



Universitat de Girona  
Escola Politècnica Superior

## Projecte/Treball Final de Carrera

**Estudi:** Enginyeria Tècn. Ind. Electrònica Ind. Pla 1994

**Títol:** CREACIÓ D'UNA XARXA DE COMUNICACIONS  
MODBUS PELS AUTÒMATS TSX Micro

**Document :** 1 Memòria

**Alumne:** MARTA PÉREZ ALBIOL

**Director/Tutor:** ALBERT FIGUERAS  
**Departament:** Electrònica, Informàtica i Automàtica  
**Àrea:** ESA

**Convocatòria:** Gener 2006

1.	INTRODUCCIÓ .....	5
2.	DESCRIPCIÓ DE LA INSTAL·LACIÓ .....	6
2.1	Antecedents .....	6
2.2	Maqueta didàctica .....	6
2.3	Autòmat TSX 3705 .....	8
2.4	Software PL7 .....	10
3.	COMUNICACIÓ MODBUS.....	11
3.1	Introducció.....	11
3.1.1	Busos de camp .....	11
3.1.2	Bus Uni-telway .....	11
3.1.3	Bus Modbus .....	12
3.2	Operacions en xarxes Modbus.....	12
3.3	Modes de transmissió sèrie.....	14
3.3.1	Mode ASCII.....	14
3.3.2	Mode RTU.....	15
3.4	Estructura del missatge.....	15
3.4.1	Estructura ASCII .....	16
3.4.2	Estructura RTU .....	16
3.4.3	Camp de direcció .....	17
3.4.4	Camp de Funció.....	17
3.4.5	Camp de Dades .....	18
3.4.6	Camp de Comprovació d'Error.....	18
3.4.7	Transmissió en sèrie de caràcters .....	19
3.4.8	Mètodes de comprovació d'error.....	20
3.4.8.1	Control de paritat.....	20
3.4.8.2	Comprovació de la trama LRC .....	21
3.4.8.3	Comprovació de la trama CRC .....	21
3.5	Descripció de les funcions principals .....	22
3.5.1	Llegir N bits .....	22
3.5.2	Llegir N registres .....	23
3.5.3	Assignar un bit .....	24
3.5.4	Assignar un registre .....	24
3.5.5	Llegir el byte d'estat .....	25
4.	XARXA DE COMUNICACIONS .....	26
4.1	Autòmats TSX3705 / TSX3722 .....	26
4.2	Unitat TSX PACC 01 .....	29
4.3	Muntatge de la xarxa.....	31

4.3.1	Connexionat de les experiències .....	31
5.	DESCRIPCIÓ SOFTWARE PL7- Pro.....	33
5.1	Introducció.....	33
5.2	Direccionament dels autòmats .....	33
5.2.1	Primera programació dels autòmats d'una xarxa.....	33
5.2.2	Tipus de direccionament .....	34
5.3	Connexió dels autòmats.....	35
5.3.1	Mode local.....	35
5.3.2	Mode connectat.....	35
5.3.3	Configuració mòdul de comunicació .....	35
5.4	Funcions de comunicació.....	38
5.4.1	Estructura general de les funcions de comunicació.....	38
5.4.2	Les funcions de comunicació en Modbus .....	39
5.4.3	Direcció destinatària.....	39
5.4.4	Paràmetres de gestió .....	39
5.4.4.1	Confirmació de comunicació i d'operació.....	40
5.4.4.2	Longitud i temps d'espera .....	42
5.4.5	Límits de les funcions de comunicació.....	43
5.4.6	Descripció funcions bàsiques de comunicació.....	43
5.4.6.1	Lectura d'objectes estàndard ( READ_VAR ) .....	43
5.4.6.2	Esriptura d'objectes estàndard ( WRITE_VAR ).....	45
6.	EXPLICACIÓ DE LA PRÀCTICA .....	48
6.1	Connexionat de l'experiència .....	48
6.2	Descripció general de la pràctica d'exemple .....	50
6.3	Emulació amb la maqueta.....	51
6.4	Programa autòmat mestre.....	53
6.4.1	Llistat d'entrades i sortides.....	53
6.4.2	Descripció .....	54
6.4.3	Programa .....	56
6.4.3.1	Secció Chart.....	56
6.4.3.2	Secció Post .....	59
6.4.3.3	Sr0 : Inicialització de variables .....	68
6.4.3.4	Sr1: Activació de bits segons taula de prioritats.....	70
6.4.3.5	Sr2 : Inicialització de variables .....	71
6.5	Programa autòmat esclau .....	72
6.5.1	Llista d'entrades i sortides.....	72
6.5.2	Descripció .....	73

6.5.3	Programa .....	74
6.5.3.1	Secció Chart.....	74
6.5.3.2	Secció Post .....	75
7.	RESUM DEL PRESSUPOST .....	81
8.	CONCLUSIONS .....	82
9.	RELACIÓ DE DOCUMENTS.....	83
10.	BIBLIOGRAFIA .....	84
	ANNEX A. MANUAL DE PRÀCTIQUES.....	85
A1.	INICIACIÓ A LA COMUNICACIÓ MODBUS .....	85
A1.1	Introducció.....	85
A1.2	Bus Modbus .....	85
A1.2.1	Direccionament .....	86
A1.3	Autòmats TSX 3705 / TSX 3722 .....	87
A1.4	Unitat TSX PACC 01 .....	88
A1.5	Funcions bàsiques de comunicació .....	90
A1.5.1	Funcions de comunicació en Modbus.....	90
A1.5.2	Funció de comunicació READ_VAR.....	91
A1.5.3	Sintaxi de la funció READ_VAR.....	91
A1.5.4	Funció de comunicació WRITE_VAR .....	95
A1.5.5	Sintaxi de la funció WRITE_VAR .....	95
A1.6	Descripció de la maqueta didàctica .....	100
A1.7	Connexionat de les experiències .....	102
A2.	EXPERIÈNCIA 0. EXEMPLE .....	103
A2.1	Descripció de la pràctica .....	103
A2.2	Emulació amb la maqueta.....	104
A2.3	Realització de la pràctica .....	106
A2.3.1	Pantalla de configuració de Modbus .....	107
A2.4	Programa autòmat mestre .....	110
A2.4.1	Llistat d'entrades i sortides.....	110
A2.4.2	Descripció .....	111
A2.4.3	Programa .....	112
A2.4.3.1	Secció Chart.....	113
A2.4.3.2	Secció Post .....	116
A2.4.3.3	Sr0: Inicialització de variables .....	125
A2.4.3.4	Sr1: Activació de bits segons taula de prioritats.....	127
A2.4.3.5	Sr2 : Inicialització de variables .....	128
A2.5	Programa autòmat esclau .....	129

Creació d'una xarxa de comunicacions modbus pels autòmats telemecanique TsxMicro

A2.5.1	Llista d'entrades i sortides.....	129
A2.5.2	Descripció .....	130
A2.5.3	Programa .....	131
A2.5.3.1	Secció Chart.....	131
A2.5.3.2	Secció Post .....	132
A3.	QÜESTIONARI DE COMPRESIÓ.....	138
A4.	EXPERIÈNCIA 1. TASQUES A REALITZAR PER L'ALUMNE .....	138
A4.1	Tasca 1 .....	138
A4.2	Tasca 2 .....	139
A4.3	Tasca 3 .....	139

## **1. INTRODUCCIÓ**

L'objecte del projecte és el de crear una xarxa de comunicacions Modbus. La xarxa unirà els autòmats del laboratori de regulació Telemecanique TSX Micro (els deu autòmats TSX 37-05). Com a complement a la xarxa, es prepararà un manual de pràctiques que contindrà dues practiques. Una de les experiències servirà com a exemple i guia per l'alumne. L'altre experiència proposarà unes tasques amb la finalitat de que els alumnes puguin conèixer la connexió Modbus entre autòmats de forma pràctica. També es farà un breu qüestionari, el qual permetrà al professor saber la comprensió real de la teoria proporcionada (que és necessària perquè l'estudiant pugui realitzar les tasques assenyalades).

La xarxa es muntarà mitjançant els autòmats Telemecanique existents al laboratori de regulació (TSX 37-05), l'autòmat TSX 3722 de nova adquisició, les unitats d'aïllament TSX P ACC 01 (se n'han hagut d'adquirir 10 unitats), i els cables corresponents (de doble parell trenat blindat).

Quan s'hagi realitzat el muntatge físic de la xarxa, caldrà implementar el programa que pugui servir com a experiència de la comunicació Modbus.

L'experiència que haurà de realitzar l'alumne, es valdrà de la maqueta didàctica i del PLC, i el que hauran de fer és seguir el manual de la pràctica i realitzar la programació del PLC quan se'ls hi demani. Aquesta experiència, però, només és introductòria, és a dir, l'experiència no serà de gran dificultat i l'alumne no adquirirà grans coneixements, simplement agafarà nocions del funcionament d'una comunicació entre autòmats, en el cas concret d'una xarxa Modbus.

El manual de pràctiques contindrà una experiència de mostra que simularà el procés de buidat i neteja de contenidors. Aquesta simulació estarà explicada de manera didàctica, essent un exemple de comunicació Modbus.

## 2. DESCRIPCIÓ DE LA INSTAL·LACIÓ

### 2.1 Antecedents

Al laboratori de Regulació, fins ara s'hi han realitzat pràctiques de regulació automàtica, d'automatització industrial, control per ordinador, etc.

Per tal de dur a terme aquestes pràctiques, el laboratori de regulació disposava de:

Unitats	Equipament
10	PC's
10	Autòmats Telemecanique TSX 3705
10	Autòmats OMRON CQM1
10	Maquetes didàctiques
1	PL7 Pro V3.4

Taula 1. Antecedents

Tot aquest equipament està distribuït en deu llocs de treball, on cada zona de treball consta d'un ordinador, d'un autòmat Telemecanique TSX 3705, d'un autòmat OMRON CQM1 i una maqueta didàctica. En un d'aquests deu llocs de treball, en comptes de l'autòmat TSX 3705, s'hi ha instal·lat l'autòmat TSX 3722, que permet realitzar les mateixes funcions que el TSX 3705 i utilitzar-se com a tal (ja que és un autòmat d'una gamma més alta que el TSX 3705).

### 2.2 Maqueta didàctica

Com que la universitat no disposa de gaires recursos per a la docència, els alumnes han de realitzar les pràctiques usant eines de simulació, i la maqueta didàctica és un complement d'aquestes eines de simulació del laboratori de regulació.

La maqueta consta d'un disc que es pot fer rodar amb un moviment màxim d'aproximadament 180°. Un motor de corrent continu és el que permet aquest moviment, podent escollir si ha de ser "a dretes", "a esquerres", "a velocitat lenta", o bé, "a velocitat ràpida".

Existeixen dos finals de cursa, Fc1 i Fc2 que limiten els moviments a esquerres i a dretes respectivament. El disc conté 20 peces metàl·liques numerades que es detectaran amb el sensor de proximitat Fc3.

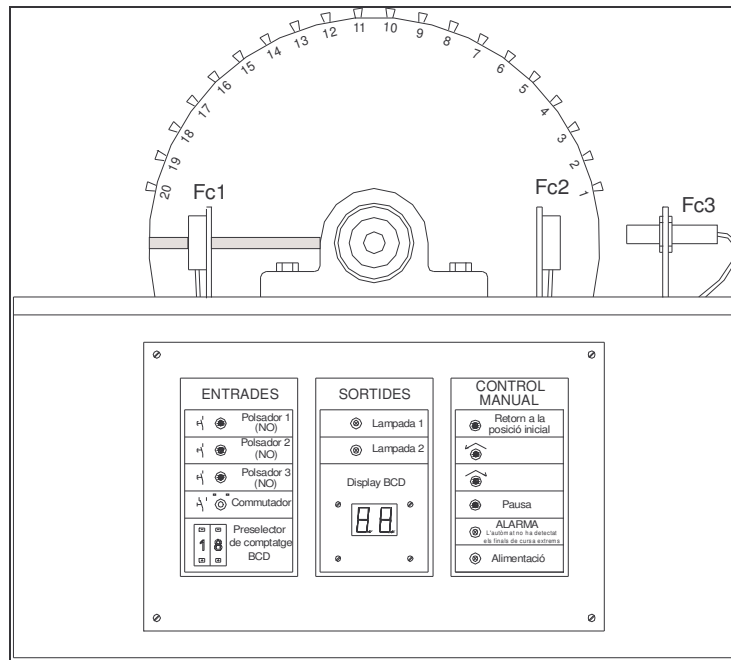


Figura 1. Dibuix de la maqueta didàctica

ENTRADA PLC	ETIQUETA	ELEMENT MAQUETA
0	Ps1	pulsador 1
1	Ps2	pulsador 2
2	Ps3	pulsador 3
3	C_e	commutador posicionat a l'esquerra
4	C_d	commutador posicionat a la dreta
5	Fc1	fi de cursa indica posició esquerra disc
6	Fc2	fi de cursa indica posició dreta disc
7	Fc3	sensor inductiu de posició pel comptatge
8		preselector BCD de unitats, pes 1
9		preselector BCD de unitats, pes 2
10		preselector BCD de unitats, pes 4
11		preselector BCD de unitats, pes 8
12		preselector BCD de desenes, pes 1
13		preselector BCD de desenes, pes 2
14		preselector BCD de desenes, pes 4
15		preselector BCD de desenes, pes 8

Taula 2. Entrades de la maqueta



SORTIDA PLC	ETIQUETA	ELEMENT MAQUETA
0	Drt	moviment a dretes
1	Esq	moviment a esquerres
2	Lnt	velocitat lenta
3	Rpt	velocitat ràpida
4	Lp1	làmpada 1
5	Lp2	làmpada 2
6		display BCD de unitats, pes 1
7		display BCD de unitats, pes 2
8		display BCD de unitats, pes 4
9		display BCD de unitats, pes 8
10		display BCD de desenes, pes 1
11		display BCD de desenes, pes 2
12		display BCD de desenes, pes 4
13		display BCD de desenes, pes 8

Taula 3. Sortides de la maqueta

La maqueta i el PLC s'uneixen mitjançant un cable de connexió.

Els comandaments del moviment del disc utilitzant els pulsadors CONTROL MANUAL té preferència sobre les ordres que l'autòmat estigui donant pel control del moviment del disc. La maqueta disposa del pulsador de control manual anomenat "retorn a la posició inicial" que permetrà a l'alumne posicionar el disc a l'esquerra, ja que les experiències tindran aquesta posició com a inicial.

### 2.3 Autòmat TSX 3705

Els autòmats existents al laboratori de regulació són els autòmats Telemecanique Micro TSX 3705, com ja s'havia dit anteriorment.

L'autòmat TSX 3705 té un mòdul de 28 entrades /sortides (16 E + 12 S) situat en el primer espai i disposa de dos espais més que permeten incorporar un mòdul de format estàndard o dos mòduls de mig format.

La capacitat màxima d'entrades / sortides és de 92 E/S TON (tot o res), instal·lant en l'espai disponible un mòdul de 64 E/S TON amb connexió mitjançant el connector HE10.



Figura 2. Autòmat TSX 3705

Els autòmats del laboratori de regulació estan equipats amb un mòdul mixt d'entrades/sortides tot o res, TSX DMZ 28DR.

Aquest mòdul inclou 28 entrades/sortides repartides en 16 entrades digitals de 24 VCC, de lògica positiva tipus 1 o lògica negativa i 12 sortides de relés.

Pel que fa a les comunicacions, el controlador TSX 3705 disposa de sèrie d'un connector terminal TER. Aquest connector és un enllaç RS485 no aïllat, format per un connector mini DIN de 8 punts. Aquest terminal es fa servir per comunicar l'autòmat amb l'ordinador mitjançant un cable convertidor RS485/RS232, i també permet connectar-hi altres equips, ja siguin autoalimentats (autòmats programables, ...) o no autoalimentats (unitats TSX PACC 01).

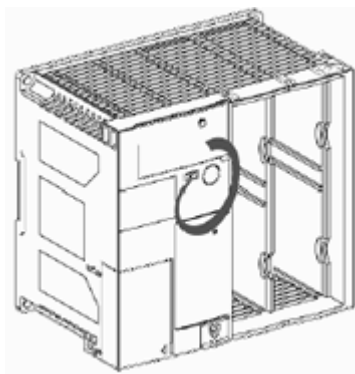


Figura 3. Connector TER

Aquest port de comunicacions es pot configurar com a Uni-telway o com a Modbus mestre o esclau.

## **2.4 Software PL7**

El software de programació PL7 permet realitzar aplicacions en els Modicon TSX Micro i/o Premium en cinc llenguatges diferents: Ladder, text estructurat, graficet, llista d'instruccions, i DFB (Diagram Function Block).

Està creat per ambient Windows i amb capacitat de multi-instància, és a dir, pot obrir-se més d'una vegada per comunicar-nos amb més d'un PLC i programar-los.

La versió del programa existent al laboratori de regulació era la V 3.4 i s'ha actualitzat a la versió V4.4 per tal de poder realitzar la comunicació Modbus (només s'ha actualitzat en el PC de l'autòmat TSX 3722, ja que en els altres PC's no era necessari).

### **3. COMUNICACIÓ MODBUS**

#### **3.1 Introducció**

##### **3.1.1 Busos de camp**

Amb la tecnologia dels busos de camp es permet la comunicació bidireccional entre els dispositius de camp i els sistemes de control, o bé, entre els propis dispositius de camp.

La falta d'estàndards ha portat a les diferents companyies a desenvolupar diferents solucions, cada una d'elles amb diferents prestacions i camps d'aplicació. Així doncs, existeixen diversos tipus de busos de camp estandaritzats, com per exemple: Profibus (alemany), Interbus (empresa Phoenix Contact), FIP-WorldFIP (francès), Lonworks (empresa Echelon - californiana), AS-I (Siemens), Ethernet, Modbus (Modicon), etc. En el cas concret de Telemecanique s'usen els busos UNI-TELWAY, FIPIO, FIPWAY, ETHWAY, MODBUS, ETHERNET.

Els busos de camp tenen configurades 4 funcions. Les tres primeres funcions són a nivell físic, nivell d'enllaç i nivell de xarxa, i la quarta funció és a nivell d'aplicació.

El nivell físic defineix per quin mitjà físic es transmet la informació entre dos sistemes.

El nivell d'enllaç defineix quin és el sistema utilitzat per transmetre dades de forma organitzada amb detecció i correcció d'errors.

El nivell xarxa defineix el mecanisme de direccionament que permet encaminar les dades entre els productes i automàticament els missatges de forma transparent.

El nivell d'aplicació defineix les funcions que permetran fer la transmissió de la informació.

##### **3.1.2 Bus Uni-telway**

La comunicació mitjançant Uni-telway permet intercanviar dades entre tots els equips connectats al bus. Es basa en una estructura jerarquitzada (un mestre i varis esclaus), on el mestre és qui gestiona el bus.

Aquest tipus de comunicació permet que s'enviïn missatges entre mestre i esclau (de mestre a esclau, d'esclau a mestre, o d'esclau a esclau).

El bus Uni-telway és el que tenen per defecte els autòmats TSX MICRO, i és el que s'usa normalment per comunicar l'autòmat amb el PC.

### **3.1.3 Bus Modbus**

El protocol Modbus proporciona un sistema estàndard de comunicació, independentment de la marca de l'autòmat.

La comunicació Modbus permet l'intercanvi de dades entre els equips connectats al bus. El protocol Modbus defineix una estructura de missatge que els controladors reconeixeran i usaran, independentment del tipus de xarxes sobre la que comuniquin. Descriu el procés que usa un controlador per demanar accés a un altre dispositiu, respon a les peticions des d'altres dispositius, i és detecten i es notifiquen els errors que puguin haver-hi. Estableix un format comú per la disposició i contingut dels camps de missatge.

Durant la comunicació en Modbus, el protocol determina com cada controlador coneixerà la seva direcció de dispositiu, reconeixerà un missatge direccionat a ell, determinarà el tipus d'acció a prendre i extraurà qualsevol dada o qualsevol altra informació continguda en el missatge. Si cal una resposta, el controlador construirà el missatge de resposta i l'enviarà usant el protocol Modbus.

La comunicació entre el mestre i els esclaus es pot realitzar en dues formes:

Una de les maneres és que el mestre envia dades a l'esclau i espera que aquest li envii una resposta.

L'altre és que el mestre fa una difusió general, és a dir, envia informació a tots els esclaus sense esperar resposta.

### **3.2 Operacions en xarxes Modbus**

Els ports standard Modbus en controladors Modicon usen una interfície sèrie compatible RS-232. La norma EIA RS-232 defineix patilles del connector, cablejat, nivells de senyal, velocitats de transmissió i control de paritat.

En el cas concret dels PLC's del laboratori de regulació, per tal de poder comunicar-los amb els PC's, cal un cable de comunicació que proporciona la mateixa casa Modicon en adquirir els controladors programables. Això és degut a que el PLC es comunica mitjançant RS-485, i el PC mitjançant RS-232, per tant, cal un convertidor de RS -232 a RS-485 (que ja va incorporat en el cable de comunicació proporcionat).

El controladors poden connectar-se directament en xarxa o via mòdem.

El protocol Modbus crea una estructura jeràrquica consistent en un mestre (que és qui gestiona tots els intercanvis de dades) i un o varis esclaus.

És el mestre l'únic que pot iniciar les operacions . Els altres dispositius (els esclaus) responen subministrant al mestre la dada sol·licitada, o realitzant l'acció demanada pel mestre.

El mestre pot direccionar esclaus individualment o pot generar un missatge per difondre a tots els esclaus. Els esclaus tornen el missatge, és a dir, donen una resposta a les peticions que se'ls ha direccionat individualment, però no es tornen respostes a les peticions fetes en un missatge comú.

El protocol Modbus estableix el format per la petició del mestre, indicant en la petició la direcció del dispositiu esclau ('0' en cas de ser un missatge per a tots els esclaus), un codi de funció que defineix l'acció sol·licitada, qualsevol dada que s'hagi d'enviar i un camp de comprovació d'error.

El missatge de resposta de l'esclau també està definit pel protocol Modbus. Conté camps confirmant l'acció presa, qualsevol dada que s'hagi de tornar i un camp de comprovació d'error. Si el missatge rebut per l'esclau és defectuós o l'esclau és incapaç de realitzar l'acció sol·licitada, construirà un missatge d'error i l'enviarà com a resposta.

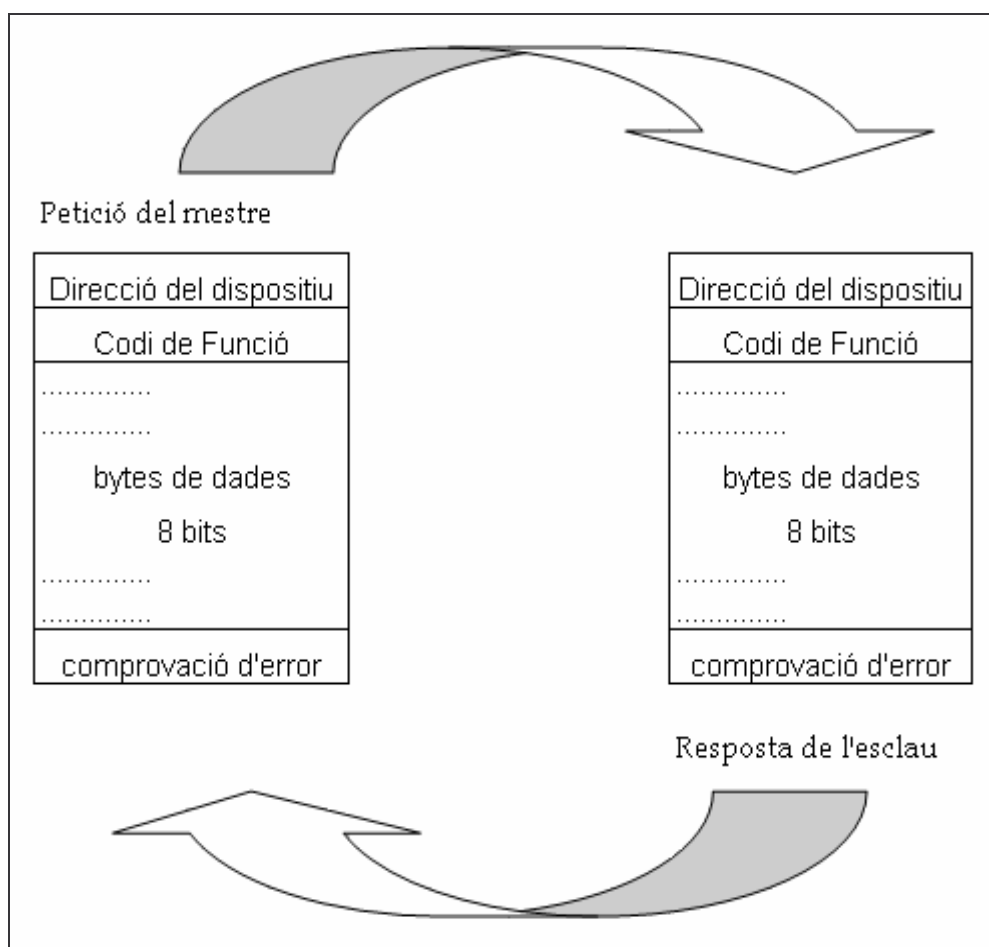


Figura 4. Format general d'una trama

### 3.3 Modes de transmissió sèrie

El mode de transmissió és l'estructura de les unitats d'informació contingudes en un missatge.

El protocol Modbus defineix dos modes de transmissió: ASCII (American Standard Code for Information Interchange) i RTU (Remote Terminal Unit).

Els modes de transmissió ASCII i RTU defineixen els bits que contenen els camps del missatge emès en la xarxa, determinen com s'ha d'empaquetar i descodificar la informació que conté cada camp del missatge.

En una mateixa xarxa de dispositius connectats mitjançant el protocol MODBUS no es poden usar diferents modes de transmissió.

Els usuaris, durant la configuració de cada controlador, escolliran el mode de transmissió i els paràmetres de comunicació del port sèrie (velocitat, paritat, etc), però cal recordar que hauran de ser els mateixos per tots els controladors d'una mateixa xarxa Modbus.

#### 3.3.1 Mode ASCII

Quan l'usuari configura el controlador en mode ASCII per realitzar una comunicació en una xarxa Modbus, cada byte (8 bits) s'envia en un missatge com a dos caràcters ASCII.

El principal avantatge que permet aquest mode és que permet intervals de temps de fins a un segon entre caràcters sense perill d'error.

Sistema de codificació
Hexadecimal caràcters ASCII 0-9 A-F Un caràcter hexadecimal contingut en cada caràcter ASCII del missatge
Bits per byte
1 bit d'arrancada 7 bits de dades (el menys significatiu s'envia primer) 1 bit per paritat parell o imparell (cap bit per No paritat) 1 bit de parada si s'usa paritat 2 bits de parada si no s'usa paritat
Camp de comprovació d'error
Comprovació Longitudinal Redundant (LRC)

Taula 4. Format per cada byte en mode ASCII

L'inici de la trama es detecta mitjançant la recepció dels caràcters “:” o per un silenci superior al temps entre caràcters.

### 3.3.2 Mode RTU

Quan en la comunicació en una xarxa Modbus s'usa el mode RTU, cada byte de 8 bits d'un missatge conté dos díigits hexadecimal de 4 bits. El principal avantatge del mode RTU és que la major densitat del caràcter permet millorar rendiment que el mode ASCII per la mateixa velocitat. Cada missatge s'ha de transmetre de forma continua.

Sistema de codificació
Binari 8 bits, hexadecimal 0-9 A-F Dos díigits hexadecimal en cada camp de 8 bits del missatge
Bits per byte
1 bit d'arrancada 8 bits de dades (el menys significatiu s'envia primer) 1 bit per paritat parell o imparell (cap bit per No paritat) 1 bit de parada si s'usa paritat 2 bits de parada si no s'usa paritat
Camp de comprovació d'error
Comprovació Cíclica Redundant (CRC)

Taula 5. Format per cada byte en mode RTU

### 3.4 Estructura del missatge

El missatge és una seqüència de caràcters que el receptor ha de poder interpretar. La seqüència de caràcters defineix la trama.

En qualsevol dels dos modes de transmissió (ASCII o RTU), un missatge Modbus té una trama amb un inici i amb un final coneguts. Això permet als dispositius receptors començar a iniciar el missatge, llegir la part de la direcció i determinar quin dispositiu s'ha direccionat (o tots els dispositius si és una difusió – direcció 0-) i saber quan s'ha completat el missatge.

El fet de que hi hagin missatges parcials pot produir errors.



### 3.4.1 Estructura ASCII

Treballant en mode ASCII, el missatge comença amb el caràcter " : " (dos punts; ASCII 3A hex), i acaba amb un parell de caràcters (CRLF) "Retorn de Carro + Avanç de línia" (ASCII 0D hex i 0A hex).

Els caràcters que es poden usar en la comunicació pels altres camps són 0-A, A-F hexadecimal.

Quan els dispositius de la xarxa reben el caràcter "dos punts" ( ' : ' ), cada dispositiu descodifica el pròxim camp (el de direcció) per saber quin ha estat dispositiu direccionat.

Dins el missatge poden existir intervals de fins a un segon entre caràcters; Si passa més temps entre caràcters, el dispositiu receptor assumeix que ha succeït un error.

INICI	DIRECCIÓ	FUNCIÓ	DADES	LRC	FINAL
caràcter " : "	2 caràcters	2 caràcters	N caràcters	2 caràcters	2 caràcters CRLF

Taula 6. Estructura típica d'un missatge ASCII

### 3.4.2 Estructura RTU

En mode RTU, els missatges comencen amb un interval silenciós d'almenys 3,5 vegades el temps necessari per transmetre un caràcter (temps de caràcter).

El primer camp que es transmet és la direcció del dispositiu destinatari.

Els caràcters que es poden transmetre per a tots els camps són 0-A, A-F hexadecimal.

Quan els dispositius reben el primer camp (el camp de direcció), cada dispositiu el descodifica per saber a qui és dirigeix el missatge.

En rebre l'últim caràcter, per què el dispositiu sàpiga que s'ha acabat el missatge, hi ha un interval silenciós d'almenys 3,5 temps de caràcter, i després d'això ja pot començar un altre missatge.

Cal recordar que el missatge complet s'ha de transmetre de forma contínua, si hi ha un interval silenciós de més de 1,5 temps de caràcter abans de completar el missatge, el dispositiu receptor pren el missatge com a erroni i assumeix que el proper byte que rebí serà el camp de direcció d'un nou missatge.

Passa una cosa semblant si un nou missatge comença abans de que passin els 3,5 temps de caràcter després d'un missatge rebut. En aquest cas, el dispositiu receptor el

considerarà una continuació de l'últim missatge rebut. Això provocarà un error ja que el valor del camp final CRC no serà vàlid.

INICI	DIRECCIÓ	FUNCIÓ	DADES	LRC	FINAL
3,5 temps	1 byte	1 byte	N bytes	2 bytes	3,5 temps

Taula 7. Estructura típica d'un missatge RTU

### 3.4.3 Camp de direcció

El camp de direcció és el primer del missatge després del temps de sincronització.

La direcció indica el dispositiu al que va dirigit el missatge. Cada dispositiu de la xarxa ha de tenir una direcció única i diferent de zero.

Quan el dispositiu respon a un missatge, ha d'enviar, primer de tot, la seva direcció per tal de que el mestre reconegui la procedència del missatge.

El camp de direcció d'un missatge conté dos caràcters (ASCII) o vuit bits (RTU). Les direccions que poden usar els esclaus han d'estar dins del rang de 1 a 247 decimal.

El protocol Modbus permet enviar missatges a tots els dispositius a la vegada usant la direcció zero, d'aquí que cap dels esclaus pugui tenir aquesta direcció.

El mestre direcciona un esclau situant la direcció de l'esclau en el camp de direcció del missatge. Quan l'esclau envia la seva resposta, posa la seva pròpia direcció en aquest camp de direcció de la resposta per donar a conèixer al mestre que l'esclau està responent. Si la direcció que envia el mestre és la direcció zero, tots els esclaus la reconeixen.

### 3.4.4 Camp de Funció

El camp de funció indica al dispositiu direccionat quin tipus de funció ha de realitzar.

Aquest camp conté dos caràcters (ASCII) o bé, vuit bits (RTU).

Els valors que pot prendre està dins el rang de 1 a 255 decimal. Algun d'aquests valors poden aplicar-se a tots els controladors Modicon, altres estan reservats a només alguns models de la mateixa casa i d'altres estan reservats per a futurs usos.

Quan el mestre envia un missatge a l'esclau, el camp de funció indica a l'esclau quina acció ha de fer. Quan l'esclau respon al mestre, usa aquest camp per indicar si la resposta és normal (sense errors) o si ha tingut algun problema i s'ha produït un error.

En el cas de que la resposta de l'esclau sigui normal, l'esclau retorna el codi de la funció original, i en el cas de que la resposta tingui un error, el que es retorna és el codi original amb el bit més significatiu posat a valor 1.

Per exemple, si el missatge enviat pel mestre a un esclau té el codi 0000 0011 (03 Hexadecimal ), si el dispositiu fa l'acció sol·licitada sense cap error, l'esclau retorna el mateix codi 0000 0011. Si es produeix un error, el que s'envia és el codi 1000 0011 (83 Hexadecimal).

Per tal de que el mestre sàpiga que s'ha produït un error, l'esclau situa un únic codi en el camp de dades del missatge de resposta, i això li indica al mestre quin tipus d'error s'ha produït.

### **3.4.5 Camp de Dades**

El camp de dades conté la informació necessària per tal de que els dispositius puguin executar les funcions sol·licitades o la informació enviada pels dispositius al mestre com a resposta a una funció.

El camp de dades està format per conjunts de dos dígit hexadecimals (dins del rang de 00 a FF hexadecimal), que segons el mode de transmissió usat, aquest conjunt estarà format per un parell de caràcters ASCII o per un caràcter RTU.

Si no hi ha cap error, el camp de dades d'una resposta emesa per un esclau, conté les dades sol·licitades, però si hi ha algun error, el camp conté un codi que el mestre pot usar per determinar la propera acció que es prendrà.

El camp de dades pot ser inexistent en algun tipus de missatges (com podria passar en cas de que el codi de funció ja especifiqués l'acció).

### **3.4.6 Camp de Comprovació d'Error**

El contingut del camp de comprovació d'error depèn del mode de transmissió que s'usi.

En mode ASCII, el camp de comprovació d'error conté dos caràcters ASCII. Aquests caràcters són el resultat d'un càlcul de Comprovació Longitudinal Redundant (LRC) que es fa sobre el contingut del missatge (sense tenir en compte els dos punts ' : ' inicials i els caràcters CRLF del final).

Els caràcters LRC s'afegeixen al final del missatge com a penúltim camp (anterior al CRLF).

En mode RTU, el camp de comprovació d'error conté un valor de 16 bits implementat com a dos bytes de 8 bits. El valor de comprovació d'error és el resultat d'un càlcul Comprovació Cíclica Redundant (CRC) realitzat sobre el contingut del missatge.

El camp CRC s'afegeix al final de tot del missatge. Per fer-ho, primer s'afegeix el byte de menys pes i llavors el de més pes. El byte d'ordre alt del CRC és l'últim byte que s'enviarà en el missatge.

El CRC és l'últim que porta la trama i permet, al mestre i als dispositius, detectar errors de transmissió. CRC assegura que els dispositius receptors o el mestre no facin accions incorrectes degut a una modificació errònia del missatge (pot ser provocat per un soroll elèctric o altres interferències).

La resposta de l'esclau consisteix en la direcció del dispositiu, el codi de la funció, el codi d'error i el CRC.

Per indicar que la resposta és una notificació d'error, el bit de més pes del codi de la funció, està activat a '1'.

### 3.4.7 Transmissió en sèrie de caràcters

En una xarxa Modbus, la transmissió en sèrie de caràcters (transmissió de cada caràcter o byte) es fa de la següent manera: es comença enviant el Bit Menys Significatiu (LSB) i es continua fins el Bit Més Significatiu (MSB).

Segons es treballi en mode ASCII o mode RTU, la seqüència de transmissió serà d'una manera o d'una altra, i també variarà segons si hi ha control de paritat o no.

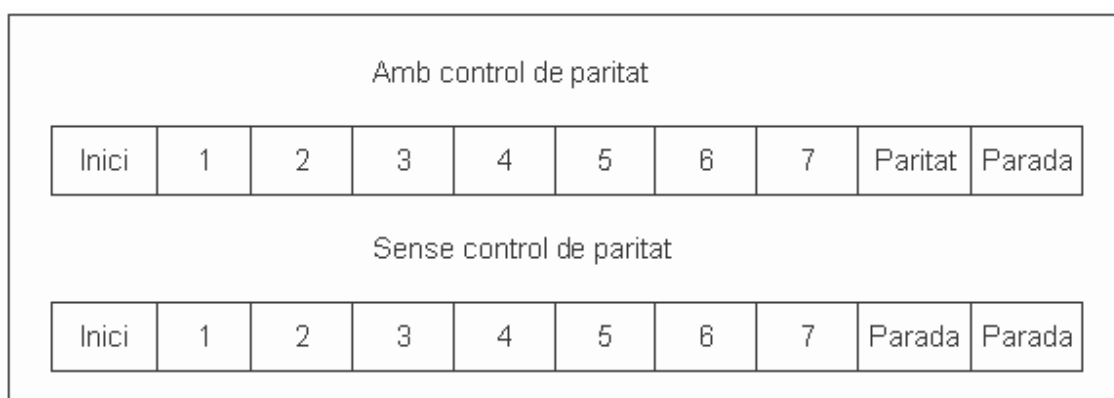


Figura 5. Ordre dels bits en ASCII

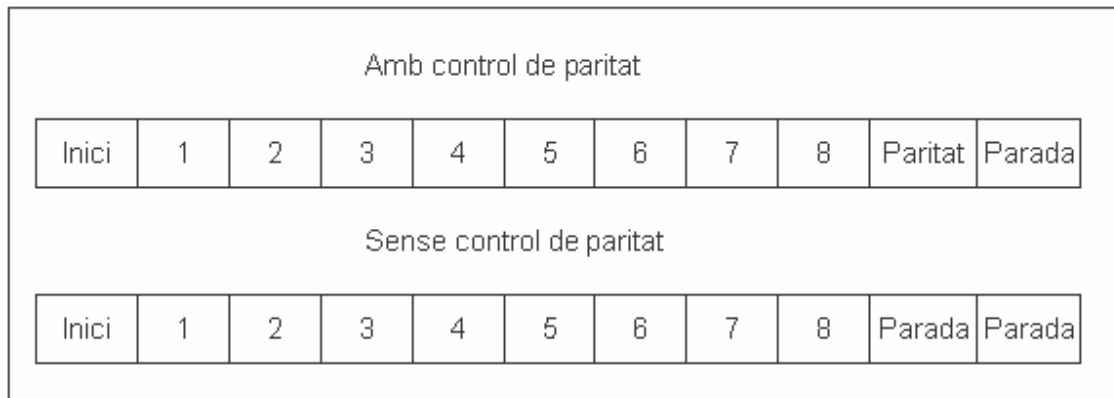


Figura 6. Ordre dels bits en RTU

### 3.4.8 Mètodes de comprovació d'error

Existeixen dos mètodes de comprovació d'error, la comprovació de la paritat (parell o imparell) o bé la comprovació de la trama (LRC o CRC). La comprovació de la paritat es pot aplicar a cada caràcter (opcional), en canvi, la comprovació de la trama s'aplica al missatge complet.

Aquests dos mètodes de comprovació d'error són creades en el mestre i aplicades als missatges abans de ser transmesos.

L'esclau, en rebre, comprova cada caràcter i la trama del missatge complet.

Ha d'existir un interval de temps (predeterminat per l'usuari) suficientment llarg per què l'esclau pugui respondre normalment. Si l'esclau detecta un error en la transmissió, no tindrà en compte el missatge i no el respondrà, i això farà que el mestre detecti que ha passat el temps d'espera i sabrà que hi ha hagut un error.

En cas de que el mestre envii un missatge a un esclau que no existeix, es produirà un error de temps excedit que s'anomena "time out".

#### 3.4.8.1 Control de paritat

L'usuari pot configurar els controladors per control de paritat parell o imparell, o bé, sense control de paritat. Això determinarà com s'iniciarà el bit de paritat en cada caràcter.

Si s'especifica un control de paritat parell o imparell, es contarà la quantitat de bits que tenen valor '1' en la porció de dades de cada caràcter (7 bits de dades per mode ASCII o 8 bits de dades per mode RTU). Al bit de paritat se li haurà de donar valor 0 o 1 per tal de tenir un número parell o imparell (al final) de bits amb valor 1.

Per exemple, tenim 8 bits de dades (1100 0101) que formen part d'un missatge de mode RTU . En aquesta trama hi ha quatre bits de valor '1'.

Si fem un control de paritat parella, el bit de paritat de la trama ha de tenir valor '0' fent que la quantitat de bits de valor '1' segueixi essent un número parell (quatre).

Si fem un control de paritat imparell, el bit de paritat haurà de tenir valor '1', imparell (cinc).

Quan el mestre envia el missatge, el bit de paritat es calcula i s'aplica a la trama de cada caràcter. Quan l'esclau rep el missatge, conta la quantitat de bits de valor '1' i, si no correspon la paritat configurada (tots els dispositius s'han de configurar amb el mateix control de paritat), dóna error.

La comprovació de la paritat només es detecta si un número imparell de bits s'han alterat durant la transmissió.

Per exemple, si s'usa control de paritat imparell i dos bits de valor '1' d'un caràcter que té originalment tres bits de valor '1', han quedat falsejats (passen a valor '0') durant la transmissió, el resultat és un número imparell de bits de valor 1 (l'error no es detecta amb aquest sistema).

Si s'especifica control sense paritat, no es transmet bit de paritat i no es fa la comprovació de paritat. Es transmet un bit de parada addicional per emplenar la trama de caràcter.

#### **3.4.8.2 Comprovació de la trama LRC**

En mode ASCII, els missatges inclouen un camp de comprovació d'error que es basa en un mètode de Comprovació Longitudinal Redundant (LRC).

Aquest camp controla el contingut del missatge (excepte el caràcter inicial ' : ' i el CRLF) i s'aplica independentment del mètode de paritat que s'hagi utilitzat pels caràcters individuals del missatge.

LRC és un byte que calcula el dispositiu mestre i que s'afegeix al missatge. El receptor calcula el LRC mentre rep el missatge i el compara amb el valor de LRC que ha rebut del mestre, i si els valors no coincideixen, es produeix un error.

#### **3.4.8.3 Comprovació de la trama CRC**

Treballant en mode RTU, els missatges inclouen un camp de comprovació d'error que està basat en un mètode Comprovació de Redundància cíclica (CRC). Aquest camp CRC controla el contingut de tot el missatge.

Igual com en la comprovació LRC, la comprovació CRC s'aplica independentment del control de paritat que s'hagi utilitzat pels caràcters individuals del missatge.

El camp CRC és de dos bytes, el calcula el mestre i l'afegeix al missatge. L'esclau calcula el CRC mentre rep el missatge i compara el valor calculat amb el valor rebut, i si els dos valors no són iguals es produeix un error.

Per tal de calcular el valor CRC, es carrega un registre de 16 bits (tots els bits a '1').

Llavors, comença un procés que agafa els successius bytes del missatge i els opera amb el contingut del registre i l'actualitza amb el resultat obtingut. Només els 8 bytes de dades de cada caràcter s'usen per generar el CRC.

Pel CRC no es tenen en compte els bits d'arrancada, parada i paritat.

Durant la generació del CRC es fa una operació booleana XOR (OR exclusiu) a cada caràcter de 8 bits amb el contingut del registre; llavors, s'aplica al resultat un desplaçament de bit en la direcció del bit menys significatiu (LSB), emplenant la posició del bit més significatiu (MSB) amb un zero.

Si LSB extret és un '1', es fa una XOR entre el registre i un valor fix predeterminat (A001 hex, que correspon al polinomi generador CRC16 'invers', que és el que s'aplica al CRC Modbus). Si l'LSB és un '0', no es fa XOR.

Aquest procés es va repetint fins que s'hagin fet 8 desplaçaments, i quan s'ha fet el vuitè desplaçament, es fa XOR amb el pròxim byte i el valor actual del registre CRC, i el procés es repeteix amb vuit desplaçaments més, i així fins a processar tots el bytes del missatge. Un cop fet tot això, el contingut final del registre és el valor del CRC.

Quan s'afegeix el CRC al missatge, primer s'afegeix el byte d'ordre baix seguit del byte d'ordre alt.

### **3.5 Descripció de les funcions principals**

#### **3.5.1 Llegir N bits**

Aquesta funció permet a l'usuari obtenir els valors lògics dels bits del dispositiu direccionat. Les dades de resposta van empaquetades en bytes de manera que el primer bit sol·licitat ocupa el bit de menys pes del primer byte de dades, i els següents van a continuació, de manera que, si no són un número múltiple de 8, l'últim byte es completa amb zeros.

Direcció de l'esclau	Codi de la Funció	Direcció del primer bit		Número de bits a llegir (màx 255)		CRC	
		MSB	LSB	MSB	LSB	MSB	LSB
1 byte	1 byte						

Taula 8. Trama mestre-esclau (bits)

Direcció de l'esclau	codi de la Funció	Número de bytes llegits	Primer byte de dades	.....	Últim byte de dades	CRC	
				.....		MSB	LSB
1 byte	1 byte	1 byte	1 byte	.....	1 byte		

Taula 9. Trama esclau-mestre (bits)

### 3.5.2 Llegir N registres

Aquesta funció permet a l'usuari obtenir els valors dels registres del dispositiu direccionat. Aquests registres emmagatzemen els valors numèrics dels paràmetres i variables del controlador. El rang de les dades varia de 0 a 65536 ( 0000h a FFFFh ). Les dades corresponents a les direccions de registres que passen de l'última direcció vàlida, s'assignen a zero (00 00).

