



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



ESCOLA TÈCNICA  
SUPERIOR ENGINYERS  
INDUSTRIALS VALÈNCIA

Curs Acadèmic:

# RESUM

---

El treball a realitzar consistirà en desenvolupar el programari que calga de manera que siga possible la comunicació entre un PLC i un robot col·laboratiu industrial emprant el protocol industrial de comunicació estàndar Modbus. Açò permetrà el desenvolupament d'aplicacions de robòtica industrial que impliquen una certa sincronització entre elements gestionats per un PLC encarregats d'algun tipus de funcionament automàtic i un robot industrial.

**Paraules clau:** PLC, robot, MODBUS.

# ÍNDEX

<b>MEMÒRIA.....</b>	<b>4</b>
1.- Introducció .....	11
2.- Antecedents.....	12
3.- Motivació.....	13
4.- Objectius del TFG.....	14
5.- Autòmats utilitzats .....	15
6.- Robot .....	17
7.- Programes utilitzats.....	20
8.- Funcionament.....	23
9.- Programació .....	28
10.- Comunicació .....	33
11.- LabView .....	38
12.- Manual del programador .....	47
13.- Conclusions.....	67
14.-Vocabulari.....	68
15.- Bibliografia.....	69
<b>PRESSUPOSTOS .....</b>	<b>70</b>
1.- Introducció .....	73
2.- Costos Unitaris.....	74
3.- Preus unitaris descompostos .....	76
4.- Pressupost total d'execució.....	83
5.- Pressupost total d'execució per contracta .....	83
6.- Pressupost base de licitació .....	83
<b>ANNEX.....</b>	<b>84</b>
1.- Relació Variables .....	86

# MEMÒRIA

---

# ÍNDEX DE CONTINGUTS

<b>1.- INTRODUCCIÓ .....</b>	<b>11</b>
<b>2.- ANTECEDENTS .....</b>	<b>12</b>
2.1.Automatització.....	12
2.2.PLC .....	12
2.3.MOVBUS .....	12
<b>3.- MOTIVACIÓ .....</b>	<b>13</b>
<b>4.- OBJECTIUS DEL TFG .....</b>	<b>14</b>
<b>5.- AUTÒMATS UTILITZATS.....</b>	<b>15</b>
<b>6.- ROBOT .....</b>	<b>17</b>
6.1.Ferramenta .....	17
6.2.Moviment .....	17
6.2.1 Graus de llibertat .....	17
6.2.2 Espai de treball .....	17
6.2.3 Càrrega màxima.....	18
6.2.4 Velocitat.....	18
6.3.Parada.....	18
6.3.1 Parada de protecció.....	18
6.3.2 Parada d'emergència .....	19
6.4.Configuració .....	19
6.5.Programació .....	19
<b>7.- PROGRAMES UTILITZATS .....</b>	<b>20</b>
7.1. SoMachine .....	20
7.2.PolyScope.....	21

<b>8.- FUNCIONAMENT .....</b>	<b>23</b>
8.1.Funcionament general.....	23
8.2.Modes de funcionament .....	23
8.2.1 Pas a pas .....	23
8.2.2 Automàtic .....	24
8.2.3 Manual.....	24
8.2.4 Parada d'emergència .....	25
8.2.5 Botó emergència del robot.....	25
8.2.6 Alarmes.....	26
<b>9.- PROGRAMACIÓ .....</b>	<b>28</b>
9.1.Cinta Transportadora .....	28
9.2.Cintes Fischer Technik .....	29
9.2.1 Cinta 1.....	29
9.2.2 Cintes23 .....	29
9.2.3 Cinta4.....	30
9.2.4 PFi .....	30
9.2.5 PEmergencia .....	30
9.3.Robot .....	31
<b>10.- COMUNICACIÓ .....</b>	<b>33</b>
10.1.Comunicació MODBUS .....	33
10.1.2 Mestre/Esclau.....	33
10.1.3 Servidor/Client.....	33
10.1.4 Comunicació MODBUS Servidor/Client entre autòmats iguals o semblants .....	35
10.1.5 Comunicació MODBUS Servidor/Client (Cinta/Robot i Cintes/Robot).....	36
10.1.5.1- Cinta/Robot.....	36
10.1.5.2- Cintes/Robot .....	37
<b>11.- LABVIEW .....</b>	<b>38</b>
11.1.MODBUS .....	38
11.2.Variables .....	38
11.3.VI.....	39
11.3.1 Diagrama de blocs .....	40

11.3.2. Manual de l'usuari: SCADA .....	41
11.3.2.1- Pantalla de visualització .....	42
11.3.2.2- Pantalla de control .....	44
<b>12.- MANUAL DEL PROGRAMADOR .....</b>	<b>47</b>
<b>12.1.Entrades/Eixides .....</b>	<b>47</b>
12.1.1 Cinta transportadora .....	47
12.1.2 Cintes Fischer.....	48
12.1.3 Robot .....	50
<b>12.2.Variables .....</b>	<b>51</b>
12.2.1 Cinta transportadora .....	51
12.2.1.1- Variables Compartides .....	51
12.2.1.2- Variables de la cinta .....	52
12.2.2 Cintes Fischer.....	52
12.2.2.1- Variables compartides.....	52
12.2.2.2- Variables de les cintes .....	53
12.2.3 Robot .....	54
12.2.3.1- Variables compartides.....	54
12.2.3.2- Variables del programa .....	55
12.2.3.3- Variables d'instal·lació.....	55
<b>12.3.Graficets .....</b>	<b>56</b>
12.3.1 Cinta transportadora .....	56
12.3.2 Cintes Fischer.....	57
<b>12.4.Programa del robot .....</b>	<b>60</b>
<b>12.5.Mòduls de funció.....</b>	<b>61</b>
12.5.1 Temporitzador .....	61
12.5.2 MOVE.....	62
12.5.3 DWORD_AS_BIT.....	62
12.5.4 WORD_AS_BIT .....	63
12.5.5 Comptador.....	63
12.5.6 Comparador.....	64
12.5.7 ADDM.....	65
12.5.8 READ_VAR.....	65
12.5.9 WRITE_VAR.....	66
<b>12.6. SFCInit .....</b>	<b>66</b>
<b>13.- CONCLUSIONS .....</b>	<b>67</b>

**14.-VOCABULARI..... 68**

**15.- BIBLIOGRAFIA..... 69**



## ÍNDIX DE FIGURES

Figura 1. Connexions dels programes i autòmats utilitzats .....	14
Figura 2. Cinta ABB .....	15
Figura 3. Motors i sensors del procés de cintes Fischer.....	16
Figura 4. Robot UR3.....	16
Figura 5. Relació entre la càrrega màxima i la compensació del centre de gravetat.....	18
Figura 6. PolyScope. Pantalla tàctil de programació del robot. ....	22
Figura 7. Comunicació mestre-esclau .....	33
Figura 8. Comunicació servidor-client.....	34
Figura 9. Diagrama de blocs del VI .....	41
Figura 10. Pantalla de visualització del SCADA.....	43
Figura 11. Pantalla de control del SCADA.....	45
Figura 12. Pantalla de moviment manual de la cinta ABB al SCADA.....	46
Figura 13. Pantalla de moviment manual de les cintes Fischer al SCADA.....	46
Figura 14. Sensors i motor de la cinta ABB.....	47
Figura 15. Sensor i motors de les cintes Fischer. ....	48
Figura 16. Pantalla de moviment del robot del PolyScope. ....	50
Figura 17. Grafcet CintaABB. ....	56
Figura 18. Grafcet el robot deixa peça a la cinta.....	56
Figura 20. Grafcet Emergencia .....	57
Figura 19. Grafcet Cinta 1.....	57
Figura 21. Grafcet Cintes23 .....	58
Figura 22. Grafcet Cinta4.....	59
Figura 23. Programa del robot. ....	60
Figura 24. Temporitzador .....	61
Figura 25. MOVE.....	62
Figura 26. DWORD_AS_BIT .....	62
Figura 27. WORD_AS_BIT.....	63
Figura 28. Comptador.....	63
Figura 29. Comparador.....	64

Figura 30. ADDM .....	65
Figura 31. READ_VAR .....	66
Figura 32. WRITE_VAR.....	66
Figura 33. SFCInit.....	66

## **ÍNDIX DE TAULES**

Taula 1. Entrades cinta ABB.....	47
Taula 2. Eixides cinta ABB.....	47
Taula 3. Entrades cintes Fischer .....	48
Taula 4. Eixides cintes Fischer .....	49

## 1.- Introducció

Actualment, la comunicació entre autòmats és molt important i ja quasi imprescindible per a l'automatització de processos. Ens resulta necessari que les màquines es troben connectades entre elles de forma que puguin intercanviar dades i coordinar-se per tal de realitzar una feina conjunta.

A més, per poder comunicar dos màquines no ens resulta necessari que siguin del mateix tipus, ni tan sols del mateix fabricant. Açò és un avantatge molt gran perquè ens permet comunicar qualsevol tipus de màquines que adquirim. Per tant, a l'hora de comprar una d'aquestes no ens hem d'estar fixant en que siga possible la comunicació amb altra, si no en coses més importants com les prestacions que pot donar, l'adaptabilitat al nostre procés i si s'ajusta al nostre pressupost.

A banda, la comunicació és molt important també per a la supervisió i control. Ens permet controlar una màquina o procés a distància, creant una pantalla de control (SCADA) i comunicant-la amb aquella màquina que volem governar.

Tant la comunicació entre autòmats com l'automatització de processos són importantíssims en aquest moment i en un futur, i a més, són punts claus en la indústria 4.0.

La indústria 4.0 és la transformació digital de les empreses. És el que s'espera amb la quarta revolució industrial, que les empreses es digitalitzen de forma que els processos tinguen major flexibilitat i siguin més efectius. Es caracteritza per la intercomunicació de les màquines de la pròpia fàbrica i la comunicació amb l'exterior, per trobar-se en contacte amb el mercat.

La clau és la creació d'un sistema organitzat mitjançant una xarxa de comunicacions i amb un intercanvi permanent i instantani d'informació. Com que el flux d'informació és regular i pot ser intercanviada internament i externament, i a més de forma molt ràpida, permet l'adaptació a situacions irregulars o canviant, tant en la pròpia fàbrica com a nivell del mercat.

A més, aquest tipus d'indústria té grans avantatges de costos: es redueixen els costos tant de mà d'obra i de materials com els costos energètics; és també, per tant, una forma de producció més neta i amb un compromís energètic major. Açò últim és un punt molt important en les empreses, ja que cada vegada se li dóna més importància a la conscienciació amb el medi ambient i al concepte d'eco-friendly. Per altra banda, serà un efecte en cadena, perquè una vegada alguna empresa pose aquesta pràctica en marxa, si les altres volen continuar al mercat, hauran d'unir-se ja que del contrari no podran competir amb els preus de les fàbriques intel·ligents.

És molt important a la indústria 4.0 la informació per a analitzar els processos, controlar-los i comunicar-los amb altres. En aquest treball realitzem la comunicació d'autòmats mitjançant una xarxa interna (Ethernet), intercanviant la informació necessària, i el control del procés amb un SCADA, per tant, aquest procés podria incorporar-se a una empresa intel·ligent, ja que amb aquest protocol de comunicació utilitzat (MODBUS), podem comunicar-nos amb qualsevol altra màquina o procés.

## **2.- Antecedents**

### **2.1. Automatització**

L'automatització és l'assignació de feines a una màquina sense necessitat d'intervenció d'una persona. A mesura que passa el temps, i amb la millora de la tecnologia, ha anat guanyant molta importància, cada vegada és major el nombre de processos que es poden automatitzar i continuarà avançant.

Amb l'automatització s'ha aconseguit un avanç molt important, ens permet realitzar feines pesades sense massa esforç i realitzar-les de forma ràpida, precisa i amb major qualitat.

Encara que es tem pels llocs de treball de la gent, l'automatització no tracta de la substitució del ser humà als llocs de treball per reduir costos salarials, si no d'obtenir una millor producció gràcies a la realització d'un treball més eficient. A més, són necessaris menys operaris per a realitzar feines molt mecàniques que poden ser realitzades per una màquina però hem de tenir en compte que darrere d'una màquina sempre hi ha un humà que s'encarrega de la seua programació, el seu manteniment...

Sense l'automatització no seria possible el nivell de producció al què s'ha arribat aquestos darrers anys.

### **2.2. PLC**

Un PLC o controlador lògic programable, com el seu nom indica, es programat per a controlar màquines o processos. Són molt importants al camp de l'automàtica, amb aquestos es poden automatitzar processos de forma molt ràpida.

Van aparèixer a la indústria de l'automòbil a finals dels anys 70, substituïen els relés. Amb aquestos es reduïa el temps i el cost dels processos. La seua programació és senzilla i més fàcil d'entendre pels programadors. Amb els anys, els PLCs són clau en l'automatització i s'ha anat reduint el tamany i el cost d'aquestos de forma que són accessible a tots, fins i tot per a màquines de poc d'ús i, a més són d'aplicació en gran varietat de camps.

A banda de la programació, els PLCs ens permeten la comunicació, accedint a altre PLC, compartint dades per formar un únic procés...

### **2.3. MODBUS**

EL MODBUS és un protocol de comunicació entre autòmats. Els avantatges de l'ús del MODBUS són que és gratuït, estàndard i no massa complicat. La història va lligada a l'aparició dels PLCs, amb la comunicació entre aquestos es va aconseguir eliminar el cablejat i amb això la reducció del cost.

Hi ha dos tipus de comunicació, mestre/esclau (RTU) i client/servidor (TCP/IP).

### 3.- Motivació

El treball que realitzem tracta de l'automatització d'un procés i la comunicació d'aquest amb un robot de forma que aconseguim un procés conjunt i connectat.

Tenim un procés de producció en el què intervé una cinta transportadora per a traslladar la peça d'un punt a altre. Podria ser una peça situada a una fàbrica que trasllada peces des d'un magatzem fins a l'inici de la producció.

A continuació tenim un robot que mou peces, introduint-les i retirant-les dels punts què ens interessin. En un procés productiu podria tractar-se de dos robots en lloc d'un, un per a deixar les peces del magatzem damunt la cinta transportadora que les durà a l'inici del procés productiu i altre per a traslladar les peces des del final de la cinta transportadora fins al començament del procés de mecanitzat.

A més, tenim un procés de mecanitzat que consta d'una línia indexada amb dos estacions de mecanitzat, una fresadora i un trepant.

Es vol aconseguir que aquests tres factors que intervenen (cinta transportadora, robot i procés de mecanitzat) es comuniquen entre ells de forma que s'aconsegueixca tindre un procés continu. Per altra banda, necessitem poder realitzar la supervisió i el control d'aquest procés des d'una interfície gràfica de forma que siguem capaços de governar el procés, veure què està ocorrent, canviar el mode de funcionament, realitzar una parada d'emergència...

## 4.- Objectius del TFG

El principal objectiu d'aquest treball és programar dos PLCs i un robot UR3 i realitzar una comunicació entre ells amb el protocol MODBUS. Per poder obtenir-lo, hem d'anar aconseguint una sèrie d'objectius.

- Programar el PLC que es troba connectat a la cinta transportadora, un Schneider M241, per tal que quan deixem una peça sobre ella la trasllade fins el final de la cinta.
- Programar altre PLC com l'anterior, M241, que en aquest cas estiga connectat a les cintes de Fischer. Amb la programació d'aquest volem que la peça entre al procés de cintes i ixca mecanitzada. S'ha de realitzar un fresat i un trepat a la peça.
- Realitzar un programa per al robot de forma que vaja traslladant peces d'un magatzem, passant-les d'un procés a altre i portant-les fins al magatzem de peces realitzades.
- Realitzar una comunicació MODBUS perquè els autòmats estiguen connectats de forma que siga un procés únic.
- Realitzar un panell de control des d'on poder governar tot el procés (un SCADA). Des d'aquesta volem poder monitorar les cintes i utilitzar controladors per a seleccionar el mode de funcionament i realitzar una parada.

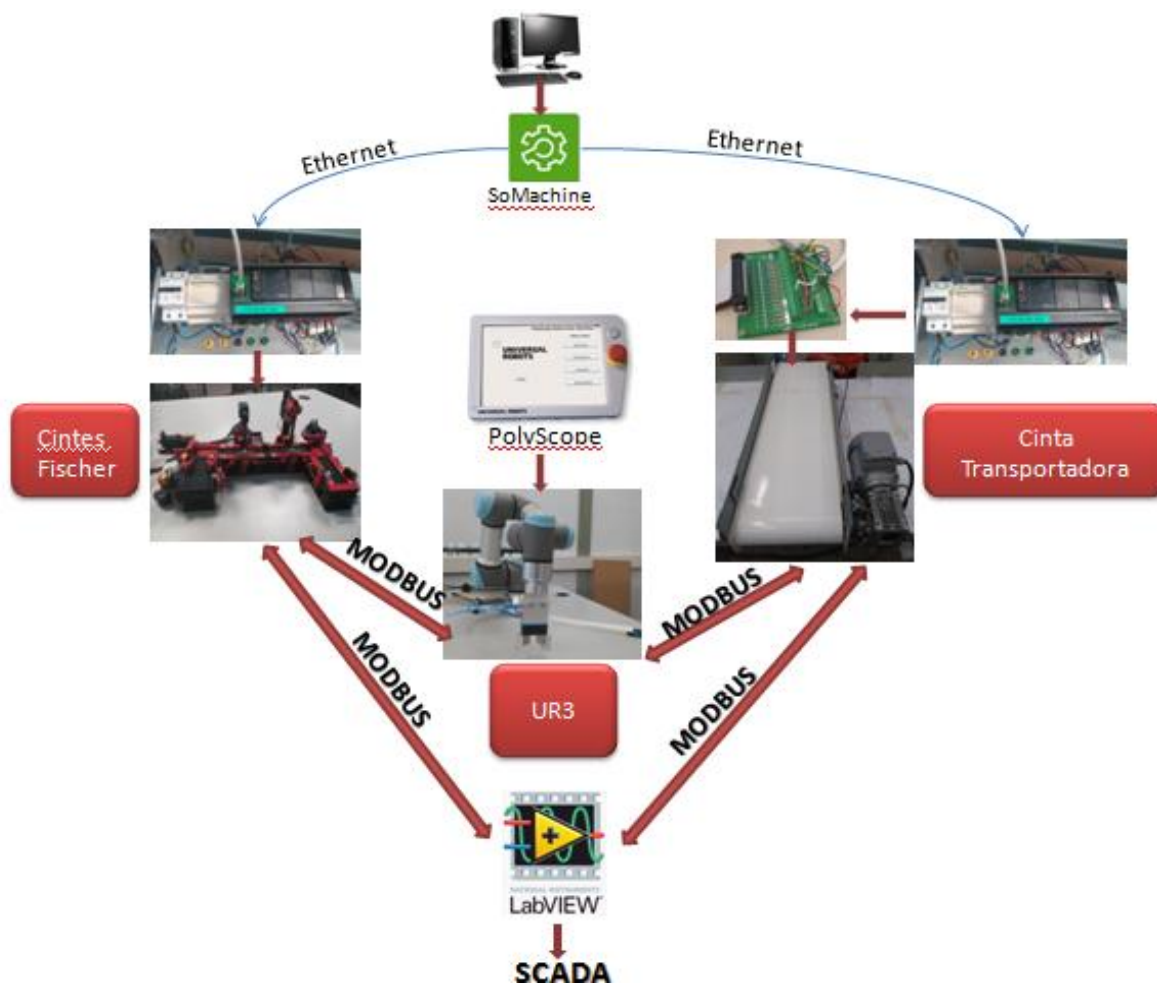


Figura 1. Connexions dels programes i autòmats utilitzats

## 5.- Autòmats utilitzats

En aquest treball hem emprat una cinta transportadora (ABB), una maqueta de Fischer Technik i un robot d'Universal Robots (UR3). Per al control de cadascun d'aquests processos ens fa falta un PLC (Programmable Logic Controller o autòmat programable) o autòmat.

Hem utilitzat l'autòmat de Schneider Electric tant per a les cintes Fischer com per a la cinta transportadora. Es tracta d'un controlador lògic de Modicon, el M241.

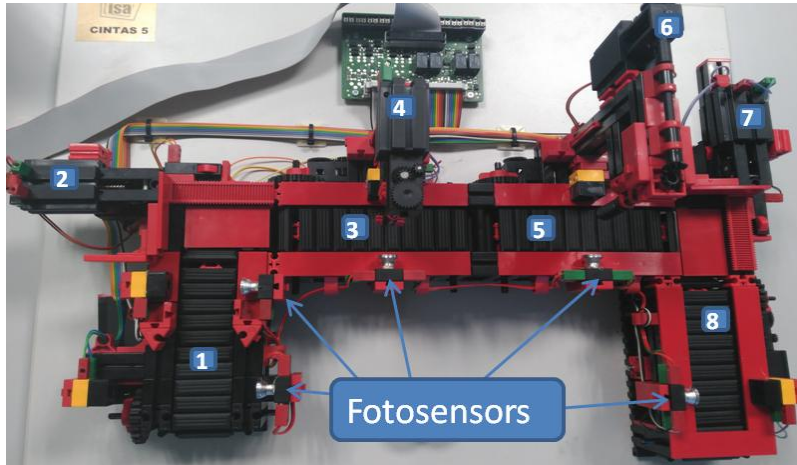
La cinta transportadora (*Figura2*) és una cinta convencional, amb dos sentits de gir i dos sensors, un a cada extrem de la cinta. En el nostre cas, utilitzem un únic sentit de gir del motor i un sensor, el de final de la cinta. Es connecta al PLC mitjançant un circuit que enllaça les entrades i eixides del PLC amb la cinta. La cinta es pot dirigir també manualment des de la pantalla del ABB, des d'on podem nomenar variables, i enllaçar accions dels motors i sensors amb el PLC per tal què des d'aquest sigam capaços de moure adequadament la cinta.(Fem referència a ella amb els següents noms: cinta, cinta transportadora o cinta ABB).



**Figura 2. Cinta ABB**

La maqueta de Fischer (*Figura3*) tracta d'una línia indexada amb dos unitats de mecanitzat (d'ara endavant l'anomenarem cintes Fischer o més abreviadament, cintes). Està formada per quatre cintes, cadascuna d'elles amb un motor independent i que sols permet el gir d'aquestes en un sentit. Dos empenyedors, un a l'inici de la segona cinta i un a l'inici de l'última, aquests tenen un motor que els mou en dos sentits (avant i arrere) i dos finals de carrera. A més, en la segona cinta trobem la primera unitat de mecanitzat, fa funció de fresadora, la fresa gira activada per un motor. I en la tercera cinta, de forma semblant a la fresadora, es disposa d'un trepant.

Hi ha també una sèrie de fotosensors (concretament cinc) de lògica invertida, és a dir, per defecte el seu valor sense detectar cap interferència és 1 i quan alguna cosa es posa al sensor aquestos donen una senyal baixa (un zero). Es disposen dos sensors a la primera cinta, un al començament i altre al final, un sensor a la fresadora i altre al trepant i l'últim sensor en la quarta cinta.



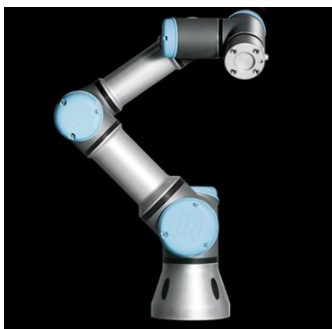
1	Cinta d'alimentació
2	Empenyedor 1
3	Cinta fresadora
4	Fresadora
5	Cinta trepant
6	Trepant
7	Empenyedor 2
8	Cinta d'eixida

**Figura 3. Motors i sensors del procés de cintes Fischer.**

En el cas del robot UR3, aquest inclou el seu propi controlador específic per a aquest robot i que per a programar-lo hem d'utilitzar una pantalla tàctil amb la seua interfase de programació, el PolyScope. Des d'aquesta interfase podem fer moltes coses diverses: configurar la seguretat del robot, definir i/o configurar les entrades i eixides del robot, realitzar la programació d'aquest, moure'l, enviar senyals a altres màquines mitjançant el MODBUS...

El robot es tracta d'un robot d'Universal Robots, l'UR3 (*Figura4*). És d'un braç robòtic articulat que permet moviments amb molta precisió i què, a més, conté una pinça que permet elevar peces de pes reduït (fins 3kg). Té un tamany menut, de forma que ocupa poc d'espai i per tant es pot utilitzar en un nombre major de processos. El seu radi de gir és de 500mm.

Es tracta d'un robot col·laboratiu, per tant hem de tindre en compte algunes consideracions. Quan xoca amb alguna cosa, es deté automàticament i entra en parada, en aquest moment, hem d'habilitar-lo de nou perquè continue amb el programa. Per tant, abans d'habilitar-lo cal assegurar-nos que hem eliminat allò que ha activat la parada per evitar que pare de nou. En cas de polsar el botó de parada d'emergència (la seta d'emergència que es troba a la pantalla tàctil del robot, PolyScope) es posen els frens, i el robot para immediatament. Per desbloquejar-lo, cal despolsar el botó de parada i inicialitzar de nou el robot des de la pantalla d'inicialització, el programa comença de nou com si fos la primera vegada que l'executem. Si el robot es troba en parada, amb tots els frens posats, encara que li donem ordres des de l'exterior, no realitza cap acció, perquè a més, fins que no l'inicialitzem, es trobarà apagat.



