

Grau en Enginyeria Electrònica Industrial i Automàtica

DISSENY I PROGRAMACIÓ D'UNA CÈL·LULA DE FABRICACIÓ FLEXIBLE

Memòria

**DANIEL DE CASTRO ALVARADO
PONENT: JOAN TRIADÓ I AYMERICH**

PRIMAVERA 2020

AGRAÏMENTS

Al meu tutor de Joan Triadó Aymerich,
per la paciència, ajuda i comprensió durant el desenvolupament del treball.

A la meva família i amics,
per donar-me els ànims necessaris per continuar endavant.

I a la meva parella Alba,
A la que espero poder tornar-li el favor
per recolzar-me en els moments més complicats.

Resum

En la documentació següent es definiran els passos necessaris per realitzar un projecte d'automatització complex, tractant el procés de disseny del sistema i posterior programació.

Els elements funcionals del disseny són els propis de la cèl·lula robotitzada del laboratori 4 del Tecnocampus. Al mateix projecte, s'inclourà també possibles millores de funcionament, així com modificacions per adequar la cèl·lula a la normativa vigent i a un ambient de treball real.

Resumen

En la siguiente documentación se definirán los pasos necesarios para realizar un proyecto de automatización complejo, tratando su proceso de diseño i posterior programación.

Los elementos funcionales del diseño son los propios de la célula robotizada del laboratorio 4 del Tecnocampus. En el mismo proyecto, se incluirán también posibles mejoras de funcionamiento, así como modificaciones para adecuar la célula a la normativa vigente y a un ambiente de trabajo real.

Abstract

The following documentation will define the steps necessary to carry out a complex automation project, dealing with its design process and subsequent programming.

The functional elements of the design are those of the robotic cell of Tecnocampus laboratory 4. In the same project, possible performance improvements will be included, as well as modifications to adapt the cell to current regulations and a real work environment.

ÍNDEX

Índex de figures	IX
Índex de taules	XI
Glossari de termes	XIII
1. Objectius	15
1.1. Propòsit	15
1.2. Finalitat	15
1.3. Objecte	15
1.4. Abast	15
1.5. Context en les línies de recerca i transferència de coneixement del Tecnocampus	16
2. Estat de l'art	17
2.1. L'automatització en el marc actual	17
2.1.1 Impacte global en l'empresa	17
2.1.2 Indústria 4.0.....	18
2.2. Normativa.....	20
2.2.1 Requisits per a l'adquisició d'equips de treball.....	21
2.2.2 Documentació de les màquines	22
2.2.3 Manteniment.....	23
2.3. Necessitat d'informació.....	24
2.4. Coneixement de les eines	25
2.4.1. Grafcet i ladder.....	25
2.4.2. RobotStudio.....	30
2.4.3. SCADA i FactoryTalk View	33
3. Especificacions tècniques i plantejament del projecte	35
3.1. Composició de la cèl·lula de fabricació	35
3.1.1. Característiques del ABB IRB 120 i Flex Pendant	36
3.1.2. Característiques del FANUC LR Mate 200iD/4S	38
3.2 Solució plantejada	39
4. Metodologia	43
5. Lògica de funcionament.....	47
5.1. Autòmat CompactLogix L32E	48
5.1.1. Interfase entre l'autòmat i la cèl·lula.....	48
5.1.2. Interfase entre l'autòmat i els robots	48
5.1.3. GRAFCET de l'autòmat.....	50

5.2. ABB IRB 120.....	56
5.2.1. Accions de l'autòmat i del programa de l'ABB IRB 120	56
5.2.2. GRAFCET del robot ABB IRB 120.....	59
5.3. FANUC LR Mate 200 iC/4S.....	60
5.3.1. Accions de l'autòmat i del programa del FANUC LR Mate 200.....	60
5.3.2. GRAFCET del robot FANUC LR Mate 200 Ic/4S	62
6. Disseny de la simulació	63
6.1. Programació de la simulació	64
6.2. Elements de la simulació.....	66
6.2.1. Dispensador.....	67
6.2.2. Base de detecció	68
6.2.3. Caixa de peces.....	69
6.2.4. Carro de transport.....	70
6.2.5. Disc Rotatori	72
6.2.6. Trepant	73
6.2.7. Verificador	75
6.2.8. Eines.....	76
6.3. Lògica d'estació i connexions.....	78
7. Anàlisi de viabilitat	81
7.1. Viabilitat tècnica	81
7.1.1. Pla de contingència.....	81
7.2. Viabilitat econòmica	82
7.2.1. Pressupost total del projecte.....	83
7.3. Viabilitat mediambiental.....	83
8. Planificació del projecte.....	85
8.1. Avantprojecte	85
8.1.1. Llistat d'activitats.....	85
8.1.2. Descripció de les activitats.....	86
8.1.3. Diagrama de Gantt de l'avantprojecte.....	87
8.2. Enginyeria de detall.....	88
8.2.1. Llistat d'activitats.....	88
8.2.2. Descripció de les activitats.....	89
8.2.3. Diagrama de Gantt de l'enginyeria de detall.....	91
8.2.4. Pla de contingència.....	92
9. Conclusions	95
10. Bibliografia i referències.....	97

Índex de figures

Figura 2.1: Instal·lacions anuals de robots industrials [Font: IFR]	17
Figura 2.2: Marcatge CE [Font: Directiva CE]	22
Figura 2.3: Elements bàsics d'un grafcet [Font: Automatització I (ESUPT)].....	26
Figura 2.4: Tipus de direccionaments dels grafkets [Font: Automatització I (ESUPT)]	26
Figura 2.5: Accionament condicionat [Font: ESUPT]	27
Figura 2.6: Exemple de ladder [Font: Automatització I (ESUPT)].....	29
Figura 2.7: Visió general del programa RobotStudio [Font: pròpia]	30
Figura 2.8: Possibilitats que ofereixen les categories de RobotStudio [Font: pròpia] ...	31
Figura 2.9: Exemple de programa RAPID [Font: pròpia].....	32
Figura 2.10: Exemple de pantalla generada amb FactoryTalk View [Font: Automatització I (ESUPT)].....	34
Figura 3.1: Localització dels diferents components que formen la cèl·lula de fabricació [Font: pròpia].....	35
Figura 3.2: Robot IRB 120 [Font: ABB].....	36
Figura 3.3: Consola Flex Pendant [Font: ABB]	37
Figura 3.4: Panell de control de la consola Flex Pendant [Font: ABB]	37
Figura 3.5: Robot LR Mate 200Id/4S [Font: FANUC]	38
Figura 3.6: Consola iPendant [Font: pròpia]	38
Figura 3.7: Botons de la consola iPendant [Font: pròpia]	39
Figura 4.1: Esquemàtic de la metodologia emprada al treball [Font: pròpia]	46
Figura 5.1: GRAFCET de l'autòmat [Font: pròpia]	50
Figura 5.2: Estat 2 del GRAFCET de l'autòmat. [Font: pròpia]	51
Figura 5.3: Estat 3 del GRAFCET de l'autòmat. [Font: pròpia]	51
Figura 5.4: Estat 4 del GRAFCET de l'autòmat. [Font: pròpia]	52
Figura 5.5: Estats 8 i 9 del GRAFCET de l'autòmat. [Font: pròpia]	52
Figura 5.6: Estats 13, 14, 15 i 16 del GRAFCET de l'autòmat. [Font: pròpia]	55
Figura 5.7: GRAFCET del robot ABB. [Font: pròpia]	59
Figura 5.8: GRAFCET del robot FANUC. [Font: pròpia]	62
Figura 6.1: Visió completa de la cèl·lula de fabricació de la simulació. [Font: pròpia]	63
Figura 6.2: Finestra que apareix al crear una nova senyal. [Font: pròpia]	65
Figura 6.3: Peces disponibles a la simulació. [Font: pròpia].....	66

Figura 6.4: Detall del dispensador de la simulació. [Font: pròpia]	67
Figura 6.5: Lògica interna del SC_Feeder. [Font: pròpia]	67
Figura 6.6: Detall de la base de detecció. [Font: pròpia].....	68
Figura 6.7: Lògica interna del SC_Sensor_Peça. [Font: pròpia].....	69
Figura 6.8: Detall del contenidor. [Font: pròpia]	69
Figura 6.9: Lògica interna del SC_Caixa [Font: pròpia].....	70
Figura 6.10: Detall del carro de transport. [Font: pròpia]	70
Figura 6.11: Lògica interna del SC_Carro. [Font: pròpia]	71
Figura 6.12: Detall del disc rotatori. [Font: pròpia]	72
Figura 6.13: Lògica interna del SC_Disc. [Font: pròpia].....	73
Figura 6.14: Lògica interna del SC_Trepant. [Font: pròpia].....	74
Figura 6.15: Lògica interna del SC_Verificador. [Font: pròpia].....	75
Figura 6.16: PinçaABB (esquerra) i PinçaFANUC (dreta). [Font: pròpia].....	76
Figura 6.17: Lògica interna del SC_Tool_ABB. [Font: pròpia].....	77
Figura 6.18: Lògica d'estació de la simulació. [Font: pròpia].....	78
Figura 8.1: Diagrama de Gantt de l'avantprojecte [Font: pròpia].....	87
Figura 8.2: Diagrama de Gantt de l'enginyeria de detall [Font: pròpia].....	91
Figura 8.3: Diagrama de Gantt del pla de contingència [Font: pròpia].....	94

Índex de taules

Taula 2.1: Estructura i ordre d'un ladder [Font: Automatització I (ESUPT)].....	28
Taula 3.1: Codi i descripció dels components de la cèl·lula [Font: pròpia].....	36
Taula 3.2: Codi i descripció d'elements de la consola Flex Pendant [Font: pròpia].....	37
Taula 3.3: Codi i descripció d'elements al panell de la consola Flex Pendant [Font: pròpia]	37
Taula 5.1: Interfase entre l'autòmat i la cèl·lula [Font: pròpia]	48
Taula 5.2: Interfase de sortides de l'autòmat cap al robot ABB [Font: pròpia]	49
Taula 5.3: : Interfase d'entrades de l'autòmat des del robot ABB [Font: pròpia].....	49
Taula 5.4: : Interfase de sortides de l'autòmat cap al robot FANUC [Font: pròpia].....	49
Taula 5.5: Interfase d'entrades de l'autòmat des del robot ABB [Font: pròpia].....	50
Taula 5.6: Taula lògica de les connexions amb Peça_noDrill. [Font: pròpia]	55
Taula 5.7: Comunicacions al moviment 'Inici' del robot ABB. [Font: pròpia]	56
Taula 5.8: Comunicacions al moviment 'Disp_Detei' del robot ABB. [Font: pròpia] ..	57
Taula 5.9: Comunicacions al moviment 'Dete_Cont' del robot ABB. [Font: pròpia] ...	57
Taula 5.10: Comunicacions al moviment 'Dete_Carro' del robot ABB. [Font: pròpia]	58
Taula 5.11: Comunicacions al moviment 'Carro_Magatzem' del robot ABB. [Font: pròpia].....	58
Taula 5.12: Comunicacions al moviment 'Inici' del robot FANUC. [Font: pròpia].....	60
Taula 5.13: Comunicacions al moviment 'pos1_Disc' del robot FANUC. [Font: pròpia]	60
Taula 5.14: Comunicacions al moviment 'pos2_Disc' del robot FANUC. [Font: pròpia]	61
Taula 5.15: Comunicacions al moviment 'Disc_Carroi' del robot FANUC. [Font: pròpia]	61
Taula 5.16: Comunicacions al moviment 'Disc_Cont' del robot FANUC. [Font: pròpia]	61
Taula 6.1: Lògica de detecció de la peça segons l'activació dels sensors. [Font: pròpia]	68
Taula 6.2: Comunicacions de sortida del controlador ABB a components de l'estació. [Font: pròpia].....	78
Taula 6.3: Comunicacions d'arribada al controlador ABB des de components de l'estació. [Font: pròpia].....	79

Taula 6.4: Comunicacions de sortida del controlador FANUC a components de l'estació. [Font: pròpia].....	79
Taula 6.5: Comunicacions d'arribada al controlador FANUC des de components de l'estació. [Font: pròpia]	80
Taula 6.6: Comunicacions entre components de l'estació. [Font: pròpia]	80
Taula 7.1: Pressupost total del projecte [Font: pròpia].....	83
Taula 8.1: Activitats relacionades amb l'avantprojecte [Font: pròpia]	86
Taula 8.2: Descripció d'activitats relacionades amb l'avantprojecte [Font: pròpia].....	86
Taula 8.3: Activitats relacionades amb l'enginyeria de detall [Font: pròpia]	89
Taula 8.4: Descripció d'activitats relacionades amb l'enginyeria de detall [Font: pròpia]	90
Taula 8.5: Nova planificació degut al pla de contingència [Font: pròpia]	93

Glossari de termes

<i>BigData</i>	Possibilitat d'explotar una gran quantitat de dades per crear nous serveis
<i>CE</i>	Conformitat Europea/ 'Conformité Européenne'
<i>Ethernet</i>	Tecnologia que connecta xarxes d'àrea local (LAN) cablejades i permet que el dispositiu es comuniqui entre si a través d'un protocol que és el llenguatge de xarxa comú
<i>ERP</i>	Manufacturing Execution Systems
<i>ESUPT</i>	Escuela Superior Politécnica del TecnoCampus
<i>FI4</i>	Fabricació Intel·ligent i Innovació Industrial
<i>Fieldbus</i>	Protocols industrials de xarxes informàtiques utilitzats per a xarxes de control industrial en temps real, estandarditzat com a norma IEC 61158
<i>Grafcet</i>	És un mètode gràfic i estructurat per a la descripció del comportament dels automatismes seqüencials mitjançant diagrames funcionals
<i>IFR</i>	International Federation of Robotics
<i>Iot</i>	Internet of Things
<i>KAREL</i>	Llenguatge de programació pels robots FANUC
<i>kW</i>	Kilowatts de potència
<i>Ladder</i>	És un llenguatge de programació gràfic molt popular
<i>MES</i>	Enterprise Resource Planning
<i>OEE</i>	Overall Equipment Effectiveness
<i>OPC UA</i>	Protocol de comunicació de Microsoft d'arquitectura unificada
<i>RAPID</i>	Llenguatge de programació d'ABB similar al llenguatge C
<i>RD</i>	Real Decret
<i>SCADA</i>	Supervisory Control And Data Acquisition
<i>Security/SSL</i>	Protocol de seguretat (Secure Sockets Layer)
<i>TFG</i>	Treball Final de Grau
<i>Usb</i>	Universal Serial Bus

1. Objectius

1.1. Propòsit

El principal objectiu del treball és realitzar el disseny d'una aplicació i programació d'una cèl·lula de fabricació flexible.

Al llarg del mateix, s'explicaran els conceptes necessaris per poder aplicar-los en casos similars, dividint el projecte en dues parts diferenciades; la simulació i el cas real. Per ambdues situacions, es tindrà com a finalitat realitzar un disseny i una programació (pròpies i diferenciades de cadascuna).

Per realitzar el treball, és necessari interioritzar el funcionament dels programes amb els quals es treballarà, alguns conceptes bàsics d'automatització i terminologia, normatives, així com conèixer l'entorn i els elements que formen la cèl·lula de treball. També es valoraran aquests aspectes a fi de millorar la comprensió general del projecte.

1.2. Finalitat

La finalitat del projecte és l'estudi del disseny i programació completa d'una cèl·lula de treball robotitzada, prenent com a model la que disposa l'ESUPT, de forma que el procés sigui comprensible, per així facilitar el procés d'aprenentatge dels dos braços robot; *ABB IRB 120* i *FANUC LR Mate 200ID 4S* i del controlador de la cèl·lula, així com proposar millores del sistema actual per a millorar la docència a l'espai del laboratori.

1.3. Objecte

L'objecte del projecte és la generació d'un sistema real automatitzat, duent a terme el seu disseny i programació, així com la millora del funcionament actual per millorar la docència a la cèl·lula o permetre demostracions.

1.4. Abast

L'abast del projecte inclou el desenvolupament de graficats i ladders del controlador, així com una simulació de funcionament i la programació necessària per els dos robots antropomòrfics.

Tot el projecte es redactarà de manera que l'estudi del funcionament de la cèl·lula per un cas concret, sigui extrapolable a futurs projectes educacionals amb el mateix entorn de treball.

El projecte es fonamenta en la utilització i manipulació d'un conjunt automatitzat ja establert sobre el qual es dissenyarà i programarà, per tant, no s'inclou el disseny físic de la cèl·lula, així com l'elecció de components, muntatge o el seu pressupost total.

1.5. Context en les línies de recerca i transferència de coneixement del Tecnocampus

La realització d'aquest projecte està alineat amb la transferència de coneixement de les línies de recerca FI4 del Tecnocampus respecte la fabricació intel·ligent i la innovació industrial, tractant conceptes com la indústria 4.0 i la robòtica avançada, entre altres.

Durant el seu transcurs, i a fi de poder proposar una solució de disseny, es treballarà amb softwares emprats a les assignatures anteriorment esmentades, com són *RSLogix 5000* i *FactoryTalk View* (propietat d'Allen Bradley-Rockwell Automation) i *RobotStudio* (propietat d'ABB).

2. Estat de l'art

L'automatització és la tecnologia mitjançant la qual es poden realitzar processos de manera autònoma. En el cas del monitoratge i processos de fabricació, s'utilitzen els robots i els controladors.

Abans de realitzar una automatització amb robots, és necessari plantejar si aquesta és necessària des del punt de vista dels beneficis (reducció de personal, augment de productivitat, disminució de l'accidentalitat, per una millora comercial...) i des d'una perspectiva econòmica; s'hi realitza una gran inversió amb l'adquisició del robot o màquina, així com els utilatges i equipament necessaris, la instal·lació i adequació de l'espai de treball, així com també involucra uns costos (implantació, configuració, formació, manteniment, socials, financers...).

Un dels punts clau en la utilització d'aquesta tecnologia és saber que s'ha de dur a terme un re-disseny dels procediments de treball, i això comporta una reenginyeria (així com una elevada inversió).

2.1. L'automatització en el marc actual

Avui en dia, els robots, sensors, actuadors i tecnologies de control continuen millorant constantment i evolucionant. Moltes de les tasques necessàries d'alta precisió o habilitat són exercides per autòmats; els processos de fresatge, tornejat i soldadura en són bons exemples. Els robots industrials i màquines són habitualment emprats per tasques d'assemblatge, manufactura o pintura de productes.

2.1.1 Impacte global en l'empresa

La IFR (acrònim de *International Federation of Robotics*) realitza un bolletí anual d'estadístiques varies respecte els robots industrials, dels quals es pot obtenir informació significativa del creixement que estan experimentant internacionalment.

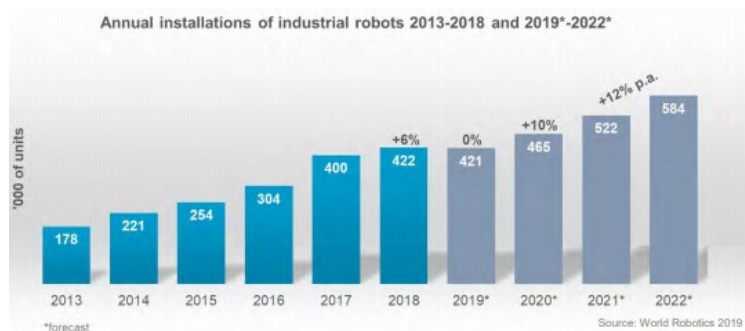


Figura 2.1: Instal·lacions anuals de robots industrials [Font: IFR]

- La previsió d'instal·lacions al 2020 fins al 2022 és creixent, fins arribar a un 12% més d'instal·lacions respecte l'any anterior.
- La regió on major creixement està experimentant l'automatització industrial és Asia, trobant com a principals clients China, Japó i Corea (la qual té l'indústria amb major proporció màquina-persona, juntament amb Singapur).
- Les principals indústries són les d'automoció, les elèctriques/electròniques i la del metall.
- Els robots col·laboratius són encara de nínxol (un 3-4% respecte les vendes de robots industrials tradicionals).
- La estadística indica que per el 2022, pràcticament s'arribarà als 4 milions de robots industrials a les factories arreu del món.

Pel que respecta a Espanya, el país ocupa actualment la novena posició al ranking europeu d'automatització industrial.

Al 2018, Espanya disposava de 168 robots instal·lats per cada 10.000 treballadors en indústries manufactureres (un 7% més que a l'any anterior). La primera de la classificació, Alemanya, es situava a la mateixa data amb 338 robots per cada 10.000 treballadors, el doble.

Tot i així, es supera la mitja mundial de 99 robots per cada 10.000 treballadors i companyies punteres del sector, com *ABB Robotics* estan invertint en espais de demostració i proves per ajudar al desenvolupament de noves solucions i tecnologies digitals que permetin a les empreses espanyoles competir amb més eficàcia en l'economia global. [1]

2.1.2 Indústria 4.0

El major repte del segle XXI és aconseguir que els robots i les màquines siguin suficientment intel·ligents per aprendre a realitzar tasques automàticament i adaptar-se a condicions de funcionament imprevistes o errors de manera robusta i predicible, sense necessitat d'orientació humana, instruccions o programació.

Aquest imminent pas en l'evolució de l'automatització ve precedit pel continu desenvolupament de tecnologies com IoT o BigData a l'empresa, i la seva integració amb els robots per controlar el procés productiu, donant pas a l'indústria 4.0.

Un dels punts vitals de l'indústria 4.0 és la digitalització de l'empresa, en el que l'IoT (*Internet of Things*) té un rol rellevant permetent obtenir informació del procés de producció i afegir-li valor per l'empresa, podent realitzar produccions de manera eficient.

La digitalització també involucra el control i organització de tot el procés de la cadena de valor.

La organització del flux d'informació té forma piramidal, amb diferents nivells ascendents (de menys a més jerarquia):

- Nivell 0: Sensors i actuadors
- Nivell 1: Nivell de control
- Nivell 2: MES/ERP
- Nivell 3: Cloud (IoT/Big Data)

Al primer nivell, es sensoritza el procés de producció a cada màquina i s'extreu dades importants pròpies de cada procés. Si s'escau, es pot arribar a monitorar la dada directament des del espai de treball amb un SCADA programat per veure índex relatiu a la realització del procés. Un dels més importants és l'OEE (*Overall Equipment Effectiveness*), una fórmula matemàtica que mesura la disponibilitat, rendiment i qualitat del procés de fabricació.

Les dades normalment s'envien al següent nivell mitjançant comunicació Ethernet, Fieldbus... on són rebuts per una computadora que recol·lecta les dades i les transforma en informació rellevant per el seu monitoratge.

La comunicació entre la resta de nivells de major jerarquia utilitzen un mateix criteri de comunicació; ja sigui OPC UA o Security/SSL. Es tracten de protocols de comunicació pensats per comunicar dades d'equips industrials que tenen com a objectiu comunicar-se amb totes les aplicacions de l'empresa a través de totes les capes empresarials.

Al nivell 2, s'hi troben els softwares de gestió de la producció; el MES (*Enterprise Resource Planning*) i l'ERP (*Manufacturing Execution Systems*). Aquests sistemes guarden i processen la informació en temps real a la planta. En estar connectades al núvol, l'accés a la informació és ràpid i eficient.

La indústria 4.0 destaca respecte la indústria convencional i és una eina molt potent en un mercat tant competitiu i àgil com el global. Tot i aquesta evidència, a Espanya, a l'any

2017, es va impulsar una iniciativa des de el govern anomenada 'Indústria Connectada 4.0' oferint ajudes per poder fer una digitalització de l'empresa, de la qual només es va exhaurir un 30% del pressupost inicial. [2]

2.2. Normativa

A l'hora de realitzar un projecte d'enginyeria d'automatització, s'ha de tenir en compte les normatives imposades per la legislació vigent per a màquines del tipus a dissenyar.

Per evitar que les normatives siguin interpretables o no suficientment específiques, el *REIAL DECRET 1644/2008, del 10 de Octubre* [3] dona una definició concisa del terme <<màquina>>:

Conjunt de parts o components vinculats entre si, dels quals almenys un és mòbil, associats per a una aplicació determinada, proveït o destinat a estar proveït d'un sistema d'accionament diferent de la força humana o animal, aplicada directament.

Conjunt com l'indicat en el primer guió, al qual només li faltin els elements de connexió a les fonts d'energia i moviment.

Conjunt com els indicats en els guions primer i segon, preparat per a la seva instal·lació que solament pugui funcionar previ muntatge sobre un mitjà de transport o instal·lat en un edifici o una estructura.

En el cas a presentar, seguidament es referencien les diferents normatives que s'empraran al projecte, juntament con un compendi dels punts amb més pes per el correcte desenvolupament del projecte; en trobem de diferents tipus i aplicacions:

- Requisits per a l'adquisició d'equips de treball (maquinaria i instal·lacions)
- Funcionament dels equips de treball
- Documentació de les màquines i aplicacions a dur a terme
- Retolació, senyals i advertències en màquines
- Responsabilitats
- Manteniment

2.2.1 Requisits per a l'adquisició d'equips de treball

La legislació específica a complir que s'aplica a aquests punts depèn de la normativa estatal i la directiva europea:

- *Reial Decret 1215/1997, del 18 de Juliol*, pel qual s'estableixen les disposicions mínimes de seguretat i salut per a la utilització pels treballadors dels equips de treball. [4]

En aquest decret, es responsabilitza a l'empresari, independentment d'altres certificacions, de seguir els següents requisits per a l'ús correcte dels equips de treball (inclosos a l'Annex I del RD):

- Els òrgans d'accionament d'un equip de treball que tinguin incidència en la seguretat hauran de ser clarament visibles i identificables, així com senyalitzats quan correspongui. També hauran d'estar situats fora de zones perilloses.
 - La posada en marxa d'un equip de treball només es pot efectuar mitjançant una acció voluntària sobre un òrgan d'accionament previst per tal efecte. El mateix cas es aplicable a una posta en marxa després d'una aturada, o per tal d'introduir un canvi en les condicions de funcionament (velocitat, pressió...).
 - L'equip de treball ha d'estar correctament il·luminat, ha de disposar de senyals d'alarma fàcilment comprensibles i les senyalitzacions necessàries per evitar riscos innecessaris.
 - Tot equip de treball ha d'estar adequat per a protegir els treballadors exposats contra el risc de contacte directe o indirecte de l'electricitat i incendis, així com de les proteccions necessàries en cas de riscos per soroll, vibracions o radiacions.
- *REIAL DECRET 1644/2008, del 10 de Octubre*, pel qual s'estableixen les normes per a la comercialització i posada en servei de les màquines.