Direcció de l'esclau	Codi de la Funció	Direcció del primer registre		Número de registres a llegir (màx 51)		CRC	
		MSB	LSB	MSB	LSB	MSB	LSB
1 byte	1 byte						

Taula 10. Trama mestre - esclau (registres)

Direcció de l'esclau	codi de la Funció	Número de bytes llegits	Valor primer registre		.....	Valor últim registre		CRC	
			MSB	LSB	.....	MSB	LSB	MSB	LSB
1 byte	1 byte	1 byte			.....				

Taula 11. Trama esclau - mestre (registres)



### 3.5.3 Assignar un bit

Aquesta funció permet a l'usuari assignar els valors dels bits del dispositiu direccionat. Per desactivar el bit, cal enviar 00h i per activar-lo, cal enviar 01h o FFh. Aquest valor s'ha d'escriure en el byte més significatiu.

Direcció de l'esclau	codi de la Funció	Direcció del bit		Valor del bit		CRC	
1 byte	1 byte	MSB	LSB	MSB	LSB	MSB	LSB

Taula 12. Assignar bit (mestre - esclau)

Direcció de l'esclau	codi de la Funció	Direcció del bit		Valor del bit		CRC	
1 byte	1 byte	MSB	LSB	MSB	LSB	MSB	LSB

Taula 13. Assignar bit (esclau - mestre)

### 3.5.4 Assignar un registre

Aquesta funció permet a l'usuari modificar el contingut dels paràmetres del dispositiu direccionat.

Els valors s'envien escalats segons el factor d'escala corresponent a cada paràmetre, en un rang entre 0000h i FFFFh.

Direcció de l'esclau	codi de la Funció	Direcció del registre		Valor del registre		CRC	
1 byte	1 byte	MSB	LSB	MSB	LSB	MSB	LSB

Taula 14. Assignar registre (mestre – esclau)

Direcció de l'esclau	codi de la Funció	Direcció del registre		Valor del registre		CRC	
1 byte	1 byte	MSB	LSB	MSB	LSB	MSB	LSB

Taula 15. Assignar registre (esclau – mestre)

### 3.5.5 Llegir el byte d'estat

Aquesta funció permet a l'usuari obtenir una lectura ràpida de l'estat del dispositiu direccionat mitjançant la lectura d'un sol byte.

Direcció del dispositiu	Codi de Funció	CRC	
1 byte	1 byte	MSB	LSB

Taula 16. Llegir byte d'estat (mestre – esclau)

Direcció del dispositiu	Codi de Funció	Byte d'estat	CRC	
1 byte	1 byte	1 byte	MSB	LSB

Taula 17. Llegir byte d'estat (esclau – mestre)

#### **4. XARXA DE COMUNICACIONS**

A la indústria actual, la major part de processos són automatitzats i la intel·ligència del procés està continguda en la unitat de control. La realització tecnològica d'aquesta intel·ligència ha adoptat diverses formes o implementacions al llarg de la història industrial, des d'automatismes purament mecànics, fins als autòmats programats actuals. Avui en dia, la majoria de processos automatitzats estan controlats per autòmats programables, la potència dels quals ha augmentat considerablement, i això ha permès que puguin fer aplicacions de control complicades en un temps mínim i amb gran fiabilitat.

En les gammes d'autòmats s'hi ha anat incorporant tota una gamma de nous mòduls que permeten realitzar funcions especials com el control d'eixos, la comunicació, etc.

##### **4.1 Autòmats TSX3705 / TSX3722**

L'autòmat TSX 3722 és d'una gamma més alta que el TSX 3705. Inclou 3 espais amb 24 V DC d'alimentació, un processador amb memòria RAM de 20 K paraules, una memòria de seguretat FLASH EEPROM, 2 espais per a targetes PCMCIA i un rellotge calendari (que permet ampliar el volum de memòria de l'aplicació i permet a l'autòmat rebre un mòdul de comunicacions). Es pot ampliar amb dos espais més.

De sèrie no incorpora els mòduls d'entrades / sortides TON, encara que s'hi poden incorporar.

L'autòmat es presenta en dos configuracions per poder respondre a les necessitats d'alimentació alterna i continua.



Figura 7. Autòmat TSX 3722

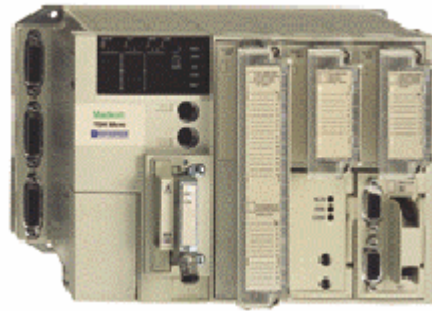


Figura 8. Autòmat amb mòduls instal·lats

Els mòduls que es poden instal·lar són:

1 mòdul d'entrades / sortides TON de format estàndard (28 o 64 E/S).

2 mòduls de mig format de tipus entrades / sortides TON (4 a 12 E/S), entrades analògiques (de 2 a 8 vies) o comptatge (1 o 2 vies).

Segons com es configuren els mòduls que es col·loquin, es pot obtenir un màxim de 256 E/S a 24 V DC.

Pel que fa a les comunicacions, els controladors TSX 37 integren un enllaç multifunció mitjançant el connector terminal. Aquest connector terminal és un enllaç RS 485 no aïllat, format per un connector mini DIN de 8 punts.

El controlador TSX 3705 disposa d'un connector terminal TER i el TSX 3722 té dos terminals diferents (tot i que funcionalment són idèntics) anomenats TER i AUX.

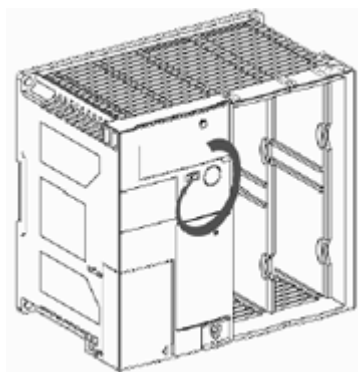


Figura 3. Connector TER

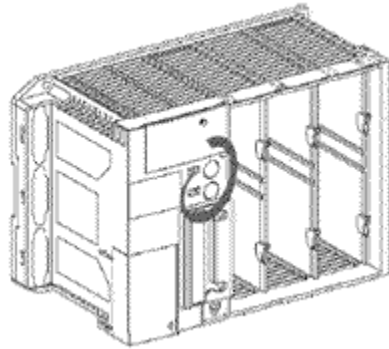


Figura 9. Connector TER i AUX

Aquest port de comunicacions es pot configurar com a Unitelway o com a Modbus mestre o esclau, però tenint en compte que la configuració escollida sigui la mateixa pels dos connectors TER i AUX.

Els autòmats del laboratori de regulació tenen instal·lat un mòdul mixt d'entrades/sortides tot o res, TSX DMZ 28DR.

Aquest mòdul inclou 28 entrades/sortides repartides en 16 entrades de 24 VCC, de lògica positiva tipus 1 o lògica negativa i 12 sortides de relés.

Per tal de dur a terme la comunicació Modbus dels autòmats, no s'ha fet ús de cap tarja PCMCIA ja que, sense tarja es poden tenir connectats al bus fins a un màxim de 28 dispositius.

Només l'autòmat TSX 3722 serà el que podrà configurar-se com a mestre, sempre i quan, la versió del software no sigui inferior a V5.0.

L'autòmat TSX 3705 és el més senzill de la gamma i una de les seves restriccions és en el mètode d'accés, és a dir, no pot ser mestre. Els autòmats TSX 3705 es configuraran com a esclaus mitjançant el connector terminal associat a nivell físic RS485.

Una altra condició per poder configurar el controlador com a mestre, és que la versió del programa PL7 Pro sigui igual o superior a V4.2.

Per poder tenir l'autòmat com a mestre, es va actualitzar la versió del software del Micro a V6.0. També es va actualitzar la versió del programa PL7 Pro a V4.4. (només es va instal·lar la nova versió al PC corresponent al TSX 3722, als altres no era necessari).

Com que el port TER és l'únic pel qual es pot accedir a un Micro des d'un PC per programar-lo, en fer ús d'aquest terminal per realitzar la comunicació, sense tarja PCMCIA no es podrà seguir la comunicació des del PC en les proves que es puguin dur a terme, és a dir, no es podrà visualitzar a la pantalla del PC.

En el cas de que haguéssim de tenir més dispositius connectats a la xarxa, llavors caldria un tarja PCMCIA, la qual permet tenir fins a 98 esclaus en connexió.

Respecte el mode de transmissió sèrie, el que s'usa és el RTU ja que si no s'usa una tarja PCMCIA no es pot usar el mode ASCII, i com ja s'ha dit, no se n'ha adquirit cap.

#### 4.2 Unitat TSX PACC 01

La connexió d'un autòmat TSX 37 (mestre o esclau) a un bus Modbus, necessita obligatòriament l'ús d'una caixa TSX PACC 01.

Només es pot connectar una unitat d'aïllament a cada controlador TSX 37.

La caixa TSX PACC 01 és una unitat d'aïllament que es connecta al terminal TER dels controladors TSX 37 mitjançant un connector mini-Din en un dels seus extrems. Aquesta caixa permet connectar diferents equips amb el connector terminal dels controladors TSX 37- 05/08/10 (per això te dos connectors mini-Din TER i AUX, equivalents als dels controladors TSX 37-■■■).

La unitat TSX PACC 01 permet connectar un controlador TSX 37 al bus Modbus/Jbus, i també, fixar la modalitat de funcionament del connector terminal, és a dir, es pot escollir si treballem en Modbus esclau, Uni-telway esclau, etc. També adapta el bus quan la caixa està connectada a un dels extrems del bus Modbus.

L'aspecte extern de la caixa TSX PACC 01 es el que es veu a la Figura 10.

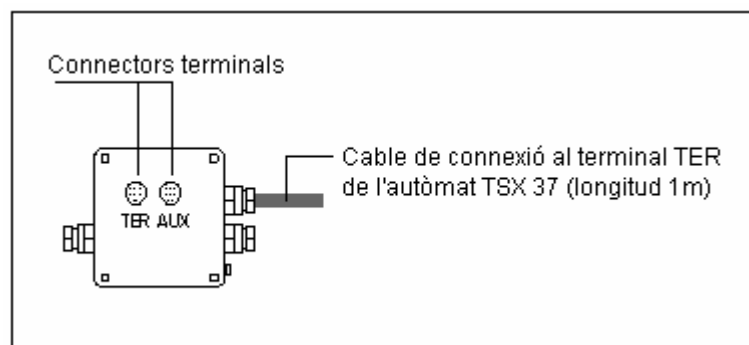


Figura 10. Unitat d'aïllament TSX PACC 01

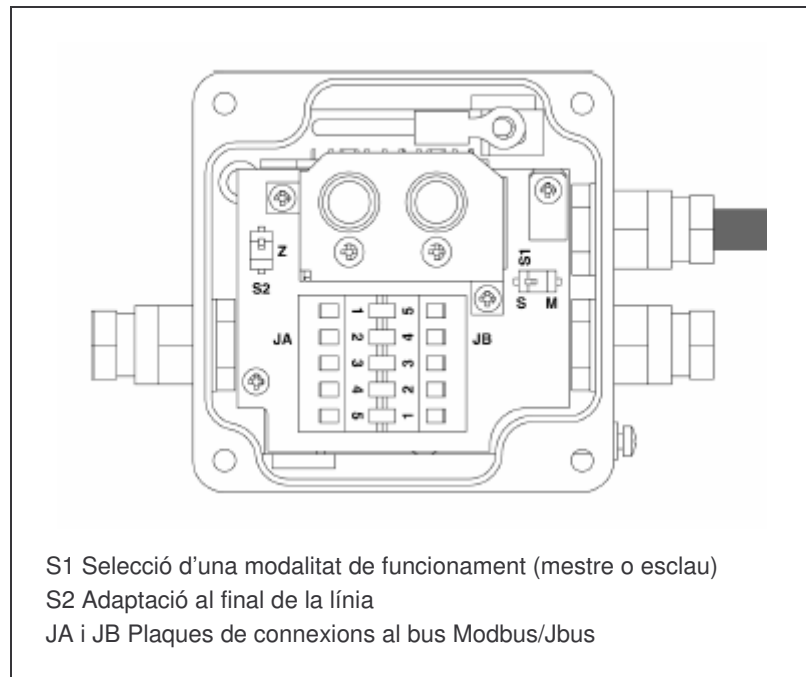


Figura 11. Vista interna de la caixa TSX PACC 01

Segons com es col·loquin els interruptors S1 i S2, escollirem la modalitat de funcionament i si la caixa es troba en un cap de la xarxa.

	OFF	ON
S2	Caixa situada en posició diferent al final de línia	Caixa situada al final de línia
S1	Uni-telway mestre	Modbus, Uni-telway esclau o Mode caràcters

Taula 18. Posició dels interruptors S1 i S2

Per realitzar una comunicació en Modbus pel port TER de l'autòmat, sempre que no tinguem tarja PCMCIA, cal posar S1 en ESCLAU, ja que és en aquesta posició quan s'estableix el pont entre els pins 7 i 5 del port TER. Aquest pont és imprescindible per poder realitzar una comunicació Modbus pel port TER.

Si es disposa d'una PCMCIA, S1 pot posar-se en MASTER o SLAVE, és indiferent, ja que no és necessari el pont anteriorment esmentat.

Aquesta unitat d'aïllament duplica el connector terminal, així doncs, en un autòmat TSX 3705 que inicialment té només un connector terminal TER, amb la caixa TSX PACC 01 tindrà un connector TER i un connector AUX.

### 4.3 Muntatge de la xarxa

De forma general, la xarxa de comunicació té la forma que es mostra a la figura 12. S'hi pot veure com es connecten els esclaus i el mestre, les unitats d'aïllament i el cablejat que cal fer-hi. Més endavant es mostraran les connexions de forma més detallada.

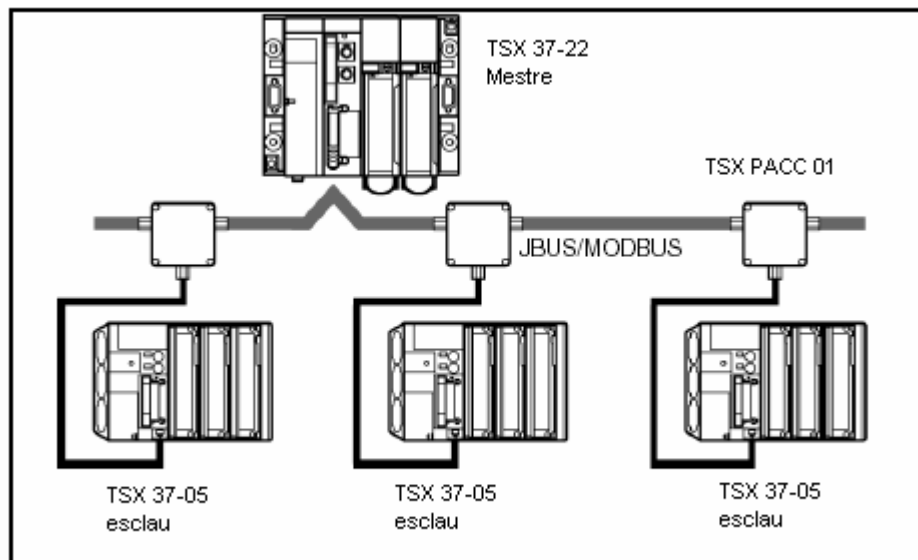


Figura 12. Forma d'una xarxa Modbus

La primera cosa que s'ha de fer és configurar les unitats d'aïllament TSX PACC 01 segons convingui (interruptors S1 i S2).

Pel que fa a l'autòmat TSX 3722, s'ha hagut d'instal·lar el sistema operatiu V6.0 per tal de poder realitzar una comunicació Modbus.

#### 4.3.1 Connexionat de les experiències

Les experiències que es faran consistiran en realitzar una comunicació entre els PLC's del laboratori de regulació (TSX 3722 i TSX 3705), una comunicació senzilla per començar a familiaritzar a l'alumne amb la comunicació Modbus. La pràctica que haurà de realitzar l'alumne tindrà tres exercicis a fer, amb un grau de dificultat progressiu en cada un d'ells.

Quan es treballi en mode connectat, les caixes TSX PACC 01 s'hauran de configurar segons el lloc que ocupin en la xarxa.

Els autòmats es connectaran amb les unitats d'aïllament mitjançant el terminal TER del propi autòmat. Els PC's no es connectaran a les caixes ja que caldria una tarja



PCMCIA per poder visualitzar el procés simulat, ja que el bus queda ocupat per la comunicació dels PLC's.

La figura 13 mostra de forma més detallada, com ha de ser el cablejat per a la comunicació Modbus.

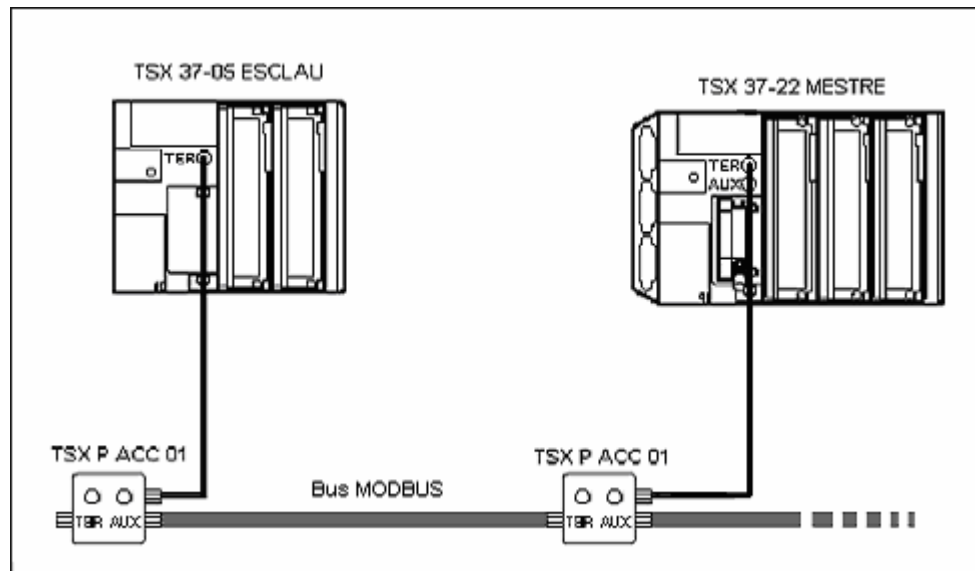


Figura 13. Connexionat de les experiències

Les direccions dels PLC's seran diferents per cada banc de treball i se'ls hi assignaran en realitzar la pràctica. Les direccions assignades correspondran al número d'esclau. El número d'esclau s'introduirà al PLC mitjançant la pantalla de configuració del PL7 Pro. Un cop configurada aquesta pantalla, mirar que la direcció que apareix en l'apartat "menú / Autómata / Definir la direcció del autómata", sigui la direcció SYS (direcció del sistema de l'autòmat ).

Seguidament, es bolcarà el programa al PLC (el PC i el PLC estaran connectats directament, no en xarxa) mitjançant el cable TSX PCU 1030. Llavors, es deixa l'autòmat en RUN i es desconnecta el cable esmentat. Es fan les connexions amb la xarxa, i ja es podran reconèixer el mestre i els esclaus en la comunicació.

## **5. DESCRIPCIÓ SOFTWARE PL7- Pro**

### **5.1 Introducció**

El programa PL7 Pro ja s'ha utilitzat en les pràctiques d'automatització industrial, i es va fer una introducció del funcionament del programa. Es va ensenyar a crear un programa, editar-lo, i comprovar el funcionament del programa quan s'executa en la CPU.

En aquest apartat s'explicarà el funcionament del software PL7 Pro referent a la comunicació Modbus, què cal saber per tal de dur a terme aquest tipus de protocol, funcions específiques per dur a terme la comunicació.

### **5.2 Direccionament dels autòmats**

Mitjançant el software PL7 Pro es pot accedir a qualsevol autòmat connectat a la xarxa. Per fer això, els autòmats que es comuniquen a través d'una xarxa, han de tenir les direccions definides per programa abans de fer qualsevol comunicació.

Per tal de definir la direcció dels autòmats cal tenir l'ordinador en mode local i donar-li la direcció mitjançant la pantalla de configuració, i un cop estiguin definides totes les direccions dels autòmats, els podrem connectar en xarxa i es podrà accedir a tots ells des de qualsevol lloc de la xarxa definint la seva adreça.

Per definir la direcció de l'autòmat cal seguir els següents passos:

Obrir el programa PL7, i un cop escollit l'autòmat amb què es treballarà, accedir a la configuració del hardware ("Herramientas / Configuración ", clicar sobre el requadre "COM"). Aquí és on s'escull si l'autòmat actua com a mestre o esclau. Si treballa com a esclau, s'especificarà quin número d'esclau és (i ja s'haurà definit la seva direcció).

#### **5.2.1 Primera programació dels autòmats d'una xarxa**

La direcció dels autòmats es defineix per programa. El primer cop que es programa, s'ha de connectar l'ordinador directament al connector TER de l'autòmat per tal de configurar la seva futura direcció. Quan cada autòmat tingui una direcció assignada (no es pot donar la mateixa direcció a dos autòmats), en la comunicació en xarxa es podrà accedir a tots ells des de qualsevol zona.

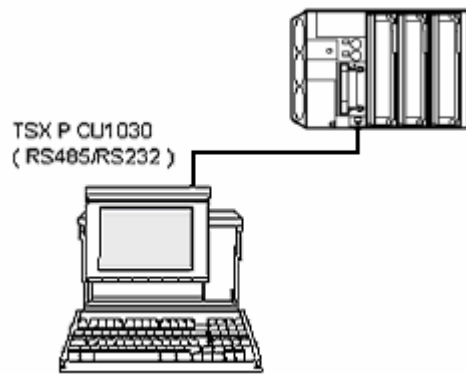


Figura 14. Esquema de connexió

### 5.2.2 Tipus de direccionament

Existeixen diversos tipus de direccionament en funció del tipus de tasca que es dugui a terme.

<b>Direccions locals</b>	
Destinatari	Direcció
Servidor UNI-TE d'un Micro/Premium Aplicació PL7 Micro Pro Esclau Modbus	SYS APP mòdul.via.número d'esclau
<b>Direccions remotes</b>	
Destinatari	Direcció
Destinatari en una xarxa remota	{xarxa.estació}direcció local
Destinatari en una xarxa local	{estació}direcció local
<b>Direccions en difusió</b>	
Destinatari	Direcció
Difusió a totes les estacions	{xarxa.ALL}direcció local
Difusió a tots els mòduls	ALL.SYS
Difusió a tots els esclaus modbus	mòdul.via.ALL

Taula 19. Tipus de direccionament

## **5.3 Connexió dels autòmats**

### **5.3.1 Mode local**

Quan es parla de mode local es fa referència a treballar sense connexió entre el PC i el PLC.

Aquest tipus de comunicació permet crear o modificar una aplicació en el terminal. Es poden realitzar diverses funcions com crear o modificar el programa (LD, IL, ST o Grafcet), modificar el període de les tasques, modificar els paràmetres dels DFB, modificar el número de paraules internes, etc.

La direcció de l'autòmat de forma predeterminada SYS, correspon a la direcció del sistema de l'autòmat.

L'aplicació es guarda al disc dur.

### **5.3.2 Mode connectat**

S'anomena així quan existeix connexió de l'autòmat amb el PC.

En aquest mode de connexió, l'aplicació es guarda a l'autòmat de manera automàtica, i si es vol guardar al disc dur, s'ha de fer amb "Archivo / Guardar".

Per passar de mode local a mode connectat, només cal anar al menú del programa PL7 i clicar a "Automata / connectar".

En el nostre cas en concret, quan es treballi en mode connectat (si es treballa amb la xarxa de comunicació) no es podrà visualitzar el procés que s'hagi introduït al PLC, això només es podrà fer si es disposa d'una tarja PCMCIA que ens permetrà visualitzar el programa i, al mateix temps mantenir una comunicació en Modbus.

### **5.3.3 Configuració mòdul de comunicació**

Un cop creada una aplicació, mitjançant l'apartat del menú "Herramientas / Configuración" o bé, mitjançant el "navegador de la aplicación", es pot accedir a la configuració del hardware. Serà aquí on s'afegiran i es configuraran mòduls de comunicació (velocitat de transmissió, paritat, via de comunicació, etc.).

La pantalla de configuració determinarà el funcionament de la comunicació.

La Figura 15 mostra de forma gràfica, la manera d'accedir a la configuració del hardware en el PL7 Pro.

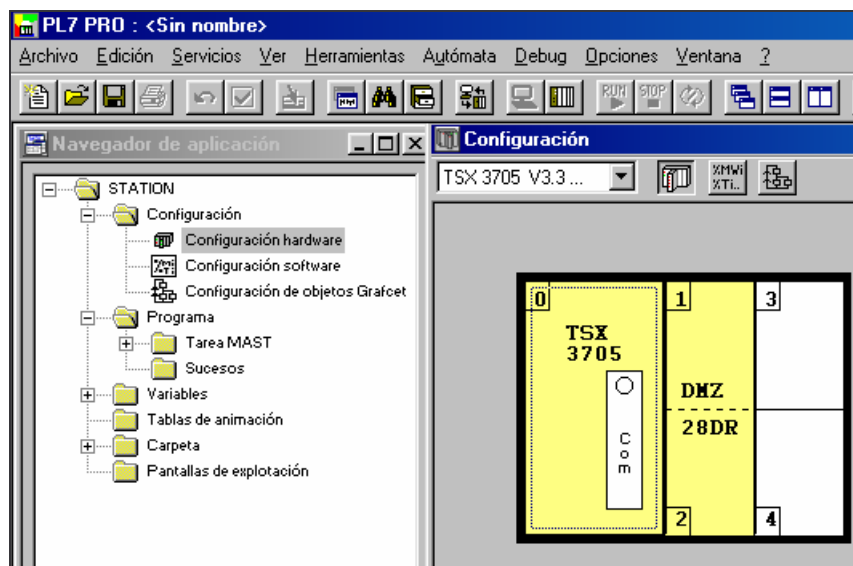


Figura 15. Configuració hardware

Quan es clica a la part de comunicació (COM), apareix una pantalla com aquesta:

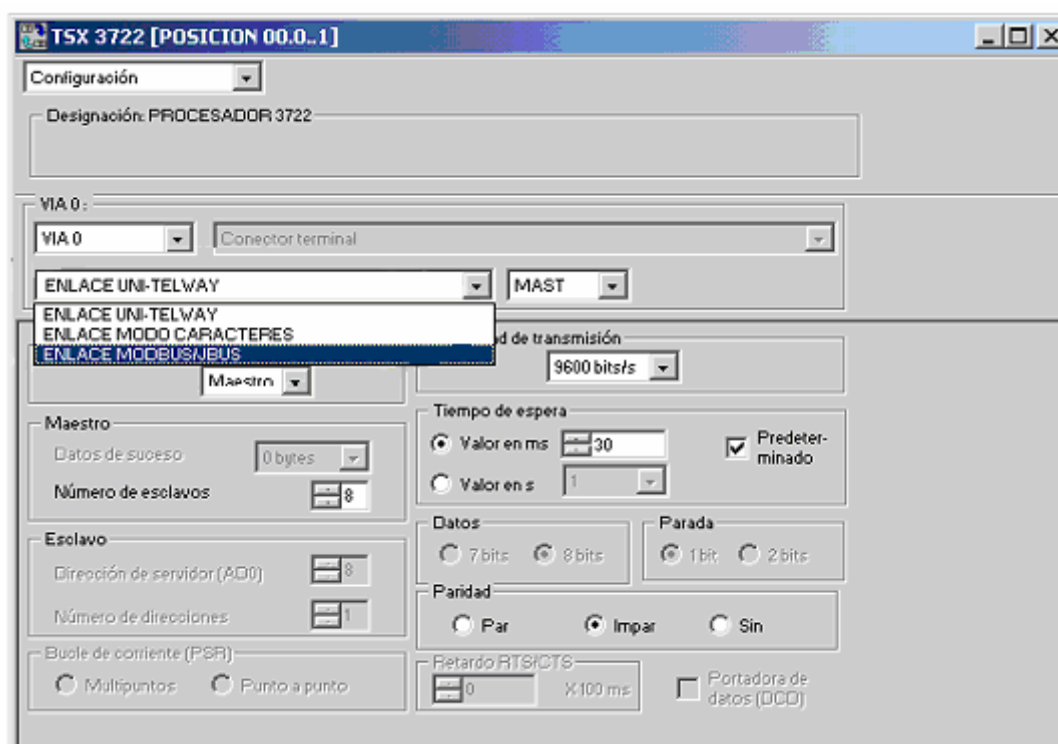


Figura 16. Pantalla de configuració

Es tracta d'una pantalla dividida en dues àrees que permet establir la via de comunicació i configurar els paràmetres necessaris per aconseguir una comunicació Modbus.

Hi apareixen uns valors per defecte que caldrà modificar si s'escau.

A la part superior s'hi troben la barra de títol, que indica la referència comercial i la posició del mòdul, una llista a la part superior permet escollir el mode de configuració o depuració. El darrer només és accessible quan s'està en mode connectat. Llavors hi ha una finestra on es visualitza una descripció del mòdul seleccionat.

En la finestra "Via" es selecciona la via de comunicació. La via 0 és la del connector terminal i la via 1 és la tarja PCMCIA (en cas de tenir-ne, llavors caldrà escollir el model de tarja) .

També es selecciona el mode de funcionament de la via, que anirà en funció de la via seleccionada (en la via 0 els modes que es poden escollir són Modbus, Uni-telway o mode caràcters). I al costat hi ha la finestreta que permet assignar el mòdul de comunicació a una tasca d'autòmat (els mòduls de comunicació sempre s'han de declarar a la tasca MAST).

A la zona inferior de la pantalla de configuració és on es seleccionen els paràmetres d'un enllaç Modbus (en aquest cas en concret) o bé, on es dóna informació dels mateixos.

La finestra tipus permet seleccionar l'opció de connexió de l'autòmat dins la xarxa, com a mestre o esclau Modbus (recordar que només pot haver-hi un mestre i que serà el controlador TSX 3722).

Si l'opció escollida és mestre, apareix una finestra que permet configurar el número de reiteracions (intents de connexió del mestre abans de declarar absent a l'esclau) i el temps de resposta (temps màxim entre l'emissió de l'últim caràcter de la petició emesa pel mestre i la recepció del primer caràcter de la petició enviada per l'esclau).

Si l'opció escollida és esclau, apareix una finestra que permet configurar el número d'esclau de l'equip.

Un cop configurada la via de comunicació, cal configurar els paràmetres de la transmissió. Aquests paràmetres són : Velocitat de transmissió (ha de ser la mateixa per a tots els equips de la xarxa), Temps entre caràcters (temps de detecció de final de la trama i del temps màxim entre dos caràcters de recepció, i es gestiona quan l'autòmat està en recepció de missatges, ja sigui mestre o esclau), Dades (permet obtenir informació del tipus de codificació usada per la comunicació, i cal configurar-se segons els altres equips de la xarxa), Parada (permet informar del número de bits de parada usats per la comunicació) i Paritat (permet definir el tipus de bit de paritat, normalment és imparell però en cas de Modbus és sense paritat).

## **5.4 Funcions de comunicació**

Totes les vies de comunicació X-WAY utilitzen un servei de missatgeria per permetre l'intercanvi de dades.

El PL7 ofereix funcions de comunicació específiques que autoritzen l'emissió i/o recepció de missatges cap/o provinents d'una entitat de comunicació.

Algunes funcions són comuns a diferents tipus de via de comunicació, i d'altres poden ser específiques d'una sola funció de comunicació.

El tractament de les funcions de comunicació és asíncron respecte el tractament de la tasca d'aplicació que ha permès activar-les. Les funcions d'emissió/recepció i de parada d'operació, són les úniques excepcions ja que la seva execució és totalment síncrona amb l'execució de la tasca d'activació.

Es diu que una funció de comunicació és síncrona quan s'executa per complet durant la tasca de l'autòmat que l'ha activat, i és diu que és asíncrona quan s'executa durant una o varies tasques de l'autòmat després de que aquestes l'haguessin activat.

### **5.4.1 Estructura general de les funcions de comunicació**

Una funció de comunicació utilitza: un paràmetre de direcció, paràmetres específics d'una operació de comunicació i paràmetres de gestió.

En general, les funcions de comunicació tenen la forma següent:

Funció (Direcció del destinatari, Paràmetres específics, Paràmetres de gestió)

El paràmetre "Funció" fa referència al tipus de funció de comunicació.

La "Direcció del destinatari" és la direcció del destinatari que participarà en l'intercanvi.

Els "Paràmetres específics" depenen del tipus de funció de comunicació i cada funció de comunicació té la seva descripció específica.

Els "Paràmetres de gestió" són comuns a totes les funcions de comunicació asíncrones. Es componen d'un paràmetre que dóna informació sobre l'activitat de la funció, d'un paràmetre que precisa el número d'intercanvi que identifica la tasca en curs, d'un paràmetre que conté la confirmació de l'intercanvi (confirmació de comunicació i confirmació de l'operació), d'un paràmetre de temps d'espera que permet controlar l'absència de respostes, i d'un paràmetre de longitud que permet memoritzar el número de bytes que s'emetrans o el número de bytes rebuts.

### 5.4.2 Les funcions de comunicació en Modbus

Les funcions de comunicació que permet el protocol Modbus són : READ\_VAR, WRITE\_VAR, SEND\_REQ i SERVER.

Aquestes funcions de comunicació són asíncrones, és a dir, s'executen durant una o varies tasques després de que aquestes les hagin activat.

Funció	Ús
READ_VAR	Llegir objectes del llenguatge de base: paraules, bits, paraules dobles, flotants interns, paraules constants, paraula i bit de sistema, temporitzadors, monoestables, programadors cíclics, registres, comptadors.
WRITE_VAR	Esriptura d'objectes del llenguatge de base: paraula, bit, paraula doble, flotant intern, paraula i bit de sistema.
SEND_REQ	Emissió de peticions i codificació de peticions UNI-TE i Modbus, i recepció de les respostes associades.
SERVER	Tracta les peticions READ_VAR i WRITE_VAR en Modbus de manera immediata (servidor immediat).

Taula 20. Funcions de comunicació per Modbus

### 5.4.3 Direcció destinatària

Aquest paràmetre es pot trobar per paraules internes (%MW) o per constants internes (%KW), o bé, es pot escriure directament en valor immediat.

Per facilitar l'intercanvi, el PL7 disposa de l'operador ADR# que permet l'assignació d'un valor (de forma immediata) de tipus direcció en una taula que sempre inclou sis paraules internes (%MW) o sis paraules constants (%KW) consecutives.

### 5.4.4 Paràmetres de gestió

Aquests paràmetres identifiquen, d'entre 4 paraules internes (%MW:4), la direcció de les dades PL7 que s'usen per controlar les funcions de comunicació. Les dues primeres paraules les gestiona el sistema i les dues últimes les gestiona l'usuari.



	Núm. Paraula	Byte més significatiu	Byte menys significatiu
Gestió del sistema	%MWk	Núm.d' intercanvi	Bit d'activitat
	%MWk+1	Confirmació de l'operació	Confirmació de la comunicació
Gestió de l'usuari	%MWk+2	Temps d'espera	
	%MWk+3	Longitud	

Taula 21. Estructura de les paraules internes

El bit d'activitat (bit 0) mostra l'estat d'execució de la funció. Es posa a 1 quan comença i es torna a posar a 0 un cop s'ha executat.

El número d'intercanvi és el número que el sistema assigna automàticament durant l'emissió, que permet identificar l'intercanvi. També permet parar l'intercanvi que s'estigui realitzant, amb la funció CANCEL.

#### 5.4.4.1 Confirmació de comunicació i d'operació

La confirmació de comunicació i d'operació formen part dels paràmetres de gestió. La confirmació és comuna a totes les funcions.

La confirmació de comunicació és significativa quan el valor del bit d'activitat passa de 1 a 0.

Valor	Confirmació de la comunicació (byte menys significatiu)
16#00	Intercanvi correcte
16#01	Parada de l'intercanvi en temps d'espera
16#02	Parada de l'intercanvi a petició de l'usuari (CANCEL)
16#03	Format de direcció incorrecte
16#04	Direcció del destinatari incorrecte
16#05	Format del paràmetre de gestió incorrecte
16#06	Paràmetres específics incorrectes
16#07	Problema d'emissió cap al destinatari
16#08	Reservat
16#09	Mida del buffer de recepció insuficient
16#10	L'acoblador de xarxa està absent
16#11	Petició absent
16#12	El servidor d'aplicació ja està actiu
16#13	Número de transacció UNI-TE V2 incorrecte

Taula 22. Primers valors de confirmació de comunicació

<b>Valor</b>	<b>Confirmació de la comunicació (byte menys significatiu)</b>
16#0A	Mida del buffer d'emissió insuficient
16#0B	Absència de recursos de sistema del processador
16#0C	Número d'intercanvi incorrecte
16#0D	No s'ha rebut cap telegrama
16#0E	Longitud incorrecte
16#0F	El servei de telegrama no està configurat
16#10	L'acoblador de xarxa està absent
16#11	Petició absent
16#12	El servidor d'aplicació ja està actiu
16#13	Número de transacció UNI-TE V2 incorrecte
16#FF	Missatge rebutjat

Taula 23. Següents valors de confirmació de comunicació

La funció pot detectar un error en els paràmetres abans d'activar l'intercanvi. En aquest cas, el bit d'activitat es manté a 0, s'inicia la confirmació amb els valors corresponents a l'error.

Aquest byte de confirmació necessita el resultat de l'operació en l'aplicació remota, però només és significatiu si la confirmació de comunicació té els valors 16#00 (intercanvi correcte) o 16#FF (missatge rebutjat).

Segons es confirmi la comunicació amb el valor 16#00 o amb 16#FF, la confirmació de l'operació adquirirà uns valors o un altres.

<b>Valor</b>	<b>Confirmació de l'operació més significativa</b>
16#00	Resultat positiu
16#01	Petició no tractada
16#02	Resposta incorrecte
16#03	Reservat

Taula 24. Comunicació amb valor 16#00

valor	Confirmació de l'operació més significativa
16#01	Falta de recursos cap al processador
16#02	Falta de recurs de línia
16#03	Equip absent o sense recurs. (*)codi únicament manejat per les targetes PCMCIA: TSX FPP20 i TSX FPP 10.
16#04	Error de línia
16#05	Error de longitud
16#06	Via de comunicació defectuosa
16#07	Error de direccionament
16#08	Error d'aplicació
16#0B	Absència de recurs de sistema
16#0C	Funció de comunicació no activa
16#0D	Destinatari absent
16#0F	Problema de direccionament entre estacions o via no configurada
16#11	Format de direccionament no gestionat
16#12	Falta de recurs destinatari
16#14	Connexió no operativa
16#15	Falta de recurs a la via local
16#16	Accés no autoritzat
16#17	Configuració de la xarxa incoherent
16#18	Connexió temporal no disponible
16#21	Servidor de l'aplicació en stop

Taula 25. Comunicació amb valor 16#FF

#### 5.4.4.2 Longitud i temps d'espera

Aquests dos paràmetres estan configurats per l'usuari.

El paràmetre de longitud s'usa per saber el número de caràcters (en bytes) que s'emetrans durant una emissió. També s'usa per memoritzar el número de caràcters (en bytes) rebuts en la recepció d'un missatge. Aquest paràmetre s'ha d'actualitzar abans d'iniciar determinades funcions de comunicació com SEND\_REQ.

El temps d'espera determina el temps màxim d'espera d'una resposta. La seva base de temps és de 100 ms (el valor 0 correspon a un valor d'espera infinit).

Si el temps d'espera ha passat, la comunicació s'acaba amb una confirmació d'error, i també, el sistema rebutja la recepció d'una resposta quan ha passat el temps d'espera.

El valor del temps d'espera d'una funció de comunicació ha de ser suficient per garantir la recepció de la resposta a la pregunta formulada, i variarà en funció de la xarxa i de la càrrega efectiva en el moment de la transacció.

#### **5.4.5 Límits de les funcions de comunicació**

El número màxim de funcions de comunicació que es poden executar a la vegada en els autòmats Micro, és de quatre (via 0).

La mida màxima de les dades trameses està en funció de la via de comunicació, de l'acoblador de la comunicació i el tipus de funció que realitza la funció. En el cas de Modbus, la trama és de 256 bytes com a màxim.

#### **5.4.6 Descripció funcions bàsiques de comunicació**

De funcions de comunicació n'hi ha moltes i diferents, i segons el protocol que s'usa per la comunicació, es fan servir unes funcions de comunicació o unes altres.

En el cas concret del protocol Modbus, les funcions que suporta són: READ\_VAR, WRITE\_VAR, SEND\_REQ i SERVER.

Una comunicació consisteix bàsicament en un intercanvi de preguntes i respostes. Per tal de dur a terme aquesta comunicació, les funcions bàsiques que s'usen són la de llegir objectes estàndard (READ\_VAR) i escriure objectes estàndard (WRITE\_VAR).

La funció SEND\_REQ és més complicada d'utilitzar que les funcions READ\_VAR o WRITE\_VAR, però serveix tant per escriure com per llegir peticions, però no es permet el seu ús en el terminal TER si el que es vol fer és una lectura de bits.

La funció SERVER només es pot usar per tractar les peticions d'un enllaç Modbus en cas d'usar una tarja PCMCIA TSX SCP 114 en un mòdul TSX SCY 21601 configurat en Modbus d'esclau amb servidor immediat. Aquesta funció permet tractar les peticions UNI-TE de forma immediata a partir del programa d'aplicació.

##### **5.4.6.1 Lectura d'objectes estàndard ( READ\_VAR )**

La funció READ\_VAR permet llegir el valor d'un o varis objectes de llenguatge com poden ser: bit intern, paraula interna, bit de sistema, paraula de sistema, paraula

constant, paraula interna doble, paraula constant doble i objectes estructurats (temporitzador, monoestable, comptador, registres, programador cíclic).

Els objectes llegits sempre han de ser consecutius i es poden trobar en una via de comunicació de tipus Modbus.

La resposta ha de contenir un número màxim de bytes que estarà en funció del protocol i del tipus de destinatari. Al final d'una lectura, la longitud de les dades rebudes es memoritza a la paraula 4 del paràmetre de gestió (paràmetre gestionat per l'usuari).

Aquesta funció de comunicació pot llegir fins a 1000 bits consecutius, sigui quin sigui el protocol usat (Modbus o Uni-telway). Si es vol llegir més de 1000 bits s'ha d'usar la funció SEND\_REQ.

La sintaxi de la funció és:

READ\_VAR(Direcció, Tipus d'objecte, N<sup>o</sup> del primer objecte, Número, Valor, Paràmetres de gestió)

Paràmetre	Descripció
Direcció	Direcció del destinatari de l'intercanvi. No es poden posar les direccions {xarxa.estació}APP, {xarxa.estació}APP.núm difusió (ALL).
Tipus d'objecte	Cadena de caràcters que especifica el tipus d'objecte que es llegirà %M : Bit intern %MW : paraula interna
N <sup>o</sup> del primer objecte	Doble paraula que indica l'índex del primer objecte que es llegirà
Número	Paraula que especifica el número d'objectes que es llegiran
Valor	Taula de paraules que conté els valors dels objectes llegits. El número de bits llegits és un múltiple de 8.
Paràmetres de gestió	Paràmetres de gestió de l'intercanvi en curs. La confirmació de l'operació pren un dels valors següents: 16#00 = Lectura correcta 16#01 = Error d'operació 16#02 = Resposta incorrecta 16#03 = Mida de resposta incorrecta

Taula 26. Paràmetres de la funció RED\_VAR

En el programa PL7, la introducció dels tipus d'objectes ha de ser coherent i s'ha de fer, o bé tot en lletres minúscules, o bé, tot en lletres majúscules. Si no es fa així, la funció reenvia 16#06 (paràmetres específics incorrectes).

Exemple:

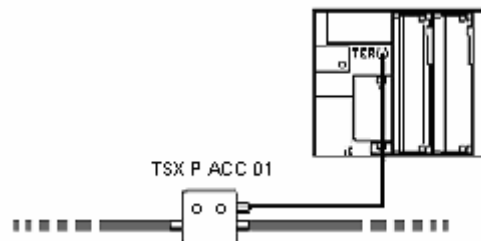


Figura 17. Connexió amb la xarxa

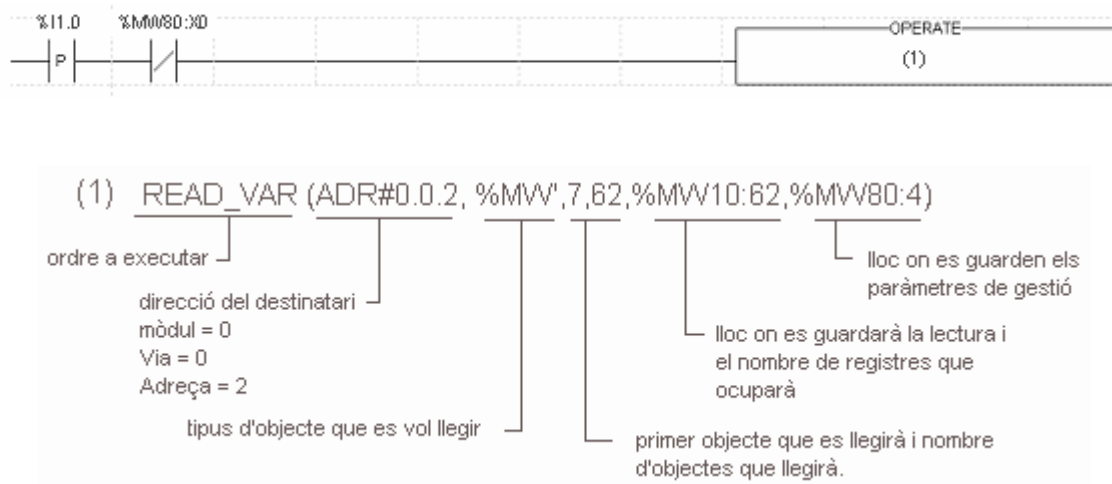


Figura 18. Funció READ\_VAR

Quan no es realitza cap transmissió, el contacte %MW80:X0 es posa a 1. El mestre llegeix les paraules internes %MW7 a %MW62 de l'esclau amb direcció 2. El valor de les paraules llegides es guardarà a la taula de paraules %MW10:62, i els paràmetres de control es guardaran a la taula %MW80.

