Universitat Autònoma de Barcelona

Dipòsit digital de documents de la UAB

This is the **published version** of the bachelor thesis:

Lakhmanets, Maksym; Grau Sala, Ramon, dir. Entorn de laboratori: Creació d'un entorn simulat utilitzant el concepte de Digital Twin. 2023. (958 Enginyeria Informàtica)

This version is available at https://ddd.uab.cat/record/272797

under the terms of the BY-NC-ND license

Entorn de laboratori: Creació d'un entorn simulat utilitzant el concepte de Digital Twin

Maksym Lakhmanets

Resum– El projecte es centra en la creació d'un entorn digital simulant un PLC (Controlador lògic programable) per realitzar proves enfocades en els aplicatius utilitzats pels operaris i altres treballadors en les línies de producció. La idea principal és replicar els PLC físics propis de les línies de producció, els quals no estan disponibles per part de l'equip de desenvolupadors. El benefici principal és tenir una eina capaç de replicar tot tipus de senyals per obtenir simulacions orientades al desenvolupament de codi de les pròpies eines que són elaborades per aquest equip, amb la finalitat de garantir un software robust i de qualitat pel benefici de l'empresa. Això s'anomena "Digital Twin"[1], un model virtual desenvolupat per reflexar amb precisió un objecte o màquina del món físic, amb l'objectiu de replicar els diferents sensors o elements que generen dades de la màquina per ser tractades en un entorn virtual i així poder executar simulacions i estudiar problemes que puguin sorgir, generant informació valuosa que després es pot aplicar al dispositiu físic original.

Paraules clau– PLC, Digital Twin, Inmation, dashboard, línia de producció, test

Abstract– The project focuses on creating a digital environment simulating a PLC (Programmable Logic Controller) to conduct tests focused on the applications used by operators and other workers on production lines. The main idea is to replicate the physical PLCs of the production lines, which are not available to the development team. The main benefit is having a tool capable of replicating all types of signals to obtain simulations oriented towards the development of code for the tools themselves that are developed by this team, in order to ensure robust and high quality software for the benefit of the company. This is known as a "Digital Twin"[1], a virtual model developed to accurately reflect an object or machine in the physical world, with the goal of replicating the different sensors or elements that generate machine data to be processed in a virtual environment and thus being able to run simulations and study problems that may arise, generating valuable information that can later be applied to the physical device.

Keywords– PLC, Digital Twin, Inmation, dashboard, production line, test

1 Introducció - Context del treball

Boehringher Ingelheim[2], empresa fundada l'any 1885 es dedica a investigar, desenvolupar, fabricar y comercialitzar productes terapèutics per a la medicina humana i animal. L'empresa està entre les 20 principals del món i es dedica la investigació i el desenvolupament (I+D).

El principal equip del departament d'IT conformat per enginyers i desenvolupadors persegueix l'objectiu comú de la transició cap a la Indústria 4.0, el qual impulsa l'automatització de processos a la línia de producció. En aquest linia de producció existeixen diferents màquines que són controlades per diferents nodes distribuïts. Dins d'aquest projecte (Factory Automation), hi participem 3 equips: l'equip del *front-end*, el *back-end*, i *integrations*.

L'objectiu d'aquests 3 equips es resumeix en portar a terme casos d'ús en benefici de l'empresa, normalment elaborant eines en forma de pàgines web o Dashboard's, utilitzats majoritàriament per altres treballadors com operaris de la línia o encarregats de la monitoratge dels diferents processos de producció.

• E-mail de contacte: max.lakh1@gmail.com
• Mención realitzada: Enginyeria de Computadors
• Treball tutoritzat per: Ramon Grau (Departament d'Arquitectura de Computadors i Sistemes Operatius
• Curs 2022/23

Aquestes eines aporten un gran valor informatiu i són l'esquelet que permet controlar els diferents processos de producció, per tant, la importància de generar i transmetre informació real sense errors és vital per poder produir i controlar els productes elaborats per les fàbriques.

Un cas d'ús molt útil per poder ajudar a aconseguir aquest objectiu comprèn una eina per poder testejar aquests dashboards per tal de comprovar aquesta integritat de dades i, per tant, agilitzar el desenvolupament del programari, atorgant un nivell més baix de generació de dades, mitjançant un bessó digital (*Digital Twin*) que simula un PLC similar als de les línies de producció

1.1 Objectius

El mètode actual del desenvolupament de programari de l'equip consisteix en treballar sobre les llibreries existents de codi, on cada desenvolupador es crea una còpia del codi a la seva carpeta. Normalment per assegurar-se que la implementació funciona correctament és necessari una sèrie de dades que estan disponibles als servidors de MongoDB.

Un problema habitual és que aquests conjunts de dades no sempre s'ajusten al cas d'ús assignat al treballador i, per tant, dificulta el testeig de la solució proposada en cada cas. Un exemple pot ser un indicador de rendiment, on les dades existents podrien mostrar un 70% però la solució del cas d'ús necessita comprovar un cas irreal com el 100%, per aquesta limitació l'enginyer pot tenir dificultats per comprovar que es pugui donar aquest cas, ja que s'ha de confiar de les dades proveïdes pel sistema real.

Els objectius d'aquest projecte proposen donar solució a aquesta problemàtica per tal d'aportar un nivell més ample quant a generació de dades; els objectius plantejats són els següents:

A. Possibilitat d'introduir processos de fàbrica manualment en qualsevol moment.

B. Disposar dels mateixos estats dels processos que a la línia de producció.

C. Poder transmetre informació en temps real.

D. Permetre que el sistema sigui escalable.

E. Dissenyar una interfície similar a la d'un PLC.

Un cop definits els objectius principals es precedirà a explicar les eines utilitzades per elaborar la solució proposada.