- *Directiva 2006/42/CE*

En aquests dos últims punts, es declara que el fabricant de la màquina, abans de la comercialització o posada en servei, ha de certificar i 'marcar' la placa identificaria de la màquina amb les inicials CE ("Conformité Européenne"), així com el nom del fabricant, any de fabricació, tipus i número de fabricació i la seva potencia en kW, com a senyal de conformitat amb la legislació obligatòria vigent.

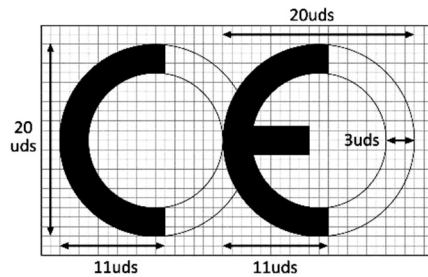


Figura 2.2: Marcatge CE [Font: Directiva CE]

Abans d'expedir la declaració CE, es indispensable que el fabricant elabori un expedient tècnic de construcció.

Encara sent un marcatge europeu per certs productes industrials, en cas de fabricació externa a la Comunitat Europea, el distribuïdor té l'obligació de certificar-lo.

2.2.2 Documentació de les màquines

S'ha d'exigir sempre al fabricant o venedor el manual d'instruccions de la màquina, redactat en la llengua original del país originari, així com una còpia en la llengua del país on serà emprada.

Aquest manual ha d'incloure els esquemes i plànols necessaris per un correcte funcionament, incloent:

- Condicions d'utilització.
- Instruccions per efectuar sense risc les tasques principals d'instal·lació, muntatge, funcionament, manteniment, desmuntatge, conservació i reparació.
- Característiques i contraindicacions d'ús.
- Plànols i esquemes en matèria de seguretat.

Tal com s'ha exposat a l'apartat anterior, tota màquina subjecta al RD 16444/2008 ha de incloure en el manual una declaració de conformitat 'CE' per part del fabricant.

En aquesta fulla de conformitat, ha de figurar com a mínim:

- Raó social i direcció completa del fabricant.

- Nom i direcció de la persona a càrrec de la realització de l'expedient tècnic.
- Descripció e identificació de la màquina (afegint la seva denominació genèrica, funció, model, número de sèrie...).
- Lloc i data de la declaració de conformitat.
- Nom, direcció i número d'identificació de l'organisme que va portar a terme l'examen CE.
- Un paràgraf en el qual s'esmeni explícitament el compliment de la *Directiva 2006/42/CE*, així com un altre per declarar l'adequació de la màquina a altres normatives comunitàries o estatals.

2.2.3 Manteniment

Inclòs al manual de la màquina, ha d'haver-hi un apartat relatiu al seu manteniment, en el qual es tracti:

- Conservació de la màquina:

Els punts de regulació, greixatge i conservació han d'estar situats fora de zones perilloses, permetent realitzar operacions relatives al manteniment amb la màquina aturada de manera segura.

En el cas de màquines automatitzades, el fabricant ha de projectar un dispositiu de connexió que permeti muntar un equip de diagnòstic per a la localització d'averies.

Continuant amb el cas d'una màquina automatitzada, és imprescindible que els elements que més pateixen el desgast o que es puguin deteriorar en cas d'un incident i, per tant, sigui necessari substituir-los amb certa freqüència, puguin desmuntar-se i tornar a muntar-se de forma senzilla i amb total seguretat. La forma de realitzar aquesta operació ha d'estar clarament definida pel constructor.

- Medis d'accés al lloc de treball o punts d'intervenció:

Si escau, el fabricant ha de proporcionar els medis necessaris que permetin arribar amb seguretat a totes les posicions de les operacions de producció, reglatge i manteniment.

- Separació de les fonts d'energia:

La màquina ha d'estar proveïda de dispositius que permetin separar cadascuna de les seves fonts d'energia. Aquests dispositius han de ser identificables, així com

permetre el seu bloqueig en cas de que la connexió de la font pugui posar en perill a les persones circumdants.

- Intervenció de l'operador:

Les màquines han d'estar dissenyades per limitar la intervenció d'operadors, i si no hi ha aquesta possibilitat, que la seva intervenció sigui segura i es doni facilitats.

- Neteja de les parts interiors:

El disseny i fabricació de la màquina ha de realitzar-se de forma que resulti possible netejar les parts interiors de la mateixa amb els menors riscos.

2.3. Necessitat d'informació

Al estar la cèl·lula robòtica en un centre educacional com l'ESUPT, s'ha investigat si hi havia algun treball o documentació vinculat amb ella que aportés coneixements addicionals per aplicar en aquest projecte.

Són els casos de les següents tesines:

- **Castillo Gutiérrez, Cristian. Disseny d'una cèl·lula de treball robotitzada que utilitzarà un robot industrial ASEA IRB 120 (2011) [6]**

En el document es pot trobar tota la informació necessària per a la posta en marxa d'una cèl·lula robòtica tant a nivell de software (mitjançant el programa RobotStudio) com de hardware que utilitzarà un robot ASEA IRB 120. D'aquesta manera, els alumnes del centre TCM, disposaran de les eines necessàries per a la realització de futures pràctiques amb aquest robot. Els passos que s'han seguit al llarg d'aquesta documentació són els que s'han considerat més fàcils d'entendre.

Aquest TFG és el naixement de la cèl·lula robòtica de l'ESUPT. Conté informació detallada del seu disseny i informació d'utilitat per realitzar un disseny de simulació a RobotStudio.

- **Monago Fernández, José. TFG Sistema de supervisió i control d'una cèl·lula robotitzada (primavera 2012) [7]**

En la documentació es presenta la informació necessària per a la realització d'un sistema de supervisió i control per a la cèl·lula robotitzada instal·lada en el Laboratori 4 del Tecnocampus. També s'inclou el estudi i la execució de les millores

funcionals de la cèl·lula, a més de posada en marxa del robot de docència SCORBOT -EV plus. L'aplicació s'ha realitzat segons les especificacions funcionals requerides per poder facilitar i orientar l'aprenentatge dels alumnes del TCM que realitzen pràctiques amb els robots instal·lats en la cèl·lula, i ser una eina de suport tècnic per al manteniment de la instal·lació.

En aquest document s'explica el disseny per realitzar un bon SCADA i també es parla del disseny de ladders per automatismes.

2.4. Coneixement de les eines

El projecte utilitza coneixements ja obtinguts a assignatures del Grau en Enginyeria Electrònica Industrial i Automàtica.

Per aconseguir continuar amb l'enginyeria de detall del projecte, és necessari saber utilitzar i dissenyar grafquets, ladders al software RsLogix 5000, una simulació al software RobotStudio, conèixer el codi de programació RAPID, realitzar una pantalla SCADA amb FactoryTalk View...

Una part de coneixements són de conèixer el concepte teòric, i altre saber com funciona el software amb el qual es programarà.

2.4.1. Grafcet i ladder

El grafcet [8] és un mètode gràfic i estructurat per a la descripció del comportament dels automatismes seqüencials mitjançant diagrames funcionals.

Hi ha diferents formes de representar un mateix grafcet depenent de la concreció del mateix; pot realitzar-se de manera més literària, o bé més tècnica (especificant ports d'entrada i sortida, referenciant a memòria o tags de l'autòmat...).

Al ser un mètode visual, és una eina optativa però de suport per realitzar després el programa ladder que s'enviarà a l'autòmat.

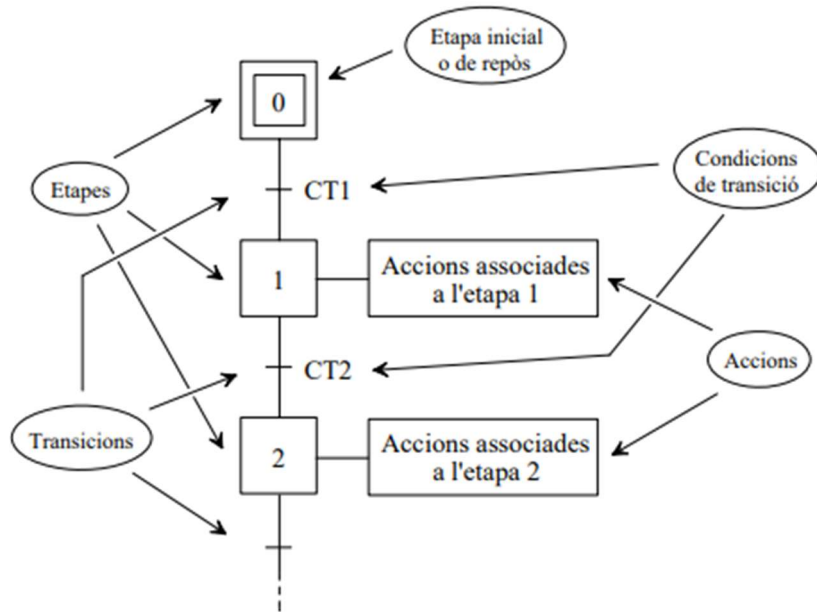


Figura 2.3: Elements bàsics d'un grafcet [Font: Automatització I (ESUPT)]

2.4.1.1 Elements i funcionament del grafcet

Un grafcet està format per etapes i transicions; l'evolució d'un procés seqüencial es pot dividir en etapes, en les quals es poden donar a terme accions. Per canviar d'una etapa a la següent, s'ha de complir la condició de transició (normalment la detecció d'un element per un sensor, l'activació d'alguna entrada o sortida...).

S'ha de tenir present que només es pot activar una etapa a la vegada, sense simultaneïtat.

Per definir varietats diferents de sistemes de control, trobem diferents situacions representades:

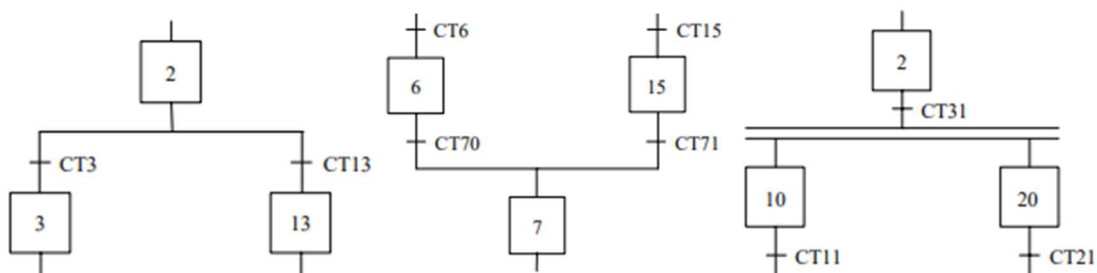


Figura 2.4: Tipus de direccionaments dels grafquets [Font: Automatització I (ESUPT)]

Bifurcacions: Dependent de la condició que s'activa una vegada finalitzada l'etapa anterior, s'activa una etapa o una altre.

Confluències: Diferents maneres d'arribar a una mateixa etapa des de dues altres diferents, dependent de quina condició es compleixi abans.

Salts: Permeten fer una transició d'etapes encara que a la representació no siguin consecutives.

Seqüències simultànies: Es divideix el graficet en seqüències asíncrones simultànies, representat amb la doble línia horitzontal. La fi de les seqüències també queda senyalada per la mateixa doble línia.

Respecte les accions, aquestes poden repetir-se a diferents etapes i en general, s'escriu el mateix tag o s'omet si no és necessària l'activació. Metodològicament, cada acció té assignat un tag booleà a cada etapa que farà d'indicador per la mateixa. Es poden condicionar accions, representant-les al graficet com un bloc amb una fletxa.

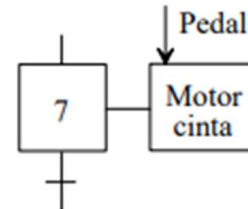


Figura 2.5: Accionament condicionat [Font: ESUPT]

En els processos de control, és habitual requerir d'eines addicionals com comptadors o temporitzadors. Aquests es poden incloure a la representació gràfica juntament amb les accions de cada etapa.

És convenient indicar que les condicions de transició són funcions lògiques d'una o més variables binaries. Per tant, es poden aplicar funcions conegudes com AND, OR i NOT.

2.4.1.2. Transformació a programa ladder

La transformació és necessària per a que l'autòmat entengui el programa. S'assigna a cada etapa un tag booleà intern de la memòria de dades de l'autòmat i es defineixen tags per a totes les accions i condicions de transició.

Els programes ladder poden ser molt complexos, es per això que es convenient seguir una estructura fixa per ubicar fàcilment cada part del programa. El programa està format per línies de control horitzontals a les que s'inclou interruptors, activadors... El conjunt d'aquestes línies donen pas al ladder, el qual permet estructurar-se de la següent manera:

Inicialització de la seqüència: Permet configurar el valor inicial que hauran de prendre algunes variables en el moment de posar l'autòmat en marxa. Comunament utilitzada per generar una etapa de repòs. En els controladors Logix de Rockwell s'hi troba un bit especial que només val 1 en el primer scan després de posar-se en marxa; es diu S:FS.

Operacions prèvies:	Construcció de flancs
Primera part: Control d'etapes	Inicialització de la seqüència
	Transicions
	Aturades
Segona part: Assignació d'accions	Posar valors inicials
	Temporitzadors
	Comptadors { ascend. descend. reset
	Sortides

Taula 2.1: Estructura i ordre d'un ladder [Font: Automatització I (ESUPT)]

Transicions: Es programen les transicions definides al graficet, fins al final del mateix (només canvis d'etapes, sense els blocs d'acció).

Aturades: És una línia de programa similar a la d'inicialització, agrupant tots els tags d'etapa i condicionant l'activació de l'etapa de repòs depenent de si la variable d'aturada està activada o no (permet tornar al repòs independentment de l'etapa). Si hi ha més d'una condició d'aturada, s'hauran de posar en paral·lel.

Posar valors inicials: En alguns casos s'ha de iniciar l'autòmat amb certs valors inicials de variables utilitzades al programa. La solució més efectiva és tornar a utilitzar el bit S:FS.

Temporitzadors i comptadors: Es disposa d'ambdós dispositius per programar tasques.

Sortides: En aquesta secció, s'indica en quines etapes es realitzen unes accions o altres. Encara que a una mateixa etapa, hi hagin dos accions diferents, s'ha d'emprar una línia de programació per a cada acció individualment.

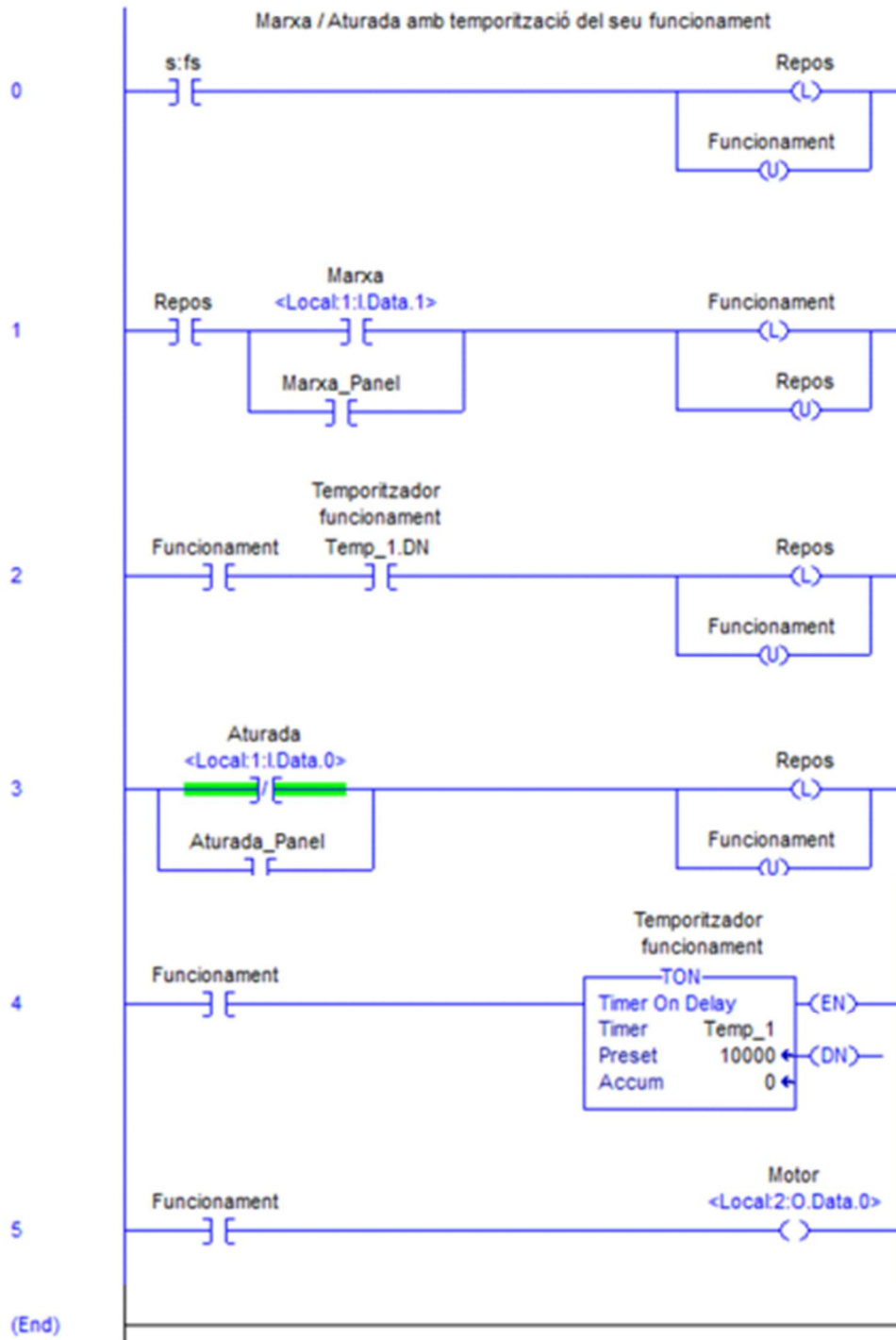


Figura 2.6: Exemple de ladder [Font: Automatització I (ESUPT)]

2.4.2. RobotStudio

RobotStudio és un software de simulació de la companyia ABB per el seu conjunt de robots industrials. Proporciona les eines per incrementar la rendibilitat d'un sistema robotitzat mitjançant tasques com la possibilitat de formació, programació i optimització, sense afectar la producció, el que proporciona nombrosos avantatges, com la reducció de riscos, engegada del sistema dissenyat més ràpid i increment de la productivitat.

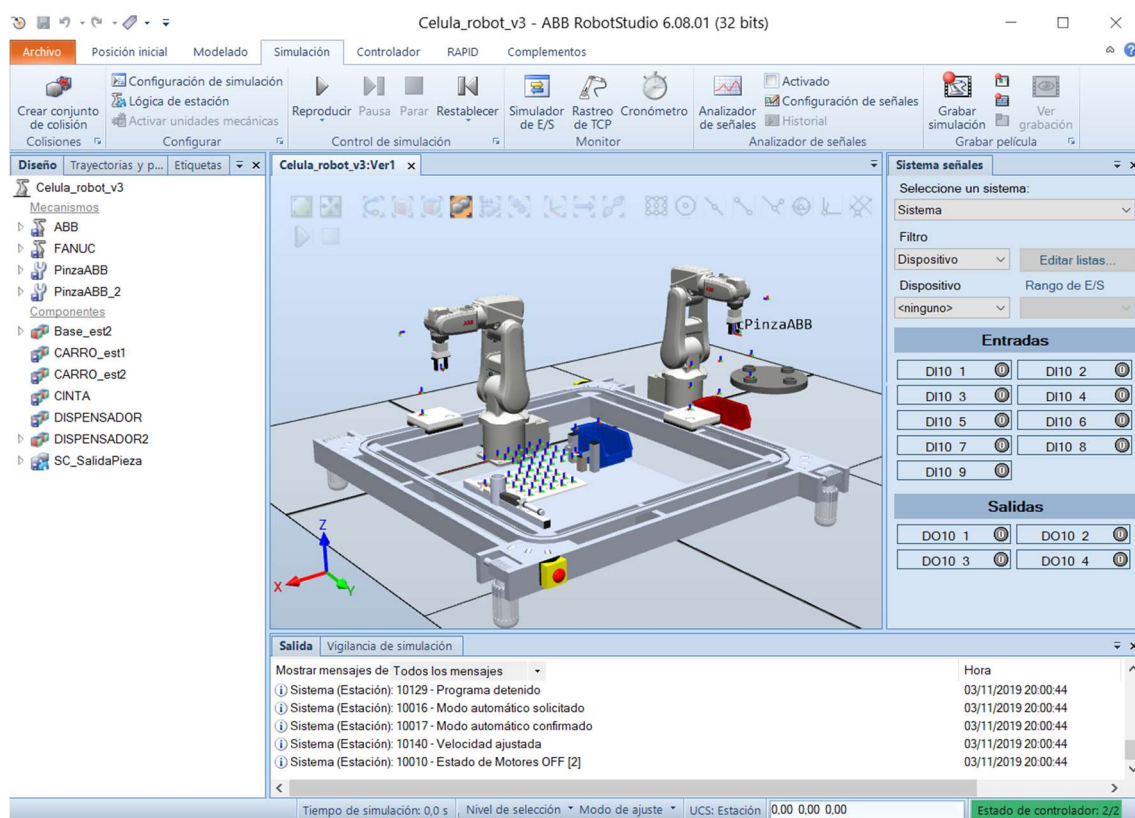


Figura 2.7: Visió general del programa RobotStudio [Font: pròpia]

El programa proporciona un entorn de treball on es poden realitzar models o generar-ne de nous, com per exemple, crear una pinça pel robot o una estació de treball (encara que és recomanable importar el disseny ja que les eines de disseny que proporciona no permeten massa complexitat).

Un dels principals atractius del programa és disposar d'una ample biblioteca de robots d'ABB per incloure al disseny, així com poder simular el controlador del robot (realment útil per després exportar el programa al propi robot).

A la barra superior del programa, es classifiquen en apartats les tasques que es poden dur a terme:

- Posició inicial: Per importar el robot i el seu sistema, així com importar geometries

i generar base de coordenades. També té l'opció 'punt', amb els quals es realitzaran les posteriors trajectòries. Altre opció molt habitual és 'veure el robot al punt', per així saber si el robot té abast suficient per arribar a la posició desitjada i evitar problemes dissenyant.

- **Modelat:** Des d'aquest apartat, és possible generar sòlids o superfícies, així com afegir components intel·ligents amb els quals realitzar una simulació més acurada a la realitat.
- **Simulació:** Conté totes les opcions per configurar la simulació de manera personal. Aquí es troben els comandaments de reproducció, pausa i aturada de la simulació, l'opció de visualitzar i interactuar amb el simulador d'E/S i fins i tot, gravar l'aplicació o simulació.
- **Controlador i RAPID:** A la pestanya 'Controlador' es pot afegir un nou controlador, reiniciar-ne un després d'haver efectuat canvis al programa RAPID o modificat la seva configuració, veure les seves propietats... mentre que la pestanya 'RAPID' s'enfoca en opcions diverses per tindre control sobre les línies de codi.

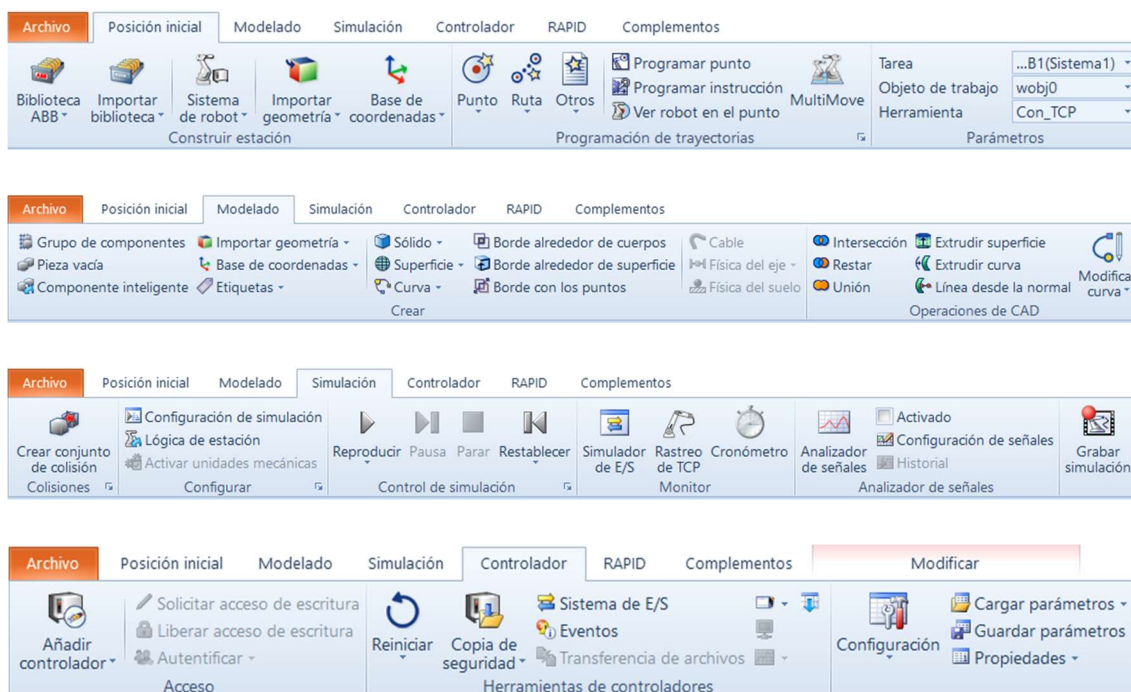


Figura 2.8: Possibilitats que ofereixen les categories de RobotStudio [Font: pròpia]

2.4.2.1. Programació en RAPID

La programació en RAPID pel robot ABB IRB 120 és força intuïtiva degut a les semblances amb el llenguatge C.