**Figura 4. Robot UR3 (www.universal-robots.com)**



## 6.- Robot

El robot que ha sigut utilitzat és el UR3, un robot col·laboratiu molt flexible amb una precisió molt menuda per adaptar-se fàcilment al treball en les zones de producció d'espai molt reduït.

Serveix per automatitzar feines de fins 3 kg de càrrega i a més, amb fàcil programació i configuració.

Es capaç d'adaptar-se a qualsevol indústria i a pràcticament qualsevol feina, i ens permet l'optimització de processos al llarg de la línia de producció.

### 6.1.Ferramenta

El robot es programa per a moure una ferramenta que es situa al final del seu braç. Amb ella es realitza l'operació desitjada.

Al treball, s'ha utilitzat com a ferramenta una pinça. Amb aquesta hem aconseguit agafar peces i traslladar-les fins el punt que ens interessava.

Aquesta pinça pot ser substituïda per qualsevol altra ferramenta segons l'operació que vulguem realitzar. Podem utilitzar un trepant per a fer un forat, una fresadora... i fins i tot una càmera.

El robot té uns límits establerts en quan el Punt Central de la Ferramenta (*PCH, Punto Central de la Herramienta*), es possible la configuració d'aquestos limitant els moviments del robot.

### 6.2.Moviment

#### 6.2.1 Graus de llibertat

El robot consta de sis graus de llibertat, pot pujar i baixar, avançar i retrocedir i girar.

A més, està format per sis articulacions (base, muscle, colze i tres manetes) que poden girar 360°. En el cas de la base, el gir permès és major.

El moviment es troba restringit pels límits de junta. Aquestos límits fan referència a les articulacions. Els podem definir des de la configuració de seguretat. Quan el robot arriba a un d'aquestos límits es produeix una parada de protecció. En el cas de trobar-nos movent el robot amb moviment lliure, si ens aproximem a un límit de junta, notarem una força per avisar-nos.

La programació del moviment del robot i les formes de moure'l es troben explicades al *Manual del programador, Robot*.

#### 6.2.2 Espai de treball

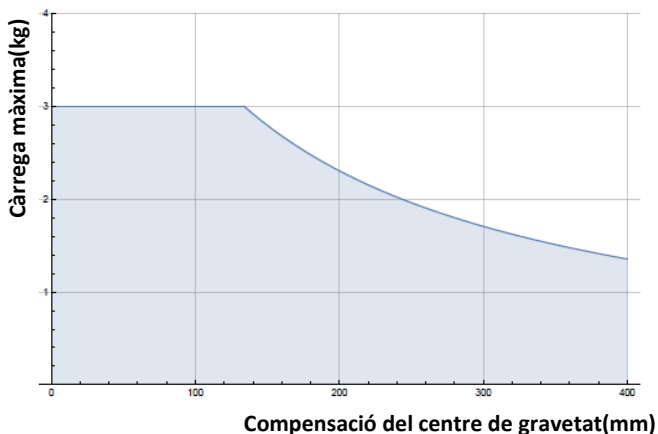
El robot es mou dins d'un radi de 500mm. L'espai què ocupa és menut i serveix per a realitzar feines que es troben pròximes a ell. Per seguretat, quan el robot es pose en funcionament, hem d'allunyar-nos a mig metre d'aquest per evitar que ensopegue amb nosaltres.

A més, hem de tenir en compte un volum cilíndric tant a la part de dalt com la de baix de la base. L'hem de considerar a l'hora de realitzar la instal·lació i la programació per evitar que la ferramenta del robot es trobe en aquest volum. Quan la ferramenta s'aproxima a aquest, les juntes es mouen de manera accelerada, més ràpid que la ferramenta, i la seguretat deixa de funcionar de forma adequada, per tant, podria ocórrer alguna fallada .

### 6.2.3 Càrrega màxima

El UR3 pot automatitzar feines de fins 3kg. En realitat, la càrrega màxima permesa depèn de la compensació del centre de gravetat. Aquesta compensació és la distància que hi ha entre el centre de gravetat i el centre de la ferrament. A major distància (major compensació del centre de gravetat) menor és la càrrega màxima que pot suportar el robot. Observem aquesta relació a la *Figura 5*.

En el nostre cas, a la configuració de la instal·lació hem establert una càrrega de 0.3 kg, que per a alçar la peça que utilitzem al procés és més que suficient.



**Figura 5.** Relació entre la càrrega màxima i la compensació del centre de gravetat (*UniversalRobots-manual d'usuari*)

### 6.2.4 Velocitat

Al robot tenim l'opció de modificar la velocitat a la que volem que es moga. D'aquesta forma tenim l'opció de posar una velocitat més baixa quan estem començant a programar o fent proves, per evitar que en el cas de xoc es produeixca cap problema i augmentar aquesta velocitat quan ja tenim tot el programa ben definit i ens interessa una velocitat més elevada.

Podem configurar els límits de junta. Aquests defineixen la velocitat angular màxima de juntes individuals i limiten la velocitat del braç robòtic.

La velocitat que ha sigut establerta per a la realització del vídeo és del 65% de la velocitat màxima definida.

## 6.3.Parada

Quan es realitza una parada, el robot deixa d'executar el programa. Al robot hi ha dos opcions de parada: parada de protecció i parada d'emergència.

### 6.3.1 Parada de protecció

La parada de protecció s'activa quan el robot es troba prop d'un límit de junta, la velocitat és pròxima a la velocitat màxima de límit de junta o quan el robot xoca amb alguna cosa.

Es tracta d'una parada que funciona com si pausarem el programa. En realitzar-se aquesta parada ens apareix un avís indicant-nos que s'ha efectuat aquest tipus de parada i indicant-nos que per a continuar hem d'habilitar el robot.

Abans d'habilitar el robot perquè continue amb el seu programa, hem d'assegurar-nos que hem eliminat allò que li ha fet al robot activar aquesta parada, del contrari, es tornarà a activar de nou.

És una parada prou freqüent, l'alimentació no s'apaga i el robot roman engegat.

### **6.3.2 Parada d'emergència**

La parada d'emergència s'activa en polsar la seta d'emergència, es tracta d'un botó (una seta) roig que es troba a la pantalla del robot, al PolyScope.

Quan aquesta s'activa el robot es para immediatament, detenint el seu programa per complet. És a dir, en aquest cas el programa es para, de forma que quan es torne a posar el programa en funcionament començarà pel principi en compte de continuar per on s'havia quedat.

Quan polsem aquest botó, ens apareix un avís a la pantalla indicant-nos que s'ha produït aquest tipus de parada i que hem d'anar a la pantalla d'inicialització per poder desactivar-la.

El primer que cal fer per tornar a posar el robot en funcionament és desactivar la seta. El robot s'ha parat, és necessari engegar-lo i alliberar els frens que s'han posat en polsar la parada d'emergència.

Aquesta parada té un ús poc freqüent, sols s'utilitza en cas d'emergència, no ocorre en tots els cicles.

## **6.4. Configuració**

Al robot tenim una finestra de configuració on podem configurar les variables d'instal·lació, la seguretat, la comunicació MODBUS...

El més important és la configuració de seguretat on es defineix la càrrega màxima, els límits de junta, la velocitat màxima de límit de junta, a més de les entrades i eixides de seguretat com les parades de seguretat i emergència.

## **6.5. Programació**

La programació del robot és la part més important. Amb aquesta ens encarreguem d'aconseguir que compleixca amb el funcionament indicat per a la realització de feines concretes. Li senyalem com volem que es desplaci, quines accions volem que faci...

La forma de programar el robot, el llenguatge que s'utilitza... es troba explicat a l'apartat *PolyScope*.

## 7.- Programes utilitzats

Per a la realització d'aquest treball hem fet ús d'una sèrie de programes tant per a la programació dels PLCs com per a la comunicació i control d'aquests.

Per a programar els PLCs hem utilitzat el *SoMachine*, ja que aquests ho requereixen. Hem emprat com a PLCs tant per a la cinta transportadora com per a les cintes el M241 de Schneider (com hem esmentat abans) i per a la programació d'aquests és necessari fer ús del *SoMachine*.

Per a la programació del robot, aquest porta inclosa al seu paquet de compra una pantalla tàctil que serveix d'interfície per a la programació del robot, s'anomena *PolyScope*.

A més, per a poder realitzar una visió i control del procés, hem usat el *LabView*. Amb aquest, hem creat un SCADA, que permet la representació del procés i el control, mitjançant l'ús d'indicadors i controls (a l'apartat *LabView* s'explica amb més detall).

### 7.1. SoMachine

El *SoMachine* és un software que serveix per a l'automatització de màquines. Es connecta al PLC mitjançant un cable Ethernet i indicant la direcció IP d'aquest. Aquest programa ens permet tant l'execució del programa realitzat (en línia o amb descàrrega) com la simulació, amb l'avantatge de poder comprovar que les funcions que hem introduït i tot allò que hem programat té cap error sense necessitat d'enviar-ho al PLC (per aquells casos que no tenim el PLC al davant en eixe moment).

El primer que hem de fer quan creem un programa nou, és incloure el tipus de controlador que anem a utilitzar i començar a programar obrint la configuració del controlador seleccionat. Aquest programa permet la creació de diversos POU de forma que tots s'activen al mateix temps.

El programa consta de tres finestres: arbre de ferramentes, arbre de dispositius i arbre d'aplicacions. El més important és l'arbre de dispositius, en aquest es configura el controlador: indiquem la seua direcció IP, afegim línies de sèrie, configurem l'Ethernet... A l'arbre d'aplicacions és on creem els distints POU i realitzem la programació d'aquests. A l'arbre de ferramentes es troben les distintes biblioteques.

En quant al llenguatge de programació, té sis llenguatges distints: graficet (*SFC, Sequential Function Chart*), diagrama de contactes (*LD, Ladder Contact*), diagrama de funcions (*FBD, Function Block Diagrams*), text estructurat (*ST, Structured Text*), llistat d'instruccions (*IL, Instructions List*) i CFC (*Continuous Function Chart*).

Tenim un apartat de declaració de variables globals (són aquelles que s'utilitzen en qualsevol part del programa), GVL, ací podem crear variables de qualsevol tipus (booleana, paraula, vectors...) i a més, assignar-los una direcció. Declarem les distintes eixides i entrades. Les eixides tenen el prefix **Q**, les entrades **I** i els registres **M**. Seguidament indiquem el tipus de variable, **X** si es booleana, **W** si és una paraula, **D** si és una paraula doble (*Double Word*)... Per exemple **%QX0.1**.

A banda, tenim també les variables internes. Són les que fan referència a un POU concret i, per tant, únicament es poden utilitzar a aquest. En cas d'utilitzar-les fora del POU on han sigut declarades no són reconegudes i dona error de variables no declarades.

Dins de la programació es poden crear blocs de funcions que poden ser temporitzadors, convertidors, comptadors... S'expliquen els utilitzats a l'apartat *Mòduls de funció*.

Podem utilitzar com a condicions: etapes d'un SFC (si es troba l'etapa activa, *Step.x* o per a condicions temporals, *Step.t...*), el compliment d'un bloc de funció (si s'ha realitzat, *BlocFuncio.Q* o *BlocFuncio.Done* segons la funció)...

Una vegada creat tot el programa cal compilar-lo i en cas de tindre el PLC a l'abast (no estar en simulació) connectar-nos a aquest i enviar-li el programa, i ja a continuació posar-ho en marxa i veure que funciona correctament. Una vegada ja connectats tenim varies opcions en línia com per exemple compilar, reset calent (serveix per fer que comence des de l'inici, com si fóra la primera vegada que s'executa), reset fred...

Quan l'hem posat en marxa ens permet veure com avança el funcionament, les variables que es troben actives i el valor d'aquestes. També l'etapa a la què es troben, el pas del temps als temporitzadors, l'execució dels blocs de funcions. Podem parar el programa, llevar la pausa...

És un programa amb opcions molt amplies, que a més de programar el PLC també ens permet la comunicació d'aquest, podem crear mòduls de funció específics per a una connexió MODBUS servidor/client, crear un línia de sèrie on indiquem qui farà de mestre i qui d'esclau en una connexió MODBUS...

## 7.2.PolyScope

El *PolyScope* és la interfície del robot on realitzem la seua programació (pantalla tàctil, *Figura 6*).

El primer que s'ha de fer per poder posar en marxa el robot és engegar-lo. Quan engeguem la pantalla ens apareix una finestra d'inicialització, on cal dir-li què volem iniciar el robot i després engegar-lo. Una vegada el robot en marxa, es passa a la seua programació.

Consta de cinc finestres: programa, instal·lació, moure, entrades/eixides i registres.

El més important és dins la finestra d'**instal·lació** la configuració de seguretat. Ací s'indica la càrrega màxima del robot, els límits de juntes, la velocitat màxima, en quin moment para el robot... També en aquesta finestra tenim altres opcions com variables, que és on es creen les variables d'instal·lació, són aquelles que el programa guarda el seu valor encara que parem el programa i el tornem a encendre i no es poden inicialitzar. A més, trobem el client MODBUS, on s'indica a qui volem connectar-nos i què volem compartir. La finestra d'instal·lació serveix per a la configuració del robot.

En l'apartat d'**entrades i eixides** apareixeran totes aquelles que es troben al programa i des d'ací podem observar i canviar el seu valor.

A la finestra de moviment (**moure**) ix representat el robot en la seua posició actual. Es veuen els eixos coordinats. Baix de la imatge del robot, trobem les opcions de moviment de les distintes articulacions del braç robòtic (base, muscle, colze i les tres manetes), podem polsar les fletxes de gir de les articulacions segons si volem girar en un sentit o altre dins dels límits de junta. A l'esquerra de la pantalla hi ha una sèrie de fletxes (*Figura 16*), tant el gir com el desplaçament fan referència a la ferramenta (és la part del robot que treballa, en el nostre cas una pinça). A més, en aquesta mateixa

finestra està l'opció de moviment lliure, un botó que mentre es troba polsat desactiva els frens del robot i ens permet moure'l amb llibertat.

En la finestra de **programació** és on indiquem allò que volem que faça el robot. L'estructura del programa consta d'un conjunt de comandaments. El llenguatge de programació és similar a l'utilitzat en informàtica (el C) o matemàtiques (el Matlab), ja que utilitza condicions (*if... else...*), bucles (*while*)...

Dins d'aquesta finestra tenim quatre subfinestres: comandament, gràfics, estructura i variables.

En la subfinestra de **comandaments** indiquem què volem que realitzi en un punt concret de l'estructura del programa, com per exemple escriure una funció, definir un punt de pas...

Els **gràfics** serveixen per a veure la situació en què es troba el robot.

A la finestra de **variables**, es troben les d'instal·lació amb el seu valor.

A l'**estructura**, que serveix per editar el programa, afegim els comandaments i, a més, tenim l'opció de moure'ls, retallar, copiar i pegar. Els tipus de comandaments que podem afegir són tres: bàsics, avançats i assistents. Entre els comandaments més utilitzats trobem el moviment i el punt de pas, l'ajust de variables, l'assignació, el bucle, les condicions(*if...*), i a més podem crear també subprogrames, esdeveniments, advertències, comentaris, avisos...

Està també l'opció d'inicialitzar variables, és a dir, definir un valor inicial predeterminat per a aquestes, de forma que cada vegada que el programa s'inicia de zero, les variables prenen aquest valor. La inicialització es fa al començament del programa i únicament es poden inicialitzar les variables creades dins d'aquest(variables internes).

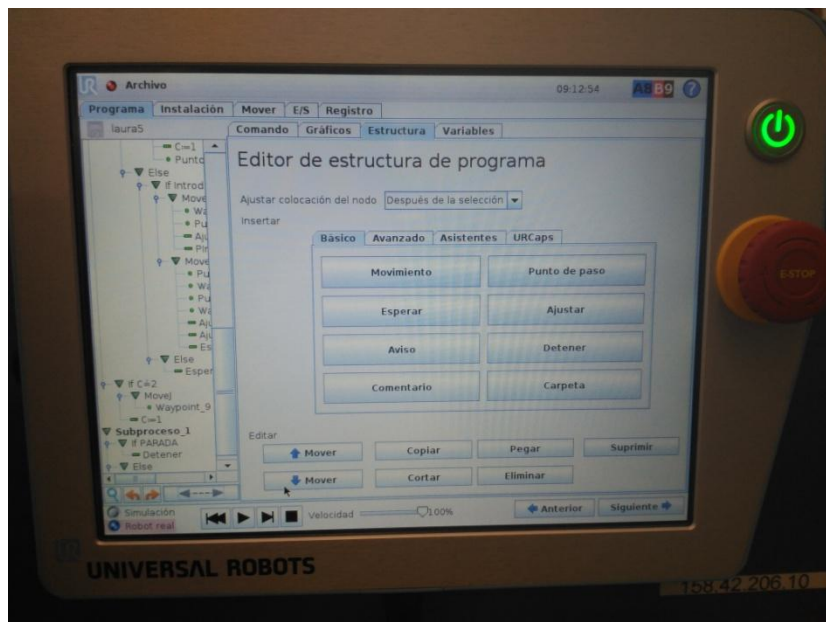


Figura 6. PolyScope. Pantalla tàctil de programació del robot.

## 8.- Funcionament

### 8.1.Funcionament general

El funcionament que volem aconseguir en aquest projecte és l'automatització de dos processos, la cinta transportadora i la cinta de Fischer i, a més, unir-los mitjançant un braç robòtic que s'encarrega de traslladar les peces entre les posicions que no es troben comunicades per cap procés.

El funcionament general tracta d'una peça en una posició inicial preestablerta, podria ser un magatzem on es troben les peces "noves" esperant a ser recollides. El braç robòtic la recollirà i la durà fins el primer procés, la cinta transportadora. Quan la peça es deixada sobre la cinta, aquesta es posa en marxa. El robot una vegada deixa la peça, es mou fins una posició de repòs. El motor de la cinta transportadora es mourà fins que la peça siga detectada pel sensor situat al final de la cinta. En aquest moment, el robot arreplega la peça i la trasllada fins el següent procés, les cintes Fischer.

Quan el robot situa la peça a l'inici d'aquestes cintes, el sensor d'inici de cinta detecta la peça i el motor de la primera cinta es mou. Mentrestant, el robot torna a la posició de repòs. La peça recorre tot el procés d'aquestes cintes i quan arriba al final, el robot va a recollir-la i la deixa en una posició final. Aquesta posició final podria ser altre magatzem diferent a l'inicial on les peces ja mecanitzades s'emmagatzemen.

El funcionament descrit es duria a terme en aquest ordre si únicament volem mecanitzar una peça. En cas d'haver més d'una, el robot aniria introduint peces a la cinta transportadora, sense parar-se en la posició de repòs. Caldria definir preferències per saber què cal que faça el robot si es troba davant de dos possibles situacions al mateix temps, com podrien ser: recollir peça de la cinta transportadora i recollir peça del final de les cintes de Fischer. Ens interessa en primer lloc recollir les peces que ja estan mecanitzades i en últim introduir-ne de noves. Per tant donarem preferència en aquest ordre: Recollir les peces finalitzades -Recollir peces de la cinta transportadora -Introduir peces noves.

### 8.2.Modes de funcionament

Per a variar el procés segons les nostres preferències en cada moment, definim diferents modes de funcionament, com són l'automàtic, pas a pas, manual, parada d'emergència i botó d'emergència del robot.

#### 8.2.1 Pas a pas

Aquest mode, al contrari que l'automàtic, és el que s'utilitza per a fer proves, per a la primera posada en marxa, per fer revisions, per quan hi ha algun problema saber on es troba...

En aquest mode, el funcionament és pas a pas, hem d'anar indicant que ja es pot passar a la següent etapa, d'aquesta forma si trobem algun problema, evitem fer malbé tot el procés, produir una parada d'emergència o el trencament d'alguna màquina perquè evitem que altra peça comence el procés fins assegurar-nos que tot funciona correctament. Ens trobarem en aquest mode quan no estiga activa la senyal de mode automàtic, *NOT (AUTO)*. A més, per a indicar-li que pot realitzar la següent operació (per canviar d'etapa) cal prémer el polsador *BotoManual*. Tenim dos polsadors, un per a cada procés, de forma que si ens trobem en aquest mode i polsem s'active aquesta senyal únicament al procés que ens interessa.

En el cas de la cinta transportadora, el funcionament no varia massa, en compte de ser el robot qui avisa a la cinta què ha deixat una peça sobre ella, ha de ser l'usuari qui polsant un botó iniciï el procés de la cinta (*BotoManual*). Després la cinta farà el mateix que en mode automàtic, portar la peça fins al final, quan és detectada per un sensor i tornaria a l'etapa inicial.

En les cintes de Fischer, cada vegada que vulguem que faci el següent moviment hem d'indicar polsant un botó (*BotoManual*). Pel demés tot funciona de la mateixa forma, fins que no estiga buida una zona, no es pot introduir altra peça nova. El graficet de l'última zona es reinicia quan la peça arriba al final, ara no s'ha d'esperar a què el robot retire la peça. Ara hi ha major nombre de condicions, abans cada vegada que acabava una operació podia passar a la següent mentre que ara si no polsem el botó no realitza cap operació. Per exemple, si la peça arriba a la fresadora, estarà esperant fins que polsem per indicar-li que pot començar aquesta operació.

El robot en aquest mode es trobarà parat, ja que volem provar els moviments de les cintes pas a pas per revisar que tot funciona correctament. Per veure què el robot funciona com toca, hem d'anar fixant-nos al mateix temps que el programem, des del seu PolyScope, ja què de l'altra forma ens resultaria més complicat. A més, no te massa sentit perquè per veure si el robot va a xocar o per dir-li que realitzi un moviment concret és millor fer-ho tenint el robot davant, i per açò la forma de fer-ho és amb la seua pròpia ferramenta de programació.

### **8.2.2 Automàtic**

Quan seleccionem el mode automàtic (*AUTO*), el funcionament serà com el descrit a l'apartat anterior (*Funcionament general*) i amb les preferències indicades. En aquest mode hem de seleccionar el número de peces que ens interessa realitzar de forma que el robot introduïxca a la cinta transportadora tantes peces com haja indicat l'usuari. A més, també tenim l'opció de refrescar el nombre de peces realitzades.

Tant les cintes de Fischer, com la cinta transportadora com el robot rebran aquesta senyal i es posaran a funcionar de forma adequada.

El mode automàtic ens interessa utilitzar-lo quan ja ens hem assegurat que el funcionament és el correcte.

### **8.2.3 Manual**

Aquest mode serveix per a realitzar els moviments des del panell de control. Ens serveix per a moure la cinta transportadora (activant el motor de la cinta) i les cintes de Fischer (amb la possibilitat de moure cadascun dels motors d'aquesta). Tindrem un polsador per a cadascun dels botons, mentre estiga polsat el botó adequat i no arribe al seu final de carrera, el motor corresponent es trobarà en marxa. Per evitar ocasionar cap problema, hem posat també les condicions de fi de carrera perquè així si polsem més temps del què deuriem, deixi de funcionar el motor.



## 8.2.4 Parada d'emergència

Quan alguna cosa no funciona bé, ja siga per perill d'un operari, d'una màquina... l'usuari ha de tindre l'opció de parar tot el procés polsant un botó que es trobe al seu abast (al panell de control). Quan s'activa la parada d'emergència tot deixa de funcionar. La variable que activa aquesta parada és *BotoParada*.

Es congelen tots els SFC i en desaparèixer aquesta senyal el procés déu començar de nou de zero. En cas d'estar en mode automàtic, el nombre de peces realitzades tornarà a posar-se a zero.

Per aconseguir que el procés es detinga i comence des del principi quan desaparega aquesta senyal, hem d'activar l'opció de *SFCInit* a tots els SFC que volem detenir. Després d'activar aquesta opció caldrà posar una acció de forma què quan es done la parada d'emergència, és a dir, en polsar *BotoParada*, s'activen els *SFCInit* de tots els SFC. Amb açò, aquests SFC detenen les accions que estaven fent relacionades amb alguna etapa, reinicialitzen el SFC (comencen a l'etapa inicial) però de moment es troba congelat a les etapes inicials. En desaparèixer la senyal d'emergència (deixem de polsar *BotoParada*), les *SFCInit* ja no es trobaran actives, es descongela el SFC. S'iniciarà des de les etapes inicials amb normalitat.

