicació de l'SCADA amb l'exterior). Aquesta funcionalitat es troba a Special-> Acces Names... Fem clic a Add o Modify i apareix la finestra de la figura.

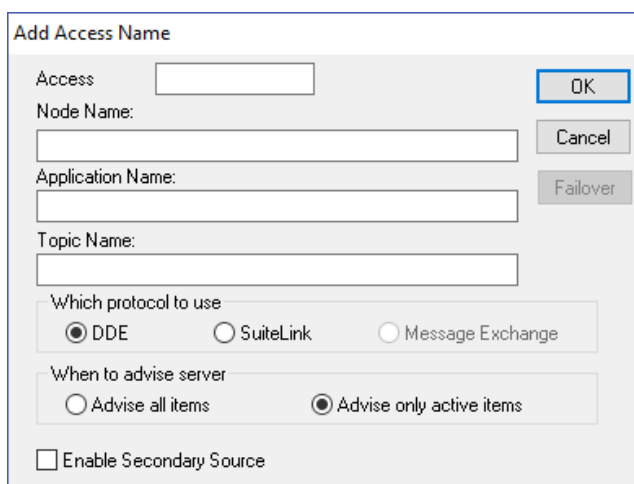


Figura 6.1.5. Configuració Access.

En aquesta figura els espais buits tenen el significat següent:

Access: És el nom amb que s'identifica l'Access a Intouch. És un nom triat pel dissenyador de la interfície.

Node Name: Nom o direcció IP del node on s'executa l'aplicació. En aquest cas serà localhost ja que l'aplicació s'estarà executant al mateix node que l'SCADA. Si l'SCADA estigués situat en un ordinador diferent al de l'aplicació aquí hi aniria la direcció IP o nom del Node que conté la OI Gateway.

Aplicacion Name: Nom de l'aplicació. Aquest nom és fix i ha de ser Gateway.

Topic Name: Carpeta on hi ha els tags seleccionats i el nom que se'ls hi ha donat. El nom és generat per la OI Gateway.

Protocol: DDE o Suitelink. En aquest cas és indiferent.

Advise server: En aquest cas és indifent.

Enable Secondary Source: Cal que estigui desactivat.

Finalment cal configurar els tags. En aquesta comunicació se n'utilitzen 3: sensor actiu, temperatura de la sonda, i bateria del sensor. El que cal fer és anar a la pestanya Special-> Tagname Dictionary..., seleccionar un nom i el tipus necessari. Aquests són:

Sensor Actiu: I/O Discrete.

Temperature Sonda: I/O Real

Bateria: I/O Integer

Finalment es selecciona l'Access Name i es diu el nom de l'Item(nom que hem assignat a la Gateway). Ara ja es pot guardar el tag.

És important que totes les parts casin a la perfecció, sinó la comunicació no funcionarà.

La HistData és una aplicació d'Intouch que permet el bolcat de dades a un csv. Hi ha una forma automàtica de crear aquesta comunicació, però si es fa d'aquesta manera es perd l'opció de canviar paràmetres importants de la configuració, sobretot el més important és el temps de mostreig. Per aquest motiu s'escull fer la configuració de manera manual, més laboriosa però amb la possibilitat de ser molt més modelable. La informació sobre com fer aquesta configuració s'ha extret en la seva totalitat de l'ajuda d'Intouch, al capítol "Manually Extracting Log Data With HistData". Els passos a seguir són:

- Crear un Access Name per l'HistData
- Crear els I/O tags pel HistData
- Crear la finestra HistData
- Executar l'aplicació HistData

A més, cal escriure unes línies de codi ja definides que s'executaran quan es faci clic als botons de la finestra.

Com configurar al complet aquesta aplicació s'explica a l'ajuda d'Intouch, a l'apartat Accessing Historical Tag Values from Other Applications, inclòs dins Data Manager.

6.1.2. Configuració Rapid SCADA

Rapid SCADA té una estructura marcada i diferenciada que permet una clara visió de la configuració. Consta de les aplicacions:

- Server Application: Configuració del servidor Web
- Communicator Application: Gestiona les connexions amb l'exterior, Modbus, OPC, GSM, Email i SNMP.
- Webstation Application: Controla com es crea la pàgina web (és una aplicació interna no modificable per l'usuari).
- Administrator Application: És la base de dades de l'aplicació.
- Table Editor Application: Es defineixen els canals que es volen registrar, mostrar en gràfic i en temps real.
- Scheme Editor Application: Es defineix la interfície gràfica amb imatges valors en temps real i animacions (molt limitat).

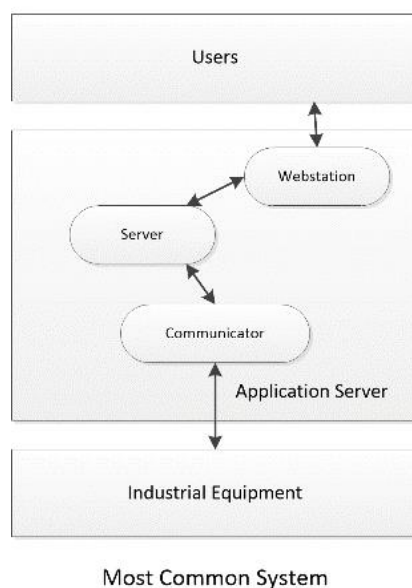


Figura 6.1.6. Estructura del sistema



Aquest tipus d'SCADA es comunica amb l'exterior (PLCs, sensors, actuadors..) amb l'aplicació Communicator. Les dades són recollides i organitzades al servidor que és un servidor Web accessible des de qualsevol punt de la xarxa. Es poden configurar diferents usuaris i donar diferents formats en funció de qui vol accedir al sistema (administrador, personal de producció, manteniment...).

En el tractament de les dades s'utilitzen les aplicacions Communicator i Administrator. Aquí és on es defineixen les línies de comunicació, els dispositius, i els canals (un canal seria l'equivalent a un tag).