És un llenguatge molt estructurat, en el qual un programa senzill consta de parts diferenciades que es veuran amb un exemple:

1. Declaració de constants CONST
2. Declaració de variables VAR
3. Programa principal main
4. Rutines (PROC i/o TRAP)

```

1  MODULE Module1
2  CONST robtarget Repos:=[[300,0,300],[0,0,1,0],[0,0,0,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
3  CONST robtarget Target_10:=[[300,125,200],[0,0,1,0],[0,0,0,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
4  CONST robtarget Target_20:=[[450,125,200],[0,0,1,0],[0,0,0,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
5  CONST robtarget Target_30:=[[450,-125,200],[0,0,1,0],[-1,0,-1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
6  CONST robtarget Target_40:=[[300,-125,200],[0,0,1,0],[-1,0,-1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
7
8  VAR speeddata velocity:=v500;
9  VAR intnum velocitat_seguretad;
10 VAR intnum velocitat_normal;
11 VAR intnum atura;
12
13 PROC main()
14   VAR num estat:=0;
15   CONNECT velocitat_seguretad WITH security_speed;
16   CONNECT velocitat_normal WITH normal_speed;
17   CONNECT atura WITH stop_trap;
18   ISignalDI SEGURETAT,1,velocitat_seguretad;
19   ISignalDI SEGURETAT,0,velocitat_normal;
20   ISignalDI ATURADA,1,atura;
21   Rep;
22   WHILE TRUE DO
23     IF START=1 THEN
24       Path_10;
25       estat:=1;
26     ENDIF
27     IF START=0 AND estat=1 THEN
28       Rep;
29       estat:=0;
30     ENDIF
31   ENDWHILE
32 ENDPROC
33
34 PROC Rep()
35   MoveL Repos,velocity,fine,Con_TCP\WObj:=wobj0;
36 ENDPROC
37 PROC Path_10()
38   MoveL Target_10,velocity,fine,Con_TCP\WObj:=wobj0;
39   MoveL Target_20,velocity,fine,Con_TCP\WObj:=wobj0;
40   MoveL Target_30,velocity,fine,Con_TCP\WObj:=wobj0;
41   MoveL Target_40,velocity,fine,Con_TCP\WObj:=wobj0;
42   MoveL Target_10,velocity,fine,Con_TCP\WObj:=wobj0;
43 ENDPROC
44 TRAP security_speed
45   velocity:=v100;
46 ENDTRAP
47 TRAP normal_speed
48   velocity:=v500;
49 ENDTRAP
50 TRAP stop_trap
51   StopMove;
52   WaitUntil (ATURADA=0 AND START=1);
53   StartMove;
54 ENDTRAP
55
56 ENDMODULE

```

Figura 2.9: Exemple de programa RAPID [Font: pròpia]

- 1 En aquest apartat es declaren les constants de les coordenades dels punts per generar les trajectòries (el propi programa ja les declara).
- 2 Es declaren les variables globals numèriques, booleanes, de speeddata (velocitat de trajectòria) que s'utilitzaran al programa principal o a les seves subrutines.
- 3 Programa principal del robot. Des del main() es poden cridar subrutines, declarar variables locals, activar/desactivar senyals digitals i utilitzar funcions similars a les de C, com el WHILE i IF.
- 4 Les subrutines de tipus PROC es generen al transferir la informació de les trajectòries a llenguatge RAPID. N'hi ha de diferent tipus, com les TRAP, que permeten interrompre el flux del programa per actuar en una eventualitat externa.

2.4.3. SCADA i FactoryTalk View

SCADA (acrònim de 'Supervisory Control And Data Acquisition') és un concepte que s'utilitza per nombrar un software per ordinadors que permet controlar i supervisar processos industrials a distància.

Els components que formen el sistema són:

- **HMI:** És la interfície que connecta a l'home amb la màquina presentant les dades del procés davant l'operari mitjançant un sistema de monitorització. A més, controla l'acció a desenvolupar a través d'una pantalla (en l'actualitat tàctil).
- **Sistema de supervisió o MTU (Ordinador / Ordinador):** Té la funció de recopilar les dades del procés i enviar les instruccions mitjançant una línia d'ordres.
- **Unitats Terminals Remotes (RTU):** Són microprocessadors (Ordinadors Remots) que obtenen senyals independents d'una acció per enviar la informació obtinguda remotament perquè es processa. Es connecten a sensors que converteixen els senyals rebuts en dades digitals que l'envien a l'ordinador o sistema de supervisió (MTU)
- **PLCs:** Comunament denominats autòmats programables, aquests són utilitzats en el sistema com a dispositius de camp a causa de que són més econòmics, versàtils, flexibles i configurables que les RTUs comentades anteriorment.

- **Xarxa o sistema de comunicació:** S'encarrega d'establir la connectivitat de l'ordinador (MTU) a les RTUs i PLCs. Per a això utilitza connexions via mòdem, Ethernet, Wifi o fibra òptica.
- **Sensors:** Són dispositius que actuen com a detectors de magnituds físiques o químiques, denominades variables d'instrumentació, i les converteixen en variables o senyals elèctrics.
- **Actuador:** És un dispositiu mecànic que s'utilitza per actuar o oferir moviment sobre un altre dispositiu mecànic.

El software HMI *FactoryTalk View* de Rockwell Automation té una bona sinergia amb els programes del RsLogix 5000, facilitant l'intercanvi de dades d'interès.

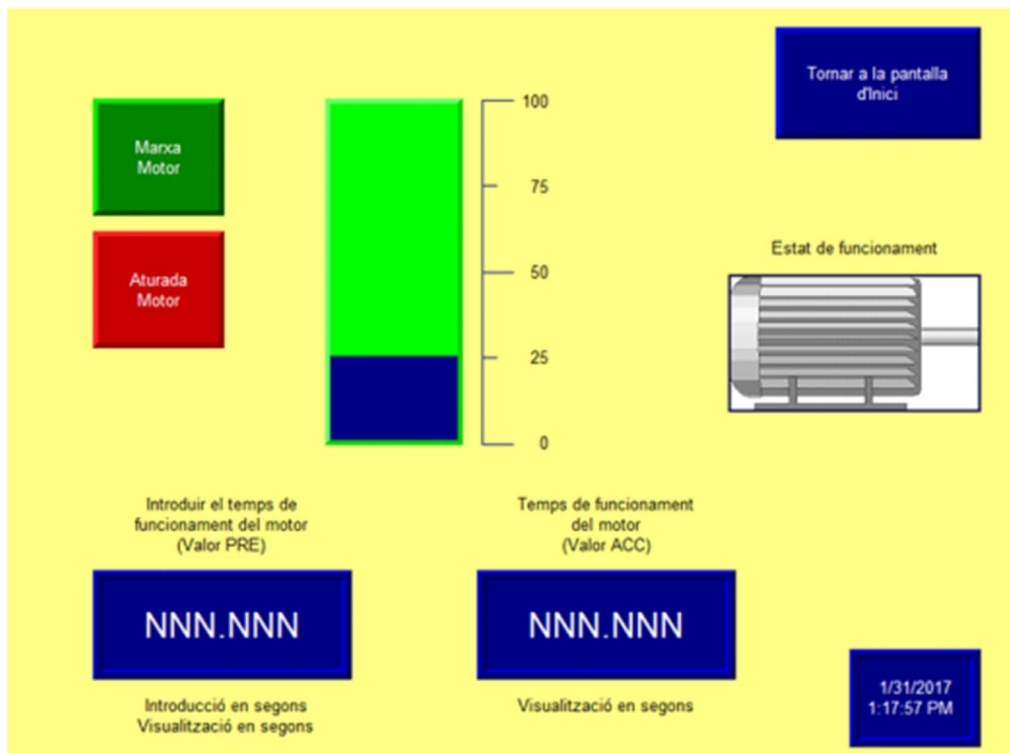


Figura 2.10: Exemple de pantalla generada amb FactoryTalk View [Font: Automatització I (ESUPT)]

Entre les possibilitats que ofereix aquesta aplicació, es poden configurar diferents pantalles i configurar-les tant en ordre com les preferències de color.

Es poden inserir botons gràfics que interaccionen amb els actuadors del ladder i permeten controlar-lo de forma senzilla i visual.

Si s'escau, es poden visualitzar els temps dels temporitzadors, així com incloure botons per navegar entre pantalles, gràfics variables depenent d'una variable o comptador, inserir la data i l'hora, incloure comentaris...

3. Especificacions tècniques i plantejament del projecte

3.1. Composició de la cèl·lula de fabricació

El món industrial és molt extens i els processos manufacturadors són diversos depenent del producte, el lloc de treball... és per això que, amb les eines que aporta l'automatització, hi ha solucions diferents per a cada cas.

En el cas del projecte que ens ocupa, al treballar sobre la cèl·lula de la que disposa l'ESUPT, la seva composició ja està definida.

El moviment de la peça al llarg de la cèl·lula és possible gràcies a el muntatge d'una cinta transportadora de forma quadrada i dimensions 2x2 metres (accionada per quatre motors), formant el mòdul primari.

Seguidament, es definiran i localitzaran els diferents components que la conformen.

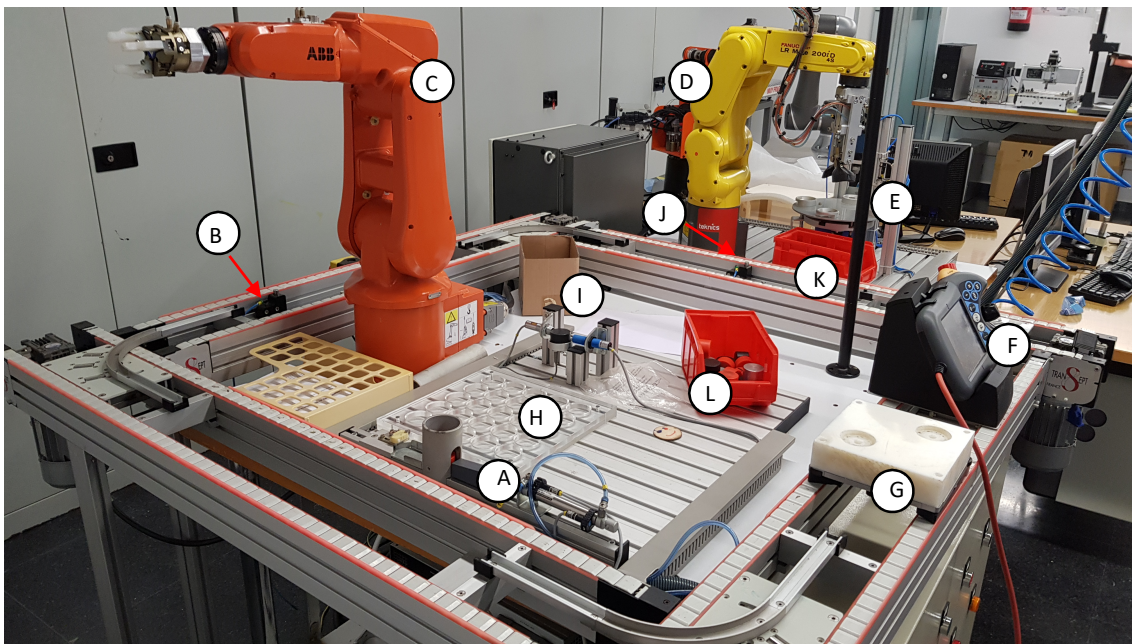


Figura 3.1: Localització dels diferents components que formen la cèl·lula de fabricació [Font: pròpia]

A	Dispensador de peces
B	Estació primària
C	Robot ABB IRB 120
D	Robot FANUC LR Mate 200iD/4S
E	Mòdul secundari (base rotativa amb trepant i validació del forat)
F	Consola Flex Pendant
G	Carro de transport
H	Magatzem
I	Mòdul de detecció (conformat per tres sensors; òptic, capacitiu i inductiu)
J	Estació secundària
K	Dipòsit de resta secundari
L	Dipòsit de resta primari

Taula 3.1: Codi i descripció dels components de la cèl·lula [Font: pròpia]

Els principals elements de control de la cèl·lula són els dos braços antropomòrfics; l'ABB IRB 120 (principal) i el FANUC LR Mate 200ID 4S (secundari), comandats pels seus respectius controladors.

3.1.1. Característiques del ABB IRB 120 i Flex Pendant

Era el robot industrial multiús més petit de la companyia ABB fins a l'arribada de l'IRB 120T (una versió del mateix robot millorada).



Figura 3.2: Robot IRB 120 [Font: ABB]

- Disposa de 6 eixos de rotació.
- Pesa només 25kg.
- Payload màxim: 3kg de càrrega.
- Té un rang màxim de 0.58m.
- Armload límit: 300g. Aquesta és la càrrega aplicada directament al braç del robot (sigui una càrrega directa o l'efecte de cables i tubs associats a l'eina del robot).
- Repetibilitat: 0.01mm.
- Utilitza un controlador IRC5 Compact/IRC5 Single Cabinet.

Les característiques dinàmiques depenen del pes i forma de l'eina utilitzada i de la pròpia peça subjectada.

El robot disposa d'una consola d'actuació manual anomenada Flex Pendant, amb la qual es poden inserir programes via USB, editar, gravar posicions del robot per definir trajectòries així com programar directament des de la mateixa.



Figura 3.3: Consola Flex Pendant [Font: ABB]

A	Connector
B	Pantalla tàctil
C	Botó d'aturada d'emergència
D	Joystick
E	Port USB
F	Botó de sosteniment
G	Punter
H	Pulsador de restabliment

Taula 3.2: Codi i descripció d'elements de la consola Flex Pendant [Font: pròpia]

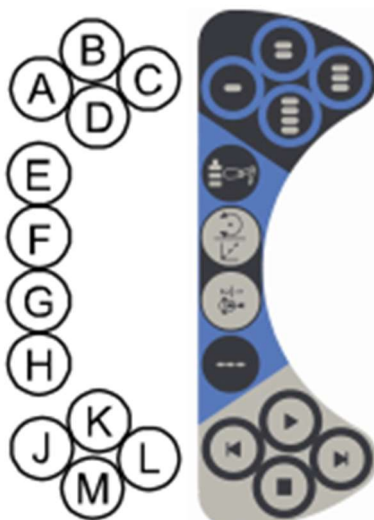


Figura 3.4: Panell de control de la consola Flex Pendant [Font: ABB]

A-D	Tecles programables 1-4
E	Seleccionar unitat mecànica
F	Mètode de moviment (reorientació o lineal)
G	Mode moviment (eixos 1-3 o 4-6)
H	Activar/desactivar increments
J	RETROCEDIR una instrucció
K	INICIAR execució del programa
L	AVANÇAR una instrucció
M	ATURADA execució del programa

Taula 3.3: Codi i descripció d'elements al panell de la consola Flex Pendant [Font: pròpia]

És compatible amb el software de simulació RobotStudio, propietat d'ABB, amb el qual es pot programar el robot amb codi en llenguatge RAPID (que proporciona una fàcil programació al tenir grans similituds al llenguatge C). Té una versió de prova disponible per un mes.

3.1.2. Característiques del FANUC LR Mate 200iD/4S



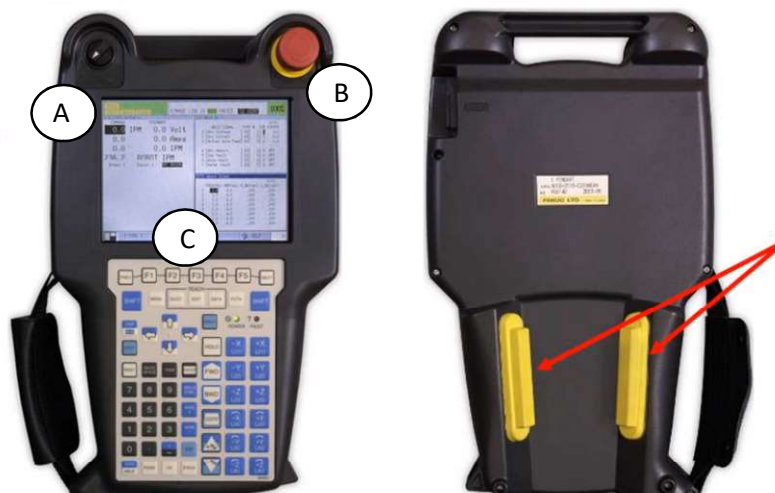
Figura 3.5: Robot LR Mate 200iD/4S
[Font: FANUC]

És un minirobot molt compacte, amb el braç més curt de la gama LR Mate, fent que sigui adequat per automatitzar processos en diverses indústries. FANUC assegura la seva fiabilitat, així com una relació pes/capacitat de càrrega excel·lent.

- 6 eixos de rotació.
- Pesa només 25kg.
- Té un rang màxim de 0.55m.
- Payload màxim: 4kg de càrrega.
- Repetibilitat: 0.01mm.
- Utilitza un controlador R-30ib Plus.

Les característiques dinàmiques depenen del pes i forma de l'eina utilitzada i de la pròpia peça subjectada.

És compatible amb el software de simulació Roboguide, propietat de FANUC, amb el qual es pot programar el robot amb codi en llenguatge KAREL. Aquest programa de simulació no té versió lliure, encara que es pot contactar via correu per obtenir una prova del mateix.



A: Interruptor ON/OFF control manual

B: Parada d'emergència

C: Pantalla

Detectors d'home mort (dead-man grips): És necessari polsar-los al operar amb el robot.

Figura 3.6: Consola iPendant [Font: pròpia]

El robot disposa d'una consola d'actuació manual anomenada iPendant, des de la qual es requereix fer la programació del robot (s'escriu el codi directament a la consola degut a la falta d'entrada USB).

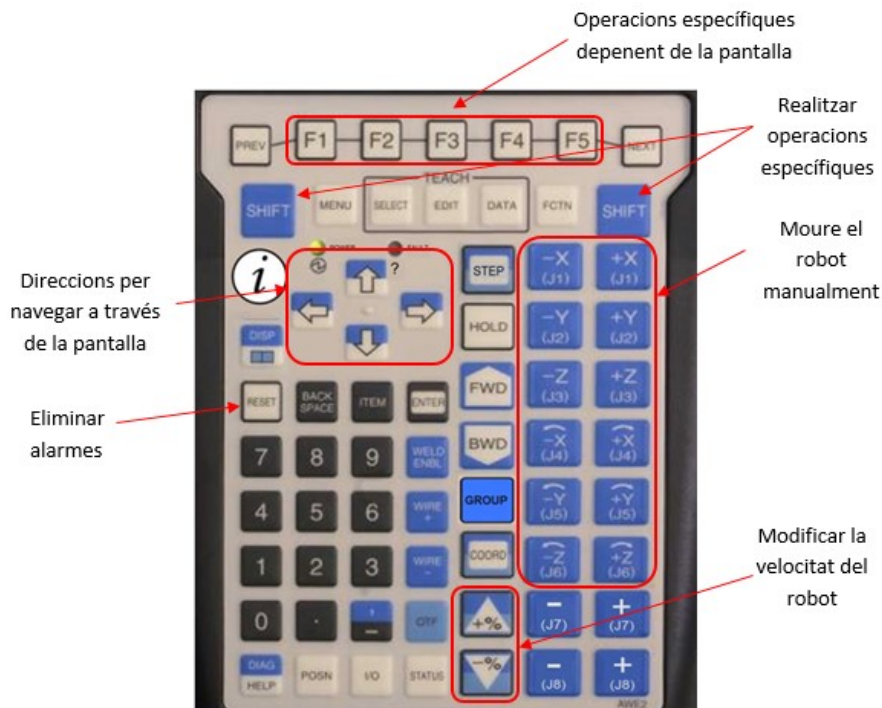


Figura 3.7: Botons de la consola iPendant [Font: pròpia]

3.2 Solució plantejada

A l'hora de dur a terme un projecte a una cèl·lula de fabricació, que té com objectius el disseny i programació per una possible aplicació industrial real, és convenient conèixer el disseny físic de la mateixa i les seves limitacions i avantatges per diferents tipus de processos.

Normalment el que trobaríem en un cas real a l'indústria seria la concessió del disseny sencer de la màquina i la seva programació a fi de dur a terme una tasca concreta. En el cas particular d'aquest projecte, per poder realitzar una demostració pràctica final, aquest sistema real serà adaptat a les possibilitats i les eines de treball a disposició dels estudiants de grau mecatrònics i, per tant, parteix de la limitació de tenir ja definida amb anterioritat la cèl·lula.

L'aplicació que és realitzarà intenta simular un procés típic industrial, en el que l'usuari pugui demanar una certa quantitat de peces d'un cert tipus, es realitzi el procés productiu per aconseguir el producte final, i el circuit automatitzat li retorni al magatzem de producte acabat la comanda.

La implementació de l'aplicació, comptarà amb els robots industrials ABB IRB120 i FANUC LR Mate 200Id 4S, així com els programes software RobotStudio (propietat d'ABB) i RSLogix 5000 (propietat de Rockwell Automation), disponibles al centre d'estudis.

Aprofitant la disposició actual dels elements de l'entorn de treball dels robots, el procés a seguir a la tasca a realitzar és el següent:

1. Inicialitzar el sistema.
2. Comanda: Demanar una certa quantitat de peces d'un tipus concret (entre 3 diferents; de plàstic vermelles, de plàstic negres o metàl·liques).
3. El braç robòtic ABB IRB120 agafa la peça del dispensador de peces.
4. Porta la peça fins a un mòdul de detecció, on s'avalua i distingeix.
5. Si la peça és del tipus demanat, el braç deposita la peça a un carro de transport a la primera estació de treball. Si és d'un tipus diferent al sol·licitat, es diposita al dipòsit de resta principal.
6. Al seguir amb el procés anterior i arribar a la capacitat màxima del carro (dues peces), aquest es trasllada fins a l'estació secundària, on s'atura i és detectat.
7. El braç robòtic FANUC LR Mate 200Id 4S agafa la peça de l'estació i la porta a un mòdul en forma de disc rotatori.
8. Es realitza el trepat de la peça i posterior validació del forat; si la peça no és vàlida, es retira de circulació deixant-la al dipòsit de resta secundari.
9. El braç torna la peça al carro situat a l'estació secundària.
10. Quan el carro és ple, aquest surt de la estació secundària en direcció a l'estació primària.
11. Al arribar i ser detectat, el braç ABB agafa les peces i les diposita al magatzem.

Aquest procés es repetirà cíclicament fins a obtenir el número de peces sol·licitades per l'usuari.

L'objectiu és maximitzar l'eficiència del procés, amb la intenció d'aconseguir dos carros en circulació i fer esperar el mínim possible els carros buits a les diferents estacions de treball.

4. Metodologia

Ja que no és un treball purament teòric ja que té una part pràctica, es segueix una metodologia més comú en els treballs d'enginyeria tècnica.

Els punts a seguir cronològicament durant el desenvolupament del treball són els següents:

1) Establiment de les especificacions

Primerament, s'estableixen les normes, exigències i procediments que s'han de complir en el projecte. Normalment, és el client qui estableix les especificacions d'un projecte d'enginyeria.

En aquest cas, es demana el disseny i programació d'un sistema real automatitzat ja existent. Per tant, la viabilitat dels requeriments i posteriors especificacions tenen una relació amb el què permet realitzar el sistema.

2) Estudi del sistema

Un estudi a fons del sistema i la seva distribució i característiques permet conèixer quines especificacions són realistes i quines han de descartar-se o canviar per altres.

En aquest estudi, es disseccionen els diferents components de la cèl·lula i s'estudia i verifica el seu funcionament per tal de comprovar el seu estat.

Altre punt que afecta a les propostes per tal de resoldre les especificacions és enfocar el sistema en termes generals; no només el hardware que el forma, sinó també el software a utilitzar i el seu funcionament.

Al obtenir una visió completa del sistema i coneixent les especificacions que es requereix incloure al treball, es pot realitzar una aproximació a com es volen solucionar les mateixes.

3) Proposta d'alternatives

A nivell intern, una vegada establides les especificacions i estudiat el sistema, és necessari elaborar un llistat d'alternatives que permetin arribar al fi del projecte complint totes les expectatives.

Amb aquestes alternatives, s'ha de seleccionar únicament una per seguir-la durant el treball.

4) Elecció de les alternatives

L'elecció de l'alternativa es fa seguint uns criteris respecte la seva viabilitat (tècnica, econòmica i mediambiental) i la idoneïtat de la mateixa, des d'un punt de vista referent a la seva complexitat.

Aquesta solució s'ha de plantejar i justificar degudament.

En aquest cas, l'alternativa escollida és un disseny similar amb el que la cèl·lula ja ha treballat amb anterioritat amb la inclusió d'alguns canvis i millores que fan que el funcionament pugui semblar similar, però amb modificacions profundes en el disseny i programació.

5) Establiment de les parts i de les interaccions entre elles

Una vegada escollida l'alternativa, es imprescindible dividir el projecte en parts de menys pes per simplificar la seva organització.

El projecte, tal com el propi títol indica, té dues parts molt diferenciades; disseny i programació. Aquestes dues parts estan connectades entre elles, pel qual, es necessari programar que s'ha de fer en cada moment.

6) Programació de les tasques

Abans de començar a dissenyar les diferents parts, és inevitable programar les tasques.

Per realitzar el projecte, s'ha decidit començar dissenyant la simulació del funcionament de la cèl·lula i els dissenys del ladder i SCADA, per acabar amb una última part de programació sobre els robots.

Un punt a tenir present és com condicionarien factors externs al projecte a cadascuna de les parts. Amb la realització d'un pla de contingència, la programació d'algunes tasques es veu modificada i es reestructura satisfactòriament.

7) Disseny de cadascuna de les parts i de les interaccions

Pel disseny de la simulació s'utilitza el software *Robotstudio* i el codi RAPID per enviar les comandes al robot ABB. Aquesta interacció es possible degut a que tant el programari com el robot són propietat de *Rockwell*; per programar el robot

FANUC és necessari codificar des de la consola manual iPendant (tot i que per a la simulació, es 'simularà' com un altre robot antropomòrfic ABB IRB 120).

Els programes per l'autòmat, són l'*RSTLogix Studio 5000* per dissenyar els ladders, i *FactoryTalk View* per el disseny del SCADA. Aquests dos últims softwares, gràcies a pertànyer com el propi autòmat a *Allen-Bradley* (també propietat de *Rockwell*), es comuniquen entre ells amb un altre programa anomenat RSLinx.

En el cas d'utilitzar el pla de contingència, seria necessari un quart programa, *RSTLogix Emulator 5000*, que permet emular l'autòmat o controlador.

8) Test en simulació

La simulació es pot testejar des del programa *Robotstudio*, el qual dedica tot un apartat a verificar el funcionament del disseny.

El funcionament es pot provar mentre es dissenya, facilitant la seva pròpia elaboració i poden observar amb deteniment cada paràmetre dels robots una vegada en marxa.