En cas d'haver una parada d'emergència cal retirar la peça manualment. Tenim dos opcions, o que siga retirada per un operari que es desplace fins el procés i l'elimine o amb la pantalla de moviment manual, com sabrem on es troba la peça, podem desplaçar-la per les cintes fins que arribe al final d'una d'aquestes, ja siga de la cinta transportadora o del procés de cintes i ahí puga ja ser retirada per l'operari. Dependrà de si resulta un perill posar algun motor en marxa per retirar-la mitjançant la pantalla de moviment, de si posem en perill l'operari anant ell mateix a retirar-la, o en el cas de no haver cap dels dos problemes, decidirem el què més còmode ens resulte.

## 8.2.5 Botó emergència del robot

Com hem comentat anteriorment, el robot disposa d'un botó d'emergència que permet la seua detenció i la del seu programa. Si es polsa aquest botó és degut a què hi ha alguna cosa al robot que no està funcionant com cal, però únicament al robot i no al procés en general, ja que en aquest cas caldria polsar el botó de detenció del procés (explicat a *l'apartat anterior*). Així, tant la cinta transportadora com l'altre procés de cintes poden continuar funcionant, condicionats per la no participació del robot.

La cinta transportadora està prou limitada ja què l'única cosa que pot fer és traslladar la peça què ha deixat el robot fins el final de la cinta quan la peça es detectada pel sensor final. La cinta detectarà que el robot està pausat degut a una emergència si passa un minut sense que el robot haja deixat cap peça nova. En aquest moment activa una variable que fa referència a l'emergència (*Emergencia*) i es trobarà detesa fins que aquesta senyal desaparega.

Les cintes de Fischer deuen permetre que les peces que es troben dins d'aquest procés arriben a la seua fi. Es detectarà que el robot es troba en situació d'emergència si han passat tres-cents segons (cinc minuts) sense que cap peça haja sigut introduïda a l'inici de les cintes o cent vint segons sense que cap peça haja sigut recollida al final d'aquestes. En aquest cas s'activa una variable d'emergència

(*Emergencia*) que canvia les condicions del procés. Ara ens interessa que totes les peces arriben a la fi de les cintes, encara que ja hi haja una peça al final esperant a ser recollida. Per tant, la condició de l'última zona que implica que si hi ha una peça ja en aquesta zona no puga haver cap altra, serà eliminada (afegim en la transició corresponent la variable d'emergència per tal que quan aquesta es dóna, la transició es compleixca).

### 8.2.6 Alarmes

A més, per saber si ha hagut qualsevol problema al procés de les cintes de Fischer, posem una sèrie d'alarmes. Aquestes alarmes sols funcionen en mode automàtic, ja que és quan ens interessa que ens avisen quan alguna cosa ha deixat de funcionar correctament perquè no ens trobem davant del procés. Quan ens trobem al mode manual no són necessàries perquè aquest mode serveix per a revisions, és a dir, estem pendents del procés mentre aquest està en marxa i si ocorre qualsevol avaria ho detectem immediatament, sense necessitat de cap alarma.

Únicament les posem a les cintes Fischer. A la cinta transportadora no ens calen perquè com el procés és curt, ens serveix la mateixa senyal que utilitzem quan hi ha emergència al robot (*Emergencia*).

Posem sis alarmes, de forma similar a com ho hem fet en el cas d'haver una emergència al robot.

La primera alarma, s'activarà quan es produeixca qualsevol problema a la primera cinta. Si passen deu segons des que la peça ha sigut detectada pel sensor inicial de la primera cinta i la peça encara no ha entrat a la segona zona de cintes (el SFC *Cintes23* es troba en l'etapa inicial), s'activa la senyal **Alarma1**.

Les tres següents alarmes fan referència a la segona zona de les cintes. La segona alarma s'activa quan passen cinc segons des que l'empenyedador es posa en marxa, en cas que no avanci d'etapa, és a dir, si està cinc segons movent-se cap avant sense trobar el sensor de final de carrera davanter (cinc segons activada l'etapa Step0 del SFC *Cintes23*). Si s'activa aquesta alarma, sabem que ha hagut un problema amb l'empenyedador mentre es movia cap avant, pot ser que se'n haja eixit de les guies, que el motor no funcione adequadament... La variable que s'activa fent referència a aquesta alarma és **Alarma2**.

La tercera alarma serveix per al tram des de l'empenyedador fins la fresadora. Per a poder saber quan s'activa, posem cada vegada que passa per l'etapa de la fresadora, la variable  $F=1$ , de forma que si es troba a una etapa concreta (Step1 SFC *Cintes23*) i  $F$  és zero, s'activa **Alarma3**. Necessitarem també afegir a l'etapa inicial del SFC *Cintes23* una acció que pose  $F=0$ , per canviar el valor de  $F$  de 0 a 1 i al contrari, utilitzem un MOVE. Aquesta alarma ens serveix per saber si ha hagut cap problema des de l'empenyedador fins a la fresa, poden haver passat dos coses: l'empenyedador s'enganxa i no es mou adequadament (el més probable) o la cinta no funciona.

La quarta alarma ens indica mal funcionament d'una peça que es troba entre la fresadora i el trepant, com en aquesta zona hi ha canvi entre dos cintes, podria quedar-se la peça enganxada. Per detectar-ho, deixem passar sis segons des que s'activa l'etapa de moviment simultani de les dos cintes (Step3 SFC *Cintes23*), si quan ha transcorregut el temps, encara es troben a la mateixa etapa (no han sigut detectades pel sensor del trepant), s'activa la senyal **Alarma4**.

La quinta alarma funciona com la segona, ja què comença en la tercera zona, a l'empenyedor i es per controlar si ha hagut algun problema en aquest mentre avançava. Si avança durant cinc segons sense trobar el final de carrera davanter (es trobe cinc segons a l'etapa corresponent, Step0 del SFC *Cinta4*), s'activa la senyal d'alarma, **Alarma5**.

La sexta alarma serveix pel mateix que la tercera, és a dir, per a l'empenyedor dos i la cinta d'eixida. Detecta fallada en el retrocés del segon empenyedor o bé, en el motor de la cinta d'eixida. Si passen cinc segons (Step1 del SFC *Cinta4*) des què l'empenyedor ha començat a retrocedir i la peça no ha arribat al sensor final, es detecta un problema i s'activa l'**Alarma6**.

## 9.- Programació

La part més important del treball és la programació de cadascun dels PLCs utilitzats de forma que tot funcione com desitgem. Com que tenim tres PLCs hem de fer un programa distint per a cadascun d'ells. Aquests programes tindran relació entre ells per tal d'aconseguir comunicar-los.

La cinta transportadora i les cintes de Fischer estan governades pel mateix PLC (M241), la programació d'aquest controlador es du a terme amb el software SoMachine. Aquest programa té llenguatges de programació molt diversos LD, CFC, SFC, FBD... Permet la creació de POU's i la utilització de molts diversos mòduls de funció: temporitzadors, convertidors de tipus de variable, de lectura, escriptura... Es poden realitzar tots els POU's desitjats, que poden funcionar al mateix temps o amb l'ordre que més ens interessa.

En aquest treball utilitzem principalment dos llenguatges de programació: el SFC (*Sequential Function Chart*), consten d'un conjunt d'etapes (amb una etapa inicial), transicions al final de cada etapa (condicions que quan es compleixen es poden travessar, és a dir, es passa d'etapa), ramificacions per poder tindre dos camins segons es compleixca una transició o altra, salts que fan que tornem a l'etapa que indiquem, a més hi ha altres coses que no hem utilitzat com macro etapes... i el LD (*Ladder Contact*) per a la definició de les transicions, accions, activació de variables...

L'UR3, com hem esmentat abans, inclou el seu programador, una pantalla tàctil amb la qual es programa, PolyScope. La programació és molt distinta al SoMachine. Aquesta interfície té molts comandaments distints: bucles (*while*), condicions (*if*), moviments, ajustos, inicialització de variables... Per a moure el robot, cal indicar-li els punts de pas per on volem que vaja el robot, no serveix dir-li que pugui o baixi si no que hem d'establir un punt o diversos concrets. També des de la configuració de la instal·lació permet canviar la seguretat del robot, les variables d'instal·lació, definir la comunicació de MODBUS client... A més des de la pantalla tàctil podem veure com es mou el robot i tenim un botó d'emergència per si necessitem parar-lo en qualsevol moment.

### 9.1.Cinta Transportadora

El programa d'aquesta consta d'un SFC principal, anomenat **CintaABB**.

Aquest comença en una etapa d'espera, on la cinta estarà parada, fins que si ens trobem en mode automàtic el robot deixi una peça sobre aquesta o en cas d'estar en mode manual, polsem el botó manual perquè comence. En este moment, la cinta es mou (sempre en la mateixa direcció), fins que la peça és detectada per un sensor que es troba situat al final de la cinta. La cinta es trobarà en aquesta etapa fins que la peça deixi de ser detectada pel sensor, moment en què la cinta comença de nou el procés explicat.

Com què volem que el robot pugui deixar més d'una peça, necessitem guardar la informació què ens indica que el robot l'ha deixada. Pot ser què en aquest moment el SFC es trobe en una etapa que no correspon a la transició de si el robot ha deixat peça o no, degut a què hi ha altra peça anterior avançant o esperant a la cinta. Amb aquesta informació, pot passar la transició quan es reinicia el SFC.

Per aconseguir-ho, hem creat un SFC, anomenat **DeixarPRobot**, què activa una etapa quan el robot deixa una peça (aquesta etapa fa de condició al SFC principal per saber si hi ha peça a la cinta) i es desactiva al flanc de pujada del Step0, SFC *Cinta ABB* (després de complir-se la condició que indica que hi ha peça). Fins que el robot no deixi una nova peça es troba en una etapa d'espera.

Aquest procés es du a terme mentre no hi haja cap senyal d'emergència activada. En el cas què el robot entre en parada, s'activa la senyal *Emergencia* (explicat a l'apartat *Botó emergència del robot*)

## 9.2.Cintes Fischer Technik

El programa de les cintes queda dividit en tres zones (tres SFC), la Cinta1 que fa referència únicament a la cinta d'entrada, les Cintes23 que consta del primer empenyedor, de la segona cinta on es troba la fresadora i de la tercera on es troba el trepant, i l'última zona, Cinta3 de la qual formen part el segon empenyedor i l'última cinta. A més, hi ha també com al cas de la cinta transportadora un SFC per al cas que el robot entre en parada d'emergència (**PEmergencia**) i altre SFC per a la recollida de peces del final d'aquest procés (**PFI**).

Hem seccionat el programa en tres perquè ens interessa que pugui haver més d'una peça a la vegada dins del procés i que a més hi haja una única en cadascuna de les zones. En cas d'haver més d'una peça, la segona peça s'espera a l'inici de la zona, fins que la primera ix d'aquesta. A l'inici de les zones dos i tres els anomenem primera i segona àrea de descans.

A continuació, descrivim el mode automàtic. Hem de tenir en compte què el mode pas a pas es com l'automàtic però afegint etapes d'espera perquè sigam nosaltres polsant un botó qui decidim que la peça pot continuar fins el següent pas (etapa), és a dir, començar a realitzar la següent acció.

### 9.2.1 Cinta 1

El SFC comença en una etapa inicial d'espera, que estarà activa durant dos segons, aquesta espera és necessària perquè els sensors de les cintes tarden uns segons en posar-se en funcionament, del contrari, la cinta començaria a moure's independentment que la peça haguera sigut detectada o no.

El moviment de la primera zona comença quan el primer sensor detecta que hi ha peça, el motor de la cinta s'engega fins després de tres segons que la peça haja passat pel sensor que hi ha al final d'aquesta. El motiu pel què deixem tres segons més funcionant la cinta, després que la peça haja sigut detectada pel sensor final, és per assegurar-nos que la peça ix de la cinta i arriba a la següent zona.

Al finalitzar el SFC la peça es troba en la primera àrea de descans, a l'entrada de la zona dos (*Cintes23*). En aquest moment, una nova peça ja pot començar aquest mateix procés.

### 9.2.2 Cintes23

El segon SFC també consta d'una etapa inicial d'espera. Aquesta zona es trobarà parada fins que s'activa la variable que ens indica que la peça ja es troba en la primera àrea de descans. La variable que hem utilitzat és **FSInit1** i el seu valor serà 1 en el flanc de baixada (negatiu) de l'última etapa del SFC *Cinta1* (Step1).

Una vegada ja es posa en marxa la segona zona, l'empenyedor mou la peça cap a la següent cinta (fins que toca el seu final de carrera), el motor de la cinta s'engega (i al mateix temps l'empenyedor

retrocedeix fins l'altre final de carrera, es queda en la posició inicial) fins que el sensor de la fresadora detecta la peça. La fresa treballa durant tres segons, en acabar el motor de la cinta es torna a engegar i amb aquest també el de la tercera cinta. La peça es transportada fins el trepant (detectada per un fotosensor que hi ha instal·lat en aquest), el trepant treballa també durant tres segons. Quan aquest finalitza, únicament la tercera cinta es posa en moviment fins tres segons després que el sensor del final de la cinta detecta la peça.

### 9.2.3 Cinta4

D'igual forma que en els dos SFC anteriors, el de l'última cinta conté també una etapa inicial d'espera. Ens trobarem en aquesta etapa fins que la peça arriba a la segona àrea de descans (sempre que no hi haja cap altra peça en esta zona, en eixe cas deuríem esperar també a que la peça fóra primer retirada). Sabem que la peça ha arribat a la zona indicada, igual que en el cas del segon SFC, quan estiga activa la variable **FSInit2**. Açò ocorre en el flanc negatiu (o de baixada) de l'última etapa del SFC anterior (*Cintes23*), Step5.

Quan ja s'ha activat la variable corresponent, comença el moviment de l'última zona. El segon empenyedor trasllada la peça des de la segona àrea de descans fins l'última cinta (fins que arriba al seu final de carrera), s'engega la cinta i l'empenyedor ara es mou en sentit contrari (cap arrere) fins que toca el seu posterior final de carrera, la cinta està en marxa fins dos segons després que l'últim foto sensor detecta la peça.

La peça déu ser retirada pel robot. Fins que no es porta a terme aquesta acció, una altra peça no podrà avançar per aquesta zona. Tenim un SFC per saber si hi ha peça al final i el robot ha d'anar a recollir-la.

### 9.2.4 PFi

En aquest SFC tenim una etapa inicial d'espera. Es troba activa fins que la peça arriba al final de l'última zona de cintes (Step4, *Cinta4*), en aquest moment es travessa la primera transició i hi ha una nova etapa què activa la variable **FIP** què permet que el robot vaja a arreplegar la peça. Aquesta etapa es desactiva al flanc de pujada de l'inici del SFC (*Init*).

### 9.2.5 PEmergencia

Hem creat un SFC per a canviar la forma d'actuar en cas que el robot entre en parada d'emergència. L'etapa inicial es troba activa mentre no hi haja cap parada al robot. En el moment què detecte que el robot ha parat (mitjançant les condicions definides a l'apartat de botó d'emergència del robot, dins l'aparat de programació), es passa a altra etapa on s'activa una variable anomenada **Emergencia**. Esta etapa es troba activa mentre no es desactiva la parada del robot. Com que el robot no pot enviar cap senyal perquè no es troba en funcionament, en aquelles transicions on es troba una condició que implica el robot, estarà també la condició d'**Emergencia** (una condició de OR), de forma que en cas què el robot no estiga en funcionament, es passa d'etapa directament (el funcionament està també descrit a l'apartat indicat).

### 9.3.Robot

El robot utilitza la interfàç PolyScope, que és propietat d'Universal Robot, per a programar-lo.

El programa del robot comença amb una inicialització de variables, de forma que cada vegada que el robot para el seu programa, ja siga per una parada d'emergència o perquè el detenim voluntàriament, les variables que es troben dins del comandament d'inicialització de variables es posaran al valor d'inicialització que hem predeterminat. Del contrari, hi ha una sèrie de variables, anomenades d'instal·lació que no es poden utilitzar en aquest comandament ja que conserven sempre l'últim valor que ha guardat el programa.

Després d'inicialitzar les variables, tenim un **BeforeStart**. Ací s'indica el què volem que faça la primera vegada que s'executa el programa. Per evitar que el robot detecte un bucle infinit si encara no s'ha engegat la cinta (en cas d'engegar primer el robot), posem una etapa d'espera, de forma que el robot espere per començar el programa fins que la cinta ABB comence el seu funcionament. La variable que hem utilitzat per dir-li que la cinta es troba en marxa es **INICI**.

A continuació comença el programa del robot. Primerament, hem ajustat la variable **DeixarP=0**, per assegurar-nos que la cinta no tinga aquesta variable activada i comence a moure's sense que el robot haja deixat cap peça. A més, tenim una espera d'un segon per evitar problemes (podríem disminuir aquest temps si ens interessara).

El robot té dos opcions, depenent del valor que tinga la variable **c**. Aquesta variable conserva el seu valor encara que hi haja cap emergència o parem el programa (variable d'instal·lació). L'utilitzem per saber si el robot s'ha quedat en una zona amb perill de xoc (**c=2**), en aquest cas hem de portar-la a un punt superior abans d'iniciar el funcionament normal. Es troba en un punt de perill a l'inici de les cintes de Fischer, perquè és un punt prou baixet i amb la cinta transportadora prop. Tenim dos condicions, **c=1** comença el programa i **c=2** puja el robot a un punt segur i canvia el valor de **c** a **c=1**.

Quan **c=1**, primer indiquem que les peces realitzades són zero, igual que hem fet amb **DeixarP**, per si de cas. Tot seguit, entra en un bucle sempre i quan ens trobem en mode automàtic i no es trobe la parada d'emergència polsada. Dins d'aquest, tenim diverses condicions per indicar-li al robot què ha de fer en cada cas, establint preferències.

El funcionament general del robot és recollir una peça d'una posició inicial on es troba aquesta (com podria ser un magatzem de peces d'estoc), traslladar-la fins la cinta, arreplegar-la d'aquesta quan arriba al seu fi, portar-la a les cintes de Fischer i quan acaba tot el procés anar a per ella i dipositar-la en la seua posició final (magatzem de peces realitzades). El robot duria aquest ordre si únicament volem mecanitzar una peça, en el cas contrari el robot no deuria estar parat, si no anar introduint peces segons les preferències establertes. Donem preferència a recollir les peces realitzades (final de les cintes de Fischer), després arreplegar-les de la cinta transportadora i en últim lloc la introducció de peces.

Quan el robot va a arreplegar una peça del magatzem inicial per a transportar-la a la cinta ABB, en el moment que l'agafa incrementem la variable **PIntroduides** en una unitat. Anirà a arreplegar peces sempre i quan el nombre de peces que ha introduït siga menor que el nombre de peces que volem realitzar. Açò ho controlem amb una variable, anomenada **Introduir**, i amb una condició if, de forma que quan la variable es trobe activa, el robot introdueixca peça.

D'igual forma, quan recull una peça del final de les cintes, el nombre de peces finalitzades s'incrementa en una unitat, la variable que utilitzem per a dur aquest compte és **PFinalitzades**.

A més, perquè deixi la peça o la reculli, cal obrir o tancar la pinça, cal què en la posició adequada ajustem la variable **PINZA** a nivell baix quan volem obrir i a nivell alt quan volem tancar.

Hem posat una espera per al cas què no estiga activada la senyal d'arreplegar peça de la cinta transportadora ni del final de les cintes, també per evitar que entre en bucle infinit. L'espera es troba activa fins que s'activa una de les variables què li indica al robot que ha d'anar a recollir peça (**RobotArreplega o FiP**) o si Introduir es troba activa. Serveix per al cas que ja ha introduït el nombre de peces necessàries i les peces es troben dins d'algun procés, també per si li indiquem que volem realitzar una peça més, en aquest cas es tornaria a activar la variable Introduir.

A més, tenim un subprograma. El subprograma funciona de forma independent al programa i es troba en funcionament al mateix temps. L'hem utilitzat per al cas què polsem la parada d'emergència des del Labview es detinga el programa del robot. Li hem indicat que es trobe en espera fins que detecte la senyal **PARADA**, i en aquest moment detinga el programa.

Quan el robot deixa la peça a la cinta transportadora, déu enviar una senyal a la cinta perquè es posei en marxa. Per a la comunicació MODBUS amb altres PLCs ens fa falta escriure durant el programa als registres d'eixida el valor què volem enviar. Per avisar què el robot ha deixat una peça o l'ha recollit, cal què assignem a les variables de MODBUS corresponents un valor de 1.

D'igual forma, el robot espera rebre una senyal de les cintes Fischer avisant-lo que ja pot anar a arreplegar la peça. Necessitem comunicar el robot amb els altres dos autòmats perquè tot el procés es trobe enllaçat.

Per al moviment del robot, hem d'indicar-li els punts de pas concrets pels que ha de recórrer la seua trajectòria. Cal comprovar amb molt de deteniment com passa d'una posició a altra evitant que xoqui amb res (en aquest cas es produiria una parada de seguretat). Cada vegada que el programa comença hem de moure el robot fins al punt indicat com a punt inicial del programa (polsant un botó fins què el robot es troba en la posició adequada, moment en què pot començar el seu programa, determinem una posició inicial concreta prou elevada per evitar xocs, *Waypoint\_10*), per açò és important saber en quina posició s'ha quedat el robot. Si este es troba en un punt de perill de xoc i es dirigeix cap al punt inicial, haurem d'indicar-li un punt de pas abans per a evitar que es produísca cap parada (i d'ací la utilització de la variable c).

El robot té un botó d'emergència, en el moment en que la polsem, el robot es deté, el PolyScope ens du a la pantalla d'inicialització on després de llevar l'emergència haurem d'engegar de nou el robot i ja podem tornar a començar el programa.



## 10.- Comunicació

La definició de comunicar és intercanviar informació entre dos o més participants a través d'un canal. Comunicar autòmats és el mateix, els participants són els PLCs i el canal depèn del tipus de comunicació.

La comunicació d'autòmats consisteix en connectar-los de forma que puguin intercanviar dades. L'intercanvi d'informació entre ells és necessària per fer que treballen conjuntament. A més de l'intercanvi de dades i informació, la comunicació també ens serveix per al control del procés.

En aquest treball ens cal una comunicació entre autòmats ja que necessitem que tres processos (la cinta transportadora, les cintes i el robot) es coordinen i obtindran així un procés conjunt continu. També comunicarem els PLCs amb un SCADA per poder controlar tot el procés.

Entre els distints tipus de comunicació (servidor, exploració d'entrades/eixides, model publicador/subscriptor...), hem triat la connexió via MODBUS.

### 10.1. Comunicació MODBUS

El MODBUS és un protocol de comunicació que permet connectar diversos PLCs. Dos tipus de comunicació:

#### 10.1.2 Mestre/Esclau

En aquest tipus de comunicació es tracta d'un protocol de sol·licitud-resposta. Ha d'haver un PLC que faça de mestre i un o diversos que seran els esclaus. El mestre és qui envia la sol·licitud als esclaus del que volem fer, ja siga llegir, escriure o ambdues coses i els esclaus (o l'esclau a qui se li envia la petició) han de respondre. El mestre deu ser l'element de control principal (HMI o SCADA).



Figura 7. Comunicació mestre-esclau

#### 10.1.3 Servidor/Client

Consta d'un servidor i diversos clients. També pot haver un client que accedeixca a més d'un servidor al mateix temps. Els clients són qui accedeixen al servidor (indicant la direcció IP d'aquell a qui volen accedir) i realitzen una petició i els servidors contesten els requeriments dels clients. Els clients inicien la comunicació i els servidors esperen a que els clients es comuniquen en ells per respondre'ls. La comunicació es realitza via Ethernet (a través d'una xarxa).

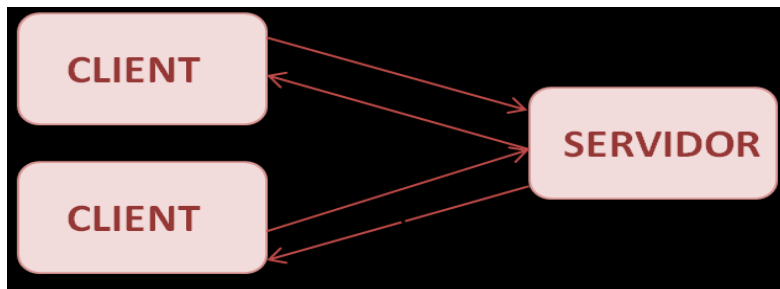


Figura 8. Comunicació servidor-client

Al nostre treball comuniquem una cinta transportadora amb un robot mitjançant MODBUS.

La cinta està governada per un controlador lògic Modicon M241 (de Schneider). Per programar-lo hem utilitzat el programa SoMachine, específic per a aquest tipus d'autòmat. Després d'estudiar el programa, hem observat que la millor i més senzilla forma de comunicar-la amb altre autòmat amb el MODBUS es amb una connexió servidor-client. En aquest tipus de connexió, el client accedeix al servidor d'on llig i escriu allò que li resulta d'interès.

Per començar, hem fet un senzill programa i hem comunicat aquest autòmat amb altre igual per veure el funcionament del MODBUS abans de passar al robot.