#### 5.4.6.2 Escriptura d'objectes estàndard ( WRITE\_VAR )

La funció WRITE\_VAR permet escriure un o varis valors d'un o varis objectes de llenguatge del mateix tipus (bit intern, paraula interna, paraula constant, bit de sistema, paraula de sistema, doble paraula interna, doble paraula constant).

Els objectes que s'escriuen han de ser consecutius.

Poden provenir d'una unitat central remota o d'un equip connectat a una via de comunicació de tipus Modbus, Uni-telway, connector terminal, Fipway, Modbus Plus, Ethway.

La sintaxi de la funció és:

WRITE\_VAR (Direcció, Tipus d'objecte, N<sup>o</sup> del primer objecte, Número, Valor, Paràmetres de gestió)

Paràmetre	Descripció
Direcció	Direcció del destinatari de l'intercanvi. No es poden usar les direccions {xarxa,estació}APP, {xarxa,estació}APP.núm difusió (ALL).
Tipus d'objecte	Cadena de caràcters que especifica el tipus d'objecte a escriure: %M : bit intern %MW : paraula interna
N <sup>o</sup> del primer objecte	Doble paraula que indica l'índex del primer objecte que s'escriurà.
Número	Paraula que especifica el nombre d'objectes que s'escriuran. Només s'ha d'inicialitzar després de l'activació de la funció.
Valor	Taula de paraules que conté els valors dels objectes que s'emetraran.
Paràmetres de gestió	Paràmetres de gestió de l'intercanvi. La confirmació de l'operació pren un dels valors següents: 16#00 = Escripura correcta 16#01 = Error d'operació 16#02 = Resposta incorrecta

Taula 27. Paràmetres de la funció WRITE\_VAR

Exemple:

(El connexionat és el mateix que en l'exemple de la funció READ\_VAR).

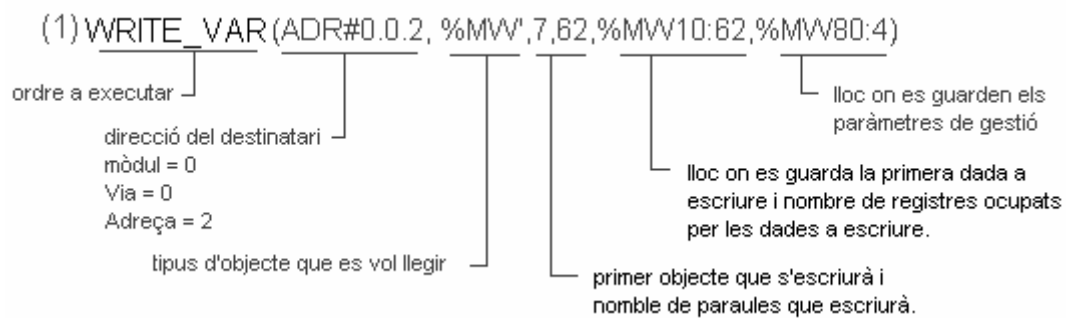


Figura 19. Funció WRITE\_VAR



## **6. EXPLICACIÓ DE LA PRÀCTICA**

La comunicació Modbus és un tipus de comunicació entre autòmats que es pot donar en qualsevol tipus de PLC, és una comunicació standard i que sovint es pot trobar al món de la indústria.

L'objectiu de les pràctiques és introduir a l'alumne en la comunicació Modbus entre autòmats, i es pretén que l'alumne es familiaritzi amb l'estructura d'aquesta comunicació i les funcions que s'usen per dur-la a terme.

Aquesta pràctica tindrà una part inicial on s'explicaran els trets característics de la comunicació Modbus i les funcions específiques de comunicació més bàsiques. A continuació hi haurà una segona part on es mostrarà un exemple de comunicació Modbus (explicat i solucionat, amb un cert grau de dificultat) que servirà a l'alumne com a ajut per realitzar les experiències que se li demanaran.

Les pràctiques que hauran de realitzar els alumnes seran introductòries, i consistirà en una mateixa experiència amb tres apartats de diferent grau de dificultat, de menys a més.

S'ha cregut útil aquest sistema (donar una experiència feta de més dificultat que la que l'alumne haurà de realitzar), per que el que interessa és que l'alumne pugui assolir els conceptes que se li donin , i que pugui completar les experiències en dues o tres sessions.

### **6.1 Connexionat de l'experiència**

Com s'ha dit, hi haurà una experiència d'exemple i una realització pràctica per part de l'alumne. El connexionat per dur a terme tant la primera com la segona part, serà el que es mostra en la figura 13.

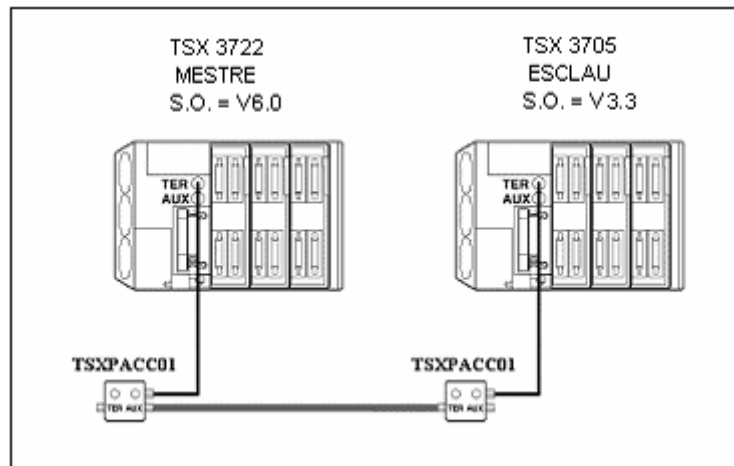


Figura13 . Connexionat experiència

La configuració de les caixes TSX PACC 01 serà la que es mostra a continuació quan estiguin situades al final del bus de comunicació, i si es treballa sense PCMCIA (ja funcioni com a mestre o esclau).

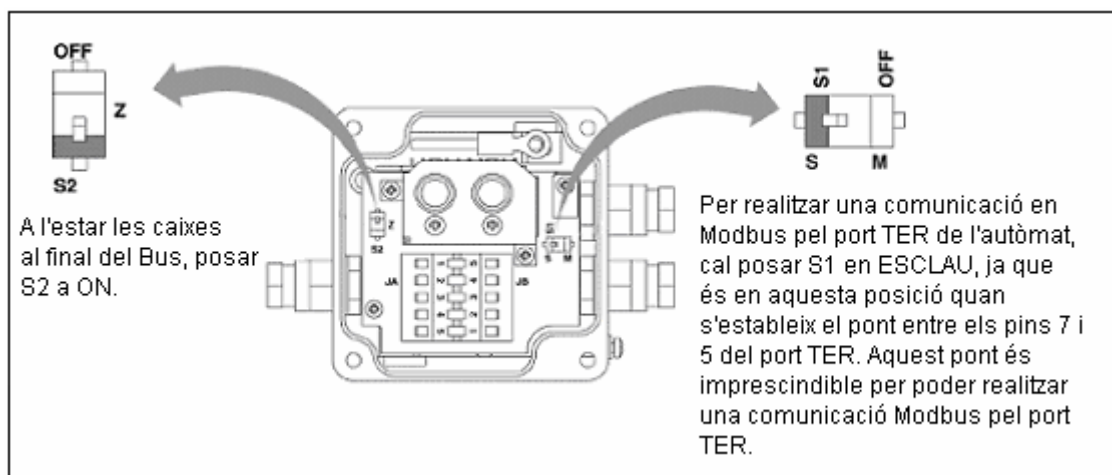


Figura 20 . Configuració de TSX PACC 01

Com que el port TER és l'únic per el que es pot accedir a un PLC des d'un PC per programar-lo, en aquest muntatge no es podrà seguir la comunicació des del PC. Aquest problema es solucionaria mitjançant una tarja PCMCIA, però les pràctiques que s'han plantejat, han estat pensades per tal de que no sigui necessària la tarja (els programes que transferirem al PLC es podran visualitzar amb la maqueta didàctica o bé amb les botoneres de que disposa el laboratori de Regulació).

Aquest connexióat serà vàlid per tota la pràctica, tant per l'experiència exemple com per les experiències a realitzar per l'alumne, l'única diferència serà el nombre de PLC's que s'hi connectaran.

## 6.2 Descripció general de la pràctica d'exemple

L'experiència d'exemple consistirà en simular un procés de buidat i neteja de contenidors de camions. El buidat es fa a la TASCA 1 i la neteja a la TASCA 2.

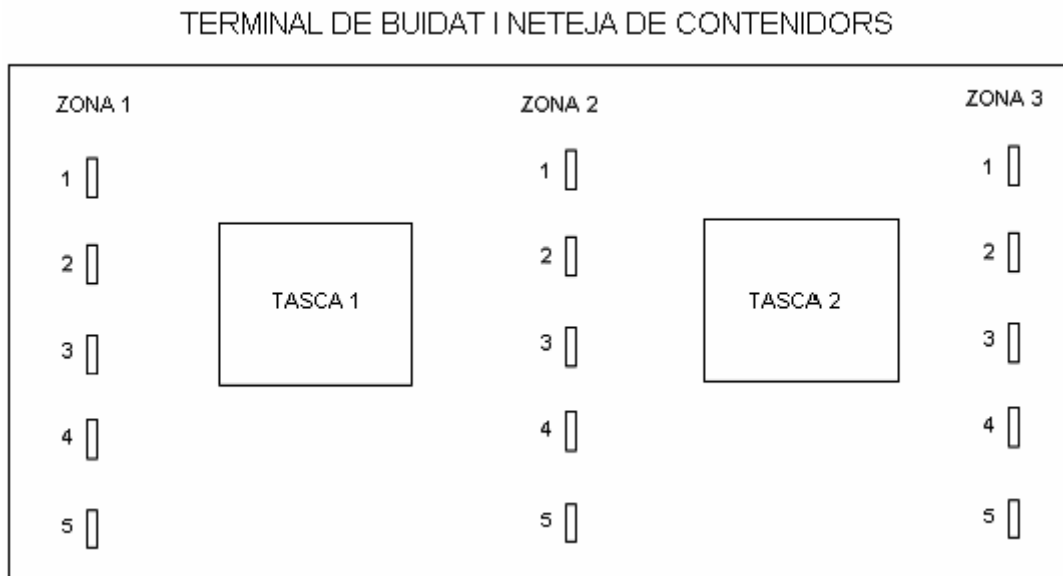


Figura 21. Terminal de buidat i neteja de contenidors

Els camions arriben a la terminal carregats amb productes diferents.

El procés comença quan un camió arriba a la terminal on es fa el buidat i la neteja dels contenidors.

Es considerarà que el camió està a la primera zona (ZONA 1), quan el conductor del camió premi el polsador d'entrada. Això servirà per saber quin camió ha arribat a la terminal. Seguidament, si no hi ha cap camió realitzant la TASCA 1, s'encendrà un led que indicarà al camió que ja pot accedir a la TASCA 1 (buidat dels contenidors) i s'obrirà la barrera d'accés, en cas contrari, s'haurà d'esperar.

Un cop el camió hagi acabat la TASCA 1, aquest es situa a la ZONA 2 (sensor d'entrada de la TASCA 2). Igual com en la ZONA 1, si hi ha algun camió fent la TASCA 2, caldrà esperar a que aquest acabi. La TASCA 2 consisteix en la neteja del contenidor del camió.

Quan el camió hagi fet les dues tasques (buidat i neteja), anirà cap a la sortida de la terminal (ZONA 3). Fins que el camió no arribi a la ZONA 3, no es donarà pas al següent camió.

### 6.3 Emulació amb la maqueta

En aquest apartat s'expliquen les equivalències entre el procés de buidat i neteja dels contenidors i la maqueta.

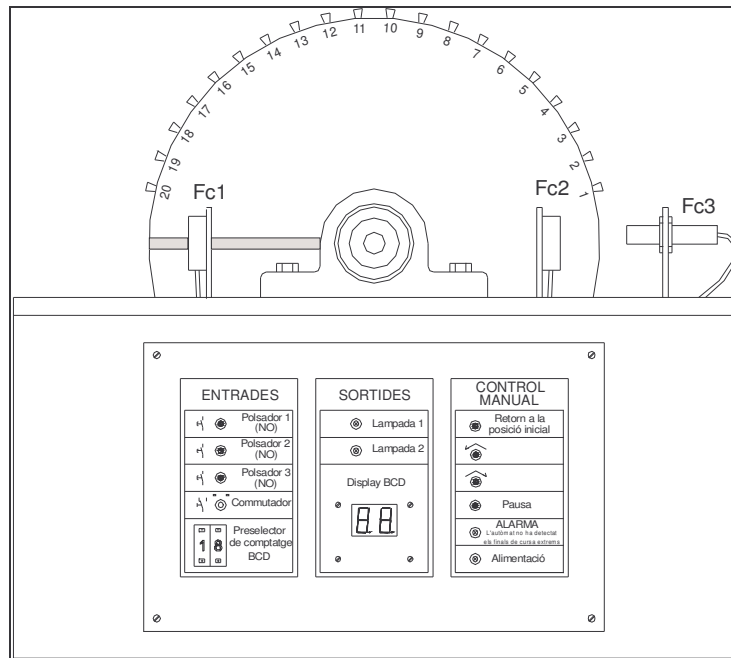


Figura 1. Dibuix de la maqueta didàctica.

El polsador de la ZONA 1 vindrà representat pel polsador 1 (entrada del PLC número 0) i s'encendrà el led Lp1 (sortida del PLC número 4).

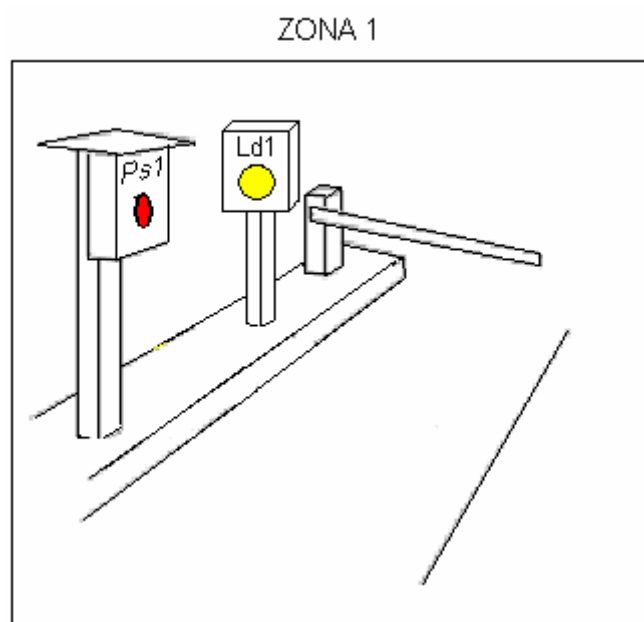


Figura 22. Zona 1 (entrada a la terminal )

Per representar l'acció d'obrir la barrera, Fc3 haurà de comptar tres peces metàl·liques. El sensor inductiu de posició per comptatge (Fc3) correspon a l'entrada del PLC número 7.

La TASCA 1 consisteix en una parada de 2 segons i llavors, el Fc3 avançarà fins a la peça número 7.

Un cop acabada la TASCA 1, el Fc3 comptarà fins a la peça metàl·lica número 10 (ZONA 2 ) s'encendrà el led 2 (sortida 5 del PLC) que serà la mostra de que ja es pot accedir a la TASCA 2.

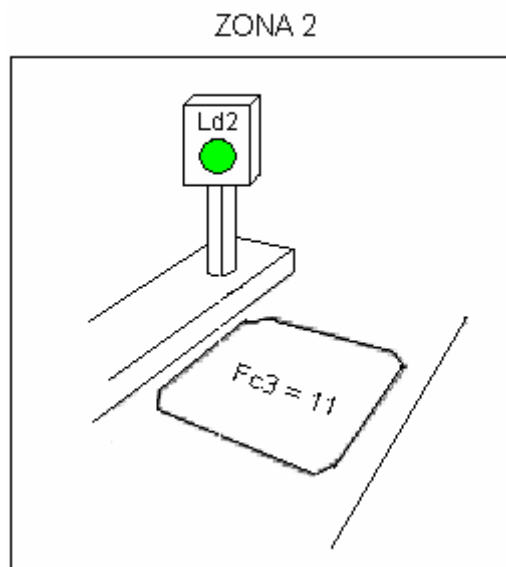


Figura 23. Zona 2 (entrada a la tasca 2 )

La TASCA 2 consistirà en fer una parada de 3 segons i llavors Fc3 comptarà fins a la peça metàl·lica número 15. Quan la TASCA 2 hagi acabat, el camió passa directament a la sortida (ZONA 3).

La ZONA 3 es representa amb que s'avanci fins al fi de cursa Fc2 (entrada número 6 del PLC) i s'aturi. Llavors, el disc girarà fins a Fc1, és a dir, torna a la posició inicial.

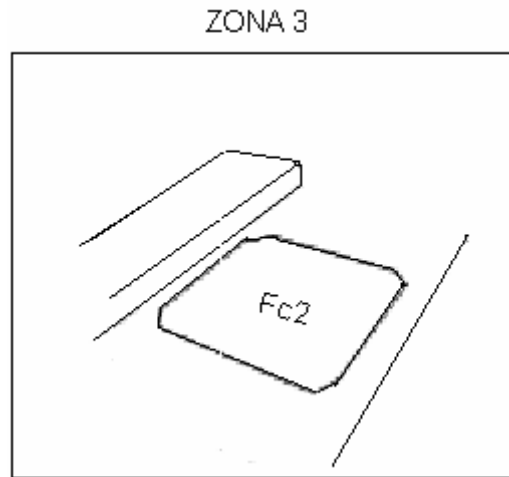


Figura 24. Zona 3 (sortida de la terminal)

#### **6.4 Programa autòmat mestre**

El mestre serà qui gestionarà el pas dels camions a les tasques corresponents (TASCA 1 o TASCA 2).

Per accedir a la TASCA 1, els camions arribaran a la ZONA 1 i, independentment de l'ordre real d'arribada, serà el mestre qui decidirà qui entra primer a fer el buidat del contenidor. El pas dels camions es donarà de forma manual.

En la TASCA 2 però, es crearà un ordre de prioritats (estructura FIFO). Aquest ordre serà el mateix que s'hagi donat en la ZONA 1, és a dir, el primer que accedeixi a la TASCA 1, serà el primer d'entrar a fer la TASCA 2, i fins que no hagi acabat, no podrà accedir-hi cap més camió. El pas del camió, el mestre el donarà de forma automàtica.

##### **6.4.1 Llistat d'entrades i sortides**

El programa autòmat mestre es valdrà de la botonera per tal de donar pas als camions que arribin a la terminal i també per saber en quin estat es troba cada camió.

Els interruptors serviran per donar pas, de forma manual, als camions que hagin d'accedir a la TASCA 1. Els leds serviran per mostrar de forma visual, l'estat dels camions (si ha arribat algun camió a la ZONA 1, si hi ha camions a la ZONA 2 que estiguin en espera, si estan realitzant la TASCA 2, o bé, si han finalitzat tots els processos).

ENTRADA PLC	ETIQUETA	ELEMENT BOTONERA
1	PE1	Interruptor que dona pas a l'esclau 1
2	PE2	Interruptor que dona pas a l'esclau 2
3	PE3	Interruptor que dona pas a l'esclau 3
4	PE4	Interruptor que dona pas a l'esclau 4
5	PE5	Interruptor que dona pas a l'esclau 5
11	ENT	Interruptor per entrar esclau a taula de prioritats
15	INI	Interruptor que activa primer cop lectura esclau

Taula 28. Entrades al PLC ( botonera )

SORTIDA PLC	ETIQUETA	ELEMENT BOTONERA
1	L1A	Led esclau 1 esperant en zona 1
2	L2A	Led esclau 2 esperant en zona 1
3	L3A	Led esclau 3 esperant en zona 1
4	L4A	Led esclau 4 esperant en zona 1
5	L5A	Led esclau 5 esperant en zona 1
6	L1B	Led esclau 1 esperant en zona 2
7	L2B	Led esclau 2 esperant en zona 2
8	L3B	Led esclau 3 esperant en zona 2
9	L4B	Led esclau 4 esperant en zona 2
10	L5B	Led esclau 5 esperant en zona 2

Taula 29. Sortides del PLC (botonera)

#### 6.4.2 Descripció

El control del mestre es farà mitjançant la botonera i també de forma automàtica. Els leds 12, 13, 14 i 15, no es poden fer servir ja que el mòdul d'entrades /sortides que es té instal·lat al laboratori és de 16 E i 12 S. Pel contrari, els interruptors si.

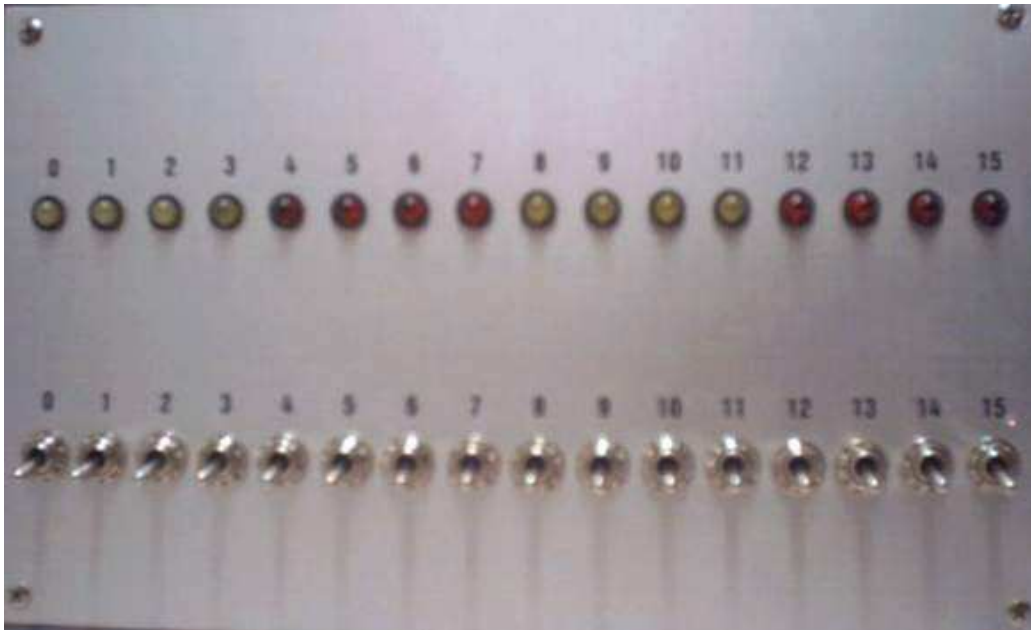


Figura 25. Botonera del laboratori de regulació

Quan un camió arriba a la ZONA 1 i prem el polsador 1, s'encén el led corresponent a l'esclau (L1A per l'esclau 1, L2A per l'esclau 2, etc). El mestre dóna pas a qui creu convenient mitjançant l'interruptor assignat a cada esclau (l'interruptor per l'esclau 1 és PE1, per l'esclau 2 PE2, etc).

Quan el supervisor veu que el llum de l'esclau (LxA) a qui s'ha donat pas (sabrà qui està realitzant la TASCA 1 perquè l'interruptor queda amunt) s'ha apagat, voldrà dir que l'esclau ja ha acabat la TASCA 1. Llavors, ja podrà donar pas al següent esclau. En arribar a la ZONA 2, s'encendrà el led corresponent (L1B per l'esclau 1, L2B per l'esclau 2, L3B per l'esclau 3, etc). En funció de l'ordre de pas que ha donat el mestre en la ZONA 1, es crearà la prioritat d'accés a la TASCA 2, mitjançant una estructura de memòria FIFO. Quan l'esclau estigui realitzant la TASCA 2, el led que li correspon a la ZONA 2, farà intermitent, i s'apagarà en el moment en què el camió acabi la neteja del contenidor i surti de la terminal.

Això permetrà saber en tot moment, de forma visual, quin és l'estat dels esclaus.



### 6.4.3 Programa

El programa s'ha editat mitjançant grafcet. S'ha fet ús de seccions editades en llenguatge de contactes (LD).



Figura 26. Estructura del programa mestre

El programa principal consta de la part que conté els grafquets (Chart) i de la part que conté el programa en llenguatge de contactes (Post).

S'han creat tres subrutines que són: Sr0, Sr1 i Sr2. Les Sr0 i Sr2 serveixen per inicialitzar variables (bits i paraules). La Sr0 conté inicialitzacions que només es faran la primera vegada que s'executi el programa, i la Sr2 conté inicialitzacions que es faran sempre que el programa accedeixi a l'etapa inicial del grafcet principal.

La Sr1 serveix per fer saber quin esclau té pas a la TASCA 2.

#### 6.4.3.1 Secció Chart

Els grafquets que han calgut per a l'autòmat mestre han estat tres. Un dels grafquets és el principal, l'altre és el que serveix per omplir la taula de prioritats i un últim que serveix per llegir la posició inicial dels esclaus, és a dir, llegeix quan un camió ha arribat a la ZONA 1.

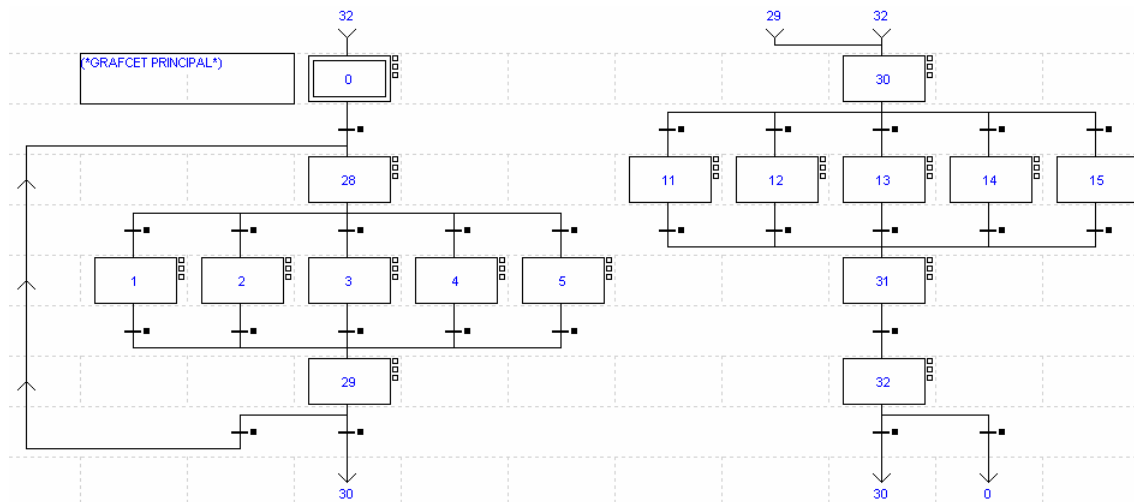


Figura 27. Grafcet principal autòmat mestre

L'etapa 0 és l'etapa d'inici. En les etapes 1, 2, 3, 4 i 5 el que es fa és que, si s'ha encès algun led (camió a la ZONA 1) i si es dona pas mitjançant l'interruptor corresponent, el mestre envia l'ordre de pas a l'esclau i espera fins que l'esclau arriba a la segona posició per donar pas a un altre esclau. Quan això succeeix, s'activa l'etapa 29, que enllaça directament amb l'etapa 30.

Per llegir la posició inicial dels esclaus, s'ha de fer de forma ordenada per tal d'evitar errors de lectura o escriptura. Més endavant s'explicarà de forma més amplia com s'ha fet aquesta forma de lectura.

L'etapa 30 comprova quin esclau té prioritats per entrar a la TASCA 2 i li dona pas.

Les etapes 11, 12, 13, 14 i 15 serveixen per enviar l'ordre de pas als esclaus i llegir quan l'esclau ha acabat.

L'etapa 31 borra el valor que ocupava l'esclau dins la taula de prioritats.

L'etapa 32 serveix per a resetejar alguns bits i per incrementar el punter de lectura (%MW50) de la taula de prioritats.

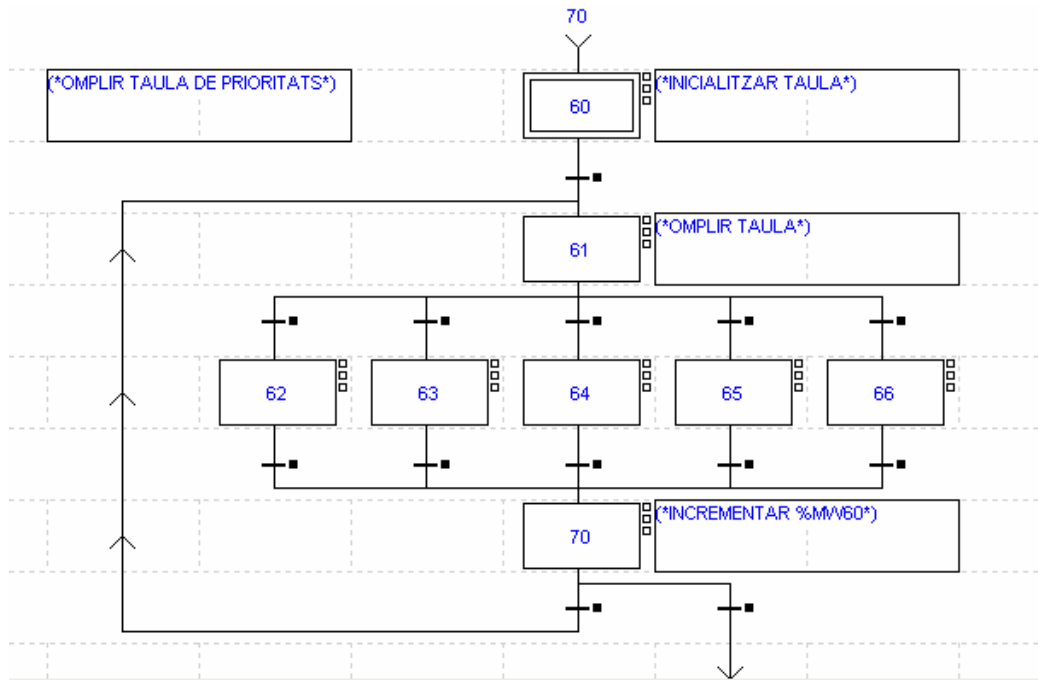


Figura 28. Grafcet omplir taula de prioritats

En l'etapa inicial (etapa 60) s'inicialitza el punter d'escriptura (%MW60) de la taula de prioritats.

De l'etapa 60 a la 61 s'hi accedeix directament.

Quan es dona pas als esclaus mitjançant els interruptors de la botonera, a la vegada, s'ha d'activar l'interruptor 11 per tal d'omplir la taula de prioritats. A l'etapa 61 s'omple la taula a mida que el mestre ha donat pas als esclaus (a la primera part del grafcet principal).

A l'etapa 70 s'incrementa el punter d'escriptura de la taula. Quan la taula estigui plena, es tornarà a l'etapa inicial d'aquest grafcet (etapa 60) on s'inicialitzarà el punter.

Mentre la taula no s'hagi completat, s'anirà augmentant una posició de la taula i s'hi guardarà el següent valor.

La taula de prioritats (escriptura) te la forma següent: %MW500[%MW60].

El punter d'escriptura %MW60 s'inicialitza a 1.

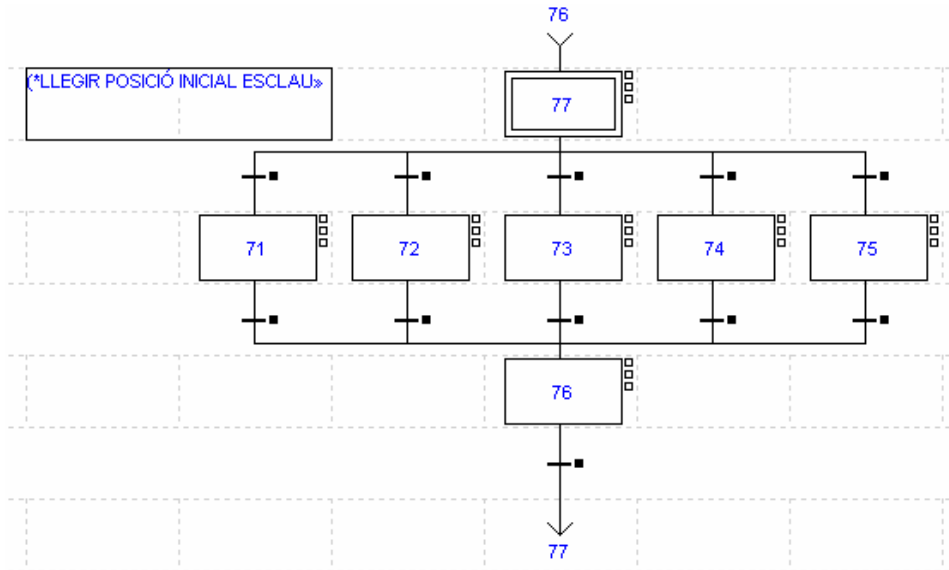


Figura 29. Grafcet llegir posició inicial dels esclaus

A l'etapa 77 s'hi resetejen els bits que serveixen per indicar que ja s'ha llegit la posició de l'esclau.

La lectura de la posició inicial dels esclaus (ZONA 1) s'ha de fer de forma ordenada per evitar conflictes en la comunicació. Primer es llegeix un esclau, llavors un altre, etc. Mentre l'esclau no hagi arribat a la ZONA 1, és a dir, mentre no s'hagi premut el polsador 1, el mestre anirà llegint aquest esclau. Quan el camió estigui a la posició final, és a dir, que surti de la terminal, llavors es tornarà a fer la lectura d'aquell esclau. Si l'esclau està realitzant qualsevol tasca, i mentrestant es prem el polsador 1 de nou, el mestre no reconeixerà com si hagués arribat un nou esclau, el mestre no ho llegirà i, en conseqüència, no s'encendrà el led corresponent a l'esclau. Això es fa per evitar conflictes amb l'esclau (si l'esclau fa alguna tasca o està esperant per fer-la, i per equivocació, es prem el polsador 1, no passarà res).

#### 6.4.3.2 Secció Post

El programa es podria dividir en tres parts que es correspondrien als grafquets. Igual que en els grafquets, els programes s'entrellacen i un necessita de l'altre per tal d'accedir a les etapes corresponents. S'explicarà el programa per ordre d'aparició a la pantalla i no per ordre d'ús, és a dir, si es mira el programa, les primeres instruccions que es veuen no són les que primer s'usen ja que, per exemple, la part de lectura de la primera posició de l'esclau està escrit al final del programa i és la primera cosa que es fa.

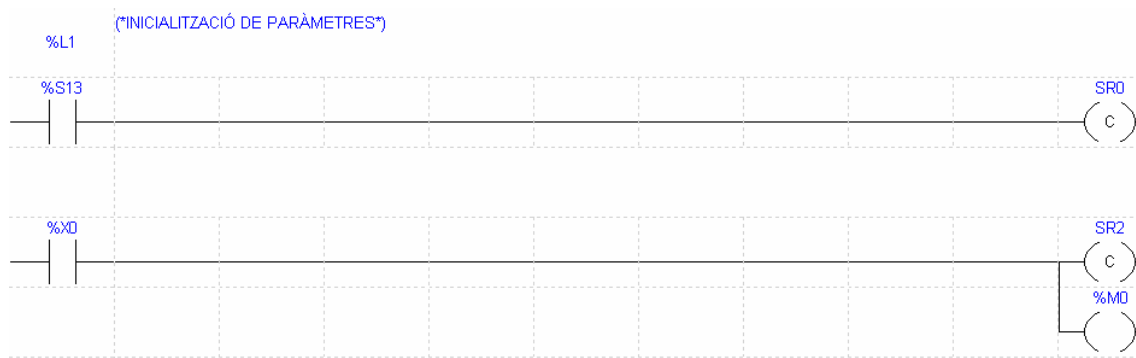


Figura 30. Inicialització de variables

El bit de sistema %S13 només s'activa quan el programa es posa a RUN, per tant, la Sr0 només s'activarà un cop, el primer cop.

Quan el bit associat a l'etapa 0 s'activi, també s'activaran la Sr2 i el bit %M0. El bit %M0 és un bit que serveix per accedir a l'etapa 28 del grafcet principal.

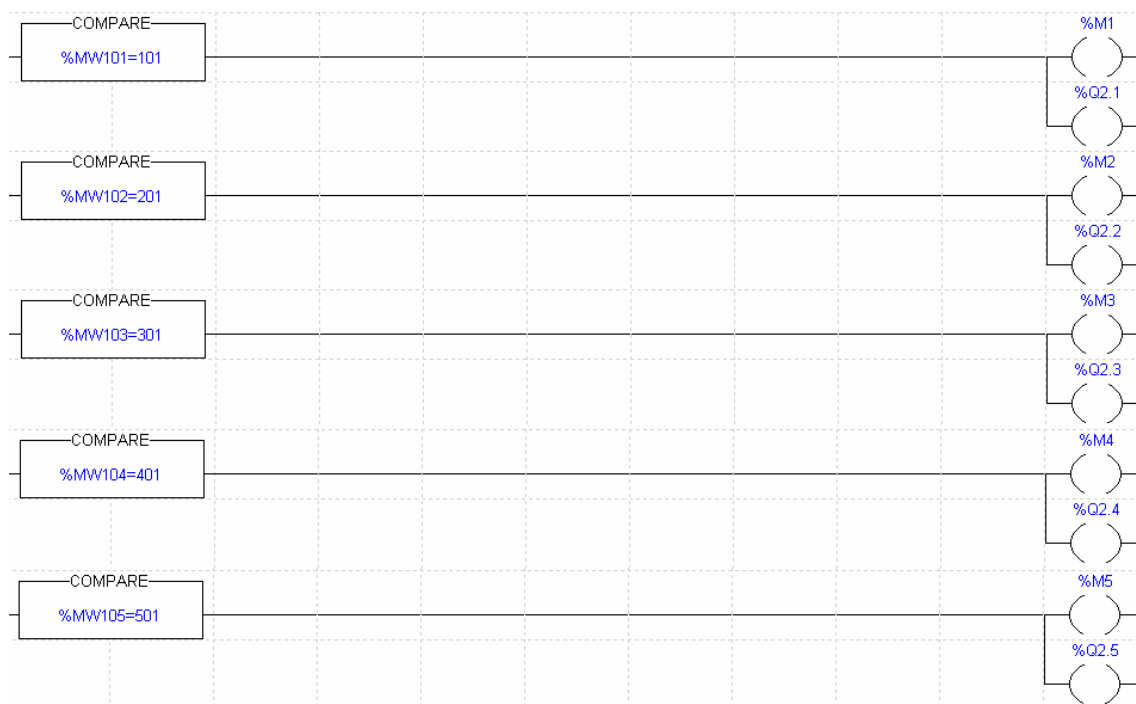


Figura 31. Esclaus a la posició 1

Les paraules %MW101, %MW102 .... %MW105 guarden la posició dels 5 esclaus. Les posicions seran les mateixes per a tots els esclaus (ZONA 1, ZONA 2, ZONA 3 o una zona intermitja que representa que l'esclau està realitzant la TASCA 2), però el que les diferenciarà serà el primer número.

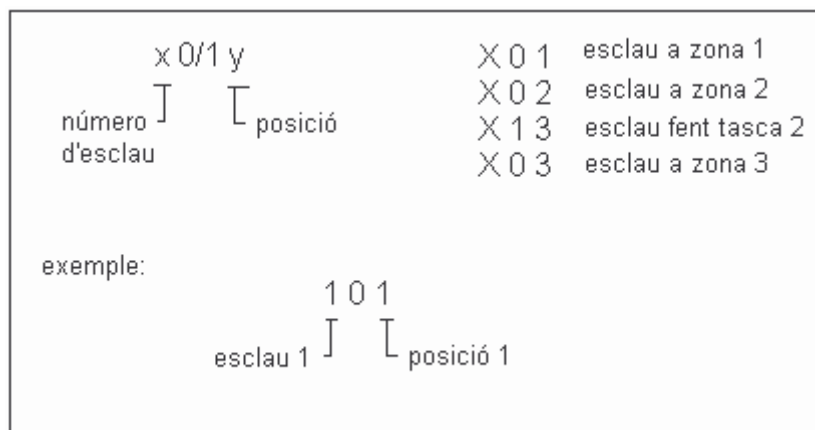


Figura 32. Grafia de les posicions i esclaus

Quan l'esclau estigui a la ZONA 1 de la terminal, s'encendrà el led corresponent i s'activarà el bit que permetrà que el mestre doni pas a l'esclau per realitzar la TASCA 1 quan aquest activi l'interruptor corresponent.

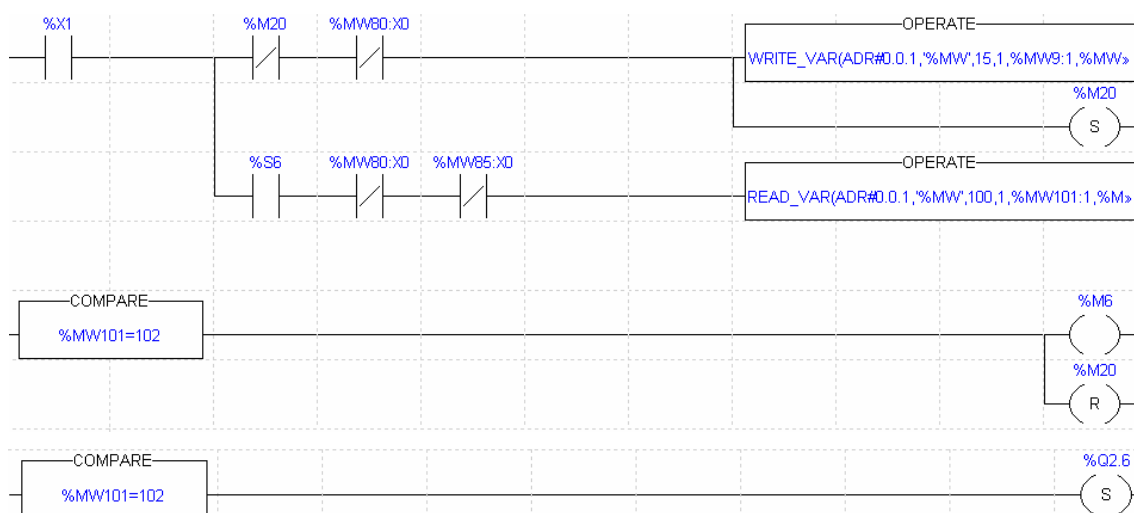


Figura 33. Etapa 1 (grafcet principal)

Quan s'activin %M1 i %I1.1, llavors s'activarà l'etapa 1 del grafcet (%X1). En aquesta etapa el mestre dóna pas a la TASCA 1 a l'esclau 1 enviant-li la paraula %MW15 (que valdrà 1). Aquesta instrucció (WRITE\_VAR) només l'envia una vegada, llavors espera a que l'esclau arribi a la ZONA 2 (READ\_VAR), la qual cosa voldrà dir que l'esclau 1 ja ha acabat la TASCA 1. Quan això succeeixi, s'activa el bit %M6 i es reseteja el bit %M20 per tal de que quan es torni a accedir a l'etapa 1, es pugui tornar a fer tota l'operació.

Pels esclaus 2, 3, 4 i 5 es fa igual, l'única diferència és que, per exemple, per activar l'etapa 2 del grafcet, s'han d'activar %M2 i %I1.2, per l'etapa 3 %M3 i %I1.3, etc.

Llavors, quan els esclaus arribin a la ZONA 2 s'activaran %M7 (per l'esclau 2), %M8 (per l'esclau 3), etc, i s'encendran els leds %Q2.6 (per l'esclau 1), %Q2.7 (per l'esclau 3), fins a %Q2.10 (per l'esclau 5).

El bit de sistema %S6 fa que cada segon es produeixi una lectura de la posició de l'esclau. El mestre llegeix la posició que l'esclau guarda a la paraula %MW100 i ho guarda a la paraula %MW10X (per l'esclau 1: %MW101, per l'esclau 2: %MW102, etc).



Figura 34. Etapa 29 (grafcet principal)

Quan s'activi qualsevol dels bits %M6, %M7... %M10, s'activarà l'etapa 29 del grafcet. Aquesta etapa es podria dir que és de transició. Quan s'activa l'etapa 29, s'activa el bit %M29 el qual permet activar l'etapa 30, i a la vegada, sempre retorna a l'etapa 28 (per tal de que es pugui enviar el pas a la TASCA 1 a un altre esclau).

