



**Escola Politècnica Superior
de Castelldefels**

UNIVERSITAT POLITÈCNICA DE CATALUNYA

TREBALL DE FI DE CARRERA

TÍTOL DEL TFC: Migració Scada en plataforma Labview i Comunicacions estàndard Modbus

TITULACIÓ: Enginyeria Tècnica de Telecomunicació, especialitat Telemàtica

AUTOR: Alejandro Borrego Gómez

SUPERVISOR: Rubén Quesada

DIRECTOR: Ángel Jaime

DATA: 24 de maig de 2011

Títol: Migració Scada en plataforma Labview i Comunicacions estàndard Modbus

Autor: Alejandro Borrego Gómez

Supervisor: Rubén Quesada

Director: Angel Jaime

Data: 24 de maig de 2011

Resum

Actualment, qualsevol planta industrial disposa de processos automatitzats visualitzats en tot moment per sistemes SCADA. En aquest cas, el projecte consistirà en migrar l'actual sistema a una aplicació SCADA d'última generació, tot això en una instal·lació totalment obsoleta. L'avaluació dels equips existents, marcarà la solució final que haurà d'aportar l'enginyeria encarregada dels canvis contractats.

Les comunicacions entre màquines es un altre dels punts forts del món de l'automatització. Els diferents elements situats en una instal·lació normalment es troben separats per distàncies importants. Unes comunicacions eficients constitueixen la robustesa dels processos. L'estàndard utilitzat serà el protocol obert de comunicacions MODBUS, molt estès actualment al mercat capaç de suportar les dades amb que treballa el PLC.

Un dispositiu intermedi realitzarà les tasques d'intercanvi de dades entre PLC i SCADA. Aquest haurà de conviure amb un PLC totalment obsolet i una plataforma SCADA d'última generació.

Per últim arriba l'objectiu principal, el desenvolupament del sistema SCADA sobre la plataforma Labview 8.2. La migració entre el predecessor i la nova plataforma, ha de ser pràcticament transparent de cara al client final instal·lació, que no ha de patir cap aturada durant el procés d'adaptació.

Un compromís qualitat-preu, farà que l'enginy sigui l'eina essencial per tal de poder desenvolupar la nova aplicació amb garanties d'èxit.

Title: SCADA migration based on Labview and MODBUS communications

Author: Alejandro Borrego Gómez

Supervisor: Rubén Quesada

Director: Ángel Jaime

Date: 24th, may 2011

Overview

Nowadays, any industrial plant has automated processes, visualized through SCADA systems. In that case, the project will consist in migrating the current system to an application SCADA of last generation, all this facts installed with obsolete devices. The evaluation of the existent devices will mark the final solution that will have to contribute the engineering commissioned of the changes hired.

The communications between machines are another of the strong points in the world of automation. The different elements in an installation are usually separated by important distances. The optimized communications constitute the robustness of the processes. The standard used will be the open source communications protocol MODBUS, actually very extended to the able market to bear the data with that works the PLC.

An intermediate device will translate the data between PLC and SCADA. This will have to coexist with a totally obsolete PLC and a SCADA platform based on a new development software.

We will initiate the development of the SCADA system with Labview 8.2. The migration between the predecessor and the new platform has to be practically transparent for the final customer and the plant that does not have to suffer any stop during adaptation process.

A commitment quality-price, will define the wit as essential tool in order to develop the new application with guarantees of success.

ÍNDIX

INTRODUCCIÓ	1
CAPÍTOL 1. COMUNICACIONS MODBUS	2
1.1. Visió General del protocol MODBUS	2
1.2. Característiques físiques MODBUS	4
1.3. Característiques lògiques MODBUS.....	5
1.3.1. Accés al medi (mestre-esclau)	5
1.3.2. Estructura de les trames	8
1.3.3. Tipus de dades.....	9
1.3.4. Diàleg esclau-mestre MODBUS.....	10
1.3.5. Model de dades MODBUS	11
1.3.6. Transmissió de dades en sèrie	12
1.3.6.1. Mode de transmissió ASCII.....	12
1.3.6.2. Mode de transmissió MODBUS RTU	12
CAPÍTOL 2. ESCENARI EN INDÚSTRIA	14
2.1. Escenari hardware.....	14
2.1.1. Especificacions generals del Hardware	14
2.1.2. Interface LANCE-E.....	15
2.1.2.1. Característiques tècniques.....	16
2.1.2.2. Mode de connexió	17
2.1.2.3. Comunicació interface mitjançant MODBUS.....	18
2.1.2.4. Configuració dispositiu interface Lance-E	20
2.2. Escenari software	21
2.2.1. Què es Labview?.....	21
2.2.2. Programació gràfica Labview	22
2.3. Definició de projecte	23
2.3.1. Canvis hardware en la instal·lació.....	23
2.3.1.1. Recursos actuals com a objecte d'estudi	24
2.3.1.2. Proposta final de l'enginyeria elèctrica	25
CAPÍTOL 3. DESENVOLUPAMENT SOFTWARE	28
3.1. Test de comunicacions MODBUS	28
3.1.1. Definició de senyals en codi PLC.....	28
3.1.2. Definició de senyals en Interface Lance-E	29
3.1.3. Definició de senyals en Labview	29
3.2. Desenvolupament de software en Labview	30
3.2.1. Definició de l'estructura principal.....	30
3.2.2. Descriptiu de subprogrames Vis creats.....	32
3.2.2.1. Pantalles de control.....	32
3.2.3. Gestió de les comunicacions.....	35
3.2.3.1. Inicialització de dades amb PLC PS316	35
3.2.3.2. Actualització de dades en PLC PS316.....	36
3.2.4. Tractament i captura de dades.....	37
3.2.4.1. Creació de fitxer d'emmagatzematge de dades.....	37
3.2.4.2. Tractament de dades	38

3.2.4.3. Conversió de dades	39
3.2.5. Representació de les dades capturades	41
3.2.5.1. Captura i visualització de temperatures	41
3.2.5.2. Creació d'informes automatitzats	42
3.2.6. Validació del software desenvolupat	43
CAPÍTOL 4. IMPACTE MEDIAMBIENTAL	44
4.1. Estudi del impacte mediambiental.....	44
4.1.1. Estalvi energètic a nivell software	44
4.1.2. Aportació a la conservació del medi ambient.....	45
CAPÍTOL 5. LÍNIES FUTURES I CONCLUSIONS	47
5.1. Línies futures	47
5.2. Conclusions	48
BIBLIOGRAFIA	50

INTRODUCCIÓ

L'automatització es una solució implementada avui dia per al tractament i control dels diferents processos existents en una planta industrial. En aquest cas, ens centrarem en el marc d'una indústria càrnica on un equip PLC s'encarregarà de mantenir la temperatura idònia en les cambres frigorífiques que conservaran els diferents aliments.

De la mà de l'autòmat tindrem el sistema SCADA, els veritables ulls de la instal·lació, capaç de visualitzar les diferents variables que conté el PLC i de forçar diferents paràmetres. En aquest cas, la nostra missió serà la de proposar un nou sistema SCADA, una migració de l'antiga aplicació sobre una plataforma d'última generació. Tot això, s'haurà de desenvolupar amb la coexistència del nou equip conjuntament amb una instal·lació avui dia obsoleta. Per tal de proposar al client final el seu projecte ideal en funció de la premissa qualitat/preu, es mirarà de trobar solucions que no comportin un augment especial del pressupost final.

En el primer capítol es definirà un dels trets importants, el d'establir un protocol de comunicacions estàndard per tal de fer comunicar el PLC amb el futur sistema SCADA. Es desenvoluparà el software entorn al protocol de comunicacions de categoria software lliure MODBUS. Aquest, avui dia es implementat en multitud de dispositius en el mercat.

A continuació en el segon capítol, aprofitarem per destacar els diferents elements hardware instal·lats en la planta càrnica. Arribats a aquest punt, s'avaluarà la recuperació d'alguns dels dispositius obsolets i es farà la proposta de projecte al client.

Un cop decidits els equips en els que s'haurà de realitzar la inversió inicial, en el tercer capítol es definirà el software que actuarà com la nova aplicació SCADA, desenvolupada sobre la plataforma Labview 8.2. Es destacarà en aquest apartat la gestió de les comunicacions MODBUS en tots els equips que hauran de compartir informació.

Per finalitzar en el capítol quart, es farà una estimació de l'impacte mediambiental actual. Amb l'objectiu de reduir les emissions nocives a l'atmosfera, així com la factura elèctrica, es destacaran algunes de les solucions adoptades i es realitzarà una previsió d'estalvi un cop s'hagin implementat els canvis proposats.

Finalment, l'enginyeria elèctrica un cop hagi finalitzat el projecte, marcarà una línia de futur on es presentaran diferents accions a realitzar a la planta per tal de millorar el rendiment dels processos realitzats dia a dia.

CAPÍTOL 1. COMUNICACIONS MODBUS

1.1. Visió General del protocol MODBUS

Modbus es un protocol de comunicacions que pertany al nivell 7 del model OSI i permet establir comunicacions client/servidor entre dispositius connectats a diferents tipus de busos i xarxes.

Serà a partir de 1979 quan Modicon converteix Modbus en un protocol de comunicacions estàndard al món de la indústria, que continua en la actualitat comunicant milions de dispositius electrònics. Rei en el món de l'automatització, es qualifica com un protocol superior a la resta pels següents trets distintius:

- És públic
- Fàcil implementació i requereix poc desenvolupament
- Suporta blocs de dades sense restriccions

Modbus també es utilitza per a la connexió entre un ordinador de supervisió amb una unitat remota (RTU), en sistemes d'adquisició de dades (SCADA). Existeixen versions del protocol Modbus per ser implementat sobre port sèrie i Ethernet (Modbus/TCP).

Aquest protocol pregunta/resposta ofereix serveis especificats mitjançant *function codes*, que seran elements integrats a les PDUs pregunta / resposta. A més, cal dir que està basat en la jerarquia mestre/esclau, establint una direcció única per a cada dispositiu de la xarxa Modbus. Qualsevol dispositiu té la capacitat d'enviar ordres, encara que el funcionament habitual contempla la existència d'un mestre. El fet de que contingui la direcció del destinatari no significa que la resta d'esclaus deixin de rebre la missatgeria, sinó que simplement serà el destinatari qui la executa (excepte en mode broadcast).

Una de les grans virtuts que caracteritza aquest protocol és la facilitat que ens aporta per realitzar qualsevol tipus de comunicació amb multitud d'arquitectures de xarxa. Tal i com es pot observar a continuació (Fig. 1.1), tots els tipus de dispositius (PLC, HMI, Panells de control – SCADA, Dispositius E/S) poden utilitzar Modbus per tal d'iniciar una operació remota. A més, ens permetrà adaptar les comunicacions tant a través de sèrie com d'Ethernet TCP/IP. Finalment, gràcies a l'ajuda de passarel·les podrem establir diàleg entre diferents tipus de busos i xarxes.

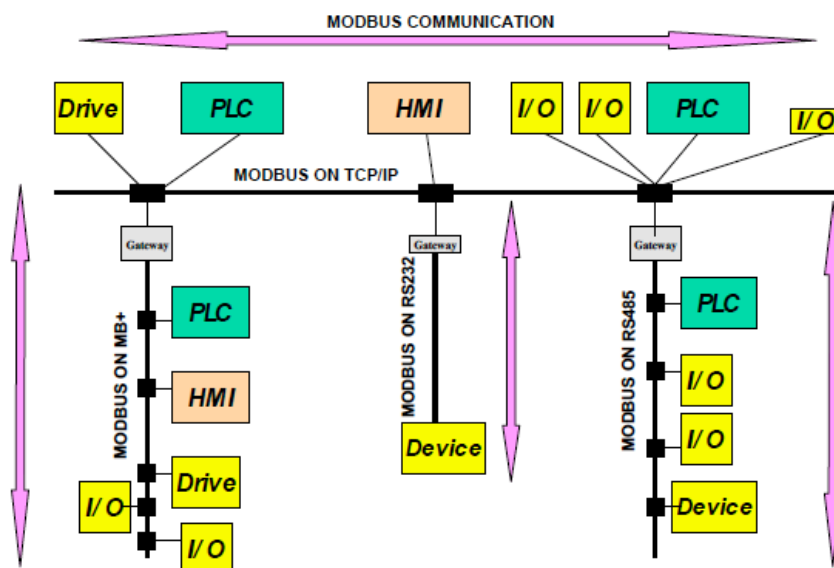


Fig. 1.1: Exemple arquitectura de xarxa MODBUS

MODBUS es un protocol de missatgeria situat a la capa d'aplicació (nivell 7) de la pila OSI. Actualment es troba implementat en els següents àmbits:

- TCP/IP a través d'Ethernet.
- Transmissions sèrie sobre una gran varietat de medis (EIA/TIA-232-E, EIA- 422, EIA/TIA-485-A; fibra, ràdio, etc.)
- MODBUS PLUS, un xarxa d'alta velocitat basada en pas de token.

Layer	ISO/OSI Model	
7	Application	MODBUS Application Protocol
6	Presentation	Empty
5	Session	Empty
4	Transport	Empty
3	Network	Empty
2	Data Link	MODBUS Serial Line Protocol
1	Physical	EIA/TIA-485 (or EIA/TIA-232)

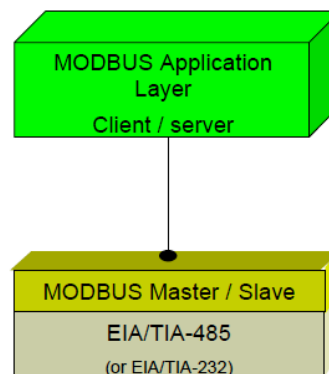


Fig. 1.2: Arquitectura de capes definida per Modbus.

Gràcies al nivell 7 que ens ofereix Modbus sobre el model OSI, aquest ens proveeix de comunicacions client/servidor entre dispositius connectats al bus o sobre xarxes.

Modbus també defineix un protocol que pertany a les capes 1 y 2 de la pila OSI, física i enllaç respectivament, per tal de permetre la comunicació sèrie entre un dispositiu mestre (master) i/o diversos esclaus (slaves). Aquest protocol es denomina: MODBUS Serial Line Protocol.

1.2. Característiques físiques MODBUS

Les comunicacions Modbus poden realitzar-se sobre enllaços punt a punt o multipunt, depèn de si el missatge s'envia de forma unicast o broadcast.

Com pot veure's en la figura 1.3, l'enllaç físic pot ser cablejat mitjançant RS-232, RS-422 o RS-485 o pot ser a través d'enllaços radials, típicament del tipus spread spectrum, que tenen l'avantatge de no requerir llicència per a la seva instal·lació. També mitjançant comunicacions telefòniques via MÒDEM, que fins i tot contemplen la possibilitat de realitzar un control a través de telefonia GSM.

Els enllaços poden ser de tipus half-duplex o full-duplex indistintament però, com es tracta d'un protocol tipus master/slave, un enllaç half-duplex serà suficient per a la seva implementació.

Els baud rates suportats depenen de les característiques dels equips utilitzats, encara que típicament van des de 1200 fins a 38400bps.

Les taxes de 9600bps i 19.2Kbps son requerides i 19.2 és el requerit "default". Altres índexs poden ser implementats opcionalment: 1200, 2400, 4800, 38,400Kbps, 56Kbps,115Kbps. Cada índex de baud implementat ha de suportar un 1% de marge d'error dins de situació de transmissió, i ha d'acceptar un error de 2% dins de la situació de recepció.