A part de la simulació, els ladders es poden verificar al propi programa de disseny si es té present el que comporta cada activació d'una entrada o sortida, encara que és més laboriós.

9) Posta en marxa

Per últim, i a fi de comprovar el funcionament de les diferents programacions en un mateix entorn, es posa en marxa la cèl·lula robòtica.

De forma presencial s'experimenta i solucionen els possibles inconvenients respecte el funcionament esperat, com a la realitat de l'indústria.

A cada sessió presencial s'anoten els avanços i problemes trobats en format d'informe.

10) Documentació en paral·lel de cada part

Durant el transcurs del projecte, es va portant a terme el redactat de tot el material d'interès pel projecte, així com del procés de realització de tots els punts anteriors, aconseguint amb la seva finalització la memòria del projecte.

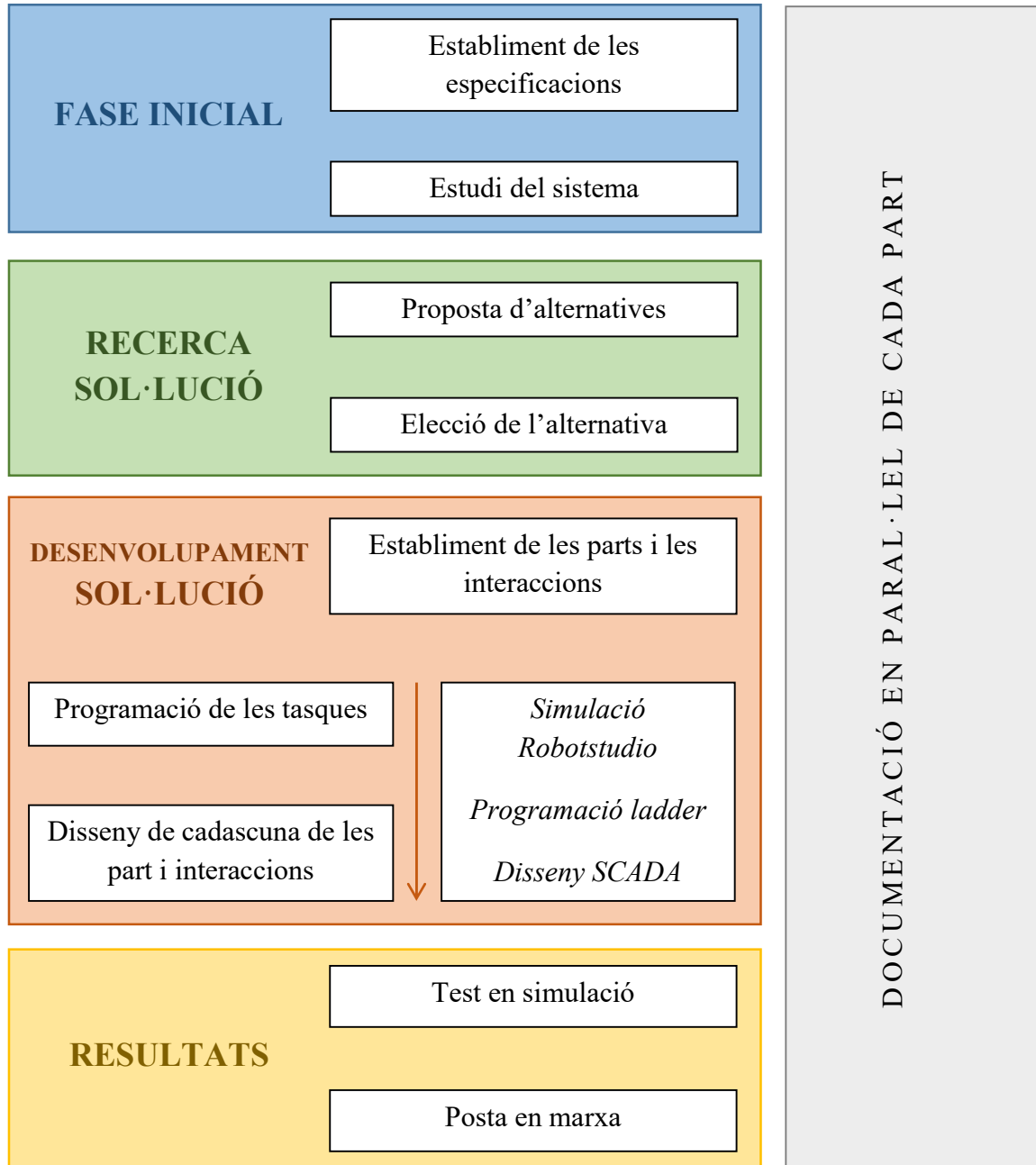


Figura 4.1: Esquemàtic de la metodologia emprada al treball [Font: pròpia]

5. Lògica de funcionament

El cas que es dissenya a continuació parteix de la premissa d'aconseguir dos carros de transport de peces funcionals a la vegada.

Aquest canvi comporta una dificultat afegida respecte el funcionament amb un sol carro, on el funcionament era seqüencial; s'ha de tenir en compte que els dos robots treballaran a vegades de forma simultània.

Alguns dels canvis a tenir presents són:

- De partida, els dos carros estan en qualsevol posició de la cinta. Un cop l'usuari activa la marxa, s'activen les cintes de la base. El primer carro que arriba i es detectat a l'estació 1 serà el carro 1, i el següent es nomenarà carro 2 (durant tota l'execució de l'aplicació).
- Una vegada omplert amb les peces sol·licitades el carro 1 pel robot ABB, el carro 1 es mou per la cinta mentre que el carro 2 queda aturat a l'estació 1, on s'omple el carro amb la mateixa operativa que s'ha omplert el carro 1. Mentrestant, durant l'execució de l'emplenament del carro 2, el carro 1 ha arribat a l'estació 2 i, al ser detectat, s'atura i comença el programa del robot FANUC en el que participa el mòdul rotatori de mecanitzat i validació.
- Durant el funcionament del programa a la cèl·lula es possible les col·lisions entre carros quan un dels dos està aturat a una estació i l'altre també arriba a la mateixa. Aquests casos no han sigut plantejats com a un problema durant el disseny, ja que la velocitat dels carros és petita i la magnitud de la col·lisió no afecta negativament al funcionament del programa. Si es volgués eliminar aquesta casuística, una de les opcions seria aturar el tram de cinta de les estacions 1 o 2 quan es detecta un carro al sensor inductiu de l'estació, encara que el temps de trajecte dels carros d'una estació a l'altre augmentaria uns segons. Per millorar encara més el funcionament, la sensorització dels carros és una possible solució, implantant sensor de col·lisió (de proximitat) a la part davantera i posterior dels carros i fent que aquestes senyals, juntament amb les dels sensors de les estacions permetin aturar el corresponent tram de cinta transportadora.

Per a la realització del disseny funcional de la cèl·lula, s'ha decidit emprar el GRAFCET, com a eina de baix nivell però molt potent que permet desenvolupar els algorismes per

els tres dispositius lògics de la cèl·lula; l'autòmat programable CompactLogix L32E i els dos robots. A partir d'aquesta base, és possible desenvolupar el ladder de l'autòmat i els programes per les controladores dels robots.

5.1. Autòmat CompactLogix L32E

5.1.1. Interfase entre l'autòmat i la cèl·lula

Autòmat			
Dispositiu	ENTRADA	SORTIDA	ALIAS
Sensor_FC_Dispensador	1:IData9	6:OData2	Peça_dispensada (A)
Pols_rearmament (polsador)	1:IData2	-	-
Marxa (polsador)	1:IData0	4:OData7	Marxa_motors_cinta
Treu_peça (Dispensador) (A)	3:IData11	4:OData0	Dispensa_peça
Sensor_E1	1:IData8	4:OData9	Enclavament_E1
Sensor_E2	2:IData8	5:OData15	Enclavament_E2
Sensor_òptic (Base detec.)	1:IData5	-	-
Sensor_capacitiu (Base detec.)	1:IData6	-	-
Sensor_inductiu (Base detec.)	1:IData7	-	-
Sensor mòdul rotatori (Disc)	1:IData15	5:OData15	Rotar90
Trepant_FC	1:IData16	5:OData13	Baixa_Trepant
Verificador_FC	1:IData17	5:OData14	Baixa_Verificador
Sensor_Verificador	1:IData18	5:OData16	Peça_noDrill

Taula 5.1: Interfase entre l'autòmat i la cèl·lula [Font: pròpia]

5.1.2. Interfase entre l'autòmat i els robots

Autòmat			Control ABB
ALIAS	SORTIDA		ENTRADA
ABB_repos	6:OData1	→	DI10_1
ALIAS	SORTIDA		ENTRADA

Peça_dispensada	6:OData2	→	DI10_2
p_incorrecta	6:OData3	→	DI10_3
p_correcta	6:OData8	→	DI10_4
Recull_peça	6:OData11	→	DI10_6
Set_Pinza	6:OData12	→	DI10_7 → DO10_1
Peça_pos1	6:OData13	→	DI10_8
Peça_pos2	6:OData14	→	DI10_9

Taula 5.2: Interfase de sortides de l'autòmat cap al robot ABB [Font: pròpia]

Autòmat			Control ABB
ALIAS	ENTRADA		SORTIDA
Treu_peça (Dispensador)	3:IData11	←	DO10_4
Peça_detector	3:IData12	←	DO10_2
Peça_al_carro	3:IData13	←	DO10_3

Taula 5.3: Interfase d'entrades de l'autòmat des del robot ABB [Font: pròpia]

Autòmat			Control FANUC
ALIAS	SORTIDA		ENTRADA
FANUC_repos	5:OData6	→	DI[107]
Recollir_peça	5:OData7	→	DI[101]
Peça_bona	5:OData8	→	DI[103]
Peça_dolenta	5:OData10	→	DI[105]
Set_PinçaF1	5:OData11	→	RO[1]
Set_PinçaF2	5:OData12	→	RO[2]

Taula 5.4: Interfase de sortides de l'autòmat cap al robot FANUC [Font: pròpia]

Autòmat			Control FANUC
ALIAS	SORTIDA		ENTRADA
Fi_càrrega	3:IData1	←	DO[111]

Taula 5.5: Interfase d'entrades de l'autòmat des del robot ABB [Font: pròpia]

5.1.3. GRAFCET de l'autòmat

Durant l'execució del programa de l'autòmat, és necessari controlar ambdós robots independentment de l'estat en que es troba l'altre. Es per això que a l'hora de fer el disseny GRAFCET, una vegada l'usuari activa l'entrada Marxa, es realitza una divergència en 'Y', representada amb una doble línia horitzontal paral·lela.

El disseny té un total de 17 estats, pertanyent 9 estats a accions i moviments d'interacció amb el robot IRB 120 (part esquerra de la figura 5.1), 6 estats en el cas del LR Mate 200 (part dreta de la figura 5.1) i 2 estats comuns.

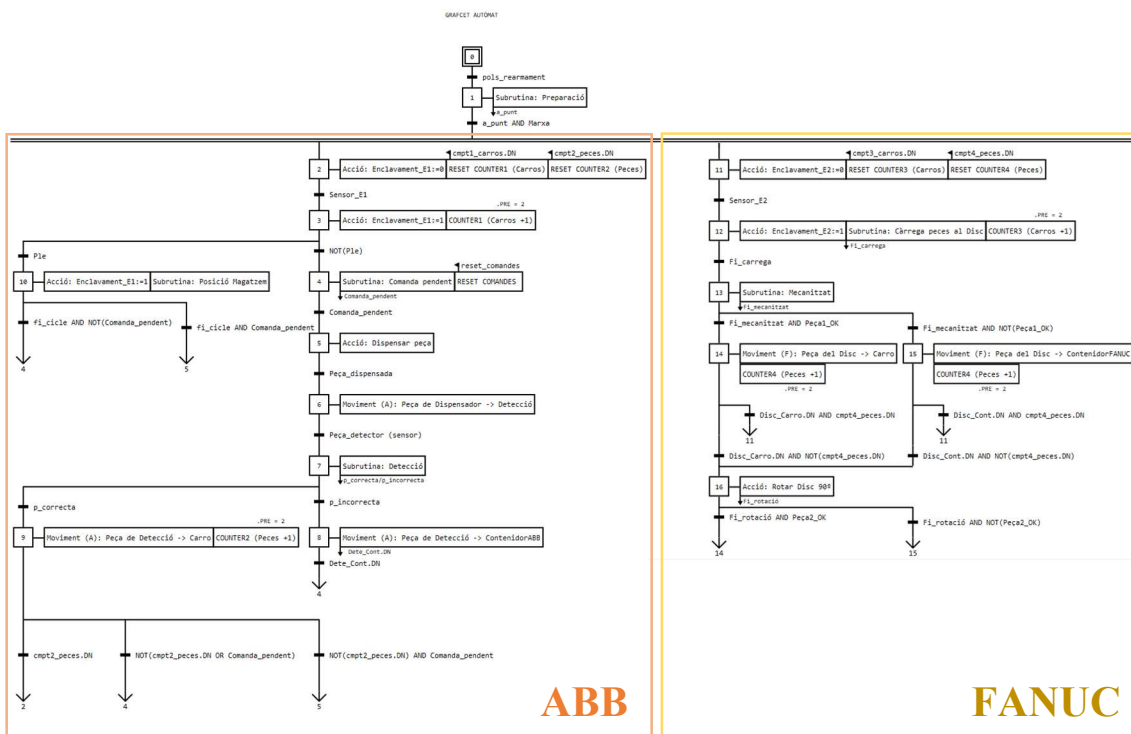


Figura 5.1: GRAFCET de l'autòmat [Font: pròpia]

[NOTA: Es pot trobar una versió amb major resolució inclosa a l'annex V]

Per el correcte funcionament del disseny, s'ha utilitzat diferents variables booleans, així com comptadors en les etapes on és pertinent.

Com s'ha mostrat a l'apartat anterior, es proporcionen 'alias' a les entrades i sortides de l'autòmat per evitar utilitzar la nomenclatura física d'I/O.

L'**Estat 0** és l'estat de repòs de la màquina. Quan es rep una activació manual al pulsador de rearmament, l'estat canvia a l'**Estat 1**.

A l'**Estat 1**, l'autòmat activa la subrutina 'Preparació':

- Preparació: Les senyals de sortida digitals ABB_repòs i FANUC_repòs s'activen. Així com també s'encarrega de posar a 0 les sortides que controlen la posició del disc, trepant i del verificador (Rotar90, Baixa_Trepant i Baixa_Verificador) per així assegurar-se que estan en la posició més elevada. Un cop realitzada l'operació, s'activa la variable booleana 'a_punt'.

Aquesta variable de sortida de la subrutina 'Preparació' (a_punt), juntament amb l'activació manual de la senyal Marxa, s'utilitza com condició de transició cap a l'**Estat 2** (ABB) i l'**Estat 11** (FANUC).

Seguint a l'**Estat 2**, es reseteja el valor de 'Enclavament_E1', que controla l'aturada del carro

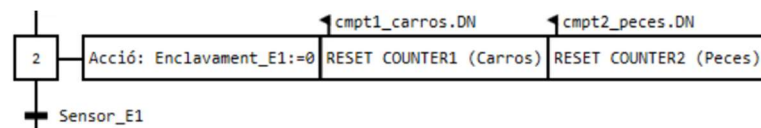


Figura 5.2: Estat 2 del GRAFCET de l'autòmat. [Font: pròpia]

a l'estació 1. També s'inclou el reset dels comptadors 1 i 2 (per a controlar el carro que es troba aturat a l'estació i el número de peces que conté, respectivament). Els comptadors tornen a 0 quan els seus valors arriben al valor de preset (.DN = .PRE), que en els dos casos és 2. Per tant, són cicles 0 → 1 → 2, on es canvia al següent valor per cada iteració a l'etapa que conté el comptador.

Al estar 'Marxa' activat, els carros es mouen per la cinta, fins que al detectar-se un mitjançant el Sensor_E1, es realitza la transició d'estat cap a l'**Estat 3**.

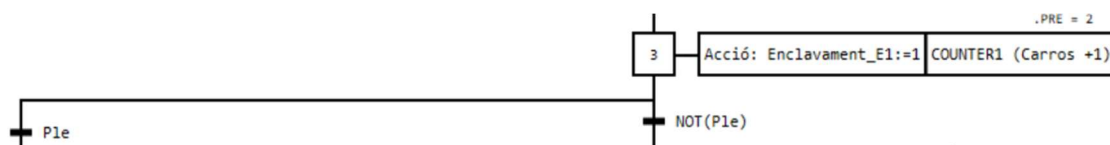


Figura 5.3: Estat 3 del GRAFCET de l'autòmat. [Font: pròpia]

A l'**Estat 3** s'activa l'enclavament de l'estació 1 (Enclavament_E1) i el comptador 1 de carros augmenta una unitat. En aquest estat, el carro pot estar ple (1 o més peces) o buit. Amb la variable Ple, es pot discriminar els dos casos. En tot moment es té control de les

peces que hi ha a cada carro mitjançant variables objecte anomenades Carro1 i Carro2, on es guarda la posició al carro segons la posició al propi objecte (per exemple, les posicions del primer carro són Carro1[1] i Carro1[2]). La selecció entre Carro1 i Carro2 i entre la posició 1 i 2 de l'objecte depèn de la posició del comptador 1 (carros) i el comptador 2 (peces), respectivament.

Si el carro està buit, l'autòmat canvia a l'**Estat 4**. En aquest estat, s'activa la subrutina 'Comanda Pendent':

- **Comanda Pendent:** Dintre de la subrutina hi ha tres comptadors (amb .PRE = 1), un per a cada tipus de comanda (vermella, negra o alumini). La sol·licitud de peça s'incorpora al programa mitjançant variables tipus bool, que activen els respectius comptadors. Si es detecta que algun dels comptadors arriba al seu valor preset (.DN = .PRE), llavors activa la variable anomenada 'Comanda_pendent'.

Aquesta mateixa etapa disposa d'un reset de comandes per entrada manual o al aturar-se l'execució.

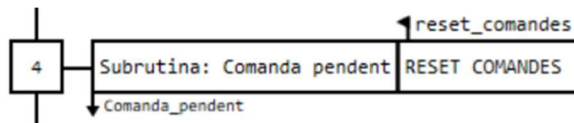


Figura 5.4: Estat 4 del GRAFCET de l'autòmat. [Font: pròpia]

La variable 'Comanda_pendent' és la condició de transició cap a l'**Estat 5**, on s'envia la senyal 'Dispensa_peça' al

cilindre pneumàtic del dispensador. Quan la peça arriba al final de cursa del detector, s'avisava a l'autòmat activant la entrada 'Sensor_FC_Dispensador' i es canvia a l'**Estat 6**, en el qual s'ordena al robot ABB que agafi la peça del dispensador per portar-la a detecció. Un cop a detecció, el robot ABB envia un senyal a l'autòmat que activa la seva entrada 'Peça_detector' i permet canviar a l'**Estat 7**.

A l'**Estat 7** es du a terme la subrutina 'Detecció':

- **Detecció:** L'autòmat rep les tres entrades dels sensors (capacitiu, òptic i inductiu) de la base de detecció, amb els que detecta el tipus de peça i la compara amb la comanda realitzada. Si la peça és l'adient, activa una variable 'peça_correcte'; depenent de si el seu valor és 0 o 1, s'envia el senyal 'p_incorrecta' o 'p_correcta'

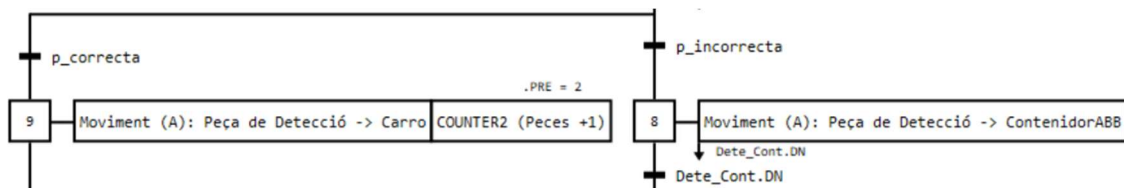


Figura 5.5: Estats 8 i 9 del GRAFCET de l'autòmat. [Font: pròpia]

al robot ABB (sent aquesta la condició per arribar a l'Estat 8 o a l'Estat 9, respectivament).

Considerant la casuística en que la peça detectada és incorrecta, a l'**Estat 8** es dona avís al robot ABB per moure la peça de la base de detecció al contenidor, on queda dipositada. Al acabar el moviment, es torna a l'Estat 4.

En el cas que la peça detectada és correcta, es passa a l'**Estat 9**. En aquest estat, es fa que el robot ABB mogui la peça fins el carro disponible a l'estació 1. Al tindre el carro dues posicions, és necessari incloure un comptador de peces (Counter 2) de .PRE= 2 amb el qual es va comptant les peces que es pugen al carro i que a la vegada, serveix per escollir entre dues trajectòries del robot ABB; la que porta la peça a la posició 1 del carro, i la que la porta a la posició 2.

En aquest estat, hi ha tres possibles transicions a diferents estats:

- Si el comptador 2 no ha arribat al seu valor de preset i la variable 'Comanda_pendent' val 1, torna a l'**Estat 5**, on es dispensa la peça pendent.
- Si el comptador 2 no arriba al seu valor de preset i la variable 'Comanda_pendent' val 0, torna a l'**Estat 4**, on s'executa la subrutina que permet afegir i gestionar les comandes.
- Si s'activa el .DN del comptador 2 (quan s'arriba al preset), torna a l'**Estat 2**, on la sortida 'Enclavament_E1' passa a ser 0, permetent el moviment del carro a la següent estació.

L'últim estat en el que participa el robot ABB és l'**Estat 10** (succeeix a l'Estat 3 quan el carro està ple, un cop ha donat la volta completa a la base de l'estació). En aquest punt, s'activa la sortida 'Enclavament_E1' per aturar el carro a l'estació i es procedeix a executar la subrutina 'Posició Magatzem':

- Posició Magatzem: Aquesta rutina serveix per ubicar les peces que el robot ABB agafa del carro i transporta al magatzem. Al tindre una capacitat limitada (25 posicions) i per no col·locar dues peces a la mateixa ubicació, s'utilitza un comptador intern que s'activa per cada vegada que es recull una peça per portar-la cap al magatzem, i amb el qual també es selecciona la trajectòria adient per la posició de magatzem indicada. Quan no queden peces al carro, s'activa una variable booleana 'fi_cicle'.

Si 'fi_cicle' està activa i no hi ha comandes pendents, l'autòmat torna a l'**Estat 4**. Si 'fi_cicle' està activa però hi ha comandes pendents, torna a l'**Estat 5**.

L'**Estat 11** és el primer dels estats de la branca que conté moviments del robot FANUC.

Aquest resetja el valor de 'Enclavament_E2', que controla l'aturada del carro a l'estació 2. També s'inclou el reset dels comptadors 3 i 4 (per a controlar el carro que es troba aturat a l'estació i el número de peces que conté, respectivament). Els comptadors funcionen per cicles $0 \rightarrow 1 \rightarrow 2$, tal com passa amb els comptadors 1 i 2.

Al estar 'Marxa' activat, els carros es mouen per la cinta, fins que al detectar-se un mitjançant el Sensor_E2, es realitza la transició d'estat cap a l'**Estat 12**.

En aquest estat, s'activa la sortida 'Enclavament_E2' per evitar el moviment del carro, es porta el compte del número de carro que es troba a l'estació (Comptador 3) i s'inicia la subrutina 'Càrrega peces al Disc':

- Càrrega peces al Disc: Aquesta subrutina consisteix en l'avís per part de l'autòmat al robot FANUC per agafar una peça i deixar-la a la posició inicial del disc. Un cop la primera peça es troba al disc, l'autòmat envia un senyal de sortida 'Rotar90' al disc per moure'l 90°. Finalitzats aquests dos passos, es torna a demanar al robot que agafi la peça restant del carro i la porti a la posició inicial del disc. Un cop realitzades aquestes accions, s'activa l'entrada de l'autòmat 'Fi_càrrega'.

Al finalitzar les operacions de la subrutina anterior i activar-se 'Fi_càrrega', es canvia a l'**Estat 13**, on s'executa la subrutina 'Mecanitzat':

- Mecanitzat: En aquest punt, s'ha de coordinar per fases les operacions. Primer de tot, el disc gira 90° (sortida Rotar90), fins deixar la primera peça al disc sota el trepant, i es procedeix a trepanar (Baixa_Trepant). Seguidament, al acabar de foradar la peça i pujar el trepant a repòs, el disc torna a girar 90°; ara hi ha la primera peça (la trepanada) sota el verificador, i la següent sota el trepant. Es duen a terme les dues operacions (Baixa_Trepant i Baixa_Verificador) i s'espera a portar ambdós dispositius a repòs abans de continuar amb un nou gir de 90°. Al realitzar-se la verificació, el sensor indica si la peça es bona o dolenta (si s'ha trepanat incorrectament) mitjançant 'Peça_noDrill'. El valor d'aquesta entrada es

guarda en diferents variables booleanes a l'autòmat dependent d'un comptador intern .PRE=2.

	Variable	ESTAT LÒGIC	
	Peça_noDrill	0	1
1 ^a Validació (peça1)	Peça1_OK	1	0
2 ^a Validació (peça2)	Peça2_OK	1	0

Taula 5.6: Taula lògica de les connexions amb Peça_noDrill. [Font: pròpia]

Un cop arriba la primera peça a la posició on originalment havia sigut depositada (quan dona la volta completa), la rutina acaba i ho indica amb la variable booleana 'Fi_mecanitzat'.