Per editar la configuració gràfica s'utilitza el Table Editor i l'Schemes Editor. Amb l'Schemes Editor es crea la interfície gràfica. Aquí és on s'afegeixen les imatges, les animacions... El Table Editor mostra les dades que es volen tractar. També formen part de la interfície gràfica, ja que es poden crear gràfiques i exportar les dades. Aquest és el punt més fluïd d'aquest programa, doncs no té cap llibreria amb imatges predissenyades, botons i objectes animats. Això fa que la interfície de l'Scada sigui molt rígida i no hi hagi moviment. En aquest cas hem d'entendre el disseny gràfic com una simple forma de mostrar valors numèrics o missatge de forma "agradable".

La creació de l'històric amb aquest Scada es molt simple, ja que des de la pàgina web hi ha una opció d'exportar un .xml. Fitxer que s'obre perfectament amb Excel i que és capaç de captar una dada cada 30 segons com a temps mínim. Suficient en aquest projecte.

En la creació d'un Scada amb aquest programa es recomana seguir aquests passos:

1. Anar a l'aplicació Administrator. Aquí crear un dispositiu (a la realitat pot ser físic o virtual) i crear una línia de comunicació o utilitzar una existent. Seguidament fer clic amb el botó dret a Input channels i seleccionar Refresh
2. Passar la configuració de la base de dades al servidor fent clic al botó , situat a la barra d'eines. Reiniciar el servei de servidor i el comunicador, amb la opció que es troba a la mateixa barra.
3. Obrir l'aplicació Communicator. Fer clic amb el botó dret sobre communications lines i escollir la opció import communication lines and devices, la opció amb l'icona . Importem les opcions que necessitem i eliminem les configuracions que no ens serveixin. Es fa clic al tic per aplicar els canvis.
4. Es configuren les opcions de comunicació a l'aplicació Communications.
5. Es torna a l'aplicació Administrator i es creen els canals (tags). La millor opció és que el programa els generi de manera automàtica. Això s'ha de fer a Service -> Create Channels.
6. Seguidament es creen els gràfics (esquemes i taules) amb les respectives aplicacions.
7. Finalment es defineixen els rols (administrador, convidat..) i es defineix que pot veure o modificar cada un d'ells.

Aquesta memòria no té la intenció de ser un manual de com utilitzar aquest programa Scada concret, però si donar una petita pinzellada de com es configura perquè el lector es pugui fer una idea del seu

funcionament. Un punt molt bo d'aquest software és la documentació. Com ja s'ha dit a la anàlisi dels diferents SCADAs.

Tota la documentació esta en Anglès, però no es fa gens difícil d'entendre, a més a més es suposa que els estudiants i els professors tenen un nivell més que suficient per entendre-la. Es disposa de documentació en text, disponible a la pàgina web de Rapid SCADA, a Product -> Documentation i en format vídeo, disponibles al canal de Youtube. El canal té un total de 11 vídeos d'uns 9 minuts en versió en Rus i en Anglès.

Per oferir una primera impressió de com és aquest software s'adjunten un seguit de figures on es mostren les diferents aplicacions.