En l'intercanvi de dades de tipus servidor-client entre autòmats fem diverses funcions com són l'escriptura i lectura, a més de la conversió de l'adreça IP. Aquestes funcions s'usen al client, qui accedeix amb elles al servidor. Les funcions es relacionen amb una etapa del procés. Quan s'arriba a l'etapa corresponent, la funció s'activa i si es realitza correctament (si tant el servidor com el client es troben actius i no es produeix cap error), s'obté una senyal que indica que s'ha realitzat l'acció (DONE=1) i, a més, per exemple en el cas de la lectura, les dades llegides del servidor poden ser emmagatzemades.

Per altra banda, tenim el robot, es tracta d'un robot d'Universal Robots, l'UR3. En estudiar la forma de connexió MODBUS del robot, ens hem adonat que aquest farà de client, i serà per tant qui llegeixca i escriu del i al servidor (en aquest cas, la cinta).

Amb el robot no hem utilitzat les funcions esmentades abans (escriptura, lectura i conversió d'adreces). En aquest cas la lectura i escriptura són més directes. En la opció de MODBUS client, dins d'instal·lació, cal indicar l'adreça IP de qui farà de servidor i afegir les distintes entrades i eixides. Per a llegir hem de triar si volem fer la lectura d'una entrada o una eixida digital i indicar el número corresponent (per exemple en el cas de voler llegir l'eixida %QX0.1 indiquem que és l'eixida 1). Per a escriure, hem de seleccionar entrada o eixida de registre. Aquestes envien valors de 16 bits.

Durant el programa de robot hem utilitzat les distintes entrades i eixides obertes amb el MODBUS. D'aquesta forma les usem de condicions per tal que en el cas que el robot lligi de la cinta una cosa que ens interessa en l'estat corresponent (per exemple, en alt) faci una acció determinada. També mitjançant el programa hem escrit el valor que volíem enviar a la cinta amb una eixida de registre.

Les eixides de registre perquè la cinta pugui llegir-les bit a bit, cosa que ens interessa més perquè podem comparar un bit sabent si es compleix o no, hem utilitzat al programa de la cinta una funció que converteix una paraula d'un nombre X de bits amb X bits dels que podem fer ús per separat.

#### 10.1.4 Comunicació MODBUS Servidor/Client entre autòmats iguals o semblants

Per poder comunicar dos PLCs semblants, per exemple dos com els que hem utilitzat per fer funcionar la cinta (M241), utilitzem aquest tipus de comunicació. Cal triar quin farà la funció de servidor i quin la de client, tenint en compte que és el client qui accedeix al servidor i llig i escriu sobre aquest.

Al servidor hem de programar-lo de forma normal com si no tinguera a cap client connectat a ell, no hem d'indicar-li que algú es connectarà a ell ni quines variables són per a compartir i quines no. L'única cosa que hem de tenir en compte és reservar una sèrie de variables sobre les que escriurà el client. Indiquem a la sèrie Ethernet el seu propi valor de IP.

El client es qui es connecta al servidor i qui accedirà a variables del servidor per llegir-les o escriure sobre elles, per tant, a aquest si que li hem d'indicar a qui es connecta i què ha de fer. Primer, hem d'indicar de mateixa forma que ho fem al servidor, la direcció IP en la sèrie Ethernet. Després, per a poder realitzar les funcions de lectura i escriptura, fem uns blocs de funció específics per a aquestes accions ( **READ\_VAR** i **WRITE\_VAR**). A més, en aquestos blocs ens demana l'adreça del PLC i com la tenim en IP hem de convertir-la de forma que aquestes funcions puguem entendre-la, per això utilitzem un bloc de conversió d'adreça (**ADDM**). La utilització d'estos blocs es troba definida a l'apartat *Mòduls de funció*.

Per a utilitzar aquests blocs, els hem d'associar a l'etapa que ens interessa del SFC, és a dir, a l'etapa que volem que es faci una d'aquestes funcions. Quan l'etapa corresponent s'activa, serà quan comença a executar-se el bloc de funció.

Cal indicar el tipus d'objecte que volem llegir o escriure, si es tracta d'una entrada o eixida digital o d'un registre, la quantitat que anem a llegir i al vector què volem emmagatzemar-ho.

### 10.1.5 Comunicació MODBUS Servidor/Client (Cinta/Robot i Cintes/Robot)

En la comunicació que utilitzem al nostre procés(Servidor/Client), el robot fa de client, és a dir, és qui accedeix als autòmats, tant a la cinta transportadora com a les cintes de Fischer per obtenir les variables necessàries d'aquests i escriure'ls les que ens interessin. El motiu pel què el robot és el client és perquè, a més que en el tipus de comunicació MODBUS la seua interfície únicament li permet ser client, és qui fa de nexa entre els dos processos de cintes, necessita comunicar-se en els dos processos per tant la millor opció és que siga ell qui accedeixi als altres PLCs.

Des del robot tenim l'opció de llegir entrades i eixides digitals, únicament indicant la direcció IP del PLC i el número de l'entrada/eixida a qui volem accedir. Per exemple, per accedir a la %QX0.0, li haurem de posar un zero en eixida digital. Aquestos valors esta llegint-los tot el temps, els utilitzem al programa a les condicions. Per exemple, quan una de les eixides es posa a u, la condició es compleix i el robot realitza una acció determinada.

Per poder escriure, hem d'enviar les dades per registre. Per açò cal indicar-li igual que abans la direcció IP del PLC a qui li volem escriure i el número del registre sobre el què anem a escriure. Per escriure cal fer-ho des del programa del robot, hem d'indicar quin registre volem ajustar i a quin valor. En eixe moment s'escriu sobre el registre corresponent del PLC, aquest valor queda fixat fins que el canviem amb altra acció.

Per part del servidor, tant de les cintes Fischer com la cinta transportadora, no cal indicar res per permetre que el robot es connecte, és el client qui es connecta al servidor i no al contrari. En el cas què el robot estiga llegint variables, tant entrades com eixides, no hem de fer cap cosa perquè pugui accedir a aquestes, únicament tindre-les definides perquè no ens done cap error.

En el cas d'estar escrivint, ens interessa que ens active una variable, però com que escriu un registre sencer hem de fer la conversió. El registre escrit pel robot és emmagatzemat a un %MD, és un registre intern doble (D:Double Word). Convertim de DWord a bit (**DWORD\_AS\_BIT**), d'esta forma podem per exemple enviar un 1, i aquest activa una variable booleana que ens pot servir com a condició ens les transicions.

#### 10.1.5.1- Cinta/Robot

El robot ha d'avisar que ha deixat una peça a l'inici de les cintes i ha de saber quan anar a arreplegar-la del final d'aquesta.

En la cinta transportadora, el robot està llegint la variable **RobotArreplega** (%QX0.0), que s'activa quan la peça ha sigut detectada pel sensor de final de cinta (etapa Step1). El robot té una condició a seu programa què li indica que una vegada aquesta senyal s'activa, ha d'anar a recollir la peça, sempre que no estiga activada la senyal de final de peça de les cintes, ja què aquesta té preferència, com hem indicat a l'apartat de programació.

Necessitem també que el robot escriba quan ha deixat la peça a l'inici de la cinta, de forma què després què el robot obriga la pinça en la situació corresponent a l'inici de la cinta, en el seu programa hi ha una acció que ajusta el registre corresponent de la cinta a 1. En aquest moment se li enviarà un 1 al PLC de la cinta pel registre indicat (%MD0). Hem declarat una variable que fa referència a aquest registre, **DeixarP** i serà aquesta la que convertim a bits. Per poder passar-ho a bits hem creat un vector de bits anomenat **Rob**, d'on únicament utilitzarem el primer bit, **Rob[0]**.

**Rob[0]** ho utilitzem com a transició, quan aquesta senyal estiga activa (a 1) la cinta comença a moure's.

Quan el robot arreplega la peça del final de la cinta, no ha d'escriure res ja que tenim el sensor de final de cinta, per tant, deixa de detectar peça que seria el mateix que dir-li que la peça ha sigut recollida pel robot.

#### *10.1.5.2- Cintes/Robot*

El robot ha de saber quan anar a recollir la peça al final de tot el procés de les cintes i ha d'avisar que s'ha emportat la peça.

Per saber quan recollir la peça, llegirà l'eixida digital %QX0.1 del PLC de les cintes. Estarà llegint tot el temps aquesta senyal, i en el programa del robot tindrem una condició on l'utilitzem. Aquesta condició serà que en cas d'estar la variable activada, el robot ha d'anar a arreplegar la peça, a més, aquesta condició té preferència sobre totes les demés (com hem explicat a l'apartat de programació). Perquè la variable estiga activa, caldrà que la peça es trobe al final de l'última cinta, en l'última etapa del SFC (Step4, *Cinta4*).

Una vegada el robot ha recollit la peça ha d'avisar a les cintes que ho ha fet, al contrari que en la cinta transportadora, aquí no hi ha cap sensor final que pugui saber que la peça ha sigut retirada. Per tal d'avisar a les cintes, el robot ha d'enviar un registre al PLC de les cintes. Tenim una variable anomenada **LRobot** que emmagatzema el que el robot li envia al registre %MD0. Després, d'igual forma que a la cinta, es necessita un mòdul de conversió de MWord a bits, de forma que pugem extraure el bit que ens interessa. En aquest cas el vector de booleans que hem creat per a seleccionar els bits que volem s'anomena **Robot**, utilitzarem un únic bit, **Robot[0]**, perquè sols volem saber si la peça ha sigut recollida o no. Amb el **Robot[0]** formem una transició de forma que quan es compleixca, pugui entrar altra peça a l'última cinta del procés de cintes.

## 11.- LabView

Una vegada queda tot el procés i la comunicació d'aquest definida, ens cal ajuntar tot en un, de forma que pugam controlar tots els processos des d'un mateix panell. Per açò, utilitzem el LabView, un programa que permet el disseny mitjançant la utilització d'un llenguatge molt gràfic i visual.

Per a representar les entrades i eixides dels diferents autòmats i realitzar un panell des què l'usuari (podria ser un operari en una línia de producció) pugui canviar el valor de les entrades, reinicialitzar el procés, realitzar una parada d'emergència... hem utilitzat el LabView.

La característica principal de LabView es que podem obtenir variables de diversos autòmats diferents i treballar amb elles. Això és exactament el què ens interessa, tenir al mateix lloc les variables dels tres PLCs utilitzats al procés. Les variables a les què accedim pot ser tant per a llegir-les com per a escriure algun valor sobre elles. Aquest accés pot realitzar-se de distints modes, pujant les variables a un servidor i accedir a aquest per obtenir-les (ja siga un servidor OPC, el mateix servidor del LabView...), mitjançant MODBUS...

### 11.1.MODBUS

En el nostre cas, hem preferit una comunicació MODBUS ja què tota la comunicació dels PLCs l'hem feta d'aquesta forma, per seguir amb el mateix procediment. En aquest cas la comunicació ha sigut mestre/esclau, com hem explicat abans (apartat *Comunicació*), aquest protocol serveix per al control dels PLCs. El LabView fa de mestre, és el que controla mitjançant el SCADA. El mestre accedeix a l'esclau fent la petició adequada (lectura, escriptura o lectura i escriptura) i l'esclau respon a aquesta petició.

El primer que cal fer per connectar amb els PLCs i poder tindre les variables que ens interessa és crear un servidor d'entrades i eixides (*I/O Server*). Com volem què el LabView faça de mestre, seleccionarem l'opció de MODBUS (no la de MODBUS Slave). Triem l'opció *serial Ethernet* i introduïm la direcció IP d'un dels dos PLCs. Únicament amb aquests passos, el LabView ja pot accedir al PLC què li hem indicat. Per a connectar-lo amb l'altre PLC creem altre servidor i seguim els mateixos passos introduint en aquest cas la direcció IP què li correspon al nou PLC.

### 11.2.Variables

Quan ja tenim creat el servidor i el MODBUS per a cadascun dels autòmats, ens fa falta introduir les variables què volem utilitzar de cada PLC. Per a aquest pas, hem de tindre en compte el significat dels prefixos de les variables i la numeració per saber com estan relacionades amb les variables que hem emprat als autòmats.

Amb el prefix 0 són bobines, eixides digitals, amb l'1 tenim entrades digitals, el 3 correspon a registres d'entrada i el 4 a registres de retenció. El 0 i el 4 són tant de lectura com d'escriptura, mentre que l'1 i el 3 són únicament de lectura. L'accés amb el cas de les entrades i eixides digitals és de bits. Tenint en compte que van des del bit 0 fins al 7, és a dir, podem representar paraules únicament de huit bits, de la QW0, podem utilitzar des de la QX0.0 fins la QX0.7. Com la QX0.0 és la primera, segons com es troben definides les eixides digitals, correspon a la 00001. Per passar a la QX1.0, com sabem que hi ha huit bits de cada paraula, li hem de sumar huit, de forma que la numeració corresponent és la 00009. En el cas dels registres (tant d'entrada com de retenció) tenim

l'opció d'accés bit a bit, en aquest cas la numeració és la que té punt i el nombre de bit al que accedim, o accés a paraula. En el cas de bit a bit podem accedir a setze bits per a cada registre, per exemple la MW0 aniria des de la 40001.1 (MX0.0) fins la 40001.16 (MX0.15). Quan accedim amb paraules, l'única diferència és que li llevem el punt bit, MW0:40001.

En el nostre programa, hem utilitzat les eixides digitals per a llegir-les (podríem haver llegit també entrades digitals, però com fan referència a sensor no ens interessa) i els registres de retenció per a escriptura (accedint a la paraula). El motiu d'haver utilitzat els registres de retenció i no les variables d'eixida per a l'escriptura és precisament pel tipus d'accés que ens permet. Amb els registres, al poder accedir a paraules, ens estalviem problemes, ja que l'accés bit a bit pot fallar.

Per afegir les variables al servidor, hem d'indicar que volem crear variables (*bound variables*) i seleccionar-les de dins del MODBUS corresponent, tenint en compte el que acabem d'explicar. Una vegada afegides ja es pot operar amb elles.

El LabView ens permet realitzar operacions molt diverses amb les variables. Crear vectors, descompondre'ls, realitzar operacions lògiques, escriure sobre elles o llegir-les... En el nostre cas, ens interessa principalment la lectura i escriptura d'aquestes.

### **11.3.VI**

Després d'haver afegit el servidor i creat les variables, el següent pas es la creació d'un instrument virtual. Mitjançant aquest aconseguirem que l'usuari pugui controlar tot el procés. L'instrumenta virtual consta de dos pantalles, un diagrama de blocs i un panell frontal.

Al diagrama de blocs hem de posar tot el que volem representar al panell frontal. Hem d'introduir les variables, crear controladors o indicadors, segons ens interessa escriure o llegir... És el panell on es realitza la programació del VI, s'utilitza el llenguatge G. Aquest panell quedarà ocult quan executem el panell de control.

El panell frontal és una representació del que seria un panell de control. Des d'aquest es dirigeix el procés, l'usuari pot veure el que està passant o controlar el funcionament pulsant botons. Podem donar-li forma, de manera que ens queda amb un disseny bonic, visual i pràctic i que l'usuari (operador, per exemple) pugui des d'ací controlar el procés de manera senzilla i entendre els diferents botons i les representacions que apareixen. Està format per botons, leds i formes geomètriques. Tot açò es representa a un SCADA.

El que ens interessa al LabView és poder representar alarmes, per saber si ha hagut algun problema amb la peça i on es troba en aquest moment, o si el robot ha entrat en emergència... Representar també la posició actual de la peça. A més, el més important és poder controlar tot el procés, seleccionant el mode de funcionament, l'opció de parada...

### 11.3.1 Diagrama de blocs

En aquest panell (*Figura9*), com ja hem esmentat, es realitza la programació d'allò que volem que aparega al panell de control. Per començar, cal crear un bucle amb una condició de **Stop**, introduint al seu interior tot la programació. D'aquesta forma, l'SCADA es pot usar mentre no polsem la condició de fi de bucle, el botó de **Stop**. Cal posar el bucle perquè realitzi les operacions contínuament, del contrari, es realitzaria sols una vegada i no obtindríem cap dada.

A més, cal posar també un temps. Per indicar el període amb què es realitza cada cicle i poder saber el nombre de cicles que s'han realitzat.

Una vegada ja tenim el bucle i el temps, podem començar la programació a l'interior d'aquest.

Com hem indicat abans, llegim eixides digitals i escrivim amb registres de retenció. Cal saber què les eixides digitals les llegim com a bits i, per tant, podem directament relacionar-les amb indicadors, perquè la representació d'aquestos utilitzant leds és de bits. Per poder llegir-les, hem d'introduir les variables que ens interessin, arrossegant-les des del MODBUS què hem creat al nostre projecte fins el diagrama de control. Quan ja les tenim al diagrama, per defecte es troben en mode lectura així que no modificar canviar res, sols ens falta afegir un indicador per cadascuna de les variables. Les variables que volem llegir són els motors tant de les cintes de Fischer com de la cinta transportadora (per saber la situació de la peça), l'habilitació dels sensors, les alarmes què ens indiquen algun problema al procés, l'emergència del robot, un indicador per saber si la peça es troba al final de la primera zona (**PCintes23**) i altre per saber si es troba al final del procés(**PFin**).

Per escriure sobre els registres de retenció és un poc distint. En el cas de voler escriure quan polsem un botó, hem de realitzar una conversió. Un polsador treballa a nivell de bits, ja què únicament indica si està actiu o no. La conversió la realitzem perquè no podem passar una informació de nivell de bits directament a un registre. Cal passar de bit a un vector (*bit\_to\_array*) i a continuació de vector a número (*array\_to\_number*). El número, ara si, podem guardar-lo en un registre. En el cas de passar-li sols la informació d'un polsador, el registre contindrà el valor 1. Aquesta operació cal fer-la per a tots els polsadors. Els polsadors què tenim són la parada d'emergència, el mode de funcionament i el polsador manual. Com volem que en polsar aquest arribi la informació tant al procés de les cintes com a la cinta, enllaçarem el polsador amb dos variables de registre, una referida a cada PLC (cada una obtesa del seu corresponent servidor MODBUS). També tenim un polsador per a refrescar el nombre de peces que portem mecanitzades, de forma què en polsar, l'indicador de peces realitzades es posa a zero. Es posarà zero també després que hi haja qualsevol parada d'emergència. Tenim el polsador de retirar l'emergència obtesa des del robot.

Per altra banda, hi ha també controladors i indicadors numèrics. Amb els controladors numèrics tenim l'opció de triar un número des del panell frontal i amb l'indicador, serà el panell qui el represente. L'indicador introduït ens serveix per saber el nombre de peces realitzades, que han arribat al final de procés de mecanitzat i el robot ja les ha traslladat al magatzem de peces acabades. El controlador ens serveix per indicar el nombre de peces què volem realitzar, quan arribem al nombre de peces indicat, el procés pararà. Perquè tinga sentit, sols compta les peces realitzades en mode automàtic, ja que en el mode pas a pas som nosaltres els que tenim el control del procés i tan



sols estem fent proves, per tant, no serveix per a res dir-li què volem un nombre determinat de peces ja que seríem nosaltres els responsables de fer que es complira aquest número.

Finalment, hem indicat que cree un document en pulsar el botó de **Stop** (condició de fi de bucle). Posem un vector fora del bucle, de forma què quan ix del bucle, escriu aquest amb els valors que li indiquem, després en l'escriptura del document, hem seleccionat que traspose el vector, de forma que la informació ens aparega per columnes en lloc de en files, de forma què ens resulta més visual.

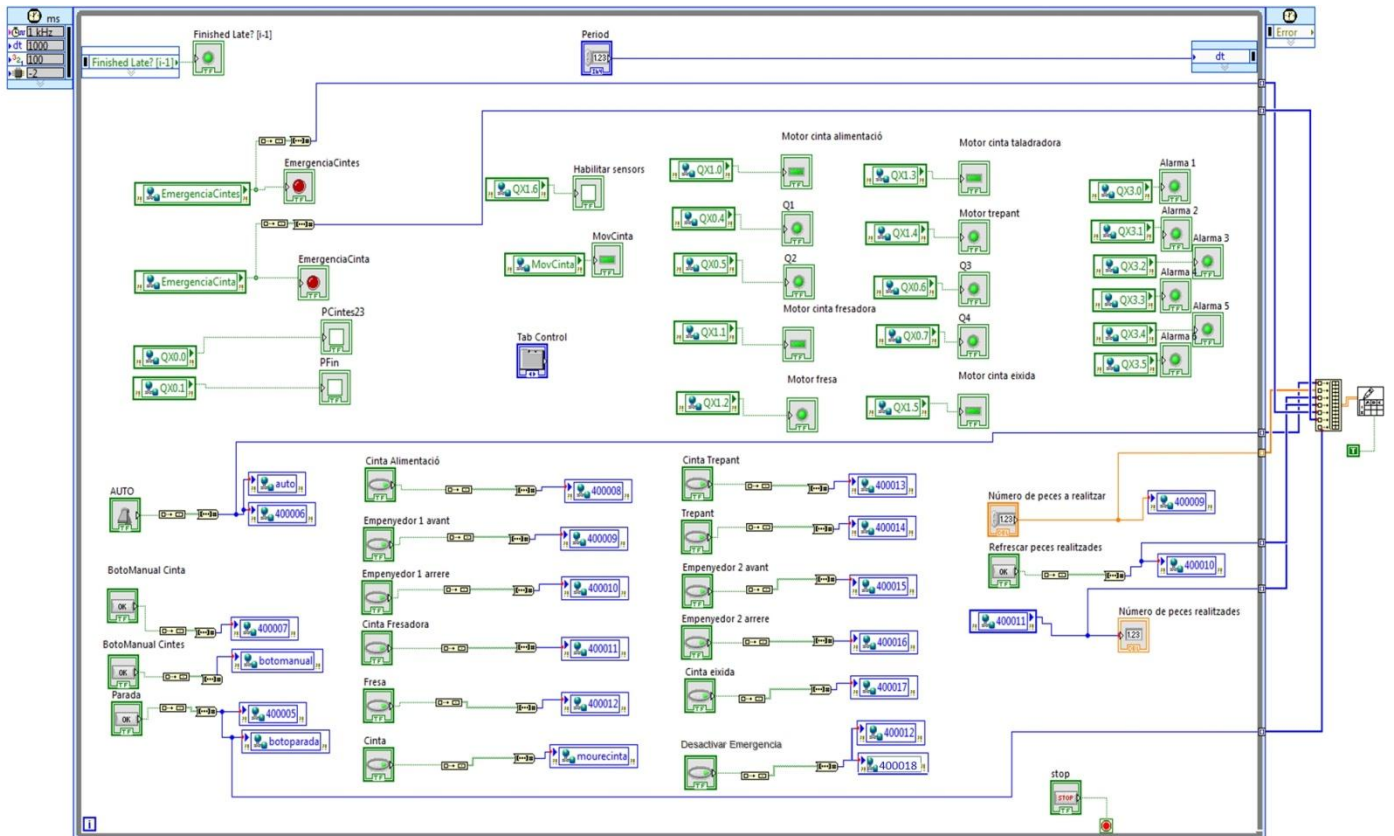


Figura 9. Diagrama de blocs del VI(Instrument Virtual)

### 11.3.2. Manual de l'usuari: SCADA

Des d'aquesta pantalla tenim la capacitat d'obtenir tota la informació que es crea al procés i controlar-la. Volem accedir a la informació necessària per tal de monitorar el procés indicant tant la situació actual de la peça com les alarmes que poden sorgir i, a més, ser capaços de realitzar un control, seleccionant el mode de funcionament desitjat, realitzant una parada d'emergència en cas de ser necessari...

En aquesta pantalla apareixen directament tots els controladors i indicadors introduïts al diagrama de blocs, a partir d'aquests podem donar-li forma a la pantalla modificant la forma o les propietats dels indicadors i controladors, introduir dibuixos geomètrics...

La forma d'utilitzar aquest panell de control és molt senzilla i visual. El nostre SCADA consta d'una pantalla que conté una carpeta amb quatre finestres, de forma que queda més organitzat i classificat de forma que segons el que vulguem fer triem una finestra o altra. A la primera es troba la pantalla de visualització, a la segona el panell de control, a la tercera el moviment manual de la cinta transportadora i a la quarta el moviment manual de les cintes Fischer.

Primer definim tot el que observem (els indicadors), que representa cada led i en quin moment s'il·lumina. Açò queda reflexat a la pantalla de visualització.

#### *11.3.2.1- Pantalla de visualització*

Per començar, hem monitorat totes les eixides corresponents als motors, tant de les cintes, els empenyedors com de la fresadora i el trepant. Quan aquestes estiguen actives s'il·lumina el led corresponent. Per a que la representació siga el més visual possible i s'entenga fàcilment quin led correspon a cada variable, hem volgut crear una representació del procés real, dibuixant les diferents cintes i guardant la mateixa distribució que tenim al procés. Quan, per exemple, el motor de la cinta d'alimentació es troba en marxa, aquesta cinta canviarà de color, indicant que es troba en moviment. D'aquesta manera a part de saber què correspon a cada cosa, podem observar on es troba la peça en cada moment sense fer falta d'anar davant les màquines per veure-ho. En aquest cas, a partir d'un led quadrat, l'hem adaptat a la forma de la cinta. (La cinta canvia de negre a blau quan es troba en funcionament).