1.2 Planificació i metodologia del projecte

Per a organitzar el projecte, s'ha utilitzat la metodologia Agile de Scrum, on cada tasca s'organitza per Sprints de dues setmanes.

El projecte va ser dividit en 10 sprints. La manera en la qual s'organitzava en un primer moment consistia en un recapte d'informació per estudiar la possibilitat i viabilitat del projecte, seguidament d'un primer contacte amb la plataforma TIA Portal de SIEMENS[3] per implementar un PLC. Un cop dissenyat el PLC i la lògica de la màquina d'estats es prosseguí amb la integració amb la plataforma àmpliament utilitzada a l'empresa anomenada Inmation. Un cop finalitzada la integració l'últim pas va ser el testeig i les simulacions per part de l'equip.

2 COMPONENTS

2.1 Siemens TIA Portal

Siemens TIA Portal és un programari de programació i configuració per a controladors programables (PLC) i sistemes d'automatització de Siemens. El programari proporciona una interfície unificada per a programar i configurar diferents dispositius, com ara PLC's, pantalles HMI, variadors de freqüència, entre d'altres. També permet la integració de diferents sistemes d'automatització, com s'explicarà més endavant; en aquest cas s'integrarà amb Inmation, middleware utilitzat a l'empresa.

Dins del software TIA Portal existeixen dos components principals i són els que s'utilitzen en aquest projecte:

1. **STEP 7:** és el programari principal de programació per als PLC's de Siemens, permetent la creació de programes de control i la configuració d'aquests dispositius.

2. **WinCC:** és un programari de supervisió i control per als sistemes d'automatització. Permet la creació de pantalles HMI i la visualització de dades de processos en temps real.

2.1.1 Disseny HMI

HMI (Human Machine Interface) es refereix a la interfície o pantalla que permet als operadors interactuar amb un sistema automatitzat, com un sistema de control de processos o una màquina industrial. Un HMI pot incloure una pantalla tàctil, un teclat o una sèrie de botons que permeten a l'operador introduir informació, veure l'estat del sistema, modificar configuracions, etc. També pot incloure gràfics i visualitzacions per ajudar a l'operador a entendre millor el sistema i prendre decisions informades.

En aquest cas, el HMI dissenyat contindrà 4 elements diferenciats (A.1 Panell PLC):

1. Transferència d'informació: Informació bàsica d'un procés (batch). Comprèn una columna per entrar els diferents caps que identifiquen el procés (columna IN) i el seu respectiu valor a la sortida (columna OUT). Els valors dels camps es componen de:

   - **BatchID:** Identificació del lot de productes tal com s'expressaria a la línia de producció.
   - **Material Number:** Numero del material.
   - **Material Desc:** Descripció del material.
   - **Order Number:** Quantitat de productes.
   - **Order Size:** Quantitat que conté el lot.
   - **Recipe Name:** Nom de la recepta.

Un cop introduïdes el botó "Transfer" (A.1 Panell PLC (1)), tal com indica el nom, transfereix la informació a la sortida.

2. Quantitats dintre un lot (batch): Aquests camps indiquen quantitats dels productes dintre un batch, que durant el procés poden anar augmentant o disminuint depenent el cas d'ús que ho requereixi. Es compon de 3 elements (A.1 Panell PLC (2)):

   – **Good Quant:** Quantitat bona de cada producte.
   – **Bad Quant:** Quantitat defectuosa de cada producte.
   – **Total Quant:** Suma de quantitat bona i defectuosa.

   Seguint la lògica anterior, s'introdueixen valors de Good o Bad Quant a la columna IN i amb el botó de + o – s'incrementa o decrementa la quantitat. La quantitat total es veurà afectada a cada acció de l'usuari. El botó RST posa a 0 el respectiu indicador a la sortida.

3. Aquesta secció es composa de 2 elements (A.1 Panell PLC (3)):

   – **Error Code:** Camp d'entrada on s'introdueix un codi d'error i s'envia mitjançant el botó "Send".
   – **Increments:** Aquesta funcionalitat suma cada 1 segon el valor introduït al camp d'entrada, al valor de sortida de Good Quant.

4. Estat del batch: Diagrama interactiu que mostra l'estat del batch en funció del diagrama d'estats (A.1 Panell PLC (4)) i l'actuació per part de l'usuari mitjançant els diferents botons. Aquest element el conformen:

   – **Diagrama d'estats:** Depenent de l'estat actual, el requadre groc indica l'estat mitjançant una intermitència.
   – **Batch Status:** Requadre que indica l'estat actual en forma de text (Idle, Starting, Running...).
   – **Polsadors de control:** Depenent de l'estat actual, es mostren o es deixen de mostrar els polsadors, els quals depenent de l'actuació de l'usuari, determinaran el pròxim estat del procés.

2.1.2 Arquitectura PLC

El PLC es compon de dos elements diferenciats, la vista (WinCC) i la lògica de programació (STEP 7). El component WinCC segueix la dinàmica de "drag and drop" per posar els elements que contindrà la vista del PLC.

STEP 7 s'usa majoritàriament per a la programació de la lògica. Els llenguatges de programació que es poden utilitzar es divideixen en 3 opcions: SCL (Structured Control Language), FBD (Function Block Diagram) i STL (Statements List). En aquest projecte només s'utilitzen els 2 primers.