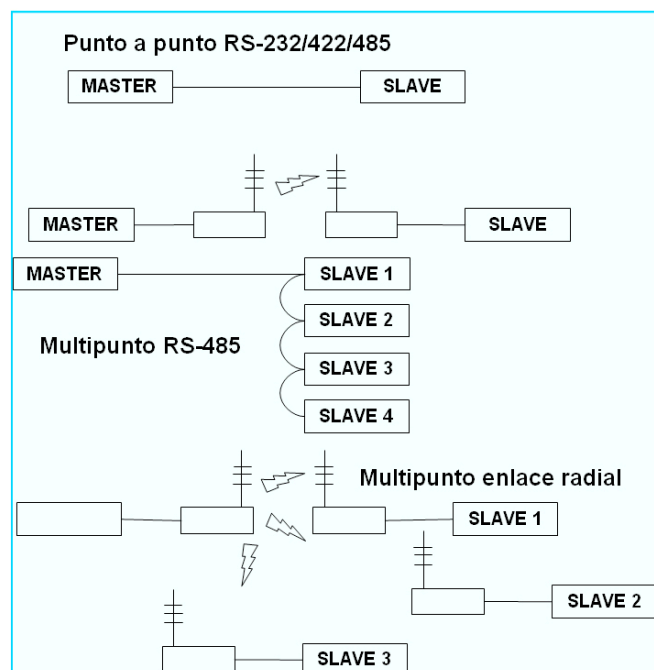


Fig. 1.3: Tipus d'enllaços Modbus.

En el cas de la figura 1.3, Modbus treballa sobre una interfície sèrie que pot variar entre una connexió RS232 o RS485. La interfície RS485 sol ser la més comuna, ja que RS232 pot penalitzar en distàncies llargues.

1.3. Característiques lògiques MODBUS

1.3.1. Accés al medi (mestre-esclau)

Com ja s'ha comentat, Modbus és un protocol mestre-esclau. Això significa que només un mestre pot tenir accés al medi en cada transmissió. A cada equip remot se li assigna un número de dispositiu en el rang de 1 a 255. El nombre màxim d'estacions és de 63 esclaus i un mestre.

A continuació (figura 1.3.1) s'observa la distribució de direccions Modbus segons estàndard:

Direccions	0	1 a 247	248 a 256
Assignada a	<i>Broadcast</i>	<i>Esclaus</i>	Reservades

Fig. 1.3.1: Assignació de direccions Modbus.

Una comunicació Modbus sempre la inicia un mestre, per la qual cosa un esclau mai transmet informació si no és a causa d'una petició del mestre i tampoc pot comunicar-se amb altres esclaus.

El dispositiu mestre pot transmetre dades en dues maneres diferents:

- Unicast:

En mode unicast el mestre es comunica amb un dispositiu esclau en concret i aquest sempre retorna un missatge de resposta a l'emissor. En una transmissió unicast, l'intercanvi d'informació Modbus sempre es compon de dos missatges: una petició del mestre i una resposta de l'esclau (Veure figura 1.3.2).

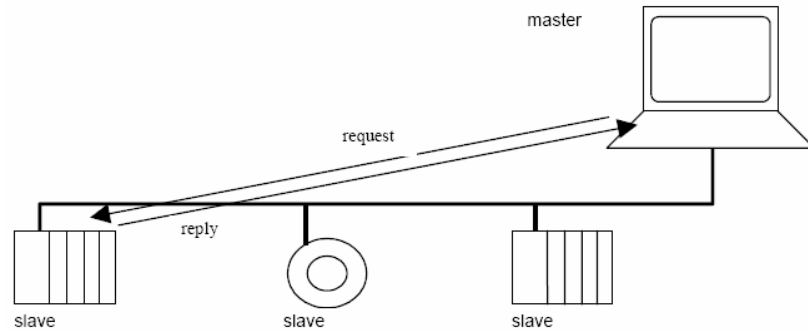


Fig. 1.3.2: Transmissió unicast.

- Broadcast:

El mestre comença una transacció que tindrà com a destí a tots els esclaus, aquests no responen a tal petició i el mestre dona per assumida la finalització de la transacció. En aquest mode no es pot assegurar que la informació arriba a tots els esclaus (Veure figura 1.3.3).

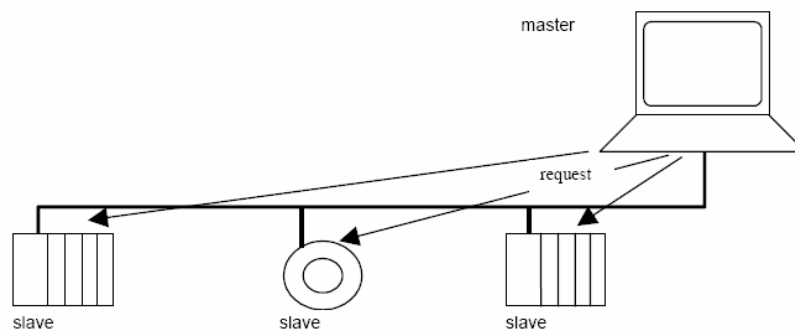


Fig. 1.3.3: Transmissió broadcast.

Plantejant l'esquema mestre - esclau s'observa que la relació entre ells és jeràrquica, el mestre té major jerarquia que els esclaus controlant i distribuint els temps des del punt de vista de les comunicacions. Seguidament es descriuen les funcions que tenen mestre - esclau en la comunicació Modbus.

- Màster Modbus

En el funcionament del protocol Modbus en una xarxa, el dispositiu Màster s'encarrega de realitzar les peticions als dispositius esclaus. Un dispositiu Màster pot ser un computador, un PLC o qualsevol mòdul capaç de portar a terme les funcions de Màster Modbus a través d'un port de comunicacions.

Els paràmetres bàsics a configurar en un dispositiu Màster són:

- Port sèrie a utilitzar
- Velocitat de comunicació
- Bit de paritat

- Bit de parada
- Mode RTU o ASCII
- Temps d'espera de resposta

Paràmetres avançats poden ser especificats depenent del dispositiu Màster Modbus utilitzat.

Un dispositiu Màster envia una trama de petició a un esclau, i espera un temps predeterminat per a rebre la resposta. Si no es rep una resposta una vegada transcorregut el temps d'espera, es genera un error de 'Temps d'Espera' i es tornarà a realitzar la petició després d'un temps determinat. Si es rep una resposta abans del temps d'espera, es porta a terme un anàlisi per a determinar la validesa de la resposta. La transmissió i la recepció de les trames ha de complir amb les regulacions del protocol Modbus.

- Esclau Modbus

Per portar a terme la comunicació Modbus en una xarxa, és necessària la presència de almenys un esclau Modbus connectat al Màster Modbus. Un dispositiu esclau pot ser un PLC, un variador de velocitat, un transmissor o qualsevol mòdul capaç de comportar-se com a esclau.

En un esclau Modbus s'han d'especificar paràmetres bàsics com:

- Direcció de l'Esclau
- Port Sèrie a utilitzar
- Velocitat de Comunicació
- Bit de paritat
- Bit de parada
- Mode RTU o ASCII

Un dispositiu esclau ha de mantenir-se en estat de recepció per poder admetre les trames enviades per un Màster Modbus. Una vegada que l'esclau Modbus rep la trama completa, l'analitza i determina si la petició va dirigida a ell. Si és així, processa la petició i envia la resposta al Màster Modbus.

En cas que la petició no vagi dirigida a ell, torna a l'estat de recepció. La transmissió i recepció de les trames han de complir amb les regulacions del protocol Modbus.

S'ha de tenir en consideració que en una xarxa Modbus ha de complir-se el següent:

- Presència d'un sol Màster Modbus
- Tots els dispositius connectats a la xarxa han de treballar a la mateixa velocitat de comunicació i similar configuració de bit de paritat i parada.
- Un sol mode de comunicació: RTU o ASCII
- Cada Esclau Modbus ha de tenir una direcció diferent

Cada dispositiu de la xarxa Modbus té una direcció única. Qualsevol dispositiu pot enviar ordres Modbus, encara que l'habitual és permet-ho a un dispositiu

mestre. Tots els dispositius reben la trama però només el destinatari l'executa. Cada un dels missatges conté informació redundat que assegura la seva integritat en la recepció. Alhora podrem controlar un dispositiu RTU per tal de modificar el valor d'algun dels seus registres o bé sol·licitar el contingut dels registres.

1.3.2. Estructura de les trames

En la figura 1.3.4 es pot observar com és la PDU (Protocol Data Unit) amb la qual treballa Modbus i que és totalment independent de les capes inferiors.

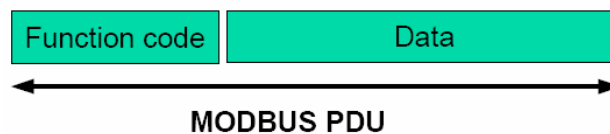


Fig. 1.3.4 PDU de Modbus

El mapeig del protocol Modbus en bus o xarxa específica acaba introduint camps addicionals creant una ADU (application data unit).

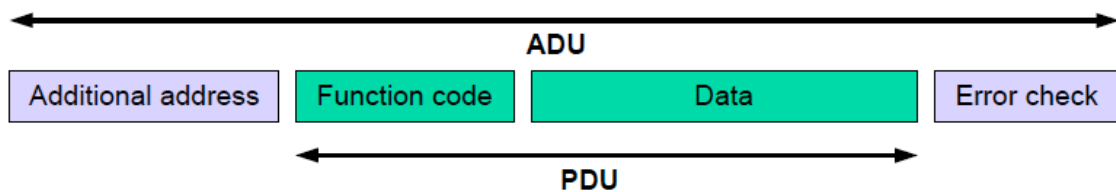


Fig. 1.3.5 ADU de Modbus

- **Address field (camp adreça, 1 byte):** En aquest camp s'especifica la direcció a la qual es transmet la informació. Si la trama la envia un dispositiu mestre, aquesta adreça pot prendre el valor 0 (broadcast) i valors entre 1 i 247 (destinada a esclaus). Quan l'esclau retorna la seva resposta, posa la seva pròpia adreça en el camp d'adreça per fer saber al mestre qui envia la trama.
- **Function code (camp funció, 1 byte):** En aquest camp s'inclou un codi corresponent a quin tipus d'informació porta l'esclau. D'aquesta manera el mestre serà informat de quina acció haurà de gestionar. Function code "0" no es un valor acceptat.

Existeixen 3 tipus de codis de funció:

- Public function code: codis de funció únics definits segons estàndard, validats per l'organisme MODBUS-IDA.

- User-defined function codes: codi de funció seleccionat i implementat per usuari, sense suport per part de l'especificació.
 - Reserved function codes: codis reservats actualment per tal de ser utilitzats per algunes empreses.
- **Data (camp de dades, 0-252 bytes):** Les dades són informació addicional que l'esclau envia al mestre per realitzar la funció especificada en el camp anterior (function code). Pot donar-se el cas que no hi hagi dades, en el qual aquest camp ocuparia 0 bytes, o que hi hagi dades estructurades en N paraules de 8 bits (1 byte) fins a aconseguir un màxim de 252 bytes.
 - **CRC (camp de checksum, 2 bytes):** Camp que inclou un detector d'errors de 16 bits. Mitjançant aquest CRC poden detectar-se errors, però no corregir-se. Per poder corregir-los el nombre de bits de dedicats al CRC hauria de ser major però, per a les funcions que ha de dur a terme Modbus, no resulta necessària la correcció d'errors. Com pot veure's, el camp CRC es compon de 2 bytes; un d'ells correspon al valor de CRCLOW i l'altre a CRCHIGH.

De les grandàries que pot suportar cada camp es pot deduir que la grandària màxima d'una trama o missatge Modbus no pot ser major que 256 bytes. Així que, es pot afirmar que es tracta d'un protocol que amb prou feines introdueix càrrega a la xarxa.

1.3.3. Tipus de dades

Les instruccions Modbus estan orientades a bit o a word (16 bits) encara que és possible llegir/escriure dades de major nombre de bits mitjançant les instruccions de lectura/escriptura de múltiples words. Aquests comandos manegen grups de bytes consecutius fins a completar el tipus de dada desitjada. Els tipus de dades més comunes són:

Tipus	Núm. Bits	Núm. Bytes	Instruccions lectura Modbus	Instruccions escriptura Modbus
Bit	1	-	Read Input Discrete Read Coil	Write Single Coil Write Multiple Coil
Byte	8	1	Read Input Register Read múltiple Register	Write Single Register Write Multiple Register
Word	16	2		
Double Word	32	4		
Float	32	4		
Double Float	64	8		

Fig. 1.3.6 Taula de dades Modbus

El protocol Modbus suporta únicament la lectura/escriptura de bits i words (16 bits). Per treballar amb dades de diferent grandària és necessari llegir/escriure el nombre de words consecutius, necessaris per formar la dada amb la grandària desitjada. Per exemple: Si desitgem llegir un double-word, haurem de llegir dos words (4 bytes) consecutius.

1.3.4. Diàleg esclau-mestre MODBUS

Un cop conegudes les trames utilitzades en protocol Modbus, es mostrarà el diàleg establert entre esclaus i mestres per tal de mantenir una comunicació lògica.

En el cas en que l'ADU enviada per un client, caracteritzada per seu function code sigui rebuda pel mestre sense errors, aquest enviarà la informació demanada en primera instància. En cas de detectar-se algun tipus d'error, el mestre generarà un "exception code" que marcarà la pauta a seguir per tal de rebre correctament l'ADU.

A continuació (Fig. 1.3.7) podem veure un diàleg exemple en que un esclau llegeix estats ON/OFF des d'un grup de sortides discretes o un grup de registres. Quan el servidor genera la resposta inclou al camp "function code" que no s'ha detectat cap error. Simplement, respondrà a l'esclau amb el "function code" original que contenia la petició inicial.

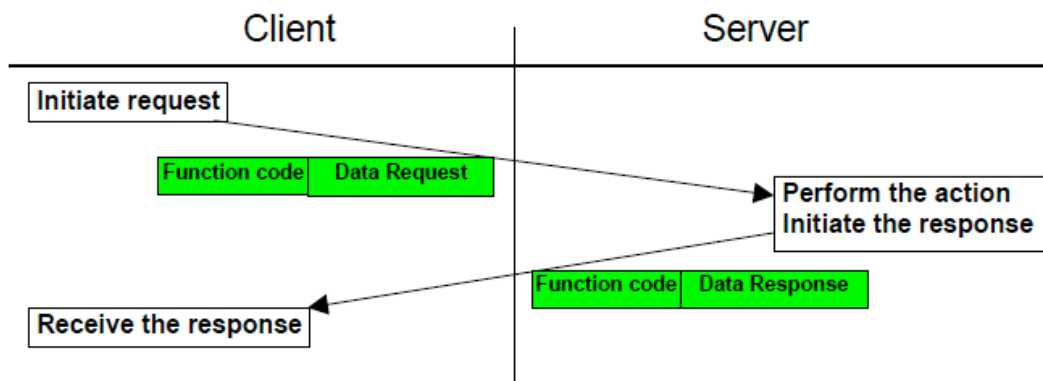


Fig. 1.3.7 Diàleg Modbus sense errors

En el cas de rebre una trama ADU amb error, el servidor generarà una "exception response". El mestre tornarà a l'esclau un function code equivalent a l'original insertat en la última petició, encarà que canviarà el valor del bit de més pes a un 1 lògic.