Respecte a la senyal d'emergència del robot, tenim dos indicadors, un referent a la senyal que obtenim des del procés de la cinta transportadora i altre per a la que obtenim de les cintes de Fischer. Aquestes senyals s'activen quan han detectat que el robot no està funcionant (en el cas de trobar-se en mode automàtic), és a dir, quan ha passat més del temps establert sense que el robot haja interactuat amb els altres processos, no haja deixat o recollit cap peça. L'**Emergencia1** fa referència a l'obtesa des de la cinta transportadora i l'**Emergencia2** a l'obtesa des de les cintes de Fischer.

Amb l'activació de la variable **PCintes23**, observem que la peça ha arribat al final de la primera zona de les cintes i entra a la segona, comença a ser empenyada pel primer empenyedors. Serveix per saber quan el robot pot introduir una nova peça al procés de les cintes. En cas de trobar-nos en mode manual, com el robot no es troba actuant, aquesta senyal ens indica quan podem introduir una nova peça perquè no es produeixca cap problema.

Amb l'indicador de **PFI**, sabem si la peça ha arribat al final del procés de les cintes i es troba esperant a ser recollida pel robot. Aquesta és la senyal que li indica al robot que vaja a arregar-la.

A més, es troben també representades les distintes alarmes que hem establert al procés. Hi ha sis i els indicadors corresponents (leds) els tenim situats al dibuix de l'SCADA a la zona on correspon cada alarma, de forma que sabent on es troba el problema i a què es degut es puga resoldre amb la major rapidesa possible. Aquests leds canvien de granat quan es troben apagats a groc en cas de produir-se cap alarma.

La primera alarma ens indica un problema a la cinta d'alimentació, la peça no ha arribat al final d'aquesta (**Alarma1**).

**Alarma2**, el motor del primer empenyedador no funciona correctament, no empenta cap avant (no arriba al seu final de carrera).

**Alarma3**, problema amb el motor de l'empenyedador movent-se cap arrere o en la segona cinta (cinta de la fresadora), la peça no arriba a la fresa.

**Alarma4**, la peça es troba entre la fresadora i el trepant, problema en alguna de les cintes corresponents o en el canvi d'una a altra.

**Alarma5**, la peça no ha sigut traslladada a l'última cinta, degut a què el segon empenyedador no es mou correctament cap avant.

**Alarma6**, el motor de l'empenyedador no funciona correctament retrocedint o el motor de la cinta d'eixida no gira, la peça no arriba al final del procés.

Tenim també un indicador per saber quantes peces portem realitzades, número de peces que han sigut emmagatzemades pel robot en finalitzar tot el procés.

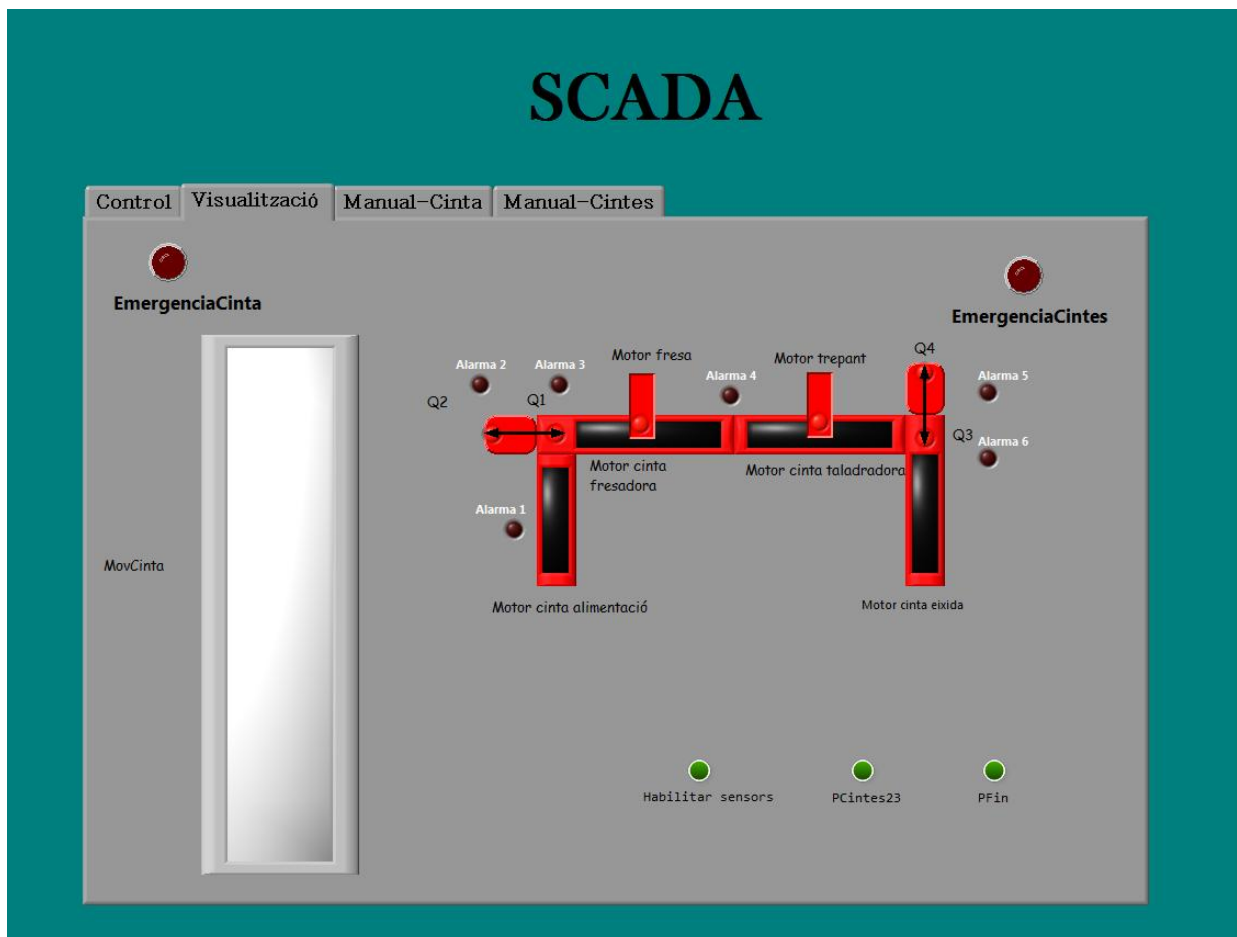


Figura 10. Pantalla de visualització del SCADA

Una vegada definits tots els leds de la pantalla de visualització, passem als distints botons que trobem. Aquests botons es troben en tres pantalles diferents. La primera serveix per a controlar el procés quan triem algun mode de funcionament (*pantalla de control*) i les altres les utilitzem per moure els distints motors forma manual des de la pantalla de (*pantalla de moviment*). Amb els polsadors passa igual què als indicadors, podem canviar-los les propietats. Podem triar si es tracta d'un botó, d'un interruptor... o si es queda pres quan el polsem o sols està activat mentre polsem, entre altres coses.

### 11.3.2.2- Pantalla de control

El primer que hem de seleccionar és el mode de funcionament, tenim un interruptor per a seleccionar-lo. Amb el mode automàtic (interruptor cap amunt, **AUTO**) hem d'indicar-li el nombre de peces que volem realitzar i es posarà a funcionar fins que haja fet tantes peces com li hem indicat. Per indicar-li el nombre de peces que volem, tenim un indicador on podem escriure un número o triar-lo polsant les fletxes cap amunt i cap avall. A més, per al mode automàtic tenim altre botó (**Refrescar peces realitzades**) que serveix per a reinicialitzar el valor de peces realitzades, podem polsar-lo nosaltres quan ens interesse i a més, es posarà a zero de forma automàtica després de realitzar una parada d'emergència.

El mode manual (interruptor cap avall, **NOT(AUTO)**) és un mode pas a pas, tenim un botó que utilitzem quan aquest mode es troba actiu què és el **BotoManual**. Amb aquest, cada vegada que el polsem deixem que el procés passe al pas següent. Per exemple, en les cintes Fischer, per començar la segona zona, fins que no polsem el BotoManual l'empenyedor no es mou cap avant. El **BotoManual** és un polsador, s'activa únicament mentre estem polsant-lo.

**BotoParada**, si ens adonem que alguna cosa no està funcionant com deuria, o hi ha qualsevol problema, polsant aquest botó detindrem immediatament tot el procés. Quan ens trobem al mode automàtic, al desactivar la senyal, el procés que havia sigut detés no continua per on es trobava si no que torna a funcionar des del principi. Si hi ha alguna peça al procés ha de ser retirada manualment per un operari abans de desactivar la senyal d'emergència. En cas d'estar al mode pas a pas, si polsem aquest botó el procés es congela, continuant per on s'ha quedat en desactivar el botó. Es tracta d'un botó que una vegada el polsem es queda actiu fins que polsem de nou.

El botó de **Stop** serveix per a detenir el SCADA, quan el polsem desconnecta les connexions què té amb els diferents PLCs via MODBUS. A més, crea un document, apareix una finestra perquè li indiquem on el volem guardar. En aquest trobarem ordenada per columnes la següent informació: mode de funcionament (un 1 en cas de trobar-se en automàtic), número de peces a realitzar, número de peces realitzades, botó de refrescar (un 1 en cas de trobar-se polsat), emergència del robot detectada a les cintes, emergència del robot detectada a la cinta i finalment la parada d'emergència.

L'emergència detectada del robot a les cintes i a la cinta transportadora ha de donar-se al mateix temps, però com què no la detecten a l'hora pot ser que una o varies línies els valors d'aquestes emergències no coincideixca.

Hi ha també un polsador per a desactivar l'emergència que s'ha detectat del robot als dos processos, en polsar-lo, es desactiva la condició d'emergència, i els graficets s'inicien de nou.

La representació d'aquesta pantalla es troba a la *Figura 9*.

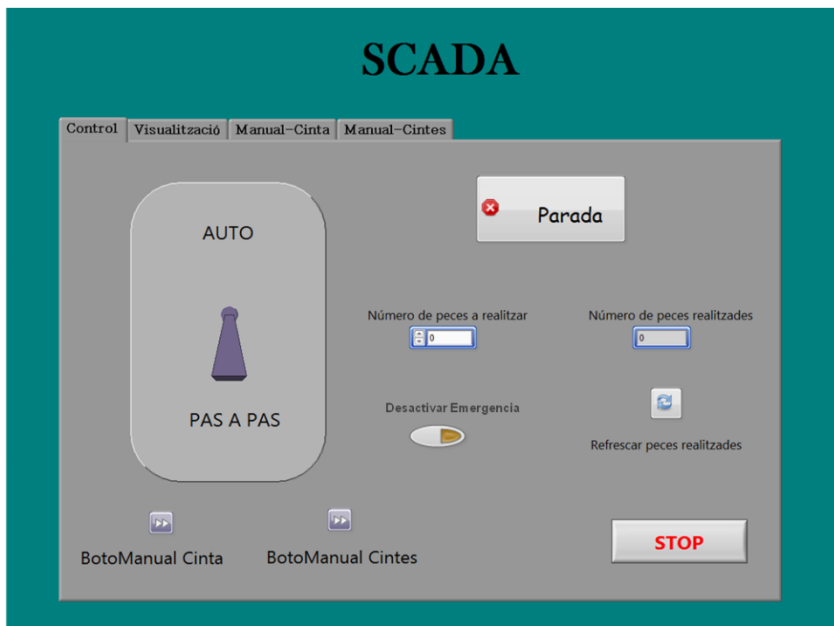


Figura 11. Pantalla de control del SCADA.

### 11.3.2.3- Pantalla de moviment

Tenim dos pantalles de moviment, una per a cada autòmat (*Figures 12 i 13*). Des d'aquestes tenim la capacitat de realitzar els moviments de les cintes, tant de les Fischer com de la cinta transportadora, polsant els botons corresponents, sense necessitat de realitzar cap programa específic. Aquests botons es troben enllaçats amb les eixides de l'autòmat corresponent.

Serveix principalment per quan s'ha produït una avaria i ens interessa engegar algun motor del procés.

Aquests controladors són polsadors, es troben actius únicament mentre els mantenim polsats. Encara que no tenen cap programa assignat i els movem nosaltres amb l'ordre que desitgem, si que respecten els finals de carrera, ja que aquestes accions de moviment es troben dins d'un programa condicionat, que permet el moviment mentre no trobe cap final de carrera.

En el cas de la cinta transportadora, l'única variable que ens interessa poder canviar de valor és la que mou la cinta (**Cinta**), perquè el seu procés és bastant senzill i no tenim més variables que controlar. Pel contrari, en el procés de les cintes de Fischer si que hi ha gran número de variables d'eixida que podem governar des d'aquesta pantalla, tots els motors que es troben a aquest procés: el motor de cadascuna de les diferents cintes, el dels empenyedors movent-se en els dos sentits (avant i arrere) i el de la fresadora i el trepant. Tot açò queda representat mitjançant polsadors a la pantalla de moviment manual.

Quan volem moure manualment hem de tenir en compte què si l'interruptor es troba en mode automàtic el robot està en funcionament i si ens trobem en pas a pas no. Segons si ens interessa que el robot funcione amb normalitat o no hem de tenir l'interruptor en una direcció o altra. Com

normalment, aquest funcionament és per a quan entrem en emergència, mentre estiga la parada d'emergència pulsada és indiferent la posició de l'interruptor ja que el robot no es mou. Però, si volguérem realitzar moviments quan no ens trobem en emergència és important tindre present aquesta consideració.

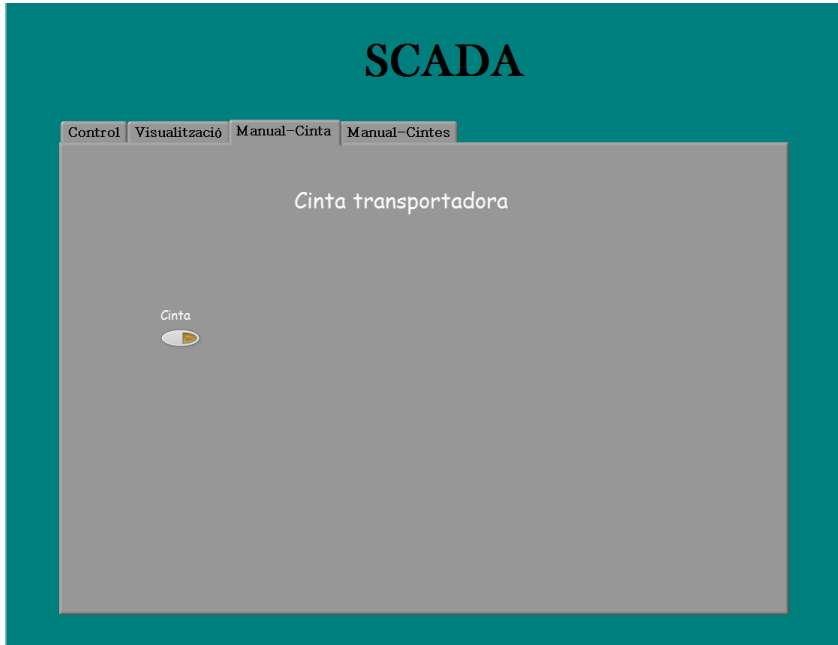


Figura 12. Pantalla de moviment manual de la cinta ABB al SCADA.

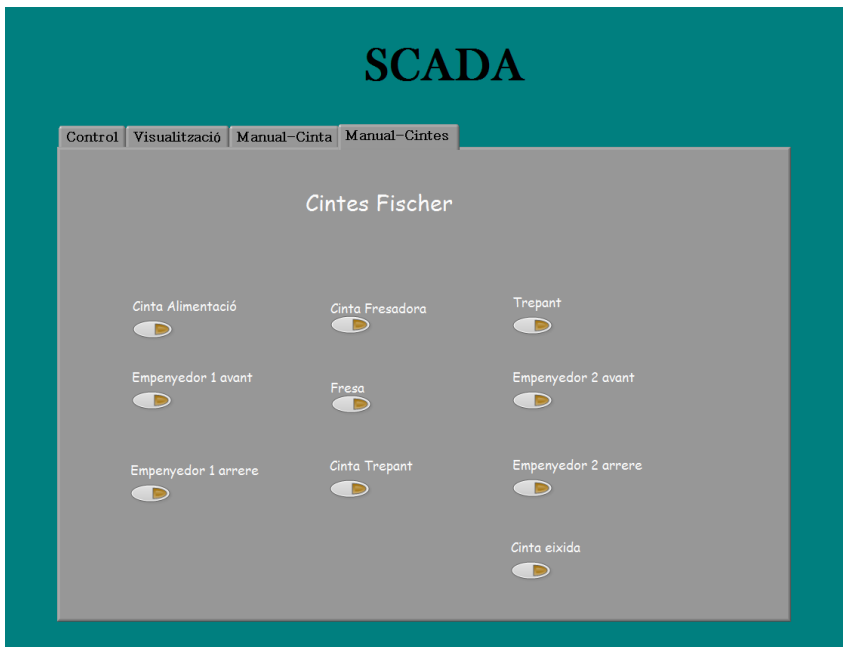


Figura 13. Pantalla de moviment manual de les cintes Fischer al SCADA.

## 12.- Manual del programador

En aquest manual farem referència a tot allò que hem utilitzat durant el treball, de forma que una persona amb coneixements bàsics sobre aquesta matèria, veient els programes i amb aquestos aclariments puga comprendre perfectament què és el que s'està fent a cada moment.

### 12.1.Entrades/Eixides

Els PLCs utilitzats tenen una sèrie d'entrades i eixides físiques assignades. Segons l'autòmat utilitzat aquestes es designen de forma diferent. Al M241 de Schneider, les eixides són %QX i les entrades %IX.

#### 12.1.1 Cinta transportadora

En el cas de les eixides i entrades de la cinta, hi ha un conjunt d'aquestes que es troben disponibles i s'ha programat perquè estiguen relacionades amb les que més ens interessaven. En aquest cas tenim el moviment de la cinta en un sentit i el sensor de final de cinta.

Entrada	Direcció	Descripció
I3	%IX0.3	Fotosensor final de la cinta

Taula 1. Entrades cinta ABB

Eixida	Direcció	Descripció
Q8	%QX0.8	Motor de la cinta en sentit 1

Taula 2. Eixides cinta ABB

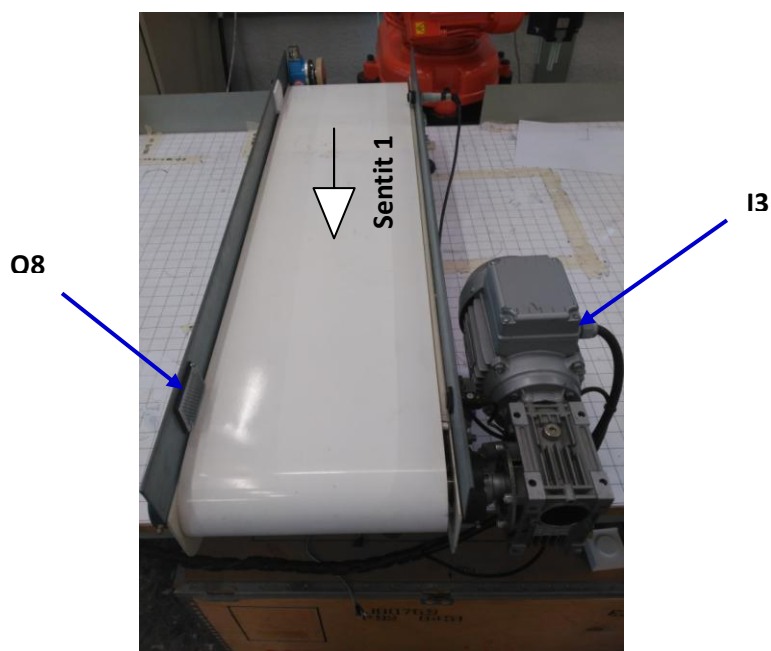


Figura 14. Sensors i motor de la cinta ABB.

### 12.1.2 Cintes Fischer

En aquest cas hi ha entrades i eixides ja cablejades per a controlar tot el procés. Com a entrades trobem els fotosensors, que són de lògica inversa (per defecte es troben actius i en detectar peça donen 0) i els finals de carrera. Com a eixides tenim tots els motors de les "màquines": els de les cintes, els empenyadors, la fresadora i el trepant.

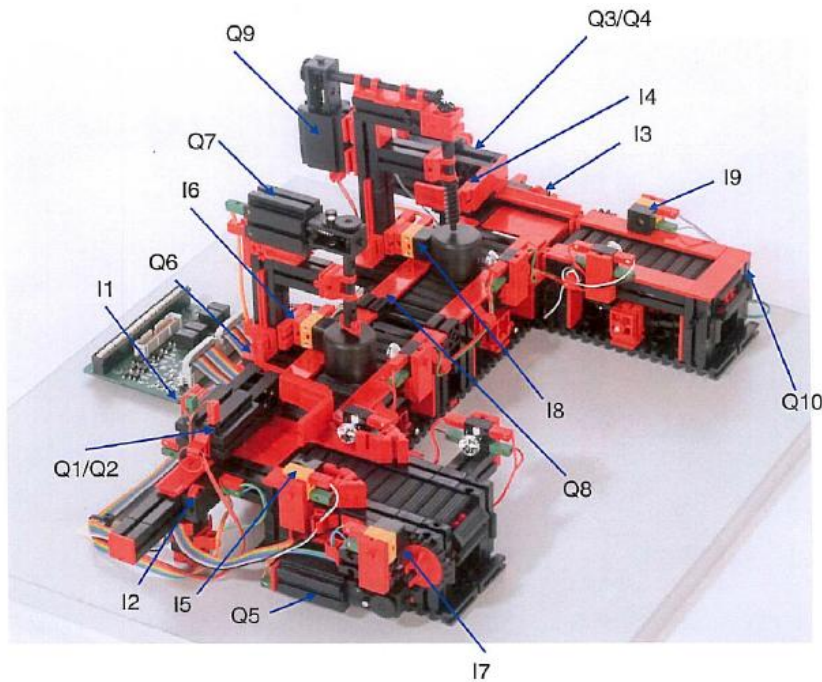


Figura 15. Sensor i motors de les cintes Fischer(pdf *Procesos FischerTechnik(Entradas-Salidas)* de l'assignatura de LAC).

Entrada	Direcció	Descripció
I1	%IX0.0	Final de carrera frontal empenyedor 1
I2	%IX0.1	Final de carrera darrer empenyedor 1
I3	%IX0.2	Final de carrera frontal empenyedor 2
I4	%IX0.3	Final de carrera darrer empenyedor 2
I5	%IX0.4	Fototransistor empenyedor 1
I6	%IX0.5	Fototransistor fresadora
I7	%IX0.6	Fototransistor estació de càrrega
I8	%IX0.7	Fototransistor trepant
I9	%IX1.0	Fototransistor cinta d'eixida

Taula 3. Entrades cintes Fischer



Eixida	Direcció	Descripció
Q1	%QX0.4	Motor empenyedador 1 cap avant
Q2	%QX0.5	Motor empenyedador 1 cap arrere
Q3	%QX0.6	Motor empenyedador 2 cap avant
Q4	%QX0.7	Motor empenyedador 2 cap arrere
Q5	%QX1.0	Motor cinta d'alimentació
Q6	%QX1.1	Motor cinta fresadora
Q7	%QX1.2	Motor fresadora
Q8	%QX1.3	Motor cinta trepant
Q9	%QX1.4	Motor trepant
Q10	%QX1.5	Motor cinta eixida
Q11	%QX1.6	Habilitar sensors i empenyedors

**Taula 4. Eixides cintes Fischer**

### 12.1.3 Robot

En el cas del robot és bastant diferent. En general, no hem d'activar una variable concreta perquè es moga, si no que li hem d'indicar el punt fins on volem que vaja. Per a indicar-li-ho tenim dos opcions, moviment lliure o amb les fletxes o variables que es troben al comandament de moviment.

Amb el moviment lliure, hem de mantenir polsant el botó de moviment lliure per tal que els frens del robot es desactiven i poder moure el robot manualment a la posició que més ens interessa. És més senzill que amb les fletxes, però té un inconvenient, es necessiten dos persones, una per prémer el botó i altra per moure el robot.

Per altra banda, tenim el moviment mitjançant fletxes, d'aquesta forma polsant les fletxes (que realitzen moviments relatius a la ferramenta, si indiquem que pugui moure la ferramenta cap a dalt) movem el robot i podem també realitzar els girs de les articulacions.

Els possibles moviments que pot realitzar són girs de cadascuna de les seues articulacions, desplaçaments i girs respecte la ferramenta i moure la ferramenta cap a dalt i cap

Els moviments es troben limitats per els límits de junta, de forma que quan el robot es troba molt pròxim a un d'ells entra en parada de protecció.