L'estructura de la programació es pot dividir en diversos blocs de funcionament paral·lel, encara que els utilitzats en aquest projecte són els OB, FC i Global DB (1):

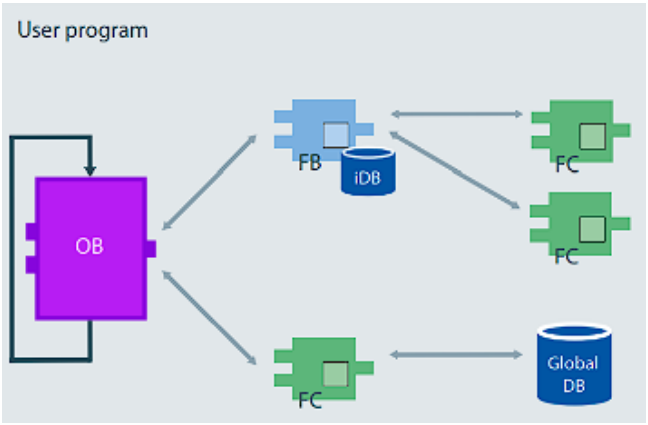


Fig. 1: Diagrama de funcionament

Cada bloc es defineix com:

- **OB (Organitzation Block):** És l'equivalent al "main" d'un programa i decideix l'ordre d'execucions dels FC. Aquest bloc és executat en bucle cada 100ms refrescant els blocs de següent nivell.
- **FC (Functions):** Es defineixen amb dades d'entrada i dades de sortida. Es on es produeix la lògica del codi que defineix el funcionament del PLC.
- **Global DB:** Tal com el seu nom indica, es tracta d'una base de dades global que conté les variables utilitzades pel PLC.

2.1.3 Lògica PLC

L'organització de blocs conté diversos elements, cadascun proporcionant la seva part de lògica a la implementació del PLC.

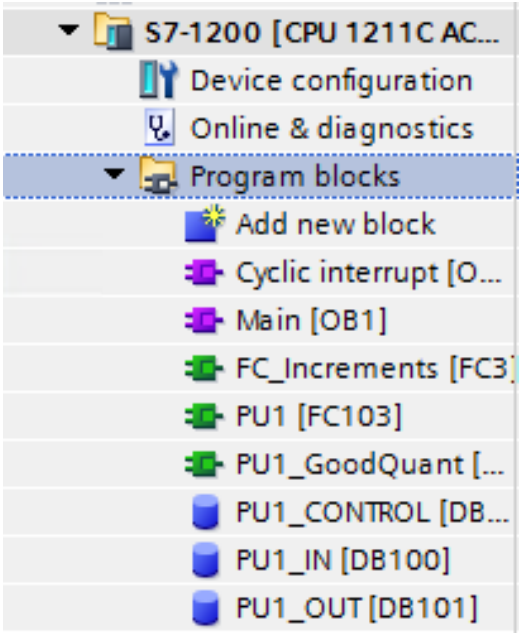


Fig. 2: Blocs del projecte

Aquests blocs es defineixen com:

- **Main:** Conté l'execució dels FC i està programat en llenguatge FBD. Indica els blocs FC i la seva execució (A.3.1 Main).

- **PU1:** FC que executa la màquina d'estats (A.2 Maquina d'estats). Es tracta d'una màquina d'estats finits estesa (EFSM) on les transicions dels estats venen donades per l'estat actual i una sèrie de condicions. El codi conté dos blocs diferenciats: el bloc d'estats i el bloc de transicions. El primer s'encarrega de definir els estats i les variables que conté aquest, com per exemple mostrar o amagar certs botons. El segon comprova les condicions per canviar-se o mantenir l'estat. Al final del codi es realitza una assignació del tipus: NextState = CurrentState. El llenguatge usat es SCL (A.3.3 PU1), ja que és el més adient per programar una màquina d'estats.

- **PU1_GoodQuant:** FC que conté la lògica única i exclusivament de l'element 2 (A.1 Panell PLC (2)). Es tracta d'una sèrie de condicions que efectuen les operacions aritmètiques dels elements del HMI. El llenguatge utilitzat és el SCL.

- **FC_Increments:** FC dedicat al 3r element (A.1 Panell PLC (3)), en aquest cas empra el llenguatge FDB (A.3.2 FC_Increments), ja que conté un element d'interrupció d'1s per cada operació aritmètica de suma.

- **PU1_CONTROL:** (A.4 Database) DB de variables utilitzades per al control de la màquina d'estats. Conté variables com els propis estats i altres variables de control.

- **PU1_IN:** DB que conté les variables d'entrada que són introduïdes per l'usuari.

- **PU1_OUT:** DB que conté les variables de sortida pro-por cionades per la lògica del PLC que seran enviades al dashboard.

2.2 Inmation

Inmation[4] és un programari d'automatització i control industrial desenvolupat per l'empresa AspenTech. És utilitzat per monitorar i controlar processos industrials en temps real, incloent-hi la recollida de dades, l'anàlisi i la presa de decisions. També es pot usar per integrar diferents sistemes i dispositius en una sola plataforma, el que facilita la gestió i el control dels processos.

Algunes de les característiques d'Inmation inclouen:

- Recollida de dades: Inmation pot recollir dades d'una varietat de fonts, com ara PLC's, sensors, SCADA, bases de dades i aplicacions web.

- Anàlisi de dades: Inmation proporciona eines per analitzar les dades recollides, com ara gràfics, taules i estadístiques.

- Presa de decisions: Inmation permet crear regles i algoritmes per automatitzar la presa de decisions d'acord amb les dades recolli-des i analitzades.

- Integració: Inmation pot integrar-se amb una varietat de sistemes i dispositius, com ara PLC's, sensors, bases de dades i aplicacions web.

- Interfície d'usuari: Inmation proporciona una interfície d'usuari intuïtiva que permet als usuaris navegar i analitzar les dades fàcilment.