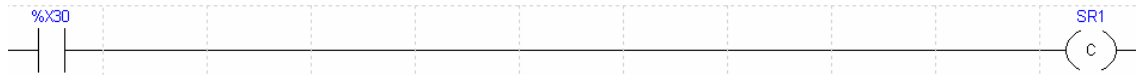


Figura 35. Etapa 30 (grafcet principal)

L'etapa 30 activa la Sr1. Més endavant s'explicarà completament el que fa aquesta subrutina, però per entendre l'etapa 30 cal saber que Sr1 activa els bits %M11 fins %M15 en funció de la zona que ocupen en la taula de prioritats. Aquests bits permetran accedir a les etapes 11, 12, 13, 14 i 15 respectivament.

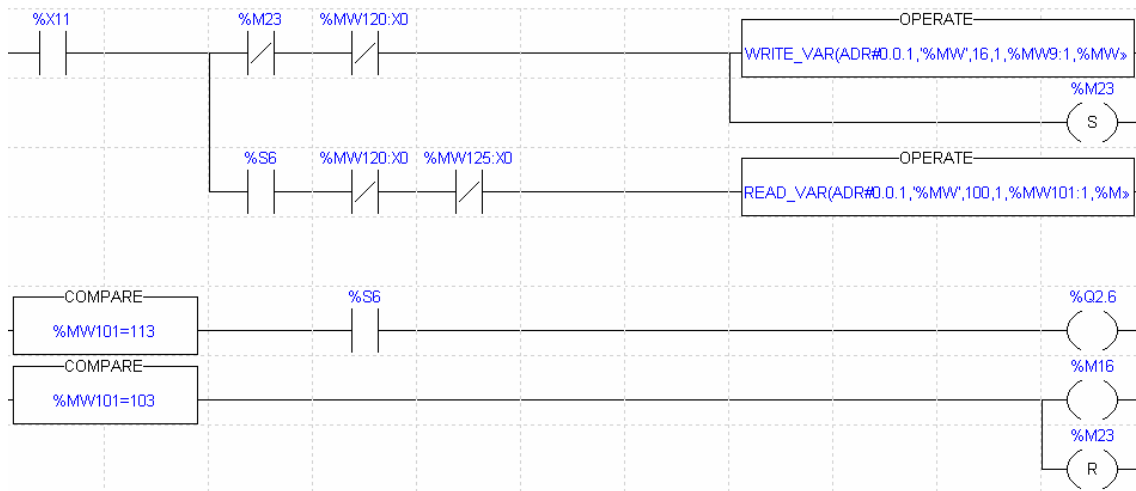


Figura 36. Etapa 11 (grafcet principal)

Quan l'esclau 1 tingui prioritats per accedir a la TASCA 2, s'activarà el bit %M11 (que farà que s'activi l'etapa 11 del grafcet).

En aquesta etapa el que es fa és donar pas a l'esclau a la TASCA 2 enviant a l'esclau la paraula %MW16 (que valdrà 1) amb la funció WRITE\_VAR. Igual que en l'etapa 1, aquesta instrucció només s'envia una vegada.

Quan l'esclau estigui realitzant la TASCA 2, la posició que es llegirà serà "x13". En aquest moment, el led corresponent a cada esclau per a la ZONA 2, es posarà a fer intermitent fins que arribi a la ZONA 3. Llavors el led s'apagarà i voldrà dir que el camió ja ha acabat (s'activarà el bit %M16 que permetrà activar l'etapa 31).

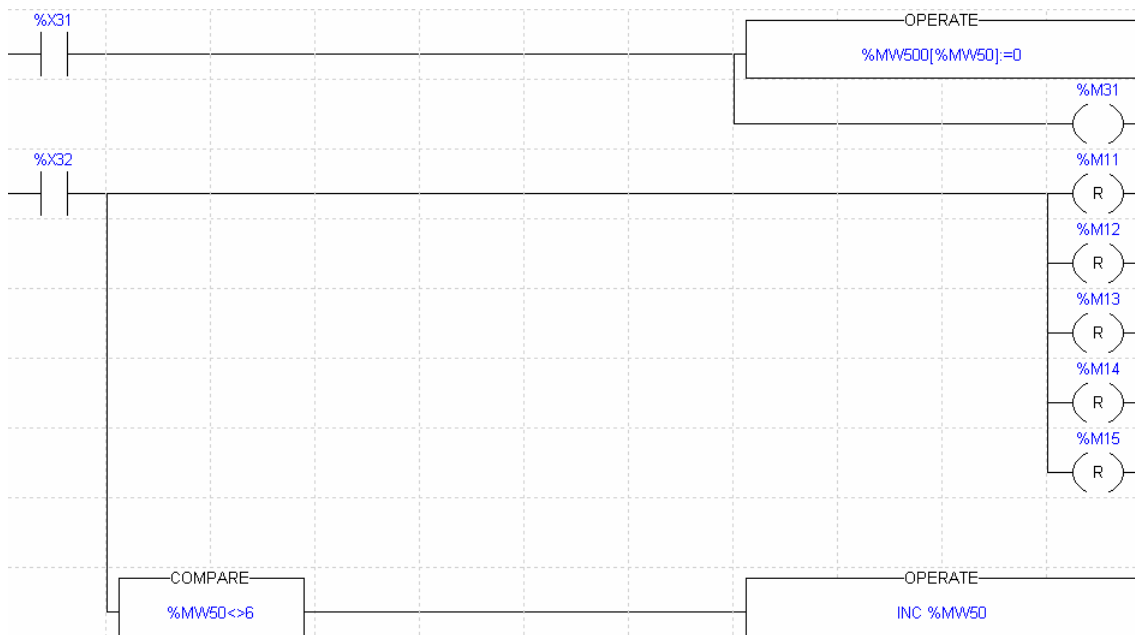


Figura 37. Etapes 31 i 32 (grafcet principal)

L'etapa 31 esborrarà el valor que ocupava l'esclau en la taula de prioritats. El bit %M31 permet activar l'etapa 32 del grafcet.

L'etapa 32 reseteja els bits que permeten accedir a les etapes 11, 12, 13, 14 i 15, i a la vegada, si el punter de lectura de la taula de prioritats (%MW50) és diferent de 6, és a dir, si no s'ha llegit tota la taula, incrementa el punter per tal de llegir la següent posició de la taula.

Si %MW50 és diferent a 6, el grafcet retorna a l'etapa 30, i si és igual a 0, passarà a l'etapa inicial (etapa 0).



Ara s'explica el programa referent a l'emplenat de la taula de prioritats. Aquest programa forma part del programa principal, no és una subrutina.

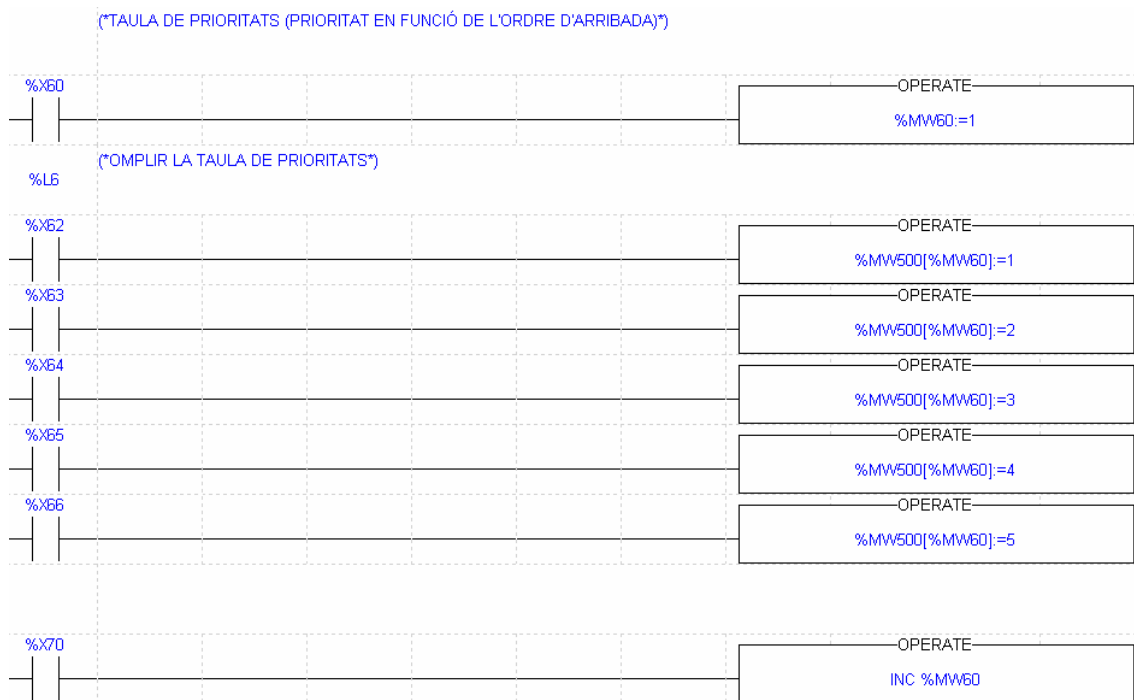


Figura 38. Omplir taula prioritats

L'etapa 60 serveix per inicialitzar el punter d'escriptura de valors a la taula (%MW60). La taula de prioritats consisteix en una taula de paraules de longitud 5 (%MW500:5). Per omplir-la, el que s'ha de fer és, quan el mestre dona pas a un esclau a la TASCA 1 mitjançant l'interruptor corresponent, a la vegada cal que també s'activi l'interruptor %I1.11. Quan s'activen els interruptors %I1.1 i %I1.11, s'activarà l'etapa 62, si s'activen els %I1.2 i %I1.11, s'activarà l'etapa 63 i així, successivament.

El valor que es guarda a la taula és 1, 2, 3, 4 o 5 segons s'hagi donat pas a un esclau o a un altre.

Per exemple: suposem que arriben 3 camions a la ZONA 1 i s'encenen els tres leds corresponents a aquests esclaus. El supervisor donarà pas a l'esclau que li sembli, sense tenir en compte quin led s'ha encès primer. L'ordre que es doni per accedir a la TASCA 1 serà el que es mantindrà per accedir a la TASCA 2, amb l'única diferència que per accedir a la TASCA 2, el pas es donarà de forma automàtica.

En l'etapa 70 s'incrementa el punter d'escriptura %MW60. Si aquest punter és diferent de 6 retorna a l'etapa 61, i si és igual a 6, o sigui, si la taula està plena, torna a l'etapa 60.

## Creació d'una xarxa de comunicacions modbus pels autòmats telemecanique TsxMicro

La part del programa que llegeix la posició inicial dels esclaus és una mica complex d'entendre. Per facilitar la comprensió, primer s'explicarà la lectura pròpiament i llavors s'explicarà l'activació dels bits que ho permeten.

La lectura de la primera posició s'haurà de fer de forma ordenada. Primer es llegirà un esclau i llavors un altre. Això es fa així per evitar conflictes entre els esclaus i assegurar que tots els esclaus rebin les ordres que els corresponen.

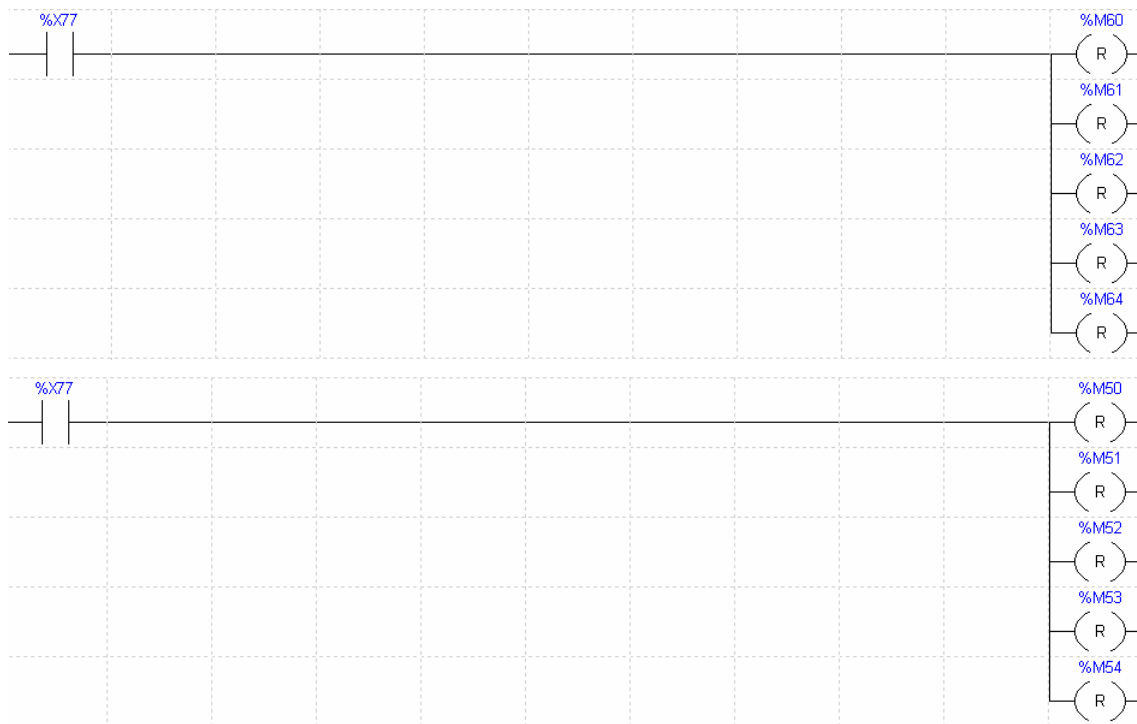


Figura 39. Etapa inicial (grafcet llegir primera posició esclaus)

L'etapa 77 és l'etapa inicial del grafcet on es llegeix la primera posició dels esclaus, és el grafcet que avisa al mestre de que ha arribat un camió a la terminal.

Aquí es resetejen els bits que permeten la lectura (%M60 fins %M64) i els bits que fan que s'activin de l'etapa 71 fins a la 75 (%M50 fins %M54).

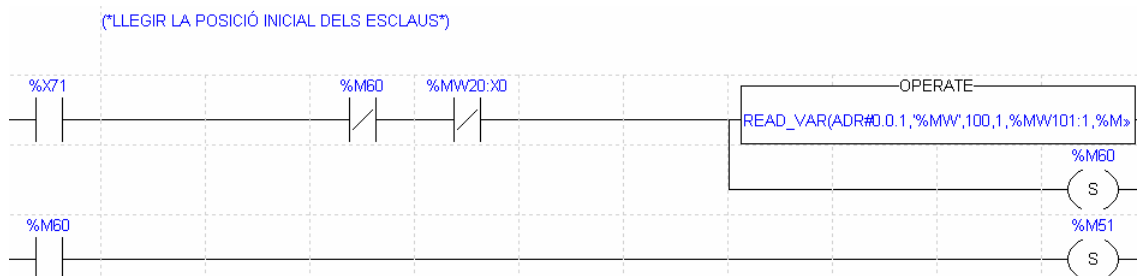


Figura 40. Lectura posició inicial esclaus



de 6 segons). Cal que la durada de l'impuls permeti que els altres timers tinguin temps d'activar-se.



Figura 42. Lectura ordenada dels esclaus (2)

Quan %TM3 hagi comptat els 6 segons, activarà el bit %M201 i resetejarà el bit %M200. Llavors, quan el bit %M201 s'activi, el %TM4 comptarà 2 segons i s'activarà de nou el bit %M200 i resetejarà el %M201.

Aquesta acció permetrà fer que el timer %TM3 s'activi i torni a començar el procés de lectura de la primera posició dels esclaus.

%TM4 es configura a 2 segons per tal de que els bits tinguin temps d'activar-se i desactivar-se.

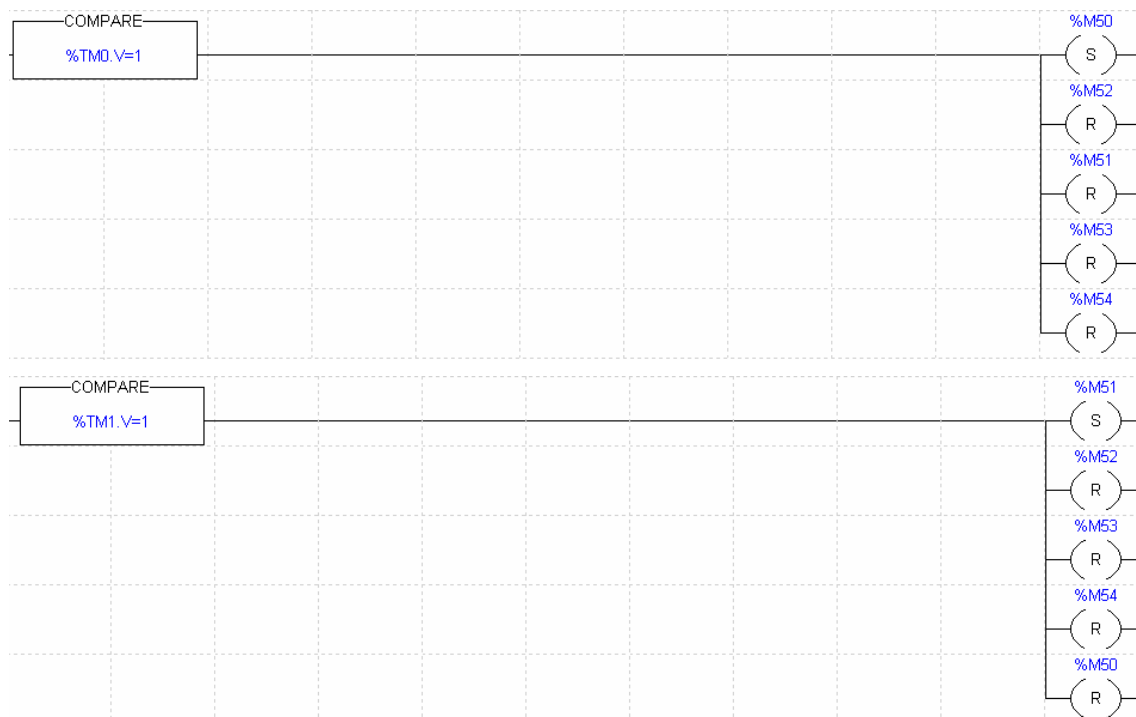


Figura 43. Activació i desactivació de bits %TMO i %TM1

Quan els timers %TM0, %TM1, %TM2, %TM5 i %TM6 han comptat el temps predeterminat, s'activarà un bit un o un altre i resetejarà els altres per evitar que no es llegeixi cap esclau que no sigui el que toca.

Així doncs, quan %TM0.V = 1 s'activarà %M50, quan %TM1.V = 1 s'activarà %M51, quan %TM2.V = 1 s'activarà %M52, etc.

Cal recordar que aquests bits (%M50 fins %M54) són els que permeten activar les etapes 71, 72, 73, 74 i 75, que són les que fan la lectura pròpiament.

### 6.4.3.3 Sr0 : Inicialització de variables

La Sr0 té la funció de realitzar inicialitzacions de variables però que només es faran una vegada, quan es posi a RUN el programa i prou.

Es resetejen els bits que permeten accedir a les etapes 1, 2, 3, 4 i 5 on es dona pas als esclaus per realitzar la TASCA 1.

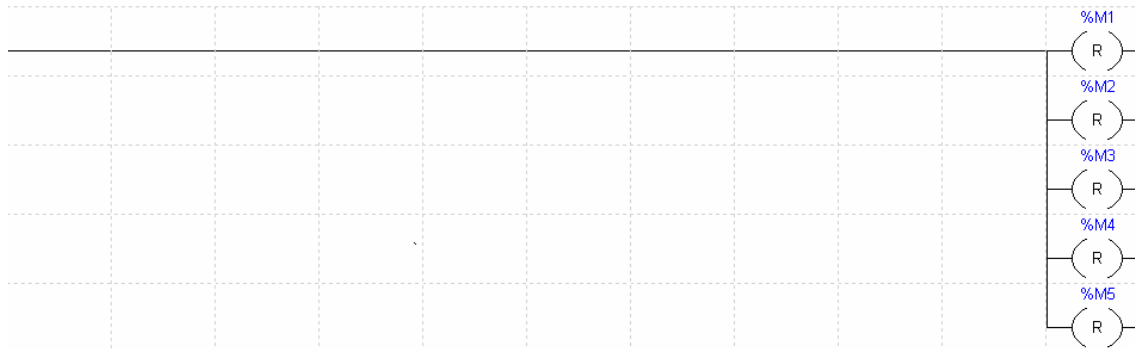


Figura 44. Inicialització de bits de pas a TASCA 1

Es resetejen els bits que donen per acabades les etapes 1, 2, 3, 4 i 5.

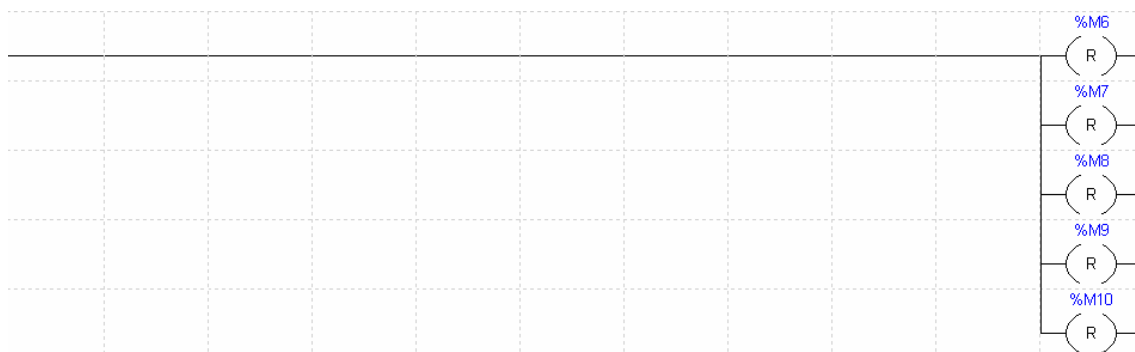


Figura 45. Inicialització de bits de fi de TASCA 1

A la Sr0 s'hi inicialitzen les paraules i punter que són necessaris per donar pas als esclaus.

%MW15 i %MW16 no caldria inicialitzar-los a 1 perquè quan el mestre escriu a l'esclau i li envia aquestes paraules, el que fa és enviar lo que conté la paraula %MW9 i ho envia amb la paraula %MW15 o %MW16, però per assegurar l'operació, es fa la inicialització a 1.

Recordar que la variable %MW50 és el punter de lectura de la taula de prioritats.

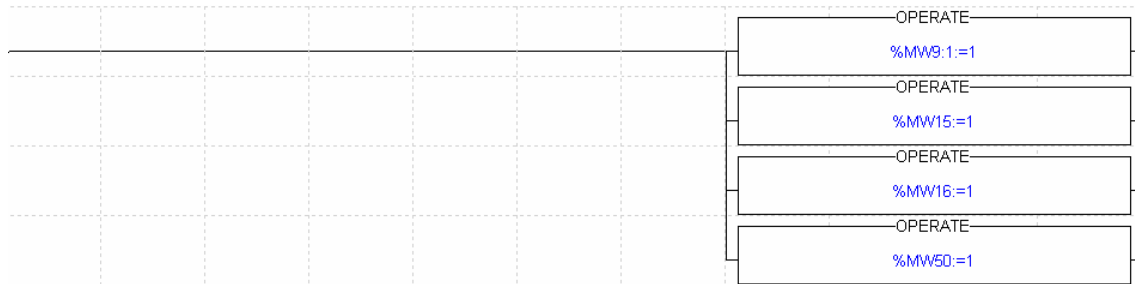


Figura 46. Inicialització de paraules i punter

Es resetejen els bits que permeten enviar a l'esclau la paraula de pas per accedir a la TASCA 1.

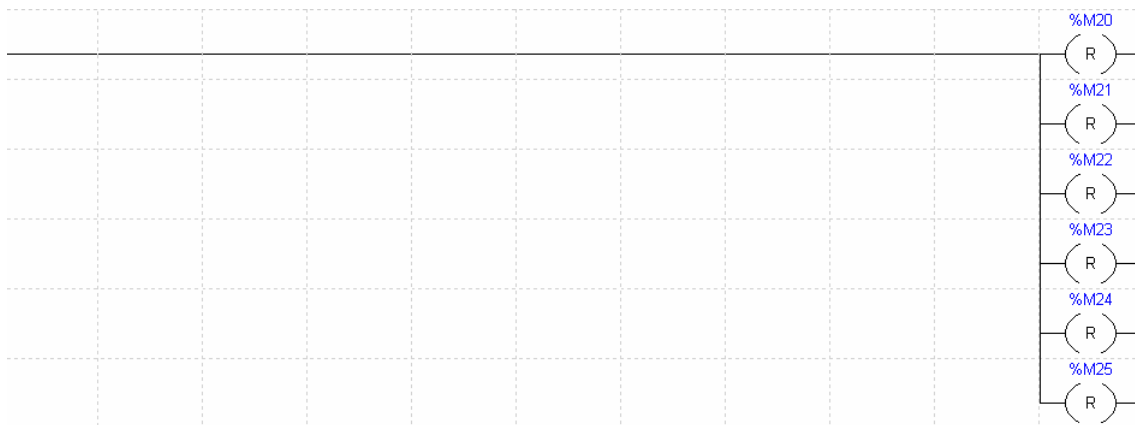


Figura 47. Inicialització de bits d'escriptura

Es resetejen els bits que permeten accedir a les etapes 11, 12, 13, 14 i 15 del grafcet principal, i els bits que assenyalen que els esclaus han arribat a la ZONA 3.

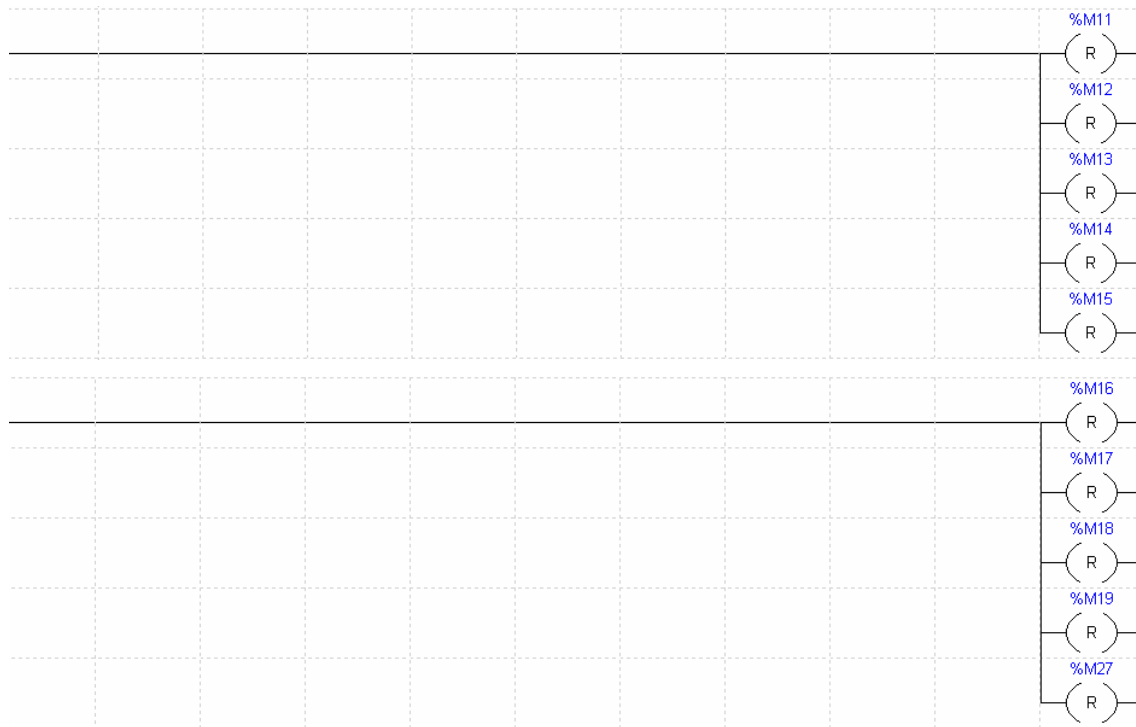


Figura 48. Reset dels bits de pas a TASCA 2 i esclau a ZONA 3

També es resetejen els bits que donen per acabada la lectura dels esclaus en la primera posició.

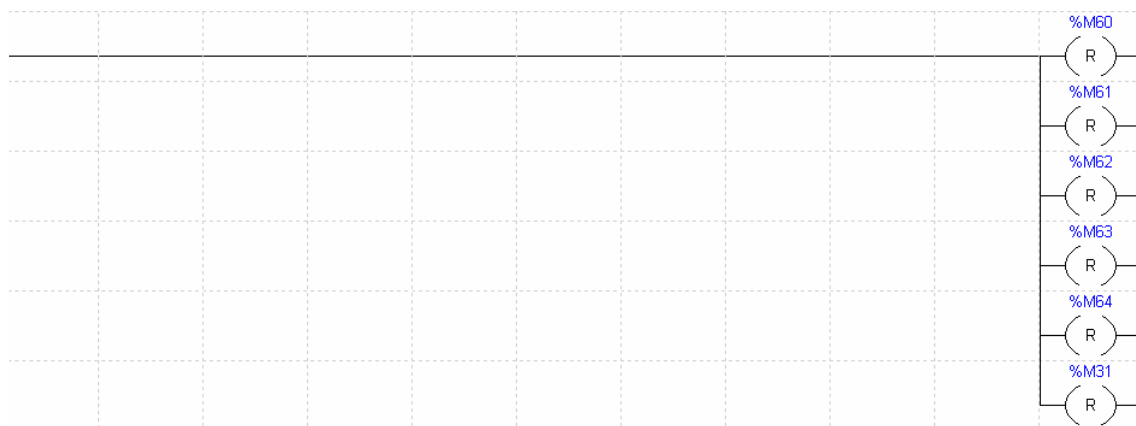


Figura 49. Reset dels bits de lectura posició 1 dels esclaus

#### 6.4.3.4 Sr1: Activació de bits segons taula de prioritats

Quan s'activa l'etapa 30 del grafcet principal, s'activa la Sr1. En aquesta part del grafcet principal, el mestre dóna pas als esclaus en funció de la taula de prioritats.

Aquest pas es dóna de manera automàtica segons l'ordre que ocupin els esclaus dins la taula de prioritats.

Per exemple, si al primer lloc de la taula hi ha guardat un 3, el bit que s'activarà serà el %M13, el qual permetrà activar l'etapa 13 on s'envia el pas a l'esclau 3.

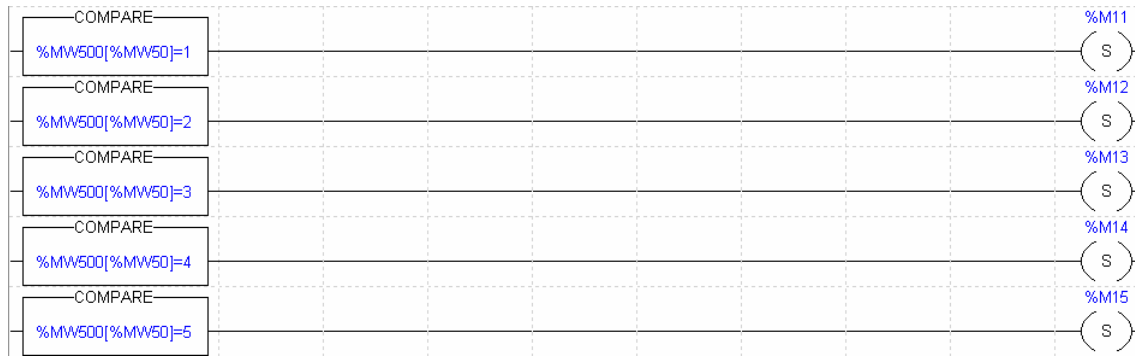


Figura 50. Activació de bits segons taula de prioritats

#### 6.4.3.5 Sr2 : Inicialització de variables

La Sr2 també serveix per inicialitzar variables, però aquestes inicialitzacions es faran cada vegada que s'activi l'etapa 0 del graficet principal.

Es resetejen els bits usats per l'escriptura als esclaus, els bits per accedir a etapes i bits que serveixen per fer saber al mestre que han arribat a una zona en concret.

Això es fa per tots els esclaus però la figura 51 només mostra les variables de l'esclau 1. No s'ensenyen les variables dels altres esclaus perquè seria molt extens i ja s'han esmentat anteriorment.



Figura 51. Inicialització dels bits referents a l'esclau 1



El punter de lectura de la taula de prioritats també s'inicialitza a 1 per tal de situar-lo a la primera posició de la taula.

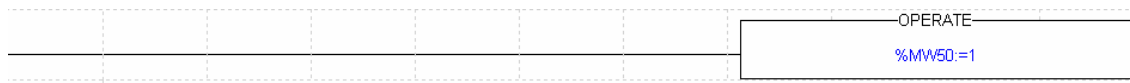


Figura 52. Inicialització punter lectura taula de prioritats

## 6.5 Programa autòmat esclau

El programa de l'autòmat esclau simularà el procés amb el suport de la maqueta didàctica. Fent girar la maqueta representarà els diversos punts del procés, i haurà d'esperar a rebre els permisos del mestre per fer una cosa o una altra. El programa dels autòmats esclaus serà el mateix per a tots, amb l'única diferència que les posicions dels esclaus en les diverses parts de la simulació seran diferents per a cada un d'ells.

### 6.5.1 Llista d'entrades i sortides

La simulació del procés es fa mitjançant la maqueta didàctica.

ENTRADA PLC	ETIQUETA	ELEMENT MAQUETA
0	Ps1	Polsador 1
5	Fc1	Fi de cursa indicant posició esquerra del disc
6	Fc2	Fi de cursa indicant posició dreta del disc
7	Fc3	Sensor inductiu de posició de comptatge

Taula 30. Entrades al PLC (maqueta).

SORTIDES PLC	ETIQUETA	ELEMENT MAQUETA
0	Drt	Moviment a dretes
1	Esq	Moviment a esquerres
2	Lnt	Velocitat lenta
3	Rpd	Velocitat ràpida
4	Lp1	Làmpada 1
5	Lp2	Làmpada 2

Taula 31. Sortides del PLC (maqueta).

### 6.5.2 Descripció

El procés comença amb el disc a la seva posició inicial, és a dir, a l'esquerra (Fc1 = 1). En prémer el polsador 1 (Ps1) de la maqueta, el mestre rep el senyal de que hi ha un esclau a la ZONA 1 i que està a punt per entrar a fer la TASCA 1.

Si no hi ha cap esclau realitzant la primera tasca, l'esclau rebrà l'ordre de pas del mestre (%MW15 = 1) i s'encendrà el led 1 (Lp1) que indicarà el pas de forma visual. Llavors, el disc haurà de comptar 3 peces metàl·liques (per simular que la barrera s'aixeca i arriba a la TASCA pròpiament), parar durant 2 segons i s'haurà d'apagar el led 1. Passats els 2 segons, el disc girarà i comptarà 5 peces metàl·liques més, i llavors pararà 1 segon. Fet això, ja haurà acabat la TASCA 1.

El disc girarà i es comptaran 4 peces metàl·liques més, s'aturarà i esperarà a rebre l'ordre del mestre (%MW16 = 1). En aquest moment, l'esclau estarà situat a la ZONA 2, esperant per entrar a realitzar la TASCA 2.

Quan %MW16 = 1, el led 2 (Lp2) s'encendrà i el disc girarà comptant 6 peces més. S'aturarà 3 segons i el led 2 s'apagarà. El disc haurà de girar fins que el fi de cursa Fc2 el detecti i llavors es considerarà que la TASCA 2 haurà finalitzat.

Per tornar a la posició inicial el disc haurà de girar cap a l'esquerra i a velocitat ràpida. Quan el sensor de posició esquerra del disc (Fc1) detecti, haurà de parar.

L'esclau podrà prendre 4 possibles posicions que permetran al mestre saber on i com es troba en qualsevol moment. Aquestes posicions seran les mateixes per a tots els esclaus però no s'escriuran igual. Això es fa per evitar confusions al mestre.

SIMBOLOGIA	ZONA
X 0 1	ZONA 1
X 0 2	ZONA 2
X 1 3	ZONA INTERMITJA
X 0 3	ZONA 3

Taula 32. Simbologia de les zones

La "X" es substituirà a cada programa esclau per l'esclau corresponent. Per l'esclau 1 les posicions seran 101, 102, 113 o 103, per l'esclau 2 seran 201, 202, 213 o 203, etc.

### 6.5.3 Programa

El programa dels esclaus és senzill i, tot i que s'ha fet ús del grafcet, el grafcet de l'esclau no és necessari. El programa és el mateix per a tots però canvien les posicions i les direccions dels esclaus.



Figura 53. Estructura del programa esclau.

#### 6.5.3.1 Secció Chart

Pel programa esclau no és necessari fer el grafcet però, per tal de facilitar la comprensió del procés, s'ha escrit el grafcet. El grafcet només serveix per veure de manera més clara el passos que segueix l'esclau però el programa no s'hi referència. El grafcet és el mateix per a tots els esclaus.

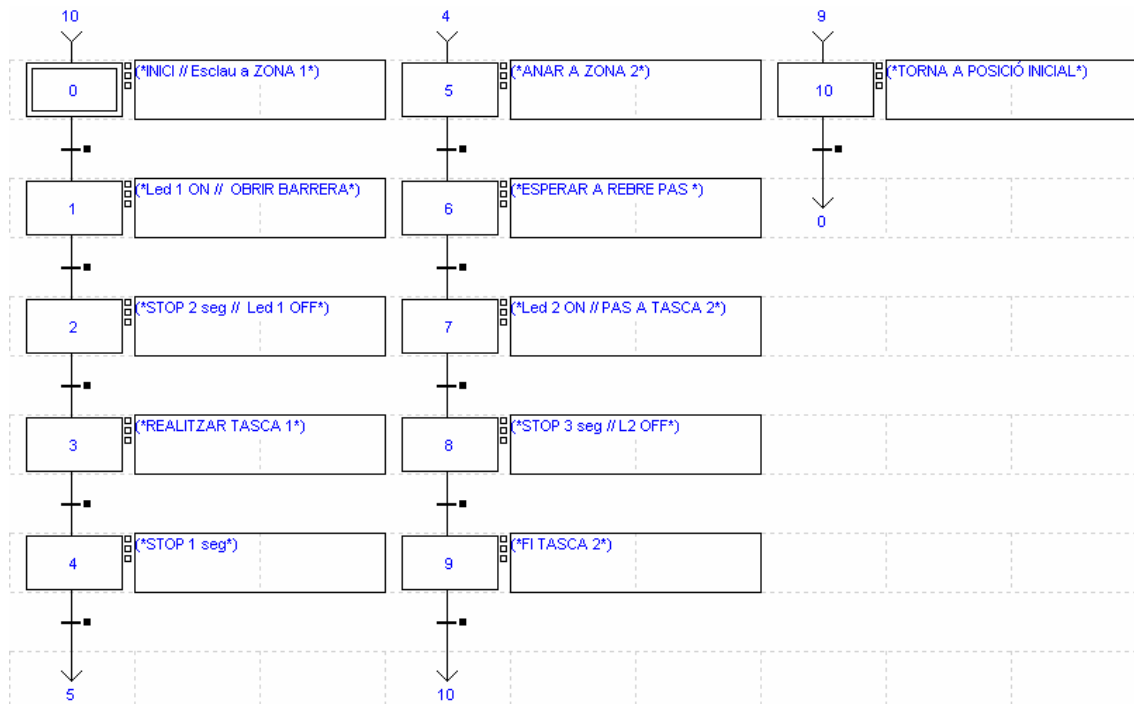


Figura 54. Graficet dels esclaus.

### 6.5.3.2 Secció Post

El programa esclau s'ha escrit en llenguatge de contactes (LD).

Primer de tot, cal assegurar-se que el disc de la maqueta estigui a la posició inicial.

En començar el programa s'inicialitzen les paraules que donen pas a l'esclau per accedir a la TASCA 1 i a la TASCA 2. Quan aquestes paraules valguin 1, l'esclau realitzarà les tasques.

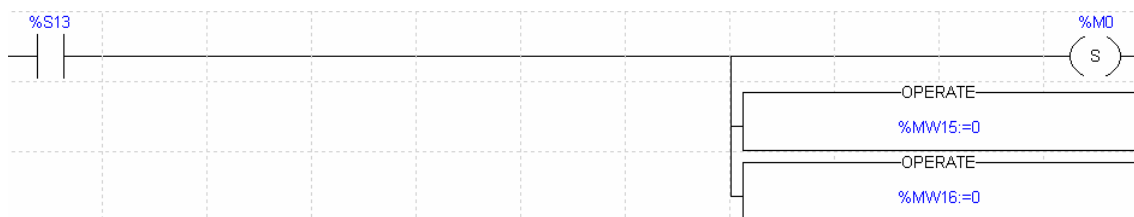


Figura 55. Inicialització de variables de l'esclau

En activar-se el bit %M0 s'iniciarà el procés i permetrà resetejar tots els comptadors.

Quan es premi el polsador Ps1 (%I1.0), l'esclau estarà a la primera posició de la terminal de buidat i neteja de contenidors. La posició que es mostra a la Figura 56 correspon l'esclau 1. Per l'esclau 2 la posició serà 201, 301 per l'esclau 3, etc.

La paraula %MW100 és on es guarda el valor de la posició dels esclaus i aquesta paraula és igual per a tots els esclaus.

Quan el mestre doni pas a l'esclau (%MW15 = 1), s'encendrà el led 1 i el disc girarà a dretes i a velocitat lenta.

Quan s'activi %M1, la paraula de pas per accedir a la TASCA 1 es posarà a 0.

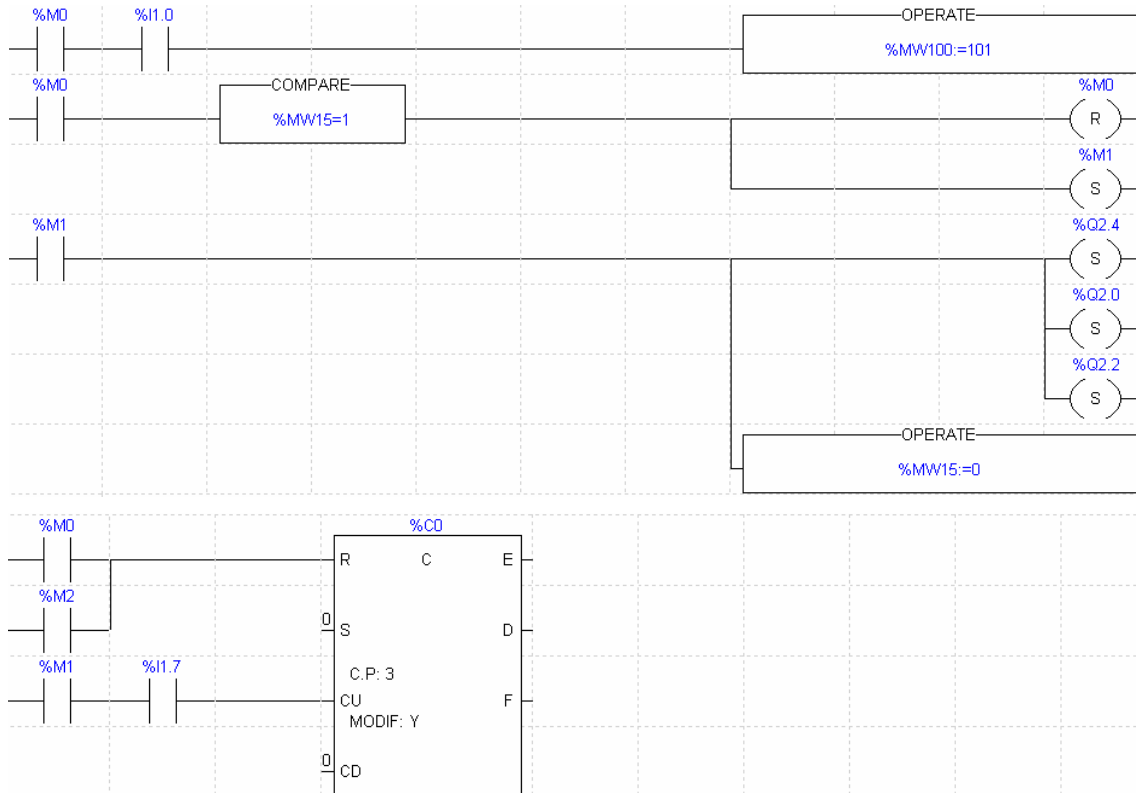


Figura 56. Accedir a la TASCA 1

El comptador %C0 s'activarà i comptarà 3 peces metàl·liques. El comptador es resetejarà quan s'activin els bits %M0 o %M2.

Quan s'hagin comptat les 3 peces, el disc s'atura durant 2 segons (%TM0).

Passats els 2 segons es reseteja el bit %M2 i s'activa el %M3. Aquest darrer bit permetrà que l'esclau faci la TASCA 1.