El que hem indicat del moviment del robot serveix per a la seua programació, en el programa dins d'un punt de pas hem de realitzar una de les operacions de moviment indicades i després, dins del programa, ja s'encarrega ell d'activar les variables necessàries per a moure's fins al punt corresponent.

Pel contrari, per a indicar-li si ha d'obrir o tancar la pinça de la ferramenta, ho fem al programa mitjançant una acció Ajustarem la pinça a nivell alt quan vulguem que es trobe tancada i a nivell baix quan vulguem que estiga oberta. Eixa és l'única acció (eixida) que activem, en el nostre cas, des del programa.

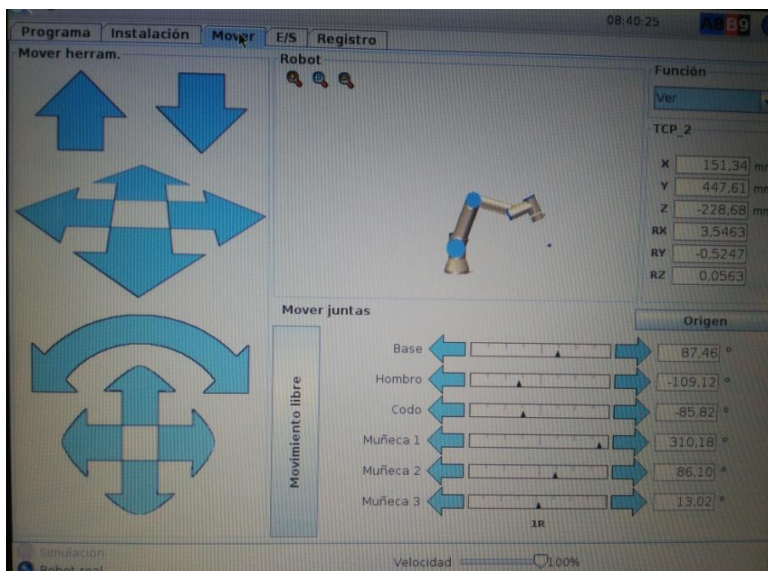


Figura 16. Pantalla de moviment del robot del PolyScope.

## 12.2.Variables

Per a realitzar la programació hem hagut de crear un conjunt de variables per activar senyal, compartir-les amb altres PLCs, realitzar conversions... Depenent de l'ús que els donem, les assignem o no a una direcció concreta i triem el tipus de dades que seran: booleans, vectors, paraules...

En aquest apartat detallem les variables més importants què ens ha fet falta crear. Les creades específicament per a utilitzar-les a algun mòdul de funció (al SoMachine), s'expliquen més avant a l'apartat corresponent.

La majoria de les variables que hem necessitat crear, són variables compartides, és a dir variables necessàries per a rebre o enviar informació de i a altres PLCs. Les variables que reben els PLCs des de el robot(registres) s'emmagatzemen en les direccions **%MD** i les què reben del LabView en les **%MW**. El robot les què envia ho ha per registre i les que rep (eixides) queden guardades a variables d'eixida de MODBUS.

### 12.2.1 Cinta transportadora

#### 12.2.1.1- Variables Compartides

Amb el robot:

- **DeixarP**, %MD0, es enviada des del robot quan aquest deixa una peça sobre l'inici de la cinta.
- **RobotArreplega**, %QX0.2, s'envia de la cinta al robot, quan aquesta té una peça al final, la peça ha sigut detectada pel sensor de final de cinta (Step1).
- **AutoRobot**, %QX0.1, quan la cinta rep del LabView la senyal d'automàtic, la cinta li ho comunica al robot mitjançant aquesta senyal.
- **ParadaRobot**, %QX0.3, igual que amb la variable anterior, quan la cinta rep del LabView la senyal de parada, aquesta li l'envia al robot perquè es detinga.
- **PRealitzades**, %MD2, el robot envia el nombre de peces que han sigut realitzades completament, han finalitzat tot el procés.
- **Introduir**, %QX0.9, el robot llig aquesta eixida digital i mentre es trobe activa (igual a 1), el robot continua introduint peces a la cinta.

Amb el Labview:

- **Automatic**, %MW5, és la variable on s'emmagatzema el mode de funcionament enviat des de LabView, en cas de ser automàtic, donarà 1.
- **BM**, %MW6, es rep del LabView, fa referència al botó de manual, serà 1 quan el botó estiga polsat.
- **BP**, %MW4, el LabView escriu un 1 en aquesta variable quan entra en parada d'emergència, és a dir, en polsar el botó de parada.
- **MovCinta**, %MW7, serveix per activar el motor de la cinta des de LabView, mitjançant la pantalla de moviment manual.
- **PecesARealitzar**, %MW8, des del LabView introduïm el nombre de peces que volem realitzar.
- **Refresh**, %MW9, per saber quan hem de reinicialitzar el nombre de peces realitzades, serà 1 en polsar el botó de refrescar del LabView.

- **PecesRealitzades**, %MW10, enviem el nombre de peces realitzades al LabView per posar-ho a l'indicador corresponent.
- **LlevarEmergencia**, %MW11, quan ha aparegut al LabView l'emergència del robot tenim l'opció de llevarla polsant el botó que activa aquesta senyal

### 12.2.1.2- Variables de la cinta

- **Rob**, és un vector de booleans que utilitzem per a guardar el que s'envia des de la senyal *DeixarP* del robot, utilitzarem el primer booleà (*Rob[0]*) com a condició per a saber si el robot ja ha deixat una peça o no.
- **Cinta**, %QX0.0, aquesta variable s'activa en engegar-se el motor de la cinta, és a dir, quan Q8 es troba activada. L'hem creada perquè des del LabView sols llig els huit primer bits de cada entrada o eixida digital (llig fins la %QX0.7) i com que la Q8 es troba a la direcció %QX0.8 no la pot llegir directament.
- **BotoParada**, és el que obtenim de la conversió de paraula a bit de la variable *BP*. Fem aquesta conversió per a poder utilitzar-la com a condició al programa.
- **BotoManual**, és el que obtenim de la conversió de paraula a bit de la variable *BM*.
- **Auto**, és el que obtenim de la conversió de paraula a bit de la variable *Automatic*.
- **MCinta**, és el que obtenim de la conversió de paraula a bit de la variable *MovCinta*.
- **Condicio**, hem creat aquesta variable per a gastar-la de condició al SFC que ens serveix per saber si el robot ha deixat una peça nova. Es dona en el flanc de pujada del Step0 del SFC *Cinta ABB*.
- **Emergencia**, és el que s'obté de la conversió de *LlevarEmergencia* a booleà.

## 12.2.2 Cintes Fischer

### 12.2.2.1- Variables compartides

Amb el robot:

- **FinPieza**, %QX0.1, quan la peça arriba al final de les cintes (Step4), aquesta variable s'activa i li indica al robot que ja pot anar a recollir la peça.
- **LRobot**, %MD0, aquesta variable és 1 quan el robot ha retirat la peça.

Amb LabView:

- **PiezaCintes23**, %QX0.0, indica que la peça ha arribat al final de la primera zona, va a entrar o ja ha entrat a la segona zona (*Cintes23*).
- **FinPieza**, %QX0.1, igual que el descrit a les variables compartides amb el robot.
- **BP**, %MW4, igual que a les cintes.
- **BM**, %MW6, igual que a les cintes.
- **Automatic**, %MW5, igual que a les cintes.
- **Alarma1**, %QX3.0, al LabView hi ha un led que s'encén quan aquesta s'activa. S'activa si hi ha problema a la cinta d'alimentació.

- **Alarma2**, %QX3.1, s'activa si el motor de l'empenyedador 1 no funciona cap avant.
- **Alarma3**, %QX3.2, s'activa si el motor de l'empenyedador 1 no funciona cap arrere o la cinta de la fresa.
- **Alarma4**, %QX3.3, si hi ha problema en les cintes 2 i/o 3 (de la fresa i el trepant).
- **Alarma5**, %QX3.4, s'activa si el motor de l'empenyedador 2 no funciona cap avant.
- **Alarma6**, %QX3.5, s'activa si el motor de l'empenyedador 2 no funciona cap arrere o la cinta d'eixida.
- **CintaAlimentacio**, %MW7, per a moure des de la pantalla de moviment manual la cinta d'alimentació.
- **Emp1avant**, %MW8, per a moure des de la pantalla de moviment manual l'empenyedador 1 cap avant.
- **Emp1arrere**, %MW9, per a moure des de la pantalla de moviment manual l'empenyedador 1 cap arrere.
- **CFresa**, %MW10, per a moure des de la pantalla de moviment manual la cinta de la fresadora.
- **Fresa**, %MW11, per a moure des de la pantalla de moviment manual el motor de la fresa.
- **CTrepant**, %MW12, a moure des de la pantalla de moviment manual la cinta del trepant.
- **Trepant**, %MW13, a moure des de la pantalla de moviment manual el motor del trepant.
- **Emp2avant**, %MW14, per a moure des de la pantalla de moviment manual l'empenyedador 2 cap avant.
- **Emp2arrere**, %MW15, per a moure des de la pantalla de moviment manual l'empenyedador 2 cap arrere
- **CEixida**, %MW16, a moure des de la pantalla de moviment manual la cinta d'eixida.
- **LlevarEmergencia**, %MW17, com a la cinta ABB.

#### 12.2.2.2- Variables de les cintes

- **FSinit1**, per donar permís al segon SFC (*Cintes23*) perquè comence, s'activa al finalitzar el primer SFC, al flanc de baixada del Steps 1 i 6.
- **FSinit2**, quan aquesta s'activa, el SFC tercer (*Cinta4*) ix de l'etapa inicial. Açò ocorre en el flanc de baixada dels Steps 5 i 17.
- **Robot**, vector de booleans per a convertir el *LRobot* de forma que utilitzem el primer bit(*Robot[0]*) com a condició per a saber si el robot ja ha recollit la peça del final de les cintes.
- **F**, aquesta variable en ens indica si la peça ha arribat a la fresa. La posem a zero a l'inici del segon SFC(*Cintes 23*) i a 1 quan el fotosensor de la fresadora detecta que hi ha peça. La creem per a utilitzar-la com a condició a l'hora d'activar l'*Alarma3*.
- **BotoParada**, com a la cinta transportadora.
- **BotoManual**, com a la cinta transportadora.
- **Auto**, com a la cinta transportadora.
- **Alimentacio**, és el que obtenim de la conversió de paraula a bit de la variable *CintaAlimentacio*.
- **E11**, és el que obtenim de la conversió de paraula a bit de la variable *Emp1avant*.
- **E12**, és el que obtenim de la conversió de paraula a bit de la variable *Emp1arrere*.

- **E21**, és el que obtenim de la conversió de paraula a bit de la variable *Emp2avant*.
- **E22**, és el que obtenim de la conversió de paraula a bit de la variable *Emp2arrere*.
- **CF**, és el que obtenim de la conversió de paraula a bit de la variable *CFresa*.
- **MotorFresa**, és el que obtenim de la conversió de paraula a bit de la variable *Fresa*.
- **CT**, és el que obtenim de la conversió de paraula a bit de la variable *CTrepant*.
- **MotorTrepant**, és el que obtenim de la conversió de paraula a bit de la variable *Trepant*.
- **CE**, és el que obtenim de la conversió de paraula a bit de la variable *CEixida*.
- **Refrescar**, és el que obtenim de la conversió de paraula a bit de la variable *Refresh*.
- **Emergencia0**, com a la *Cinta ABB*.

### 12.2.3 Robot

#### 12.2.3.1- Variables compartides

En el cas del robot, les variables compartides són les que creem com a variables d'instal·lació al MODBUSClient. Tant les que obtenim al llegir dels autòmats (eixides digitals), com les que escrivim (registres).

De les cintes Fischer:

- **FiP**, eixida digital 1, la llegim del PLC i l'utilitzem com a condició al programa, per indicar-li al robot que quan aquesta es trobe activa haurà d'anar a recollir una peça al final de les cintes.
- **PRecollida**, eixida de registre 0, indica que el robot ja ha recollit la peça, al programa l'ajustem a 1 en el moment que això es compleixca i se li envia al PLC.

De la cinta transportadora:

- **RobotArreplega**, eixida digital 2, la llegim del PLC i ens serveix com a condició al programa, per a que el robot sàpiga quan ha d'anar a arreplegar la peça al final de la cinta.
- **DeixarP**, eixida de registre 0, es per a saber que la peça ha sigut deixada pel robot a l'inici de la cinta. Al programa s'ajusta a 1 després que el robot deixi la peça on toca i s'envia la senyal al PLC.
- **Auto**, eixida digital 1, per saber el mode de funcionament en que ens trobem, si és 1 ens trobe en mode automàtic i és el cas en que el robot pot funcionar. La llig el robot des de l'autòmat.
- **PARADA**, eixida digital 3, quan polsem al LabView el botó de parada d'emergència, s'activa aquesta variable prohibint que el robot funcione. El robot la llig des de l'autòmat.
- **INICI**, eixida digital 6, serveix perquè el robot estiga en mode espera fins que el programa de la cinta transportadora es posa en marxa, en aquest moment la variable aquesta serà 1 i el robot començarà a moure's.
- **Realitzades**, eixida de registre2, li envia a l'autòmat el nombre de peces que ja ha finalitzat, l'autòmat li ho envia al LabView des d'on pot observar-ho l'usuari.
- **Introduir**, eixida digital 9, llig l'eixida corresponent de l'autòmat i mentre aquesta estiga activada, podrà continuar introduint peces a la cinta.

### *12.2.3.2- Variables del programa*

Són aquelles variables que creem al mateix programa, aquestes al contrari que les d'instal·lació sí que es poden inicialitzar.

- **PFinalitzades**, es una variable que inicialitzem a zero i cada vegada què el robot deixa una peça a la posició final (magatzem final), s'incrementa en una unitat, de forma que funciona com un comptador del nombre de peces realitzades.

### *12.2.3.3- Variables d'instal·lació*

- **c**, aquesta variable pot tindre el valor 1 o 2, inicialitzem el seu valor a 1 i quan el robot es troba a l'inici de les cintes Fischer el seu valor canvia a 2. Es deu a què si ocorre una parada i el robot ha de tornar a la posició inicial, si es troba en una posició baixeta ( $c=2$ ), té perill de xocar contra la cinta transportadora, per tant, hem afegit una condició què en cas de ser  $c=2$ , el robot s'ha de moure primer a un punt alt de la zona de perill i evitar així que al començar de nou el funcionament normal, xoque.
- **pinza**, és una variable que va enllaçada directament amb una eixida, el moviment de la pinça, quan aquesta es troba a nivell alt, la pinça es tanca i a nivell baix s'obri.

## 12.3.Grafcets

### 12.3.1 Cinta transportadora

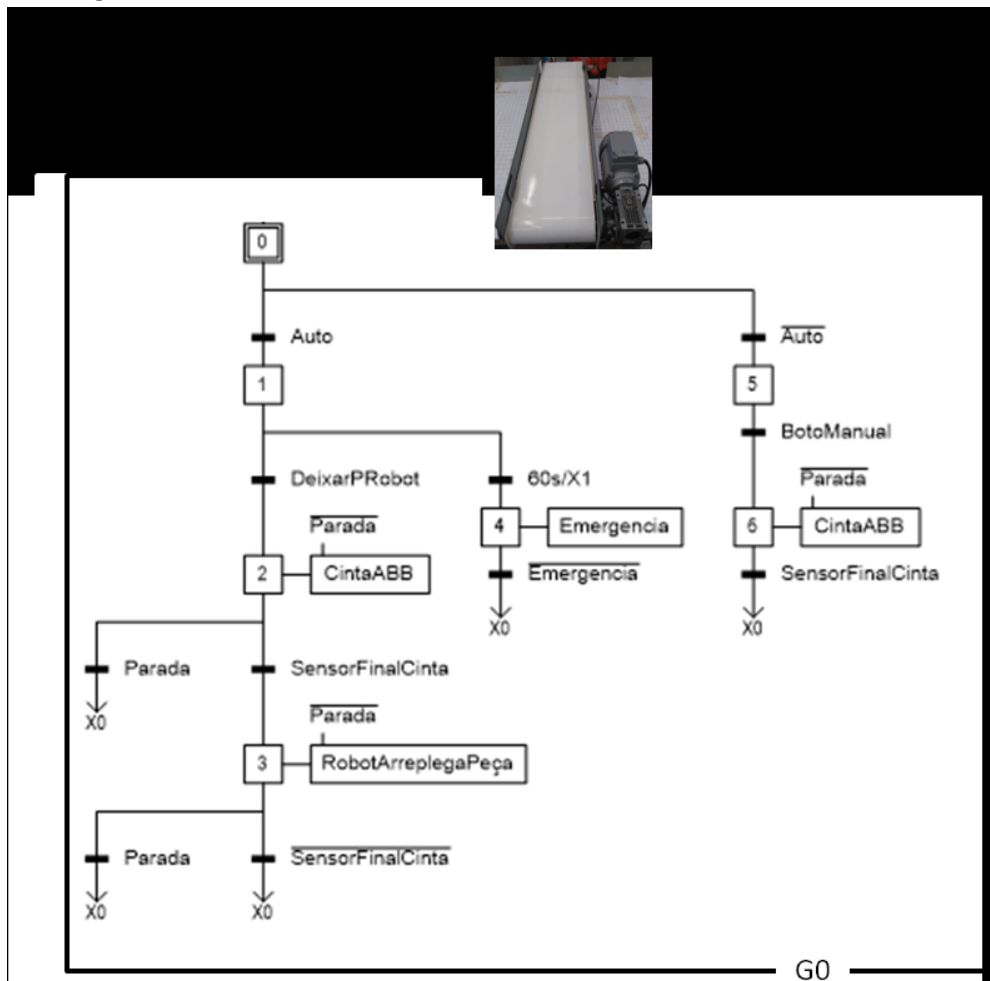


Figura 17. Grafcet CintaABB.

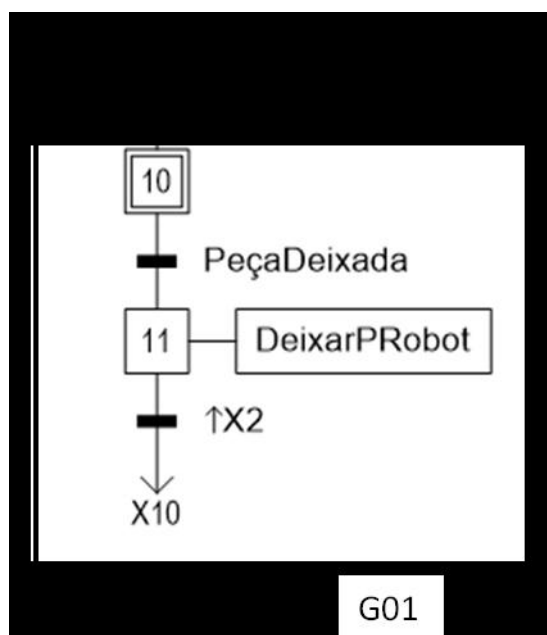


Figura 18. Grafcet el robot deixa peça a la cinta.



### 12.3.2 Cintas Transportadoras

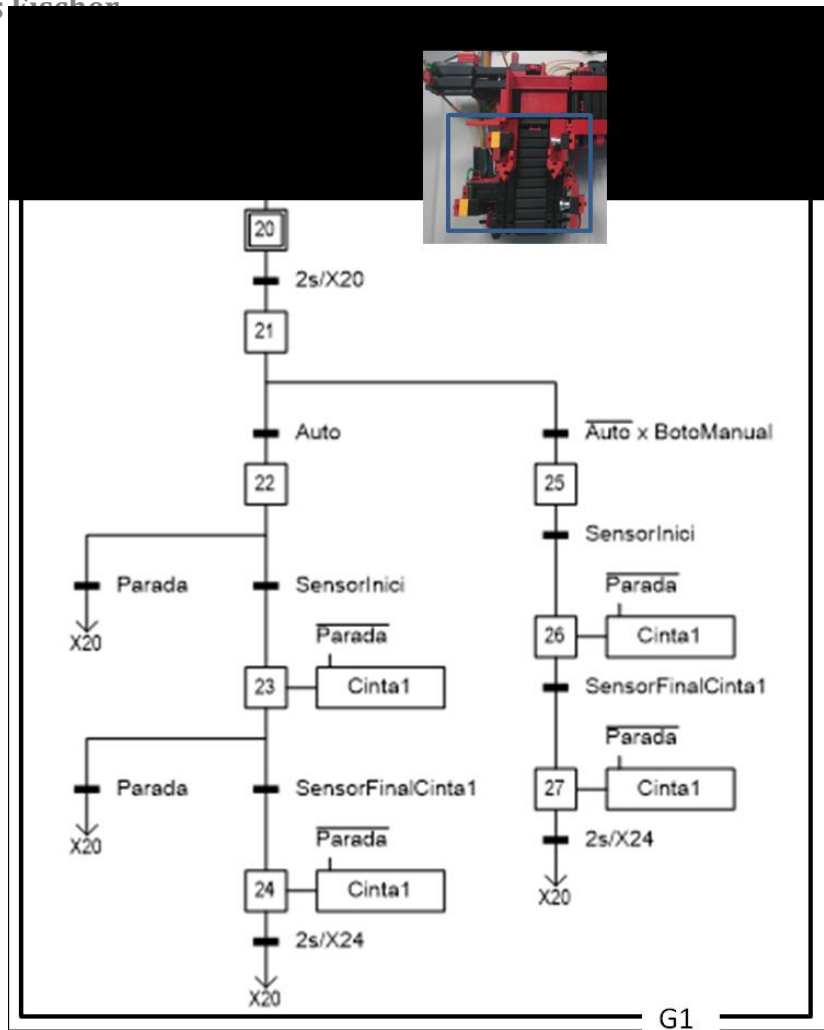


Figura 19. Grafcet Cinta 1.