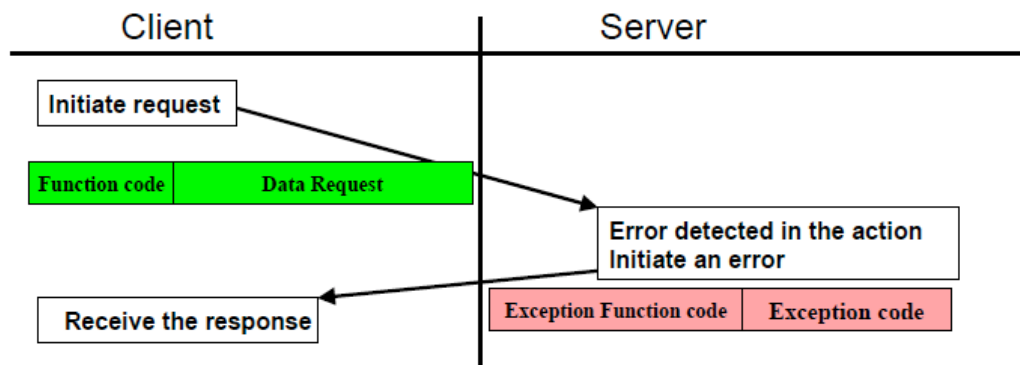


Fig. 1.3.8 Diàleg Modbus amb errors

1.3.5. Model de dades MODBUS

El protocol Modbus, en la seva versió original, suporta 4 tipus de dades:

- **Sortides digitals (adreces 0001-09999):** són sortides físiques discretes. Requereixen un bit que pot prendre els valors 0 o 1 i permeten accés d'escriptura.
- **Entrades digitals (adreces 10001-19999):** són entrades físiques discretes. Requereixen un bit que pot prendre els valors 0 o 1 i permeten accés d'escriptura/lectura.
- **Entrades analògiques (adreces tipus 30001-39999):** són entrades físiques analògiques que funcionen amb registres de 16 bits i que permeten accés d'escriptura.
- **Sortides analògiques (adreces 40001-49999):** es tracta de sortides físiques analògiques o registres interns de l'equip. També conegudes amb el nom de holding registers. Registres de 16 bits que permeten accés de lectura i d'escriptura.

En la taula 1.3.9 es poden veure aquestes mateixes dades però de forma més clara:

Tipus de dades	Tipus d'accès	Rang de memòria
Sortides digitals	Espectura	00001-09999
Entrades digitals	Espectura / Lectura	10001-19999
Entrades analògiques	Espectura	30001-39999
Registres de memòria	Espectura / Lectura	40001-49999

Tabla 1.3.9 Estructura de les dades Modbus.

1.3.6. Transmissió de dades en sèrie

El protocol MODBUS és capaç d'utilitzar dues maneres de transmissió de dades a través d'una xarxa MODBUS: el mode RTU i el mode ASCII. La missió d'aquesta diferenciació és definir de quina manera es realitzarà la transmissió de dades en sèrie, el contingut en bits dels diferents camps del missatge, com la informació s'empaquetarà dins del camp de missatge i com es descodificarà.

Mentre que el mode ASCII, malgrat ser requerit específicament per algunes aplicacions i dispositius, és tan sols un mètode opcional a l'hora d'implementar-ho, el mode RTU és obligatori i ha de ser implementat en tots els dispositius MODBUS, per la qual cosa, aquest últim és el mètode seleccionat per defecte. Cal destacar que el mètode de transmissió i els paràmetres dels ports sèrie triats han de ser iguals per a tots els dispositius d'una mateixa xarxa MODBUS.

En aquest projecte s'implementarà el mode MODBUS RTU (a través de port sèrie), que acabarà traduint-se en MODBUS TCP a través de cablejat ethernet i en la connexió final amb el nou sistema SCADA.

1.3.6.1. Mode de transmissió ASCII

En mode ASCII totes les trames comencen amb un caràcter ":" i acaben amb CRLF (tornada de carro i alimentació de línia, codis ASCII 13 i 10 respectivament). Tots els bytes transmesos entre tots dos delimitadors són caràcters ASCII entre '0'-'9' i 'A'-'F' representant, en hexadecimal, els valors binaris a transmetre. A manera d'exemple, el valor 00111111 (BIN) = 3F (HEX), es transmet com '3' seguit de 'F'.

En aquest mode les trames són del doble de la longitud que en RTU per tal de complir la mateixa funció. L'avantatge principal de la utilització d'ASCII és que és més fàcil d'implementar, atès que no requereix temporitzacions precises. És per això que molts equips només suporten aquesta manera de comunicació. Des del punt de vista del diagnòstic, és també més simple, ja que les trames es poden veure directament. Per contra, presenta com a desavantatge que les trames tenen una longitud d'aproximadament el doble que les trames RTU, la qual cosa té l'efecte net de reduir la velocitat de comunicació a la meitat. La manera ASCII es presta millor per al desenvolupament de drivers de comunicació mitjançant llenguatges d'alt nivell.

1.3.6.2. Mode de transmissió MODBUS RTU

En els missatges enviats en mode RTU, cada byte conté un caràcter compost per dues paraules de 4 bits en hexadecimal.

El gran avantatge d'aquest mode és que amb la mateixa taxa de transferència s'obté un "throughput" major que utilitzant ASCII, ja que s'utilitzen més bits per a cada paraula. D'altra banda, cal destacar que cada missatge ha de ser transmès en un flux continu de paraules o caràcters. Així doncs, és més eficient ja que les trames són més curtes, i aprofita millor els enllaços lents com els radials o per una altra part, aporta un menor temps d'actualització quan la xarxa té un gran nombre d'esclaus presents.

Per contra, el seu principal desavantatge és que és més difícil d'implementar, ja que requereix de temporitzacions precises per marcar la fi del missatge.

CAPÍTOL 2. ESCENARI EN INDÚSTRIA

Un cop descrites les característiques del protocol de comunicacions Modbus a utilitzar, en el següent capítol es descriuran els equips hardware que interactuaran en l'escenari que ens presenta el client.

2.1. Escenari hardware

2.1.1. Especificacions generals del Hardware

A continuació s'especificarà el hardware que ens trobem "in situ" per tal de realitzar la migració en aplicació SCADA obsoleta actualment.

- PLC Klöckner Moeller PS316

El PS316 és un PC modular, que podem configurar segons les necessitats de la instal·lació. Com a màxim podem posar 64 targetes d'entrades digitals i d'altres de sortides, així mateix, podem posar targetes intel·ligents, com poden ser les targetes d'entrades o sortides analògiques.

A continuació es mostra quadre de característiques del PLC PS316:

Model	SUCOMAT PS316
Bits de dades	8
Paritat	None
Velocitat (Baud rate)	9.600
Protocol de comunicació	SUCOMA
Mode de comunicació	Punt a punt
Adaptador	RS485

Tabla 2.1.1 Característiques PLC Klöckner Moeller PS316.

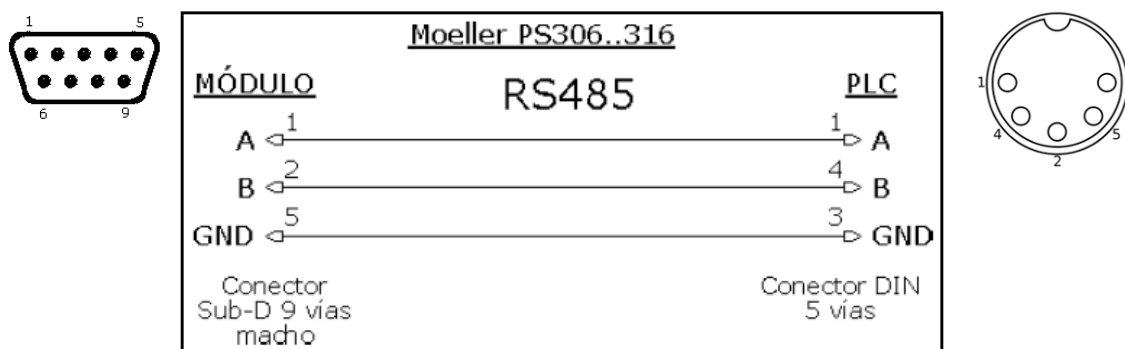


Tabla 2.1.2 Configuració de pins DIN 5 / DB9.

Podem extreure finalment que gràcies a l'adaptador de comunicacions RS485 que presenta Klöckner Moeller serem capaços de llegir i escriure dades en el PLC amb la preparació d'un cable interface Connector DIN5-DB9. Veure configuració de pins (Fig. 2.1.2).

- SCADA MITOR Dos Based

Aplicació SCADA de nom MITOR basada en sistema MS-DOS. Plataforma desenvolupada durant els anys 80 que estableix comunicació punt a punt amb PLC PS316.

2.1.2. Interface LANCE-E

El Lancelan © LANCE-E és un adaptador de comunicacions universal ethernet/sèrie per a la comunicació amb la majoria de PLC's del mercat de forma local o remota.

Aquestes són només algunes dels avantatges que caracteritzen al LANCE-E:

- Alta velocitat de procés en cicles molt curts
- Protocol genèric veritablement simple per a l'accés a les dades del PLC
- Protocol MODBUS per compatibilitat amb equips que només comuniquin mitjançant aquest protocol
- El port sèrie proporciona una "connexió flexible" a elements HMI, terminals d'operador, PCs i dispositius basats en microprocessador
- Disseny robust i compacte per a la seva integració en ambients industrials

Les possibles aplicacions del mòdul LANCE-E són, entre unes altres:

El registre cíclic de valors dels registres, la connexió amb un programa de control de producció per a la captura de dades de procés o la supervisió del procés en marxa, ja sigui local o remota, petites aplicacions SCADA, etc..

Gràcies a que el LANCE-E utilitza indistintament una estructura de protocol obert o el protocol MODBUS, qualsevol aplicació podrà comunicar amb ell.

Aquest podrà ser també usat per aplicacions escrites en llenguatges d'alt nivell tals com: Visual *Basic, C i C++ o *Java, encara que també podrà ser usat per aplicacions basades en microprocessador o, fins i tot, escrites en assemblador. ió en ambients industrials.

2.1.2.1. *Característiques tècniques*

Gràcies a les prestacions que ens presenta l'interface escollit dins del ventall de productes que podem trobar al mercat, haurem de realitzar les següents connexions sobre el Lance-E:

- **Port sèrie:** Connectarem el cable provinent del PLC Klöckner Moeller PS316 (de DIN5 a DB9). Per tant tindrem un connector DB9 connectat al dispositiu interface i el configurarem segons els següents paràmetres:
 - Adaptador: RS485
 - Velocitat: 9600 bauds
 - Paritat: None
 - Número de bits: 8

- **Port ethernet:** A través del cable ethernet farem comunicar el PC que allotjarà el sistema SCADA amb el propi Interface mitjançant el protocol TCP/IP. Assignarem una direcció IP tant al PC SCADA com al Lance-E, que es trobin en el mateix rang.

- **Alimentació:** Connectarem dos fils provinents d'una presa de corrent amb subministrament 220AC. No tindrem problemes de consum ja que l'interface indica unes especificacions a nivell de potència de 3W.

- **Port de configuració:** Haurem de carregar les diferents configuracions al dispositiu interface a través del connector jack situat al lateral de l'equip. El cable utilitzat serà Jack / DB9.



Fig. 2.1.3 Cable de configuració Jack / DB9

#	Esquema	Tipus de port
1		Port sèrie configurable a RS485
		De 1200 a 187.500 bauds
		Paritat none, even o odd
		7 o 8 bit
		1 o 2 stop bits
		Connector sub-D femella
2		Port ethernet/10
		Connector RJ45
		Protocol TCP/IP
		Missatgeria UDP i TCP
		Servidor web
		Servidor telnet
3		Alimentació
		220Vac
		Potència 3W
		Connexió per regleter amb cargol
4		Port de configuració
		Sèrie RS232
		Connector jack estereo situat al lateral
		Permet configuració interna

Fig. 2.1.4 Característiques connexionat interface Lance-E

2.1.2.2. Mode de connexió

En mode local efectua les tasques de conversió de protocol necessàries per a la comunicació amb PLC's i dispositius sèrie situats en l'entorn on s'executa l'aplicació. D'aquesta manera, el PC pot comunicar amb el Lance-E indistintament pels ports sèrie 0 i 2 o el port ethernet utilitzant missatgeria TCP o UDP. Tot això serà implementat en protocol MODBUS/RTU.

L'interface Lance-E utilitza el protocol específic del PLC connectat per efectuar l'operació de lectura o escriptura de valors interns o I/O, generant una resposta al finalitzar.

Un cop conegudes les propietats bàsiques en quant a modes de connexió, es mostra un diagrama de xarxa on s'observa la connectivitat que presentaran els nostres equips en l'escenari real (Veure Fig. 2.1.5).

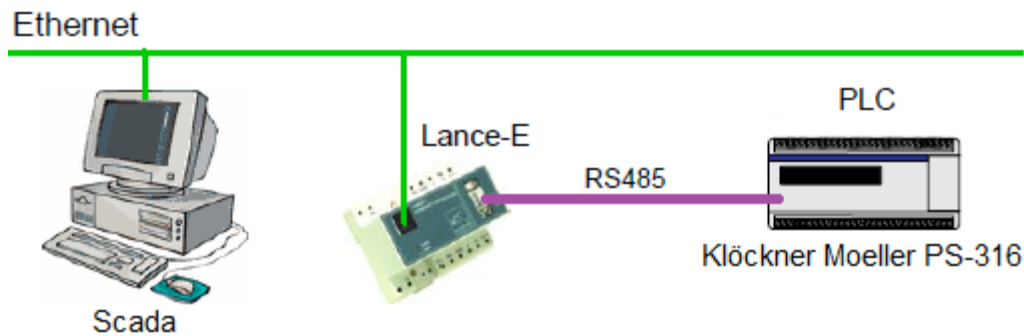


Fig. 2.1.5 Diagrama connexió PLC-Interface-SCADA

Tal i com s'ha apuntat en apartats anteriors, el PLC Klöckner Moeller PS-316 es comunica amb la perifèria mitjançant RS485 amb sortida DIN5. A partir d'aquí es prepara un cable PLC que realitzi la conversió de DIN5 a DB9. Un cop preparada aquesta connexió, l'intercanvi de dades PLC-Interface es realitzarà mitjançant el protocol MODBUS/RTU.

L'interface escollit presenta total compatibilitat amb l'antic model de PLC PS-316, per tant, internament s'encarregarà de realitzar les peticions de lectura en el cas en que el nostre SCADA vulgui mostrar dades o en contraposició, escriurà sobre els registres, marques i entrades que contingui el PLC, si el que volem es canviar paràmetres de la instal·lació.

2.1.2.3. Comunicació interface mitjançant MODBUS

El Lance-E implementa el protocol Modbus amb la finalitat de facilitar la comunicació amb PLC's de diferents marques a aquells dispositius que només suportin aquest protocol. Pel port sèrie 2 implementa el protocol MODBUS/RTU. Pel socket base+1 del protocol TCP sobre ethernet implementa el protocol MODBUS/TCP.

Per utilitzar el protocol Modbus és necessari associar les adreces de les dades que apareixeran en les trames del protocol Modbus amb les adreces específiques del PLC amb el qual es desitja comunicar.

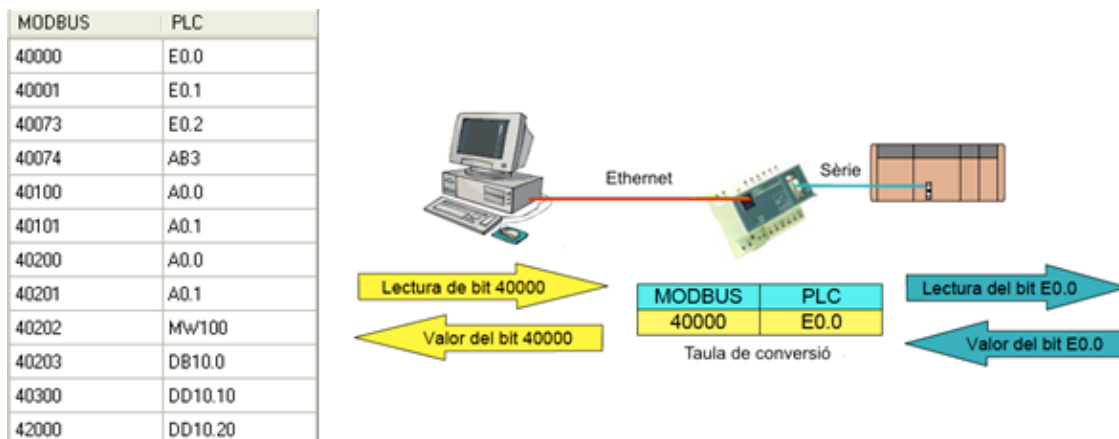


Fig. 2.1.6 Taula exemple associació registre Modbus

Aquesta taula té dues columnes:

En la de l'esquerra s'introdueix l'adreça Modbus mentre que en la de la dreta s'introduirà el registre o element d'E/S que anirà associat a aquesta adreça.