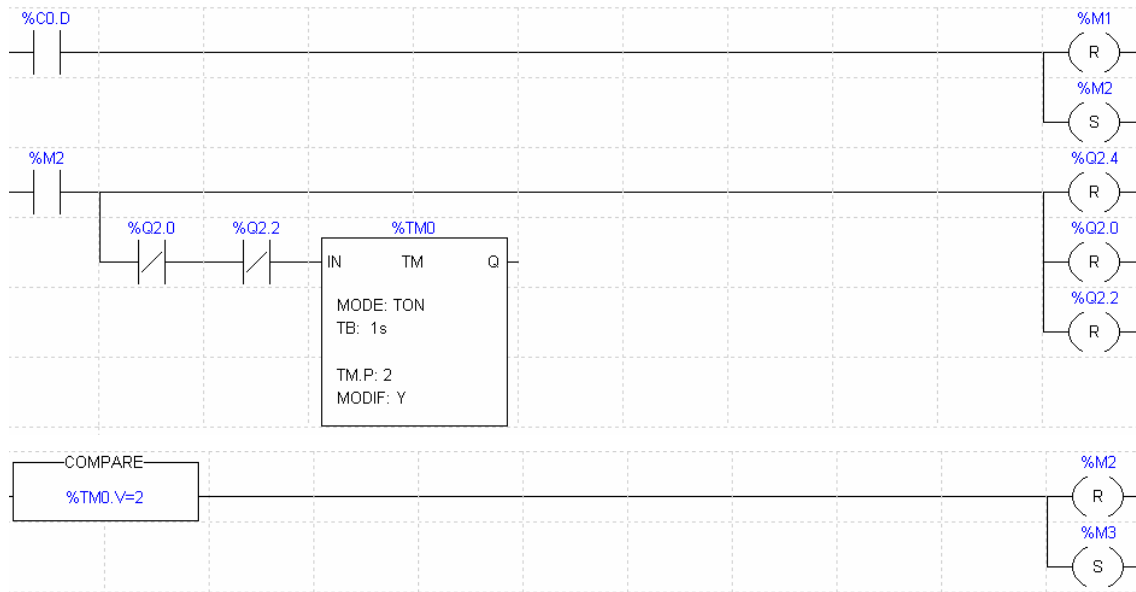


Figura 57. Entrar a realitzar TASCA 1

La TASCA 1 consisteix en comptar 5 peces metàl·liques i fer una parada d'un segon.

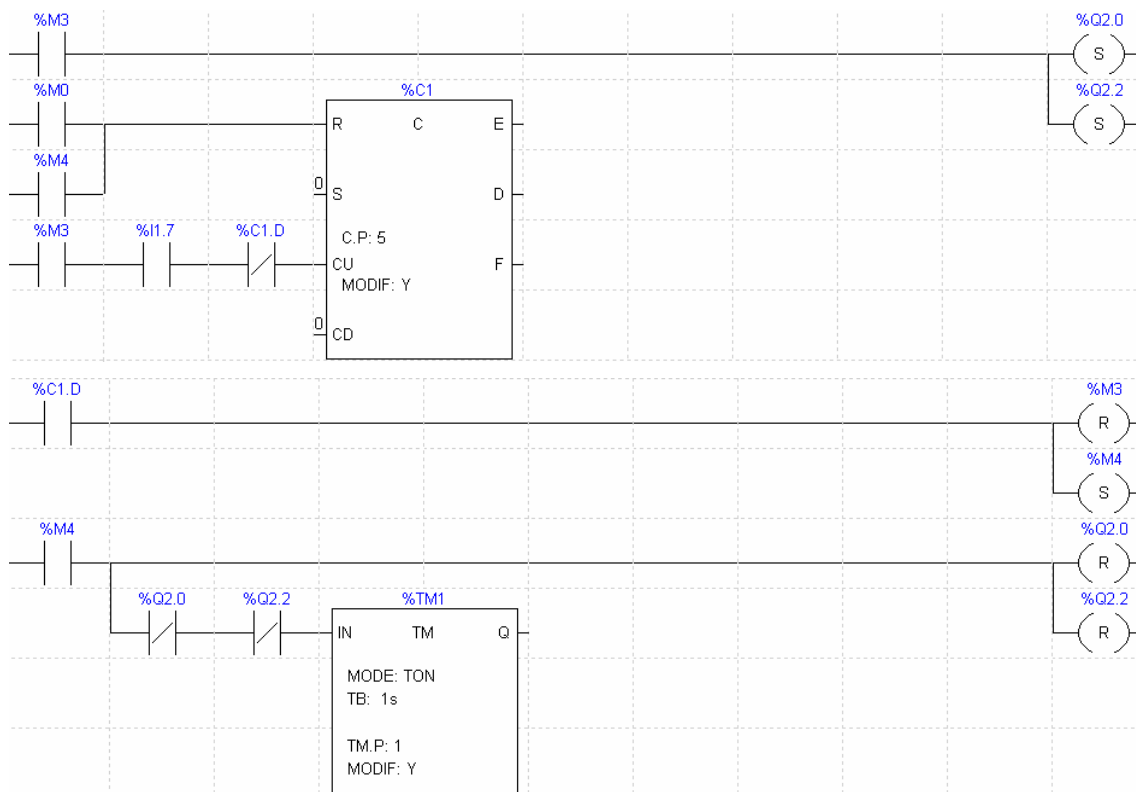


Figura 58. Realització de la TASCA 1

Quan l'esclau hagi simulat la neteja del contenidor s'haurà de dirigir cap a la ZONA 2 (zona on l'esclau haurà d'esperar fins que no hi hagi cap esclau realitzant la TASCA 2 per tal d'accedir-hi).

Per simular-ho, el comptador %C2 comptarà 4 peces metàl·liques.

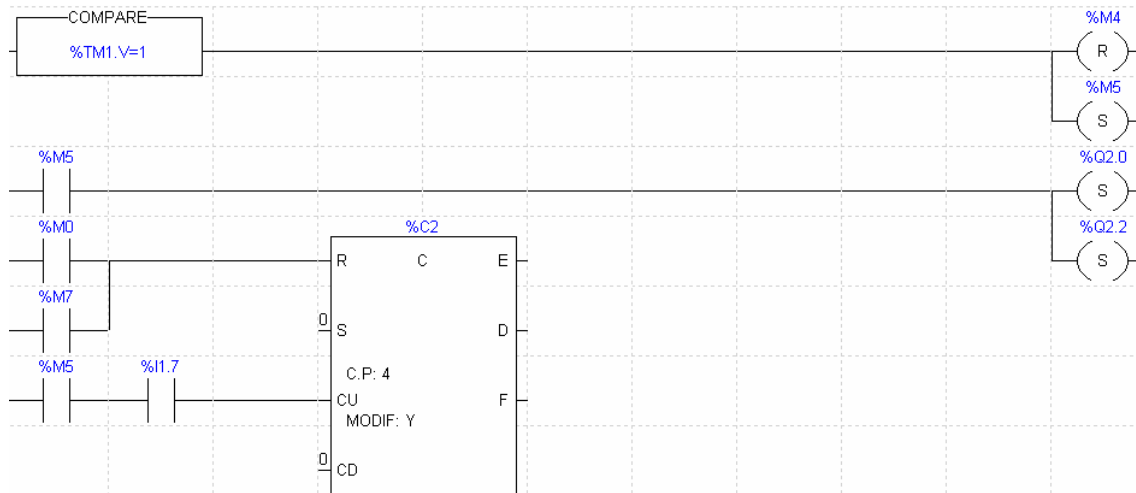


Figura 59. Anar a la ZONA 2

Quan s'hagi arribat a la ZONA 2, la posició de l'esclau serà x02 (on "x" serà l'esclau). Llavors caldrà esperar a que %MW16 valgui 1 per poder accedir a la TASCA 2.

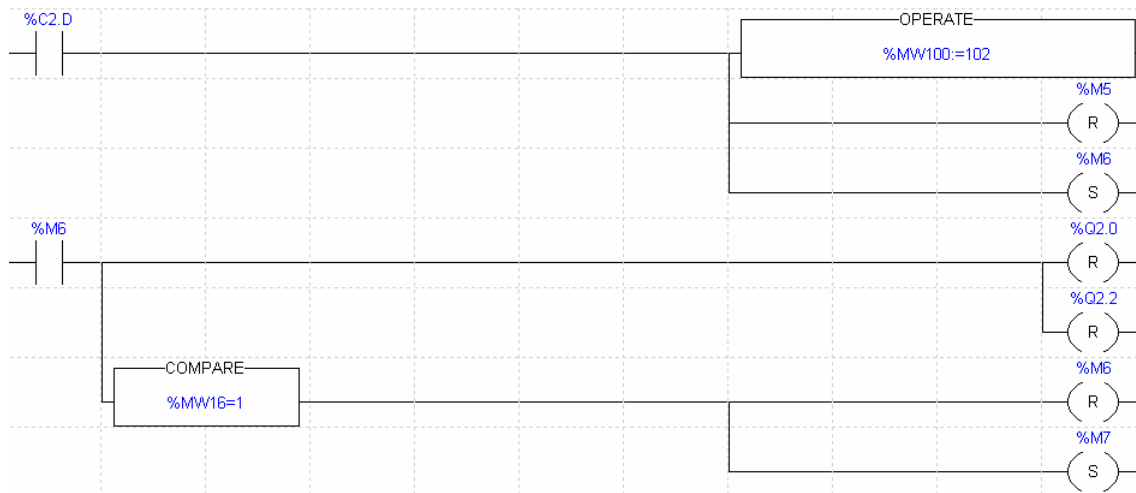


Figura 60. ZONA 2 i esperar per accedir a TASCA 2

Quan el mestre doni permís a l'esclau per realitzar la TASCA 2, el led 2 (%Q2.5) de la maqueta didàctica s'encendrà.

Mentre s'estigui realitzant la TASCA 2, l'esclau estarà a una posició intermitja que és la x13 (on "x" variarà segons l'esclau).

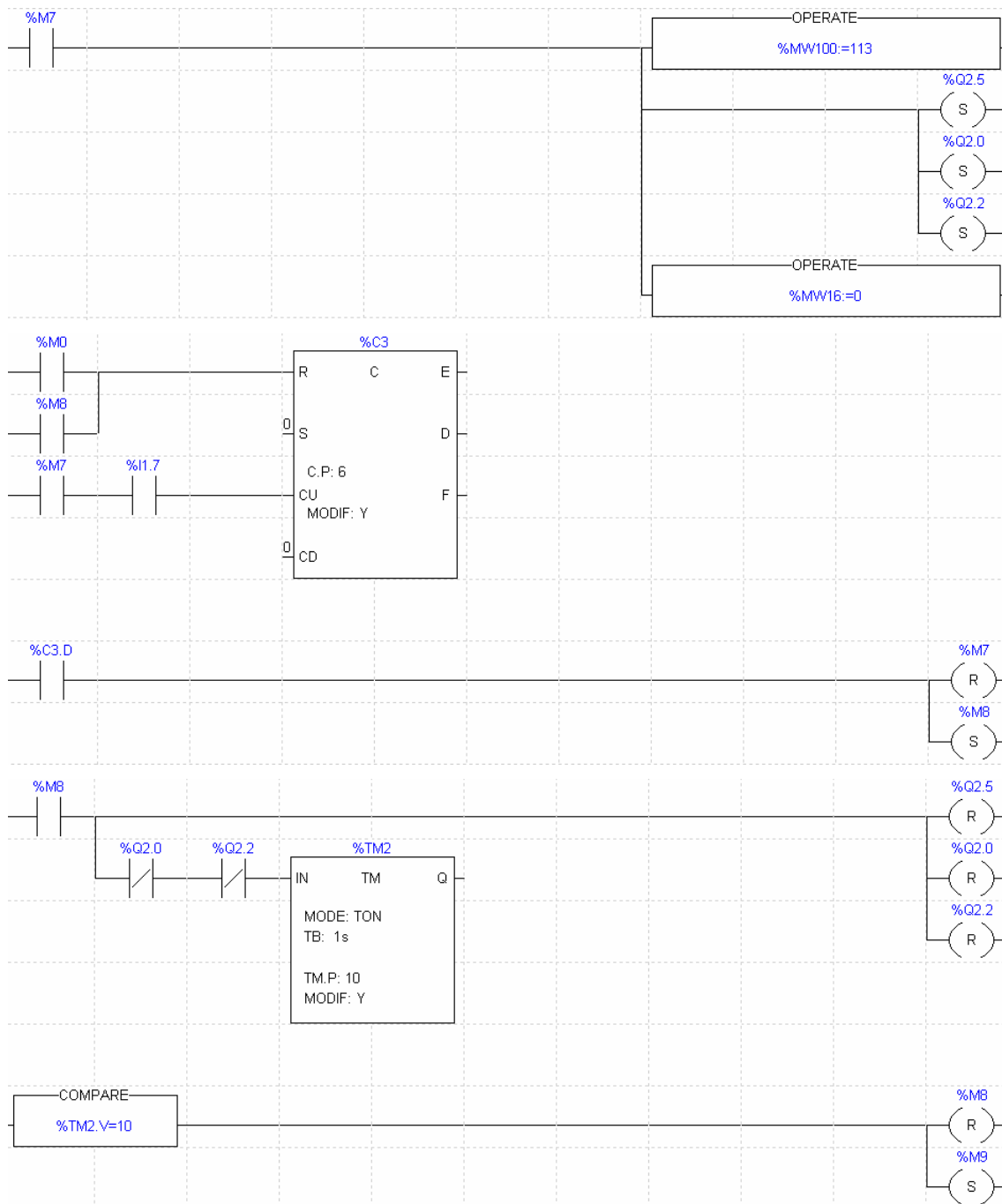


Figura 61. Realitzar TASCA 2

La TASCA 2 consistirà en comptar 6 peces metàl·liques i fer una parada de 10 segons. Un cop acabada la neteja dels contenidors dels camions (esclaus), s'hauran de dirigir cap a la ZONA 3 (zona de sortida de la terminal de buidat i neteja de contenidors). Quan el fi de cursa Fc2 detecta (%I1.6), llavors s'arriba a la posició x03.



## Creació d'una xarxa de comunicacions modbus pels autòmats telemecanique TsxMicro

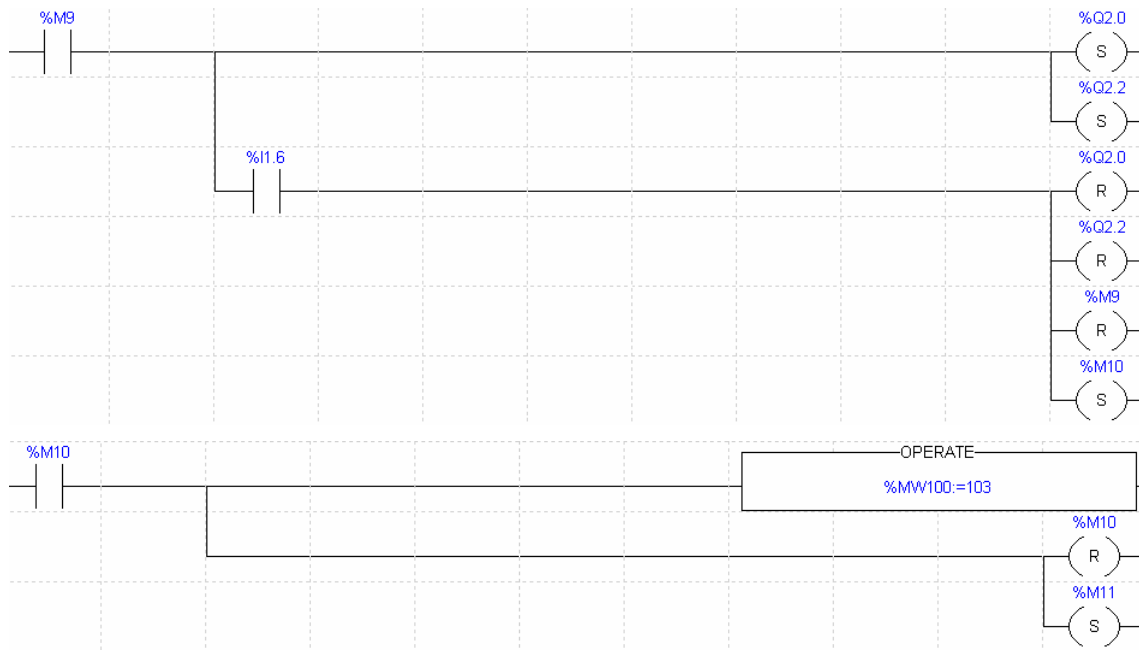


Figura 62. ZONA 3 de la terminal

Quan l'esclau arribi a l'última posició haurà de retornar de manera automàtica a la posició inicial (%I1.5).

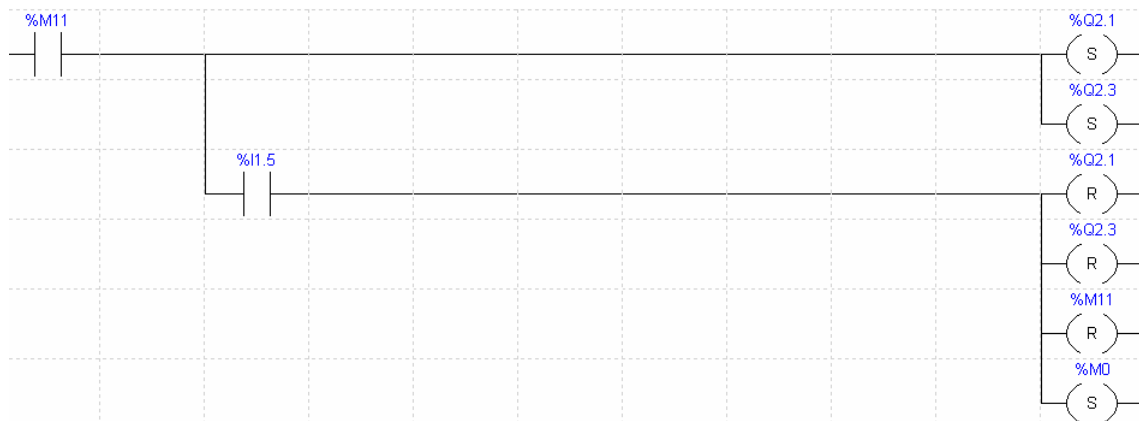


Figura 63. Retorn a la posició inicial.

## **7. RESUM DEL PRESSUPOST**

El pressupost inclou el material necessari per realitzar la part del muntatge de la xarxa de comunicacions Modbus i la part de programació, així com el cost de mà d'obra previst per realitzar el muntatge de la instal·lació.

El projecte tindrà un preu de dos mil dos-cents quaranta vuit euros i vuitanta - dos cèntims d'euro, sense iva.

## **8. CONCLUSIONS**

Un cop muntada la xarxa de comunicacions Modbus pels autòmats del laboratori de Regulació, es pot afirmar que els objectius del projecte s'han assolit.

La xarxa de comunicació instal·lada és un punt de partida per a futures ampliacions, com podria ser, per exemple, en el nombre d'autòmats que hi estan connectats (caldrà tenir en compte que s'hauran d'adquirir més unitats d'aïllament TSX PACC 01, una per cada autòmat), o bé, la visualització dels processos que es vulguin simular (per això cal adquirir una tarja PCMCIA ).

Com a complement s'ha realitzat una experiència simulant el procés de buidat i neteja de contenidors que serveix com a exemple als alumnes i que s'ha documentat convenientment per a l'ús didàctic. També s'han proposat unes pràctiques de poca dificultat a realitzar pels estudiants i un qüestionari de comprensió.

Marta Pérez Albiol  
Enginyera Tècnica Industrial

Girona, 20 de desembre 2005

## **9. RELACIÓ DE DOCUMENTS**

Els documents que formen part d'aquest projecte són la memòria, el plec de condicions, l'estat d'amidaments i el pressupost.

En la memòria es descriu la finalitat del projecte i el procés que s'ha seguit per dur-lo a terme, així com es fa una explicació teòrica de la comunicació modbus. La memòria té un annex que consisteix en un manual de pràctiques on hi ha una pràctica exemple convenientment explicada per a l'ús didàctic, i on es proposen experiències pels alumnes i un qüestionari. En el plec de condicions s'hi exposen les condicions tècniques i econòmiques necessàries per tal de poder dur a terme la realització del projecte. En l'estat d'amidaments s'hi defineixen les unitats de cada objecte o les unitats d'obra que configuren la totalitat dels productes (nombre d'unitats, característiques, models i tipus). En el pressupost s'hi determina el cost econòmic del projecte i es basa en l'estat d'amidaments. S'hi reflexen els preus unitaris, pressupost parcial i pressupost total. El pressupost té un annex on s'hi especifica el cost de la realització del projecte.

## **10. BIBLIOGRAFIA**

KELLEY, DEAN. Teoria de autómatas y lenguajes formales.

MAJOL I BADIA. Automatas programables.

ORGANIZACION AUTOMATAS. Informació sobre comunicació Modbus.

(<http://www.automatas.org>, 14 de juny de 2004)

PIEDRAFITA MORENO, R. Ingenieria de la automatizacion Industrial.

SCHNEIDER ELECTRIC. Descripció del software PL7, Modicon – Telemecanique.

SCHNEIDER ELECTRIC. Documentació tècnica del PL7 V4.3.

([http://155.210.154.15/Doc\\_Soft/IntrSp/menu.htm](http://155.210.154.15/Doc_Soft/IntrSp/menu.htm), 14 de juny de 2004)

SCHNEIDER ELECTRIC. Empresa dedicada a l'automatització.

(<http://www.schneiderelectric.es> , <http://www.schneiderautomation.com>, 14 de juny de 2004)

SCHNEIDER ELECTRIC. Manual electrotecnico telesquemario.

## **ANNEX A. MANUAL DE PRÀCTIQUES**

El manual de pràctiques permetrà a l'alumne iniciar-se en la comunicació modbus entre autòmats Telemecanique TSX Micro.

A l'alumne se li proporcionarà una pràctica convenientment explicada, unes pràctiques que haurà de realitzar el propi alumne i un qüestionari que permetrà comprovar al professor el nivell de comprensió de l'estudiant.

### **A1. INICIACIÓ A LA COMUNICACIÓ MODBUS**

#### **A1.1 Introducció**

Aquest manual té l'objectiu d'iniciar a l'alumne en la comunicació entre autòmats Telemecanique TSX Micro, en aquest cas concret, una comunicació Modbus.

Es pretén que l'alumne es familiaritzi amb aquest tipus de protocol. En ser un protocol estàndard (es pot donar en qualsevol tipus de PLC), pot ser de gran utilitat quan l'estudiant s'introdueixi en el món laboral.

Les experiències són introductòries i l'alumne haurà d'aplicar els coneixements previs que hagi adquirit amb anterioritat (en pràctiques anteriors), caldrà que l'alumne tingui uns mínims coneixements per poder-la realitzar.

Per realitzar les experiències, els alumnes faran servir els nou autòmats TSX 3705, l'autòmat TSX 3722, les unitats d'aïllament TSX P ACC 01 i la maqueta didàctica (com a eina de simulació). El software que s'usa per fer la programació de la simulació del procés és el PL7 pro versió V 4.4 per el TSX 3722, i versió V 3.4 pels TSX 3705 .

#### **A1.2 Bus Modbus**

El protocol Modbus proporciona un sistema estàndard de comunicació, independentment de la marca de l'autòmat.

La comunicació Modbus permet l'intercanvi de dades entre els equips connectats al bus. El protocol Modbus defineix una estructura de missatge que els controladors reconeixeran i usaran, independentment del tipus de xarxes sobre la que comuniquin. Descriu el procés que usa un controlador per demanar accés a un altre dispositiu, respon a les peticions des d'altres dispositius i es detecten i es notifiquen els errors que puguin haver-hi.

Durant la comunicació en Modbus, el protocol determina com cada controlador coneixerà la seva direcció de dispositiu, reconeixerà un missatge dirigit a ell,

determinarà el tipus d'acció a prendre i extraurà qualsevol dada o qualsevol altra informació continguda en el missatge.

El protocol Modbus crea una estructura jeràrquica consistent en un mestre (que és qui gestiona tots els intercanvis de dades) i un o varis esclaus.

És el mestre l'únic que pot iniciar les operacions. Els altres dispositius (els esclaus) responen subministrant al mestre la dada sol·licitada, o realitzant l'acció demanada pel mestre.

### **A1.2.1 Direccionament**

El mestre pot direccionar esclaus individualment o pot generar un missatge per difondre a tots els esclaus. Els esclaus tornen el missatge, és a dir, donen una resposta a les peticions que se'ls ha direccionat individualment, però no es tornen respostes a les peticions fetes en un missatge comú.

El protocol Modbus estableix el format per la petició del mestre, indicant en la petició la direcció del dispositiu esclau ('0' en cas de ser un missatge per a tots els esclaus), un codi de funció que defineix l'acció sol·licitada, qualsevol dada que s'hagi d'enviar i un camp de comprovació d'error.

El missatge de resposta de l'esclau també està definit pel protocol Modbus. Conté camps confirmant l'acció presa, qualsevol dada que s'hagi de tornar i un camp de comprovació d'error. Si el missatge rebut per l'esclau és defectuós o l'esclau és incapaç de realitzar l'acció sol·licitada, construirà un missatge d'error i l'enviarà com a resposta.

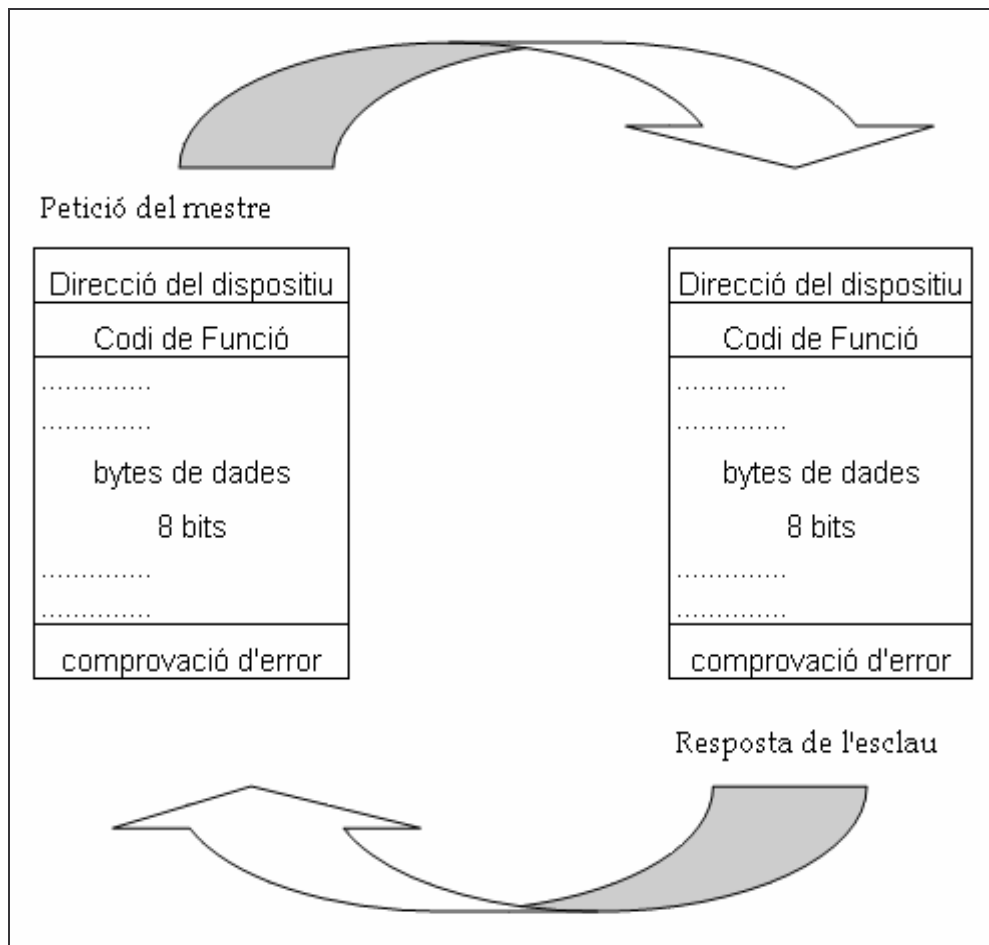


Figura 4. Format general d'una trama.

### A1.3 Autòmats TSX 3705 / TSX 3722

Els autòmats es poden comunicar a través del connector terminal, una tarja PCMCIA o un mòdul de comunicació.

Els autòmats de que es disposa al laboratori de regulació són TSX 3705 i TSX 3722. L'autòmat TSX 3722 és d'una gamma més alta que el TSX 3705, i conté dos espais per a targetes PCMCIA .

Pel que fa a les comunicacions, els controladors TSX 37 integren un enllaç multifunció mitjançant el connector terminal.

El controlador TSX 3705 disposa d'un connector terminal TER i el TSX 3722 té dos terminals diferents (tot i que funcionalment són idèntics) anomenats TER i AUX.

Aquest port de comunicacions es pot configurar com a Uni-telway o com a Modbus mestre o esclau, però tenint en compte que la configuració escollida sigui la mateixa pels dos connectors TER i AUX.



Els autòmats del laboratori de regulació tenen instal·lat un mòdul mixt d'entrades/sortides tot o res, TSX DMZ 28DR (16 E i 12 S).

La connexió d'un autòmat TSX37 ( mestre o esclau ) a un bus Modbus/Jbus necessita obligatòriament l'ús d'una caixa TSXPACC01 .

#### **A1.4 Unitat TSX PACC 01**

La caixa TSX PACC 01 és una unitat d'aïllament que es connecta al terminal TER dels controladors TSX 37 mitjançant un connector mini-Din en un dels seus extrems. Aquesta caixa permet connectar diferents equips amb el connector terminal dels controladors TSX 37- 05/08/10 (per això te dos connectors mini-Din TER i AUX, equivalents als dels controladors TSX 37-■■■).

La unitat TSX PACC 01 permet connectar un controlador TSX 37 al bus Modbus/Jbus, i també, fixar la modalitat de funcionament del connector terminal, és a dir, es pot escollir si treballem en Modbus esclau, Uni-telway esclau, etc. També adapta el bus quan la caixa està connectada a un dels extrems del bus Modbus.

L'aspecte extern de la caixa TSX PACC 01 es el que es veu a la Figura 10.

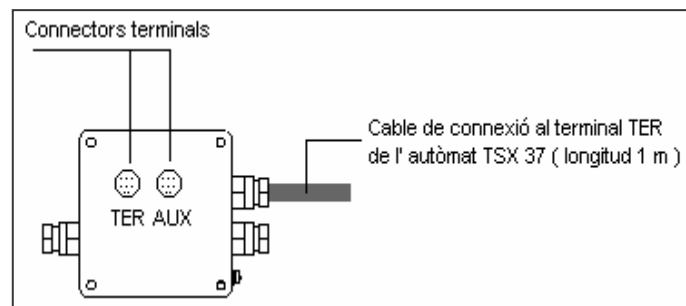


Figura 10. Unitat d'aïllament TSX PACC 01

Dins la caixa hi ha els interruptors que permetran escollir la modalitat de funcionament (S1 i S2). L'interruptor S1 permet escollir si es treballa en mestre o esclau i si es fa en Modbus o Uni-telway. L'interruptor S2 serveix per fer l'adaptació de la línia (final de línia o altres posicions), es col·loca en una posició o una altra segons si la caixa està al final de la xarxa de comunicació o no.

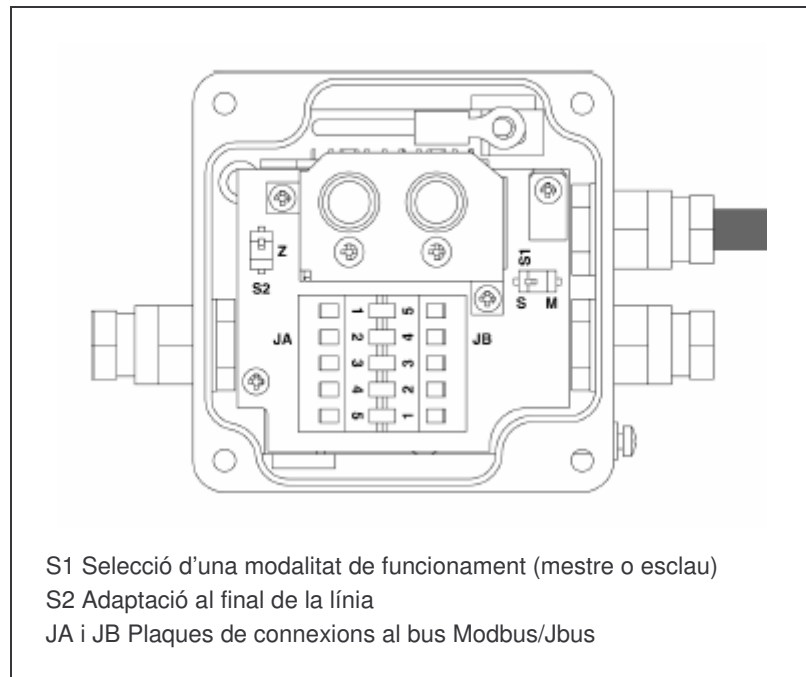


Figura 11. Vista interna de la caixa TSX PACC 01

	OFF	ON
S2	Caixa situada en posició diferent al final de línia	Caixa situada al final de línia
S1	Uni-telway mestre	Modbus, Uni-telway esclau o Mode caràcters

Taula 18. Posició dels interruptors S1 i S2

Per realitzar una comunicació en Modbus pel port TER de l'autòmat, sempre que no tinguem tarja PCMCIA, cal posar S1 en ESCLAU, ja que és en aquesta posició quan s'estableix el pont entre els pins 7 i 5 del port TER. Aquest pont és imprescindible per poder realitzar una comunicació Modbus pel port TER.

Si es disposa d'una PCMCIA, S1 pot posar-se en MASTER o SLAVE, és indiferent, ja que no és necessari el pont anteriorment esmentat.

Aquesta unitat d'aïllament duplica el connector terminal, així doncs, en un autòmat TSX 3705 que inicialment té només un connector terminal TER, amb la caixa TSX PACC 01 tindrà un connector TER i un connector AUX.

## **A1.5 Funcions bàsiques de comunicació**

Per tal de dur a terme la pràctica, cal conèixer les funcions bàsiques de comunicació.

Una funció de comunicació utilitza tres paràmetres concrets que són: un paràmetre de direcció, paràmetres específics d'una operació de comunicació i paràmetres de gestió.

En general, les funcions de comunicació tenen la forma següent:

Funció (Direcció del destinatari, Paràmetres específics, Paràmetres de gestió)

Funció : tipus de funció de comunicació.

Direcció del destinatari : direcció del destinatari que participarà en l'intercanvi.

Paràmetres específics : depenen del tipus de funció de comunicació i cada funció de comunicació té la seva descripció específica.

Paràmetres de gestió : són comuns a totes les funcions de comunicació asíncrones.

Els paràmetres de gestió es componen d'un paràmetre que dóna informació sobre l'activitat de la funció, d'un paràmetre que precisa el número d'intercanvi que identifica la tasca en curs, d'un paràmetre que conté la confirmació de l'intercanvi (confirmació de comunicació i confirmació de l'operació), d'un paràmetre de temps d'espera que permet controlar l'absència de respostes, i d'un paràmetre de longitud que permet memoritzar el número de bytes que s'emetràn o el número de bytes rebuts.

### **A1.5.1 Funcions de comunicació en Modbus**

De funcions de comunicació n'hi ha moltes i diferents, i segons el protocol que s'usa per la comunicació, es fan servir unes funcions de comunicació o unes altres.

En el cas concret del protocol Modbus, les funcions que suporta són: READ\_VAR, WRITE\_VAR, SEND\_REQ i SERVER.

Una comunicació consisteix bàsicament en un intercanvi de preguntes i respostes. Per tal de dur a terme aquesta comunicació, les funcions bàsiques que s'usen són la de llegir objectes estàndard (READ\_VAR) i escriure objectes estàndard (WRITE\_VAR). La funció SEND\_REQ és més complicada d'utilitzar que les funcions READ\_VAR o WRITE\_VAR. Serveix tant per escriure com per llegir peticions, però no es permet el seu ús en mode Modbus mestre en el connector terminal TER.

La funció SERVER només es pot usar per tractar les peticions d'un enllaç Modbus en cas d'usar una tarja PCMCIA TSX SCP 114 en un mòdul TSX SCY 21601 configurat en Modbus d'esclau amb servidor immediat. Aquesta funció permet tractar les peticions UNI-TE de forma immediata a partir del programa d'aplicació.

### A1.5.2 Funció de comunicació READ\_VAR

La funció READ\_VAR permet llegir el valor d'un o varis objectes de llenguatge com poden ser: bit intern, paraula interna, bit de sistema, paraula de sistema, paraula constant, paraula interna doble, paraula constant doble i objectes estructurats (temporitzador, monoestable, comptador, registres, programador cíclic).

Els objectes llegits sempre han de ser consecutius i es poden trobar en una via de comunicació de tipus Modbus.

La resposta ha de contenir un número màxim de bytes que estarà en funció del protocol i del tipus de destinatari. Al final d'una lectura, la longitud de les dades rebudes es memoritza a la paraula 4 del paràmetre de gestió (paràmetre gestionat per l'usuari).

Aquesta funció de comunicació pot llegir fins a 1000 bits consecutius, sigui quin sigui el protocol usat (Modbus o Uni-telway). Si es vol llegir més de 1000 bits s'ha d'usar la funció SEND\_REQ.

### A1.5.3 Sintaxi de la funció READ\_VAR

READ\_VAR(Direcció, Tipus d'objecte, N<sup>o</sup> del primer objecte, Número, Valor, Paràmetres de gestió)

Paràmetre	Descripció
Direcció	Direcció del destinatari de l'intercanvi. No es poden posar les direccions {xarxa.estació}APP, {xarxa.estació}APP.núm difusió (ALL).
Tipus d'objecte	Cadena de caràcters que especifica el tipus d'objecte que es llegirà %M : Bit intern %MW : Paraula interna
N <sup>o</sup> del primer objecte	Doble paraula que indica l'índex del primer objecte que es llegirà
Número	Paraula que especifica el número d'objectes que es llegiran
Valor	Taula de paraules que conté els valors dels objectes llegits. El número de bits llegits és un múltiple de 8.
Paràmetres de gestió	Paràmetres de gestió de l'intercanvi en curs. La confirmació de l'operació pren un dels valors següents: 16#00 = Lectura correcta 16#01 = Error d'operació 16#02 = Resposta incorrecta 16#03 = Mida de resposta incorrecta

Taula 26. Paràmetres de la funció RED\_VAR.

En el programa PL7, la introducció dels tipus d'objectes ha de ser coherent i s'ha de fer, o bé tot en lletres minúscules, o bé, tot en lletres majúscules. Si no es fa així, la funció reenvia 16#06 (paràmetres específics incorrectes).

Exemple d'ús:

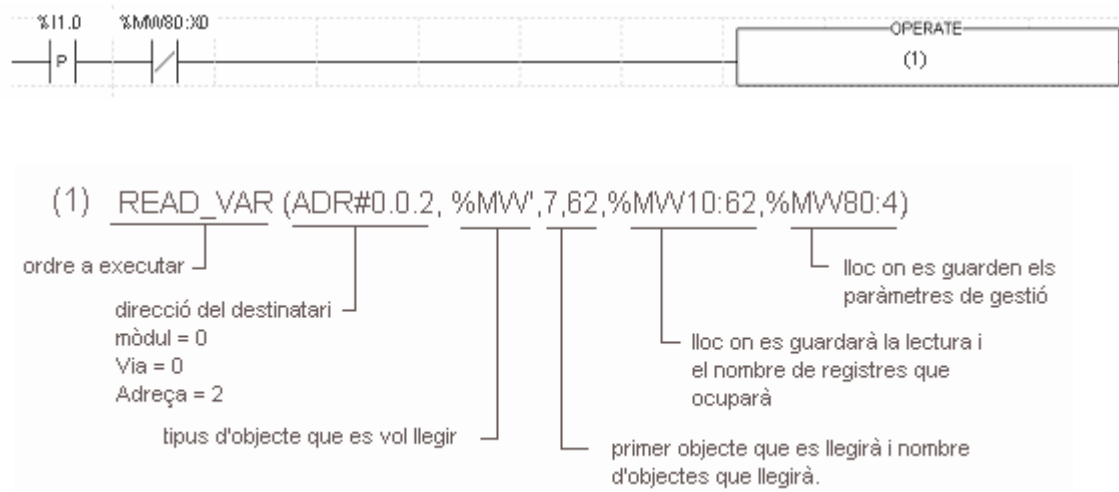


Figura 18. Funció READ\_VAR

Quan es donen les dues condicions (%I1.0 i %MW80:X0), s'activa la comunicació. %MW80:X0 és el bit d'activitat dels paràmetres de gestió (%MW80:4).

READ\_VAR : Ordre de lectura de N paraules (com a màxim 1000 bits)

ADR#0.0.2 : Direcció de l'entitat destinatària de l'intercanvi. El primer número és el mòdul, el segon número és la via de comunicació (1 si es tracta de la tarja PCMCIA i 0 si és el port TER), i el tercer és el número de l'esclau al que s'intenta accedir.

%MW : Cadena de caràcters que especifica el tipus d'objecte que es va a llegir (%MW: paraula interna o %M : bit intern)

7,62 : El primer número (en decimal) indica el registre pel que s'iniciarà la lectura. El segon (també decimal) indica el número de paraules a llegir. El Micro només permet la recepció de 1000 bits (62 paraules) després d'una petició de lectura (READ\_VAR). Per tant, el màxim de paraules que es poden llegir amb aquesta funció de lectura és de 62 paraules.

## Creació d'una xarxa de comunicacions modbus pels autòmats telemecanique TsxMicro

%MW10:62 : Aquí s'especifica la taula on es guarden les dades llegides. En aquest cas es tracta d'una taula de paraules (%MW). El primer número (decimal) indica en quin registre de la memòria del TSX Micro es guardarà la primera lectura realitzada. El segon (decimal també) indica la longitud de la taula de registres que s'utilitzarà per guardar la informació.

Cal assegurar-se de que la longitud de la taula concorda amb la quantitat de dades a llegir.

%MW80:4 : Indica on es guardaran en la memòria del TSX Micro els paràmetres de gestió de l'ordre. El primer número (decimal) indica en quin registre de la memòria del TSX Micro es guardarà el primer d'aquests paràmetres. El segon (decimal també) indica la longitud de la taula de registres que s'utilitzaran.

Les paraules d'aquesta taula poden adquirir els valors següents:

Primera paraula de la taula (%MW80):

Si el seu valor és 0: indica que l'acció s'ha realitzat (transmissió o recepció finalitzada)

Si el seu valor és 1: indica que no s'ha realitzat l'acció ( encara està transferint o rebent).

És a dir, el bit 0 d'aquesta paraula val 1 mentre s'està comunicant i 0 al acabar la comunicació.

Segona paraula de la taula (%MW81):

Si es realitza l'acció, contindrà els paràmetres de la petició:

En el byte 0, el codi de funció

En el byte 1, el codi de subfunció

Si no es realitza l'acció, contindrà el codi d'error:

16#0007 paràmetres de la petició incorrectes

16#0004 paràmetres de la qüestió incorrectes

Byte 0 = 0x80 + codi de funció

Byte 1 = codi error Modbus

(6 esclau ocupat, 1 funció desconeguda, 3 dades no vàlides)

Si el valor del byte 0 és FF (16#XXFF) indica que el missatge ha estat rebutjat, llavors, el valor del byte 1 indicarà el codi de l'error.

Si byte 0 = #FF	
Codi d'error (valor del byte1)	Significat
16#01	Falta de recurs cap al processador
16#02	Falta de recurs de línia
16#03	Equip absent sense recurs(*)
16#04	Error línia
16#05	Error longitud
16#06	Via de comunicació defectuosa
16#07	Error de direccionament
16#08	Error de l'aplicació
16#0B	Absència de recurs de sistema
16#0C	Funció de comunicació no activa
16#0D	Destinatari absent
16#0F	Problema d'encadenament entre estacions o via no configurada
16#11	Format de direccionament no controlat
16#12	Falta de recurs destinatari
16#14	Connexió no operativa (exemple: Ethernet TCP-IP)
16#15	Falta de recurs en via local
16#16	Accés no autoritzat (exemple: Ethernet TCP-IP)
16#17	Configuració xarxa incoherent (exemple: Ethernet TCP-IP)

(\*) Codi només gestionat per les targetes PCMCIA TSXFPP20

Taula 33. Codis d'error

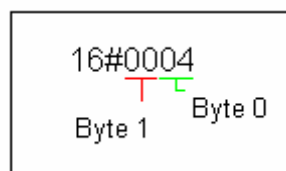


Figura 64. Descripció dels bytes 0 i 1

#### **A1.5.4 Funció de comunicació WRITE\_VAR**

La funció WRITE\_VAR permet escriure un o varis valors d'un o varis objectes de llenguatge del mateix tipus (bit intern, paraula interna, paraula constant, bit de sistema, paraula de sistema, doble paraula interna, doble paraula constant).

Els objectes que s'escriuen han de ser consecutius.

Poden provenir d'una unitat central remota o d'un equip connectat a una via de comunicació de tipus Modbus, Uni-telway, connector terminal, Fipway, Modbus Plus, Ethway.