- Alarmes i notificacions: Inmation permet configurar alarmes i notificacions per alertar als usuaris de qualsevol condició anòmala o problema en el procés.

- Accés remot: Inmation permet als usuaris accedir a les dades i controlar els processos des de qualsevol lloc amb accés a internet.

- Escalabilitat: Inmation és escalable, el que significa que es pot adaptar a una varietat d'aplicacions i entorns, des de petites instal·lacions fins a grans centrals industrials.

2.2.1 Inmation Datastudio

Datastudio es el principal client de system:inmation. Ofereix accés complet a dades en temps real i històriques amb una suite dinàmica i efectiva. A continuació el llistaran alguns dels principals components que s'utilitzen en aquest projecte:

- **Core:** Servei que processa i emmagatzema les dades del repositori i els generats pels mateixos elements que conté.

- **Data Store:** Emmagatzematge de dades generades per les diverses fonts del sistema. Té la possibilitat de connexió amb l'historiador de dades de Inmation MongoDB per als diversos tipus de dades si s'escau.

- **Data Source:** Element que exposa la informació recupera da per Inmation.

- **Connector:** Servei que recupera les dades, en aquest cas, generades pel PLC.

- **Repositoris:** Conjunt de repositoris sobre els quals s'executa la lògica del tractament de dades, utilitzant el llenguatge de LUA.

- **I/O Items:** Conjunt d'elements que proporcionen càlculs en temps real utilitzant LUA.

Per comunicar el PLC amb el sistema d'Inmation és necessari una prèvia integració de l'equip. Per tal d'assolir aquest objectiu es fa ús d'una eina creada per l'equip de desenvolupadors anomenada Shopfloor Manager.

Aquesta eina es un "addin" o extensió d'Excel la qual mitjançant una sèrie d'arxius d'Excel anomenats "spreadsheerts" la qual acompleix la tasca de crear els elements llistats anteriorment a Inmation Datastudio. Depenent del que sigui necessari crear, s'utilitzarà un Spreadsheet o altre. Per començar es crearà un Datastore (LAB_ENV) (3) per a poder guardar les dades generades pel PLC, aquest es crearà dintre del Core System:

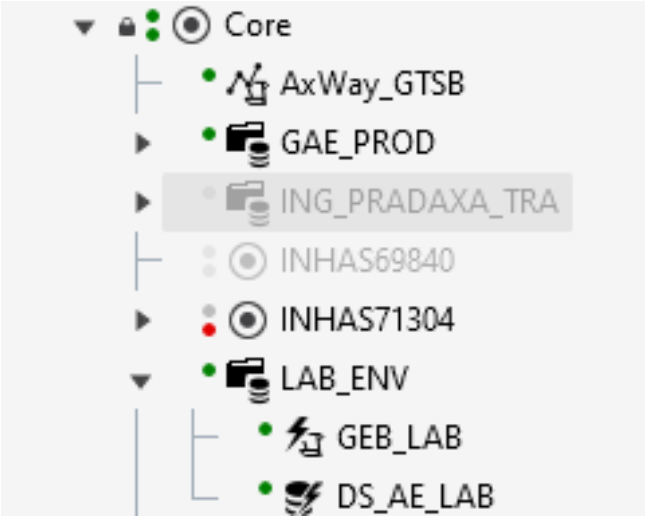


Fig. 3: Datastore

El datastore conté dos elements: GEB_LAB, es resumeix com buffer per a les dades i DS_AE_LAB per l'emmagatzematge de dades a la MongoDB. Seguidament es procedirà a crear el connector (TEST_SFM_LAB_ENVIRONMENT) (4), que serà l'encarregat de transmetre les dades des de el PLC al datastore, per tant, un dels paràmetres de la configuració al addin d'Excel, en aquest pas serà el nom del datastore. Un cop generat el connector, es procedeix a crear l'element pont entre el PLC i el sistema Inmation, aquest pas es denomina com integració. Aquest element anomenat datasource (LAB_ENVIRONMENT_DATASOURCE) conté un arbre anomenat "Equip" (4) . Aquesta ruta és la que utilitzarà Inmation Webstudio per referenciar les dades generades pel PLC.

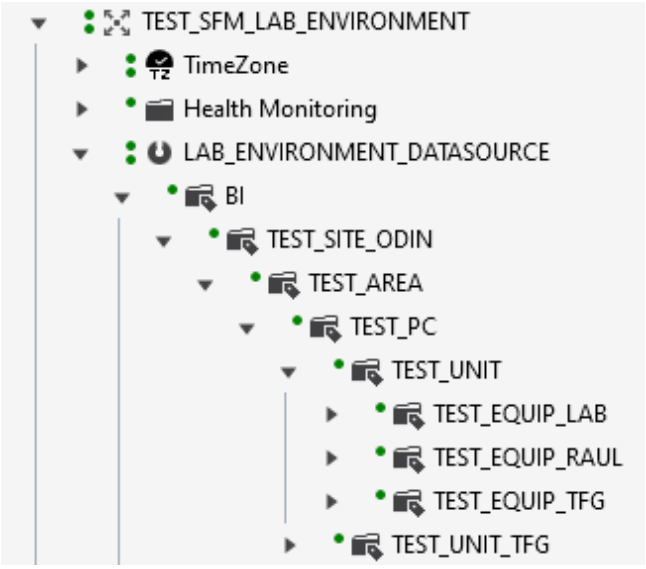


Fig. 4: Estructura Connector: Equips

Finament, el darrer pas d'aquesta integració consistirà a indicar quines son les dades generades pel PLC. En aquest últim pas és molt important indicar el tipus de dada per tal que el sistema pugui tractar-la correcta-ment, aquestes dades s'anomenen "Tags"(5):