Figura 6.1.7. Aplicacions amb les seves icones

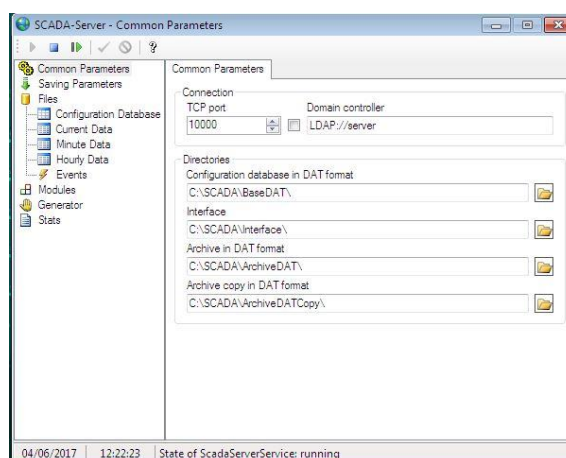


Figura 6.1.8. Aplicació SCADA Server

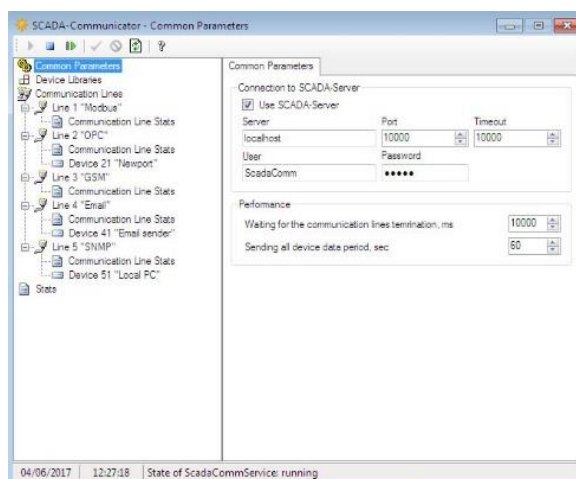


Figura 6.1.9. Aplicació SCADA Communicator

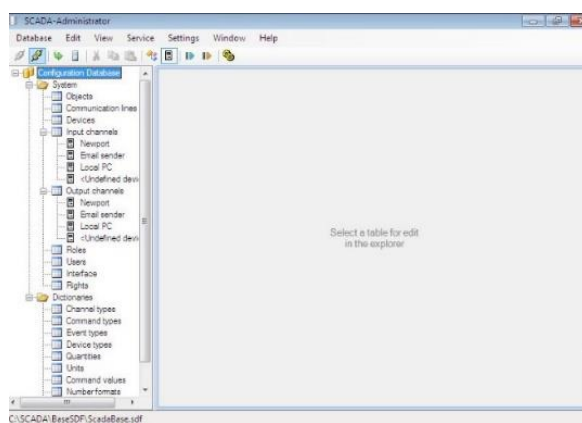


Figura 6.1.10. Aplicació SCADA Administrator

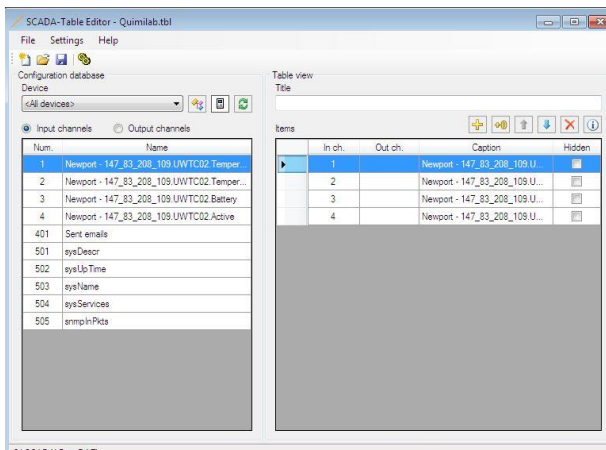


Figura 6.1.11. Aplicació SCADA Table Editor

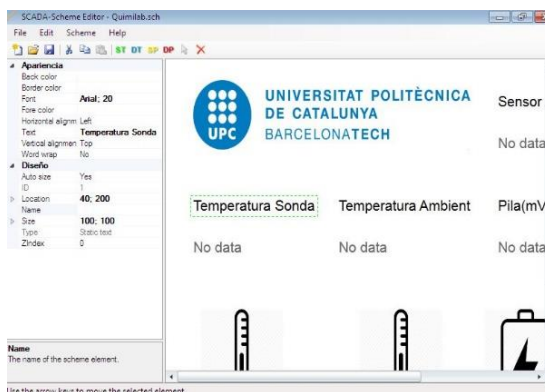


Figura 6.1.12. Aplicació SCADA Schemes Editor

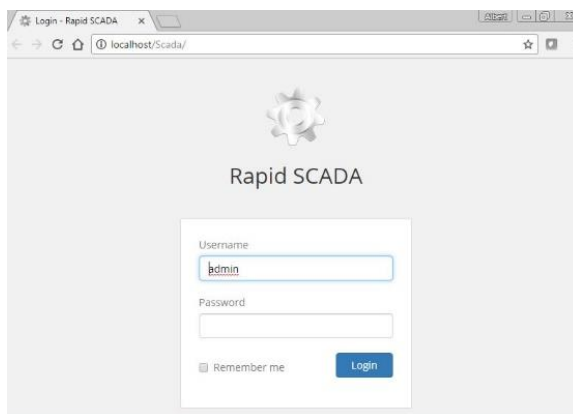


Figura 6.1.13. Pàgina Inicial Servidor Web

6.1.3. Eines de Diagnosi

En el punt de comunicar l'SCADA amb el servidor OPC em vaig trobar amb problemes de connexió, que em van portar a revisar varies vegades la configuració. Sospitava que les versions OPC no fossin compatibles i en aquest context, vaig pensar si no podia existir alguna eina que m'aportes més informació sobre que és el que fallava.

Existeix i es diu OPC Sniffer. De nou una eina gratuïta de Matrikon que ajuda a identificar problemes d'interoperabilitat entre Servidor i Client.

Els passos a realitzar són connectar-se des del client OPC a aquesta aplicació (que es servidor OPC). Llavors apareix una finestra on s'ha d'indicar el servidor amb el qual es vol comprovar la connexió. Es selecciona en aquest cas el servidor de Newport i l'aplicació dóna que hi ha problemes de comunicació.

Veient que realment sembla que hi ha una problema d'interoperabilitat entre servidor i client es busquen mil alternatives per intentar sobrepassar aquest problema (gateways). Cap de les opcions provades aporta una solució i es decideix contactar el servei tècnic de Wonderware. Després d'explicar el meu problema em comenten que es possible que aquest vingui per la configuració que per defecte utilitza Windows en les comunicacions COM/DCOM. Es revisa aquesta configuració, bàsicament de seguretat i, efectivament el problema era una configuració errònia en el protocol COM/DCOM de Windows.

Per evitar que passi el mateix a futurs usuaris es deixarà a disposició seva el document de configuració que van fer arribar el personal de servei tècnic de Wonderware.

6.2. Estructura i Interfície final SCADA

Finalment l'estructura del sistema és:

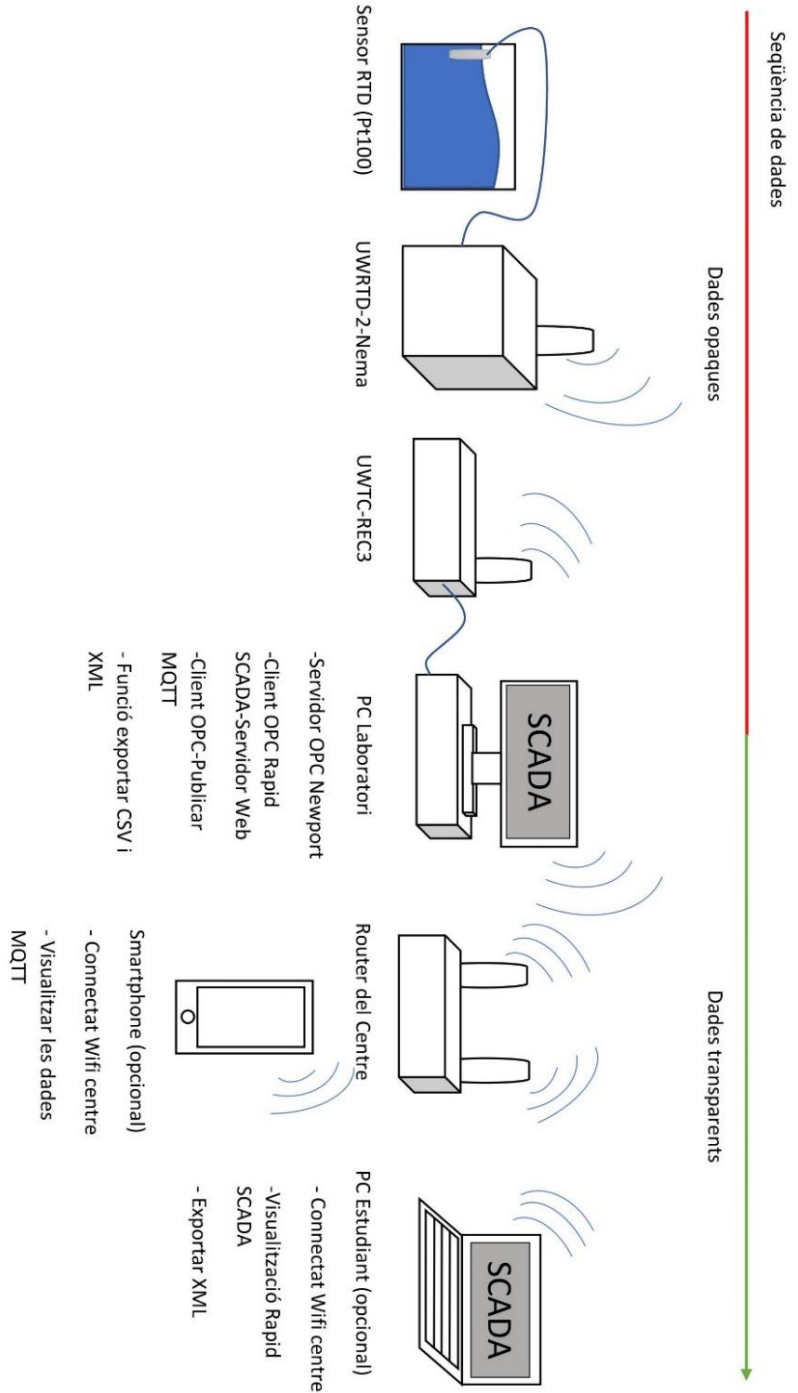


Figura 6.2.1. Estructura Sistema

La connexió entre el PC del centre i el router pot ser sense cables o amb un cable Ethernet. En aquest cas, finalment el PC del que es disposa no té targeta Wifi i la connexió es fa per cable, però això no suposa cap problema, al contrari, més velocitat. Seguidament s'adjunten imatges de l'aspecte del sistema final i de les dades exportades en XML.



Figura 6.2.2. Interfície SCADA

	A	B	C	D	E
1	27/05/2017 - 28/05/2017				
2	Generated: 28/05/2017 20:50:08				
3	Input channels:				
4	2001	OPC Newport - 147_83_208_109.UWTC02.Temperature1			
5	2002	OPC Newport - 147_83_208_109.UWTC02.Temperature2			
6	2003	OPC Newport - 147_83_208_109.UWTC02.Active			
7	2004	OPC Newport - 147_83_208_109.UWTC02.Battery			
8					
9	Time	Channel 2001	Channel 2002	Channel 2003	Channel 2004
713	20:24:00	26,1	24	1	3170
714	20:24:30	26,1	24,2	1	3170
715	20:25:00	26,1	24,2	1	3170
716	20:25:30	26,1	24,5	1	3170
717	20:26:00	26,1	24,5	1	3170
718	20:26:30	26,1	24,5	1	3170
719	20:27:00	26,1	24,5	1	3170
720	20:27:30	26,1	24,5	1	3170
721	20:28:00	26,1	24,5	1	3170
722	20:28:30	26,1	24,5	1	3170
723	20:29:00	26,1	24,5	1	3170
724	20:29:30	25,5	24,5	1	3170
725	20:30:00	26,1	24,5	1	3170
726	20:30:30	26,1	24,5	1	3170
727	20:31:00	26,1	24,5	1	3170
728	20:31:30	26,1	24,5	1	3170
729	20:32:00	26,1	24,5	1	3170
730	20:32:30	26,1	24,5	1	3170
731	20:33:00	26,1	24,7	1	3170
732	20:33:30	26,1	24,7	1	3170
733	20:34:00	26,1	24,7	1	3170

Minute data (+)


Listo 

Figura 6.2.3. Registre .xml

En un inici la intenció era utilitzar, com ja s'ha dit l'SCADA Intouch. En cas d'utilitzar aquest tipus de sistema la publicació de les dades a MQTT passava a ser un pur tràmit, doncs la OI Gateway inclou aquesta funcionalitat i implementar-la és qüestió de fer quatre clics. Mirant els manuals dels que es disposa s'utilitza la configuració descrita i el sistema no funciona. Després de revisar el sistema varies vegades i fer una recerca de possibles errors de configuració es decideix parlar amb el servei tècnic d'Intouch. Es manté una conversa via correu electrònic on s'envien les captures de pantalla de la configuració i el registre de la Gateway. Com que en un inici no saben veure on hi ha l'error es decideix quedar un dia per fer una connexió remota per revisar in situ la configuració.

Es revisa la informació i segons el servei tècnic la configuració utilitzada és la correcta. De fet el mateix tècnic va fer una prova amb la mateixa configuració i el sistema funcionava. Com que el cas no sembla gens normal el tècnic informa que buscarà quin pot ser l'error i es posarà en contacte amb mi. Al cap de dues setmanes es rep resposta on el tècnic informa de que sembla que hi ha una anomalia al software i que aquest no funciona amb alguns PCs. Es demana fer una nova connexió remota. Es planifica una connexió remota i al dia de la connexió el tècnic de Wonderware i un dels informàtics desenvolupadors del software intenten configurar un PC de Wonderware que no pot publicar a MQTT. Després de més d'un intent i de no aconseguir que el PC publiqui a MQTT el desenvolupador informa que obrirà una investigació per veure que és el que passa i quin és el factor que impedeix la comunicació en certs equips.

Arribats a aquest punt s'ha de buscar una alternativa. Es fa una recerca buscant algun projecte de conversió d'OPC a MQTT i no es troba cap software dedicat únicament a aquest tipus de conversió . No obstant en la recerca es descobreix un portal molt interessant i desconegut fins al moment: GitHub. Possiblement el seu símbol (el simpàtic octocat) us sigui familiar ja que apareix a tots els webs que comparteixen codi amb aquest portal.