En aquest exemple podem observar que hem assignat al registre Modbus número 40000 el bit del PLC I0.0. Això significa que, en ser rebuda una petició de lectura del registre 40000 el Lance-E emetrà, al seu torn, una petició de lectura del bit I0.0 del PLC, generant una resposta en Modbus en ser rebuda la contestació.

El tipus de variables que s'allotjen al PLC Klöckner Moeller PS-316 son les següents:

Variable	Descripció	Bits	Registre	Explicació	Rang
I	Entrada externa	1	In	n=bit	0-1
Q	Salida externa	1	Qn	n=bit	0-1
M	Marcador	1	Mn	n=bit	0-1
IB	Entrada externa	8	IBn	n=byte	n=byte
QB	Salida externa	8	QBn	n=byte	n=byte
MB	Marcador	8	MBn	n=byte	n=byte
IW	Entrada externa	16	IWn	n=word	0-65535
QW	Salida externa	16	QWn	n=word	0-65535
MW	Marcador	16	MWn	n=word	0-65535

Fig. 2.1.7 Taula tipus de variables en PLC Klöckner Moeller

2.1.2.4. Configuració dispositiu interface Lance-E

El dispositiu interface Lance-E tal i com s'ha especificat en capítols anteriors es adaptable per a pràcticament tota classe de PLCs comercials, i els seus ports s'adapten a multitud de configuracions i protocols de comunicacions. Per tant procedirem a configurar pas a pas el dispositiu per tal d'adaptar el nostre enllaç entre el PC SCADA y el PLC al nostre gust segons les necessitats del projecte que ens ocupa.

- **Configuració tipus de PLC:**

Amb el cable jack / DB9 ens connectarem al punt jack estèreo situat al lateral del dispositiu. Aquesta configuració es pot entrar tant via Telnet com amb l'editor Lancelan, software entregat pel fabricant per tal de fer les tasques de configuració més senzilles.

- Haurem d'indicar el codi del nostre PLC "SUCO"
- Activarem les escriptures en PLC

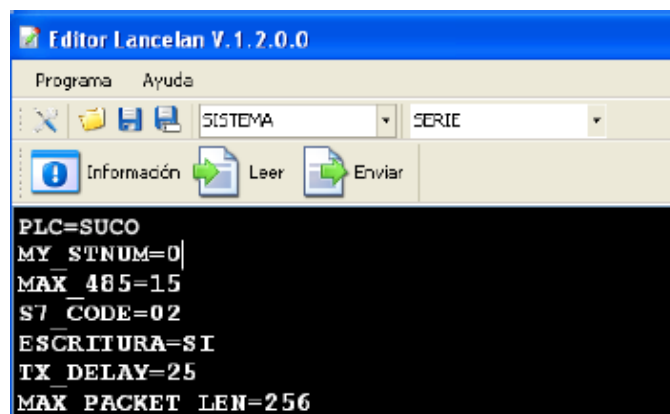


Fig. 2.1.8 Configuració tipus de PLC en editor Lancelan

- **Configuració adreça IP:**

Un altra punt important en el procés de configuració serà l'assignació d'una adreça IP en el dispositiu interface que estigui en el mateix rang que l'SCADA. D'aquesta manera es comunicaran sense problemes sobre un enllaç punt a punt. Serà també en l'editor Lancelan on carregarem l'adreça adient en el Lance-E.

- Assignarem la adreça IP 192.168.1.86

- **Configuració de taula d'assignació Modbus <-> PLC:**

Perquè la unitat remota pugui accedir a les dades del PLC serà necessari crear la taula d'assignació que relacioni els elements que requereix la unitat remota amb els elements d'E/S i registres interns del PLC.

En la columna de l'esquerra posarem el nombre de registre Modbus tal com ho implementarà la unitat remota, mentre que, en la columna de la dreta posarem els elements d'E/S (bits i words) i els registres interns (words, double word) que sapiguem que contenen les dades que ens interessin. Una vegada definida la taula l'enviarem al Lance-E prement el botó ENVIAR.

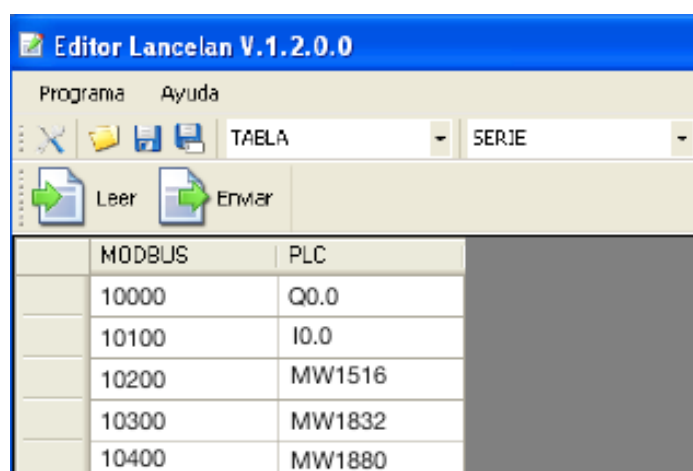


Fig. 2.1.9 Configuració taula d'assignació en editor Lancelan

2.2. Escenari software

2.2.1. Què es Labview?

LabVIEW (Laboratory Virtual Instrument Engineering Workbench, de National Instruments) és un llenguatge de programació gràfic pel disseny de sistemes d'adquisició de dades, instrumentació i control. Labview permet dissenyar interfícies d'usuari mitjançant un programa executable final. Es pot dissenyar especificant el seu sistema funcional, el seu diagrama de blocs o un programa en llenguatge informàtic.

Labview és ahora compatible amb eines de desenvolupament similars i pot treballar amb programes d'altres àrees d'aplicació, com per exemple Matlab.

2.2.2. Programació gràfica Labview

Quan es dissenyen programes amb Labview s'està treballant sempre sota el que es denomina VI, és a dir, un instrument virtual. Es pot crear un VI a partir d'especificacions funcionals. Aquest VI es pot utilitzar en qualsevol altra aplicació com una subfunció o subprograma dins d'un programa general i podem fer tantes crides a un VI com les necessitats del software ho requereixin.

L'escenari de programació presenta dues parts fonamentals, el panell frontal i el diagrama de blocs. El panell frontal és la interfície amb l'usuari i és aquí on es defineixen els controls (entrades) i els indicadors (sortides) que es mostren en pantalla.

Podríem dir que es la interfície gràfica, es la part més important de cara al client ja que s'haurà d'adaptar a la perfecció amb els requeriments definits, aportant un punt d'avenç degut a la gran capacitat de desenvolupament que Labview proporciona.

El diagrama de blocs conté les funcionalitats del panell i es programa de forma gràfica, es a dir fent connexions virtuals, mitjançant fils ó cables, de diferents blocs que realitzen funcions molt determinades per crear una funció feta a mida.

Una bona estructura de programa serà la base per muntar el trencaclosques final que suposa unir totes les VI creades per a cada necessitat. A més, resultarà vital presentar un codi de programa organitzat ja que el descontrol a nivell de software acabarà afectant a nivell hardware. Amb aquesta afirmació es vol dir, que podem tenir problemes a l'hora de llegir dades i escriure en el PLC, degut a temps de cicle molt elevats o saturació d'informació a través del RS485 que suporta l'autòmat. Elèctricament fins i tot, pot arribar un punt en que el PLC no activi les sortides comandades des del sistema SCADA. Per ajudar-nos, NI amb el seu Labview ens proporciona diferents solucions a nivell estructural.

Com a resum destacarem tres de les estructures mes utilitzades durant el nostre projecte:

- **While loop:**

Es un bucle que executarà el codi que es troba dins seu sempre i quan es compleixin unes condicions externes definides pel programador. La utilització d'aquesta estructura obliga a més a introduir condicions de parada, per tal d'evitar els bucles infinits. Això ens ajudarà a fer una parada controlada dels processos en el moment de tancar l'aplicació.

- **Case Structure:**

Definida com una estructura de programa que defineix diferents casos que seran executats de forma independent en funció de la crida externa que es faci.

- **Flat sequence:**

L'última estructura a destacar de entre les múltiples possibilitats que Labview ens ofereix, ens dona la possibilitat d'executar el codi de programa seqüencialment. Fins que no finalitzi la execució de la primera part de programa no continuarà amb la resta.

A continuació es poden visualitzar la totalitat de les funcions a nivell estructural que aporta la versió 8.2 de Labview:

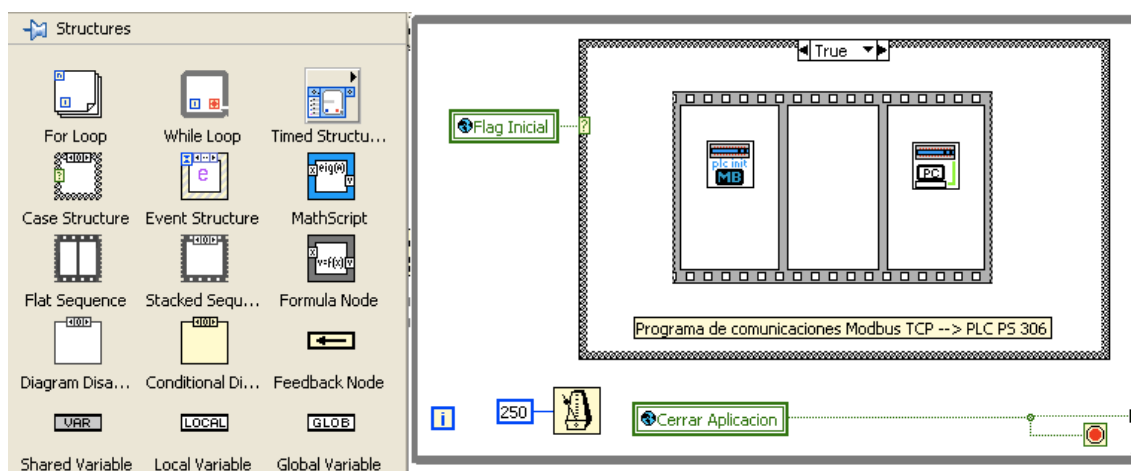


Fig. 2.2.1 Estructures de programa en Labview 8.2

2.3. Definició de projecte

En el següent apartat es definiran els requeriments de projecte en funció del pressupost del client i les possibilitats tecnològiques que aportarien nous equips. Una solució equilibrada en la balança tan important avui dia qualitat preu, serà l'objectiu que buscarà tant el desenvolupador com el client final.

2.3.1. Canvis hardware en la instal·lació

El pressupost final marca la presa de decisions per tal d'aconseguir implementar un sistema SCADA d'última generació.

2.3.1.1. Recursos actuals com a objecte d'estudi

En primera instància es necessari portar a estudi cadascun dels dispositius instal·lats en producció, valorant un canvi en funció de l'impacte econòmic generat així com la capacitat d'adaptació al nou sistema SCADA que es vol instal·lar.

Seguidament analitzarem els desavantatges que presenten cadascun dels equips existents:

- **PLC Klöckner Moeller:** L'autòmat instal·lat data dels anys 80. Klöckner Moeller, actualment Eaton Moeller, no es capaç de subministrar peces de recanvi. Una fallida de qualsevol dels components del PLC deixaria la planta sense un control automàtic de temperatures en les cambres. Cal recordar que en la indústria alimentària, la temperatura es un paràmetre crític.
- **Scada Mitor sobre màquina I486:** L'SCADA que es vol migrar a una nova plataforma està basat en MS-DOS. A més, el software s'executa sobre una màquina amb processador Intel I486, totalment obsolet des de 2007.
- **ADAM 4521 Addressable RS-422/485 to RS-232 Converter:** Dispositiu encarregat de realitzar la conversió del protocol RS485 amb que comunica el PLC Klöckner Moeller a RS232, port de comunicacions que trobem al PC on s'allotja el sistema SCADA. L'ADAM 4521 presenta les següents característiques:
 - Control de flux automàtic sobre RS-485
 - Aïllament 1000VDC
 - Velocitats de transmissió fins a 115,2kbps
 - Connexió en xarxa sobre distàncies superiors a 1200m.
 - Requeriments d'alimentació: de +10 a +30 VDC
 - Fàcil muntatge sobre carril DIN
 - Configuracions de comunicació sobre EEPROM

Una de les limitacions més importants que presenta l'ADAM es la connexió a través de port sèrie. Una xarxa ethernet facilitarà les comunicacions amb el nou sistema SCADA desenvolupat sobre Labview.

Un cop analitzats els desavantatges de cadascun dels equips, resumirem la instal·lació en el següent esquema:

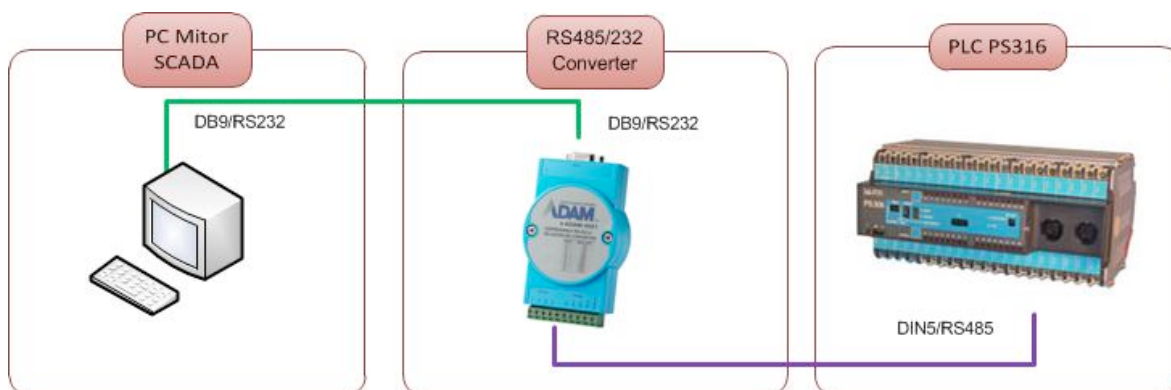


Fig. 2.3.1 Esquema dispositius en indústria

2.3.1.2. Proposta final de l'enginyeria elèctrica

La premissa per definir aquest projecte amb requisits del client, es la de minimitzar al màxim la inversió, mirant d'aprofitar els equips disponibles actualment per tal d'obtenir un pressupost molt ajustat.

A nivell software el client, amb poc coneixement informàtic vol mantenir la mateixa filosofia en el nou sistema SCADA que l'existent en el seu antecessor, per tal d'assegurar una adaptació ràpida dels treballadors de la planta. El repte dels desenvolupadors del nou software de supervisió de la instal·lació serà la de imitar una forma de treballar actualment obsoleta sobre una plataforma d'última generació.

Finalment es proposa al client un bon projecte parlant en termes qualitat-preu. Tot seguit es descriuran els equips i software finals:

- Autòmat:

El PLC es l'encarregat de portar el control dels paràmetres en la instal·lació. En el cas d'aquest projecte, activarà les cambres frigorífiques segons els valors entrats al sistema SCADA, diagnosticarà les diferents alarmes actives i mantindrà les temperatures desitjades segons el tipus de cambra.

Finalment es decideix mantenir el PLC existent Klöckner Moeller PS316 principalment degut a la gran inversió que suposaria instal·lar un PLC d'última generació. A més, diferents dispositius en el mercat ens permetran la interconnexió amb el PLC avui dia obsolet. Per últim, i per tal d'eliminar el factor de risc que suposaria la pèrdua de l'autòmat, s'aconsegueix en el mercat un PLC de les mateixes característiques com a recanvi.