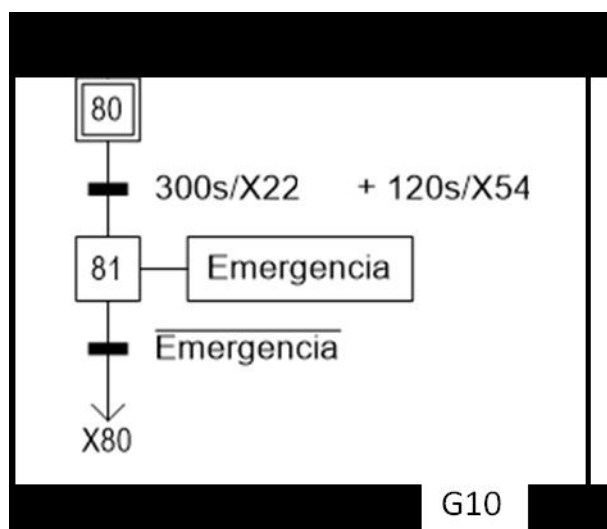
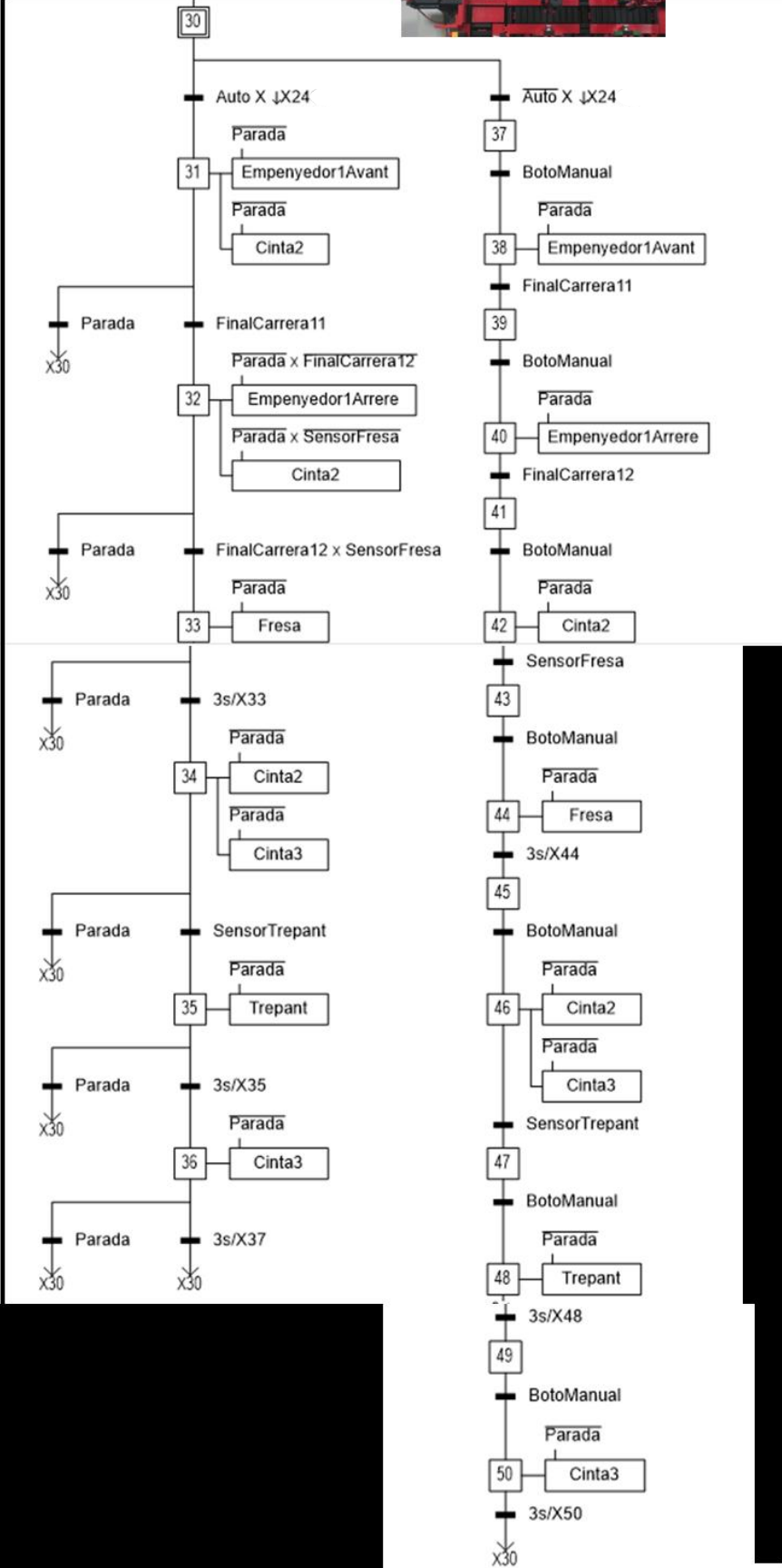


Figura 20. Grafcet Emergencia



G23

Figura 21. Grafcet Cintes23

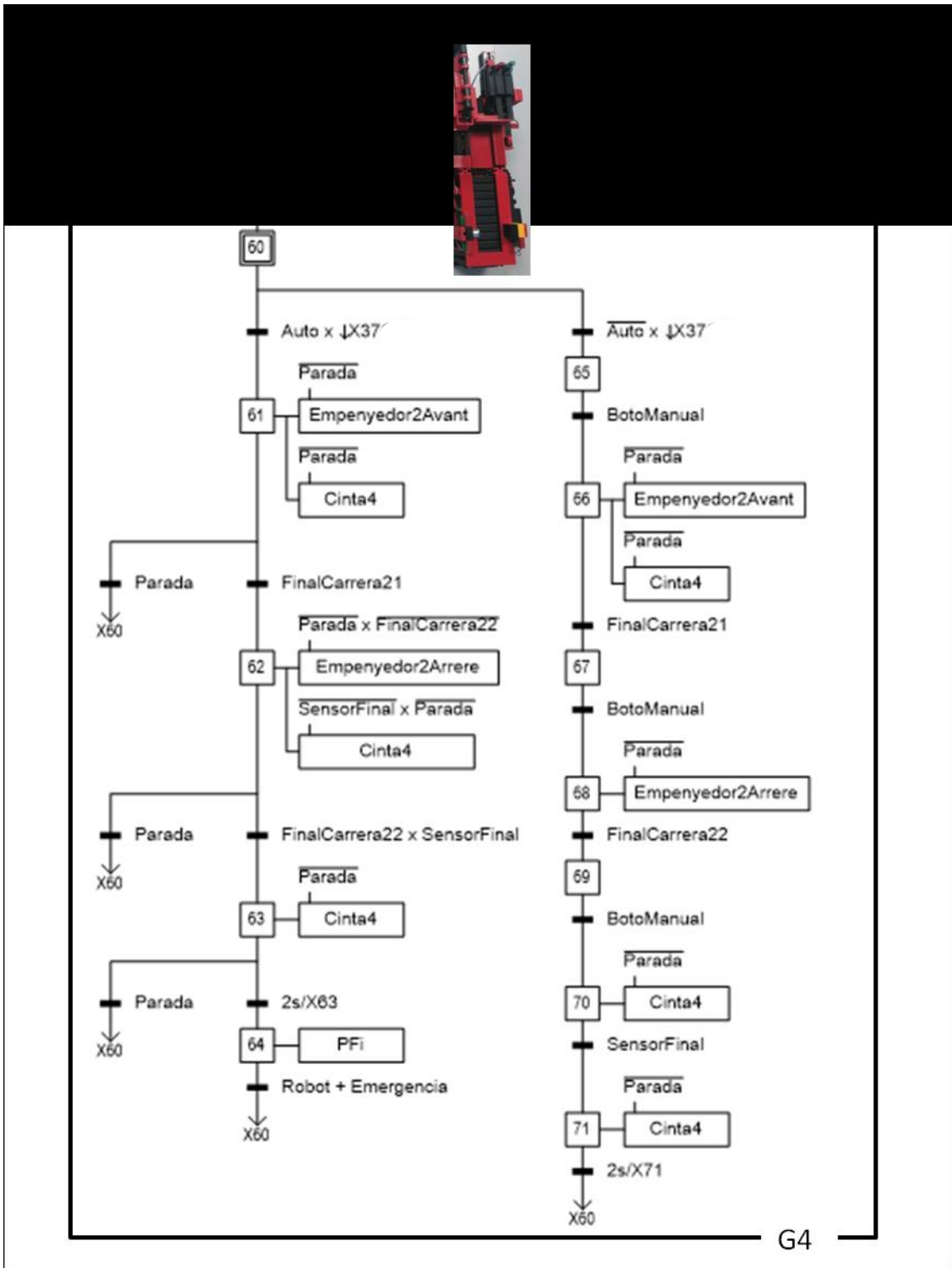


Figura 22. Grafcet Cinta4.

12.4.Programa del robot

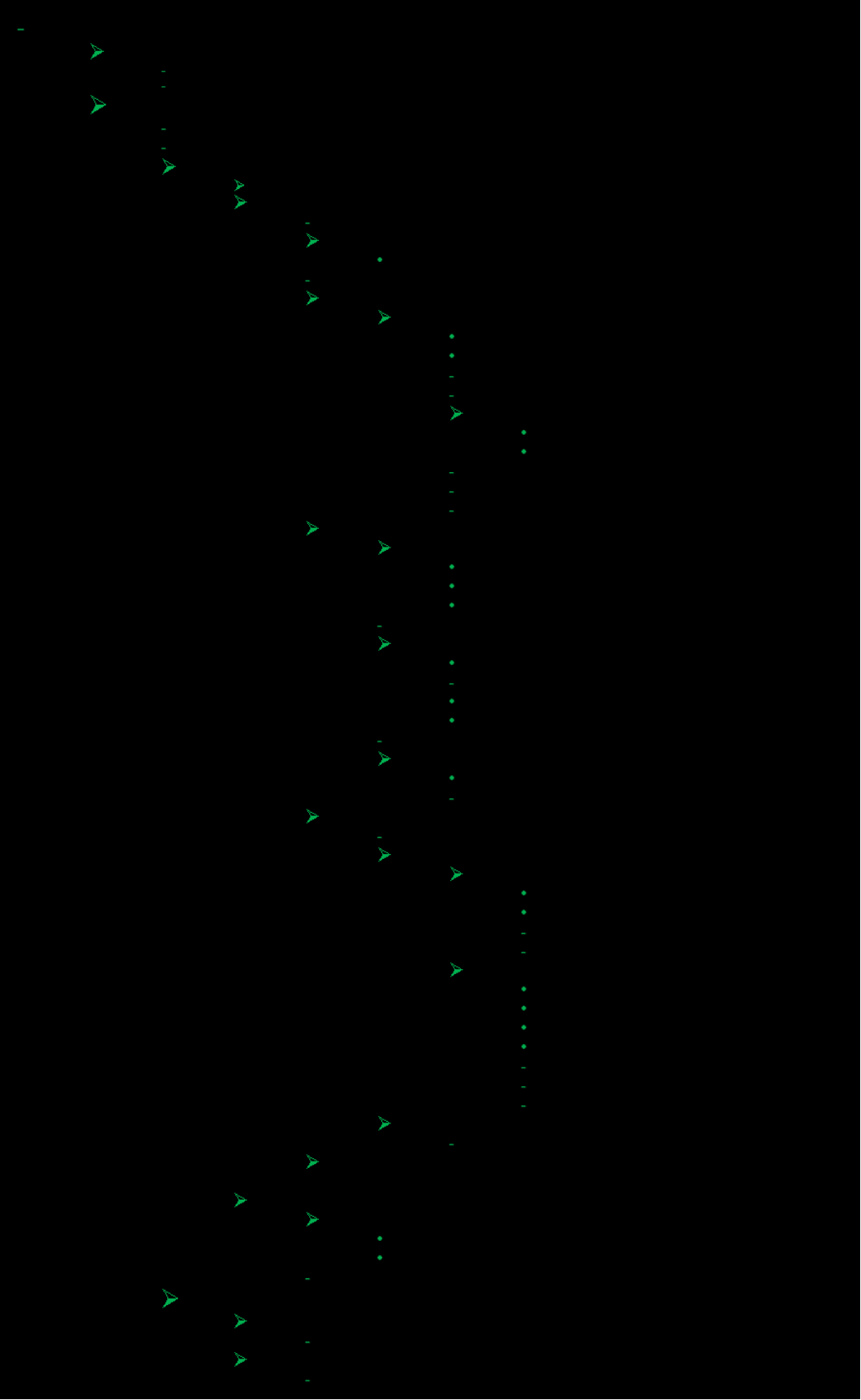


Figura 23. Programa del robot.

## 12.5.Mòduls de funció

Per la programació dels PLCs hem utilitzat diferents mòduls de funció al SoMachine, com per exemple temporitzadors, move...

Els blocs de funcions que es descriuen a continuació han d'anar associats a una etapa, tenint en compte que sols es produiran mentre aquesta etapa estiga activa. El llenguatge de programació utilitzat és el diagrama de contactes(LD, Ladder Contact).

### 12.5.1 Temporitzador

En tots els SFC tenim diverses esperes de temps entre etapes, però temporitzadors de mòdul de funció sols hem utilitzat un, al primer SFC de les cintes Fischer(*Cinta1*). L'altra forma de comptar el pas del temps sense utilitzar un temporitzador com a tal, és indicar els segons que volem que passen des que s'ha activat un pas, per exemple, si volem activar una variable tres segons després que s'active la segona etapa del SFC *Cinta1* posaríem **Cinta1.Step2.t>T#3s**, per tant quan hagen passat tres segons es compleix eixa condició. Aquest és una manera molt ràpida de posar condicions de temps en les transicions, i ens estalviem els mòduls de funció que ens donen més feina.

En el cas de la primera transició del SFC, de la forma que hem explicat abans no funciona aquest comptador la primera vegada que s'inicia el SFC, per això hem hagut de posar un temporitzador. Al temporitzador cal indicar-li quina és l'entrada que inicia el compte d'aquest, que a més, quan es dona el flanc de baixada d'aquesta entrada, es reinicialitza el compte del temporitzador, li indiquem el nombre de segons que volem que deixi passar i a més creem una variable que compta el temps que va passant des que es posa en marxa el temporitzador, en aquest cas la variable és *Timer\_Elapsedtime*, hem d'indicar també el nom del temporitzador *TON\_1*. Per a posar-lo de condició, hem de dir que quan ja s' haja complit aquest mòdul de funció, que serà quan haja passat el temps establert(en aquest cas dos segons), passe d'etapa, per això hem de posar el nom del temporitzador punt Q, és a dir, *TON\_1.Q*.

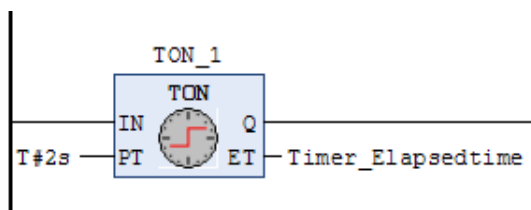


Figura 24. Temporitzador.

## 12.5.2 MOVE

Aquest mòdul de funció serveix per a traslladar informació, pot ser d'una variable a altra, donar valor a una variable... L'entrada i l'eixida han de ser del mateix tipus. Per exemple, l'utilitzem per canviar el valor de *F* de 0 a 1 i al contrari. Si volem donar-li a una variable un valor enter, posem a l'entrada l'enter i a l'eixida la variable a qui li volem assignar aquest valor. En cas de voler assignar un número en binari, cal posar davant un dos i una etiqueta(2#), per exemple 2#101.

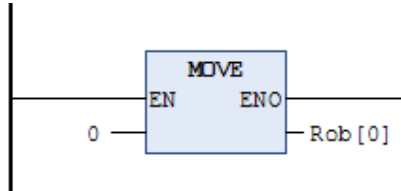


Figura 25. MOVE

## 12.5.3 DWORD\_AS\_BIT

Serveix per a convertir un DWord(*Double Word*) en bits, descomposar-ho. L'utilitzem per transformar el que guardem a un registre a partir del que ha escrit el robot, per obtenir els bits d'aquest. Sols utilitzem un bit perquè només ens fa falta saber si ha fet alguna cosa o no, per tant, amb un bit tenim suficient.

Per exemple, per a indicar al procés de les cintes que el robot ha recollit la peça que es trobava al final d'aquestes. Hem utilitzat altre com aquest per a indicar-li a la cinta transportadora que el robot ha deixat una peça sobre ella.

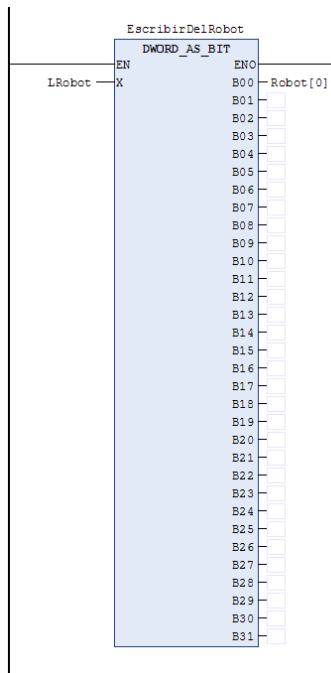


Figura 26. DWORD\_AS\_BIT

### 12.5.4 WORD\_AS\_BIT

És un convertidor de paraules a bits(booleans). L'utilitzem per convertir el registre rebut des del LabView en el SoMachine. Aquestos registres s'activen en prémer un botó per tant, sols ens interessa l'últim bit que en cas de ser un 1 ens indica que el botó està polsat i si és un 0 no es troba polsat. El convertim perquè així podem utilitzar aquesta variable booleana, per exemple com a condició en els nostres programes, per a activar(o no) una variable o com a transició en una etapa.

Tenim un convertidor d'aquest tipus per a cada botó de la pantalla de control del LabView tant a les cintes com a la cinta transportadora. Per al mode de funcionament, el botó de parada, el botó manual, el botó de refrescar. A més, per a la pantalla de moviment manual del LabView, tenim també un per cada polsador, per a l'empenyedor 1 avant, l'empenyedor 1 arrere, el motor de la fresa...

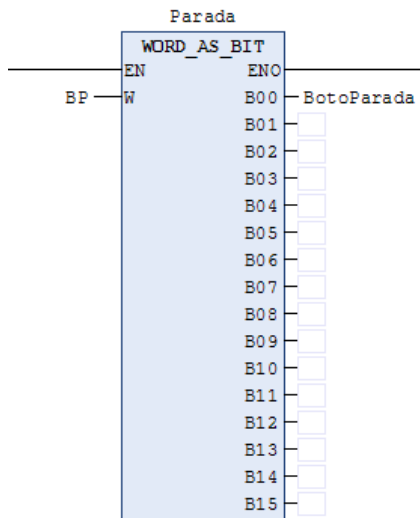


Figura 27. WORD\_AS\_BIT

### 12.5.5 Comptador

Utilitzem un comptador al programa de la *Cinta ABB* per saber el nombre de peces que hem introduït a aquesta i indicar-li-ho al robot per tal que introdueixca únicament el nombre de peces que volem realitzar.

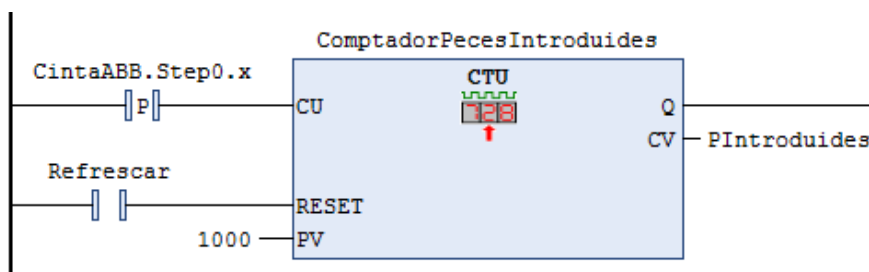


Figura 28. Comptador

**CU (Counter Up)**, incrementa el compte en una unitat cada vegada que ens trobem al flanc de pujada del Step0 del SFC *Cinta ABB*.

**CV**, ací indiquem la variable on es guarda el compte del número de peces introduïdes, *PIntroduïdes*.

**RESET**, per posar el comptador a zero, li hem d'indicar què és el que fa que aquest compte es reinicie, en el nostre cas, es reinicia en polsar el botó de refrescar el nombre de peces.

### 12.5.6 Comparador

També per a saber quantes peces ha d'introduir el robot, utilitzem un comparador, de forma que una variable es troba activa mentre el nombre de peces introduïdes siga menor al nombre de peces que li hem indicat des de LabView que volem realitzar.

Gastem un comparador de menor que.

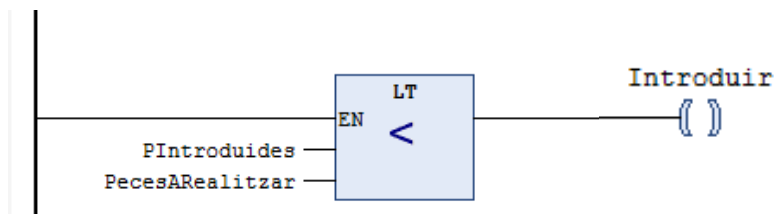


Figura 29. Comparador

La variable que es troba activa mentre es compleix la condició del comparador és *Introduir* (variable booleana). Aquesta variable s'envia al robot i es gasta com a condició perquè aquest pugui introduir peces sempre i quan aquesta variable es trobe activa.



Els mòduls utilitzats particularment per a la connexió entre dos autòmats, són els següents:

### 12.5.7 ADDM

Es tracta d'un bloc de conversió d'adreça, introduïm l'adreça IP del PLC de forma que després el que obtenim a aquest si que es pot llegir com a adreça als mòduls d'escriptura i lectura.

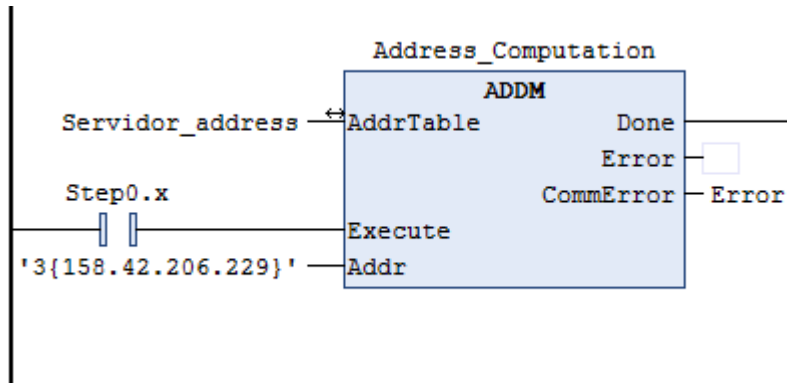


Figura 30. ADDM

Li indiquem la variable on volem que emmagatzeme l'adreça convertida, **Servidor\_address**.

S'executa(**Execute**) quan es trobe activa la condició corresponent, en aquest cas en activar l'etapa Step0.

I hem de posar l'adreça IP entre claus i davant indiquem el número de port, per defecte posem el 3, açò queda explicat al manual del SoMachine, als mòduls de funció per al modbus.

### 12.5.8 READ\_VAR

Per a llegir del servidor utilitzem el bloc de funció **READ\_VAR**(Figura 31).

En la casella **Execute** li indiquem quan s'activarà aquest mòdul, en aquest cas s'activa quan es dona l'etapa *Lectura*, per indicar que es tracta d'una etapa li posem un punt i una ix a continuació del nom de l'etapa, *Lectura.x*. En l'adreça posem la variable que hem convertit al bloc de **ADDM**, *Servidor\_address*, **Timeout** posem per exemple 10 mil·lisegons. **ObjType**, triem si es tracta d'una entrada digital(.I), una eixida digital(.Q) o una variable interna(.MW). **FirstObj** és l'objecte a partir del qual comencem a llegir i **Quantity** la quantitat que llegim. Finalment, el **Buffer** que és el vector on s'emmagatzema *ADR(Vector)*. Mentre s'esta realitzant la funció el **Busy** es troba en TRUE i quan ja s'ha realitzat s'activa el **DONE**. Si es deixa de fer la funció o hi h algú error es posen a TRUE les altres caselles, en **CommError** i **OperError** apareixen uns numerets en cas d'haver-se produït algú error que ens indiquen de quin tipus d'error es tracta. Els tipus d'errors es troben al manual *SoMachine Modbus and ASCII Read/Write Functions PLCCommunication Library Guide*.

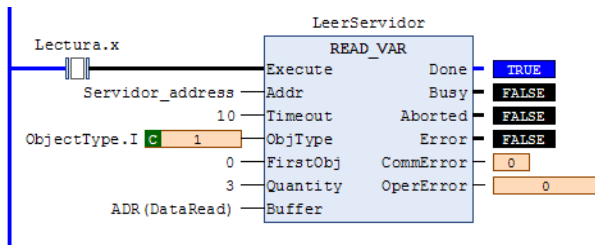


Figura 31. READ\_VAR

### 12.5.9 WRITE\_VAR

El bloc **WRITE\_VAR** s'utilitza d'igual forma que el **READ\_VAR**. En aquest cas s'utilitza per a escriure en el servidor. La diferència és que per activar-lo en aquest cas posarem la seua corresponent etapa, *Escriptura.x* i el tipus d'objecte(**ObjType**) serà un eixida digital(.Q).

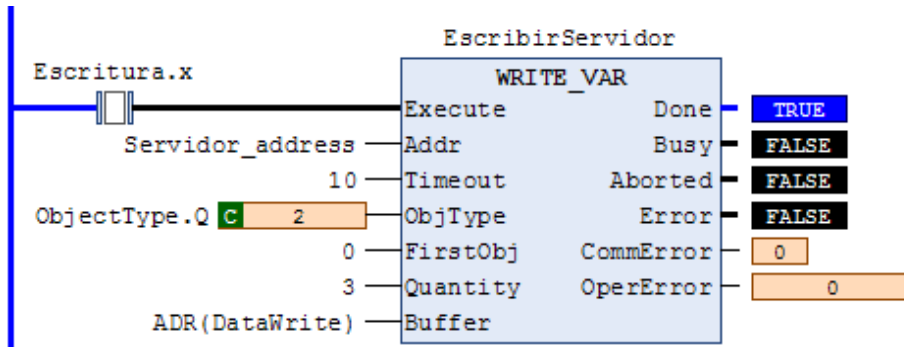


Figura 32. WRITE\_VAR

### 12.6. SFCInit

Per a reinicialitzar el programa amb normalitat després d'una parada d'emergència, utilitzem **SFCInit**. Serveix per a detenir el procés mentre estiga la senyal que l'activa en marxa(*BotoParada*, referit a la parada d'emergència) i una vegada desapareix aquesta senyal, els SFC s'inicien des de l'etapa inicial.

L'hem utilitzat sols per al cas de mode automàtic, per al pas a pas no ens interessa que comence de nou perquè com estem fent proves volem ser nosaltres qui controlem el procés.

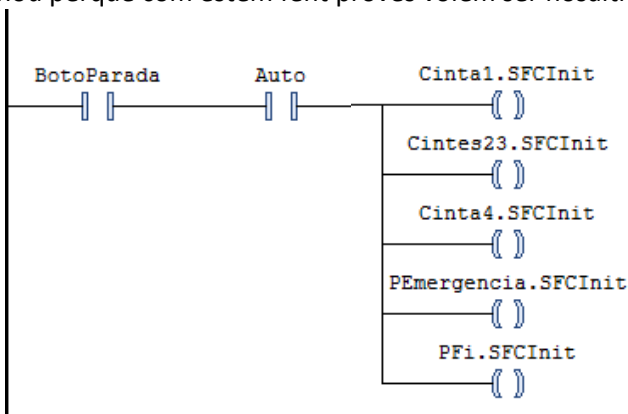


Figura 33. SFCInit

## 13.- Conclusions

El treball compleix tots els objectius establerts a l'inici d'aquest (*Objectius del tfg*).