Figura 7.2 Logotip GitHub (Font: Github.com)

GitHub és un portal on programadors de tot el món penen els seus projectes sota llicències obertes. Té com a objectiu donar un accés fàcil a codi en pràcticament tots els llenguatges de programació existent perquè persones i grups tinguin exemples i puguin aprendre programació, per aplicar al seu

àmbit i contribuir a un món més format i amb accés fàcil al coneixement. De fet no es descarta penjar el codi desenvolupat en aquest projecte al portal.

Al buscar en aquest portal hi ha un codi python que publica a mqtt dades obtingudes d'un servidor OPC. Es revisa el codi però no se n'arriba a entendre el funcionament. A més a més, python és un llenguatge desconegut.

D'assignatures anteriors del grau en electrònica industrial i automàtica es coneix el llenguatge C#. Es busca algun tipus de codi de suport per començar a treballar, però no es troba res i, per realitzar aquest tipus d'aplicació es creu que no és el més apropiat, doncs està orientat a una interfície gràfica que no té perquè existir.

Es creu que el llenguatge que pot donar millor marc a l'aplicació és Python. Es tracta d'un llenguatge simple, fàcil d'aprendre, però a la vegada potent i amb exemples de codi existent i llibreries per implementar un client OPC i publicar a MQTT.

Es segueix una sèrie de vídeos tutorials del canal de youtube "THENEWBOSTON" i s'aprenen les bases del llenguatge. Una vegada se saben les bases del software es comença per fer un client OPC. Per fer aquest client s'utiliza la llibreria OpenOPC i exemples de GitHub. Després d'aconseguir llegir les dades del servidor OPC es crea un codi independent que publica una dada fixa a MQTT. Aquest codi s'aconsegueix amb la llibreria de paho, del projecte eclipse, que inclou exemples. Ara es tenen dos codis: un client OPC i un que publica a MQTT. S'uneixen i es crea un codi OPC a MQTT.

Ara les dades són accessibles per qualsevol dispositiu amb connexió a internet i un software de client MQTT, per exemple un smartphone. Seguidament es pensa en l'opció de, ja que es disposa de les dades en un codi python, perquè no guardar-les a un fitxer .csv (comma-separated value) i crear així una base de dades simple, però no obstant accessible per multitud d'aplicacions. Així doncs es busca com crear un .csv i s'implementa aquesta funcionalitat.

7.1. Funcionament

Amb la sonda i el receptor funcionant es fa doble clic sobre l'aplicació amb l'script de Python.



Figura 7.1.1. Icona de l'Script

Aquest començarà a publicar les dades que rep del receptor a MQTT. En aquest cas al broker MQTT que s'ha instal·lat a l'ordinador. Un broker MQTT creat per Eclipse, de la comunitat open source. En el cas de la prova la configuració només funciona amb dispositius connectats a la mateixa xarxa, ja que el Broker té una IP privada. En el cas que la IP fos pública el broker seria accessible a tot internet. Sabrem que el programa s'està executant perquè apareix una finestra com la de la figura.

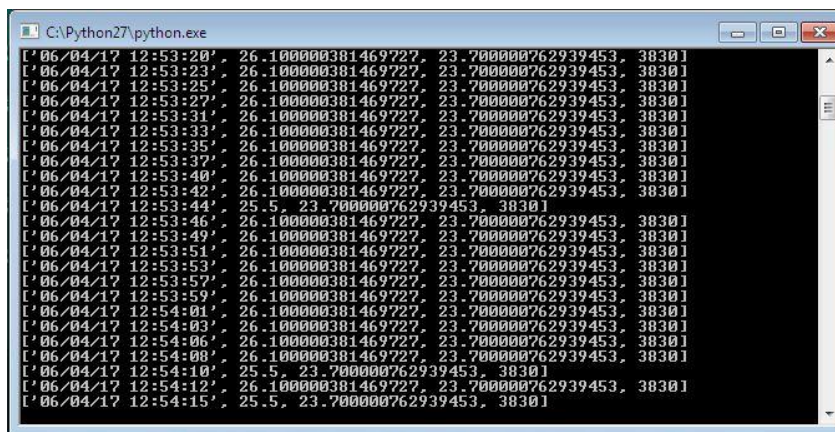


Figura 7.1.2. Cmd Python

Seguidament, si anem al nostre mòbil i, després d'instal·lar una aplicació subscriptor MQTT la configurem veure les dades que s'estan publicant.

En aquest projecte s'escull configurar l'aplicació MQTT Dash per a Android.

Primer de tot cal definir el Broker al qual l'aplicació es connectarà. Aquí se li dona un nom i s'introdueix la direcció, que és la IP de l'ordinador. El port s'utilitza l'1883, que es el port no segur, el que ofereix una configuració més fàcil i ràpida.

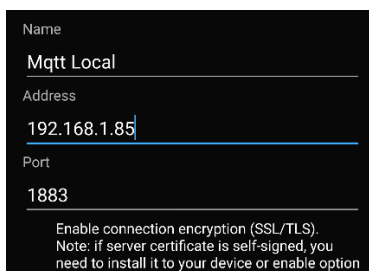


Figura 7.1.3. MQTT Broker

Seguidament es configura la subscripció als tòpics. Els tòpics són Lab/Time, Lab/Temp1, Lab/Temp2 i Lab/Bat.

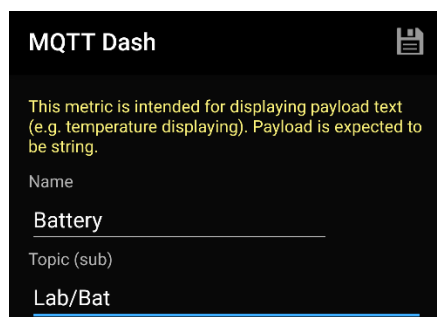


Figura 7.1.4. Subscripció als Topics

Finalment la configuració queda com es mostra a la figura 7.1.5. En aquest cas les dades s'actualitzen cada 2 segons.