- Scada:

En aquest cas, s'instal·larà un nou PC que haurà de presentar com a requisits mínims:

Processador	Pentium4/M o equivalent
RAM	1GB
Resolució de pantalla	1024x768 píxels
Sistema operatiu	XP Professional
Espai en disc dur	30GB

Fig. 2.3.2 Requisits mínims PC Scada

Sobre l'equip indicat s'instal·larà LabVIEW Run-Time Engine 8.5. Aquest paquet de National Instruments dóna la capacitat de suportar executables desenvolupats en Labview. A més, permet la visualització de panells sobre exploradors web. Com a punt a favor, cal indicar que és totalment gratuït.

L'elecció del software Labview per tal d'implementar el nou sistema Scada ens aporta grans avantatges:

- Primerament, des del punt de vista econòmic ens ajuda a presentar un pressupost molt ajustat en el que respecta a la gestió de llicències. Tan sols el desenvolupador ha de tenir una llicència (NI Developer Suite) ja que el client final tindrà prou amb el Labview Run-Time Engine 8.5, paquet de software gratuït per executar l'aplicació Scada.
- A nivell de comunicacions, Labview 8.2 presenta compatibilitat total amb Modbus. Subprogrames (VI) creats especialment per aquest protocol de comunicació facilitaran de forma notable l'intercanvi. A més definir les senyals d'intercanvi entre PLC i SCADA es converteix en una tasca molt senzilla.

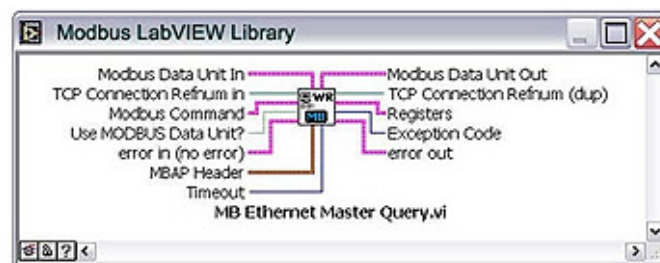


Fig. 2.3.3 Requisits mínims PC Scada

- Dispositiu Interface:

Anteriorment sobre la connexió entre PLC i SCADA es feia una conversió de RS485 (PLC) a RS232 (SCADA) a través de l'interface ADAM4521.

Finalment s'instal·larà l'equip Lance-E en substitució de l'antic ADAM4521. El nou Lance-E millorarà la qualitat de les comunicacions gràcies als segons trets:

- Realitza la conversió RS485 (PLC) a Ethernet (nou sistema SCADA). Per tant, eliminem l'antiga connexió RS232.
- Ens permet configurar fàcilment els registres d'interactuació amb el sistema SCADA. Aquest registres es traduiran en entrades, sortides i marques allotjades en el codi del PLC.

- PC Scada en xarxa:

Per últim el client marca com a requisit entrar el PC Scada en el domini de xarxa existent, per tal de permetre la impressió de les dades obtingudes des del sistema de visualització.

Finalment un cop analitzats els diferents elements que intervindran en la modificació, es mostra l'esquema final d'interconnexió:

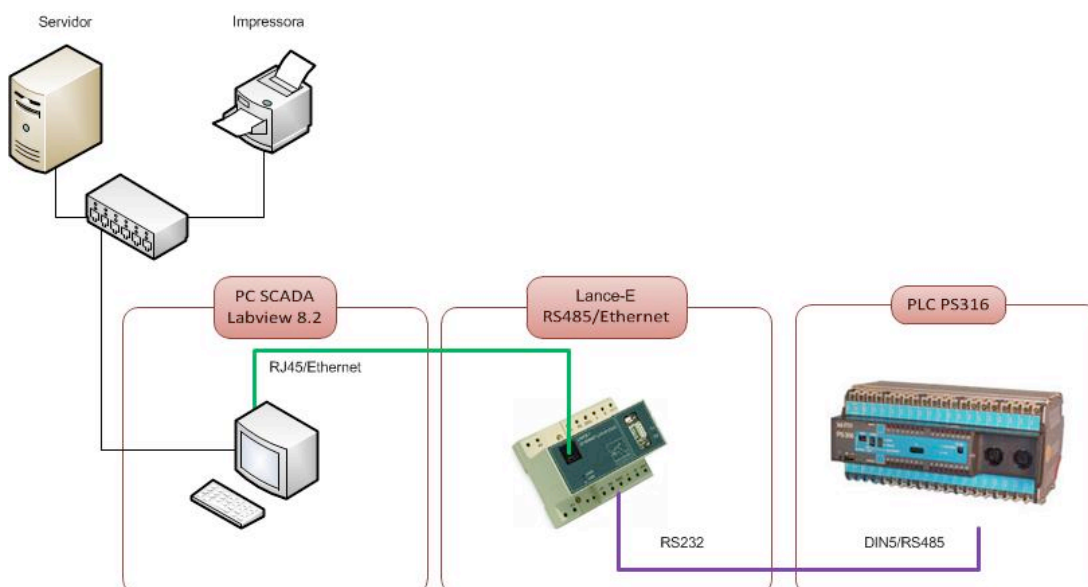


Fig. 2.3.4 Esquema d'interconnexió dispositius

CAPÍTOL 3. DESENVOLUPAMENT SOFTWARE

En el següent capítol es descriuran els processos a nivell software que s'han anat desenvolupant un cop muntat l'escenari hardware, per tal d'implementar l'SCADA Labview amb els requisits definits.

3.1. Test de comunicacions MODBUS

Per tal de llegir i escriure dades entre els sistemes SCADA y PLC, s'haurà de fer un estudi de la distribució de memòria dins del codi de l'autòmat, traduir les entrades/sortides en registres MODBUS, i per últim definir-los en el software que s'executa en el SCADA per poder visualitzar i forçar sortides.

3.1.1. Definició de senyals en codi PLC

Primerament consultarem el simbòlic del PLC, es a dir, l'assignació que s'ha donat a cada entrada i sortida del PLC. De igual manera, podem identificar cada senyal a través de l'esquema elèctric, encara que s'optarà per accedir al codi del PLC a través del software propietari de Klöckner Moeller, Sucosoft 3. Després d'aquest procés obtindrem un llistat compost de les marques, entrades i sortides definides al PLC.

A continuació es mostra un exemple en que es representa la distribució de memòria de la double word (16bits) M100.0:

Marques	Descripció
M 100.1	Inhibición evaporador 1 conservación
M 100.2	Inhibición evaporador 2 conservación
M 100.3	Inhibición evaporador 3 conservación
M 100.4	Inhibición evaporador 4 conservación
M 100.5	Inhibición evaporador 5 conservación
M 100.6	Inhibición evaporador 6 conservación
M 100.7	Inhibición evaporador 7 conservación
M 100.8	Inhibición evaporador 8 conservación
M 100.9	Inhibición evaporador 1 congelación
M 100.10	Inhibición evaporador sala 1
M 100.11	Inhibición evaporador sala 2
M 100.12	Inhibición evaporador sala 3
M 100.13	Inhibición compresor C1 conservación
M 100.14	Inhibición compresor C2 conservación
M 100.15	Inhibición compresor C3 conservación

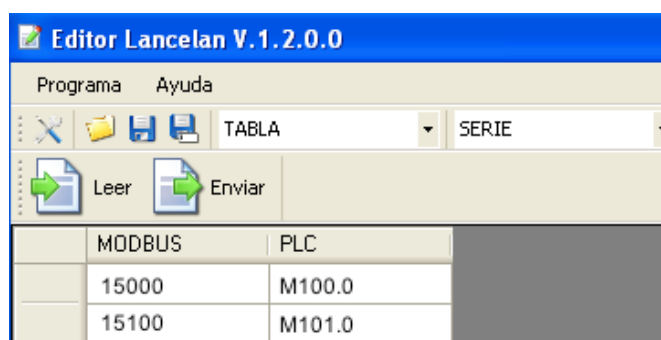
Fig. 3.1.1 Distribució de memòria double word M100.0

3.1.2. Definició de senyals en Interface Lance-E

El següent pas, i tenint present l'exemple anterior que treballava amb la double word 100.0, definirem els 16 bits a la taula d'assignació MODBUS allotjada en memòria del dispositiu Lance-E. Per tant, tan sols haurem d'assignar una codificació MODBUS per tal de relacionar les dades que ens interessin entre el PLC i el sistema SCADA.

Assignarem a la marca 100.0 el registre MODBUS 15000. Serà en el software Labview on indicarem que volem llegir els 16 bits en total. En canvi, a la taula d'assignació MODBUS hem de procurar no trepitjar àrees de memòria, aquesta serà l'única premissa per tal de distribuir les dades que necessitem.

En el següent exemple s'observa com s'assigna el registre 15000 per a la marca M100.0. Per un altre costat, el següent registre serà el 15100 per tal de no trepitjar els bits que arriben a continuació, M101.0.



MODBUS	PLC
15000	M100.0
15100	M101.0

Fig. 3.1.2 Definició de marques i registres en taula d'assignació MODBUS

3.1.3. Definició de senyals en Labview

Un cop localitzades les senyals desitjades en simbòlic del PLC i seguidament, s'han definit a la taula d'assignació MODBUS amb un registre únic, podem fer una lectura de les dades ja en el sistema SCADA.

A continuació es mostra com es tradueixen les dades entrades a la taula d'assignació MODBUS que hem carregat anteriorment a l'equip Lance-E. Es pot observar a més, que es en aquest punt on indiquem al protocol MODBUS que volem llegir 16 entrades discretes (Read Discrete Inputs , 16). Les dades demanades es carregaran sobre la variable global de nom R M100.0 – M100.15. ¿Perquè una variable global? Amb aquest format, serem capaços d'accedir a les dades que conté des de qualsevol VI carregada en el projecte Labview que s'està executant.

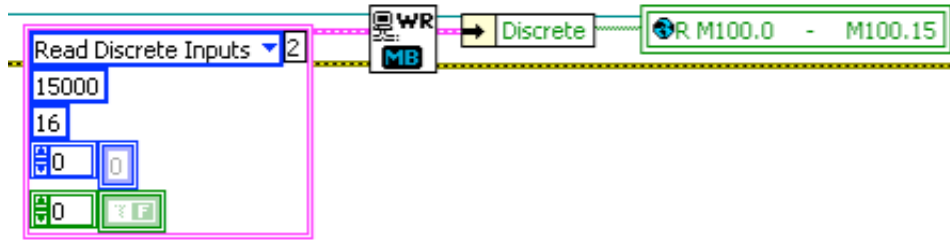


Fig. 3.1.3 Definició de marques i registres en Labview

Amb aquesta petició l'SCADA es defineix com a Master Modbus i el dispositiu Lance-E actuarà com a esclau, responent amb les dades que ha demanat el mestre.

Com hem vist, el procediment per definir les senyals Interface compartides entre PLC i SCADA es senzill, encara que s'ha de seguir una organització acurada.

3.2. Desenvolupament de software en Labview

Molts cops a l'hora de desenvolupar un nou software es busca la funcionalitat i no pas l'eficiència del codi, però en el moment en que tenim un flux de dades important poden aparèixer alguns problemes no contemplats en la fase inicial de projecte. Una nova problemàtica es genera en el moment de la fi d'execució de programa, ja que múltiples sub VI insertades en el codi hauran de patir una parada controlada.

3.2.1. Definició de l'estructura principal

Les situacions comentades han estat algunes de les que han definit l'estructura final del sistema Scada dissenyat. Com es pot comprovar (veure Fig. 3.2.1), els subprogrames pengen de "Menu Principal.vi". Aquesta aplicació arrel, s'encarregarà a grans trets de fer les crides a les pantalles, inicialitzar les comunicacions MODBUS, capturar el registre de temperatures de cadascuna de les cambres frigoríques i visualitzar en sinòptic l'estat general de la instal·lació.

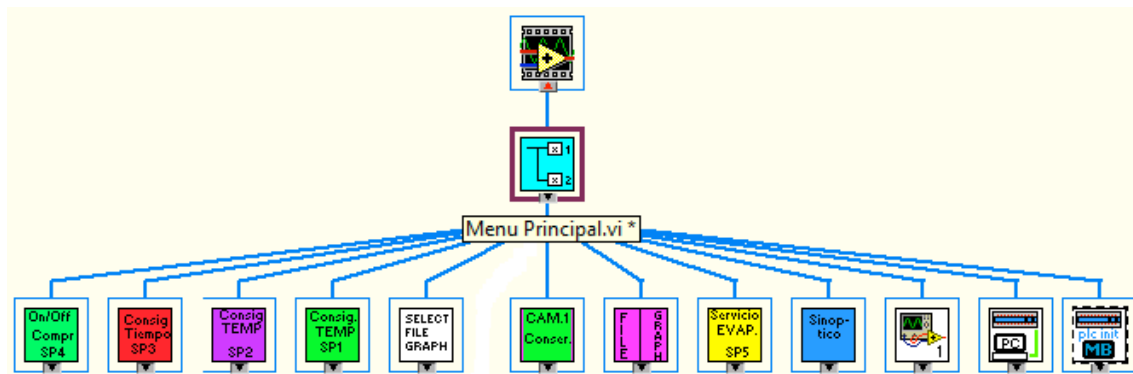


Fig. 3.2.1 Jerarquia de programa

Serà en la subfunció “Menu Principal.vi” on definirem l’estructura que suportarà el nostre sistema SCADA. Ens endinsem per tant en el codi de programació “Block diagram”.

Primerament, i per evitar ensopegades en el moment d’arrancar l’aplicació, definirem una variable global anomenada “Flag inicial”. Podríem dir metafòricament parlant, que es tractarà del tret de sortida per tal d’executar les diferents estructures definides. Com es pot observar a continuació, en la primera execució, s’establirà que “flag inicial = false”. Per tant, el forçarem a “true”.



Fig. 3.2.2 Jerarquia de programa

Un cop hem definit la variable que sincronitzarà les nostres execucions, la repetirem tants cops en codi com estructures haguem programat.

El següent pas serà introduir en programa totes les subfuncions creades al llarg del projecte, una per cada funcionalitat. El que es farà serà definir una estructura “while loop” per cada VI que es vulgui cridar. De moment tenim la variable global “flag inicial” per iniciar l’execució amb sincronia i per tant, adoptarem una nova marca de programa per indicar la fi de programa. Aquesta funció la realitzarà la variable global “cerrar aplicació”. Cada “while loop” també inclourà aquesta última variable i cadascuna d’elles es programa a mode de sèrie. Quan qualsevol de les variables “cerrar aplicació” passi a estat “true”, s’iniciarà l’aturada de totes les estructures de programa.

Per últim, un nou requeriment de projecte apareix en la fase de test del sistema SCADA. La utilització d’estructures while loop, provoquen una càrrega de processador del 98%. Dins del ventall de possibilitats que proposa el software de desenvolupament Labview, s’opta per la utilització de la subfunció

proprietària de National Instruments anomenada “Wait until next ms multiple”. Aquesta aplicació admet la entrada de “integers” per tal d’introduir un valor de temps expressat en milisegons. Amb la utilització d’aquesta opció de programa, farem que l’estructura while loop s’executi cada cop que el rellotge de la CPU contingui un valor múltiple del integer introduït com a paràmetre. Es decideix incloure una temporització de 250ms, per tant, cada cop que el rellotge arribi a un valor múltiple de 250ms, realitzarà un refresc en l’execució i iniciarà la lectura del while loop.