#### **A1.5.5 Sintaxi de la funció WRITE\_VAR**

WRITE\_VAR (Direcció, Tipus d'objecte, N<sup>o</sup> del primer objecte, Número, Valor, Paràmetres de gestió)

<b>Paràmetre</b>	<b>Descripció</b>
Direcció	Direcció del destinatari de l'intercanvi. No es poden usar les direccions {xarxa,estació}APP, {xarxa,estació}APP.núm difusió (ALL).
Tipus d'objecte	Cadena de caràcters que especifica el tipus d'objecte a escriure: %M : bit intern %MW : paraula interna
N <sup>o</sup> del primer objecte	Doble paraula que indica l'índex del primer objecte que s'escriurà.
Número	Paraula que especifica el nombre d'objectes que s'escriuran. Només s'ha d'inicialitzar després de l'activació de la funció.
Valor	Taula de paraules que conté els valors dels objectes que s'emetraran.
Paràmetres de gestió	Paràmetres de gestió de l'intercanvi. La confirmació de l'operació pren un dels valors següents: 16#00 = Escripura correcta 16#01 = Error d'operació 16#02 = Resposta incorrecta

Taula 27. Paràmetres de la funció WRITE\_VAR



A continuació s'ofereix un exemple d'ús:

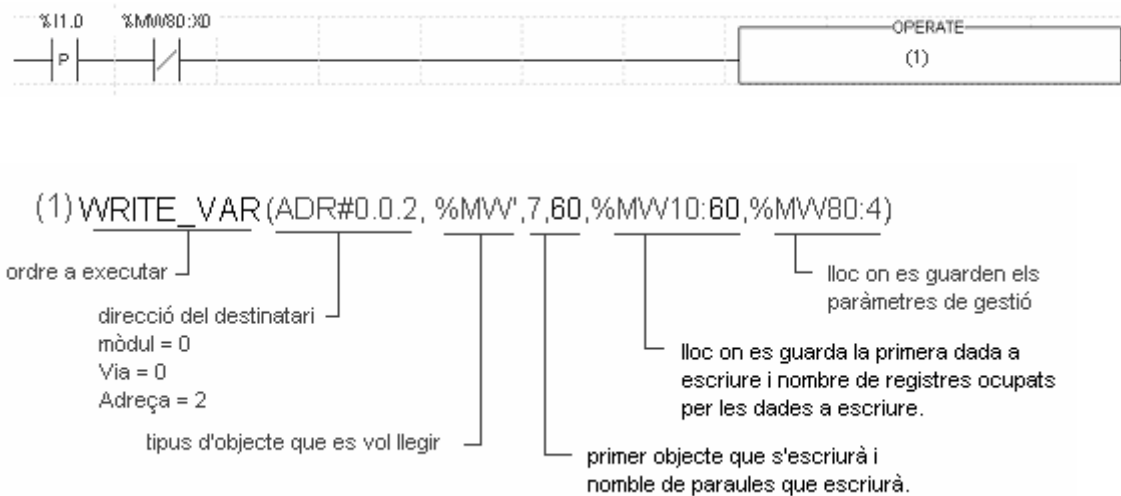


Figura 19. Funció WRITE\_VAR

WRITE\_VAR : Ordre d'escriptura de N paraules

ADR#0.0.2 : Direcció de l'entitat destinatària de l'intercanvi. El segon número és la via de comunicació (1 si es tracta de la tarja PCMCIA i 0 si és el port TER) i el tercer, és el número de l'esclau al que s'intenta accedir.

%MW : Cadena de caràcters que especifica el tipus d'objecte que es va a llegir (%MW: paraula interna o %M : bit intern).

7,60 : El primer número (en decimal) indica el registre del Micro (esclau) en el que s'iniciarà l'escriptura. El segon (també decimal) indica el número de paraules a escriure. El TSX Micro només permet la transmissió de 1000 bits (62 paraules) amb una única ordre de WRITE\_VAR. Tenint en compte que d'aquestes 62 paraules, 2 han de ser per rebre i guardar la taula dels paràmetres de gestió, només en queden 60. Aquest és el nombre màxim de paraules que es poden escriure amb una sola ordre de WRITE\_VAR.

%MW10:60 : Aquí s'especifica la taula on es guarden les dades a transmetre. En aquest cas es tracta d'una taula de paraules (%MW). El primer número (decimal) indica en quin registre de la memòria del TSX Micro està guardat el primer valor a

escriure. El segon (decimal també) indica la longitud de la taula de registres que s'ha utilitzat per guardar la informació.

Cal assegurar-se de que la longitud de la taula concorda amb la quantitat de dades a escriure.

%MW80:4 : Indica on es guarden en la memòria del TSX Micro els paràmetres de gestió de l'ordre. El primer número (decimal) indica en quin registre de la memòria del TSX Micro es guardarà el primer d'aquests paràmetres. El segon (decimal també) indica la longitud de la taula de registres que s'utilitzaran.

Les paraules d'aquesta taula poden adquirir els valors següents:

Primera paraula de la taula (%MW80):

Si el seu valor és 0: indica que l'acció s'ha realitzat (transmissió o recepció finalitzada)

Si el seu valor és 1: indica que no s'ha realitzat l'acció (encara està transferint o rebent).

És a dir, el bit 0 d'aquesta paraula val 1 mentre s'està comunicant i 0 al acabar la comunicació.

Segona paraula de la taula (%MW81):

Si es realitza l'acció, contindrà els paràmetres de la petició:

En el byte 0, el codi de funció

En el byte 1, el codi de subfunció

Si no es realitza l'acció, contindrà el codi d'error:

16#0007 paràmetres de la petició incorrectes

16#0004 paràmetres de la qüestió incorrectes

Byte 0 = 0x80 + codi de funció

Byte 1 = codi error Modbus

(6 esclau ocupat, 1 funció desconeguda, 3 dades no vàlides)

Si el valor del byte 0 és FF (16#XXFF) indica que el missatge ha estat rebutjat, llavors, el valor del byte 1 indicarà el codi de l'error. El significat dels diferents codis d'error és el mateix que per la funció READ\_VAR (taula 32).

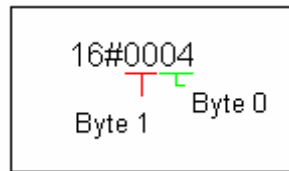


Figura 64. Descripció dels bytes 0 i 1

NOTA: Si al introduir la funció WRITE\_VAR, apareix el següent missatge d'error en la part inferior de la pantalla:

“Imposible referirse a esta instrucción: el número de versión de familia en la aplicación no corresponde al número de versión en”

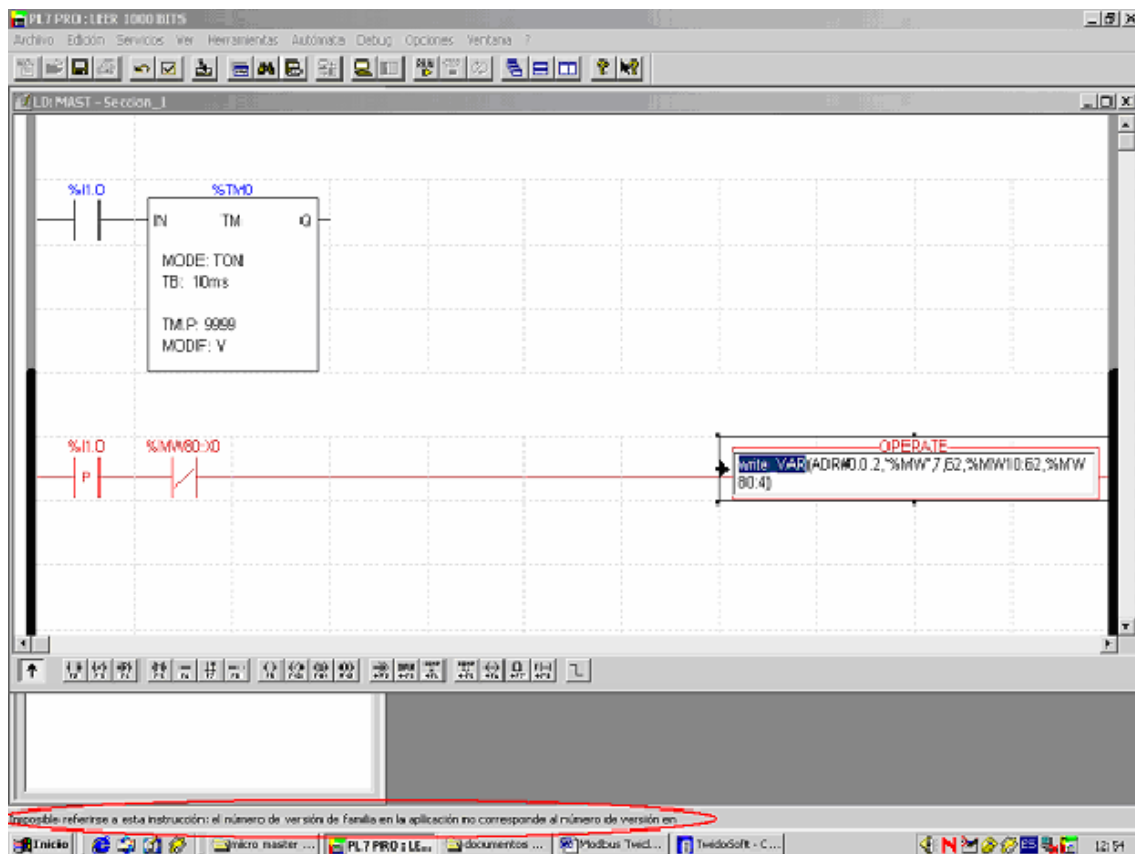


Figura 65. Missatge d'error WRITE\_VAR

Realitzar els següents passos per resoldre el problema:

Prémer ESC per tal de sortir de l'edició del bloc

Accedir al menú HERRAMIENTAS / BIBLIOTECA

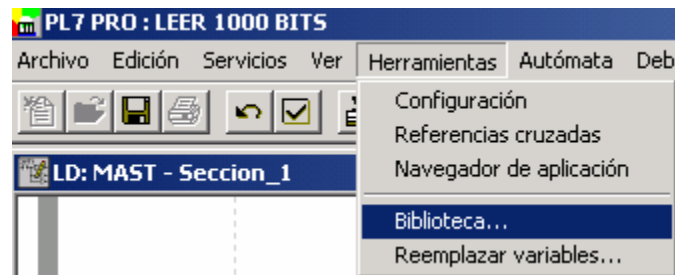


Figura 66. Menú software PL7

Apareixerà la següent finestra on es pot observar l'error (assenyalat en vermell). Es pot veure que la versió de l'aplicació és inferior a la de la biblioteca.

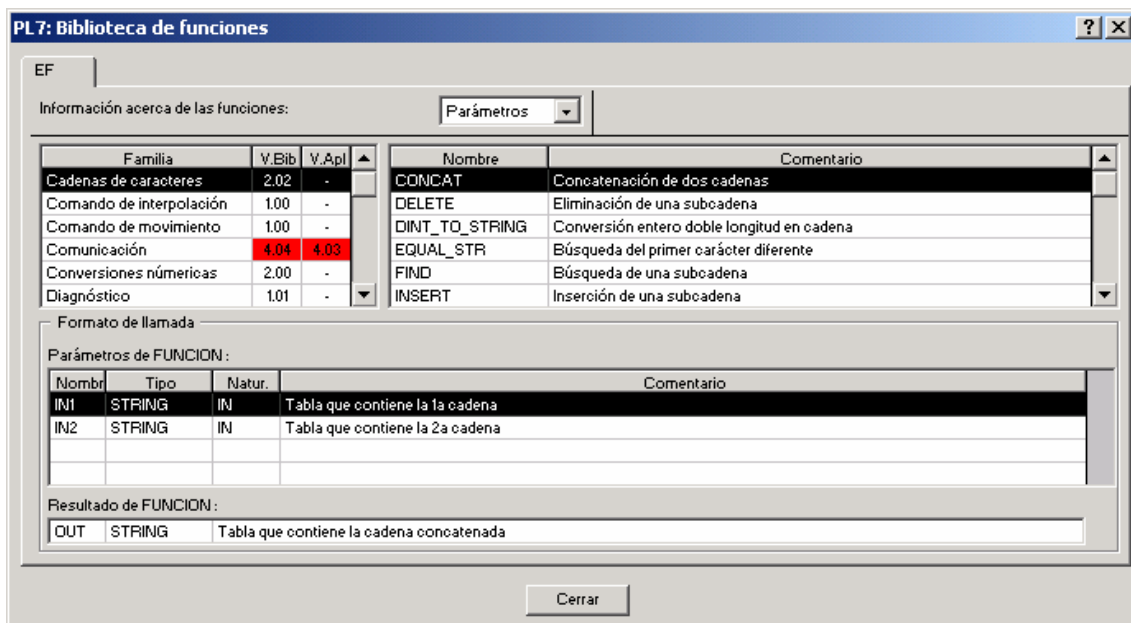


Figura 67. Pantalla biblioteca de funciones

El següent pas serà el d'actualitzar la biblioteca. Per fer-ho, caldrà pulsar sobre "COMUNICACIÓN" i en la finestra de comunicació, pulsar sobre "ACTUALIZAR FAMILIA".

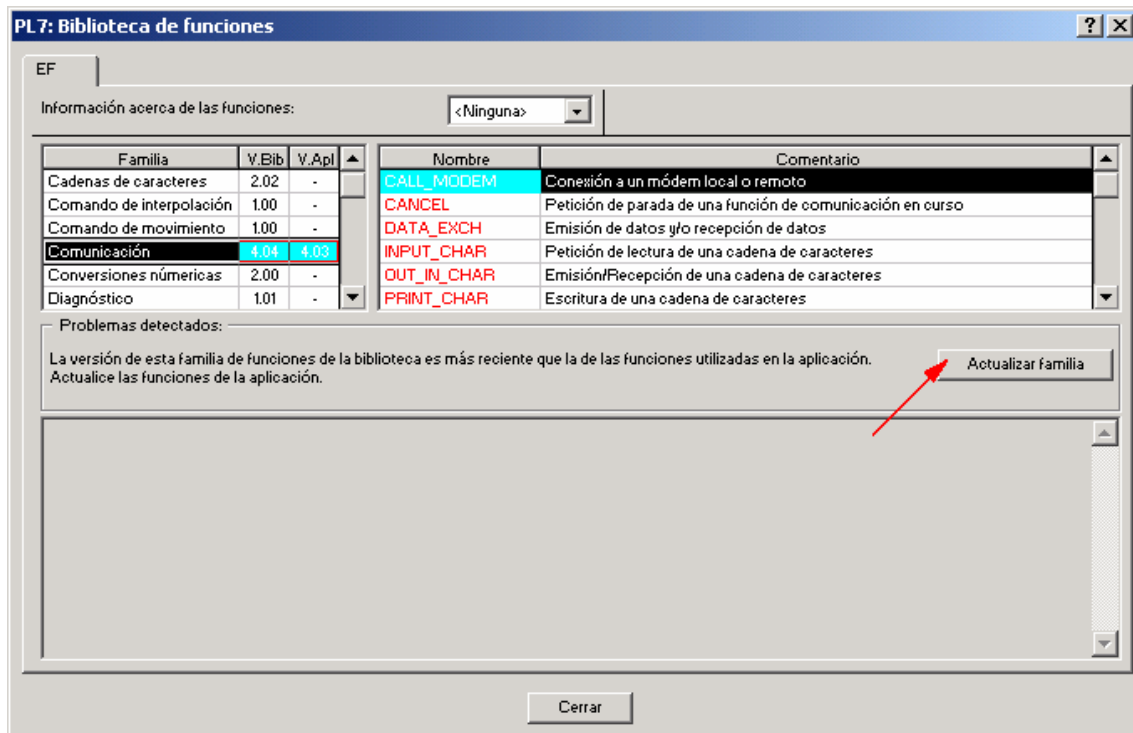


Figura 68. Pantalla actualitzar família.

Un cop s'ha acabat l'actualització, polsar sobre "CERRAR" i ja es podrà utilitzar la funció WRITE\_VAR.

### A1.6 Descripció de la maqueta didàctica

El disc es pot fer rodar amb un moviment màxim d'aproximadament 180°. Aquest disc està accionat per un motor de corrent continu que en funció de les ordres que li doni, l'autòmat podrà girar a dretes i a esquerres a velocitat lenta o a velocitat ràpida.

El moviment a dretes ha de quedar limitat pel fi de cursa Fc2 i el moviment a esquerres ha de quedar limitat pel fi de cursa Fc1.

El disc porta incorporat 20 peces metàl·liques numerades que seran detectades pel sensor de proximitat Fc3.

Tal com es reflexa en el llistat d'entrades i sortides es disposa d'altres entrades (polsadors, commutador, preselector de comptatge) i sortides (làmpades de senyalització, display BCD) que permeten ampliar les possibilitats de la maqueta.

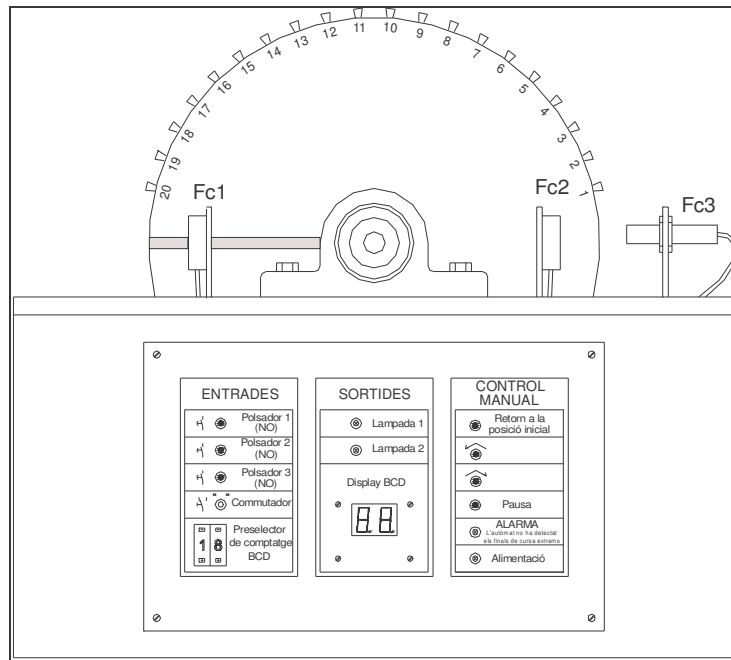


Figura 1. Dibuix de la maqueta didàctica

ENTRADA PLC	ETIQUETA	ELEMENT MAQUETA
0	Ps1	Pulsador 1
1	Ps2	Pulsador 2
2	Ps3	Pulsador 3
3	C_e	commutador posicionat a l'esquerra
4	C_d	commutador posicionat a la dreta
5	Fc1	fi de cursa indica posició esquerra disc
6	Fc2	fi de cursa indica posició dreta disc
7	Fc3	sensor inductiu de posició pel comptatge
8		preselector BCD de unitats, pes 1
9		preselector BCD de unitats, pes 2
10		preselector BCD de unitats, pes 4
11		preselector BCD de unitats, pes 8
12		preselector BCD de desenes, pes 1
13		preselector BCD de desenes, pes 2
14		preselector BCD de desenes, pes 4
15		preselector BCD de desenes, pes 8

Taula 2. Entrades de la maqueta

SORTIDA PLC	ETIQUETA	ELEMENT MAQUETA
0	Drt	moviment a dretes
1	Esq	moviment a esquerres
2	Lnt	velocitat lenta
3	Rpt	velocitat ràpida
4	Lp1	làmpada 1
5	Lp2	làmpada 2
6		display BCD de unitats, pes 1
7		display BCD de unitats, pes 2
8		display BCD de unitats, pes 4
9		display BCD de unitats, pes 8
10		display BCD de desenes, pes 1
11		display BCD de desenes, pes 2
12		display BCD de desenes, pes 4
13		display BCD de desenes, pes 8

Taula 3. Sortides de la maqueta

La maqueta i el PLC s'uneixen mitjançant un cable de connexió.

Els comandaments del moviment del disc utilitzant els pulsadors CONTROL MANUAL té preferència sobre les ordres que l'autòmat estigui donant pel control del moviment del disc. La maqueta disposa del pulsador de control manual anomenat "retorn a la posició inicial" que permetrà a l'alumne posicionar el disc a l'esquerra, ja que les experiències tindran aquesta posició com a inicial.

### **A1.7 Connexionat de les experiències**

En l'experiència exemple es connectaran cinc autòmats TSX 3705 amb les unitats d'aïllament TSX PACC 01 corresponents (una per cada autòmat), l'autòmat TSX 3722 (també tindrà una caixa TSX PACC 01), i els sis ordinadors corresponents.

En les experiències a realitzar per l'alumne, es connectaran tots els autòmats del laboratori (TSX 3705 i TSX 3722) amb les unitats d'aïllament TSX PACC 01 corresponents (una per cada autòmat) i els ordinadors corresponents del laboratori.

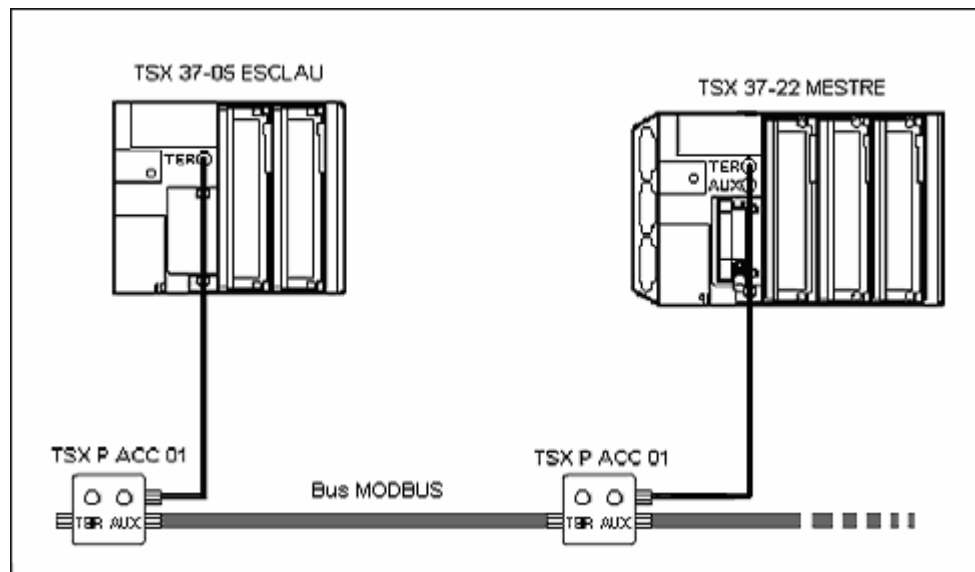


Figura 13. Connexionat de les experiències

Les caixes TSX PACC 01 es configuren com a finals de línia amb l'interruptor S2=ON, i TOTS els interruptors S1 com a ESCLAUS (ja actuïn com a mestre o com esclau). Això es fa perquè, per realitzar una comunicació en Modbus pel port TER de l'autòmat, quan S1 està en ESCLAU és quan s'estableix el pont entre els pins 7 i 5 del port TER (aquest pont és imprescindible per poder comunicar en Modbus). Si es disposa de tarja PCMCIA, l'interruptor S1 de l'autòmat mestre es pot configurar com a mestre o esclau, no influeix.

## A2. EXPERIÈNCIA 0. EXEMPLE

Exemple de comunicació Modbus. Simulació d'un procés de buidat i neteja de contenidors amb la maqueta didàctica.

### A2.1 Descripció de la pràctica

L'experiència consistirà en simular un procés de buidat i neteja de contenidors de camions que transporten diferents productes.



### TERMINAL DE BUIDAT I NETEJA DE CONTENIDORS

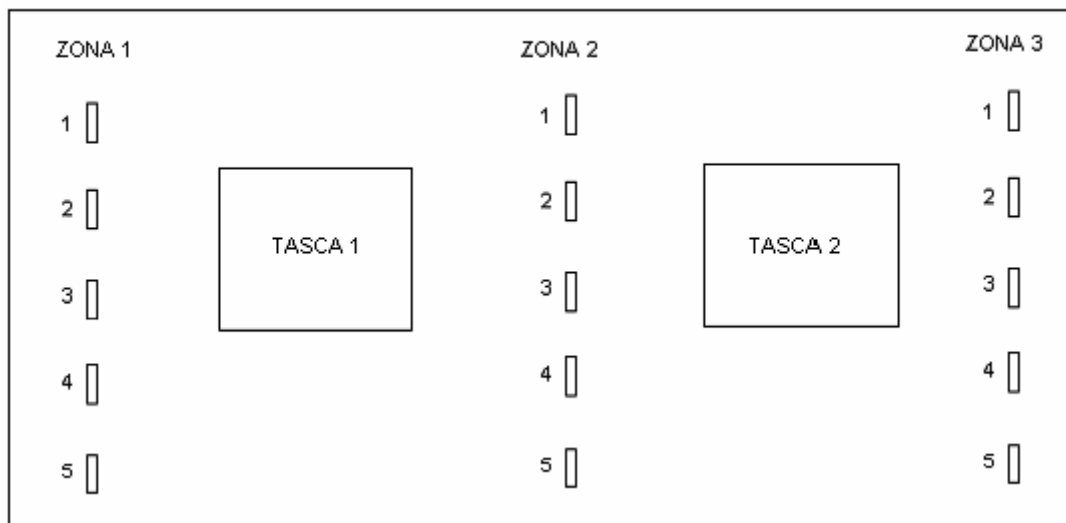


Figura 21. Terminal de buidat i neteja de contenidors

El procés comença quan un camió arriba a la terminal on es fa el buidat i la neteja dels contenidors. Es considera que el camió està a la primera zona (ZONA 1) quan el conductor del camió premi el polsador d'entrada (Ps1). Això servirà per saber quin camió ha arribat a la terminal. Seguidament, si no hi ha cap camió realitzant la TASCA 1, s'encendrà un led (que indicarà al camió que ja pot entrar a al terminal) i s'obrirà la barrera. En cas contrari s'haurà d'esperar.

Un cop el camió hagi acabat la TASCA 1, aquest es situa a la ZONA 2 (sensor d'entrada de la TASCA 2). Igual com en la ZONA 2, si hi ha algun camió fent la TASCA 2, caldrà esperar a que aquest acabi. La TASCA 2 consisteix en la neteja del contenidor del camió.

Quan el camió hagi fet les dues tasques (buidat i neteja), anirà cap a la sortida de la terminal (ZONA 3). Fins que el camió no arribi a la ZONA 3, no es donarà pas al següent camió.

#### A2.2 Emulació amb la maqueta

En aquesta secció es mostren les equivalències entre la maqueta i el procés.

Arribada de camió a la terminal i pas:

Ps1 (polsador arribada de camió a la terminal)

Lp1 (led zona 1, pas a la TASCA1)

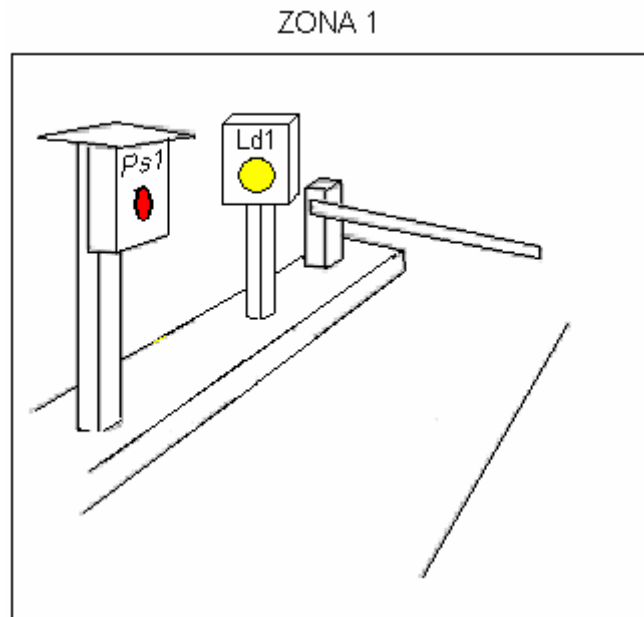


Figura 22. Zona 1 (entrada a la terminal)

Per representar l'acció d'obrir la barrera, Fc3 comptarà tres peces metàl·liques. La TASCA 1 consisteix en una parada de 2 segons i Fc3 = peça 7.

Un cop acabada la TASCA 1 es passa a la ZONA 2:

Sensor ZONA 2 (Fc3 = peça 10)

Lp2 (Led zona 2, pas a la TASCA 2)

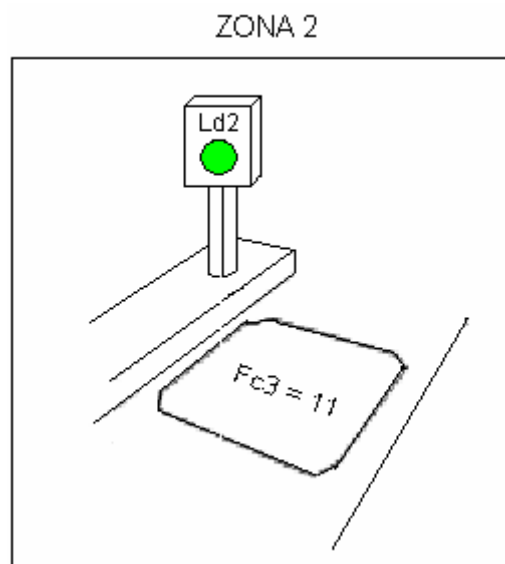


Figura 23. Zona 2 (entrada a la tasca 2 )

Quan el Fc3 = 10, si no hi ha cap camió a la TASCA 2, s'encendrà el led 2 i el camió ja podrà entrar a fer la TASCA 2. La TASCA 2 vindrà representada per una parada de 3 segons i Fc3 = peça 15.

Un cop acabada la TASCA 2, el camió passa directament a la sortida (ZONA 3) i, llavors, retorna directament a la posició inicial.

ZONA 3 (Fc2 i stop)

Fc1 (Retorn a la posició inicial)

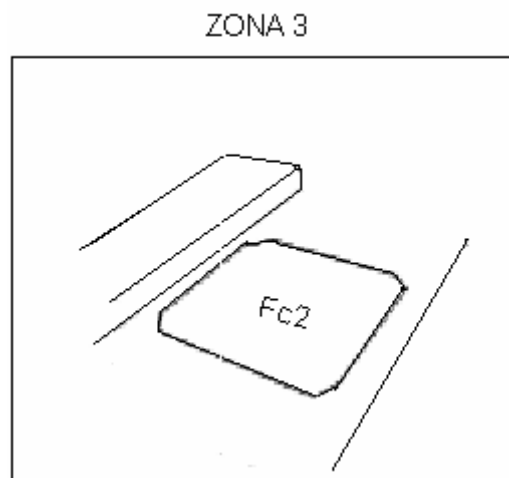


Figura 24. Zona 3 (sortida de la terminal)

### **A2.3 Realització de la pràctica**

La pràctica es fa amb el software PL7 pro V 3.4 en el cas dels TSX 3705 i PL7- pro V 4.4 en el cas del TSX 3722. Això és degut a que la comunicació Modbus mestre (configurat amb el connector TER) només es pot dur a terme amb una versió a partir de la V 4.2.

Abans de realitzar el programa mitjançant el software indicat, cal fer diverses comprovacions. La primera cosa que cal fer és comprovar que el cable TSX PCU 1030 està connectat al port sèrie (aquest cable és el que permet connectar el PC i el PLC, i passar el programa al PLC). Segon, col·locar els interruptors de les unitats d'aïllament TSX PACC 01 a la posició que correspongui (TOTES les caixes amb S1 a "S", esclau, i cal mirar si la caixa està col·locada en un cap de línia). Tercer, comprovar que l'autòmat mestre te connectada la botonera i els autòmats esclaus tenen connectada la maqueta.

El següent pas és el d'obrir el programa PL7 Pro i crear una aplicació nova (amb grafcet).

Entrar a la configuració del hardware (navegador de l'aplicació) i comprovar que hi ha instal·lat el mòdul TSX DMZ 28DR (mòdul tot o res de 16 entrades i 12 sortides, 24Vcc).

Llavors, es fa doble clic sobre el rectangle gris que diu "COM":

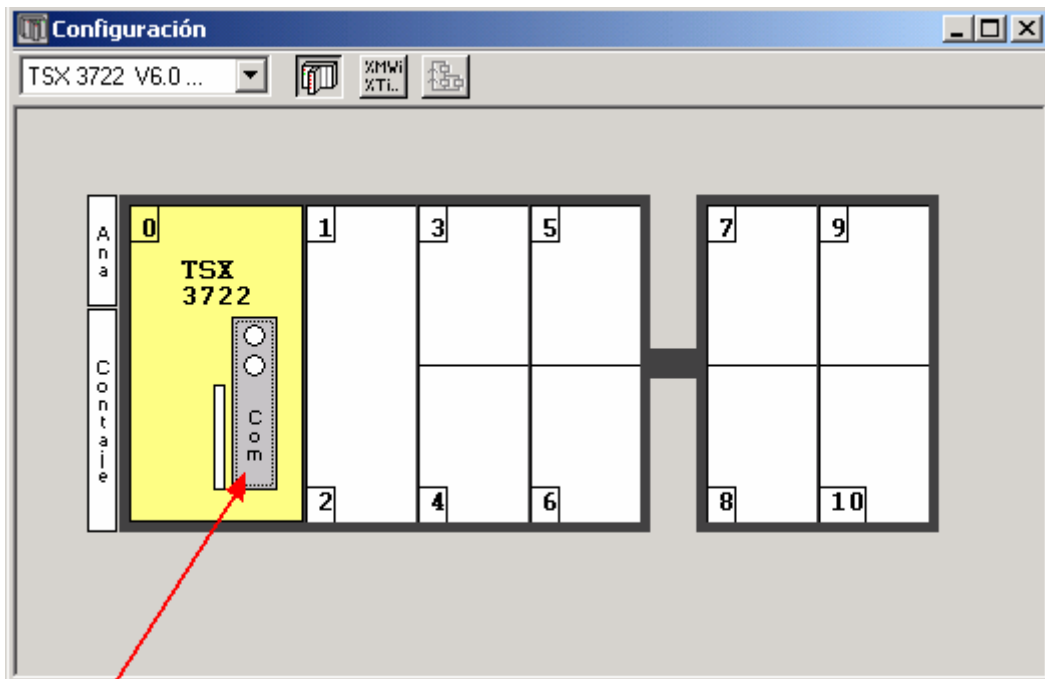


Figura 69. Pantalla configuració hardware.

Apareixerà la pantalla de configuració, i serà aquí on configurarem l'autòmat per tal de que es pugui realitzar una comunicació Modbus.

### **A2.3.1 Pantalla de configuració de Modbus**

En la següent pantalla cal configurar la via per la que es realitzarà la comunicació (via 1 si es te tarja PCMCIA i via 0 si no es te tarja PCMCIA - via que correspon al port TER-).

Si ja s'ha escollit la via, cal configurar l'enllaç per a dur a terme la comunicació. En el cas concret d'aquesta experiència, al ser una comunicació Modbus, es seleccionarà tal i com s'indica a la figura 70.

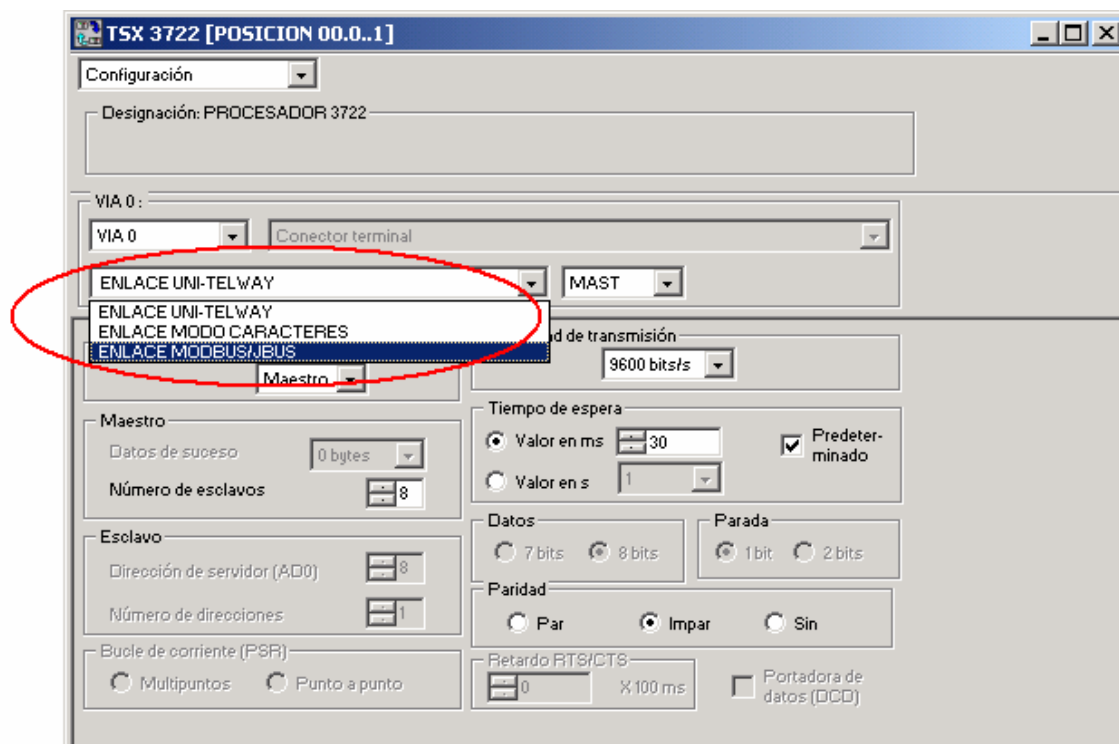


Figura 70. Pantalla de configuració

En la pantalla de configuració s'hi diferencien els paràmetres relatius a l'aplicació i els paràmetres relatius a la transmissió. Els primers són: Tipus, Funció mestre, número de reiteracions, Temps de resposta, Funció esclau i Número d'esclau. Els relatius a la transmissió són: Velocitat de transmissió Temps entre caràcters, Dades i Mode (RTU o ASCII).

Tipus : permet escollir el tipus del protocol Modbus usat pel mòdul (mestre o esclau).

Funció mestre: S'hi accedeix quan s'escull el tipus mestre. Permet configurar els paràmetres "número de reiteracions" i "temps de resposta".

Número de reiteracions: nombre d'intents de connexió que fa el mestre abans de declarar l'esclau com a absent. El valor predeterminat és 3, i els valors estan compresos entre 0 i 15. El valor 0 indica que no hi ha reiteracions per part del mestre.

Temps de resposta : és el temps de demora entre la petició emesa pel mestre i la reiteració de la mateixa en cas de no haver-hi resposta per part de l'esclau. Es correspon amb al temps màxim entre l'emissió de

l'últim caràcter de la petició emesa pel mestre i la recepció del primer caràcter de la petició enviada per l'esclau. El valor predeterminat és 1 segon ( $100 * 10 \text{ ms}$ ) i els valors estan compresos entre 10 ms i 10 segons.

Funció esclau : S'hi accedeix quan s'escull el tipus esclau. Permet configurar el paràmetre "número d'esclau".

Número d'esclau: anirà en funció del lloc de treball que s'ocupi (serà el professor qui assigni el número corresponent).

Velocitat de transmissió: la velocitat per defecte és de 9600 bits/s. Les velocitats disponibles són 1200, 2400, 9600 i 19200 bits/s. Les velocitats de 300 i 600 bits/s estan disponibles només amb la tarja PCMCIA TSX SCP 111.

Temps entre caràcters: és el temps de detecció de final de trama i del temps màxim que separa dos caràcters en recepció. Es gestiona quan l'autòmat està en recepció de missatges (indiferentment de si és esclau o mestre). Es recomana l'ús de valors predeterminats (4 ms ) amb configuracions sense mòdem i sense equip intermig. En cas contrari caldrà utilitzar valors superiors. El valor predeterminat depèn de la velocitat de transmissió.  
Per una velocitat de transmissió de 9600 bits/s tenim un temps entre caràcters de 26 ms i un nombre màxim de 23 caràcters.

Dades: permet obtenir informació del tipus de codificació usat per comunicar-se amb Modbus.

Mode RTU : Caràcters codificats en 8 bits.  
L'inici i final de trama es detecten amb un silenci d'almenys 3,5 caràcters.

Mode ASCII : Caràcters codificats en 7 bits.  
L'inici de la trama es detecta mitjançant la recepció dels caràcters ":" o per un silenci superior al temps entre caràcters.

El final de la trama es detecta mitjançant un silenci de durada superior al temps entre caràcters.

En el cas concret d'aquesta experiència, només es pot usar el mode RTU perquè no es disposa de tarja PCMCIA.

**IMPORTANT:** Cada vegada que es vulgui canviar algun paràmetre de la comunicació, caldrà que es repeteixin aquests mateixos passos (excepte el pas de seleccionar l'enllaç de comunicació que no caldrà que es modifiqui).

Per canviar qualsevol paràmetre cal desconnectar l'autòmat perquè així deixarà d'executar els programes immediatament.

Un cop configurat l'autòmat ja es pot passar a introduir el graficet de l'aplicació.

#### **A2.4 Programa autòmat mestre**

El mestre serà qui donarà pas als camions a les tasques corresponents (TASCA 1 o TASCA 2).

Per accedir a la TASCA 1, els camions arribaran a la ZONA 1 i, independentment de l'ordre real d'arribada, serà el mestre qui decidirà qui entra primer a fer el buidat del contenidor. El pas dels camions es donarà de forma manual.

En la TASCA 2 però, es crearà un ordre de prioritats (estructura FIFO). Aquest ordre serà el mateix que s'hagi donat en la ZONA 1, és a dir, el primer que accedeixi a la TASCA 1, serà el primer d'entrar a fer la TASCA 2, i fins que no hagi acabat, no podrà accedir-hi cap més camió. El pas del camió, el mestre el donarà de forma automàtica.

##### **A2.4.1 Llistat d'entrades i sortides**

El programa autòmat mestre es valdrà de la botonera per tal de donar pas als camions que arribin a la terminal i també per saber en quin estat es troba cada camió.

Els interruptors serviran per donar pas, de forma manual, als camions que hagin d'accedir a la TASCA 1. Els leds serviran per mostrar de forma visual, l'estat dels camions (si ha arribat algun camió a la ZONA 1, si hi ha camions a la ZONA 2 que estiguin en espera, si estan realitzant la TASCA 2, o bé, si han finalitzat tots els processos).

ENTRADA PLC	ETIQUETA	ELEMENT BOTONERA
1	PE1	Interruptor que dona pas a l'esclau 1
2	PE2	Interruptor que dona pas a l'esclau 2
3	PE3	Interruptor que dona pas a l'esclau 3
4	PE4	Interruptor que dona pas a l'esclau 4
5	PE5	Interruptor que dona pas a l'esclau 5
11	ENT	Interruptor per entrar esclau a taula de prioritats
15	INI	Interruptor que activa primer cop lectura esclau

Taula 28. Entrades al PLC ( botonera )

SORTIDA PLC	ETIQUETA	ELEMENT BOTONERA
1	L1A	Led esclau 1 esperant en zona 1
2	L2A	Led esclau 2 esperant en zona 1
3	L3A	Led esclau 3 esperant en zona 1
4	L4A	Led esclau 4 esperant en zona 1
5	L5A	Led esclau 5 esperant en zona 1
6	L1B	Led esclau 1 esperant en zona 2
7	L2B	Led esclau 2 esperant en zona 2
8	L3B	Led esclau 3 esperant en zona 2
9	L4B	Led esclau 4 esperant en zona 2
10	L5B	Led esclau 5 esperant en zona 2

Taula 29. Sortides del PLC (botonera)

#### A2.4.2 Descripció

El control del mestre es farà mitjançant la botonera i també de forma automàtica.

Els leds 12, 13, 14 i 15, no es poden fer servir ja que el mòdul d'entrades /sortides que es té instal·lat al laboratori és de 16 E i 12 S. Pel contrari, els interruptors si.



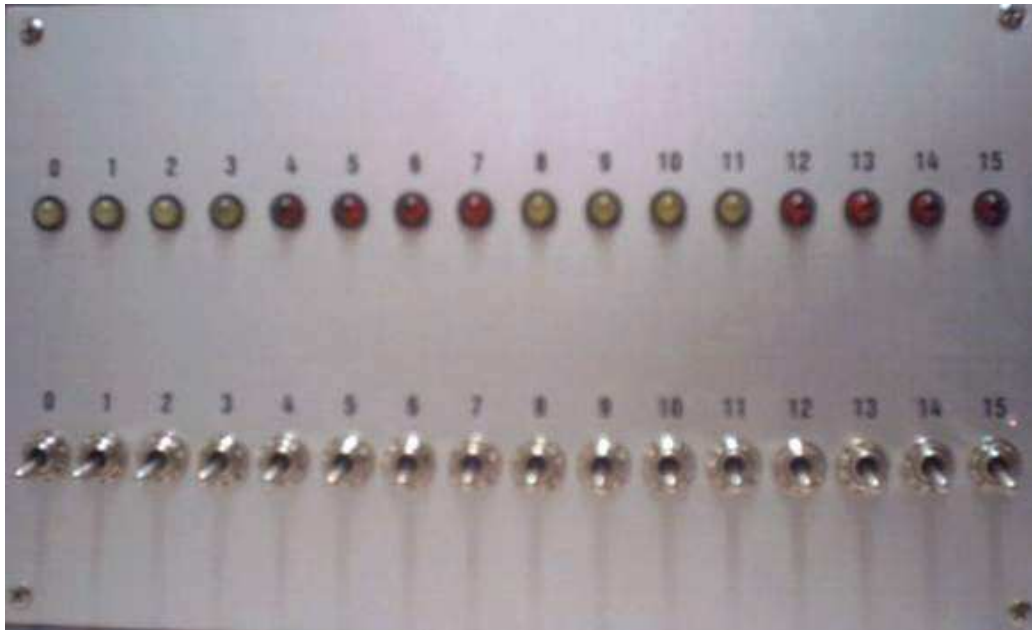


Figura 25. Botonera del laboratori de regulació

Quan un camió arriba a la ZONA 1 i prem el polsador 1, s'encén el led corresponent a l'esclau (L1A per l'esclau 1, L2A per l'esclau 2, etc). El mestre dóna pas a qui creu convenient mitjançant l'interruptor assignat a cada esclau (l'interruptor per l'esclau 1 és PE1, per l'esclau 2 PE2, etc).

Quan el supervisor veu que el llum de l'esclau a qui s'ha donat pas (sabrà qui està realitzant la TASCA 1 perquè l'interruptor queda amunt) s'ha apagat, voldrà dir que l'esclau ja ha acabat la TASCA 1. Llavors, ja podrà donar pas al següent esclau.

En arribar a la ZONA 2, s'encendrà el led corresponent (L1B per l'esclau 1 o per l'esclau 2, L2B per l'esclau 2 o per l'esclau 3, etc). En funció de l'ordre de pas que ha donat el mestre en la ZONA 1, es crearà la prioritat d'accés a la TASCA 2, mitjançant una estructura de memòria FIFO. Quan l'esclau estigui realitzant la TASCA 2, el led que li correspon a la ZONA 2, farà intermitent, i s'apagarà en el moment en què el camió acabi la neteja del contenidor i surti de la terminal.