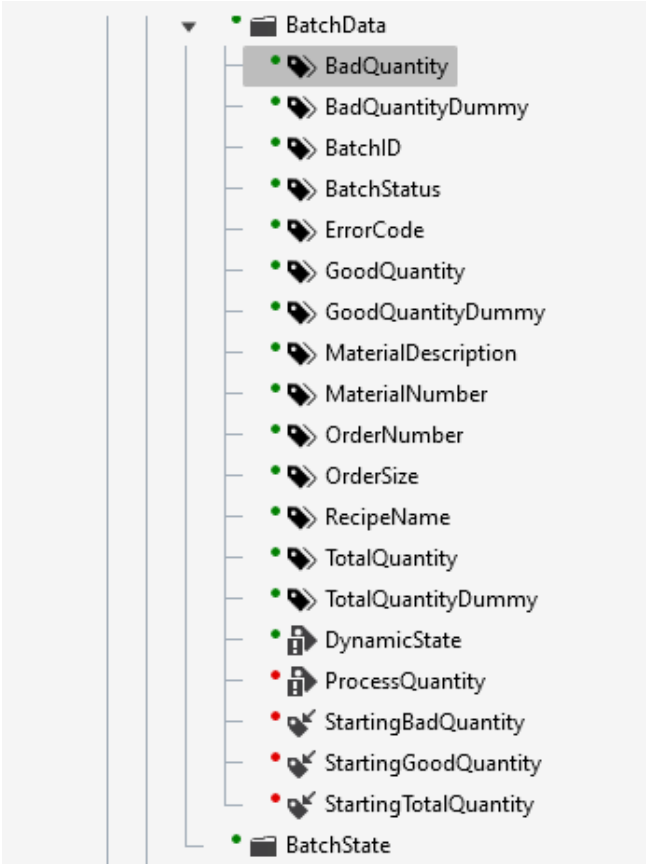


Fig. 5: Tags

Tot aquest conjunt de tags són el punt de sortida generats pel PLC que posteriorment seran utilitzats per mostrarse al dashboard.

2.2.2 Inmation Webstudio

Inmation Webstudio es un client de system:inmation en forma d'interfície web que accedeix a les dades proveïdes per Inmation Datastudio. Aquest webservice permet un accés ràpid i complet en temps real de la informació i permet una interacció entre l'operari i el sistema, mitjançant la visualització efectiva de gràfiques, indicadors i altres elements interactius com per exemple formularis, selectors i actuadors, mitjançant una API de web.

Aquesta aplicació web, programada pel proveïdor en llenguatge Angular, permet incloure els elements descrits anteriorment en forma d'estructura JSON per part de l'equip de desenvolupadors. Dins l'empresa existeixen diferents dashboards els quals permeten visualitzar diferent tipus d'informació, en el cas d'aquest projecte s'ha decidit utilitzar el dashboard OEE-Realtime (A.5 Dashboard), el qual proporciona informació d'un procés a la línia de producció.

3 Utilització de l'eina

Primerament s'accedirà al Dashboard OEE-Realtime (A.5 Dashboard) mitjançant el buscador web. El primer que apareix és un selector i l'usuari haurà de seleccionar l'equip que es vol monitorar (4) en aquest cas, TEST_EQUIP_LAB; aquest equip conté els tags (5) proporcionats pel PLC. Un

cop seleccionat l'equip s'obrirà el Dashboard mostrant l'estat inicial del batch "IDLE". Aquest estat indica que el batch està a l'espera de què l'usuari introdueixi les dades al PLC.

Paral·lelament, s'obrirà el programa WinCC Runtime Loader amb el PLC compilat. El PLC (A.1 Panell PLC) mostrarà la pantalla amb totes les dades actuals on l'estat serà el mostrat al dashboard. El primer pas serà introduir les dades del batch i seguidament es pressionarà el botó de "Transfer". Un cop transferides les dades podrem comprovar com els valors al dashboard s'actualitzen instantàniament. A continuació es procedirà a pressionar el botó de RUN, per passar de l'estat IDLE al estat STARTING. En aquest estat es posaran en marxa diferents comptadors que comptabilitzen diferents mètriques del procés. Un cop completat l'estat STARTING es pot procedir a l'estat RUNNING. Aquest últim estat permet començar a afegir les diferents quantitats, manualment (A.1 Panell PLC (2)) o de forma gradual (A.1 Panell PLC (3)), i són les que donen sentit a les mètriques del dashboard de rendiment, disponibilitat, quantitats per segon, etc. A més a més, aquest estat possibilita procedir als estats contigus HELD i PAUSED, necessaris per indicar que un procés pot ser pausat durant la seva execució. En qualsevol estat es podrà enviar codis d'error (A.1 Panell PLC (3)) i un cop realitzat el procés si l'usuari ho requereix es procedirà a completar el batch passant a l'estat COMPLETE. Independentment o seguint el diagrama, aquests processos es podran passar en qualsevol moment a STOPPED o ABORTED i per reiniciar aquest procés, es podrà indicar a través del polsador Reset, tornant a l'estat inicial de IDLE.