A continuació s’adjunta un exemple on apareixen els elements estructurals del nostre sistema SCADA:

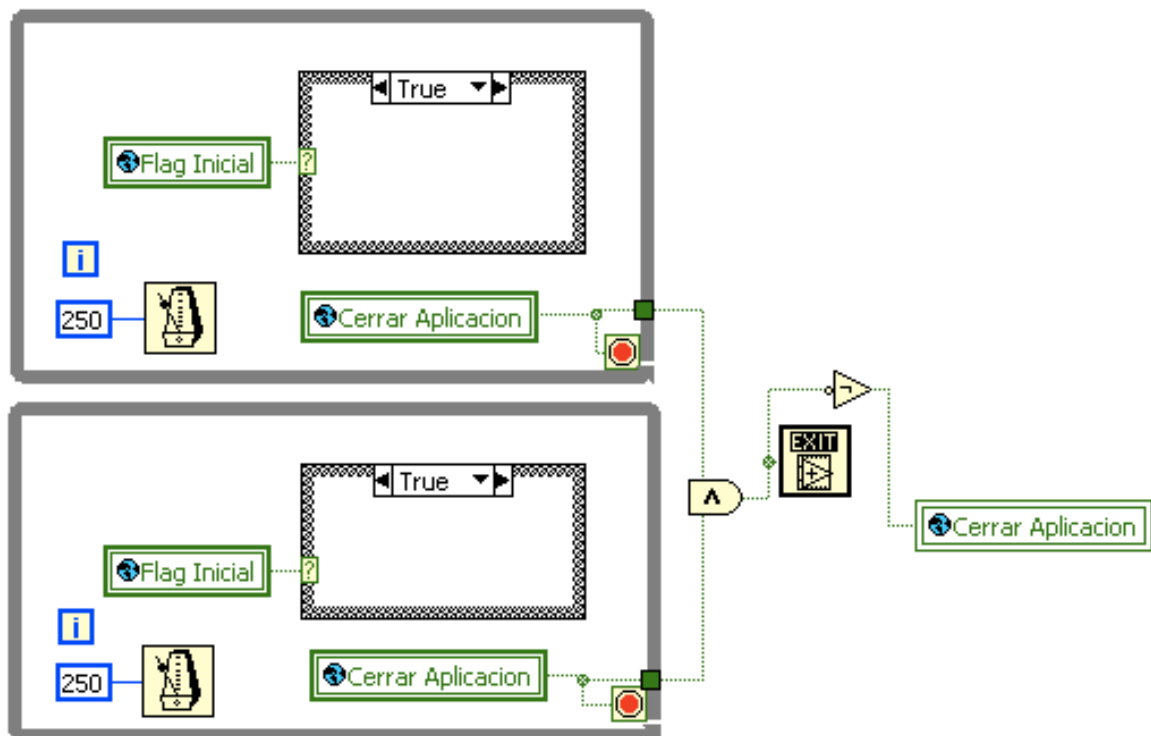


Fig. 3.2.3 Elements estructurals de programa

3.2.2. Descriptiu de subprogrames Vis creats

A continuació es resumeixen algunes de les VI (Virtual Instrument) o subprogrames creats en aquest projecte per satisfer els requisits mínims acordats amb el client.

3.2.2.1. Pantalles de control

- **Sinóptico:** Aquesta serà la pantalla principal i la més utilitzada pels operaris que controlen la planta alimentària. D'una ullada serem capaços

de visualitzar les cambres en servei, la temperatura actual de cadascuna d'elles, així com l'estat d'alarma i règim de treball de motors i compressors.

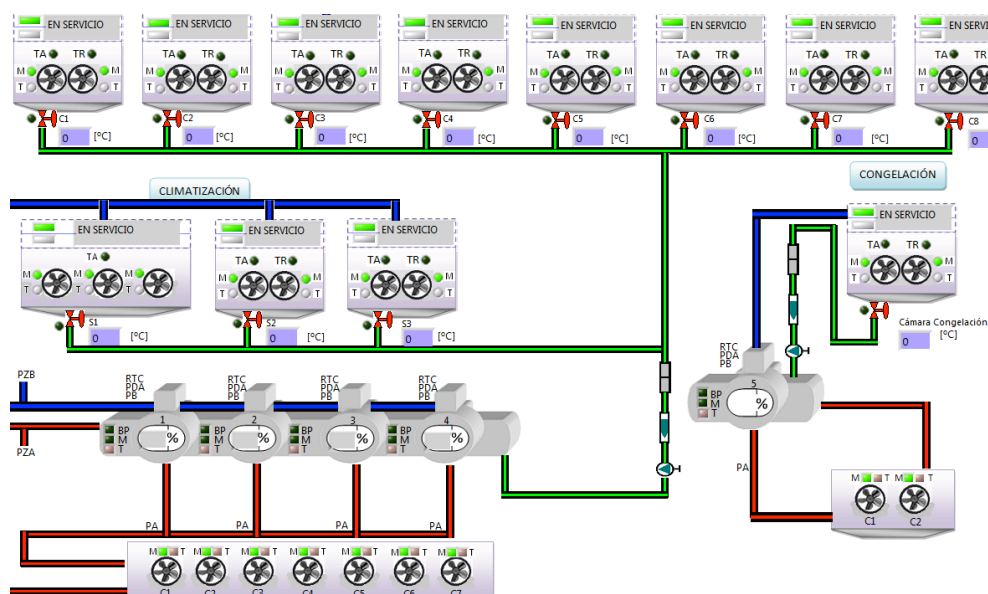


Fig. 3.2.4 Pantalla sinòptic en sistema SCADA

- Consigna temperatura de Trabajo:** En aquesta pantalla assignarem la temperatura que volem en cada cambra. A més, s'incorpora un nou paràmetre anomenat "histèresi consigna". En aquest camp introduïrem la variació de temperatura acceptada abans de tornar a arrancar els compressors. S'incorpora com una mesura per aconseguir un estalvi energètic important, es reduiran les emissions CO2 i es conservarà la mecànica dels equips que hi intervenen.
- Consigna temperatura desescarhe:** Podrem iniciar el procés per eliminar la gebrada que es crea a baixes temperatures a l'evaporador. Un cop alguna de les cambres presenti una temperatura molt superior a la temperatura de treball, programarem el procés per eliminar el gel format i a la temperatura que el volem iniciar. També s'incorporarà una temperatura d'histèresi per deixar un marge de maniobra abans d'iniciar el procés d'eliminació de gebrada.
- Consigna tiempo desescarhe:** S'assignarà el temps que volem que duri el procés d'eliminació de gebrada. A més, s'incorpora un nou paràmetre en el que podem definir el temps entre processos d'eliminació de gebrada. D'aquesta manera podem preveure que sempre que s'iniciï el procés, tindrem gel acumulat. Actuarà com una nova mesura per a l'aportació d'estalvi energètic.

- **Servicio compresores:** Serà una de les pantalles més utilitzades pels operaris encarregats de les cambres frigorífiques. Podrà forçar l'encesa i parada de les cambres frigorífiques desitjades.
- **Tendencia cámaras:** Aquesta no serà la pantalla més utilitzada però sens dubte la més important. En aquesta opció escollirem la cambra desitjada i veurem una gràfica amb l'evolució de les temperatures. Podrem seleccionar una cambra en concret i dia/hora, i d'aquesta manera consultarem l'històric de temperatures. Un cop visualitzades les dades desitjades, podrem imprimir un informe oficial.

La importància d'aquesta pantalla recau en la capacitat de realitzar informes de temperatures per cambra. La indústria alimentària en qüestió, setmanalment es examina per una inspecció de salut i higiene. A més, els controls veterinaris que passen les carns tractades a la planta, es basen en l'evolució de les temperatures sota les quals els aliments han estat exposats. Per tant, s'implementa una subfunció per generar informes de forma automàtica.

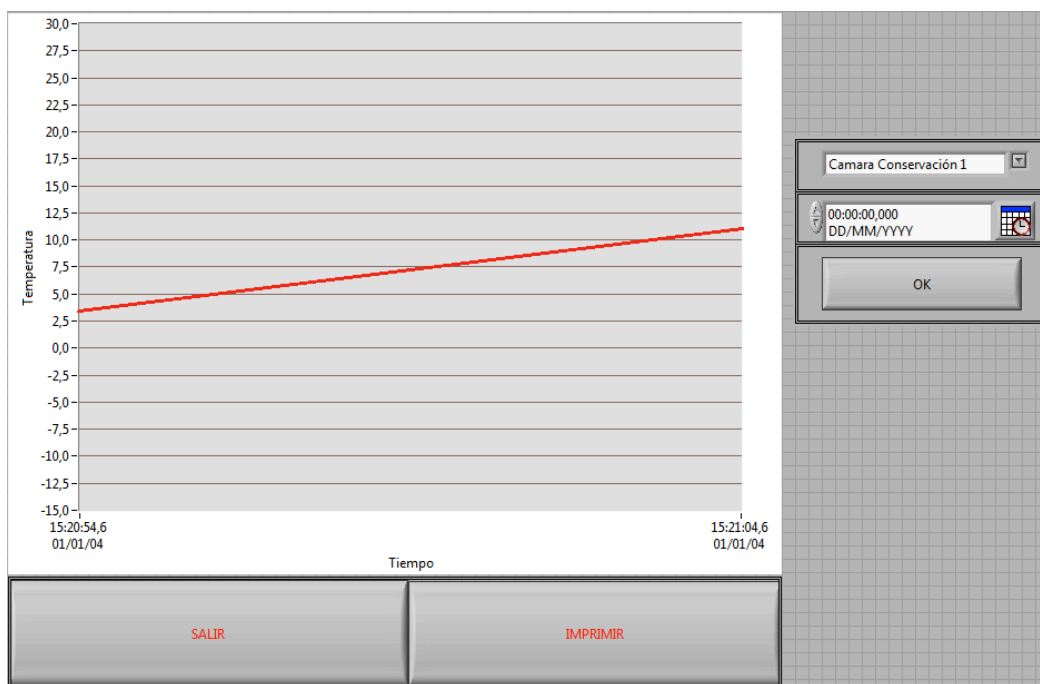


Fig. 3.2.5 Gràfiques de tendència de temperatura en cambra

Per últim, i segons requisits del client, es crea una subfunció de programa que s'encarrega de protegir les pantalles en que es pot manipular algun paràmetre amb contrasenya.

Cal indicar, que totes les pantalles s'han dissenyat amb un format semblant a les existents en l'antic sistema Scada, així que l'adaptació dels operaris al nou sistema no ha de ser gens traumàtica. Es manté la mecànica utilitzada els últims anys en la intervenció de la planta.

3.2.3. Gestió de les comunicacions

3.2.3.1. Inicialització de dades amb PLC PS316

Gràcies a la utilització de Labview, la nostra tasca a l'hora de preparar les comunicacions amb el PLC resultaran bastant simples, sempre i quan tinguem clars els registres definits a la taula d'assignació del Lance-E. A més, aprofitarem les subfuncions MODBUS implementades en el software de National Instruments.

Adaptarem per tant, el VI "Modbus TCP Master Query" per establir la connexió TCP a través del equip interface Lance-E. La subfunció resultant desenvolupada s'anomenarà "Inicializa variables PLC.vi".

En aquesta subfunció de programa, primerament iniciarem la sessió TCP amb el Lance-E. Com a paràmetres, haurem de senyalar la direcció IP del dispositiu Interface i el port remot. Els valors introduïts al software de l'SCADA hauran de coincidir amb els carregats al Lance-E a través de cable sèrie.

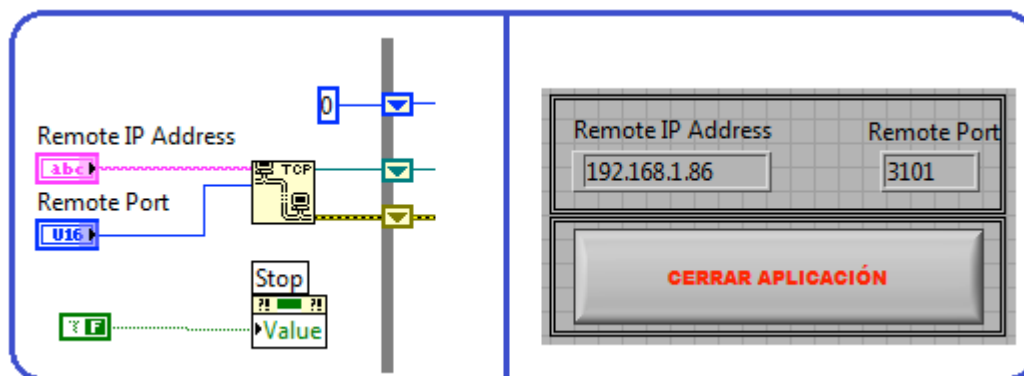


Fig. 3.2.6 Configuració de connexió TCP

Un cop s'estableix connexió TCP amb el Lance-E procedirem a llegir les dades que ens interessin a través de MODBUS. La conjunció MODBUS y TCP serà interpretada pel Lance-E que acabarà traduint les peticions al PLC a través de RS485.

Arribats a aquest punt, farem peticions de lectura de tots els registres definits en la taula d'assignació Lance-E. Totes les respostes rebudes es guardaran en les variables globals i s'actualitzaran tots els valors en les pantalles definides. Les peticions segons la classe de dades existents, seran del tipus:

- **Read discrete inputs:** Bits del tipus I0.0
- **Read input registers:** Paraules del tipus MW1500

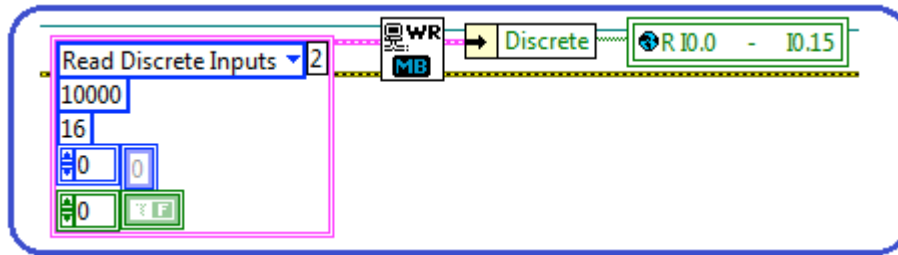


Fig. 3.2.7 Inicialització de variables

Amb aquesta operació (Fig.3.2.7), tindrem actualitzat el valor de cada variable, segons la informació que conté el PLC.

3.2.3.2. Actualització de dades en PLC PS316

La filosofia serà molt semblant que la seguida en l'apartat anterior (3.2.3.1), encara que en aquest cas, es realitzaran peticions d'escriptura en el PLC.

En primera instància, i de igual manera que en la inicialització de variables, iniciarem la sessió TCP amb el dispositiu Lance-E, mantenint els paràmetres (IP, port remot) habituals.

Un cop establerta la connexió TCP, farem una comparació per saber si hem d'escriure una dada o no, compararem la variable global de lectura i la d'escriptura. Amb aquesta operació es controla si el que tenim assignat a la pantalla coincideix amb la variable de lectura, aquesta última actualitzada amb la informació del PLC. En cas de no coincidir, farem una petició d'escriptura. En el que cas en que la variable allotjada al PLC sigui diferent de la mostrada, també s'actualitzarà però aquest cas en el sistema SCADA.