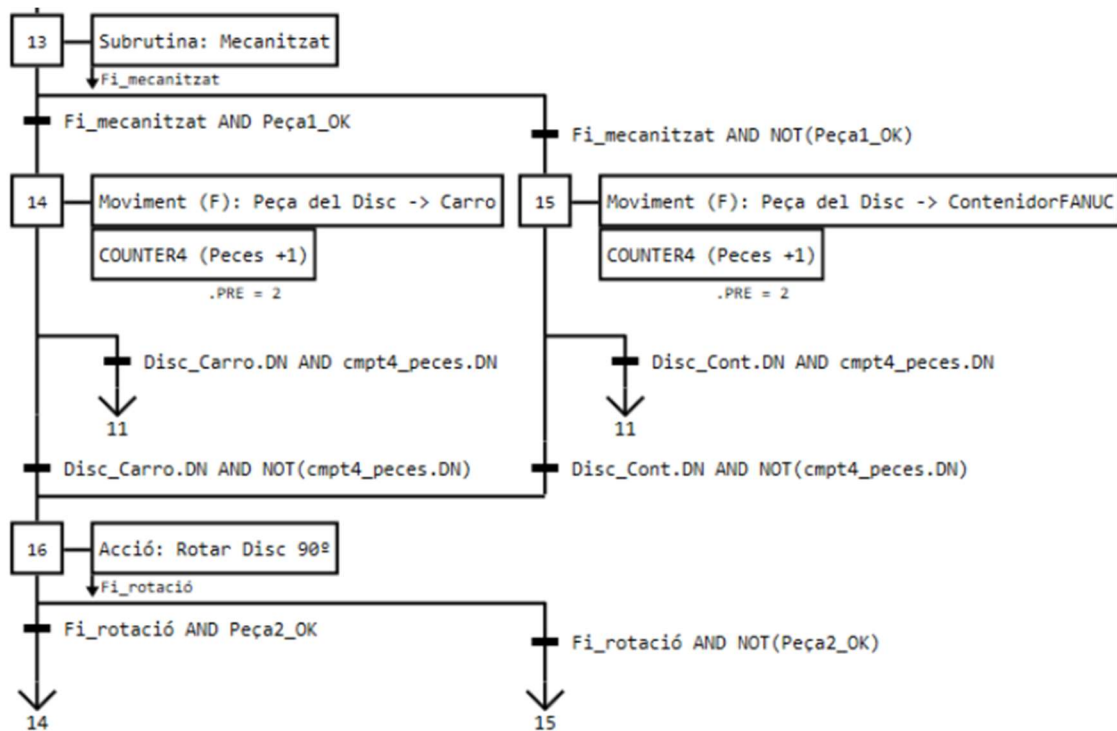


Figura 5.6: Estats 13, 14, 15 i 16 del GRAFCET de l'autòmat. [Font: pròpia]

Acabat el mecanitzat (Fi_mecanitzat) i un cop es coneix si la peça és correcta o no, es passa a l'Estat 14 o Estat 15. Les dues etapes tenen un comptador (Counter 4) de .PRE=2 que comparteixen i el qual serveix per conèixer que peça s'està tractant i per conèixer quan ja no queden més peces al disc (cmpt4_peces.DN).

A l'**Estat 14**, s'agafa la peça de la posició inicial del disc i es porta al carro. Un cop acabat el moviment (Disc_carro.DN), si el comptador 4 ha assolit el valor preset, significa que

no queden més peces al disc i, per tant, es torna a l'Estat 11, on l'Enclavament_E2 es posa a zero, alliberant el carro per circular de tornada fins l'estació 1 i el comptador 4 es reseteja.

Si, en canvi, no ha assolit el valor de preset, es passa a l'**Estat 16**, on s'acciona el gir del disc (Rotar90), quan acaba l'acció s'activa el senyal Fi_rotació i, un cop es verifica el valor de Peça2_OK, es torna a l'Estat 14 si la peça és bona, o a l'Estat 15 si la peça no és vàlida.

El cas de l'**Estat 15** implica la pèrdua d'una peça; l'autòmat encomana al robot FANUC que agafi la peça de la posició inicial del disc i la depositi al contenidor. Un cop acabat el moviment, es tenen les mateixes opcions que a l'Estat 14; o bé la primera de les peces ha sigut dolenta i s'encamina a l'Estat 16 per rotar el disc, o bé la segona peça és dolenta i per tant, al no haver més peces, es torna a l'Etapa 11 per alliberar el carro (contenint una o cap peça).

5.2. ABB IRB 120

5.2.1. Accions de l'autòmat i del programa de l'ABB IRB 120

Armament: El robot ha de posar totes les sortides (DO10_2, DO10_3 i DO10_4) a zero i obrir la pinça (DO10_1=0), així com resetejar el comptador 1 (que controla el número de peces dipositades al magatzem).

El robot ha de fer 5 moviments: Inici, Mov. del Dispensador a Detecció, Mov. De Detecció al Contenedor, Mov. De Detecció al Carro i Mov. Del Carro al Magatzem:

- Inici: Quan el robot rep un 1 a DI10_1, ha d'anar a repòs i al finalitzar el moviment ha de posar a 1 la sortida DO10_4 durant un segon i advertir a l'autòmat.

AUTÒMAT			ROBOT	
Ves a repòs (ABB_Repos)	6:OData1	→	DI10_1	Ha d'anar a repòs
Treu peça del dispensador (Treu_peça)	3:IData 11	←	DO10_4	Ja sóc a repòs (1 s activat (puja a 1) i es desactiva (torna a 0))

Taula 5.7: Comunicacions al moviment 'Inici' del robot ABB. [Font: pròpia]

- Moviment del Dispensador a Detecció: Quan el robot rep un 1 a la DI10_2, ha d'anar a recollir la peça del final de base del dispensador, portar-la fins la base de detecció de peça i retirar-se a una distancia de seguretat per evitar problemes en la detecció.

Quan acaba, el robot posa a 1 la sortida DO10_2 durant un segon i advertir a l'autòmat.

AUTÒMAT			ROBOT	
<i>Peça al final de la base del dispensador</i> <i>(Peça_dispensada)</i>	6:OData2	→	DI10_2	Agafa peça i la posa a la base de detecció
Detecta <i>(Peça_detector)</i>	3:IData 12	←	DO10_2	<i>Ja he deixat la peça i m'he separat una mica</i> (1 s activat (puja a 1) i es desactiva (torna a 0))

Taula 5.8: Comunicacions al moviment 'Disp_Detei' del robot ABB. [Font: pròpia]

En aquest instant, els moviments i accions entre l'autòmat i el robot ABB canvien depenent del tipus de peça detectada. Si és incorrecta, el robot ha de moure la peça al contenidor, i si és correcta, l'ha de portar al carro.

- Moviment de Detecció al Contenidor: Quan el robot rep un 1 a la entrada DI10_3, ha de agafar la peça a la base de detecció i portar-la fins el contenidor. Després, torna a repòs.

AUTÒMAT			ROBOT	
<i>Peça incorrecta</i> <i>(p_incorrecta)</i>	6:OData3	→	DI10_3	Va a llençar-la al contenidor I després va a repòs

Taula 5.9: Comunicacions al moviment 'Dete_Cont' del robot ABB. [Font: pròpia]

- Moviment de Detecció al Carro: Quan el robot rep un 1 a la entrada DI10_4, ha de agafar la peça a la base de detecció i portar-la fins el carro. Després, torna a repòs.

AUTÒMAT			ROBOT	
<i>Peça correcta (p_correcta)</i>	6:OData8	→	DI10_4	Va a portar la peça al carro i després va a repòs
<i>Peça al carro</i>	3:IData13	←	DO10_3	<i>Ja he deixat la peça al carro i vaig a repòs (1s)</i>

Taula 5.10: Comunicacions al moviment 'Dete_Carro' del robot ABB. [Font: pròpia]

Un cop arriba el carro a l'estació 1 ple o semiplè, s'ha de programar un nou moviment per guardar les peces.

- Moviment del Carro al Magatzem: Quan el robot rep un 1 a la entrada DI10_6, ha d'agafar les peces disponibles al carro i portar-les al magatzem. Després, torna a repòs.

AUTÒMAT			ROBOT	
<i>Ves a agafar les peces del carro (Recull_peça)</i>	6:OData3	→	DI10_6	Va a deixar les peces disponibles al magatzem (i després va a repòs)

Taula 5.11: Comunicacions al moviment 'Carro_Magatzem' del robot ABB. [Font: pròpia]

5.2.2. GRAFCET del robot ABB IRB 120

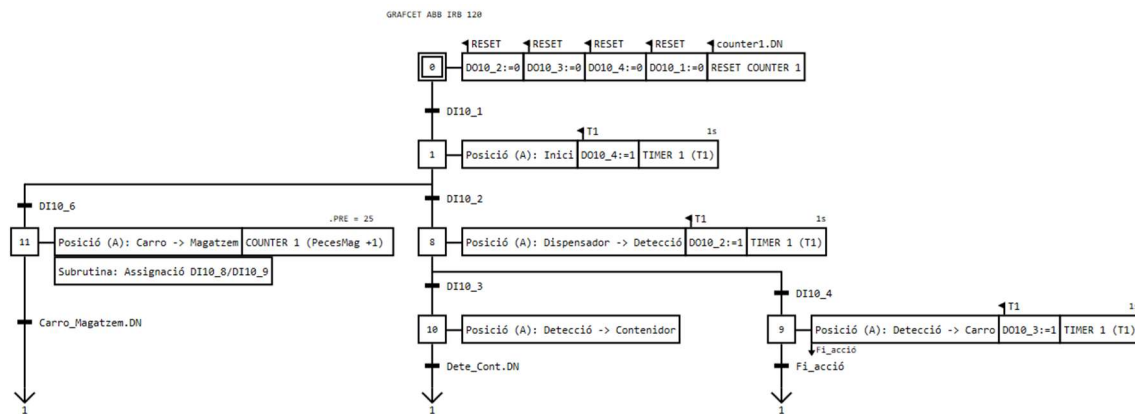


Figura 5.7: GRAFCET del robot ABB. [Font: pròpia]

[NOTA: Es pot trobar una versió amb major resolució inclosa a l'annex VI]

El GRAFCET de la figura 5.7 és el resultat d'adjuntar els anteriors moviments i donar-li un sentit lògic concatenant-los. Té un total de 6 estats i 5 moviments principals; els moviments 'Detecció → Carro' i 'Carro → Magatzem' tenen diferents variants depenent en el primer cas si es mou la primera peça al carro o ja n'hi ha una (de l'elecció s'encarrega l'autòmat mitjançant el comptador 2), i en el segon cas, el moviment es divideix en dues parts: la recollida del carro (que depèn de la posició on sigui la peça) i la posició al magatzem on es vol deixar (hi ha un total de 25 caselles disponibles).

Si bé els moviments o posicions, així com les transicions entre estats s'han comentat anteriorment, s'aprofita aquest apartat per detallar el funcionament de la subrutina 'Assignació DI10_8/DI10_9'.

- Assignació DI10_8/DI10_9: Aquesta subrutina és tan simple com enllaçar les sortides de l'autòmat Peça_pos1 i Peça_pos2 amb les entrades del controlador del robot DI10_8 i DI10_9.

Altre punt a tenir present, és que per moure la peça durant una trajectòria, s'ha de tancar la pinça a la ubicació d'inici i obrir-la a la ubicació final. És per això que, durant l'execució dels moviments, la sortida DO10_1 va modificant el seu valor a 1 o 0 depenent de l'ordre que es requereix realitzar (obrir i tancar, respectivament).

5.3. FANUC LR Mate 200 iC/4S

5.3.1. Accions de l'autòmat i del programa del FANUC LR Mate 200

Armament: El robot ha de posar la sortida (DO[111]) a zero i obrir la pinça (RO[1]=0, RO[2]=0).

El robot ha de fer 5 moviments: Inici, Peça pos.1 del Carro al Disc, Peça pos.2 del Carro al Disc, Peça del Disc al Carro i Peça del Disc al Contenidor:

- Inici: Quan el robot rep un 1 a DI[107], ha d'anar a repòs i esperar l'activació de DI[101] per part de l'autòmat.

AUTÒMAT			ROBOT	
<i>Ves a repòs</i> (FANUC_repòs)	5:OData6	→	DI[107]	Ha d'anar a repòs

Taula 5.12: Comunicacions al moviment 'Inici' del robot FANUC. [Font: pròpia]

- Peça a la posició 1 del Carro al Disc: Quan el robot rep un 1 a l'entrada DI[101] s'ha de moure per recollir la peça a la posició 1 del carro i col·locar-la al disc.

AUTÒMAT			ROBOT	
<i>Agafa la peça del carro</i> (Recollir_peça)	5:OData7	→	DI[101]	Mou la peça de la pos.1 del carro a la pos. inici del disc

Taula 5.13: Comunicacions al moviment 'pos1_Disc' del robot FANUC. [Font: pròpia]

- Peça a la posició 2 del Carro al Disc: Amb la mateixa activació de l'entrada DI[101] i un cop finalitzat el moviment de l'anterior peça al disc, el robot s'ha de moure per recollir la peça a la posició 2 del carro i col·locar-la al disc.

AUTÒMAT			ROBOT	
<i>Agafa la peça del carro</i> (Recollir_peça)	5:OData7	→	DI[101]	Mou la peça de la pos.1 del carro a la pos. inici del disc

Comença a mecanitzar (Fi_càrrega)	3:IData1	←	DO[111]	<i>Avisa que ambdues peces estan situades al disc i torna a repòs. (1s)</i>
--------------------------------------	----------	---	---------	---

Taula 5.14: Comunicacions al moviment 'pos2_Disc' del robot FANUC. [Font: pròpia]

- Peça del Disc al Carro: Quan el robot rep una entrada DI[105] de valor 1, ha de agafar la peça de la posició inicial del disc i portar-la al carro.

AUTÒMAT			ROBOT	
<i>Torna la peça al carro</i> (Peça_bona)	5:OData6	→	DI[105]	Porta la peça de la posició inici del disc al carro

Taula 5.15: Comunicacions al moviment 'Disc_Carroi' del robot FANUC. [Font: pròpia]

- Peça del Disc al Contenidor: Quan el robot rep una entrada DI[103] de valor 1, ha de agafar la peça de la posició inicial del disc i abocar-la al contenidor de peces no vàlides.

AUTÒMAT			ROBOT	
<i>Descarta la peça</i> (Peça_dolenta)	5:OData8	→	DI[103]	Porta la peça de la posició inici del disc al contenidor

Taula 5.16: Comunicacions al moviment 'Disc_Cont' del robot FANUC. [Font: pròpia]

5.3.2. GRAFCET del robot FANUC LR Mate 200 Ic/4S

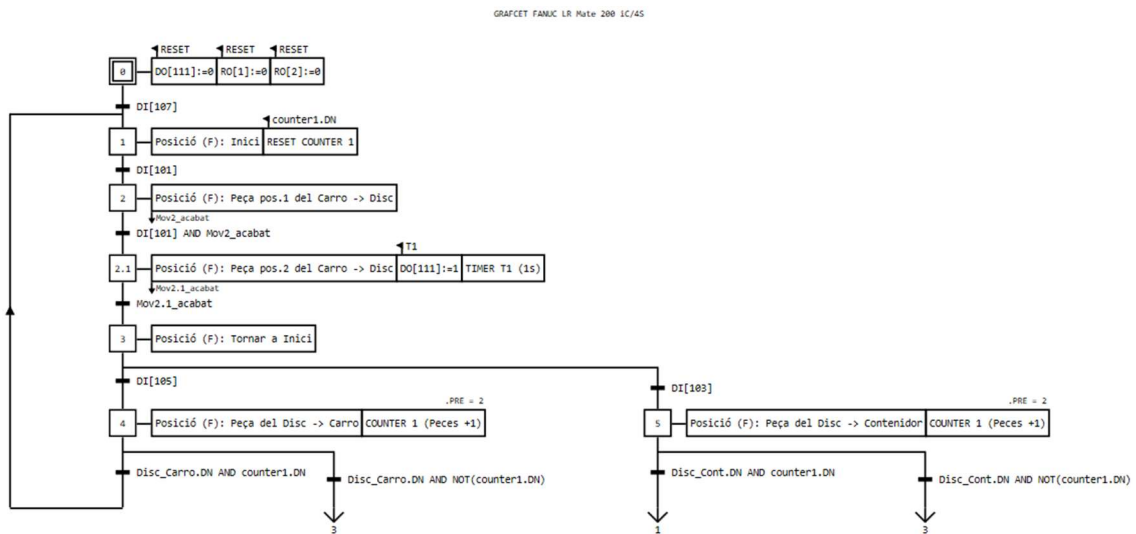


Figura 5.8: GRAFCET del robot FANUC. [Font: pròpia]

[NOTA: Es pot trobar una versió amb major resolució inclosa a l'annex VII]

El GRAFCET de la figura 5.8 és el resultat d'adjuntar els anteriors moviments i donar-li un sentit lògic concatenant-los. Té un total de 7 estats i 5 moviments principals; el moviment 'Inici' i 'Tornar a Inici' són idèntics.

El comptador 1 apareix als estats 4 i 5 i serveix per portar el compte de quantes peces s'han verificat al disc. Per exemple, en el cas de que ambdues peces siguin dolentes, aquest comptador permet saber que han passat dues peces pel verificador encara que no hagin arribat al carro, per així posar-lo en marxa encara que estigui vuit.

El timer 1 s'inclou a l'etapa 2.1 per activar i desactivar la sortida DO[111] en el temps establert (1s).

Tal com passa amb el robot ABB, per moure la peça durant una trajectòria, s'ha de tancar la pinça del robot FANUC a la ubicació d'inici i obrir-la a la ubicació final. És per això que, durant l'execució dels moviments, les sortides RO[1] i RO[2] van modificant el seu valor a 1 o 0 depenent de l'ordre que es requereix realitzar. En aquest cas, cada sortida controla l'apertura d'un costat de la pinça, així que per obrir 'completament' les dues han de valdre 0, i per tancar la pinça, el valor de RO[1] i RO[2] ha de ser 1.

6. Disseny de la simulació

Per dur a terme la simulació, les dues opcions més realistes per al projecte eren els softwares Robotstudio i Roboguide, al ser aquests programes propietat d'ABB i FANUC, fabricants dels models de robot que incorpora la cèl·lula. Ambdós softwares tenen una versió de prova, si bé la versió de Robotstudio és més accessible per descarregar des de la web d'ABB a diferència de la de Roboguide, per la qual cal contactar i demanar-la a la companyia FANUC.

S'ha escollit el software Robotstudio 6.08 degut a que és un programa ja conegut, més complet que el de la competència i amb el qual es pot programar de forma senzilla el robot ABB IRB 120 amb llenguatge RAPID gràcies a les seves similituds amb C+.

El principal inconvenient a l'hora d'escollir un entorn de treball on treballen robots de diferents companyies és la incompatibilitat del codi de programació; en aquest cas, el codi RAPID del robot FANUC no és extrapolable pel controlador del LR Mate 200 iC/4S i, per tant, per portar a terme el disseny, s'ha decidit substituir-lo per un altre robot IRB 120.

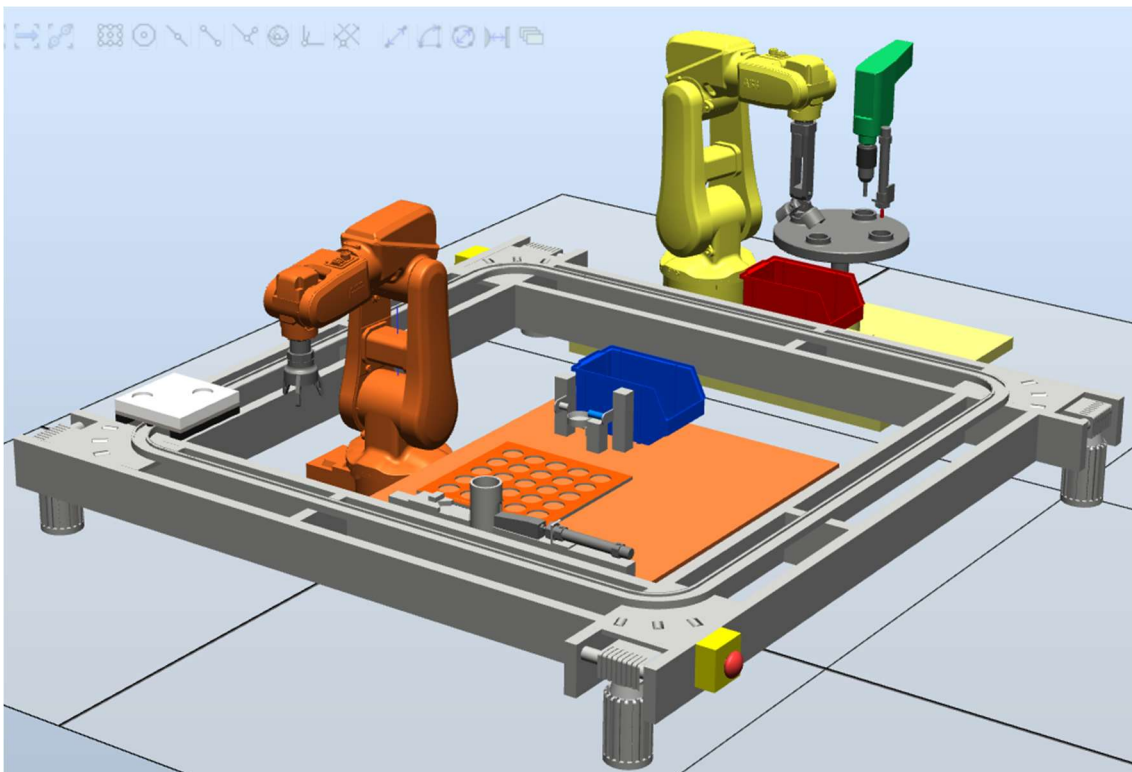


Figura 6.1: Visió completa de la cèl·lula de fabricació de la simulació. [Font: pròpia]

La complexitat de codificació per portar a terme la simulació amb dos carros condiciona el disseny a un únic carro. D'aquesta forma, simplificant l'operativa de la cèl·lula en parts més petites, s'observen millor els passos que ha de realitzar una peça des de que es realitza la comanda fins que arriba finalment al magatzem.

6.1. Programació de la simulació

Per portar a terme la simulació, és necessari incloure dos sistemes o controladors, un per cada robot.

Això comporta que cada robot tingui la seva pròpia programació independent, així com les trajectòries necessàries per dur a terme l'aplicació.

Els passos a seguir per programar la simulació des del programa Robotstudio són els següents:

- (1) Crear una nova estació buida (Archivo → Nuevo → Estación vacía).
- (2) Importar els objectes o Smart Components que configuren la cèl·lula arrossegant-los cap a l'estació i ubicar-los en el lloc on correspongui (botó dret sobre l'objecte → Posición).
- (3) Importar el robot desitjat (Posición inicial → Biblioteca ABB o Importar biblioteca).
- (4) Afegir els controladors als robots (Posición inicial → Sistema de robot → Nuevo Sistema).
- (5) Definir bases de coordenades per cada sistema (Posición inicial → Base de coordenades).
- (6) Importar o dissenyar les eines dels robots (si la geometria de l'eina és complexa, es recomana utilitzar softwares alternatius més potents; les eines de disseny gràfic no són molt potents).
- (7) Crear els punts on ha d'anar el robot. És convenient en aplicacions tipus Pick-and-Place situar un punt allunyat un cert offset sobre la posició on el robot a de recollir o deixar peça per baixar amb seguretat. (Posición inicial → Punto → Crear punto). Els icones translúcids a la part superior de la pantalla de l'estació poden ser d'ajuda per seleccionar parts, objectes... o com en aquest cas, per ubicar els punts amb major precisió (ajustant a objectes, a centres...). Altre característica útil és

- ‘Ver el robot en el punto’ (es troba a Posición inicial → Programación de trayectorias), que permet observar si el punt està dintre de l’abast del robot.
- (8) Formar trajectòries amb els punts (es seleccionen els punts convenients → botó dret → Nueva trayectoria). Es pot ajustar com de precís ha de passar la trajectòria per cada punt i la seva velocitat configurant-lo clicant botó dret sobre el punt en qüestió.
- (9) Carregar les trajectòries a RAPID (botó dret al controlador → Sincronizar con RAPID).
- (10) Crear les entrades i sortides necessàries per cada controlador (Controlador →

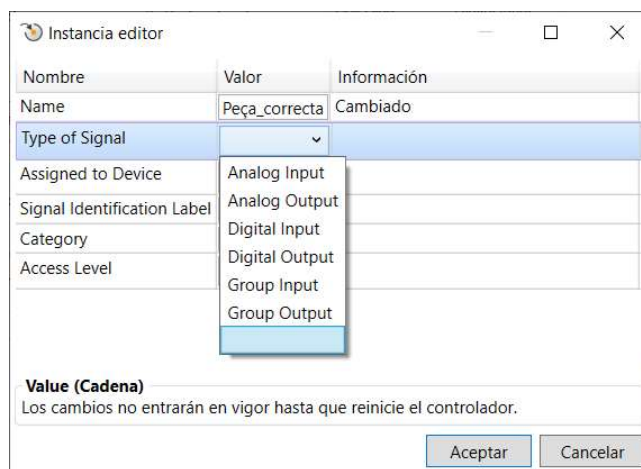


Figura 6.2: Finestra que apareix al crear una nova senyal. [Font: pròpia]

Configuración → I/O System → botó dret sobre Signal → Nuevo Signal). S’obre una finestra on es pot escollir el tipus de senyal que es vol introduir. És indispensable reiniciar els controladors al afegir o eliminar senyals per assimilar els canvis (part superior, a ‘Herramientas de controladores’).

- (11) Dissenyar la lògica de simulació (Simulación → Configurar → Lógica de simulación).
- (12) Disseny del programa RAPID. Si s’està editant, tot nou canvi que es vulgui incorporar a la simulació ha de ser aplicat (RAPID → Controlador → Aplicar)
- (13) Sincronitzar el programa RAPID (RAPID → Acceso → Sincronizar → Sincronizar con Estación).
- (14) Configurar la simulació. Permet actualitzar els canvis d’objectes a l’estació. (Simulación → Configurar → Configuración de simulación).
- (15) Reproduir (Simulación → Control de simulación → Reproducir). Des d’aquest mateix panell es pot gestionar l’execució de la simulació, podent pausar o aturar. És convenient, si s’utilitzen sistemes complexos o es requereix alguna activació de senyal manual, utilitzar el simulador d’E/S.

6.2. Elements de la simulació

Hi ha dos tipus de components al disseny de la simulació; actius (participen en el moviment de les peces o es desplacen), i passius (són estàtics, no reactius).

Majoritàriament, al ser una simulació, es requereixen una sèrie de sensors extra per facilitar determinats processos com el de distribució de les peces que en el sistema real no són necessaris. Es per aquest motiu que la majoria d'elements del disseny són actius, sent la pròpia base de la cèl·lula l'element passiu més important. La base té una mida de 1500x1500mm i disposa de quatre motors per posar en marxa les cintes que faran que el moviment entre estacions dels carros sigui possible. També té una guia per la trajectòria dels carros.