4 ANÀLISI DEL BENEFICI

Com s'ha esmentat anteriorment, el desenvolupador treballa sobre les llibreries existents sobre les dades històriques generades per equips reals. Això comporta una sèrie d'implicacions les quals limiten i afecten l'hora d'idear solucions al cas d'ús sorgit. A continuació s'exposaran una sèrie de problemàtiques, la solució proposada amb aquest projecte i el benefici obtingut:

4.1 Problemes

El principal problema, descrit anteriorment és la disponibilitat de dades adients per utilitzar-los a l'hora de testejar una solució a un bug o una nova "feature" necessària pel dashboard. Fins ara el procediment variava depenent del cas d'ús, però l'usual en aquest cas era disposar de les dades existents; o per part dels desenvolupadors, una solució més adient era generales mitjançant un script. Aquesta solució comportava certes limitacions, la principal; no es podria modificar certs elements o tags en temps real, és a dir, un cop executat l'script, si es volia fer un canvi s'havia d'esperar a la seva execució. Això comporta una espera per part del desenvolupador i endarrereix la seva feina. Un altra problemàtica és a l'hora d'introduir segons quines dades. No sempre és possible provar tots els casos, ja que, com s'ha esmentat anteriorment, el mètode de simular un batch és a través d'un script executat en llenguatge LUA, per tant, aquest ha de ser compilat i és possible l'aparició errors que no tenen a veure amb el testeig del dashboard, això implica un contratemps afegit i generalment dificulta encara més el desenvolupament del cas d'ús.

Una altra problemàtica és degut a la naturalesa d'aquest cas d'ús. La monitorització implica una variable temporal, es a dir, per veure una tendència, com per exemple el rendiment, el temps en que es generen les "Good Quantities" o "Bad Quantities" a plaer, és un factor que no pot ser manipulat en temps d'execució del script.

4.2 Solucions

La solució plantejada proporciona l'habilitat de solventar les problemàtiques descrites anteriorment:

- Modificació dels tags en temps real depenent de la necessitat de l'usuari
- Possibilitat d'introduir dades en formats no previstos per testejar la resiliència del dashboard.
- Reducció quantitativa del temps de testeig, ja que, la transmissió de les dades es instantània.

En general aquesta eina proporciona un nivell molt més ample a l'hora de procedir amb les proves del Dashboard.

4.3 Beneficis

4.3.1 Implementació robusta

L'habilitat de poder testejar casos poc usuals com l'exemple presentat a l'apartat 1.1 Objectius proporciona un nivell més profund de testeig, proveint una capa extra de resiliència al sistema, proporcionant informació valuosa davant d'imprevistos que podrien sorgir a les línies de producció; mitjançant aquesta informació el valor que aporta és molt útil per preparar el dashboard sobre situacions que podrien endarrerir el procés de producció estalviant costos temporals i monetaris davant d'un error inusual.

4.3.2 Descongestió del sistema

Com s'ha descrit anteriorment, per a la realització de proves, anteriorment s'utilitzava un script que s'havia d'anar executant a l'entorn de treball Inmation Datastudio. Això implica l'ús de recursos del Core, el qual executa tots els elements de forma paral·lela; l'execució d'aquest script necessita recursos adicionals, ja que també s'executa de forma paral·lela. En aquest cas, la possibilitat de poder calcular i generar dades a través d'un PLC millora en general el rendiment del sistema, ja que s'assimila a un sistema més real i, per tant, està en línia amb la idea de tota la infraestructura de l'empresa.

4.3.3 Estalvi de costos

El principal estalvi és el temporal. Aquesta eina agilitza en gran mesura el desenvolupament de codi, ja que com s'ha comentat anteriorment, possibilita el testeig en temps real sense haver d'utilitzar codi adicional en forma de script i elimina la problemàtica de la limitació temporal per part de la naturalesa d'un procés de la línia.

Aquest últim benefici en comporta un altre: si es redueix el temps per testejar una solució, inevitablement repercuteix en el cost que l'empresa té sobre el desenvolupament de codi.

5 Impacte dels beneficis

El principal impacte d'aquest projecte te cabut en l'àmbit temporal i afecta directament al cost per cada desenvolupador. Ja que es tracta d'una solució que es presenta en forma d'eina utilitzada per casos d'ús on es desenvolupa una solució per una problemàtica, el càlcul de l'impacte es farà tenint en compte l'experiència prèvia d'aquesta eina en l'àmbit del testeig i el temps mitjà que es triga per realitzar una tasca de testeig; agafant com a punt de partida el mètode anterior (utilització d'un script i dades existents) i la utilització del Digital Twin elaborat en aquest projecte.

Es posarà per cas, que és necessari testejar un batch el qual passa pels estats IDLE, STARTING, RUNNING, COMPLETED, on a part es vol veure 5 quantitats de 10, 20, 30, 40, 50 unitats de producte cada ordre.

TAULA 1: Operacions per eina

Eina	Operacions	Temps aproximat
Script	5(cada quantitat)	5x2min = 10min
Digital Twin	1	Canvis Instananis

Tal com s'observa a la taula (1) la millora és pràcticament absoluta ja que, com s'ha esmentat abans, s'elimina el factor de les esperes per execució del script.

6 Escalabilitat

S'ha estudiat la possibilitat d'escalar el projecte a altres casos d'ús. Un d'ells es podria utilitzar en el dashboard ODINS-EYE, el qual a part de veure algunes de les mètriques anteriorment vistes, inclou la visualització d'alarmes.

Les alarmes són una sèrie de senyals generades per diferents PLC's que indiquen dades com temperatures límit, quantitats excessives de producte i altres mètriques rellevants. Aquestes alarmes es generen si és compleixen una sèrie de condicions com per exemple una temperatura massa elevada en algun procés de producció. A continuació podem veure un exemple de com es podria implementar:

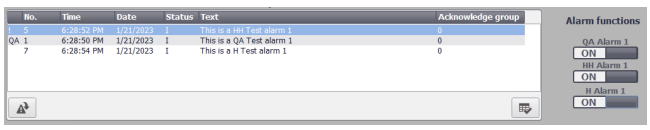


Fig. 6: Alarmes

Es pot observar com, els actuadors de la dreta poden activar diverses alarmes a plaer de l'usuari, que poden ser enviades immediatament al sistema d'Inmation, per a realitzar diferents proves.