- **Write multiple coils:** Bits del tipus M21.0
- **Write multiple registers:** Paraules del tipus MW1800

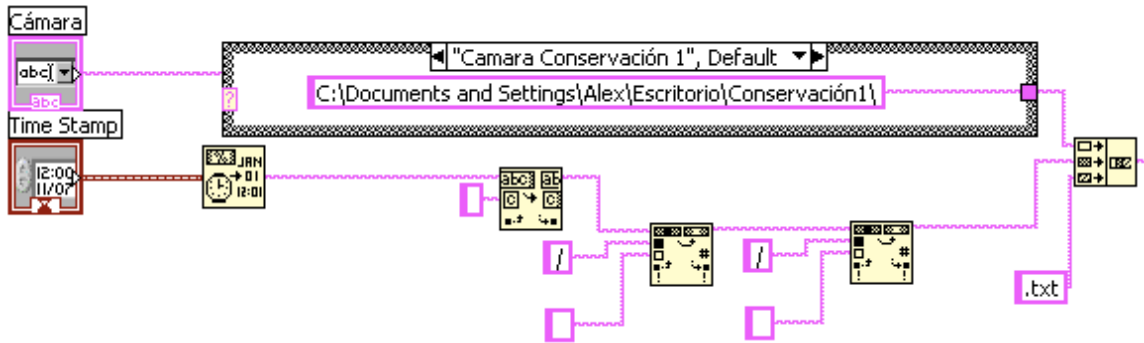


Fig. 3.2.9 Creació de fitxer .txt

En canvi, si l'arxiu ja havia estat creat en la carpeta corresponent, tan sols l'obrirem sempre que calgui refrescar la informació que allotja.

3.2.4.2. Tractament de dades

Un cop gestionats els fitxers on escriurem de forma contínua les temperatures de cadascuna de les cambres, es té en compte el volum de dades en disc.

Amb la missió de minimitzar l'espai destinat a l'emmagatzematge, es decideix guardar les dades en arxius amb format .txt, i l'enregistrament de dades es realitzarà cada cinc minuts. Amb aquesta quantitat de valors, es genera una gràfica amb una bona resolució, on s'evidencia l'evolució de les temperatures.

Per saber l'instant en que hem d'escriure dins de cadascun dels directoris (un per cambra), agafarem l'hora del sistema operatiu en que s'ha instal·lat l'Scada. Desgranarem el senyal horari i comprovarem si ens trobem en el minut "0" o minut "5". Si es compleixen les condicions, escriurem el valor de temperatura i l'hora d'escriptura en el fitxer txt. Aquesta operació serà controlada per la subfunció desenvolupada amb nom "Actualiza 5 minutos.vi".

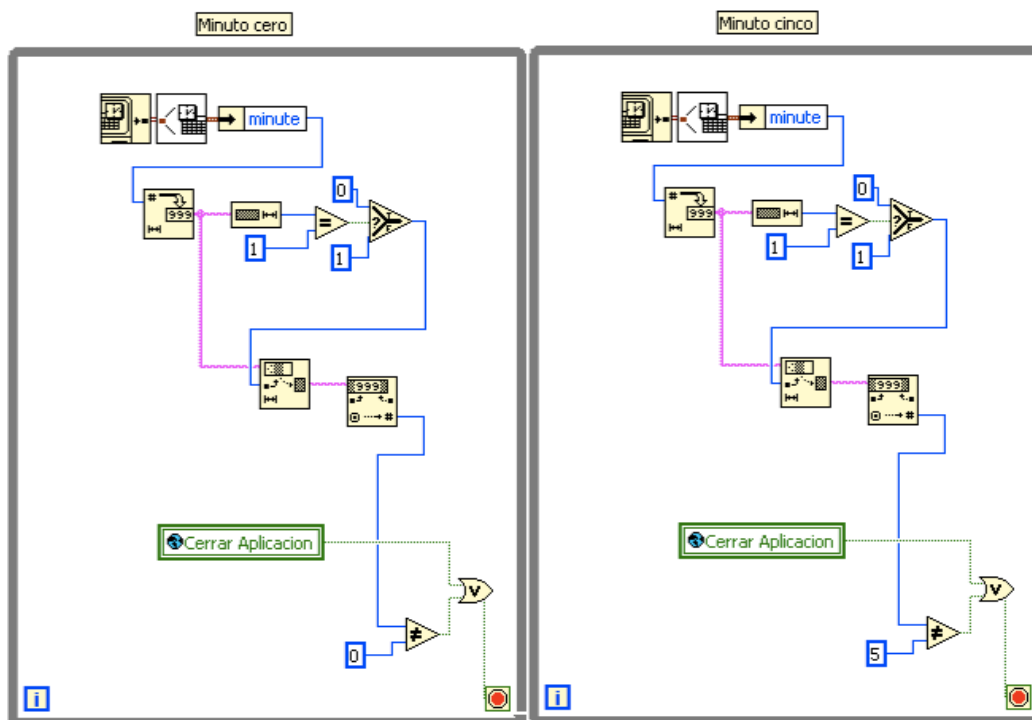


Fig. 3.2.10 Actualització en minut 0 i minut 5

3.2.4.3. Conversió de dades

El tractament de dades en PLC i SCADA es diferent depenent del dispositiu en que ens trobem. Per tant, es crearan diferents subfuncions per traduir les dades al format desitjat.

- **Array to Bool / Bool to Array:**

Amb la subfunció “Array to Bool” seleccionarem les double words (16 bits) llegides directament des del PLC amb la petició de lectura “Read Input Registers”. De cada double word seleccionarem els bits que ens interessin indicant un subíndex comprès entre 0 i 15 (16 bits). Finalment podrem treballar a nivell de bit .

La subfunció “Bool to Array” realitzarà el procés invers i l'utilitzarem per empaquetar els 16 bits en double word, per acabar realitzant una escriptura en el PLC del tipus “Write multiple coils”.

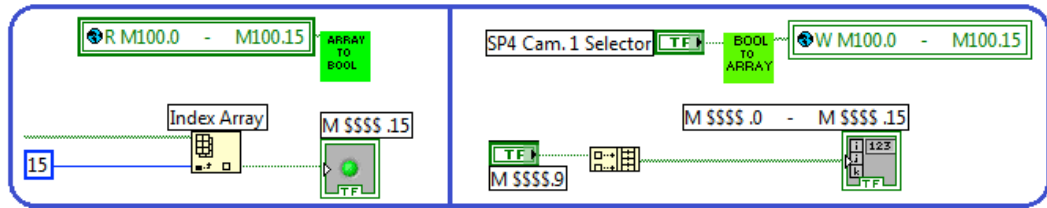


Fig. 3.2.11 Definició funció “Array to Bool / Bool to Array”

- **Register to Array / Array to Register:**

Amb la subfunció “Register to Array” seleccionarem les double words (16 bits) llegides directament des del PLC amb la petició de lectura “Read Input Registers”. De cada double word seleccionarem els bits que ens interessin indicant un subíndex comprès entre 0 i 15 (16 bits).

La subfunció “Array to Register” realitzarà el procés invers i l'utilitzarem per empaquetar els 16 bits en double word, per acabar realitzant una escriptura en el PLC del tipus “Write multiple registers”.

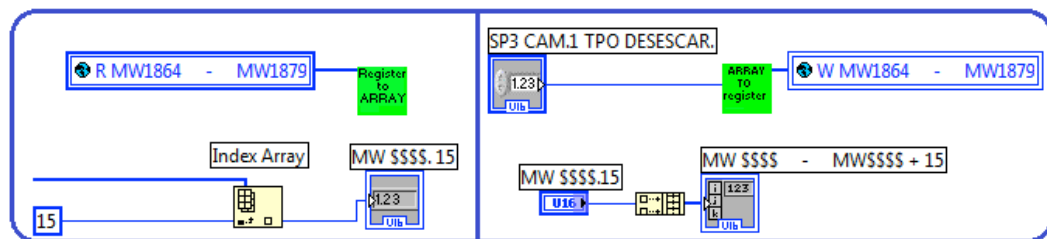


Fig. 3.2.12 Definició funció “Register to Array / Array to Register”

- **Unsigned to Signed / Signed to Unsigned:**

L'última conversió de dades afecta a la gestió dels integers i a nivell d'SCADA, a la lectura i escriptura de les temperatures. El veritable problema recau en les temperatures negatives, ja que la definició del signe a nivell de bit no coincideix amb la seva representació en pantalla. En la taula següent (Fig. 3.2.13) s'observa la representació de diferents temperatures tant en SCADA com en PLC.

Temperatura en SCADA (Signed)	Temperatura en PLC (Unsigned, nivell de bit)
+1	1
0	0
-1	65535

Fig. 3.2.13 Representació de valors Signed/Unsigned

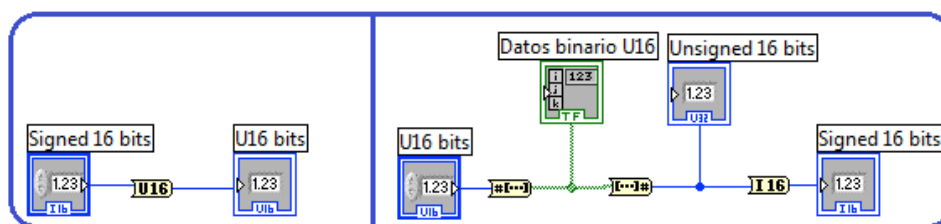


Fig. 3.2.14 Definició funció “Unsigned to Signed / Signed to Unsigned”

3.2.5. Representació de les dades capturades

3.2.5.1. Captura i visualització de temperatures

Després del tractament de dades aplicat, s'emmagatzemarà la informació en el arxiu “txt” tal i com volem que siguin representades les temperatures tant a les gràfiques de control de temperatures com als informes automatitzats.

En primer lloc haurem d'indicar quines dades volem observar, ajudant-nos de la següent aplicació que ens donarà la opció d'escollir alguna de les cambres de la instal·lació i la data on iniciarem la visualització de temperatures. Finalment obtindrem un índex que guardarem en una variables de tipus “string” i presentarà el següent format:

“Directori \ n° de cambra \ DDMMAAAA.txt”

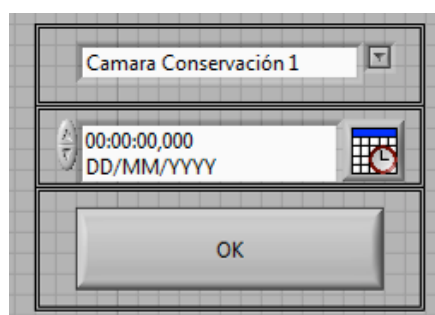


Fig. 3.2.15 Selecció de dades a visualitzar

Aquesta petició de lectura coincidirà amb el nom dels fitxers que s'han anat enregistrant en el nostre terminal SCADA i que contenen l'històric de temperatures.

Seguidament, consultarem les dades que ha demanat l'usuari i definirem els eixos X/Y que dibuixaran les temperatures en funció del temps. El nostre objectiu serà el de mostrar una gràfica amb una representació temporal a l'eix X i el valor de les temperatures sobre l'eix Y.

En la figura 3.2.16 veurem la forma amb que es llegeixen les dades per tal de representar-les sobre un objecte de tipus XY Graph.

En primer lloc es llegirà la variables de tipus string generada amb la petició de lectura de l'usuari. La nostra aplicació SCADA es posicionarà en el directori marcat i farà dos tipus de lectures:

- Per un cantó es llegiran els valors de format integer que corresponen a les temperatures enregistrades durant la jornada seleccionada. Formarem per tant l'eix Y.
- Per una altra banda, farem la lectura temporal emmagatzemada en l'arxiu en qüestió i l'abocarem sobre l'eix X.

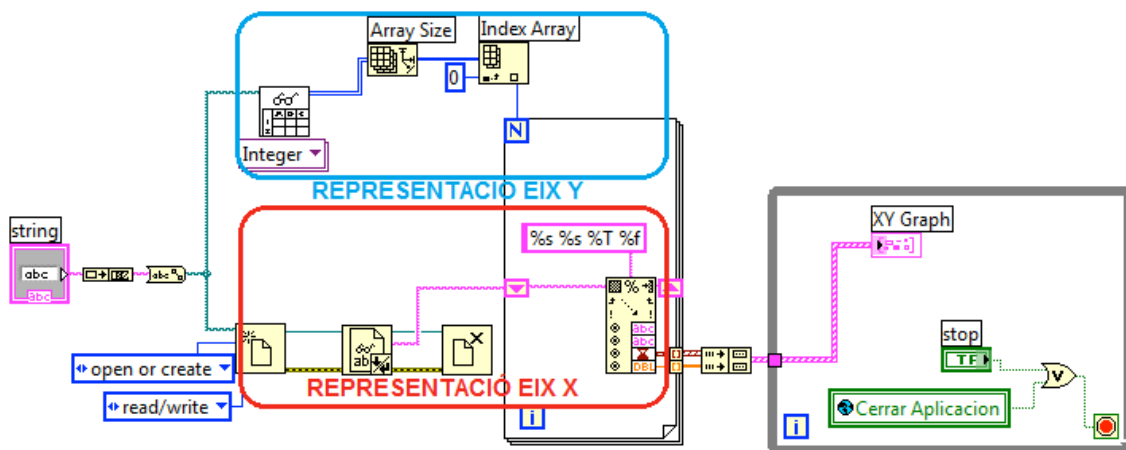


Fig. 3.2.16 Definició d'eixos XY

3.2.5.2. Creació d'informes automatitzats

Una de les premisses del nostre client i una de les motivacions més importants per tal d'emprendre aquest projecte, era el de tenir la capacitat de crear informes automatitzats on es portés un control de la temperatura exhaustiu. Aquesta informació seria utilitzada com a objecte d'anàlisi a nivell veterinar, on es certificaran diferents processos aplicats sobre els productes alimentaris en funció de la estabilitat a nivell tèrmic.

Aquesta informació a més seria de caire confidencial i tan sols un grup d'usuaris en funció de els seus privilegis han de ser capaços d'activar l'autogeneració d'informes.

En la figura 3.2.17 es pot observar un nou subVI on primerament trobem la protecció que afegeix l'aplicació "Password" per tal d'activar la generació del document oficial i obliga a introduir un número secret. Un cop introduït arribarà una senyal a true a la entrada del següent panell per tal d'imprimir el report de temperatures.

Serà en aquest pas on afegirem les dades llegides anteriorment mitjançant la variable local "XYGraph". Com a elements informatius introduïrem diferents texts i logos de l'empresa que servirà com a document certificat.

Un cop executat tot el procés descrit, obtindrem una plana pdf on visualitzarem l'històric de temperatures.

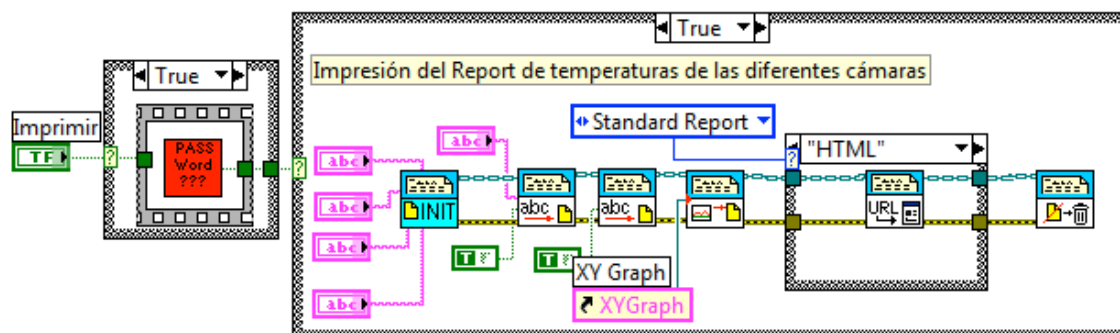


Fig. 3.2.17 Creació d'informes automatitzats

3.2.6. Validació del software desenvolupat

Un cop desenvolupat el software que farà les funcions d'SCADA arriba el torn de la validació amb el client.

Primerament, es realitza un curs de formació on es presenten les novetats que introdueix el nou sistema desenvolupat amb Labview 8.2. Entre elles, destaca la facilitat amb que es poden consultar temperatures de les cambres frigorífiques en un històric, així com la generació d'informes. Cal comentar, que els informes automatitzats van estar supervisats pel tècnic encarregat de fer les inspeccions veterinàries en funció de les temperatures a les que s'han sotmès els aliments.

A nivell de comunicacions, en la primera execució del nou SCADA es van trobar petits "bugs" de programa dins del microprocessador que conté el Lance-E que van ser corregits pel fabricant. Aquestes incongruències, feien que el forçat de l'encesa de les cambres des del sistema SCADA no fos el correcte.