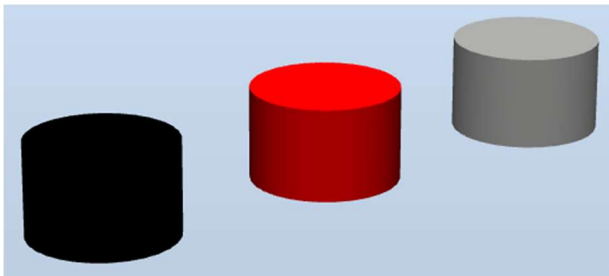


Figura 6.3: Peces disponibles a la simulació. [Font: pròpia]

Les peces disponibles del sistema mesuren 40mm de diàmetre i 25mm d'altura, i estan disponibles en color negre, vermell o acabat en alumini.

Quan es requereix una interacció entre un element del sistema i altre element o el propi controlador, és necessari convertir aquests elements en Smart Components, els quals disposen de entrades i sortides i contenen una lògica interna pròpia, independent de la programada en RAPID (comparteixen I/Os, treballen simultàniament i un canvi en la codificació pot comportar un canvi al funcionament del Smart Component).

Els Smart Components que formen part del disseny de la cèl·lula, ordenats de manera que segueixin l'ordre lògic d'aparició a l'execució, són els següents:

- Dispensador (SC_Feeder)
- Base de detecció del tipus de peça (SC_Sensor_Peça)
- Caixes de peces indesitjades (SC_Caixa/SC_CaixaF)
- Carro de transport (SC_Carro)
- Disc rotatori (SC_Disc)
- Trepant (SC_Trepant)
- Verificador (SC_Verificador)

6.2.1. Dispensador

Està format per una base amb un allotjament vertical per introduir les peces i un cilindre pneumàtic per fer-les arribar a la ubicació final de la base segons convingui dependent de l'activació d'una sortida digital de l'autòmat.

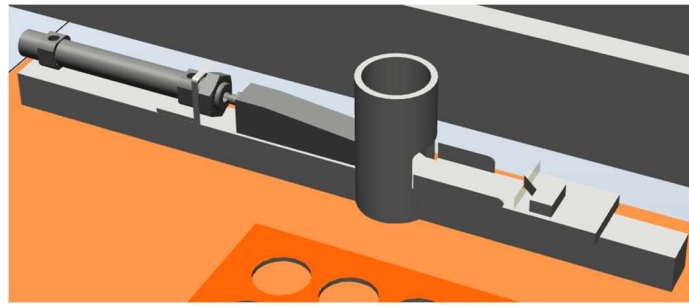


Figura 6.4: Detall del dispensador de la simulació. [Font: pròpia]

Encara que la seva funcionalitat és bàsica, per dissenyar-lo a la simulació s'ha de tenir present que l'entrada no és manual. Es per això que s'inclouen 3 entrades digitals (VERMELLA, ALUMINI I NEGRE) perquè l'usuari pugui escollir la peça a introduir a la simulació. També disposa d'una última entrada (Marxa) per activar la simulació del moviment de la peça cap a la posició final de la base. La única sortida digital d'interès del component és DO_POSICIO, que indica precisament l'aturada de la peça.

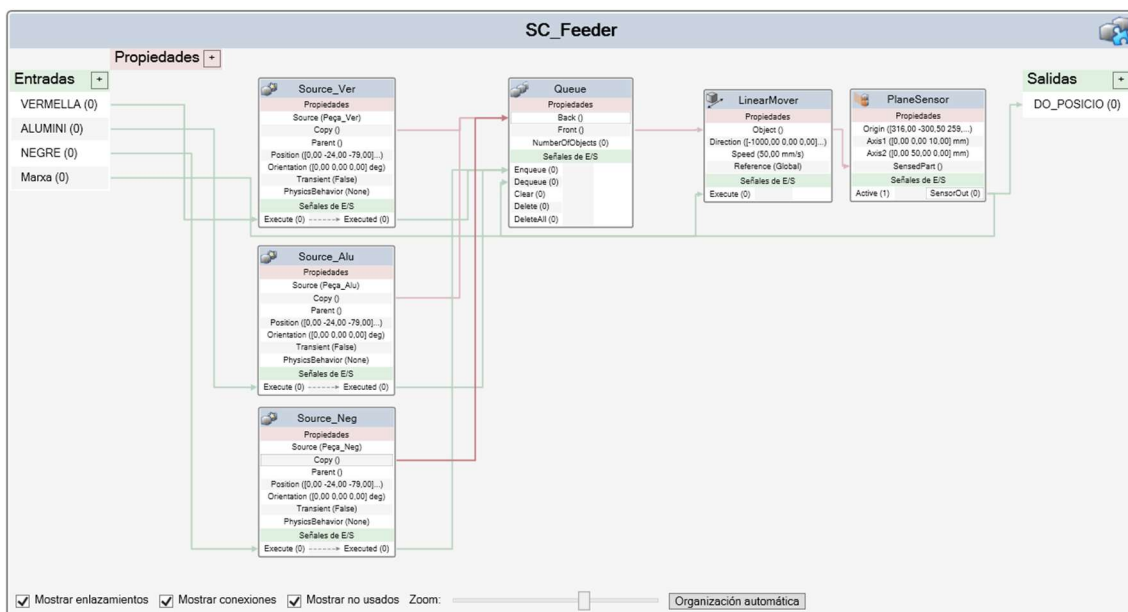


Figura 6.5: Lògica interna del SC_Feeder. [Font: pròpia]

Internament, el component està compost per diversos components subordinats.

Hi ha tres Sources, els quals generen còpies de les peces desitjades per l'usuari (quan activa les DI) i s'apilen a un Queue, o cua. Al activar la marxa del dispensador, el component agafa l'última peça de la cua i la transporta amb un LinearMover fins arribar al PlaneSensor convenientment situat al final de la base, on s'atura el moviment i s'elimina la peça de la cua.

6.2.2. Base de detecció

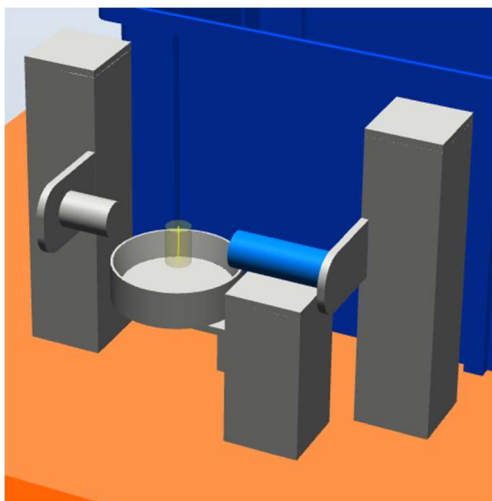


Figura 6.6: Detall de la base de detecció. [Font: pròpia]

La base de detecció de peces està equipada amb tres sensors diferents (capacitiu, òptic i inductiu).

Depenent del conjunt de sortides dels sensors, es pot identificar quin tipus de peça està a la base, i si és o no la sol·licitada per l'usuari.

El sensor capacitiu funciona com un detector de proximitat, per saber si hi ha peça. El sensor inductiu permet distingir la peça d'alumini de les altres dues. Per diferenciar les dues peces de plàstic de color negre i vermell, s'utilitza el

sensor òptic, ja que al tenir la peça negra un grau de reflectivitat baix, no es detectada pel sensor.

Tipus de peça	Negre (plàstic)	Vermella (plàstic)	Alumini (metall)
Sensor capacitiu	1	1	1
Sensor inductiu	0	0	1
Sensor òptic	0	1	1

Taula 6.1: Lògica de detecció de la peça segons l'activació dels sensors. [Font: pròpia]

En aquest cas, Robotstudio proporciona subcomponents de molta utilitat i es poden simular directament tres sortides digitals booleanes que indicaran el tipus de peça detectada.

A la base s'ha inclòs un LineSensor, que detecta la peça. Amb el GetParent, s'aconsegueix obtenir el Parent a partir del Child; les peces vermelles, negres o d'alumini com objectes són difícils de comparar, ja que al generar-les al dispensador copiant-les d'una peça origen, és un nou element per si mateix amb un altre nom i no les identificaria com a iguals. Per aquest motiu, al disseny s'ha classificat les peces en grups, de major a menor jerarquia, com per exemple: Peces > Peces vermelles > PeçaVer4.

En aquest cas, no es pot comparar la PeçaVer4 amb una possible PeçaVer5, però si ambdues es troben al mateix grup 'Peces Vermelles', llavors comparteixen Parent i podem afirmar que són del mateix tipus.

Això mateix és el que es duu a terme amb els ObjectComparer per aconseguir discriminar el tipus de peça a la base.

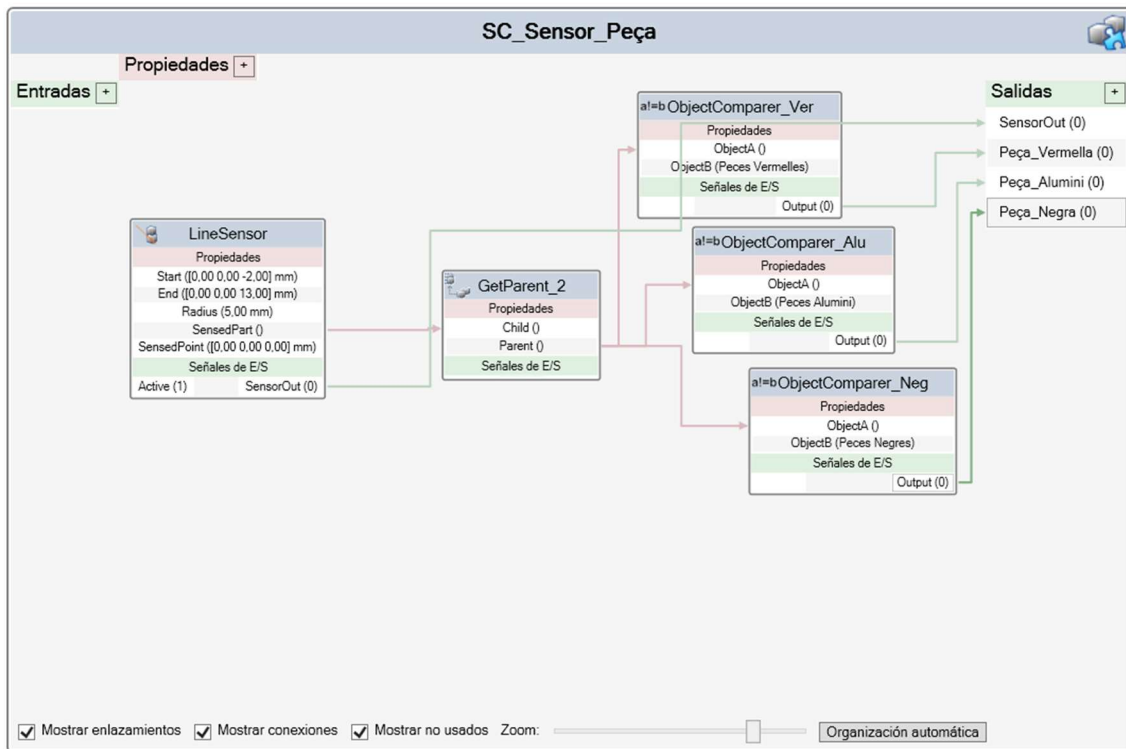


Figura 6.7: Lògica interna del SC_Sensor_Peça. [Font: pròpia]

6.2.3. Caixa de peces

Bàsicament, és on s'ubiquen les peces no desitjades que es descarten durant l'execució de l'aplicació.

Al llarg del recorregut per la cèl·lula de les peces, una caixa es troba al costat del detector de tipus de peça i l'altre, entre l'estació 2 i el mòdul rotatori, per descartar les peces de tipus incorrecte i a les que el trepat no ha sigut satisfactori, respectivament.

Al món real, el robot corresponent es situa en un punt a sobre de la caixa, obre la pinça i deixa caure la peça. En el cas de la simulació, i degut a

l'impossibilitat d'afegir físiques a la peça per fer-la caure a la caixa (funció no disponible a la versió 6.08 de Robotstudio), s'ha simulat la caiguda.

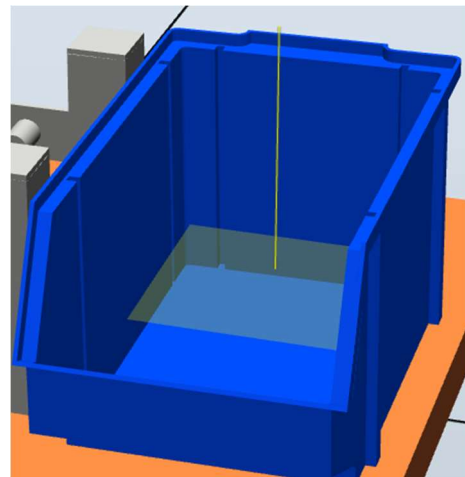


Figura 6.8: Detall del contenidor. [Font: pròpia]

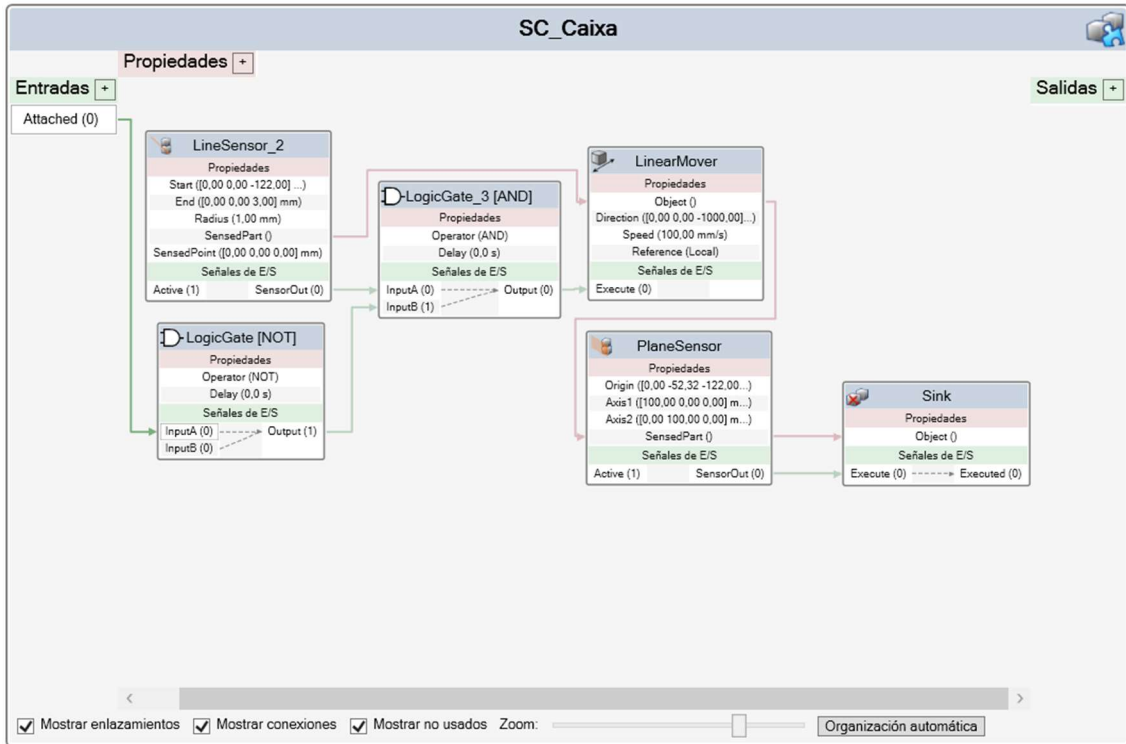


Figura 6.9: Lògica interna del SC_Caixa [Font: pròpia]

Amb una entrada digital anomenada 'Attached' (que proveeix un enllaç amb la pinça corresponent), s'aconsegueix que, quan la peça és detectada i la pinça la ha alliberat (Attached = 0), s'actua el LinearMover fins que el moviment de la peça arriba al PlaneSensor al fons de la caixa i activa el subcomponent Sink, que elimina l'objecte.

6.2.4. Carro de transport

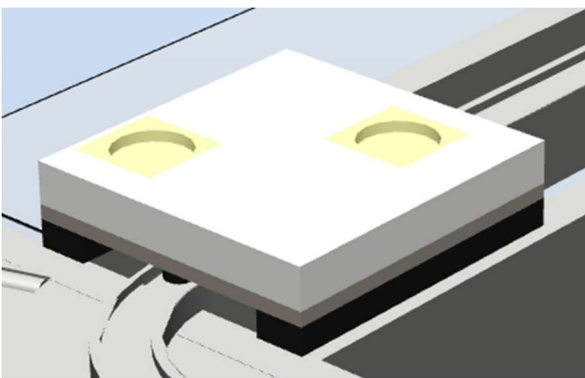


Figura 6.10: Detall del carro de transport. [Font: pròpia]

El carro de transport és un element fonamental al sistema. S'encarrega de portar les peces d'una estació a una altre. En el disseny físic, la base de l'estació disposa de dos sensors inductius amb enclavament, que al detectar l'arribada del carro, s'activen i eviten el seu moviment. Aquests punts de la base de la cèl·lula s'han decidit nombrar com Estació 1 (on opera el robot ABB) i Estació 2 (on opera el robot FANUC).

A la part superior, el carro disposa de dos allotjaments per transportar fins a dues peces a la vegada evitant pèrdues durant el trajecte.

El disseny del Smart Component és complex i és pot dividir en dues parts; la primera, necessària per fixar la peça al carro i fer-la moure simultàniament amb el carro (part esquerra de la figura 6.11), i la segona, soluciona el moviment del carro entre estacions (part dreta de la figura 6.11).

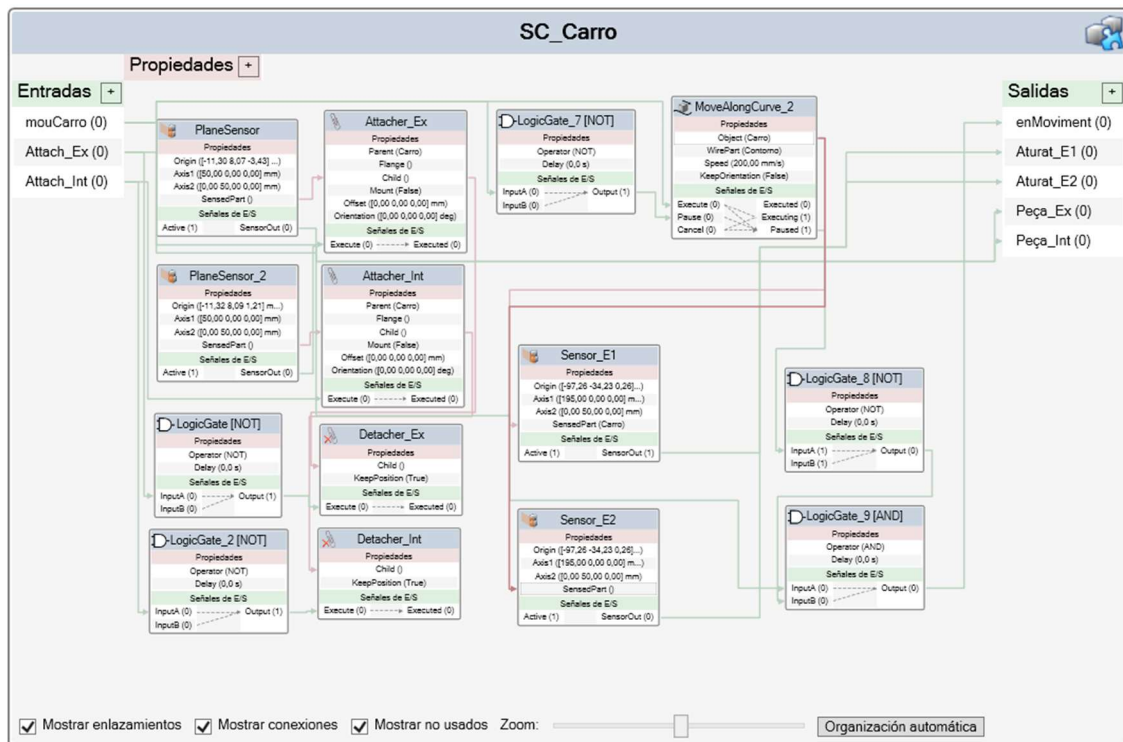


Figura 6.11: Lògica interna del SC_Carro. [Font: pròpia]

El carro simulat incorpora dos PlaneSensor connectats al propi carro (significa que es mouen de forma coordinada amb el carro, mantenint la mateixa posició respecte l'origen local del carro). Aquests sensors, un cop activades les DI Attach_Ex o Attach_Int, els Attacher permeten 'fixar' la peça exterior o interior al carro, respectivament. En aquestes condicions, una vegada el carro estigui aturat en alguna de les dues estacions i es requereixi que el corresponent robot agafi les peces, s'atorguen valors 0 a les dues entrades per activar els Detacher i alliberar la peça del carro.

Amb l'entrada digital 'mouCarro' s'activa o es pausa el moviment del carro. La trajectòria del moviment ve definida pel MoveAlongCurve, al qual s'ha afegit un contorn tipus Wire del circuit que ha de realitzar el carro.

Com és important monitoritzar la posició i estat del carro, s'extreuen sortides digitals com 'Aturat_E1' i 'Aturat_E2' per conèixer en quina estació està aturat el carro, o 'Peça_Ex'

i 'Peça_Int', que indiquen si els allotjaments del carro estan plens, falta alguna peça o estan buits.

6.2.5. Disc Rotatori

El disc rotatori és el component principal i base del mòdul rotatori de mecanitzat i validació. Té quatre posicions per ubicar peces i permet realitzar girs de 90° fins donar la volta completa al mòdul.

En el disseny físic, el plat del disc té un diàmetre exterior de 300mm i 25mm de gruix.

A les dues últimes posicions de gir s'hi troba el trepant i el verificador.

Per simular el disc, s'han sensoritzat totes les posicions del disc amb un total de 4 sensors connectats al disc, pel que mantenen la seva

posició respecte l'objecte disc. Pot semblar d'inici que aquest pas no és necessari, al únicament ocupar dues posicions dels disc quan es col·loquen peces al disc, però no es així. Per visualitzar-ho amb un exemple, si es suposa que el robot FANUC carrega les dues peces al disc, i aquestes fan el cicle complet correctament, quan el robot deixa l'última de les peces al carro que estigui aturat a l'estació 2, el disc ja no torna a girar fins a carregar una nova peça. Això significa que en el cas d'utilitzar només dos sensors, l'última peça de la segona carrega al disc no seria detectada. Aquest succés passa a cada nou cicle de carrega de peces.

És vital detectar la peça; el funcionament del moviment en el disc és diferent del tractat anteriorment amb el carro. El funcionament és més bàsic i ofereix una alternativa de programació per moviments coordinats.

En aquest cas, l'Smart Component té una única entrada 'Rotate90', amb la qual es dona l'ordre de girar 90° el disc, i una sortida 'EnMarxa', que indica amb un 1 al controlador quan està en marxa el disc en temps real.

A cada senyal de 'Rotate90', s'activen els PlaneSensor (P1, P2, P3 i P4) i el disc es mou al executar-se el 'Rotator_Disc', component rotatori on s'estableix un gir de -90° i un

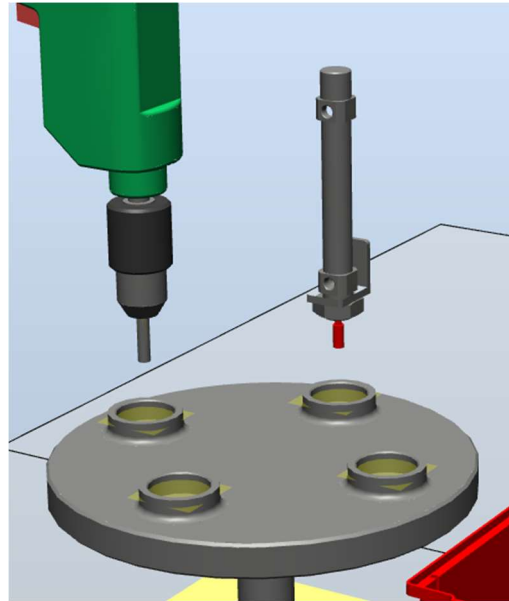


Figura 6.12: Detall del disc rotatori. [Font: pròpia]

temps de moviment de 2 segons. Si mentre es mou el disc, algun dels sensors detecten una peça, s'executa el corresponent 'Rotator_Peça', que té les mateixes condicions de gir que el propi disc, veient-se a la simulació el moviment sincronitzat.

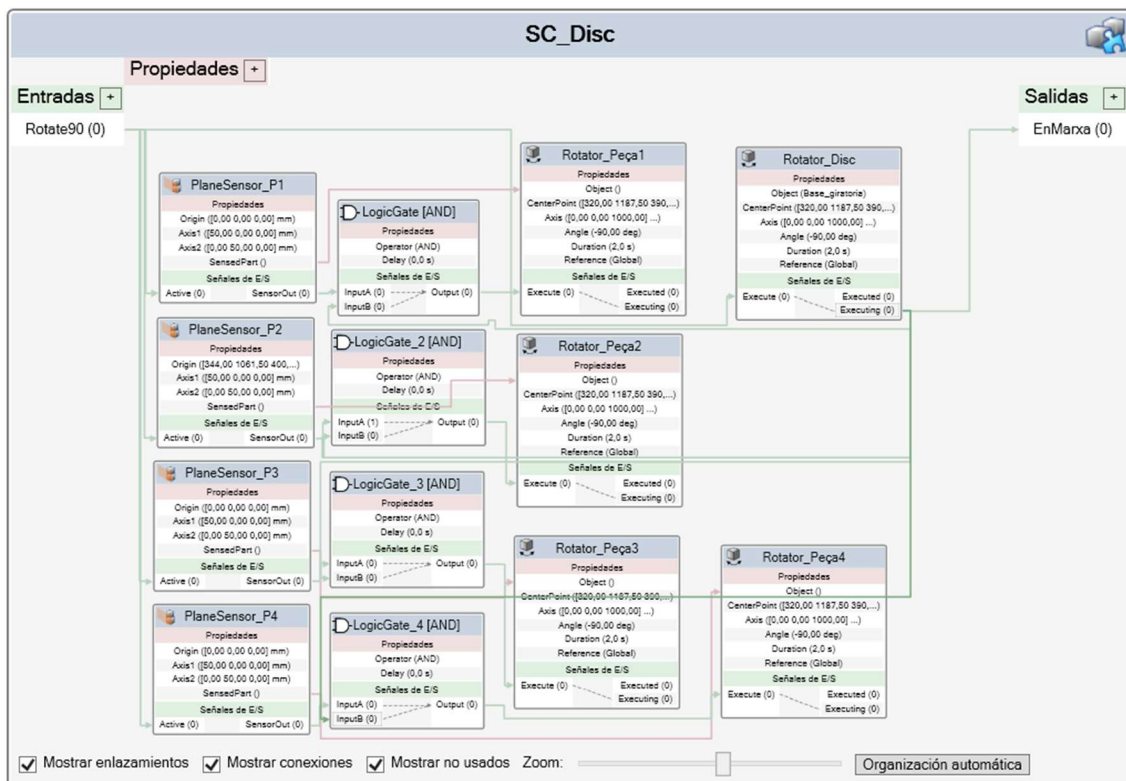


Figura 6.13: Lògica interna del SC_Disc. [Font: pròpia]

D'aquesta forma no és necessari utilitzar els Attacher i Detacher, s'evita tenir que emprar més entrades i en aquest cas, com s'explicarà més endavant, facilita l'execució de les operacions al mòdul (sobretot de la simulació del trepanat).