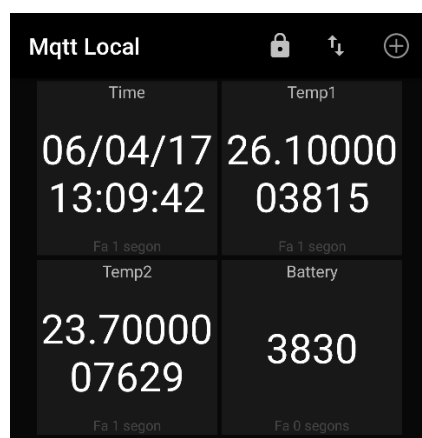


Figura 7.1.5. Aspecte de l'escriptori

A més a més, l'aplicació va guardant les dades a un fitxer .csv. Aquest fitxer es crea a la mateixa carpeta on hi ha l'script que estem executant. Aquest fitxer es pot importar a un full de càlcul com Excel i fer els tractaments que es creguin necessaris. Abans de fer cap càlcul el millor es convertir tots els punts del CSV a comes perquè siguin interpretades correctament.

	A	B	C	D
1	Time	Temperature1	Temperature2	Battery
2	06/04/17 13:11:54	25.5	23.7000007629	3830
3	06/04/17 13:11:57	25.5	23.7000007629	3830
4	06/04/17 13:11:59	25.5	23.7000007629	3830
5	06/04/17 13:12:01	25.5	23.7000007629	3830
6	06/04/17 13:12:03	25.5	23.7000007629	3830
7	06/04/17 13:12:06	25.5	23.7000007629	3830
8	06/04/17 13:12:07	25.5	23.7000007629	3830
9	06/04/17 13:12:09	25.5	23.7000007629	3830
10	06/04/17 13:12:11	25.5	23.7000007629	3830
11	06/04/17 13:12:13	25.5	23.7000007629	3830
12	06/04/17 13:12:16	25.5	23.7000007629	3830
13	06/04/17 13:12:18	25.5	23.7000007629	3830
14	06/04/17 13:12:20	25.5	23.7000007629	3830
15	06/04/17 13:12:22	26.1000003815	23.7000007629	3830
16	06/04/17 13:12:24	26.1000003815	23.7000007629	3830
17	06/04/17 13:12:26	26.1000003815	23.7000007629	3830
18	06/04/17 13:12:28	26.1000003815	23.7000007629	3830
19	06/04/17 13:12:30	26.1000003815	23.7000007629	3830
20	06/04/17 13:12:33	26.1000003815	23.7000007629	3830
21	06/04/17 13:12:35	26.1000003815	23.7000007629	3830
22	06/04/17 13:12:37	26.1000003815	23.7000007629	3830
23	06/04/17 13:12:39	26.1000003815	23.7000007629	3830
24	06/04/17 13:12:41	25.5	23.7000007629	3830
25	06/04/17 13:12:43	25.5	23.7000007629	3830
26	06/04/17 13:12:45	25.5	23.7000007629	3830
27	06/04/17 13:12:47	25.5	23.7000007629	3830
28	06/04/17 13:12:50	25.5	23.7000007629	3830
29	06/04/17 13:12:52	26.1000003815	23.7000007629	3830
30	06/04/17 13:12:54	26.1000003815	23.7000007629	3830
31	06/04/17 13:12:56	25.5	23.7000007629	3830
32	06/04/17 13:12:59	25.5	23.7000007629	3830
33	06/04/17 13:13:00	25.5	23.7000007629	3830

Figura 7.1.6. Exemple .csv a Excel

8. Anàlisi de l'impacte ambiental

El present treball no té cap impacte negatiu per al medi ambient. És important tenir uns bons costums d'apagar l'ordinador i les sondes sempre que s'acabin les sessions de pràctiques per evitar el malbaratament d'energia. En aquest sentit també s'ha d'ajustar el màxim el nombre de mostres per minut que es desitja que el sensor adquireixi, doncs un nombre de mostres elevat incrementa el consum de la pila i aquesta s'haurà de substituir amb més freqüència. Es tracta d'un sensor que funciona amb una pila ja que és un sensor sense fils i quan sigui necessari substituir la pila aquesta s'ha de dipositar al contenidor adequat.