El propòsit d'aquesta captura és el de mostrar que la facilitat d'implementació d'altres casos d'ús pels diferents dashboards usats a l'empresa es pot realitzar de manera fàcil i eficient sense posar en compromís altres funcionalitats. Com es pot observar les possibilitats són quasi infinites.

7 Conclusions

Elaborar un projecte en una empresa proporciona un nivell més elevat de compromís. Formar part d'un equip ajuda a entendre millor la problemàtica presentada. En moltes ocasions s'ha desenvolupat aquest projecte paral·lelament amb altres tasques més prioritàries pròpies de l'equip, endarrerint la formulació d'aquest.

Gràcies a la solució presentada s'ha pogut no només agilitzar aquesta feina sinó que també s'ha proporcionat informació valuosa al desenvolupament del cas d'ús orientat a l'eina. A part del descrit anteriorment, també millora l'entorn Inmation Datastudio, un objectiu no descrit i que ha resultat ser positiu.

En definitiva es pot concloure que el projecte elaborat s'ha pogut completar de manera satisfactòria, s'han assolit els objectius proposats al principi del treball i a part, s'han dut a terme reunions posteriors per a determinar l'escalabilitat i les possibilitats més enllà de la versió actual.

Agraïments

En primer lloc, voldria donar les gràcies a la meva família pel suport atorgat durant aquesta etapa universitària i dedico aquest treball al meu gran amic Enzo Ciompi, ja que no podria haver cursat aquest any universitari sense la seva ajuda.

Per altra banda, voldria agrair al meu tutor del treball de recerca, Ramon Grau Sala per guiar-me i donar feedback en tot moment.

Per últim, donar les gràcies al meu equip del Front-End, als Product-Owner Marco Gallo, Sergi Alà i Josep Reverter i la meva responsable d'equip Montse Portella. Sobretot donar les gràcies al mentor d'aquest projecte per part de l'empresa, l'enginyer Alexander Dussler, qui m'ha ajudat en tot moment malgrat les limitacions temporals degut al seu càrrec dins l'empresa.

Referències

[1] SIEMENS, ≪SIEMENS,≫ © Siemens 1996 – 2023, [En línia]. Available: https://www.siemens.com/global/en/company/stories/industry/ the-digital-twin.html. [Últim accés: 29 08 2022].

[2] Boehringher Ingelheim, ≪Boehringher Ingelheim,≫ © 2010-2023 Boehringer Ingelheim España, S.A., 01 01

2023. [En línia]. Available: https://www.boehringer-ingelheim.es/. [Últim accés: 3 11 2022].

[3] SIEMENS, ≪SIEMENS,≫ © Siemens México 2023, 2023. [En línia]. Available: https://new.siemens.com/mx/es/productos/ automatizacion/ industry-software/automation-software/tia-portal/software.html. [Últim accés: 22 08 2022].

[4] Inmation, ≪Inmation,≫ Copyright ©2013-2023 inmation Software GmbH. All Rights Reserved. , [En línia]. Available: https://www.inmation.com/en/. [Últim accés: 12 9 2022].