6.2.6. Trepanat

El trepanat és l'eina escollida per mecanitzar la peça (fer-li un forat al centre).

En el sistema real, el trepanat està acoblat a una base que es desplaça verticalment, fent pujar o baixar el trepanat. Per controlar el funcionament del trepanat, són necessàries com a mínim dues senyals d'accionament; una per l'activació del gir del trepanat, i altre per desplaçar-ho de d'alt a baix.

Per a la seva simulació, i a fi de millorar la representació gràfica i seguir utilitzant nous elements de Robotstudio, s'ha decidit incloure al disseny els components necessaris per realitzar el trepanat i que el canvi sigui visible.

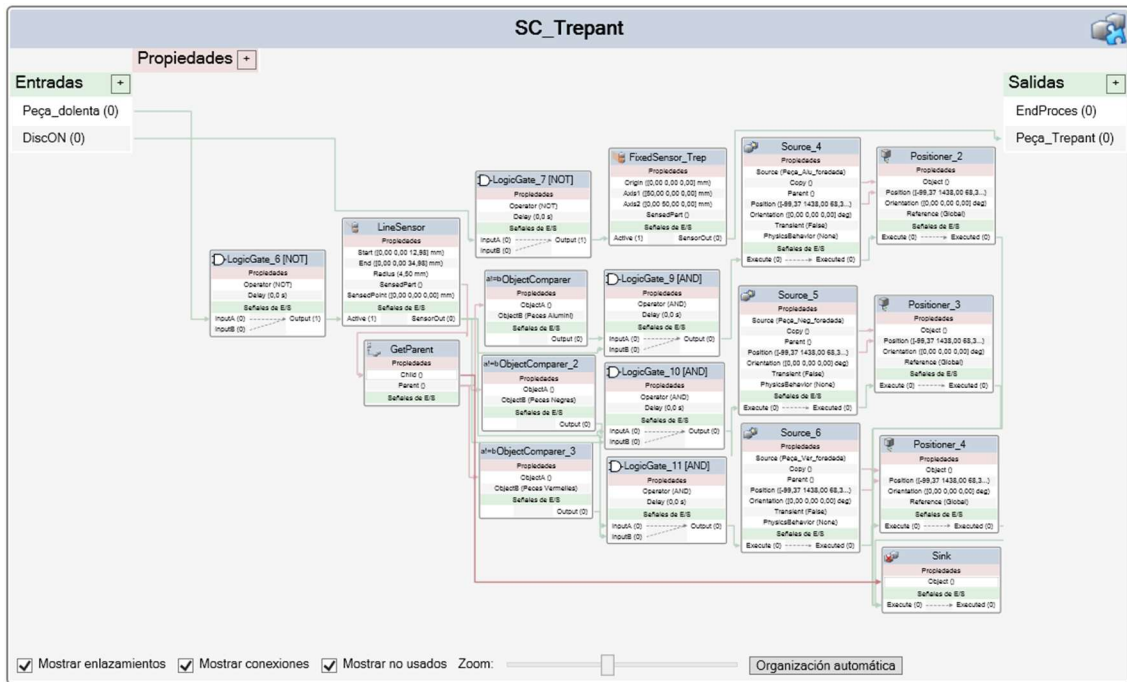


Figura 6.14: Lògica interna del SC_Trepant. [Font: pròpia]

En aquest cas, el trepant s'ha convertit en una eina per simular mitjançant el gestor d'esdeveniments el seu moviment: si la sortida del controlador 'Trepant_ON' és 0, el trepant es troba en la seva posició inicial (més elevada), i si val 1, en la seva posició de final de cursa. Al crear l'eina es poden generar posicions i vincular-les a canvis a les entrades/sortides, afegint un temps de transició entre elles.

Les entrades de l'Smart Component són 'Peça_dolenta' (si l'usuari estableix el valor d'aquesta variable a 1 durant l'execució de la simulació, les peces no es foradaran i seran dolentes) i DiscON (es troba activa mentre s'està duent a terme la rotació del disc).

Les sortides del component són 'EndProces' i 'Peça_Trepant', que està connectada al sensor fix sota el trepant; aquest sensor permet conèixer si hi ha peça a foradar o no.

Al propi trepant, se l'ha afegit un LineSensor a la punta de la broca. La funció del sensor és que, depenent si la peça és bona o no, s'activa permetent el trepat; es canvia d'objecte quan el trepant està a la seva posició més baixa (s'ha dissenyat noves peces de cada color amb un forat al centre). Seguint amb el procés de canvi de peça, s'obté el Parent de la peça a foradar i es compara amb els diferents grups de peces disponibles (tal com s'ha fet a la base de detecció). Un cop coneixem el color de la peça que s'està foradant, amb un Source es fan còpies del nou objecte amb forat i amb un Positioner es situa a la posició absoluta on estava la peça anterior (aquests components es necessiten incloure per cada

color). L'últim pas, és eliminar la peça antiga sense forat amb un component Sink per evitar deteccions desitjades al voler moure la peça pel disc.

Tal com s'ha comentat a l'apartat anterior, on es parla del component intel·ligent 'SC_Disc', el fet de no utilitzar Attachers ni Detachers facilita l'operativa de creació de peça, al no tenir que fer un Detach de la peça antiga i després un Attach amb la nova. Amb la planificació de component actual, al realitzar la substitució de peça, els sensors de cada posició del disc, la detecten i la fan girar, independentment si és l'objecte amb forat o no.

6.2.7. Verificador

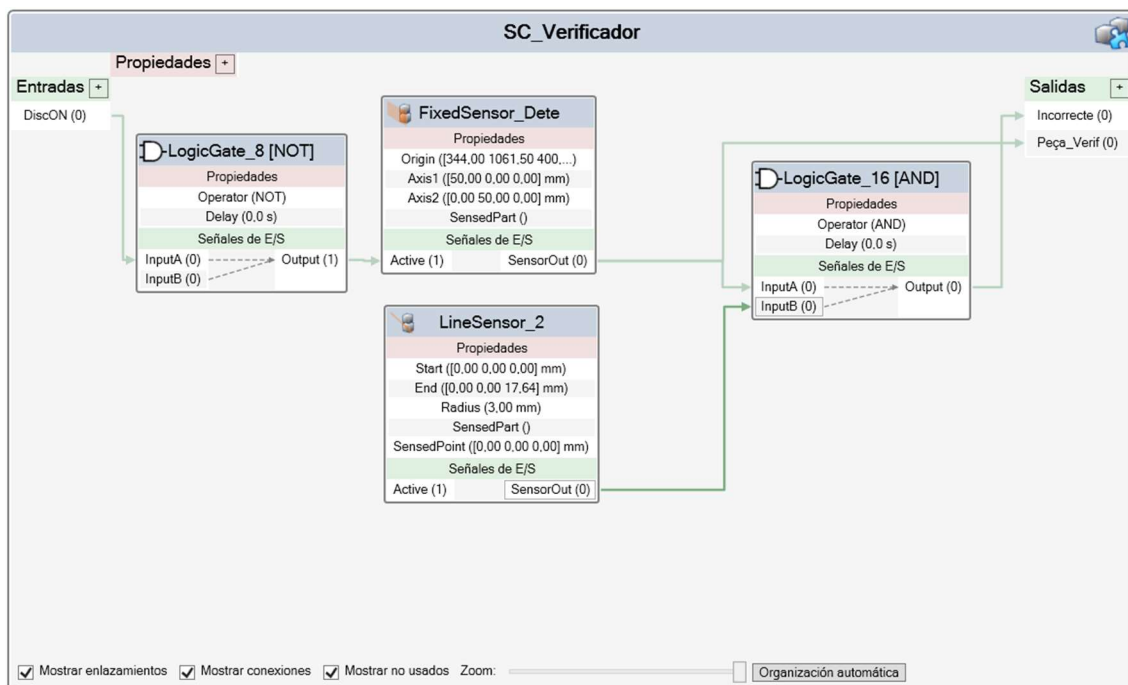


Figura 6.15: Lògica interna del SC_Verificador. [Font: pròpia]

El verificador consta d'un cilindre pneumàtic sense èmbol i amb sensor a la tija, el qual al donar-li el senyal d'activació, baixa fins al seu final de carrera o fins col·lisionar amb la peça, obtenint un senyal de tornada a l'autòmat (0 si la peça és correcta o 1 si és incorrecta).

El disseny del component a la simulació és bàsic. Consta de dos sensors i la corresponent lògica interna per obtenir les sortides desitjades. Té una entrada 'DiscON' i dues sortides, 'incorrecte' i 'Peça_Verif', on la primera serveix per avisar d'una peça incorrecta (sortida

del LineSensor ubicat a la punta del verificador) i la segona per conèixer quan executar el cicle del verificador (sortida del PlaneSensor fixat sota el propi verificador).

Tal com s'ha dut a terme amb el trepant, el verificador s'ha transformat en una eina a la qual se li ha assignat unes posicions (pose) concretes i un temps de transició entre ambdues (el qual s'ha igualat amb el temps entre posicions del trepant per minimitzar el temps d'operació).

6.2.8. Eines

Els Smart Components poden incloure eines que treballen i es mouen en sintonia amb els robots. Aquest és el cas de les pinces utilitzades en els robots antropomòrfics. El disseny de les pinces intenta assimilar-se a les disponibles al laboratori. La lògica de funcionament en els dos casos és la mateixa.

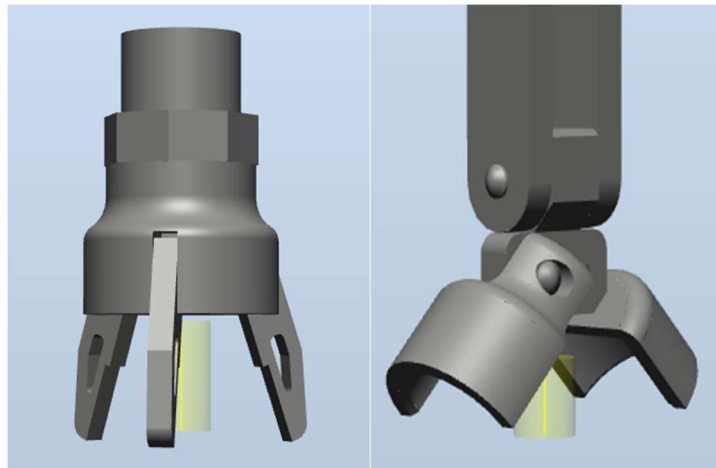


Figura 6.16: PinçaABB (esquerra) i PinçaFANUC (dreta). [Font: pròpia]

- Eina IRB 120 (SC_Tool_ABB)
- Eina LR Mate 200 iC/4S (SC_Tool_FANUC)

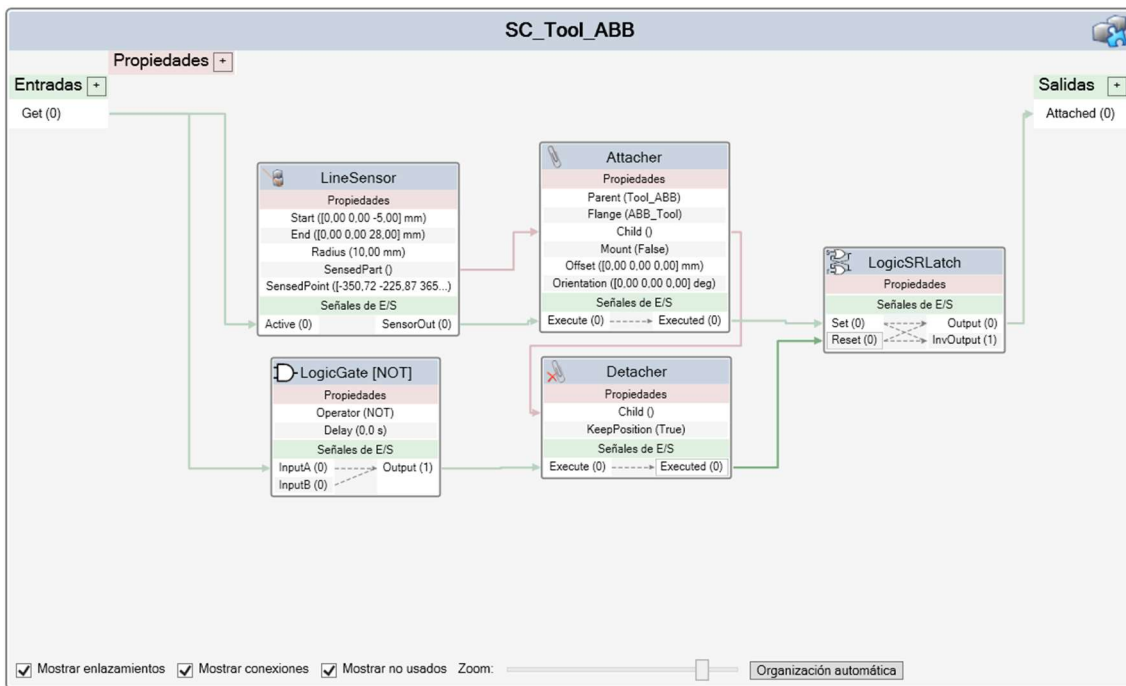


Figura 6.17: Lògica interna del SC_Tool_ABB. [Font: pròpia]

Abans de dissenyar l'Smart Component de les eines, s'ha de convertir la seva geometria a eina dins del Robotstudio. Un dels motius és per poder-la acoblar als respectius robots, i altre per poder donar-li l'animació d'apertura i tancament de la pinça (mitjançant 'poses' com s'ha dut a terme anteriorment amb el trepant i el verificador).

El component té una entrada 'Get', que s'activa quan es vol agafar una peça, i una sortida 'Attached' per assegurar que la pinça i la peça s'han connectat satisfactòriament i assegurar el correcte moviment entre dues posicions d'una trajectòria.

Per connectar la pinça amb l'eina és necessari incloure un LineSensor, que al activar-se l'entrada 'Get', s'executa el Attacher (i es tanca la pinça), mentre que si l'entrada és 0, s'executa el Detacher (i s'obre la pinça).

El subcomponent 'LogicSRLatch' serveix per donar valors a una sortida dependent de les variables que es connecten a les entrades Set (Attach) i Reset (Detach).

6.3. Lògica d'estació i connexions

La lògica d'estació permet connectar les entrades i sortides dels controladors de forma eficient amb la resta de components de la cèl·lula.

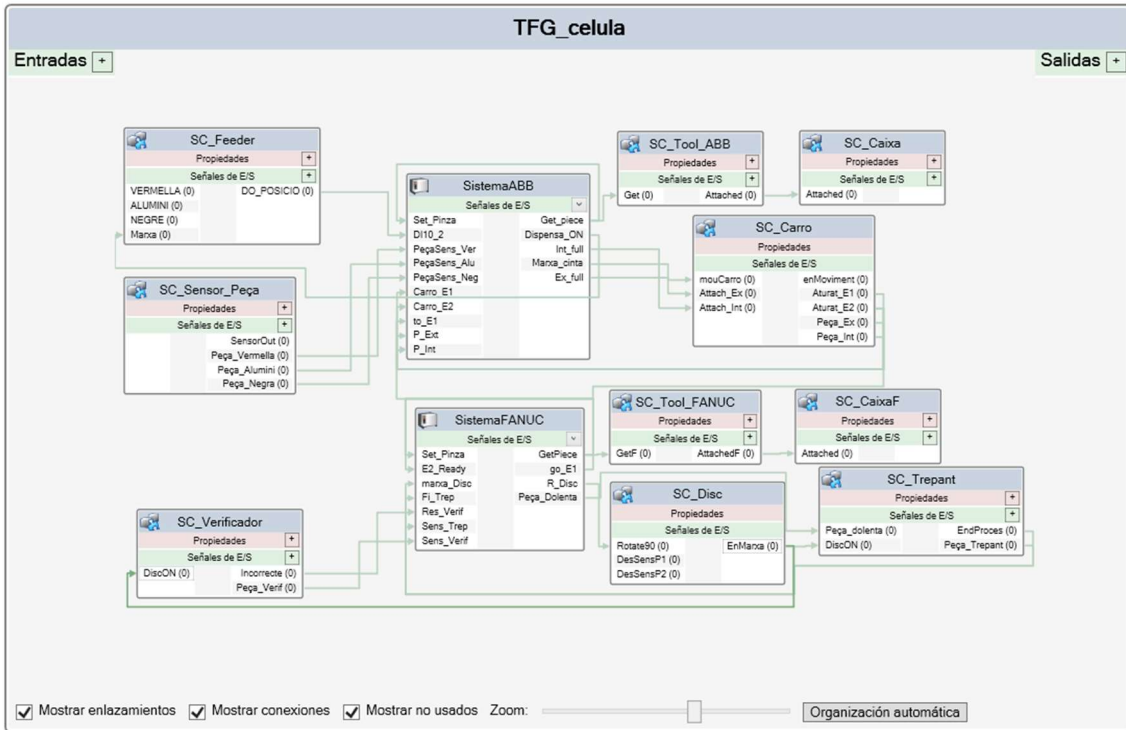


Figura 6.18: Lògica d'estació de la simulació. [Font: pròpia]

CONTROLADOR ABB (Sortida)		COMPONENT (Entrada)	
Get_pieza	→	Get	<i>SC_Tool_ABB</i>
Dispensador_ON	→	Marxa	<i>SC_Feeder</i>
Int_full	→	Attach_Int	<i>SC_Carro</i>
Ex_full	→	Attach_Ext	
Marxa_cinta	→	mouCarro	

Taula 6.2: Comunicacions de sortida del controlador ABB a components de l'estació. [Font: pròpia]

CONTROLADOR ABB (Entrada)		COMPONENT (Sortida)	
Set_Pinza	←	Get_piece	<i>CONTR. ABB</i>
to_E1	←	Go_E1	<i>CONTR. FANUC</i>
DI10_2	←	DO_POSICIO	<i>SC_Feeder</i>
PeçaSens_Ver	←	Peça_Vermella	<i>SC_Sensor_Peça</i>
PeçaSens_Alum	←	Peça_Alumini	
PeçaSens_Neg	←	Peça_Negra	
Carro_E1	←	Aturat_E1	SC_Carro
Carro_E2	←	Aturat_E2	
P_Ext	←	Peça_Ex	
P_Int	←	Peça_Int	

Taula 6.3: Comunicacions d'arribada al controlador ABB des de components de l'estació. [Font: pròpia]

CONTROLADOR FANUC (Sortida)		COMPONENT (Entrada)	
GetPiece	→	GetF	<i>SC_Tool_FANUC</i>
go_E1	→	to_E1	<i>CONTR. ABB</i>
R_Disc	→	Rotate90	<i>SC_Disc</i>
Peça_Dolenta	→	Peça_dolenta	<i>SC_Trepant</i>

Taula 6.4: Comunicacions de sortida del controlador FANUC a components de l'estació. [Font: pròpia]

CONTROLADOR FANUC (Entrada)		COMPONENT (Sortida)	
Set_Pinza	←	GetPiece	CONTR. FANUC
E2_Ready	←	Aturat_E2	SC_Carro
marxa_Disc	←	En_marxa	SC_Disc
Fi_Trep	←	End_proces	SC_Trepant
Sens_Trep	←	Peça_Trepant	
Res_Verif	←	Incorrecte	SC_Verificador
Sens_Verif	←	Peça_Verif	

Taula 6.5: Comunicacions d'arribada al controlador FANUC des de components de l'estació. [Font: pròpia]

COMPONENT	COMPONENT (Sortida)		COMPONENT (Entrada)	COMPONENT
SC_Tool_ABB	Attached	→	Attached	SC_Caixa
SC_Tool_FANUC	AttachedF	→	Attached	SC_CaixaF
SC_Disc	EnMarxa	→	DiscON	SC_Trepant/Verificador

Taula 6.6: Comunicacions entre components de l'estació. [Font: pròpia]

7. Anàlisi de viabilitat

7.1. Viabilitat tècnica

Al treballar a un laboratori d'un centre educacional, al llarg del curs es duen a terme pràctiques i demostracions amb la cèl·lula automatitzada, com pot ser el cas de l'assignatura de Robòtica de quart curs del *Grau en Electrònica i Automatització Industrial* o les exposicions a visitants a la universitat.

Físicament, l'estat de conservació de la cèl·lula és bo, es poden executar programes correctament; els robots antropomòrfics funcionen adequadament (i consegüentment, els controladors també), juntament amb les seves consoles (Flex Pendant i iPendant).

El circuit de transport del palet, formada per les quatre cintes transportadores amb els seus motors, també funcionen amb normalitat. El palet es desplaça entre les dues estacions sense aturar-se en cap punt no desitjat (els dos cilindres pneumàtics permeten la parada a les dues estacions sense problema).

L'autòmat està connectat a ambdós robots de la cèl·lula a través de les seves entrades i sortides digitals (del controlador de l'autòmat CompactLogix L32E). A aquest, mitjançant una connexió Ethernet es comunica amb el PC per utilitzar el ladder dissenyat al RSLogix 5000. La comunicació entre ambdós dispositius s'ha corroborat i la informació arriba a l'autòmat.

Pel que respecta al software que s'utilitzarà per implementar l'aplicació planificada al projecte, s'ha verificat el seu correcte funcionament per evitar conflictes per diferents versions de programa que puguin no ser compatibles amb el hardware disponible. A continuació s'enllista el software a utilitzar:

- RSLogix 5000 20.03
- FactoryTalk View 8.20.00
- RobotStudio 5.14.03

7.1.1. Pla de contingència

[NOTA: Degut a la situació excepcional de crisi sanitària, els medis per realitzar el projecte han disminuït, afectant el treball a realitzar. En aquest cas, s'ha executat el pla de contingència explicat en el capítol 7.2.4]

En cas de no poder fer ús de la cèl·lula de fabricació per falta de disponibilitat horària o averia, s'ha realitzat un pla d'actuació a seguir.

Com que una part del projecte és basa en l'experimentació i programació a la cèl·lula de forma presencial, s'ha d'eliminar tota activitat relacionada. Això comporta eliminar la programació dels braços robot i de l'autòmat, el que validaria el correcte funcionament del projecte.

A fi de solucionar aquest problema, es durà a terme tota la fase de disseny i programació de forma no presencial amb els diagrames Grafcet del sistema; el del controlador i els dels dos braços antropomòrfics.

Amb aquesta alternativa de contingència, el projecte es veu alterat el menys possible i permet la utilització de tot el material programat via GRAFCET al hardware que compon la cèl·lula de fabricació de l'ESUPT si finalment hi ha la possibilitat de donar-li ús. Es descriurà el procediment necessari per poder convertir aquests diagrames a un codi comprensible per el controlador.

La part simulada del projecte es manté inalterada, ja que no necessita de la participació in-situ a la cèl·lula.

Per compensar la falta d'experimentació presencial, es reforça la part didàctica sobre software per millorar la comprensió del treball realitzat i a fi de aconseguir fer-lo el més pedagògic possible. S'explicaran els processos més complexos que es duen a terme durant el disseny de la simulació i la creació dels Grafquets, recolzant-se en il·lustracions segons convingui.

7.2. Viabilitat econòmica

La viabilitat econòmica del projecte s'ha realitzat tenint present l'abast del mateix, on s'especificava la no inclusió del disseny físic de la cèl·lula ni el seu pressupost total.

L'estudi econòmic inclourà el pressupost de la realització del projecte (recerca d'informació, disseny de la simulació, grafquets i ladder de control, programació a màquina, verificació del funcionament...) i la redacció dels passos portats a terme fins la seva resolució (així com els costos relacionats amb els mateixos durant el desenvolupament del projecte).

7.2.1. Pressupost total del projecte

El pressupost del projecte detallat es pot trobar a l'estudi econòmic adjunt a aquest projecte, incloent els costos de realització de l'avantprojecte i els d'enginyeria de detall.

A la següent taula es pot veure un extracte resumit de les conclusions obtingudes:

PRESSUPOST		
Capítol	Descripció	Cost (€)
I	Elaboració de l'avantprojecte	15.120,00€
	Elaboració de l'enginyeria de detall	15.900,00€
	<i>COST TOTAL CAPÍTOL I (sense marge)</i>	31.020,00€
	<i>COST TOTAL CAPÍTOL I (+25% marge)</i>	38.775,00€
II	Materials	307,65€
III	Amortitzacions	1.079,98€
TOTAL FINAL		40.162,63 €

Taula 7.1: Pressupost total del projecte [Font: pròpia]

7.3. Viabilitat mediambiental

La finalitat del projecte és l'estudi del disseny i programació completa d'una cèl·lula de treball robotitzada.

Per tant, al fonamentar-se en l'estudi i disseny, es pot afirmar que l'impacte mediambiental serà molt reduït. Els possibles impactes negatius venen donats per l'ús de la pròpia màquina, així que, un cop finalitzat els dissenys per l'autòmat i la simulació, posar en marxa el disseny a la cèl·lula de fabricació si pot comportar aspectes nocius (encara que un disseny exhaustiu i minuciós pot reduir el temps emprat en fer-lo funcionar).