9. Planificació

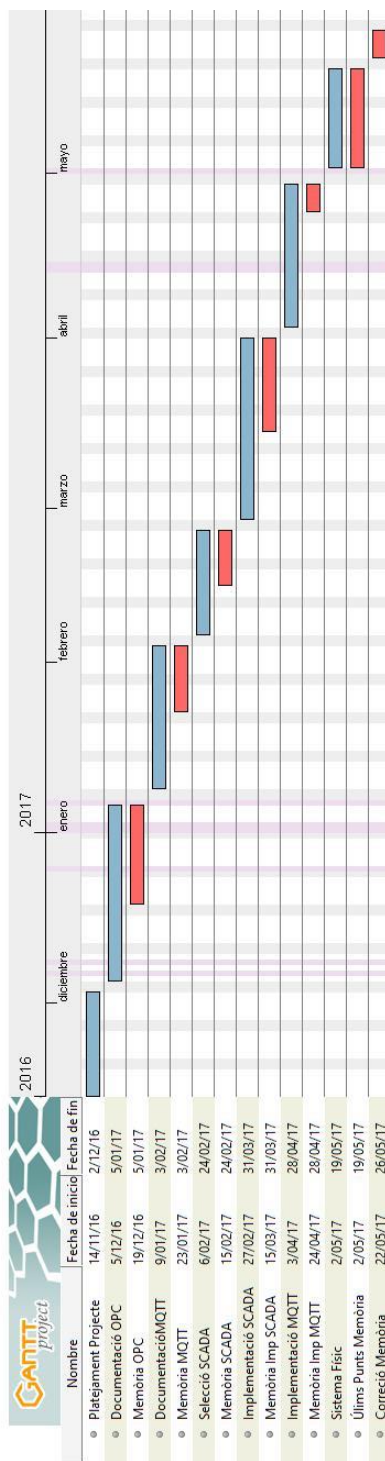


Figura 9.1 Planificació Gantt

Sempre abans de començar un projecte s'ha de fer una planificació del mateix amb les tasques i el temps que s'assigna a cada una d'elles. Una bona opció és fer un diagrama de Gantt, que és una forma molt visual de veure les tasques. En aquest projecte no es va fer una planificació estricta i això va ser un error. Es va fer una planificació, però sense un calendari marcat. A part, es van tenir molts problemes i imprevistos que van endarrerir les tasques fins al punt de que l'autor es planteja aplaçar el termini d'entrega. Finalment no va ser necessari, i amb un esforç final, treballant els caps de setmana, es va aconseguir arribar amb el projecte acabat a la data de presentació, però sense tenir temps de fer una revisió exhaustiva del treball i donar temps al tutor per a que se'l revisés. No és el marc ideal, però sovint surten imprevistos, que s'han de preveure al Gantt, però que poden fer que s'allargui el projecte. Per fer el diagrama de Gantt s'ha utilitzat el programa Gantt Project. Aquest programa va molt bé perquè simplifica molt la feina de fer el projecte i dona opcions a destinar recursos (persones o material) a determinades feines. A més a més, el software es sense llicència així que es pot descarregar gratuïtament i de forma fàcil.

Conclusions

En el marc acadèmic es creu que s'han assolit els objectius del projecte. S'ha aconseguit crear un sistema amb un registre de dades de temperatura en uns formats estàndards i accessibles. A més a més, s'ha aconseguit crear un entorn SCADA encara que molt reduït, amb semblances a un industrial, que pot ser un primer contacte pels estudiants. Aquest era un dels motius principals del projecte, ja que un sistema com aquest pot ser molt més econòmic d'implementar si s'utilitzen sondes USB connectades directament a un ordinador.

On no s'han pogut complir les demandes del co-director és en el punt que es volia un sistema on l'estudiant hagués de fer petites configuracions per introduir-lo al món de les comunicacions industrials.

A més a més s'ha aconseguit un sistema creat a partir de software lliure i software creat per al projectista en la mesura del possible. Això ha fet un sistema sense límits de versions demostració o llicències a renovar. Ara queda que el temps el posi a prova i es vegi si els resultats són bons i el sistema és robust, que és un objectiu que també es perseguia. S'ha intentat deixar el màxim de documentació possible per a facilitar el manteniment, modificació o ampliació del sistema.

S'ha fallat en la programació. Punt molt important en tot projecte. Aquesta no va ser bona i això ha fet complicat arribar a la data de presentació.

En el tema personal, com que aquest és (sempre que s'aprovi) l'últim treball del grau, m'agradaria afegir comentaris, opinions i percepcions totalment personal que he tingut al llarg del treball.

Aquest treball m'ha ajudat a obrir i explorar una porta que desconeixia la complexitat que arriba a presentar. Aquest és el món de les comunicacions industrials. Al llarg del treball he tingut molts temps morts, en els quals no avançava ni una mil·lèsima en una setmana i que et porten a la desesperació, fins que decideixes demanar ajuda i en això també m'ha servit. En veure que quan portes molt de temps encallat en un punt has de demanar ajuda a professors i servei tècnic. Encara que ells no et resolguin el problema (o sí), però et donen una nova visió del problema que potser no tenies i et permet resoldre'l. O et proporcionen informació que desconeixies o no sabies on buscar, ja que en temes avançats sovint no saps on trobar informació.

És possible que aquest treball m'ajudi també a l'hora de decidir el meu futur. A part, també m'ha donat idees de projectes personals que vull portar a terme. M'ha descobert el món del programari lliure per SCADAs i el portal GitHub, que considero molt útils en el món de l'ensenyament i dels projectes personals, per la seva versatilitat, possibilitat de modificació i, també, pel seu baix cost.

Pressupost i Anàlisi Econòmica

Per a l'execució d'aquest projecte s'ha utilitzat material ja adquirit pel departament de química. En aquest cas el cost del projecte ha estat pràcticament nul. No obstant es farà el pressupost com si el sistema s'hagués d'adquirir completament nou.

Concepte	Preu unitari	Quantitat	Preu total
Sensor UWRTD-2-Nema	196,89 €/unit	1	196,89
Receptor UWTC-REC3	268,52 €/unit	1	268,52
PC Windows + Monitor + Teclat + Ratolí estàndard	275€/unitat	1	275
OPC-Server Llicència	283,07 €/unit	1	283,07
Software Rapid SCADA	0€/unit	1	0
Plug-in Chart PRO	4,47 €/unit	1	4,47
Software Python 2.7	0 €/unit	1	0
Hores Enginyeria	70 €/hora	50	3500
Base Imposable			4527,95
IVA		21%	950,86
Total			5478,81

S'ha estimat l'hora de treball d'una empresa d'enginyeria en 70€/hora i que un enginyer amb experiència en el sector amb una mica més d'una setmana laboral normal tindria el projecte. En aquestes hores es divideix el temps entre anàlisis, disseny i implementació. Que dividirien el temps a parts iguals. Òbviament jo he necessitat tot un quadrimestre i l'ajuda de molta gent. També cal tenir present que en el cas de contractar una empresa externa s'hauria hagut d'adquirir una llicència de SCADA i la part IoT segurament s'hauria obviat.

Amb les dades a la mà no és un projecte viable, doncs un pressupost tant elevat per un procés que es basa en la lectura de només un sensor (ni que en fossin 4) no està justificat. No obstant, donat que es disposa dels dispositius i el sistema l'ha fet un estudiant es creu que es una opció viable. A part dota el laboratori de química d'una interfícies (els SCADAs) molt utilitzada a la indústria i dona les dades en dos formats diferents (.xml i .csv) , també molt utilitzats i amb els que segurament hauran de treballar en un futur.