S'ha aconseguit fer de tres processos diferents fer un únic, relacionant-los i comunicant-los amb el MODBUS. El funcionament final del procés és l'esperat: el robot agafa una peça del magatzem de peces no mecanitzades, la situa a la cinta transportadora, aquesta trasllada la peça fins el final, el robot la recull i la deixa a la línia indexada on la peça es mecanitzada (es realitza un fresat i un trepat).

També s'ha complert amb la realització d'una pantalla SCADA amb la qual es realitza la supervisió i el control del procés. A més, en aquesta el resultat final és millor de l'esperat. S'han separat la supervisió i el control per finestres i s'han afegit dos finestres més per a realitzar el moviment manual dels motors dels processos. I, la finestra de supervisió (pantalla de visualització) té un disseny molt atractiu i que representa amb claredat la distribució del procés real, utilitzant fins i tot els mateixos colors que tenen les maquetes.

En quant a la comunicació, s'ha realitzat correctament i utilitzant el protocol desitjat (MODBUS). Si bé no hem realitzat una comunicació directa del LabView amb el robot (ha resultat complicat i finalment, s'ha decidit fer-ho d'altra forma), si que s'ha aconseguit que es trobaren connectats (hem realitzat la comunicació utilitzant els PLCs com a intermediaris entre aquests, LabView <-> PLC <-> Robot).

A més, a nivell personal, el treball m'ha ajudat a ampliar els meus coneixements amb la utilització de nous programes i el protocol MODBUS (fins el moment desconegut per a mi). També m'ha hagut d'enfrontar amb els problemes que anaven sorgint i prendre decisions a l'hora de resoldre'ls. Els resultats obtesos són per a mi satisfactoris.

## 14.-Vocabulari

- **PLC**, Programmable Logic Controller(controlador lògic programable o autòmat), és un aparell electrònic programable per l'usuari i que s'utilitza per al control.
- **POU**, Program Organization Unit, s'utilitza per a la creació de codis que després seran executats pel PLC.
- **GRAFSET**, Graphe Fonctionnel de Commande Etape Transition(gràfic de control d'etapes i transicions), és un tipus de llenguatge per a la programació de POU's. Com el seu nom indica és la representació d'un sistema i està format per un conjunt d'etapes i transicions.
- **Modicon**, MOdular DIgital CONtroller(Controlador digital modular).
- **RTU** , Remote Terminal Unit(unitat terminal remota), permet l'obtenció de senyals de diferents processos i l'enviament d'aquestes a una unitat remota, com pot ser l'SCADA.
- **TCP/IP**, és el conjunt de dos protocols(els dos més importants):
- **TCP**,Transmission Control Protocol, protocol de control de transmissió.
- **IP**,Internet Protocol, protocol d'internet.
- **SCADA**, Supervisory Control And Data Acquisition (control de supervisió i adquisició de dades), serveix per al control i supervisió a distància de processos. És el nom que se li dona al software que es capaç de realitzar aquestes funcions(supervisió i control).
- **Ethernet**, és una xarxa d'ordinadors que permet que envien i reben dades, per a xarxes d'àrea local(LAN).

## 15.- Bibliografia

### Documents consultats

- *Manual de formació SoMachine* descarregat a la pàgina web d'Info PLC ([www.infopl.net](http://www.infopl.net))
- *Modicon M241 Logic Controller- Guia de programació* descarregat a la web de SchneiderElectric.-([www.schneider-electric.es](http://www.schneider-electric.es))
- *SoMachine-Guia de programació*, referència EIO0000001435.03. Descarregat a la pàgina web de SchneiderElectric. ([www.schneider-electric.es](http://www.schneider-electric.es))
- *SoMachine-Funcions de lectura/escriptura Modbus i ASCII. Guia de la biblioteca PLCComunicacion*, referència EIO0000000744.06. Descarregat a la pàgina web de SchneiderElectric ([www.schneider-electric.es](http://www.schneider-electric.es))
- *Manual d'usuari UR3*, versió 3.3.0. Descarregat a la pàgina web d'UniversalRobots ([www.universal-robots.com](http://www.universal-robots.com))

### Enllaços consultats

- Per a la realització del SCADA:  
<http://www.ni.com/white-paper/7675/es/>  
<http://www.ni.com/tutorial/13911/en/>
- Per a la redacció de la introducció i els antecedents:  
<https://www.infopl.net>  
<http://www.industriaconectada40.gob.es/Paginas/index.aspx#industria-4>

# PRESSUPOSTOS

---

# ÍNDIX DE CONTINGUTS

<b>1.- INTRODUCCIÓ .....</b>	<b>73</b>
<b>2.- COSTOS UNITARIS .....</b>	<b>74</b>
2.1.Mà d'obra.....	74
2.1.1 Enginyer industrial.....	74
2.1.2 Director del projecte .....	74
2.1.3 Tècnic.....	74
2.2.Material .....	74
<b>3.- PREUS UNITARIS DESCOMPOSTOS .....</b>	<b>76</b>
3.1.-Cinta ABB .....	76
3.1.1 Mà d'obra .....	76
3.1.2 Materials.....	77
3.2. Cintes Fischer .....	78
3.2.1 Mà d'obra .....	78
3.2.2 Materials.....	79
3.3. Robot .....	80
3.3.1Mà d'obra .....	80
3.4. SCADA .....	81
3.4.1 Mà d'obra .....	81
3.4.2 Material .....	81
3.5. Redacció .....	82
3.5.1 Mà d'obra .....	82
3.5.2 Materials.....	82
<b>4.- PRESSUPOST TOTAL D'EXECUCIÓ.....</b>	<b>83</b>
<b>5.- PRESSUPOST TOTAL D'EXECUCIÓ PER CONTRACTA.....</b>	<b>83</b>
<b>6.- PRESSUPOST BASE DE LICITACIÓ .....</b>	<b>83</b>

## ÍNDEX DE TAULES

Taula 5. Cost mà d'obra-Enginyer industrial .....	74
Taula 6. Cost mà d'obra-Director del projecte .....	74
Taula 7. Cost mà d'obra-Tècnic .....	74
Taula 8. Cost material.....	75
Taula 9. Cost mà d'obra-Cinta ABB.....	76
Taula 10. Cost material-Cinta ABB.....	77
Taula 11. Cost mà d'obra-Cintes Fischer .....	78
Taula 12. Cost material-Cintes Fischer .....	79
Taula 13. Cost mà d'obra-Robot.....	80
Taula 14. Cost material-Robot.....	80
Taula 15. Cost mà d'obra-SCADA .....	81
Taula 16. Cost material-SCADA .....	81
Taula 17. Cost mà d'obra-Redacció.....	82
Taula 18. Cost material-Redacció.....	82
Taula 19. Pressupost total d'execució .....	83
Taula 20. Pressupost total d'execució per contracta .....	83
Taula 21. Pressupost base de licitació.....	83

## ÍNDEX D'EQUACIONS

Equació 1. Amortització .....	75
-------------------------------	----



## 1.- Introducció

A aquest document realitzem un estudi econòmic de tots els recursos utilitzats al llarg del treball per obtenir el cost total del projecte.

Per a començar a definir el pressupost, indiquem el cost unitari dels distints recursos utilitzats, diferenciant el cost de mà d'obra i el cost de material. Definim les distintes unitats d'obra en què es pot dividir el projecte i obtenim el cost unitari d'aquestes i el cost general de la suma de totes, el pressupost total d'execució material. Finalment, inclourem les despeses generals i el benefici industrial (obtenint el pressupost d'execució per contracta) i afegirem l'IVA corresponent per obtenir el pressupost base de licitació.

En aquest treball realitzem l'automatització de dos processos i la comunicació d'aquests amb un robot col·laboratiu. A més, realitzem també la programació d'un panell de control (SCADA) per tal que l'usuari pugui des d'aquesta controlar tot el procés.

Suposem que disposem dels processos a automatitzar (la cinta i les cintes) i la nostra feina és automatitzar-los. Perquè siga un procés completament automàtic hem decidit adquirir un robot que s'encarrega moure les peces des del magatzem a les cintes i recollir-la en finalitzar el procés. Per tant, el preu de la cinta transportadora i el procés de cintes de Fischer no l'hem d'incloure al pressupost però el robot sí. Inclourem també tot el que ens fa falta per a realitzar la programació, comptant també amb els PLCs utilitzats.

## 2.- Costos Unitaris

Els costos unitaris es calculen en euros per hora(€/h). A continuació definim els diferents costos unitaris, distingint entre mà d'obra i materials.

### 2.1.Mà d'obra

Al projecte intervé tres classes de personal:

- L'enginyer industrial, és qui s'encarrega de la realització del projecte, es troba especialitzat en automàtica i robòtica.
- Director del projecte, és qui dóna suport l'enginyer industrial i s'encarrega de la direcció i resolució de problemes.
- Tècnic, s'encarrega del manteniment dels recursos utilitzats, i reparació en cas de ser necessari.

#### 2.1.1 Enginyer industrial.

Descripció	Cost Unitari(€/h)
Enginyer industrial	40

Taula 5. Cost mà d'obra-Enginyer industrial

#### 2.1.2 Director del projecte

Descripció	Cost Unitari(€/h)
Director del projecte	60

Taula 6. Cost mà d'obra-Director del projecte

#### 2.1.3 Tècnic

Descripció	Cost Unitari(€/h)
Tècnic de laboratori	18

Taula 7. Cost mà d'obra-Tècnic

## 2.2.Material

Per calcular el cost unitari dels recursos materials emprats, ho fem mitjançant l'amortització d'aquests. Per a 225 dies útils de treball en jornada de huit hores, considerem una vida útil de 1800h/any.

L'amortització és el preu del recurs material dividit per la vida útil d'aquest:

$$\text{Cost d'amortització(€/h)} = \frac{\text{preu(€)}}{\text{vida útil(h)}}$$

Equació 1. Amortització

Tipus	Preu(€)	Vida útil(h)	Cost Unitari(€/h)
Recursos Físics			
Robot	20000	35000	0,57
PLC	5000	3000	1,67
Ordinador Personal	800	7200	0,11
Monitor Ordinador	200	5400	0,04
Torre Ordinador	600	5400	0,11
Software			
SoMachine	0	1800	0
LabView	1859	1800	1,03
Microsoft Office	69	1800	0,04

Taula 8. Cost material

*\*La llicència del SoMachine és gratis amb l'adquisició dels PLCs de Schneider Electric.*

### 3.- Preus unitaris descompostos

Per calcular els preus unitaris, descomposem el projecte en unitats d'obra i calculem el cost de cadascuna d'elles.

El cost total de la unitat d'obra és la suma del cost de la mà d'obra i dels materials.

#### 3.1.-Cinta ABB

##### 3.1.1 Mà d'obra

Activitat	Descripció	Cost unitari(€)	Rendiment	Unitat	Cost(€)
<b>1.1-1 Reunions</b>					
Reunió Inicial(2 hores)	Acord dels objectius i la forma d'aconseguir-los. Establir dates límit i dies de reunions.			1	200,00
	Enginyer industrial	40,00		2 h	80,00
	Director del projecte	60,00		2 h	120,00
Reunió Consulta(1 hora)	Reunions per comprovar com van les metes fixades i per realitzar consultes i resoldre problemes			3	300,00
	Enginyer industrial	40,00		1 h	40,00
	Director del projecte	60,00		1 h	60,00
Reunió Informativa(1 hora)	Indicacions al tècnic del què anem a fer i què necessitem que monte o realitze el manteniment			1	118,00
	Enginyer industrial	40,00		1 h	40,00
	Director del projecte	60,00		1 h	60,00
	Tècnic Laboratori	18,00		1 h	18,00
<b>SUBTOTAL</b>					<b>618,00</b>
<b>1.1-2 Programació</b>					
Cinta ABB	Realitzar el programa què ha d'executar la cinta tenint en compte els modes de funcionament, seguretat...			1	1600,00
	Enginyer Industrial	40,00		40 h	1600,00
<b>SUBTOTAL</b>					<b>1600,00</b>
<b>1.1-3 Comunicació</b>					
Cinta ABB	Comunicar mitjançant MODBUS amb el robot			1	800,00
	Enginyer Industrial	40,00		20 h	800,00
<b>SUBTOTAL</b>					<b>800,00</b>
<b>1.1-4 Manteniment</b>					
Cinta ABB	Realitzar el manteniment adequat, comprovar que es troben bé els cables corresponents, funcionament dels sensors...			1	36,00
	Tècnic laboratori	18,00		2 h	36,00
<b>SUBTOTAL</b>					<b>36,00</b>
<b>TOTAL(Mà d'obra)</b>					<b>3054,00</b>

Taula 9. Cost mà d'obra-Cinta ABB

### 3.1.2 Materials

Activitat	Descripció	Cost unitari(€/	Rendiment	Unitat	Cost(€)
1.2-1 Recursos físics					
	PLC	1,67	30,00	h	50,00
	Monitor Ordinador	0,04	40,00	h	1,48
	Torre Ordinador	0,11	40,00	h	4,44
SUBTOTAL					<b>55,93</b>
1.2-2 Software					
	SoMachine	0,00	40,00	h	0,00
SUBTOTAL					<b>0,00</b>
<b>TOTAL(Cost material)</b>					<b>55,93</b>

Taula 10. Cost material-Cinta ABB

Cost d'unitat d'obra, CintaABB 3109,93€.

## 3.2. Cintes Fischer

### 3.2.1 Mà d'obra

Activitat	Descripció	Cost unitari(€/h)	Rendiment	Unitat	Cost(€)
<b>2.1-1 Reunions</b>					
Reunió Inicial(2 hores)	Acord dels objectius i la forma d'aconseguir-los. Establir dates límit i dies de reunions.		1		200,00
	Enginyer industrial	40,00	2 h		80,00
	Director del projecte	60,00	2 h		120,00
Reunió Consulta(1 hora)	Reunions per comprovar com van les metes fixades i per realitzar consultes i resoldre problemes		4		400,00
	Enginyer industrial	40,00	1 h		40,00
	Director del projecte	60,00	1 h		60,00
Reunió Informativa(1 hora)	Indicacions al tècnic del què anem a fer i què necessitem que monte o realitzi el manteniment		1		118,00
	Enginyer industrial	40,00	1 h		40,00
	Director del projecte	60,00	1 h		60,00
	Tècnic Laboratori	18,00	1 h		18,00
<b>SUBTOTAL</b>					<b>718,00</b>
<b>2.1-2 Programació</b>					
Cintes Fischer	Realitzar el programa què han d'executar les cintes tenint en compte els modes de funcionament, seguretat...		1		2400,00
	Enginyer Industrial	40,00	60 h		2400,00
<b>SUBTOTAL</b>					<b>2400,00</b>
<b>2.1-3 Comunicació</b>					
Cintes Fischer	Comunicar mitjançant MODBUS amb el robot		1		800,00
	Enginyer Industrial	40,00	20 h		800,00
<b>SUBTOTAL</b>					<b>800,00</b>
<b>2.1-4 Manteniment</b>					
Cintes Fischer	Realitzar el manteniment adequat, comprovar que es troben bé els cabes corresponents, funcionament dels sensors...		1		54,00
	Tècnic laboratori	18,00	3 h		54,00
<b>SUBTOTAL</b>					<b>54,00</b>
<b>TOTAL(Mà d'obra)</b>					<b>3972,00</b>

Taula 11. Cost mà d'obra-Cintes Fischer

### 3.2.2 Materials

Activitat	Descripció	Cost unitari(€/h)	Rendiment	Unitat	Cost(€)
2.2-1 Recursos físics					
	PLC	1,67	30	h	50
	Monitor Ordinador	0,04	40	h	1,48
	Torre Ordinador	0,11	40	h	4,44
SUBTOTAL					<b>55,93</b>
2.2-2 Software					
	SoMachine	0	40	h	0
SUBTOTAL					<b>0</b>
<b>TOTAL(Material)</b>					<b>55,93</b>

Taula 12. Cost material-Cintes Fischer

Cost d'unitat d'obra, Cintes Fischer 4027,93€.

### 3.3. Robot

#### 3.3.1 Mà d'obra

Activitat	Descripció	Cost unitari(€/h)	Rendiment	Unitat	Cost(€)
<b>3.1-1 Reunions</b>					
Reunió Inicial(2 hores)	Acord dels objectius i la forma d'aconseguir-los. Establir dates límit i dies de reunions.			1	200,00
	Enginyer industrial	40,00		2 h	80,00
	Director del projecte	60,00		2 h	120,00
Reunió Consulta(1 hora)	Reunions per comprovar com van les metes fixades i per realitzar consultes i resoldre problemes			4	400,00
	Enginyer industrial	40,00		1 h	40,00
	Director del projecte	60,00		1 h	60,00
Reunió Informativa(1 hora)	Indicacions al tècnic del què anem a fer i què necessitem que monte o realitze el manteniment			1	118,00
	Enginyer industrial	40,00		1 h	40,00
	Director del projecte	60,00		1 h	60,00
	Tècnic Laboratori	18,00		1 h	18,00
SUBTOTAL					<b>718,00</b>
<b>3.1-2 Programació</b>					
Robot	Realitzar el programa què ha d'executar el robot tenint en compte els modes de funcionament, seguretat...			1	1400,00
	Enginyer Industrial	40,00		35 h	1400,00
SUBTOTAL					<b>1400,00</b>
<b>3.1-3 Comunicació</b>					
Robot	Comunicar mitjançant MODBUS amb la cinta ABB i les cintes Fischer			1	800,00
	Enginyer Industrial	40,00		20 h	800,00
SUBTOTAL					<b>800,00</b>
<b>3.1-4 Manteniment</b>					
Robot	Realitzar el manteniment adequat, canvi de la ferramenta del robot, ajust de la pressió de l'aire...			1	36,00
	Tècnic laboratori	18,00		2 h	36,00
SUBTOTAL					<b>36,00</b>
<b>TOTAL(Mà d'obra)</b>					<b>2954,00</b>

Taula 13. Cost mà d'obra-Robot

#### 3.3.2 Materials

Activitat	Descripció	Cost unitari(€/h)	Rendiment	Unitat	Cost(€)
<b>3.2-1 Recursos físics</b>					
	Robot	0,57		25 h	14,29
SUBTOTAL					<b>14,29</b>
<b>3.2-2 Software</b>					
SUBTOTAL					<b>0</b>
<b>TOTAL(Material)</b>					<b>14,29</b>

Taula 14. Cost material-Robot

Cost d'unitat d'obra, Robot 2968,29€.



### 3.4. SCADA

#### 3.4.1 Mà d'obra

Activitat	Descripció	Cost unitari(€)	Rendiment	Unitat	Cost(€)
4.1-1 Reunions					
Reunió Inicial(1 hores)	Acord del que es vol representar i com		1		100,00
	Enginyer industrial	40,00	1 h		40,00
	Director del projecte	60,00	1 h		60,00
Reunió Consulta(1 hora)	Reunió per resoldre dubtes i		1		100,00
	Enginyer industrial	40,00	1 h		40,00
	Director del projecte	60,00	1 h		60,00
SUBTOTAL					<b>200,00</b>
SCADA	Realitzar la programació del		1		1000,00
	Enginyer Industrial	40,00	25 h		1000,00
SUBTOTAL					<b>1000,00</b>
<b>TOTAL(Mà d'obra)</b>					<b>1200,00</b>

Taula 15. Cost mà d'obra-SCADA

#### 3.4.2 Material

Activitat	Descripció	Cost unitari	Rendiment	Unitat	Cost(€)
4.2-1 Recursos físics					
	Monitor ordinador	0,037	25	h	0,93
	Torre ordinador	0,11	25	h	2,78
SUBTOTAL					<b>3,70</b>
	LabView	1,03	25	h	25,82
SUBTOTAL					<b>25,82</b>
<b>TOTAL(Material)</b>					<b>29,52</b>

Taula 16. Cost material-SCADA

Cost d'unitat d'obra, SCADA 1229,52€.

## 3.5. Redacció

### 3.5.1 Mà d'obra

Activitat	Descripció	Cost unitari(€/h)	Rendiment	Unitat	Cost(€)
5.1-1 Reunions					
Reunió Inicial(1 hores)	Establiment dels punts de la memòria		1		100,00
	Enginyer industrial	40,00	1 h		40,00
	Director del projecte	60,00	1 h		60,00
Reunió Consulta(1 hora)	Reunió per corregir el document i millorar-lo		3		300,00
	Enginyer industrial	40,00	1 h		40,00
	Director del projecte	60,00	1 h		60,00
SUBTOTAL					<b>400,00</b>
5.1-2 Redacció					
	Enginyer Industrial	40,00	60	h	2400,00
SUBTOTAL					<b>2400,00</b>
<b>TOTAL(Mà d'obra)</b>					<b>2800,00</b>

Taula 17. Cost mà d'obra-Redacció

### 3.5.2 Materials

Activitat	Descripció	Cost unitari(€/h)	Rendiment	Unitat	Cost(€)
5.2-1 Recursos físics					
	Ordinador personal	0,11	60	h	6,67
	Impressió	50	1		50
SUBTOTAL					<b>56,67</b>
5.2-2 Software					
	Microsoft Office	0,04	60		2,3
SUBTOTAL					<b>2,30</b>
<b>TOTAL(Material)</b>					<b>58,97</b>

Taula 18. Cost material-Redacció

Cost d'unitat d'obra, Redacció 2858,97€.

## 4.- Pressupost total d'execució

És la suma del conjunt dels preus descompostos (de les diferents unitats d'obra).

Pressupost parcial	Cost(€)
Cinta ABB	3109,9259
Cintes Fischer	3972,00
Robot	2968,29
SCADA	1229,52
Redacció memoria	2858,97
<b>TOTAL(€)</b>	<b>14138,70</b>

Taula 19. Pressupost total d'execució

**El cost total d'execució és 14138,70€**

## 5.- Pressupost total d'execució per contracta

És el pressupost total d'execució afegint les despeses i el benefici industrial.

Pressupost total d'execució per contracta	Cost(€)
Pressupost total d'execució	14138,70
13% de despeses	1838,03
6% de benefici industrial	848,32
<b>TOTAL</b>	<b>16825,05</b>

Taula 20. Pressupost total d'execució per contracta

**El pressupost total d'execució per contracta és 16825,05€**

## 6.- Pressupost base de licitació

El pressupost base de licitació és el pressupost total d'execució per contracta tenint en compte l'IVA.

Pressupost base de licitació	Cost(€)
Pressupost total d'execució per contracta	16825,05
21% d'IVA	3533,26
<b>TOTAL(€)</b>	<b>20358,32</b>

Taula 21. Pressupost base de licitació

El pressupost total final del projecte, pressupost base de licitació, és:

**VINT MIL TRES-CENTS CINQUANTA-HUIT EUROS AMB TRENTA-DOS CÈNTIMS.**

# ANNEX

---

## ÍNDEX DE CONTINGUTS

<b>1.- RELACIÓ VARIABLES .....</b>	<b>86</b>
1.1.Cinta ABB .....	86
1.2.Cintes Fischer .....	87

## ÍNDEX DE TAULES

Taula 22. Relació variables grafcet i SFC-Cinta ABB .....	86
Taula 23. Relació variables grafcet i SFC-Cintes Fischer .....	87

## 1.- Relació Variables

Com que als grafquets hem donat un nom més explicatiu per a les variables, tant per a les accions com per a les transicions, aquest nom o coincideix amb el nom utilitzat al programa creat al SoMachine. A més, al manual del programador hem explicat les variables dels programes però corresponent als SFC i no als grafquets, per tant, per evitar confusions, a continuació les relacionem.

### 1.1.Cinta ABB

Variable GRAFCET	Variable SFC
Auto	Auto
CintaABB	Q8 i MCinta
DeixarPRobot	DeixarPRobot
Parada	BotoParada
SensorFinalCinta	I3
Emergencia	Emergencia
Emergencia	Emergencia0
RobotArreplegaPeça	RobotArreplega
PeçaDeixada	Rob0
BotoManual	BotoManual

Taula 22. Relació variables grafquet i SFC-Cinta ABB

## 1.2.Cintes Fischer

Variables GRAFCET	Variables SFC
Auto	Auto
BotoManual	BotoManual
SensorInici	I7
SensorFCinta1	I5
Cinta1	Q5 i Alimentacio
Parada	BotoParada
Empenyedor1Avant	Q1 i E11
Empenyedor1Arrere	Q2 i E12
Empenyedor2Avant	Q3 i E21
Empenyedor2Arrere	Q4 i E22
Cinta2	Q7 i CF
Cinta3	Q9 i CT
FinalCarrera11	I1
FinalCarrera12	I2
FinalCarrera21	I3
FinalCarrera22	I4
SensorFresa	I6
SensorTrepant	I8
Fresa	Q7 i MotorFresa
Trepant	Q9 i Motor Trepant
SensorFinal	I9
Cinta4	Q10 i CE
PFi	PFi
Robot	Robot[0]
Emergencia	Emergencia

Taula 23. Relació variables grafcet i SFC-Cintes Fischer