Per últim, es va identificar un problema en les comunicacions entre SCADA i PLC. Passat un temps d'enregistrament de temperatures, el PC que allotjava l'aplicació SCADA es saturava, deixant penjada l'aplicació. Aquestes situacions es van solventar amb la utilització de temporitzadors dins de les estructures "while loop", que generaven una execució esgraonada dels processos definits.

CAPÍTOL 4. IMPACTE MEDIAMBIENTAL

4.1. Estudi del impacte mediambiental

Dins del projecte que consistia en desenvolupar un nou sistema SCADA sobre una instal·lació obsoleta, des d'un bon principi es va treballar en la línia de treball que marcaven els requeriments del client. Això si, la veu del programador sempre va tenir molt de pes sobretot per introduir millores en la instal·lació.

Avui dia, l'impacte que la indústria genera en el medi ambient es un factor important ja que cada cop som mes conscients de la conservació de la natura. Un altre punt jugarà al nostre favor i es que normalment, la reducció d'emissions es tradueix en un estalvi energètic.

A continuació es valorarà l'aportació realitzada per tal de reduir amb els medis existents, l'impacte mediambiental.

4.1.1. Estalvi energètic a nivell software

Tal i com s'ha comentat en apartats anteriors, en el sistema SCADA existeixen diferents pantalles entre les quals, destaquen les de control de temperatura de cadascuna de les cambres. Serà aquí on s'introduirà un nou paràmetre anomenat "temperatura d'histèresi / offset".

En general, l'ajustament ve marcat per una temperatura de consigna, encara que en aquest cas, s'afegirà un nou valor que actuarà com a marge, el valor d'histèresi. Els evaporadors de les cambres no s'engegaran mentre ens trobem dins dels marges establerts. En el PLC mitjançant el software de desenvolupament SUCOSOFT, definirem un nou rang de memòria comprès entre la MW1816.0 i MW1831.0 que allotjarà la suma de la temperatura de consigna més la temperatura d'offset (histèresi).

En el següent exemple, es defineix com a MW1825.0 la temp. de consigna + temp. d'offset:

```

** Evaporador 2 Control temperatura Camara
L MW1801.0          M.Consigna Temp. Evapor.2      Conser.
ADD MW1813.0        M.Offset Temp. Evapor.2      Conser.
= MW1825.0          Consig.Temp.+ Offset Evapor.2    Conser.

```

Fig. 4.1.1 Temp. consigna + temp. d'histèresi en SUCO

Aquesta doble paraula definida que conté 16 bits (de MW1816.0 a MW1831.0), s'haurà de traspasar al sistema SCADA mitjançant el protocol MODBUS. Per tant segons el procediment habitual, definirem un registre en la taula d'assignació allotjada en el dispositiu interface Lance-e per finalment, realitzar la lectura i escriptura de la doble paraula en el software del sistema SCADA desenvolupat en Labview.

La introducció d'aquest nou paràmetre, sempre que treballem amb temperatures haurà de sofrir una transformació de valor integer (amb signe) a unsigned (sense signe). Per últim, totes les temperatures d'histeresi introduïdes s'hauran d'empaquetar en un array tot just per ser enviat al PLC novament.

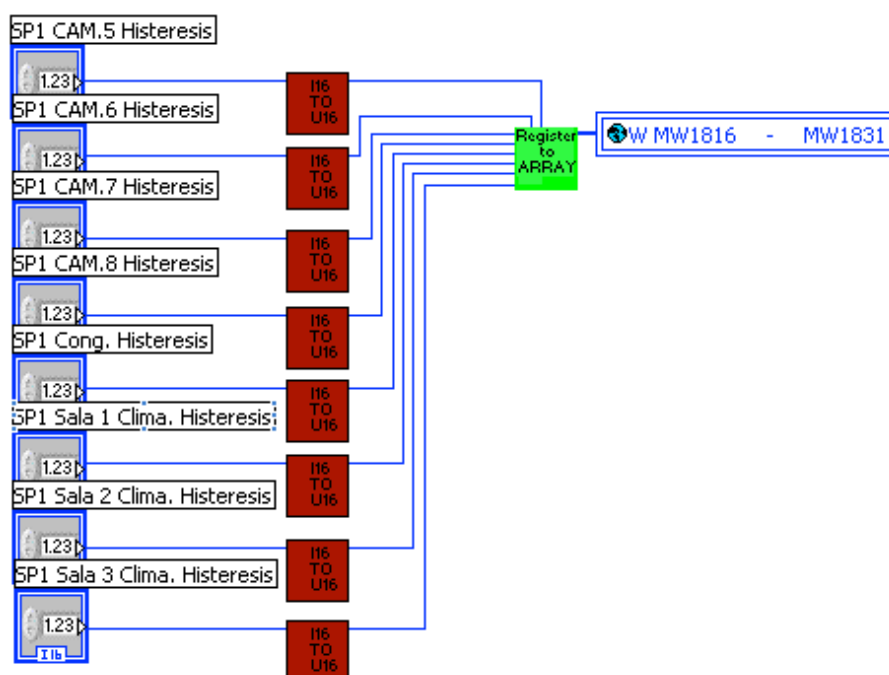


Fig. 4.1.1 Escripció de temperatures d'histeresi en PLC

4.1.2. Aportació a la conservació del medi ambient

A continuació es realitzarà una estimació de la reducció d'emissions de CO₂ aconseguida amb la modificació a nivell software tant en PLC com en sistema SCADA.

Cal comentar que no tenim a disposició les lectures mensuals de consum elèctric encara que es demostrarà, que s'aconsegueix un estalvi a tenir en compte amb valors de consum estimats.

Segons especificacions tècniques de la instal·lació obtenim les següents característiques:

- Volum de cambres frigorífiques: 3500 m^3
- Volum d'avantcambres i passadís: 550 m^3
- Capacitat en servei continu: 30.000 kg
- Temperatura mínima en cambra congelació: -27°C
- Necessitats energètiques diàries:

Els càlculs energètics es realitzen amb una mitjana de 17h de funcionament diàries, segons estadístiques enregistrades.

- Cambres frigorífiques 20 kWh
- Cambres de conservació 13kWh

El consum acumulat pujaria fins als 561kWh diaris, es a dir, uns 145.860 kWh/anuals.

Un cop implementat el canvi de software, el funcionament mitjà de la instal·lació es veu reduït en 55 minuts, el que suposa un consum de 530kWh diaris. Al cap d'un any, el consum acumulat ascendirà a 137.995kWh/anuals.

S'obté per tant, un estalvi energètic aproximat de:

- $145.860 \text{ kWh} - 137.995 \text{ kWh} = 7.865 \text{ kWh anuals}$

La factura elèctrica es veurà reduïda d'aquesta manera:

- $7.865 \text{ kWh} \times 0,08\text{€ /kWh} = 629,2\text{€}$

Per últim, es deixarà d'emetre la següent quantitat de CO^2 a l'atmosfera:

- 3,064 Tones de CO^2

Amb aquestes petites modificacions es demostra que existeix un estalvi un cop s'han implementat els canvis en el control de l'autòmat. No obstant cal comentar, que si traslladem els càlculs a la indústria en la que ens trobem semblaren xifres un tant ridícules.

Amb l'estudi realitzat, demostrarem al client, que amb petites modificacions, l'eficiència energètica de la seva instal·lació anirà millorant fins al punt de suposar un estalvi econòmic important. En el que respecta al impacte mediambiental, es deixan d'alliberar a l'atmosfera més de 3 tones de CO^2 .

CAPÍTOL 5. LÍNIES FUTURES I CONCLUSIONS

5.1.Línies futures

Un cop el projecte ha estat optimitzat per tal de satisfer les necessitats del client, es plantegen diferents línies d'actuació per tal de que la instal·lació s'adapti tècnicament a les noves tecnologies i no quedi pas endarrerida tecnològicament parlant.

Primerament, destacarem el dispositiu més important de la instal·lació, l'autòmat. Avui dia continua instal·lat el Klöckner Moeller PS316, totalment obsolet. La solució provisional per tal de suplir una possible pèrdua de l'equip, ha estat la de trobar un recanvi compatible. Segons el consell que es llança des de l'enginyeria elèctrica, la següent inversió en la instal·lació haurà de ser la de realitzar una migració del PLC i armari elèctric a un sistema actual. Com a proposta s'aconsella instal·lar un nou PLC SIEMENS CPU 315 2DP, adaptable a un gran ventall d'accessoris i perifèria ja que serà 100% compatible tant amb el sistema SCADA com amb el dispositiu interface Lance-E.

Un canvi de PLC tan sols suposaria a nivell de comunicacions, canviar un paràmetre en l'interface Lance-E, ja que aquest equip, presenta total compatibilitat amb multitud de PLCs d'última generació.

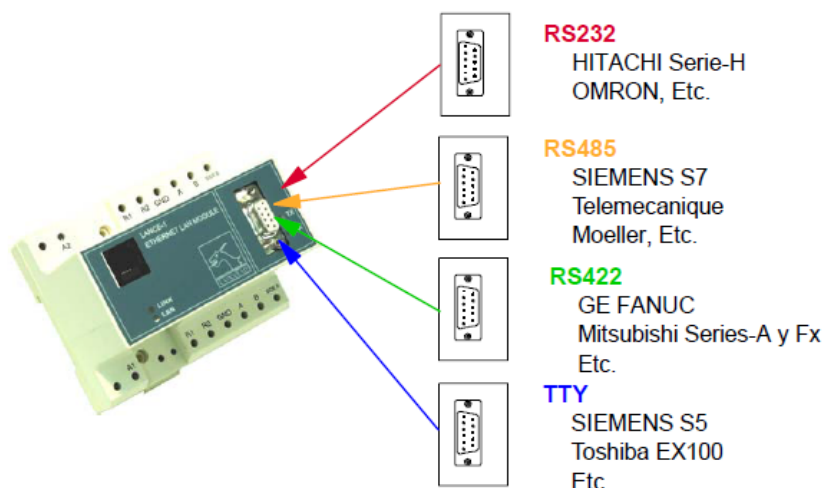


Fig. 5.1.1 Compatibilitat de Lance-E amb diferents PLC

A nivell d'SCADA, sempre i quan no modifiquem la taula d'assignació MODBUS allotjada a l'interface Lance-E, no s'haurà d'implementar cap modificació.

La següent fase de modificacions a realitzar no presenten una criticitat tan elevada com el canvi de PLC, encara que és molt important a nivell econòmic i mediambiental. La següent línia de millores ha d'englobar una sèrie de mesures per millorar l'eficiència energètica de la instal·lació. Primerament s'haurà de realitzar una auditoria energètica per part d'una empresa especialitzada.

Amb la presa de dades tant elèctriques com termogràfiques, es buscaran les millors mesures encara que la més efectiva sens dubte constarà en substituir els actuals equips encarregats de la refrigeració de les cambres frigorífiques, per evaporadors i compressors d'eficiència energètica A+. Amb aquesta mesura es reduiran tant les emissions com la factura elèctrica en un 30% aproximadament. Cal comentar, que una bona programació en la gestió de la planta, estudiant una ocupació eficient de les cambres frigorífiques, aportarà millores importants en el rendiment de la instal·lació.

Per últim, es proposarà configurar el sistema SCADA en un escenari remot, accessible a través d'Internet des de qualsevol emplaçament. El dispositiu Lance-E presenta la capacitat d'exercir com a servidor de dades Ethernet. Amb la utilització conjunta d'un servidor OPC, serem capaços de traslladar l'SCADA a qualsevol PC amb connexió a Internet. Cada cop es més important controlar la planta industrial des de l'exterior.

5.2. Conclusions

En aquest treball de final de carrera, s'ha mostrat la migració d'un sistema SCADA totalment obsolet a un altre integrat en el mateix escenari encara que desenvolupat amb eines d'última generació.

Un dels reptes més importants per a tots, era el d'aconseguir integrar el nou sistema de visualització, en un entorn de treballadors molt acostumat a l'antiga aplicació. A més, l'avaluació econòmica del projecte sempre va estar molt ajustada i es van haver de trobar solucions d'ingeni i recerca de recursos disponibles.

Primerament, s'ha demostrat que el protocol de comunicacions Modbus està molt estès en el mercat i multitud de dispositius el tenen implementat. A més, es un tret destacat la seva classificació com a software lliure, ja que aquesta situació es tradueix directament en un estalvi econòmic. L'endinsament en aquest protocol ens ha mostrat les grans virtuts que aporta i l'alta eficiència amb que es va concebre.

Seguidament apareix la gran troballa en aquest projecte, el dispositiu interface Lance-E. Segurament, aquesta és la peça que encaixa totes les parts del projecte, i sembla realment increïble, que l'empresa desenvolupadora

(Lancelan- La Factoría de Comunicaciones Aplicadas, S.L.) hagi donat suport a un PLC totalment obsolet avui dia, com es el que s'ha trobat en aquest projecte, el Klöckner Moeller PS316. El fet a més de que implementés MODBUS TCP a través d'ethernet, ens ha facilitat les comunicacions amb el sistema SCADA Labview.

Arribats a aquest punt ens faltava desenvolupar un software per al sistema SCADA assequible de cara al client final. Es aquí on neix la idea de Labview. National Instruments aporta un gran avantatge, i es que tan sols el desenvolupador requereix d'una llicència professional. El client no haurà de comprar cap tipus de llicència per executar el software. Un nou factor que jugarà en favor del preu final. Cal comentar a més, que Labview integra perfectament les comunicacions MODBUS TCP, fet que representa una compatibilitat total amb el dispositiu Interface Lance-E. A més, la programació gràfica amb que s'ha ideat Labview, mostra la facilitat amb que es poden desenvolupar aplicacions realment complexes.

Un dels temors mes grans durant l'avantprojecte, fou la resposta del PLC front a comunicacions tant avançades. Després d'uns petits ajustos al microprocessador que conté el Lance-E, el Klöckner Moeller es va poder comunicar sense problemes degut en gran mesura, al gran control de flux de dades que realitza precisament l'interface.

Per últim el procés d'adaptació dels empleats envers el nou sistema va transcorre sense problemes ja que un dels treballs més satisfactoris, va ser la complexa adaptació dels antics processos sobre una plataforma d'última generació.

BIBLIOGRAFIA

- 1 MODBUS RTU OVERVIEW. *Pàgina principal*. [en línia]. Disponible a:
< <http://www.rtaautomation.com/modbusrtu>>
- 2 Description of Modbus protocol [en línia]. Disponible a:
< <http://www.jaec.info/Home%20Automation/Protocols-buses-house/Modbus-Protocol/modbus-protocol.php> >
- 3 UNDERSTANDING THE MODBUS PROTOCOL [en línia]. Disponible a:
< <http://jamod.sourceforge.net/kbase/protocol.html>>
- 4 Klöckner moeller ps316 support [en línia]. Disponible a:
<http://downloads.applied-automation.com/sam/moeller/ps316/index.html>
- 5 SUCOS S30 [en línia]. Disponible a:
<<http://usuaris.tinet.cat/lcc/plc/sucos.htm#caracteristicas>>
- 6 Klöckner moeller ps316 modular plc controller [en línia]. Disponible a:
<<http://www.klocknermoeller.com/automation/ps316.htm>>
- 7 MODBUS APPLICATION PROTOCOL SPECIFICATION:
<<http://www.Modbus-IDA.org>>
- 8 Modbus_over_serial_line_V1_02.pdf:
- 9 Manual de usuario Lance-e. [en línia]. Disponible a:
<<http://www.lancelan.com>>
- 10 RS485 data cable coupling ps3-ps3.pdf:
- 11 CeroCO2 Iniciativa para el cuidado del clima [en línia]. Disponible a:
< <http://www.ceroco2.org/calcular/default.aspx>>