Bibliografia

- [1] OPC Foundation, *What is OPC?* (2017). Accessible a, <https://opcfoundation.org/about/what-is-opc/>
- [2] MQTT.org, *MQTT Portal*. Accessible a, <http://mqtt.org/>
- [3] Wikipedia, *Client-Server model* (2017). Accessible a, https://en.wikipedia.org/wiki/Client%E2%80%93server_model
- [4] Wikipedia, *Publish-subscribe pattern* (2017). Accessible a, https://en.wikipedia.org/wiki/Publish%E2%80%93subscribe_pattern
- [5] Zigbee Alliance, *Portal Zigbee Alliance* (2017). Accessible a, <http://www.zigbee.org/>
- [6] Digi International, *Zigbee Wireless Standard*. Accessible a, <https://www.digi.com/resources/standards-and-technologies/rfmodems/zigbee-wireless-standard>
- [7] Newport, *UWRD-2-NEMA*. Accessible a, <http://www.newportus.com/ppt/UWRD-2-NEMA.html>
- [8] Newport, *UWTC-REC3*. Accessible a, <http://www.newportus.com/ppt/UWTC-REC3.html>
- [9] The OPC Foundation, *What is OPC? UA in a minute* (2015). Accessible a, <https://youtu.be/-tDGzwsBokY>
- [10] Wonderware, *¿Qué es SCADA?* (2017). Accessible a, <http://www.wonderware.es/hmi-scada/que-es-scada/>
- [11] Electronicsforu, *8 Free and Open Source Software For SCADA* (2016). Accessible a, <http://electronicsforu.com/resources/cool-stuff-misc/8-free-open-source-software-scada>
- [12] Encada, *Indigo SCADA*. Accessible a, <http://www.enscada.com/a7khg9/IndigoSCADA.html>
- [13] Eclipse, *neoSCADA* (2017). Accessible a, https://eclipse.org/eclipsescada/news/2017/02/15/eclipse_neoscada_0_4_0_released.html
- [14] MICROSYS, *Promotic* (2017). Accessible a, <https://www.promotic.eu/en/index.htm>

- [15] Rapid SCADA, *Rapid SCADA Portal* (2017). Accessible a, <http://rapidscada.org/>
- [16] IBM Think Academy, *How it Works: Internet of Things* (2015). Accessible a, <https://www.youtube.com/watch?v=QSIPNhOiMoE>
- [17] GitHub, *GitHub Portal* (2017). Accessible a, <https://github.com/>
- [18] Pierre F., *paho.mqtt.python* (2017). Accessible a, <https://github.com/eclipse/paho.mqtt.python>
- [19] OpenOPC, *OpenOPC for Python*. Accessible a, <http://openopc.sourceforge.net/>
- [20] Wikipedia, *Modelo OSI* (2017). Accessible a, https://es.wikipedia.org/wiki/Modelo_OSI
- [21] Universitat Politècnica de València - UPV, *Modelo de referència OSI @UPV* (2016). Accessible a, <https://www.youtube.com/watch?v=vfcpqnWYl1E>
- [22] Universitat Politècnica de València- UPV, *El modelo OSI* (2011). Accessible a, https://www.youtube.com/watch?v=iN_kCtk22Ek
- [23] Lara, Eduard, *Modelo OSI*. Accessible a, <http://elara.site.ac.upc.edu/documentacion/INTERNET%20-%20UD1%20-%20Modelo%20OSI.pdf>
- [24] Eclipse, *Mosquitto Broker* (2017). Accessible a, <https://mosquitto.org/>
- [25] thenewboston, *Python 3.4 Programming Tutorials* (2016). Accessible a, <https://www.youtube.com/playlist?list=PL6gx4CwI9DGAcbMi1sH6oAMk4JHw91mC>

Annex

S'adjunta el codi Python

```
#####
#####
##### OPC Client//MQTT Publish//CSV Recording #####
#####
#####

import time
import csv
import OpenOPC
import sys
import os
import inspect
import paho.mqtt.publish as publish

def importOPC(OPCserver,taglist):
    message=[]
    try:
        opc.connect(OPCserver1)          #Connection with the OPC Server
    except:
        print('Unable to connect with the OPC Server')

    try:
        v = opc.read(tags=taglist)        #Reading tags from OPC
        str(v)
        for i in range(len(v)):
            (name,val,qual,time)=v[i]
            if val == '':
                val = 0
            message.insert(i,val)
            message.insert(0,time)
            print(message)
    except:
        print('Unable to get the values')
    return message

def AddItemsCSV(OPCserver,CSVname,taglist):
    """
    Connect with OPC server
    Extract the value of the taglist
    and append it to the csv
    taglist: list with name of all items
    """
```

```

attach = [] #Initialize attach

try:
    opc = OpenOPC.client() #Connection with the OPC Server
    opc.connect(OPCserver)
except:
    print("No connection to Server")

try:
    v = opc.read(tags=taglist) #Reading tags from OPC
    str(v)
except:
    print("Unable to read tags")

try:

    # Create a line with the values from the tag

    for i in range(len(v)):
        (name, val, qual, time) = v[i]
        if val == '':
            val = 0
        attach.insert(i,val)
    attach.insert(0,time)
except:
    print('Unable to create a row')

#Writing the line to the CSV file

try:
    b = open(CSVname,'ab')
    a = csv.writer(b)
    a.writerow(attach)
    b.close()
except:
    print('Unable to write in CSV')

#Initialize values

OPCserver1='NewportOPC.iSeries'
CSVname1='Newport.csv'
taglist1=['147_83_208_109.UWTC02.Temperature1','147_83_208_109.UWTC02.Temperature2']
variables=['Time','Temperature1','Temperature2','Battery']
publishlist=['Lab/Time','Lab/Temp1','Lab/Temp2','Lab/Bat']
UpdateRate=30

opc = OpenOPC.client()

```

```
#Creating the first row with the names of the variables

try:
    b = open(CSVname1,'ab')
    a = csv.writer(b)
    a.writerow(variables)
    b.close()
except:
    print('Unable to initialize the CSV file')

#Main program

while True:
    #CSV Function
    AddItemsCSV(OPCserver1,CSVname=CSVname1,taglist=taglist1)

    #Getting information from the OPC server

    message = importOPC(OPCserver1,taglist1)

    #MQTT Publish

    for i in range(len(message)):
        top=publishlist[i]
        value=message[i]
        publish.single(topic=top, payload=value, hostname="localhost")

    #Delay UpdateRate

    time.sleep(UpdateRate)
```