Tot i així, a la primera fase del projecte, l'ús intensiu de l'ordinador té un evident consum energètic, que a la seva vegada, comporta entre 52 y 234 grams equivalents de CO2 considerant una potencia de entre 80 y 360 watts, segons la Comissió Europea (sense contemplar la 'embodied energy', o energia oculta, que aglutina l'energia necessària per

a la fabricació i subministrament des de l'extracció de la matèria prima fins a la seva eliminació parcial o total i acaba generant centenars de kg en emissions de CO₂). [5]

Reduir aquesta petjada de carboni és possible gràcies a recomanacions que tothom pot posar en pràctica:

- Apagar el monitor si no s'utilitzarà durant 20 o més minuts.
- Apagar monitor i disc dur si l'aturada és de més de 2 hores.
- Connectar el monitor i la resta de dispositius a una regleta d'endolls i apaga el botó quan no es vagin a utilitzar.
- Evitar deixar l'ordinador en stand-by (el consum continua sent elevat).

Respecte el funcionament de la cèl·lula, a la fase de funcionament es poden patir residus electrònics (envelliment de l'equip) i residus de les pròpies peces, tant si la peça no és correcta i es descarta, com si el trepat de la mateixa no és validat.

Per fer funcionar els servos dels robots i els motors de la cinta transportadora, hem d'alimentar-los amb una tensió alterna, generant una contaminació electromagnètica, un factor de risc invisible que pot afectar a les persones. També està present als ordinadors.

Altres factors negatius pel medi ambient i que afecten a totes les màquines, són els sorolls durant la producció, l'emissió de calor generada pels sistemes de control i l'autòmat... així com en un cas més improbable, causar un incendi per sobreescalfament per ús continuat i intensiu. Per evitar aquest tipus d'accident mediambiental, és indispensable utilitzar totes les eines electròniques sota les condicions que indica el fabricant al seu manual de funcionament.

8. Planificació del projecte

Establerta l'aplicació a dur a terme i les especificacions del projecte, s'ha planificat una sèrie d'activitats a seguir per acotar en tasques menors el projecte i simplificar-lo.

Aquesta planificació comença el dia 18 de Novembre, data límit per l'adjudicació dels TFG per part de les coordinacions de TFG de cada àrea, i finalitza el 14 de Febrer i 10 de Juny (avantprojecte i memòria final, respectivament).

Únicament s'utilitzarà un recurs al llarg de tot el projecte.

8.1. Avantprojecte

La distribució horària per l'avantprojecte serà de 3 hores al matí diàries entre setmana, amb les excepcions següents:

- Setmana del 09/12/19-13/12/19 (exàmens de l'ESUPT)
- Festivitat de Nadal
- Cap d'Any
- Dia 1 de Gener

8.1.1. Llistat d'activitats

Seguidament, es numeren totes les activitats que conformaran l'avantprojecte, incloent les durades, les seves precedències i els recursos assignats a cadascuna.

Codi	Activitat	Precedències	Durada	Recursos
A	Hores dedicades a la recerca d'informació i generació de bibliografia d'interès	-	21h	Daniel de Castro
B	Estudi previ del disseny del sistema	-	15h	Daniel de Castro
C	Definició de l'aplicació (solució del projecte)	A, B	6h	Daniel de Castro
D	Definició de la metodologia a seguir	C	12h	Daniel de Castro
E	Comprovació de les viabilitats	D	8h	Daniel de Castro

Codi	Activitat	Precedències	Durada	Recursos
F	Definició de pressupostos	E	18h	Daniel de Castro
G	Redactat de l'informe	C	52h	Daniel de Castro
H	Generació d'annexos	G	4h	Daniel de Castro
I	Revisió de la documentació aportada	H	5h	Daniel de Castro

Taula 8.1: Activitats relacionades amb l'avantprojecte [Font: pròpia]

La duració de l'avantprojecte s'estipula en 135h i un total de 47 dies treballats.

8.1.2. Descripció de les activitats

Codi	Descripció
A	En aquesta activitat es buscarà informació sobre la importància de l'automatització en l'actualitat, estadístiques i el seu progrés de millora continua en la nova indústria 4.0.
B	S'estudiarà el disseny de la cèl·lula de fabricació disposada a laboratori del TCM; components que la formen, ús, normatives...
C	Es definirà la tasca a realitzar amb la cèl·lula automatitzada per simular un cas d'automatització real de l'indústria.
D	Es farà una pauta dels passos a seguir per portar a terme el projecte de detall.
E	Es realitzarà un estudi de tres viabilitats fonamentals; tècnica, econòmica i mediambiental.
F	Es faran números del cost total tant de l'avantprojecte com del projecte de detall.
G	En aquesta activitat es redactarà progressivament el que serà el document a entregar per l'avantprojecte.
H	S'inclourà informació addicional que pugui ser d'interès pel projecte, així com datasheets dels robots.
I	Es farà una revisió exhaustiva del format de tota la documentació.

Taula 8.2: Descripció d'activitats relacionades amb l'avantprojecte [Font: pròpia]

8.1.3. Diagrama de Gantt de l'avantprojecte

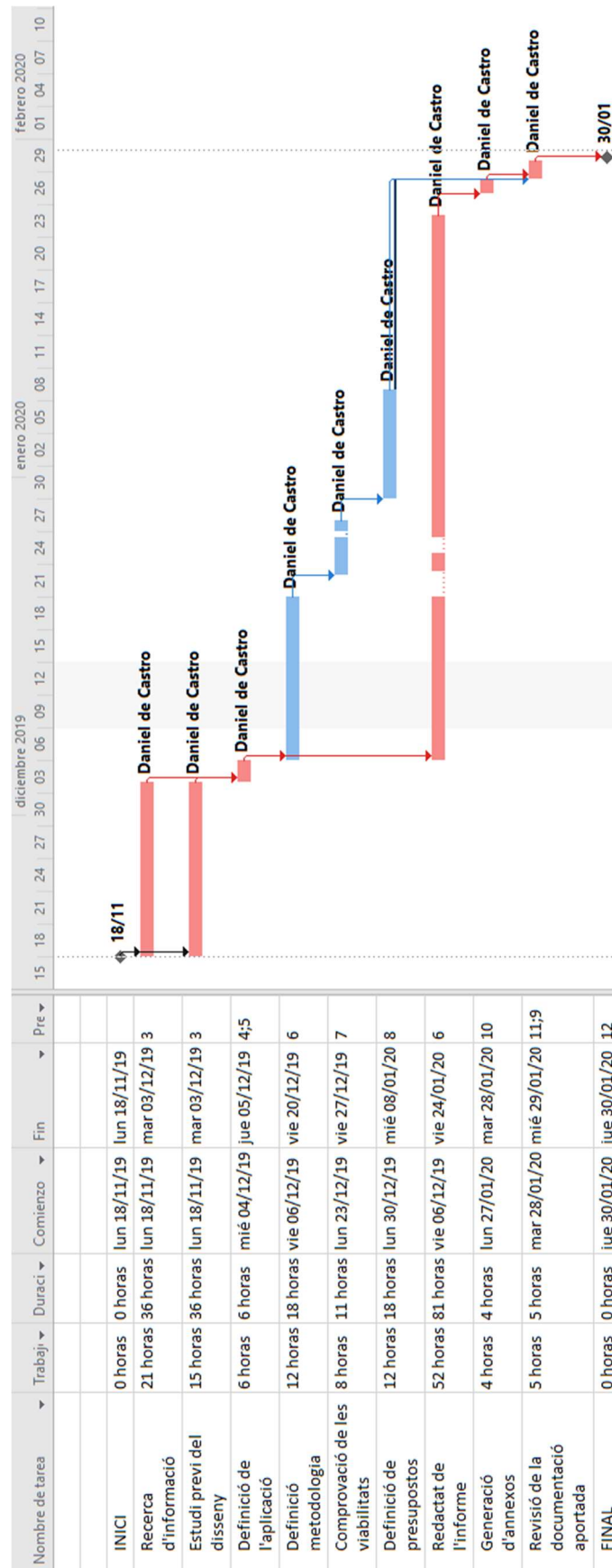


Figura 8.1: Diagrama de Gantt de l'avantprojecte [Font: pròpia]

8.2. Enginyeria de detall

[NOTA: Degut a la situació excepcional de crisi sanitària, els medis per realitzar el projecte han disminuït, afectant el treball a realitzar. En aquest cas, s'ha executat el pla de contingència explicat en el següent capítol 6.2.4]

La distribució horària per l'enginyeria de detall serà de 5 hores al matí diàries entre setmana per aconseguir (si es compleixen les dates de la planificació), arribar a l'entrega intermèdia del projecte amb les primeres proves de funcionament i un mes de marge respecte l'entrega final per depurar el projecte i evitar contratemps.

Al ser una part majorment tècnica, és molt possible que es requereixi estar presencialment al laboratori amb la cèl·lula per poder avançar i comprovar la feina feta, així com utilitzar softwares que no tenen versió gratuïta per treballar externament, donant pas a les excepcions següents:

- Setmana del 16/03/20-20/03/20 (exàmens de L'ESUPT)
- Festius a l'ESUPT (del 6 al 13 d'Abril, 23 d'Abril, 1 de Maig i 1 de Juny)

8.2.1. Llistat d'activitats

Seguidament, es numeren totes les activitats que conformaran l'enginyeria de detall, incloent les durades, les seves precedències i els recursos assignats a cadascuna.

Codi	Activitat	Precedències	Durada	Recursos
A	Revisió de l'avantprojecte i verificació de la plantejament a implantar	-	20h	Daniel de Castro
B	Repàs i recerca d'informació concreta sobre simulacions amb RobotStudio	A	25h	Daniel de Castro
C	Estudi d'algorismes de funcionament dels robots per l'aplicació a la cèl·lula	A	15h	Daniel de Castro
D	Programació de la simulació	B, C	25h	Daniel de Castro

Codi	Activitat	Precedències	Durada	Recursos
E	Repàs i recerca d'informació concreta sobre RsLogix 5000	D	10h	Daniel de Castro
F	Aproximació dels graficets de funcionament de l'autòmat	D	15h	Daniel de Castro
G	Disseny dels ladders de l'autòmat	E, F	25h	Daniel de Castro
H	Correcció del posicionament del robot ABB respecte la simulació	G	5h	Daniel de Castro
I	Programació a la consola del robot FANUC	H	5h	Daniel de Castro
J	Proves de funcionament i validació	H, I	15h	Daniel de Castro
K	Revisió final del sistema i les millores dutes a terme	J	15h	Daniel de Castro
L	Espectura de la memòria	A	80h	Daniel de Castro
M	Revisió final del projecte	L	10h	Daniel de Castro

Taula 8.3: Activitats relacionades amb l'enginyeria de detall [Font: pròpia]

La duració de l'enginyeria de detall s'estipula en 265h i un total de 53 dies treballats.

8.2.2. Descripció de les activitats

Codi	Activitat
A	L'avantprojecte es revisarà, juntament amb els procediments duts a terme per generar el plantejament.

B	Repàs de conceptes coneguts de RobotStudio i desenvolupar la capacitat de fer simulacions més realistes i complexes.
C	S'ha de prendre constància dels moviments que requereix fer cada robot pel correcte funcionament del procés per simplificar les programacions posteriors i no ser redundant.
D	Programació de la simulació de la cèl·lula completa.
E	Repàs de conceptes coneguts de RsLogix 5000 i recerca d'informació concreta com creació de subrutines o 'tips' diversos que puguin simplificar un programa llarg.
F	Es buscarà aproximar-se als graficats de funcionament de l'autòmat necessaris per aconseguir un ladder complet.
G	Disseny dels ladders del programa de l'autòmat tenint present cada entrada i sortida a cada robot.
H	Es verificarà o modificarà el posicionament als punts de trajectòria del robot ABB respecte la simulació (poden haver-hi variacions respecte altres proves al no estar la cèl·lula cimentada al sòl de l'edifici).
I	Es requerirà programar els moviments realitzats a la simulació a la consola del robot FANUC (el seu llenguatge es incompatible amb RAPID).
J	Proves de funcionament sobre la cèl·lula física al laboratori del TCM.
K	Conclusions del projecte, fent una recapitulació i revisió final del sistema amb les millores dutes a terme.
L	Es redactarà un document tècnic on s'exposi els passos duts a terme per dissenyar i programar una cèl·lula de fabricació flexible.
M	Es revisarà el format al llarg del document.

Taula 8.4: Descripció d'activitats relacionades amb l'enginyeria de detall [Font: pròpia]

8.2.3. Diagrama de Gantt de l'enginyeria de detall

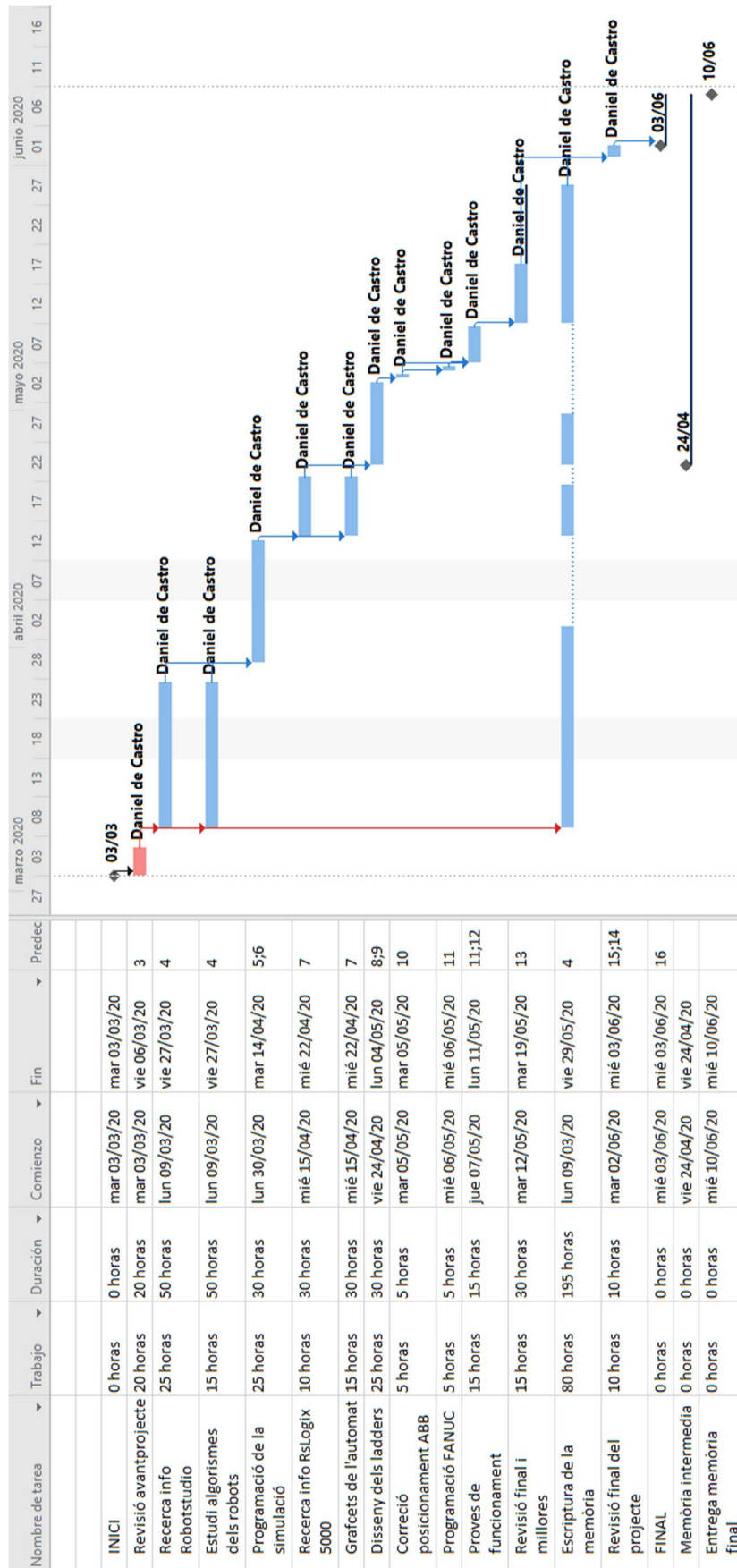


Figura 8.2: Diagrama de Gantt de l'enginyeria de detall [Font: pròpia]

8.2.4. Pla de contingència

Els canvis a efectuar en el pla de contingència afecten a algunes de les activitats del projecte de detall, eliminant-se les activitats E, G, H, I i J.

Altres, com l'activitat L original (escriptura de la memòria) es veuran afectades en la seva durada degut a la inclusió d'explicacions addicionals respecte la creació dels GRAFCETS del sistema i el disseny i programació de la simulació.

Seguidament, s'il·lustra la nova planificació del projecte:

Codi	Activitat	Precedències	Durada	Recursos
A	Revisió de l'avantprojecte	-	20h	Daniel de Castro
B	Recerca d'informació respecte Robotstudio	A	25h	Daniel de Castro
C	Recerca d'informació respecte GRAFCET-Studio	A	15h	Daniel de Castro
D	Estudi dels algorismes dels robots	B, C	25h	Daniel de Castro
E	Creació d'Smart Components	D	10h	Daniel de Castro
F	Disseny de l'estació de simulació	D	15h	Daniel de Castro
G	GRAFCET de l'autòmat	E, F	25h	Daniel de Castro
H	GRAFCET del robot ABB	G	5h	Daniel de Castro
I	GRAFCET del robot FANUC	H	5h	Daniel de Castro
J	Programació codi RAPID	D, I	10h	Daniel de Castro

K	Proves i millores	G, H, I	10h	Daniel de Castro
L	Esriptura de la memòria	A	90h	Daniel de Castro
M	Revisió final del projecte	J, K, L	10h	Daniel de Castro

Taula 8.5: Nova planificació degut al pla de contingència [Font: pròpia]

A destacar els canvis de precedències en les noves activitats J, K i M:

- J precedida per:
 - D (Estudi dels algorismes dels robots)
 - I (GRAFCET del robot FANUC)
- K precedida per:
 - G (GRAFCET de l'autòmat)
 - H (GRAFCET del robot ABB)
 - I (GRAFCET del robot FANUC)
- M precedida per:
 - G (GRAFCET de l'autòmat)
 - H (GRAFCET del robot ABB)
 - I (GRAFCET del robot FANUC)

La duració de l'enginyeria de detall s'estipula en 265h i un total de 53 dies treballats.

En aquest pla, les hores necessàries són les mateixes que les planificades originalment i, conseqüentment, no es modifica el capítol 1 del pressupost final del projecte.

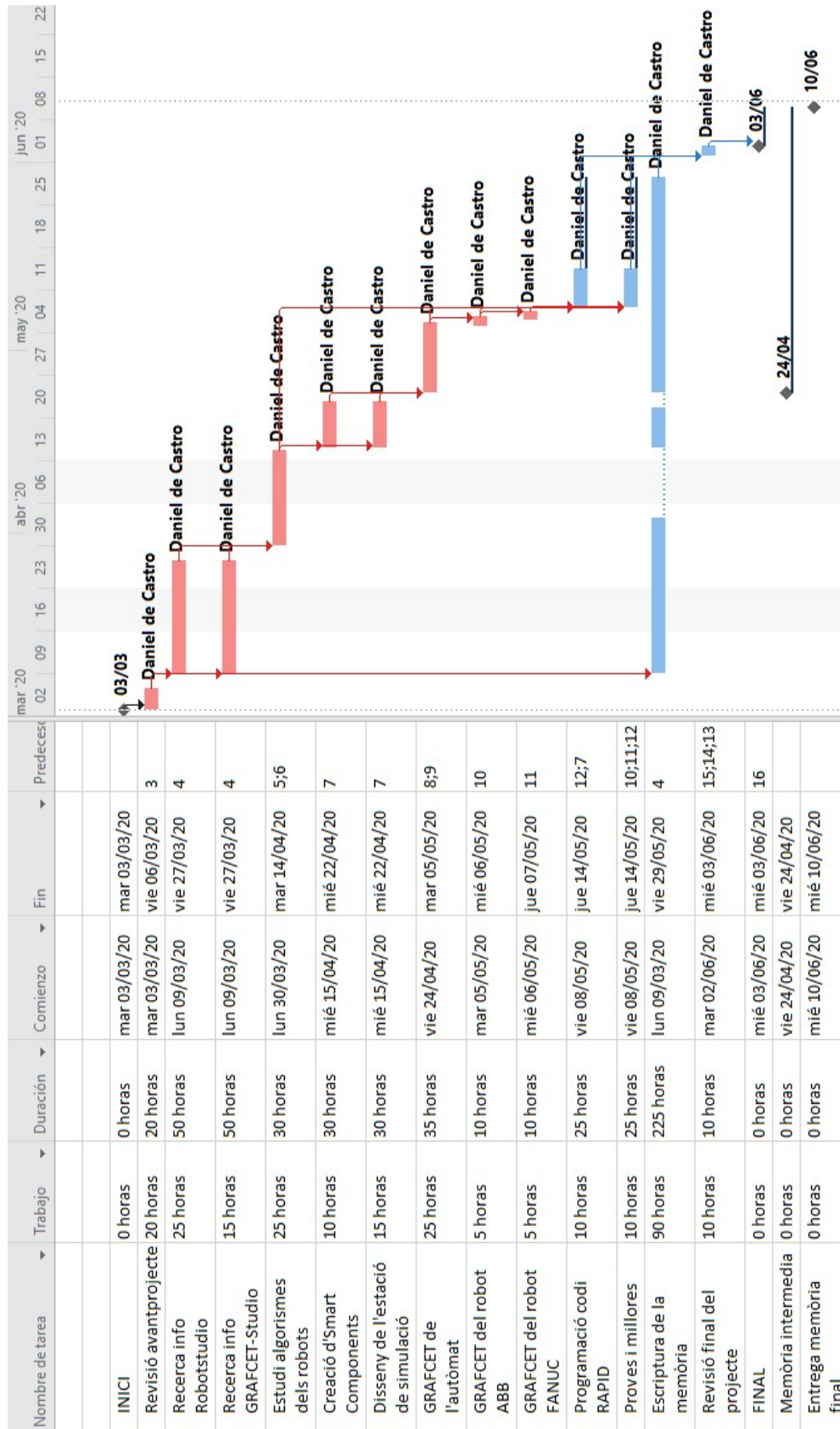


Figura 8.3: Diagrama de Gantt del pla de contingència [Font: pròpia]

9. Conclusions

L'experiència durant la realització del projecte ha estat enriquidora i he après molt sobre la simulació amb autòmats i la lògica amb la que funcionen. Potser aquest projecte ha estat un primer tast d'una possible sortida professional que fins ara no contemplava massa. La quantitat de robots que apareixen al mercat any rere any és aclaparadora, és un sector molt dinàmic i que permet un aprenentatge continu.

M'hagués agradat continuar fent més feina del treball; el pla de contingència dissenyat que he utilitzat per la part pràctica del projecte m'ha semblat força interessant i penso que pot servir per la finalitat inicial; que l'explicació sigui comprensible per així facilitar el procés d'aprenentatge dels braços robot *ABB IRB 120* i *FANUC LR Mate 200ID 4S* i del controlador de la cèl·lula.

S'ha proposat també un disseny multi carro per millorar l'actual funcionament amb un carro. Amb els *GRAFCETS*, hi ha una bona base per desenvolupar els ladders de l'autòmat i, juntament amb la programació *RAPID* pel robot *IRB 120* (pel *FANUC LR Mate* s'hauria de programar manualment), fer funcionar l'aplicació a la cèl·lula.

En aquest cas, es podria veure quins temps de cicle fa la cèl·lula amb diferents versions d'una mateixa aplicació, ver estudiar com millorar més el seu rendiment.

Lamento el fet de no haver pogut disposar de la cèl·lula robòtica per dur a terme el treball; l'experimentació in-situ segur que hagués estat molt engrescadora. Tot i així, encara les dificultats que comporta fer un disseny en un software com *Robotstudio*, sent la part del projecte a la que més temps li he dedicat, els resultats crec que han sigut satisfactoris.

La realització d'aquest projecte ha sigut una experiència intensa que segur que recordaré durant molt de temps per els condicionants que l'han envoltat, tant personals com globals. Però sobretot per recordar que l'esforç té recompensa. Han sigut uns anys llargs de vida universitària, bastant esgotadors, però tot té un final.

Espero amb ganes el nou inici que està per arribar.

10. Bibliografia i referències

- [1] LÓPEZ, ESTELA/STAVRAKY, CARLA. El Economista (2020). *ABB invierte 15 millones en un centro de I+D de robótica y automatización para España*. Disponible en <https://www.eleconomista.es/catalunya/noticias/10327573/01/20/ABB-invierte-15-millones-en-un-centro-de-ID-de-robotica-y-automatizacion-industrial-para-Espana.html>
- [2] POZO, DAVID. Siemens, S.A. (2017). *Future of Manufacturing - Visión de Siemens sobre Industria 4.0*. Disponible en <https://www.youtube.com/watch?v=GF1MYUdFcqo>
- [3] BOE. *Real Decreto 1644/2008, de 10 de octubre, por el que se establecen las normas para la comercialización y puesta en servicio de las máquinas*. Disponible en <https://www.boe.es/buscar/doc.php?id=BOE-A-2008-16387>
- [4] BOE. *Real Decreto 1215/1997, de 18 de julio, por el que se establecen las disposiciones mínimas de seguridad y salud para la utilización por los trabajadores de los equipos de trabajo*. Disponible en <https://www.boe.es/buscar/doc.php?id=BOE-A-1997-17824>
- [5] PlanetaRecicla. Ecoembes (2016). *Los ordenadores también emiten CO2*. Disponible en <https://www.ecoembes.com/es/planeta-recicla/blog/los-ordenadores-tambien-emiten-co2>
- [6] CASTILLO GUTIÉRREZ, CRISTIAN. (2011). *Disseny d'una cèl·lula de treball robotitzada que utilitzarà un robot industrial ASEA IRB 120*. Tesina disponible al repositori: <https://upcommons.upc.edu/handle/2099.1/12869>
- [7] MONAGO FERNÁNDEZ, JOSÉ. (Primavera 2012). *TFG Sistema de supervisió i control d'una cèl·lula robotitzada*. Tesina disponible al repositori: <https://upcommons.upc.edu/handle/2099.1/16023>
- [8] GUERRERO SAIZ, JOSÉ MANUEL (2019). *Programación estructurada de autómatas programables con Grafset*. Disponible per ISBN: 9788428341387