Això permetrà saber en tot moment, de forma visual, quin és l'estat dels esclaus.

### **A2.4.3 Programa**

El programa s'ha editat mitjançant grafcet. S'ha fet ús de seccions editades en llenguatge de contactes (LD).

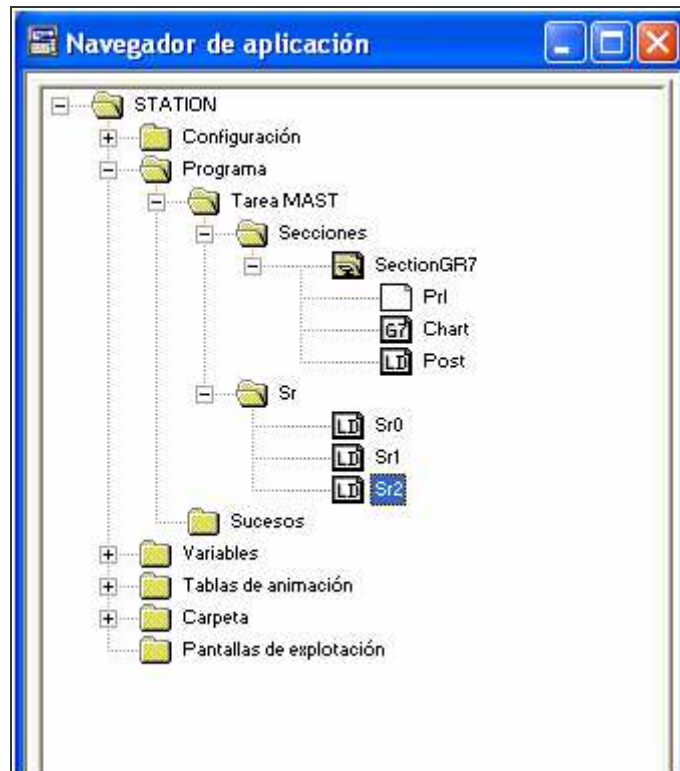


Figura 26. Estructura del programa mestre

El programa principal consta de la part que conté els graficets (Chart) i de la part que conté el programa en llenguatge de contactes (Post).

S'han creat tres subrutines que són: Sr0, Sr1 i Sr2. Les Sr0 i Sr2 serveixen per inicialitzar variables (bits i paraules). La Sr0 conté inicialitzacions que només es faran la primera vegada que s'executi el programa, i la Sr2 conté inicialitzacions que es faran sempre que el programa accedeixi a l'etapa inicial del graficet principal.

La Sr1 serveix per fer saber quin esclau té pas a la TASCAs 2.

#### A2.4.3.1 Secció Chart

Els graficets que han calgut per a l'autòmat mestre han estat tres. Un dels graficets és el principal, l'altre és el que serveix per omplir la taula de prioritats i un últim que serveix per llegir la posició inicial dels esclaus, és a dir, llegeix quan un camió ha arribat a la ZONA 1.

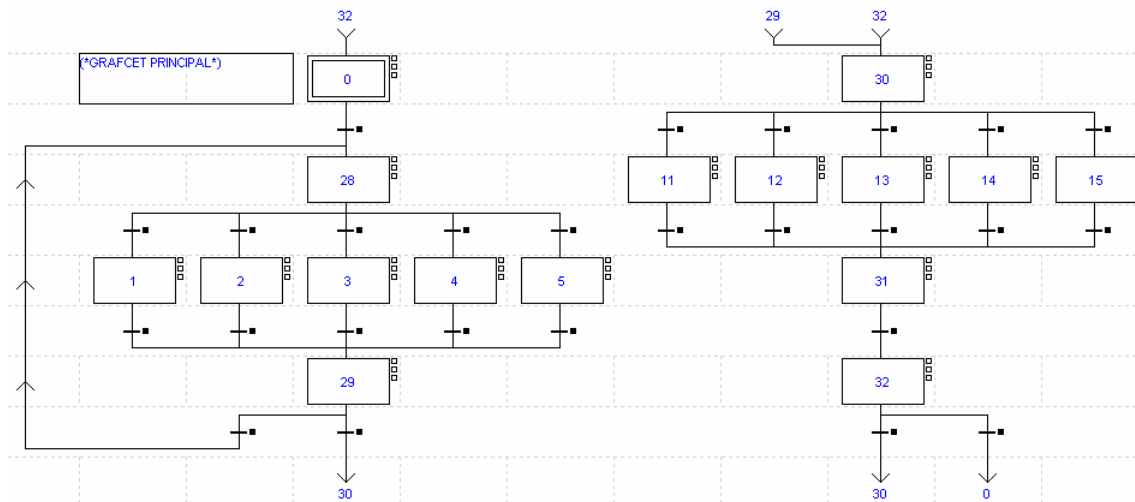


Figura 27. Grafjet principal autòmat mestre

L'etapa 0 és l'etapa d'inici. En les etapes 1, 2, 3, 4 i 5 el que es fa és que, si s'ha encès algun led (camió a la ZONA 1) i si es dona pas mitjançant l'interruptor corresponent, el mestre envia l'ordre de pas a l'esclau i espera fins que l'esclau arriba a la segona posició per donar pas a un altre esclau. Quan això succeeix, s'activa l'etapa 29, que enllaça directament amb l'etapa 30.

Per llegir la posició inicial dels esclaus, s'ha de fer de forma ordenada per tal d'evitar errors de lectura o escriptura. Més endavant s'explicarà de forma més ampla com s'ha fet aquesta forma de lectura.

L'etapa 30 comprova quin esclau té prioritats per entrar a la TASCAS 2 i li dona pas.

Les etapes 11, 12, 13, 14 i 15 serveixen per enviar l'ordre de pas als esclaus i llegir quan l'esclau ha acabat.

L'etapa 31 borra el valor que ocupava l'esclau dins la taula de prioritats.

L'etapa 32 serveix per a resetejar alguns bits i per incrementar el punter de lectura (%MW50) de la taula de prioritats.

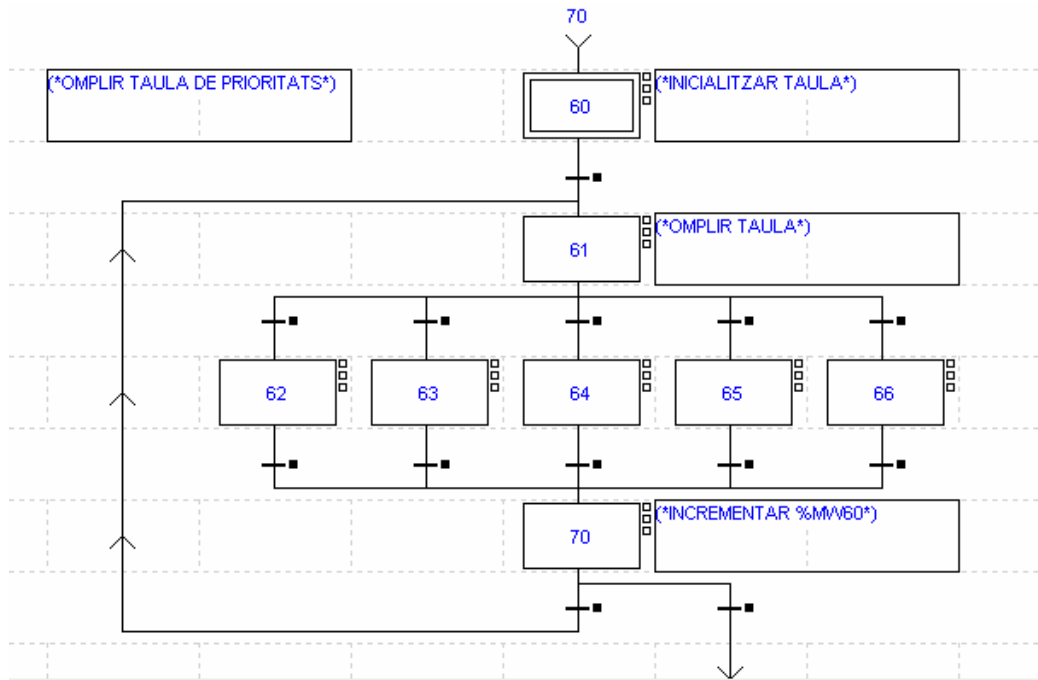


Figura 28. Grafcet omplir taula de prioritats

En l'etapa inicial (etapa 60) s'inicialitza el punter d'escriptura (%MW60) de la taula de prioritats.

De l'etapa 60 a la 61 s'hi accedeix directament.

Quan es dona pas als esclaus mitjançant els interruptors de la botonera, a la vegada, s'ha d'activar l'interruptor 11 per tal d'omplir la taula de prioritats. A l'etapa 61 s'omple la taula a mida que el mestre ha donat pas als esclaus (a la primera part del grafcet principal).

A l'etapa 70 s'incrementa el punter d'escriptura de la taula. Quan la taula estigui plena, es tornarà a l'etapa inicial d'aquest grafcet (etapa 60) on s'inicialitzarà el punter.

Mentre la taula no s'hagi completat, s'anirà augmentant una posició de la taula i s'hi guardarà el següent valor.

La taula de prioritats (escriptura) te la forma següent: %MW500[%MW60].

El punter d'escriptura %MW60 s'inicialitza a 1.

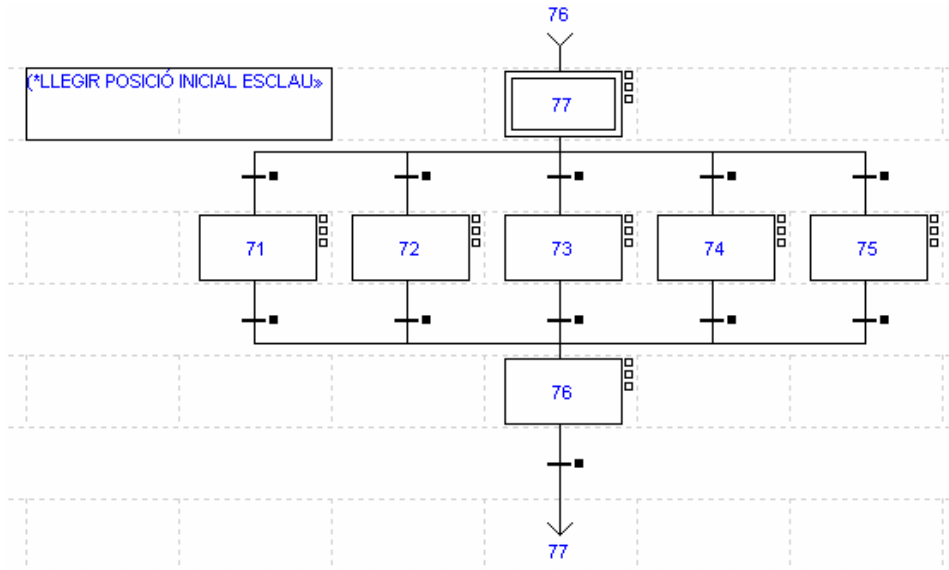


Figura 29. Grafcet llegir posició inicial dels esclaus

A l'etapa 77 s'hi resetejen els bits que serveixen per indicar que ja s'ha llegit la posició de l'esclau.

La lectura de la posició inicial dels esclaus (ZONA 1) s'ha de fer de forma ordenada per evitar conflictes en la comunicació. Primer es llegeix un esclau, llavors un altre, etc. Mentre l'esclau no hagi arribat a la ZONA 1, és a dir, mentre no s'hagi premut el polsador 1, el mestre anirà llegint aquest esclau. Quan el camió estigui a la posició final, és a dir, que surti de la terminal, llavors es tornarà a fer la lectura d'aquell esclau. Si l'esclau està realitzant qualsevol tasca, i mentrestant es prem el polsador 1 de nou, el mestre no reconeixerà com si hagués arribat un nou esclau, el mestre no ho llegirà i, en conseqüència, no s'encendrà el led corresponent a l'esclau. Això es fa per evitar conflictes amb l'esclau (si l'esclau fa alguna tasca o està esperant per fer-la, i per equivocació, es prem el polsador 1, no passarà res).

#### A2.4.3.2 Secció Post

El programa es podria dividir en tres parts que es correspondrien als grafcets. Igual que en els grafcets, els programes s'entrellacen i un necessita de l'altre per tal d'accedir a les etapes corresponents. S'explicarà el programa per ordre d'aparició a la pantalla i no per ordre d'ús, és a dir, si es mira el programa, les primeres instruccions que es veuen no són les que primer s'usen ja que, per exemple, la part de lectura de la primera posició de l'esclau està escrit al final del programa i és la primera cosa que es fa.

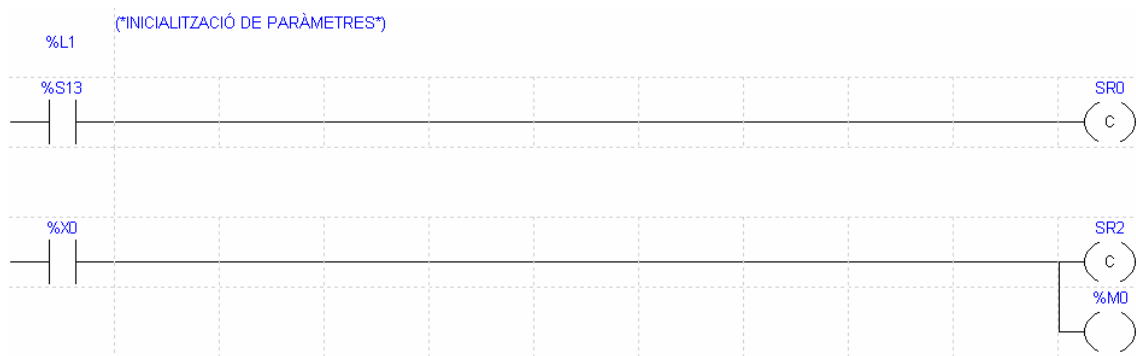


Figura 30. Inicialització de variables

El bit de sistema %S13 només s'activa quan el programa es posa a RUN, per tant, la Sr0 només s'activarà un cop, el primer cop.

Quan el bit associat a l'etapa 0 s'activi, també s'activaran la Sr2 i el bit %M0. El bit %M0 és un bit que serveix per accedir a l'etapa 28 del grafcet principal.



Figura 31. Esclaus a la posició 1

Les paraules %MW101, %MW102 .... %MW105 guarden la posició dels 5 esclaus. Les posicions seran les mateixes per a tots els esclaus (ZONA 1, ZONA 2, ZONA 3 o una zona intermitja que representa que l'esclau està realitzant la TASCAS 2), però el que les diferenciarà serà el primer número.

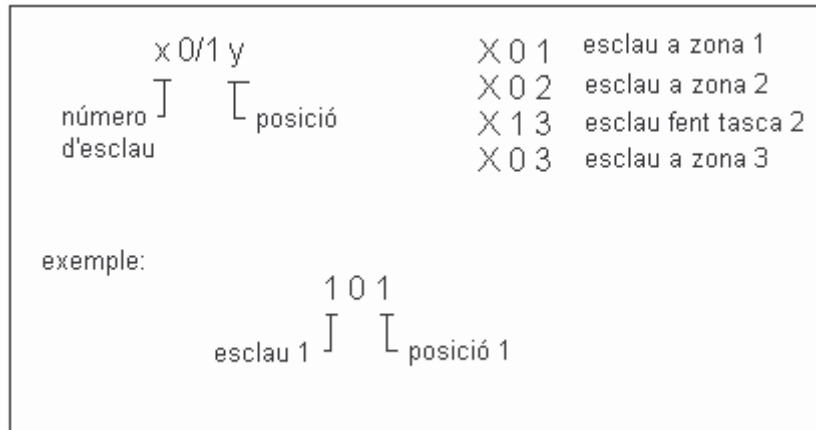


Figura 32. Grafia de les posicions i esclaus

Quan l'esclau estigui a la ZONA 1 de la terminal, s'encendrà el led corresponent i s'activarà el bit que permetrà que el mestre doni pas a l'esclau per realitzar la TASCA 1 quan aquest activi l'interruptor corresponent.

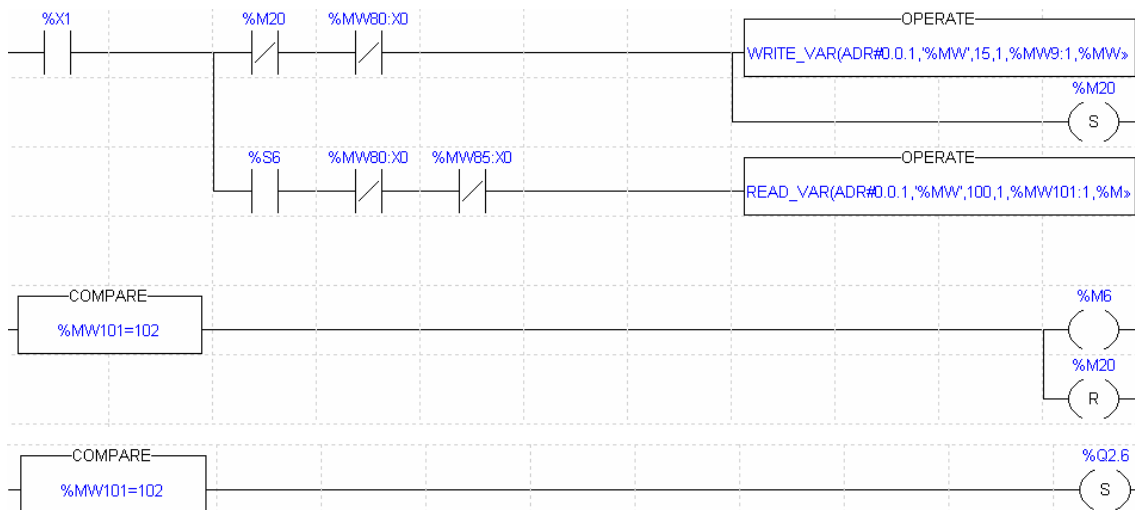


Figura 33. Etapa 1 (grafcet principal)

Quan s'activin %M1 i %I1.1, llavors s'activarà l'etapa 1 del grafcet. En aquesta etapa el mestre dóna pas a la TASCA 1 a l'esclau 1 enviant-li la paraula %MW15 (que valdrà 1). Aquesta instrucció (WRITE\_VAR) només l'envia una vegada, llavors espera a que l'esclau arribi a la ZONA 2 (READ\_VAR), la qual cosa voldrà dir que l'esclau 1 ja ha acabat la TASCA 1. Quan això succeeixi, s'activa el bit %M6 i es reseteja el bit %M20 per tal de que quan es torni a accedir a l'etapa 1, es pugui tornar a fer tota l'operació. Pels esclaus 2, 3, 4 i 5 es fa igual, l'única diferència és que, per exemple, per activar l'etapa 2 del grafcet, s'han d'activar %M2 i %I1.2, per l'etapa 3 %M3 i %I1.3, etc. Llavors, quan els esclaus arribin a la ZONA 2 s'activaran %M7 (per l'esclau 2), %M8

## Creació d'una xarxa de comunicacions modbus pels autòmats telemecanique TsxMicro

(per l'esclau 3), etc, i s'encendran els leds %Q2.6 (per l'esclau 1), %Q2.7 (per l'esclau 3), fins a %Q2.10 (per l'esclau 5).

El bit de sistema %S6 fa que cada segon es produeixi una lectura de la posició de l'esclau. El mestre llegeix la posició que l'esclau guarda a la paraula %MW100 i ho guarda a la paraula %MW10X (per l'esclau 1: %MW101, per l'esclau 2: %MW102, etc).



Figura 34. Etapa 29 (graficet principal)

Quan s'activi qualsevol dels bits %M6, %M7... %M10, s'activarà l'etapa 29 del graficet. Aquesta etapa es podria dir que és de transició. Quan s'activa l'etapa 29, s'activa el bit %M29 el qual permet activar l'etapa 30, i a la vegada, sempre retorna a l'etapa 28 (per tal de que es pugui enviar el pas a la TASCA 1 a un altre esclau).



Figura 35. Etapa 30 (graficet principal)

L'etapa 30 activa la Sr1. Més endavant s'explicarà completament el que fa aquesta subrutina, però per entendre l'etapa 30 cal saber que Sr1 activa els bits %M11 fins %M15 en funció de la zona que ocupen en la taula de prioritats. Aquests bits permetran accedir a les etapes 11, 12, 13, 14 i 15 respectivament.

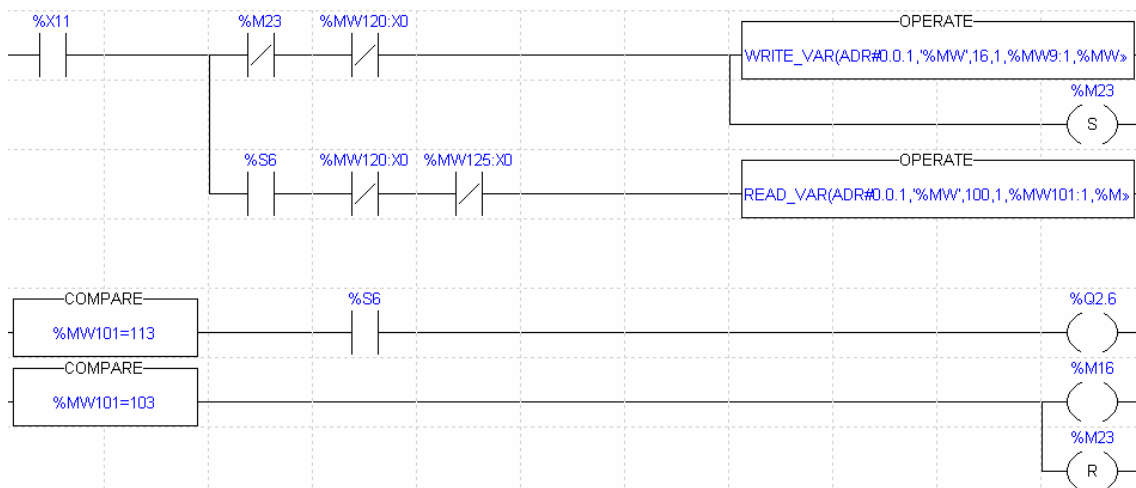


Figura 36. Etapa 11 (graficet principal)

Quan l'esclau 1 tingui prioritats per accedir a la TASCA 2, s'activarà el bit %M11 (que farà que s'activi l'etapa 11 del graficet).



En aquesta etapa el que es fa és donar pas a l'esclau a la TASCA 2 enviant a l'esclau la paraula %MW16 (que valdrà 1) amb la funció WRITE\_VAR. Igual que en l'etapa 1, aquesta instrucció només s'envia una vegada.

Quan l'esclau estigui realitzant la TASCA 2, la posició que es llegirà serà "x13". En aquest moment, el led corresponent a cada esclau per a la ZONA 2, es posarà a fer intermitent fins que arribi a la ZONA 3. Llavors el led s'apagarà i voldrà dir que el camió ja ha acabat (s'activarà el bit %M16 que permetrà activar l'etapa 31).

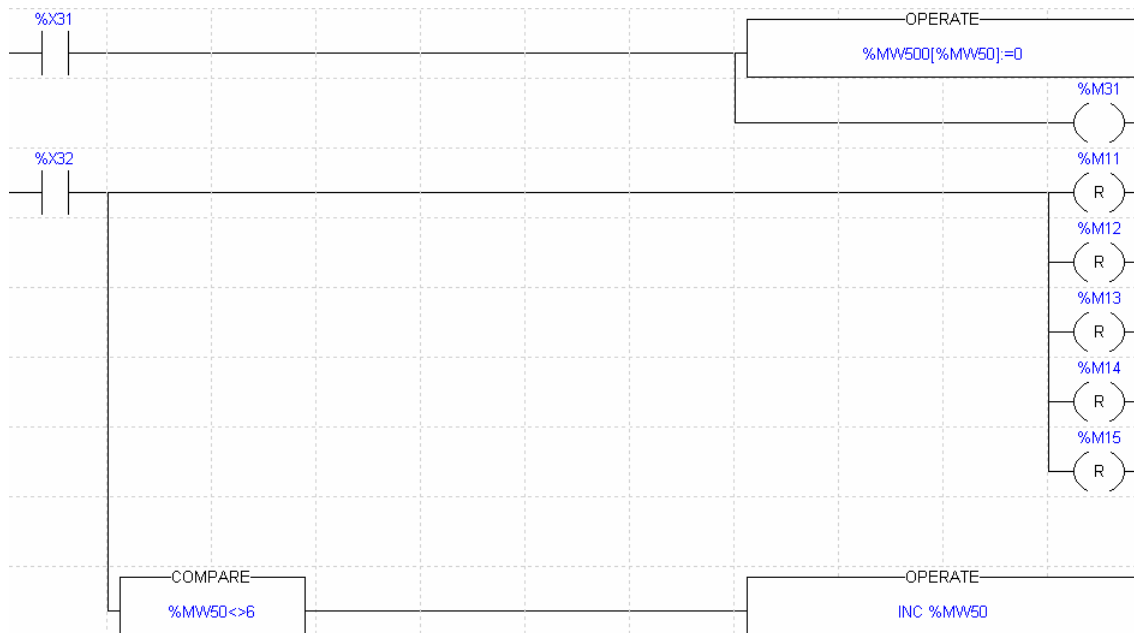


Figura 37. Etapes 31 i 32 (grafcet principal)

L'etapa 31 esborrarà el valor que ocupava l'esclau en la taula de prioritats. El bit %M31 permet activar l'etapa 32 del grafcet.

L'etapa 32 reseteja els bits que permeten accedir a les etapes 11, 12, 13, 14 i 15, i a la vegada, si el punter de lectura de la taula de prioritats (%MW50) és diferent de 6, és a dir, si no s'ha llegit tota la taula, incrementa el punter per tal de llegir la següent posició de la taula.

Si %MW50 és diferent a 6, el grafcet retorna a l'etapa 30, i si és igual a 0, passarà a l'etapa inicial (etapa 0).

Ara s'explica el programa referent a l'emplenat de la taula de prioritats. Aquest programa forma part del programa principal, no és una subrutina.

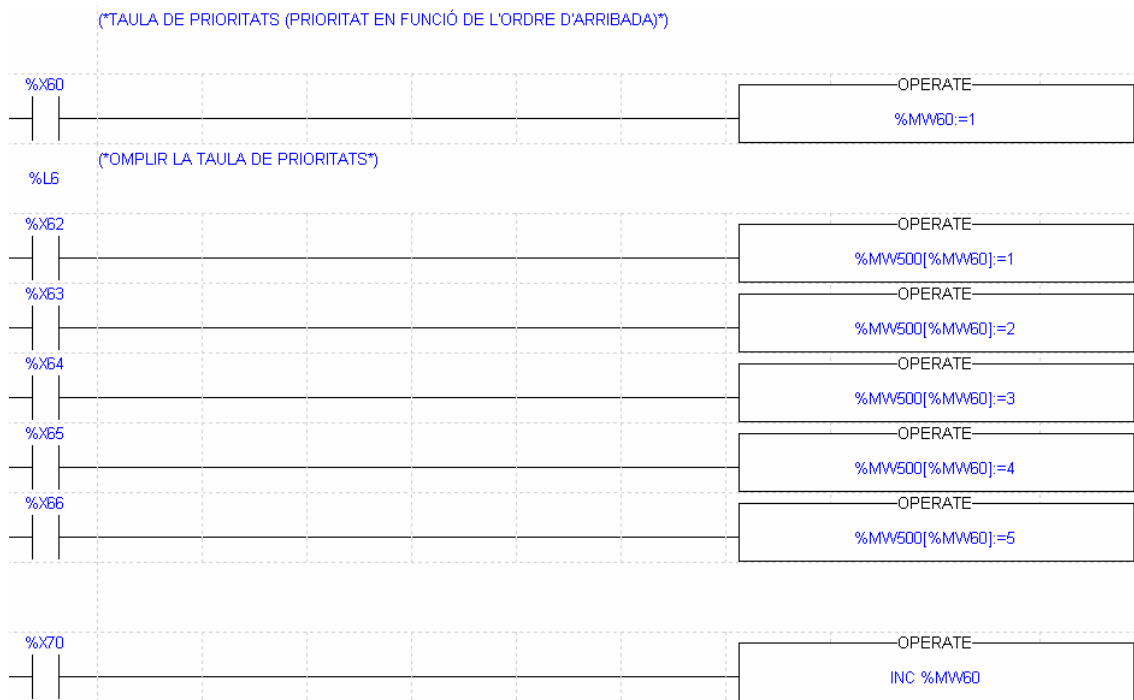


Figura 38. Omplir taula prioritats

L'etapa 60 serveix per inicialitzar el punter d'escriptura de valors a la taula (%MW60). La taula de prioritats consisteix en una taula de paraules de longitud 5 (%MW500:5). Per omplir-la, el que s'ha de fer és, quan el mestre dona pas a un esclau a la TASCA 1 mitjançant l'interruptor corresponent, a la vegada cal que també s'activi l'interruptor %I1.11. Quan s'activen els interruptors %I1.1 i %I1.11, s'activarà l'etapa 62, si s'activen els %I1.2 i %I1.11, s'activarà l'etapa 63 i així, successivament. El valor que es guarda a la taula és 1, 2, 3, 4 o 5 segons s'hagi donat pas a un esclau o a un altre.

Per exemple: suposem que arriben 3 camions a la ZONA 1 i s'encenen els tres leds corresponents a aquests esclaus. El supervisor donarà pas a l'esclau que li sembli, sense tenir en compte quin led s'ha encès primer. L'ordre que es doni per accedir a la TASCA 1 serà el que es mantindrà per accedir a la TASCA 2, amb l'única diferència que per accedir a la TASCA 2, el pas es donarà de forma automàtica.

En l'etapa 70 s'incrementa el punter d'escriptura %MW60. Si aquest punter és diferent de 6 retorna a l'etapa 61, i si és igual a 6, o sigui, si la taula està plena, torna a l'etapa 60.

La part del programa que llegeix la posició inicial dels esclaus és una mica complex d'entendre. Per facilitar la comprensió, primer s'explicarà la lectura pròpiament i llavors s'explicarà l'activació dels bits que ho permeten.

La lectura de la primera posició s'haurà de fer de forma ordenada. Primer es llegirà un esclau i llavors un altre. Això es fa així per evitar conflictes entre els esclaus i assegurar que tots els esclaus rebin les ordres que els corresponen.

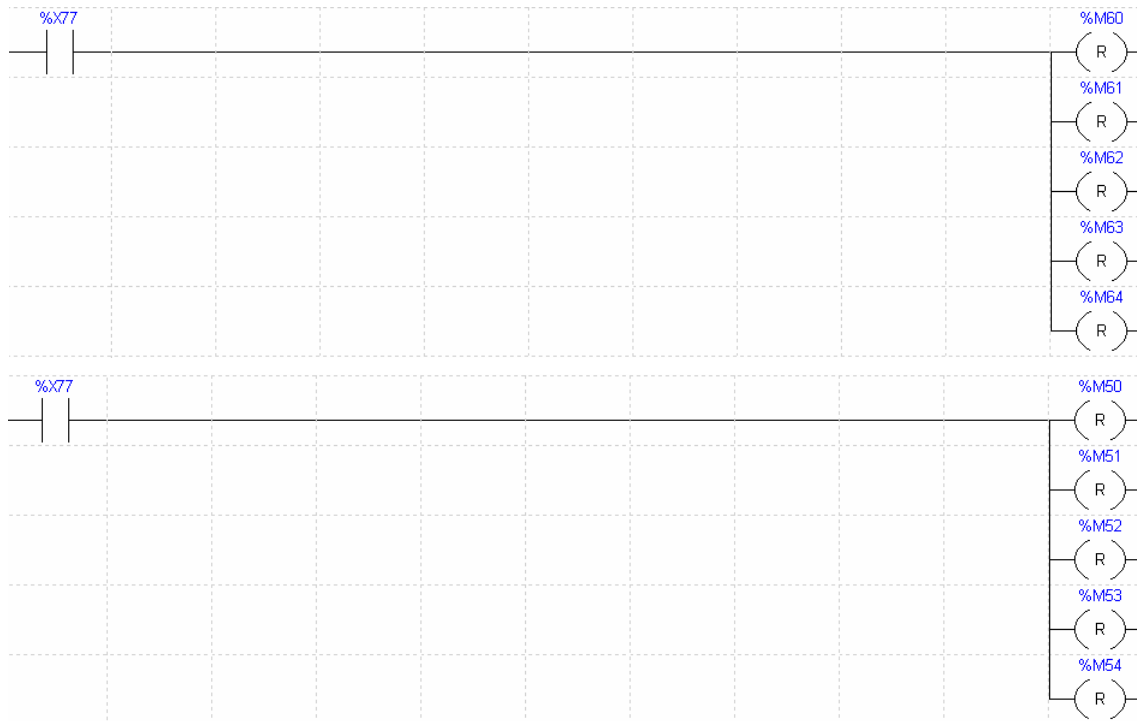


Figura 39. Etapa inicial (grafcet llegir primera posició esclaus)

L'etapa 77 és l'etapa inicial del grafcet on es llegeix la primera posició dels esclaus, és el grafcet que avisa al mestre de que ha arribat un camió a la terminal.

Aquí es resetejen els bits que permeten la lectura (%M60 fins %M64) i els bits que fan que s'activin de l'etapa 71 fins a la 75 (%M50 fins %M54).

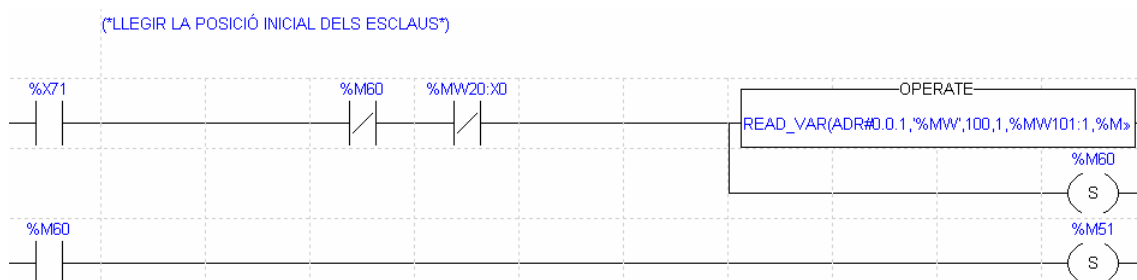


Figura 40. Lectura posició inicial esclaus

Per activar l'etapa 71 cal que estigui activat el bit %M50 i que la posició de l'esclau no sigui ni la primera, ni la segona, ni la posició intermitja, és a dir, que l'esclau no estigui a la ZONA 1, que no estigui a la ZONA 2 i que no estigui realitzant la TASCA 2.

Quan s'activi l'etapa 71 i ja s'hagi llegit la posició de l'esclau 1, seguidament s'activarà el bit que permetrà fer la lectura de l'esclau 2 (%M51) i s'activarà l'etapa 72. El procés serà el mateix pels altres esclaus.

Per tal de realitzar la lectura, s'ha organitzat una estructura amb timers que activaran els bits que permetran la lectura quan sigui convenient i de manera ordenada.

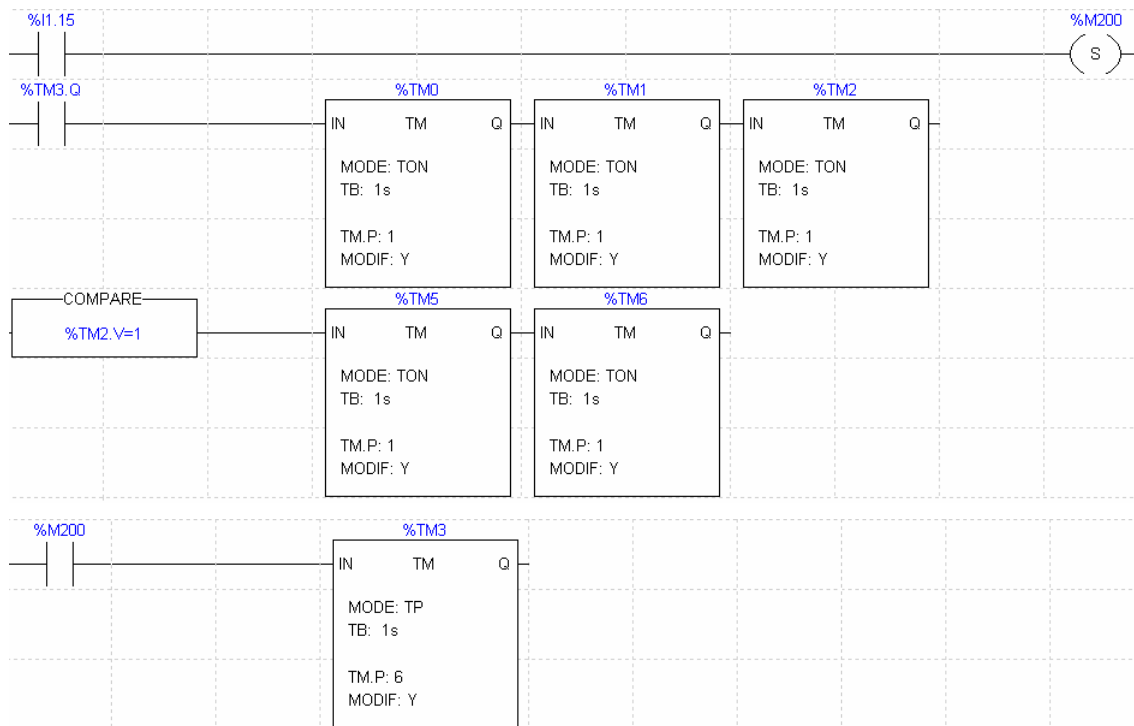


Figura 41. Lectura ordenada dels esclaus (1)

La primera vegada que s'executi el programa caldrà activar l'interruptor %I1.15. Aquest però, només s'haurà d'utilitzar un cop i prou, el primer cop per activar la lectura, i a partir d'aquí, ja s'executarà de manera automàtica.

L'interruptor activa el bit %M200 que posa en marxa el timer %TM3. Aquest timer activarà el timer %TM0. Els timers %TM0, %TM1, %TM2, %TM5, %TM6 estan programats per comptar un segon. Quan hagi passat un segon, un timer activarà l'altre (tal i com es veu a la Figura 41).

El %TM3 es programa a 6 segons i en mode TP. Aquest mode de funcionament del temporitzador permet tenir un impuls d'una durada precisa (en aquest cas concret és de 6 segons). Cal que la durada de l'impuls permeti que els altres timers tinguin temps d'activar-se.

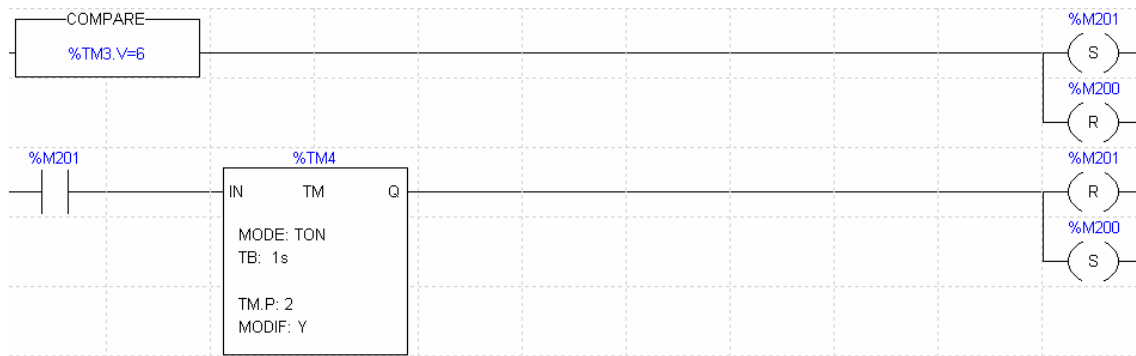


Figura 42. Lectura ordenada dels esclaus (2)

Quan %TM3 hagi comptat els 6 segons, activarà el bit %M201 i resetejarà el bit %M200. Llavors, quan el bit %M201 s'activi, el %TM4 comptarà 2 segons i s'activarà de nou el bit %M200 i resetejarà el %M201.

Aquesta acció permetrà fer que el timer %TM3 s'activi i torni a començar el procés de lectura de la primera posició dels esclaus.

%TM4 es configura a 2 segons per tal de que els bits tinguin temps d'activar-se i desactivar-se.

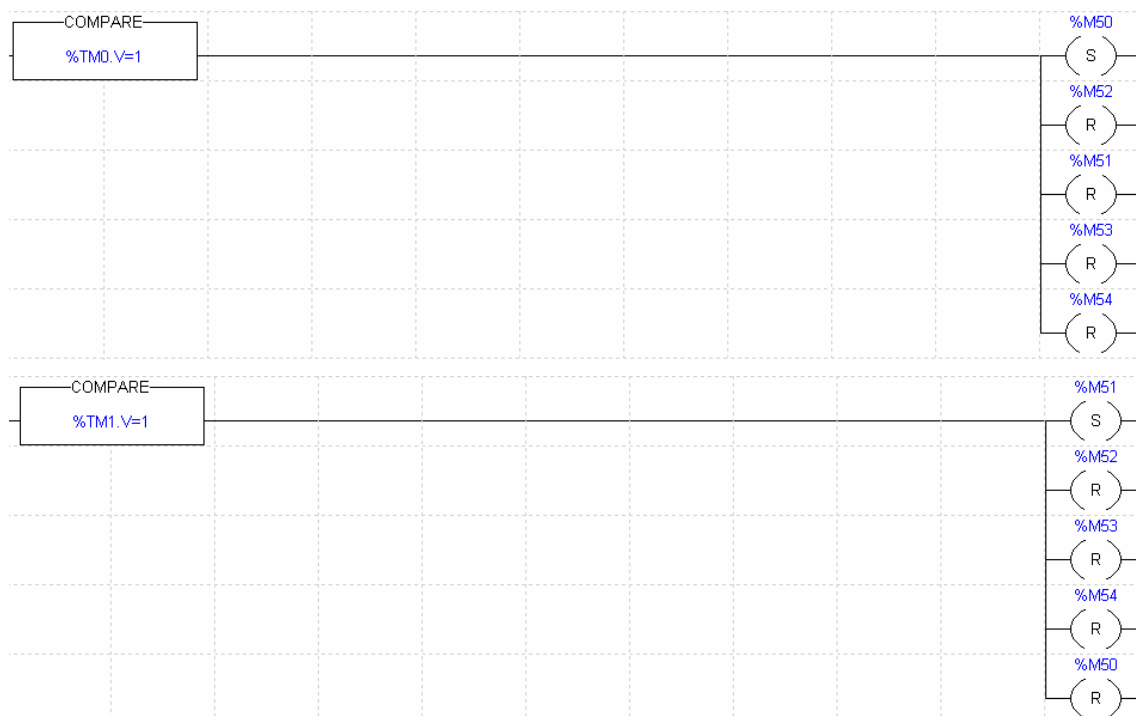


Figura 43. Activació i desactivació de bits %TM0 i %TM1

Quan els timers %TM0, %TM1, %TM2, %TM5 i %TM6 han comptat el temps predeterminat, s'activarà un bit un o un altre i resetejarà els altres per evitar que no es llegeixi cap esclau que no sigui el que toca.

Així doncs, quan %TM0.V = 1 s'activarà %M50, quan %TM1.V = 1 s'activarà %M51, quan %TM2.V = 1 s'activarà %M52, etc.

Cal recordar que aquests bits (%M50 fins %M54) són els que permeten activar les etapes 71, 72, 73, 74 i 75, que són les que fan la lectura pròpiament.

#### **A2.4.3.3 Sr0: Inicialització de variables**

La Sr0 té la funció de realitzar inicialitzacions de variables però que només es faran una vegada, quan es posi a RUN el programa i prou.

Es resetejen els bits que permeten accedir a les etapes 1, 2, 3, 4 i 5 on es dona pas als esclaus per realitzar la TASCA 1.

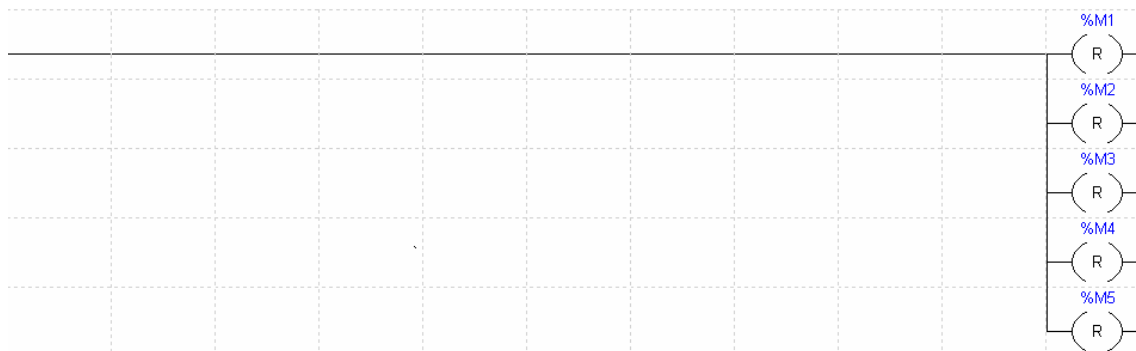


Figura 44. Inicialització de bits de pas a TASCA 1

Es resetejen els bits que donen per acabades les etapes 1, 2, 3, 4 i 5.



Figura 45. Inicialització de bits de fi de TASCA 1

A la Sr0 s'hi inicialitzen les paraules i punter que són necessaris per donar pas als esclaus.

%MW15 i %MW16 no caldria inicialitzar-los a 1 perquè quan el mestre escriu a l'esclau i li envia aquestes paraules, el que fa és enviar lo que conté la paraula %MW9 i ho envia amb la paraula %MW15 o %MW16, però per assegurar l'operació, es fa la inicialització a 1.

Recordar que la variable %MW50 és el punter de lectura de la taula de prioritats.

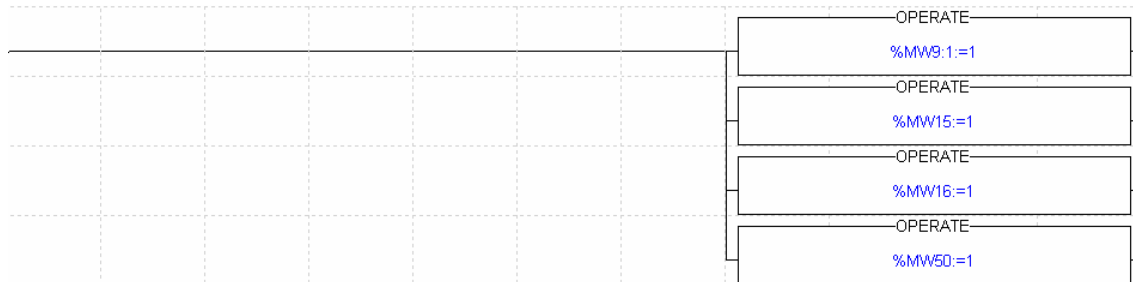


Figura 46. Inicialització de paraules i punter

Es resetejen els bits que permeten enviar a l'esclau la paraula de pas per accedir a la TASCA 1.

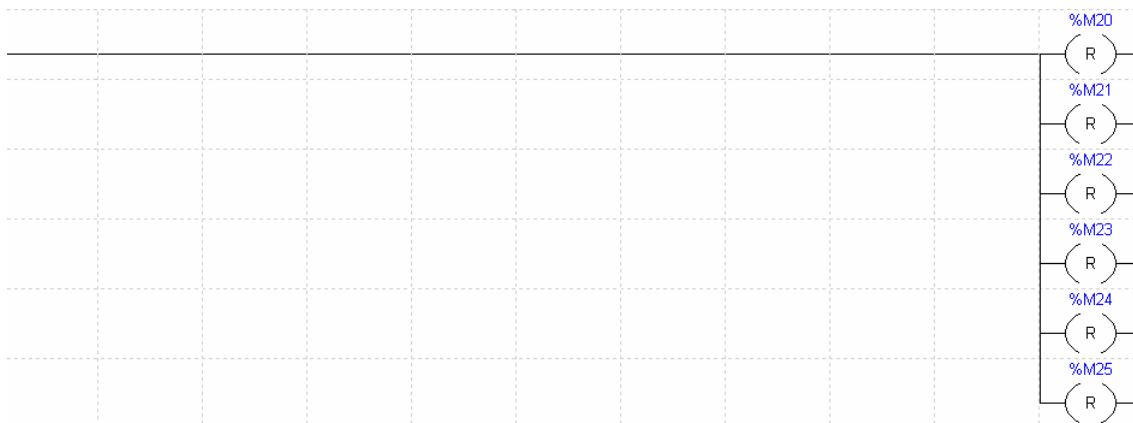


Figura 47. Inicialització de bits d'escriptura

Es resetejen els bits que permeten accedir a les etapes 11, 12, 13, 14 i 15 del grafcet principal, i els bits que assenyalen que els esclaus han arribat a la ZONA 3.

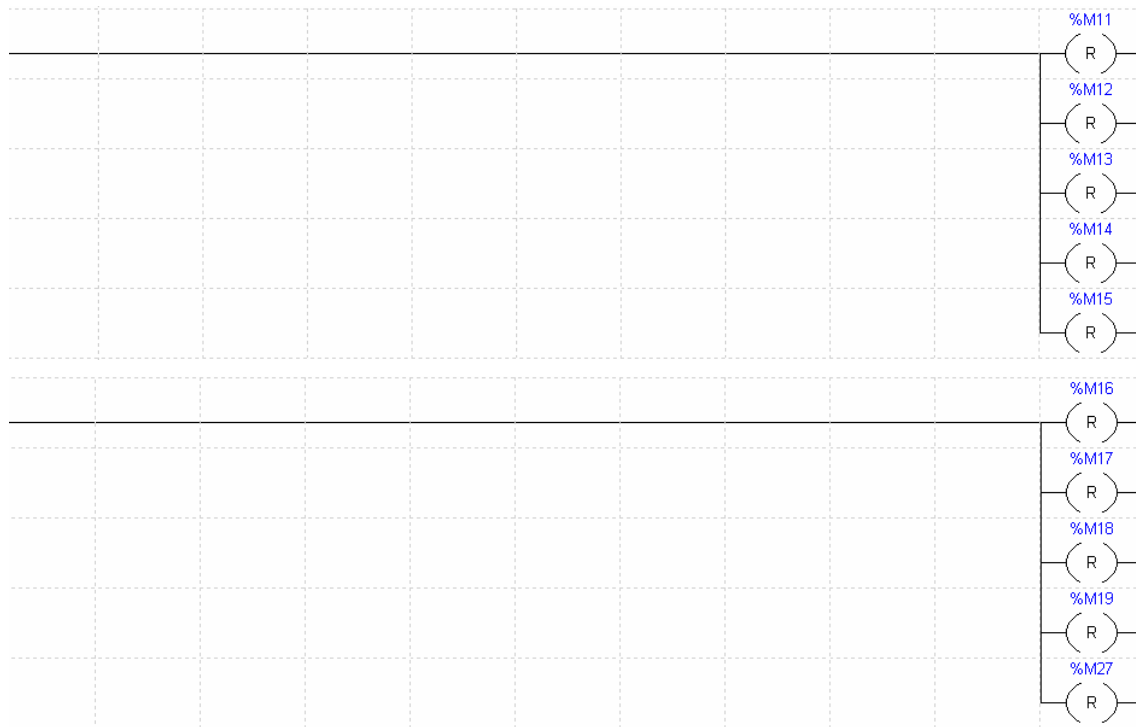


Figura 48. Reset dels bits de pas a TASCA 2 i esclau a ZONA 3

També es resetejen els bits que donen per acabada la lectura dels esclaus en la primera posició.

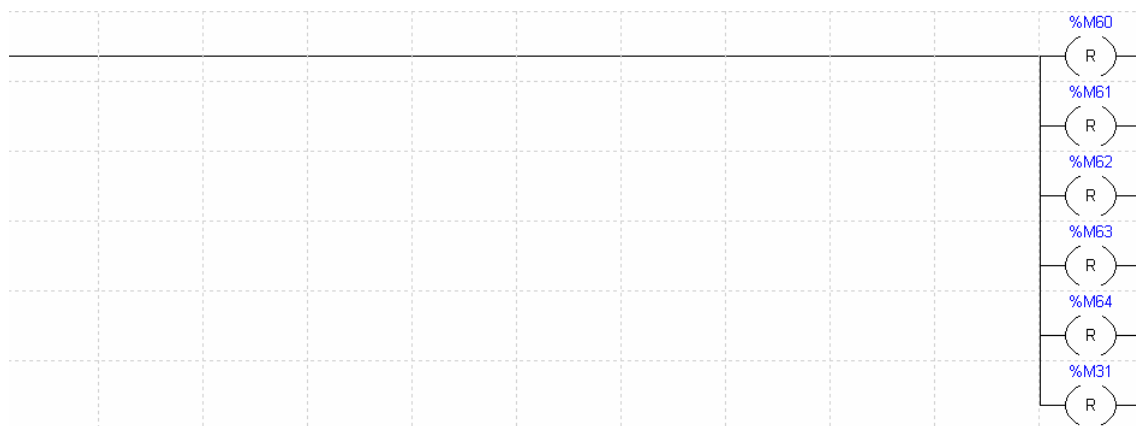


Figura 49. Reset dels bits de lectura posició 1 dels esclaus

#### A2.4.3.4 Sr1: Activació de bits segons taula de prioritats

Quan s'activa l'etapa 30 del grafcet principal, s'activa la Sr1. En aquesta part del grafcet principal, el mestre dóna pas als esclaus en funció de la taula de prioritats.



Aquest pas es dóna de manera automàtica segons l'ordre que ocupin els esclaus dins la taula de prioritats.

Per exemple, si al primer lloc de la taula hi ha guardat un 3, el bit que s'activarà serà el %M13, el qual permetrà activar l'etapa 13 on s'envia el pas a l'esclau 3.

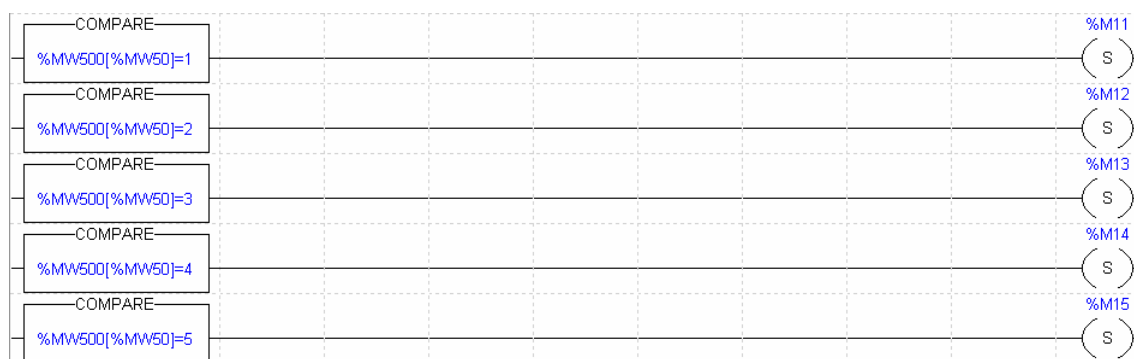


Figura 50. Activació de bits segons taula de prioritats

#### A2.4.3.5 Sr2 : Inicialització de variables

La Sr2 també serveix per inicialitzar variables, però aquestes inicialitzacions es faran cada vegada que s'activi l'etapa 0 del graficet principal.

Es resetejen els bits usats per l'escriptura als esclaus, els bits per accedir a etapes i bits que serveixen per fer saber al mestre que han arribat a una zona en concret.

Això es fa per tots els esclaus però la figura 51 només mostra les variables de l'esclau 1. No s'ensenyen les variables dels altres esclaus perquè seria molt extens i ja s'han esmentat anteriorment.

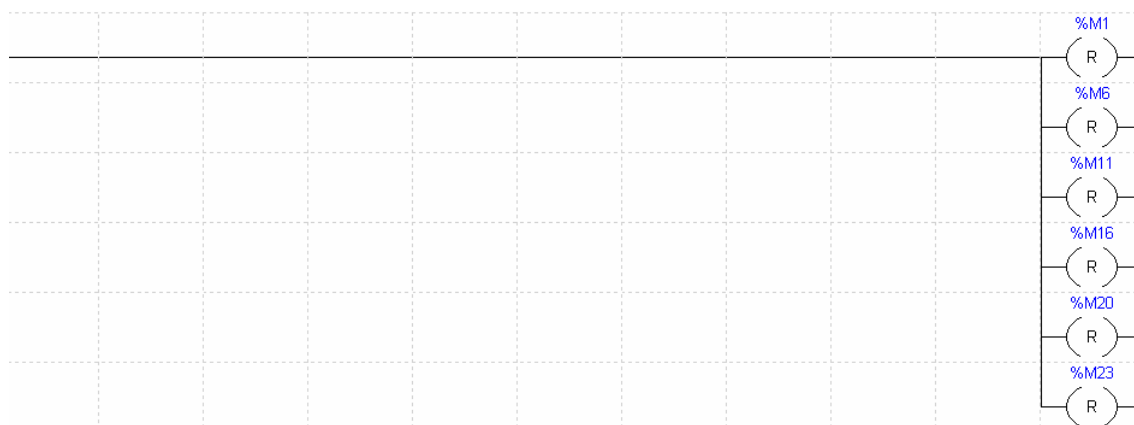


Figura 51. Inicialització dels bits referents a l'esclau 1

El punter de lectura de la taula de prioritats també s'inicialitza a 1 per tal de situar-lo a la primera posició de la taula.



Figura 52. Inicialització punter lectura taula de prioritats

## A2.5 Programa autòmat esclau

El programa de l'autòmat esclau simularà el procés amb el suport de la maqueta didàctica. Fent girar la maqueta representarà els diversos punts del procés, i haurà d'esperar a rebre els permisos del mestre per fer una cosa o una altra. El programa dels autòmats esclaus serà el mateix per a tots, amb l'única diferència que les posicions dels esclaus en les diverses parts de la simulació seran diferents per a cada un d'ells.

### A2.5.1 Llista d'entrades i sortides

La simulació del procés es fa mitjançant la maqueta didàctica.

ENTRADA PLC	ETIQUETA	ELEMENT MAQUETA
0	Ps1	Polsador 1
1	Ps2	Polsador 2
5	Fc1	Fi de cursa indicant posició esquerra del disc
6	Fc2	Fi de cursa indicant posició dreta del disc
7	Fc3	Sensor inductiu de posició de comptatge

Taula 30. Entrades al PLC (maqueta)

SORTIDES PLC	ETIQUETA	ELEMENT MAQUETA
0	Drt	Moviment a dretes
1	Esq	Moviment a esquerres
2	Lnt	Velocitat lenta
3	Rpd	Velocitat ràpida
4	Lp1	Làmpada 1
5	Lp2	Làmpada 2

Taula 31. Sortides del PLC (maqueta).

### A2.5.2 Descripció

El procés comença amb el disc a la seva posició inicial, és a dir, a l'esquerra (Fc1 = 1). En prémer el polsador 1 (Ps1) de la maqueta, el mestre rep el senyal de que hi ha un esclau a la ZONA 1 i que està a punt per entrar a fer la TASCA 1.

Si no hi ha cap esclau realitzant la primera tasca, l'esclau rebrà l'ordre de pas del mestre (%MW15 = 1) i s'encendrà el led 1 (Lp1) que indicarà el pas de forma visual. Llavors, el disc haurà de comptar 3 peces metàl·liques (per simular que la barrera s'aixeca i arriba a la TASCA pròpiament), parar durant 2 segons i s'haurà d'apagar el led 1. Passats els 2 segons, el disc girarà i comptarà 5 peces metàl·liques més, i llavors pararà 1 segon. Fet això, ja haurà acabat la TASCA 1.

El disc girarà i es comptaran 4 peces metàl·liques més, s'aturarà i esperarà a rebre l'ordre del mestre (%MW16 = 1). En aquest moment, l'esclau estarà situat a la ZONA 2, esperant per entrar a realitzar la TASCA 2.

Quan %MW16 = 1, el led 2 (Lp2) s'encendrà i el disc girarà comptant 6 peces més. S'aturarà 3 segons i el led 2 s'apagarà. El disc haurà de girar fins que el fi de cursa Fc2 el detecti i llavors es considerarà que la TASCA 2 haurà finalitzat.

Per tornar a la posició inicial el disc haurà de girar cap a l'esquerra i a velocitat ràpida. Quan el sensor de posició esquerra del disc (Fc1) detecti, haurà de parar.

L'esclau podrà prendre 4 possibles posicions que permetran al mestre saber on i com es troba en qualsevol moment. Aquestes posicions seran les mateixes per a tots els esclaus però no s'escriuran igual. Això es fa per evitar confusions al mestre.

SIMBOLOGIA	ZONA
X 0 1	ZONA 1
X 0 2	ZONA 2
X 1 3	ZONA INTERMITJA
X 0 3	ZONA 3

Taula 32. Simbologia de les zones

La "X" es substituirà a cada programa esclau per l'esclau corresponent. Per l'esclau 1 les posicions seran 101, 102, 113 o 103, per l'esclau 2 seran 201, 202, 213 o 203, etc.

### A2.5.3 Programa

El programa dels esclaus és senzill i, tot i que s'ha fet ús del grafcet, el grafcet de l'esclau no és necessari. El programa és el mateix per a tots però canvien les posicions i les direccions dels esclaus.



Figura 53. Estructura del programa esclau.

#### A2.5.3.1 Secció Chart

Pel programa esclau no és necessari fer el grafcet però, per tal de facilitar la comprensió del procés, s'ha escrit el grafcet. El grafcet només serveix per veure de manera més clara el passos que segueix l'esclau però el programa no s'hi referència. El grafcet és el mateix per a tots els esclaus.

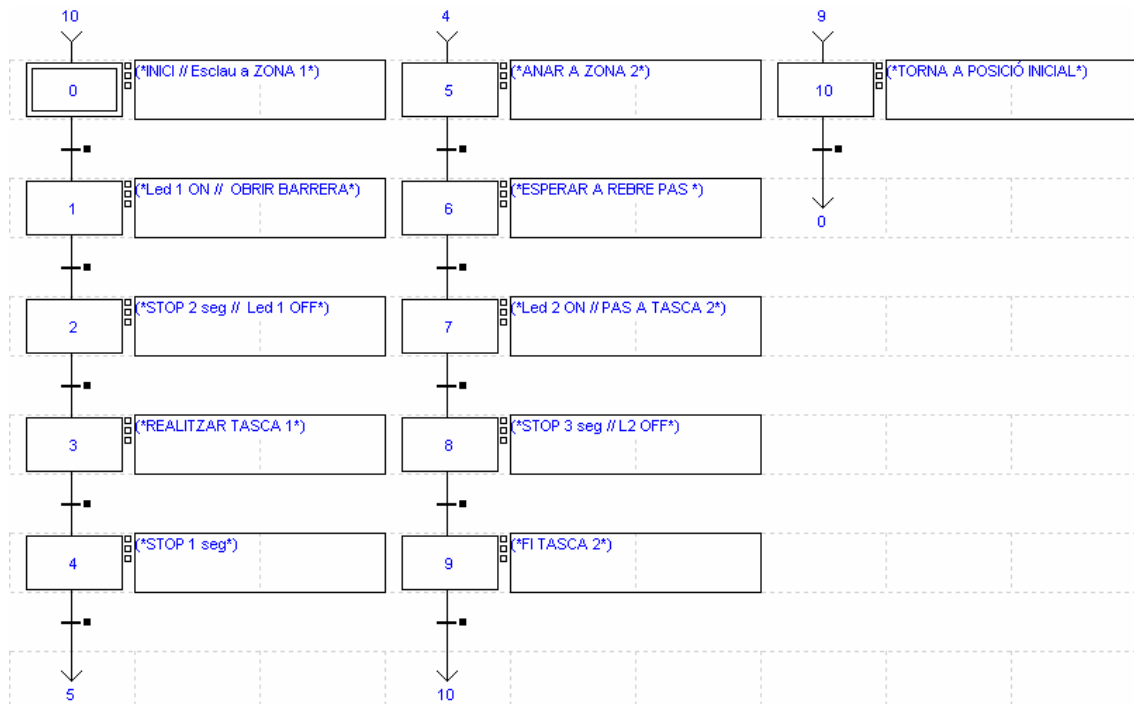


Figura 54. Graficet dels esclaus.

### A2.5.3.2 Secció Post

El programa esclau s'ha escrit en llenguatge de contactes (LD).

Primer de tot, cal assegurar-se que el disc de la maqueta estigui a la posició inicial.

En començar el programa s'inicialitzen les paraules que donen pas a l'esclau per accedir a la TASCA 1 i a la TASCA 2. Quan aquestes paraules valguin 1, l'esclau realitzarà les tasques.

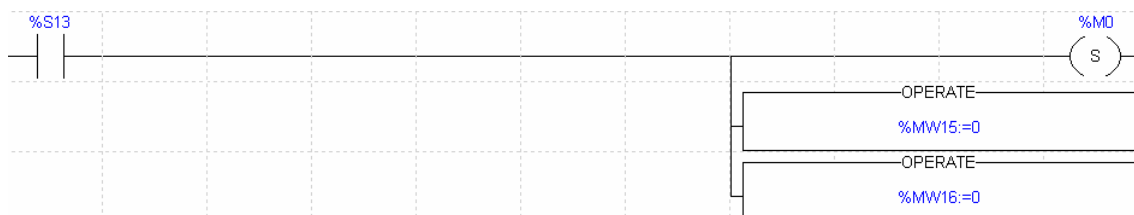


Figura 55. Inicialització de variables de l'esclau

En activar-se el bit %M0 s'iniciarà el procés i permetrà resetejar tots els comptadors.

Quan es premi el polsador Ps1 (%I1.0), l'esclau estarà a la primera posició de la terminal de buidat i neteja de contenidors. La posició que es mostra a la Figura 56 correspon l'esclau 1. Per l'esclau 2 la posició serà 201, 301 per l'esclau 3, etc.

La paraula %MW100 és on es guarda el valor de la posició dels esclaus i aquesta paraula és igual per a tots els esclaus.

Quan el mestre doni pas a l'esclau (%MW15 = 1), s'encendrà el led 1 i el disc girarà a dretes i a velocitat lenta.

Quan s'activi %M1, la paraula de pas per accedir a la TASCA 1 es posarà a 0.

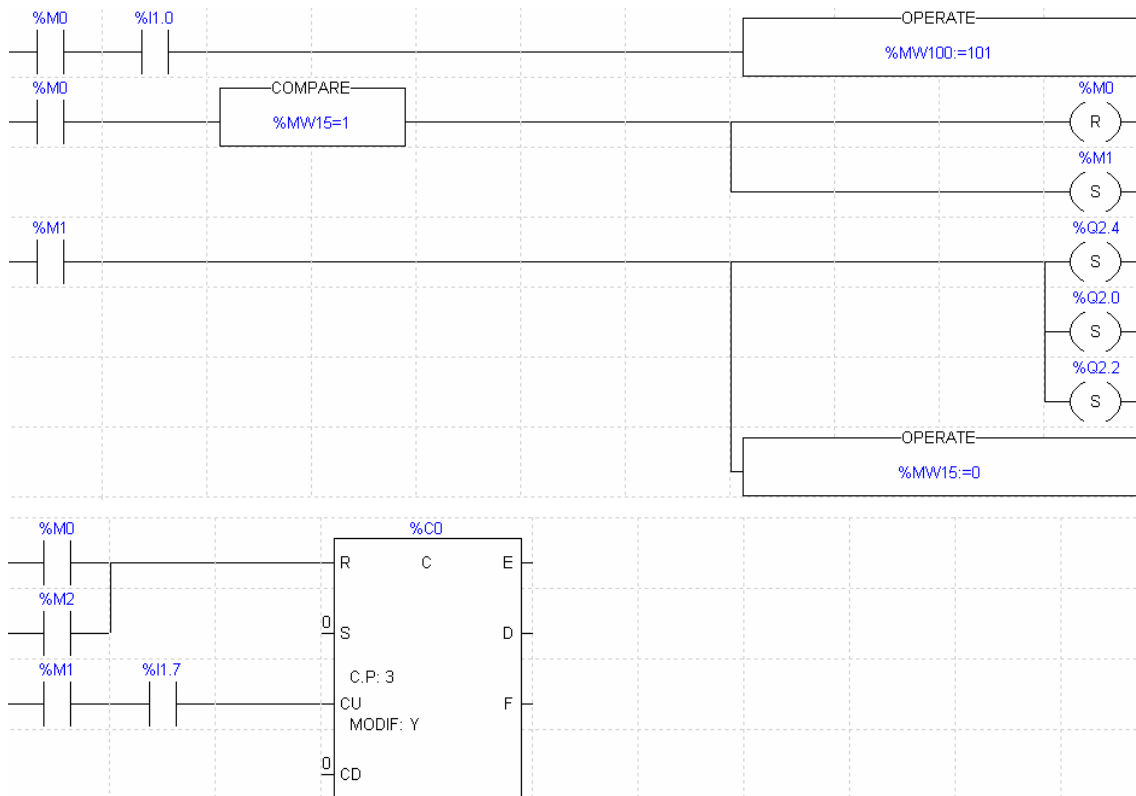


Figura 56. Accedir a la TASCA 1

El comptador %C0 s'activarà i comptarà 3 peces metàl·liques. El comptador es resetejarà quan s'activin els bits %M0 o %M2.

Quan s'hagin comptat les 3 peces, el disc s'atura durant 2 segons (%TM0).

Passats els 2 segons es reseteja el bit %M2 i s'activa el %M3. Aquest darrer bit permetrà que l'esclau faci la TASCA 1.

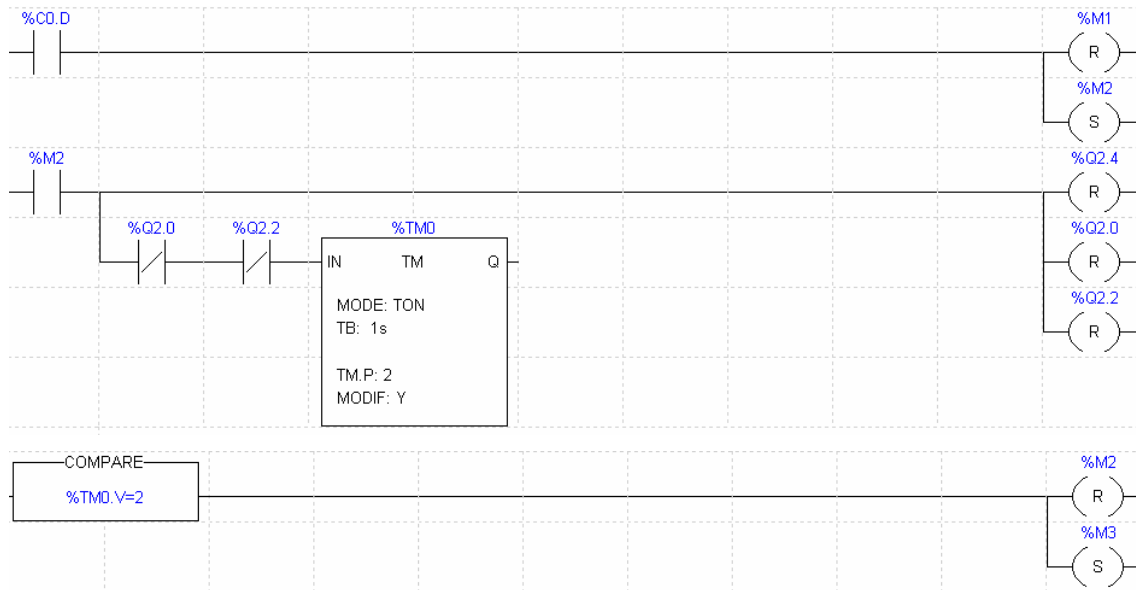


Figura 57. Entrar a realitzar TASCA 1

La TASCA 1 consisteix en comptar 5 peces metàl·liques i fer una parada d'un segon.

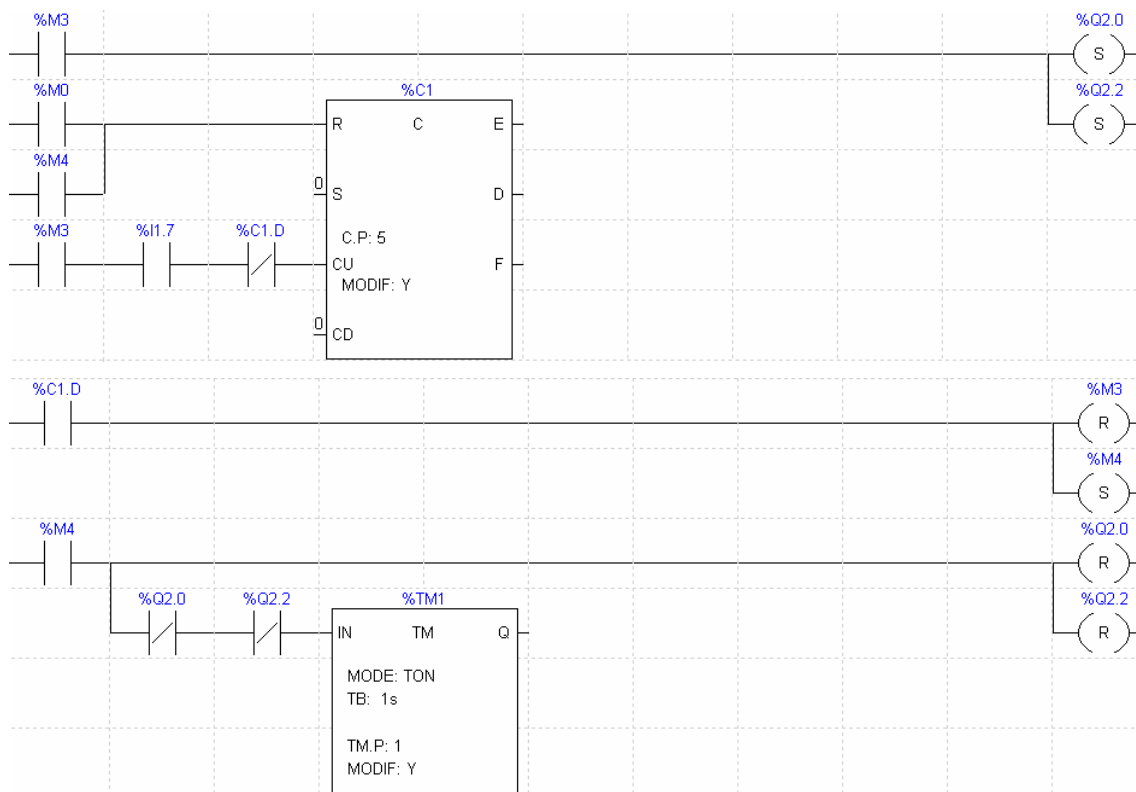


Figura 58. Realització de la TASCA 1

Quan l'esclau hagi simulat la neteja del contenidor s'haurà de dirigir cap a la ZONA 2 (zona on l'esclau haurà d'esperar fins que no hi hagi cap esclau realitzant la TASCA 2 per tal d'accedir-hi).

Per simular-ho, el comptador %C2 comptarà 4 peces metàl·liques.

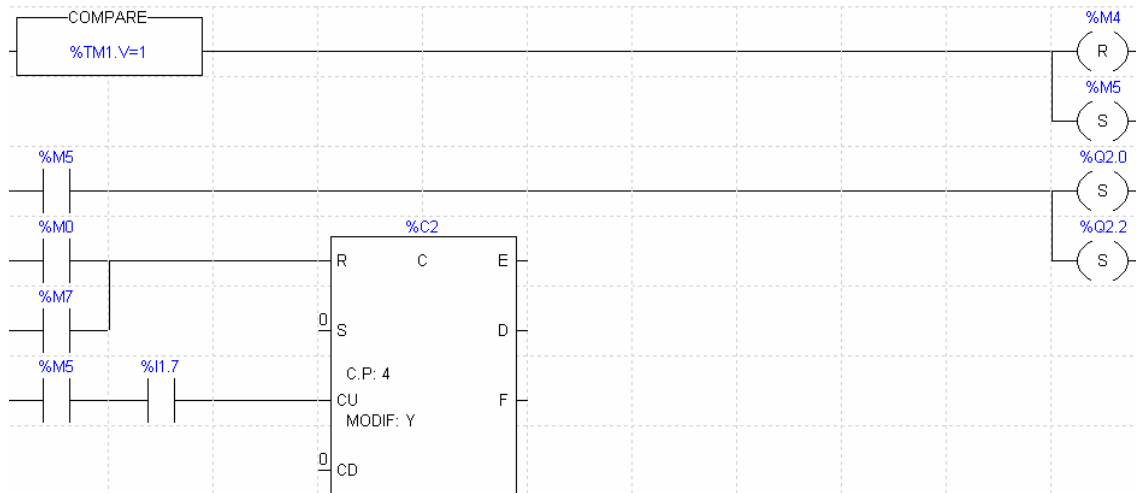


Figura 59. Anar a la ZONA 2

Quan s'hagi arribat a la ZONA 2, la posició de l'esclau serà x02 (on "x" serà l'esclau). Llavors caldrà esperar a que %MW16 valgui 1 per poder accedir a la TASCA 2.

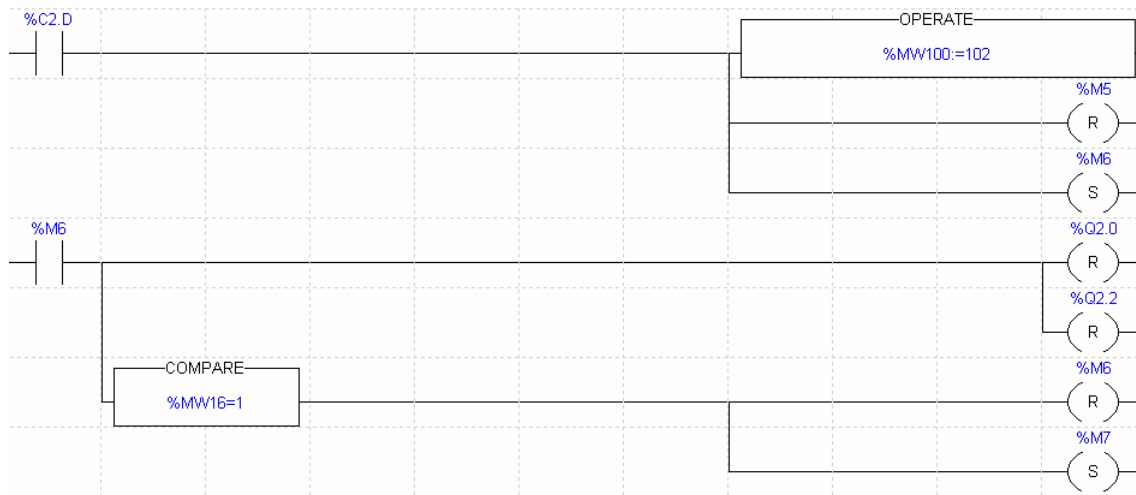


Figura 60. ZONA 2 i esperar per accedir a TASCA 2

Quan el mestre doni permís a l'esclau per realitzar la TASCA 2, el led 2 (%Q2.5) de la maqueta didàctica s'encendrà.



Mentre s'estigui realitzant la TASCA 2, l'esclau estarà a una posició intermitja que és la x13 (on "x" variarà segons l'esclau).

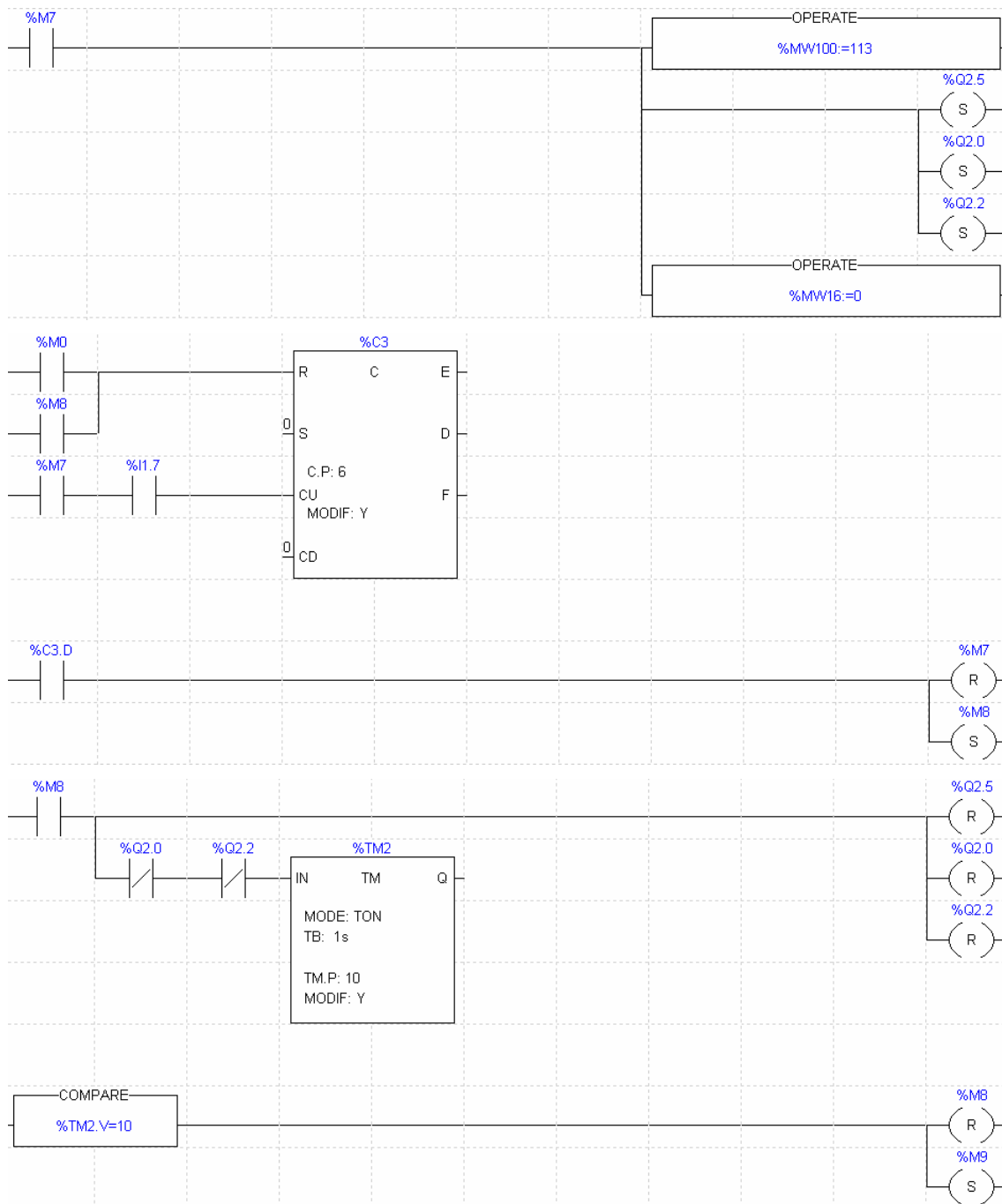


Figura 61. Realitzar TASCA 2

La TASCA 2 consistirà en comptar 6 peces metàl·liques i fer una parada de 10 segons. Un cop acabada la neteja dels contenidors dels camions (esclaus), s'hauran de dirigir cap a la ZONA 3 (zona de sortida de la terminal de buidat i neteja de contenidors). Quan el fi de cursa Fc2 detecta (%I1.6), llavors s'arriba a la posició x03.

## Creació d'una xarxa de comunicacions modbus pels autòmats telemecanique TsxMicro

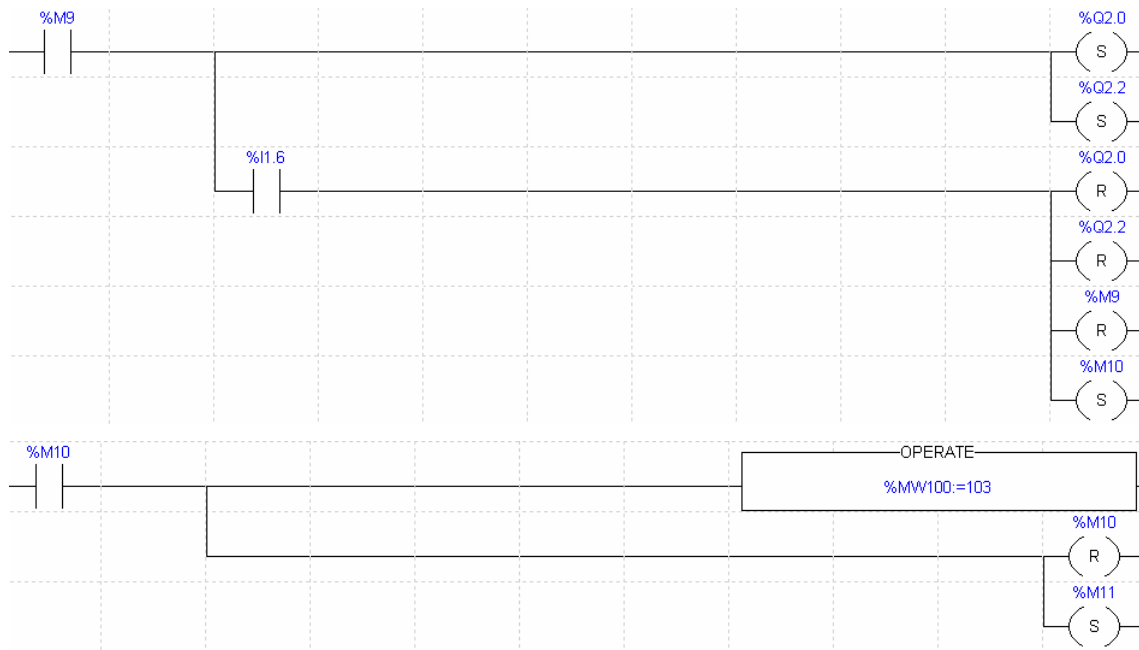


Figura 62. ZONA 3 de la terminal

Quan l'esclau arribi a l'última posició haurà de retornar de manera automàtica a la posició inicial (%I1.5).

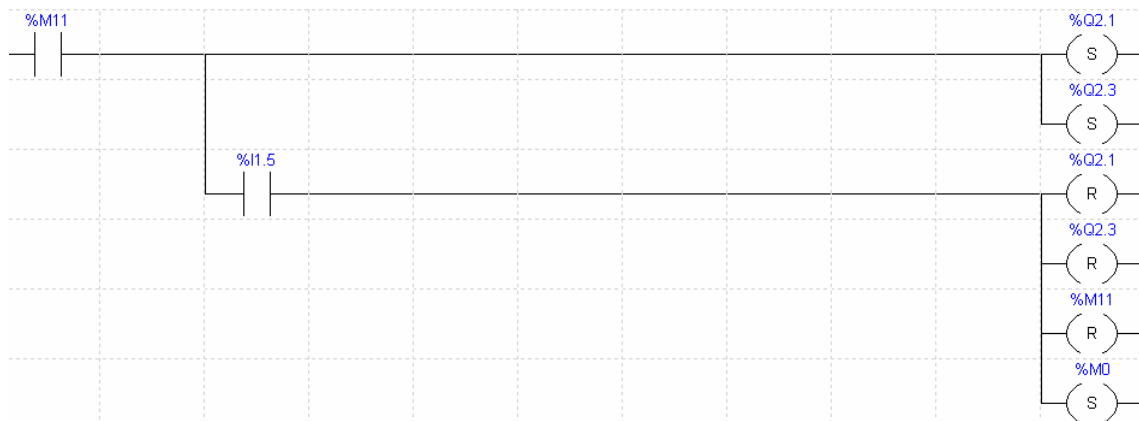


Figura 63. Retorn a la posició inicial.

### **A3. QÜESTIONARI DE COMPRENSIÓ**

El qüestionari te l'objectiu de valorar si l'alumne comprèn el material que se li ha proporcionat, és a dir, si te clar els conceptes bàsics d'una comunicació modbus.

Perquè creus que, podent usar una comunicació Uni-telway (que és la pròpia dels autòmats Telemecanique), s'usa una comunicació Modbus.

Quina estructura segueix una comunicació modbus.

Quina és la funció de l'autòmat mestre.

Per a què serveix la caixa TSX PACC 01.

Com es configuren els interruptors S1 i S2.

Quines són les funcions bàsiques que s'usen per a una comunicació modbus.

Donades les funcions següents, d'on llegirà el mestre les dades i on les guardarà en els dos casos.

```
READ_VAR(ADR#0.0.4,'%MW',1,4,%MW3:6,%MW80:4)
```

```
WRITE_VAR(ADR#0.0.4,'%MW',1,4,%MW3:6,%MW80:4)
```

Quan s'hagi de configurar la comunicació (en el cas concret de les pràctiques realitzades al laboratori de regulació), perquè s'ha d'escollir el mode RTU i no el mode ASCII.

### **A4. EXPERIÈNCIA 1. TASQUES A REALITZAR PER L'ALUMNE**

Un cop vista i entesa la simulació del procés de buidat i neteja de contenidors, i servint com a referència, l'alumne haurà de realitzar les tasques següents.

#### **A4.1 Tasca 1**

Fer un programa que faci moure les maquetes del laboratori de regulació, de forma sincronitzada i per ordre, és a dir, primer haurà de girar la que està a la direcció 1, llavors la 2, etc.

Primer, el disc haurà de girar fins a la peça número 10, començant per la maqueta de l'esclau 1, continuant per la 2, etc. Llavors, quan la última hagi realitzat el gir, haurà de tornar a la posició inicial (començant per la última maqueta, i tornant cap a la que havia estat la primera).

Caldrà realitzar el programa dels autòmats esclaus (s'aconsella usar el llenguatge de contactes LD) i el programa de l'autòmat mestre (no és necessari fer el graficet).

El programa serà el mateix per a tots els esclaus.

#### **A4.2 Tasca 2**

Fer un programa que faci moure les maquetes (igual com abans, comptar 10 peces i tornar) segons l'ordre que se li hagi donat prèviament amb el mestre. S'haurà de crear una memòria FIFO on s'introduiran les direccions per establir la prioritat. També seria útil fer una taula d'animació per poder canviar les prioritats i veure el comportament del procés.

Caldrà realitzar el programa dels autòmats esclaus (s'aconsella usar el llenguatge de contactes LD), i el programa autòmat mestre (no és necessari fer el graficet ).

#### **A4.3 Tasca 3**

Fer un programa que, havent passat un temps determinat, si l'esclau no respon, fer que el mestre salti l'esclau i passi al següent.

Per fer això, caldrà que prèviament es faci una mesura de la durada del procés de cada esclau.

Si es supera el temps determinat per a cada esclau, s'encendrà un led de falla (indicador de l'esclau que no respon), i el procés haurà de continuar.

Es recomana usar la botonera de que disposa el laboratori.

Caldrà realitzar la mesura de la durada del procés de cada esclau, construir el programa de l'autòmat esclau i construir el programa de l'autòmat mestre.