APÈNDIX

A.1 Panell PLC

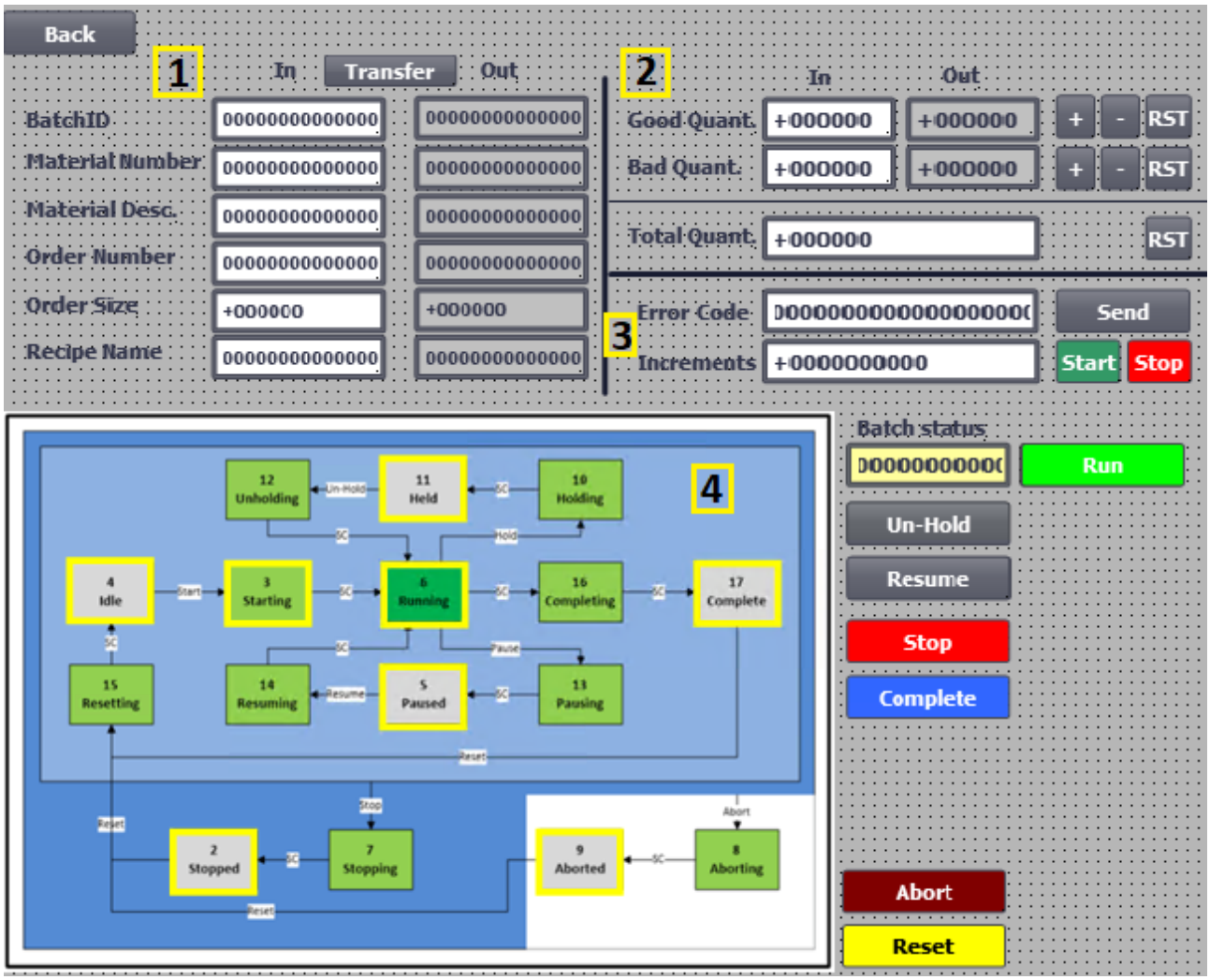


Fig. 1: Aquest panell representa la vista que té l'usuari o desenvolupador. Els números dintre dels requadres grocs indiquen els diferents blocs descrits al treball (A.1 Panell PLC (X)) on X es refereix al numero dins cada requadre.

A.2 Màquina d'estats

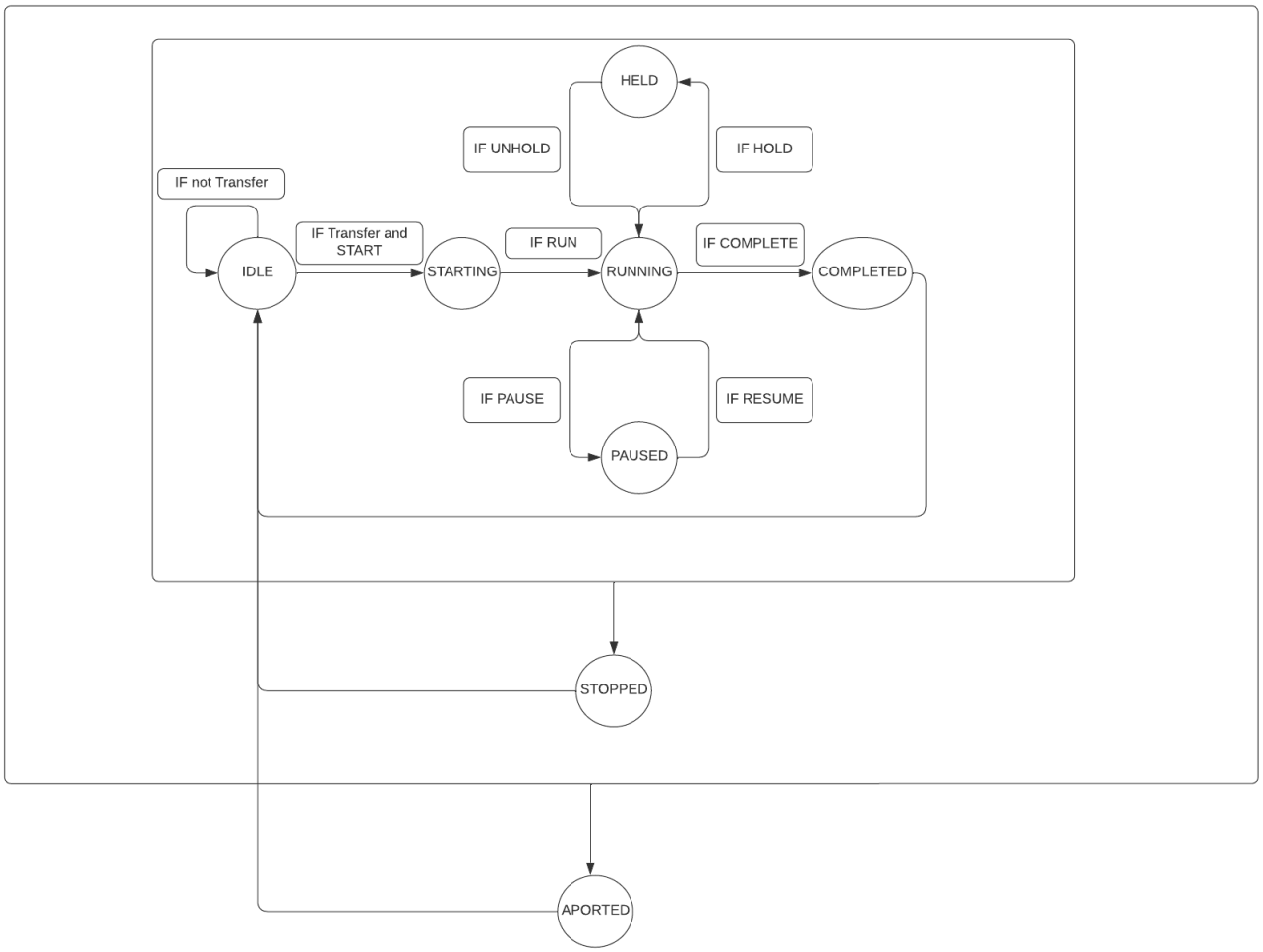
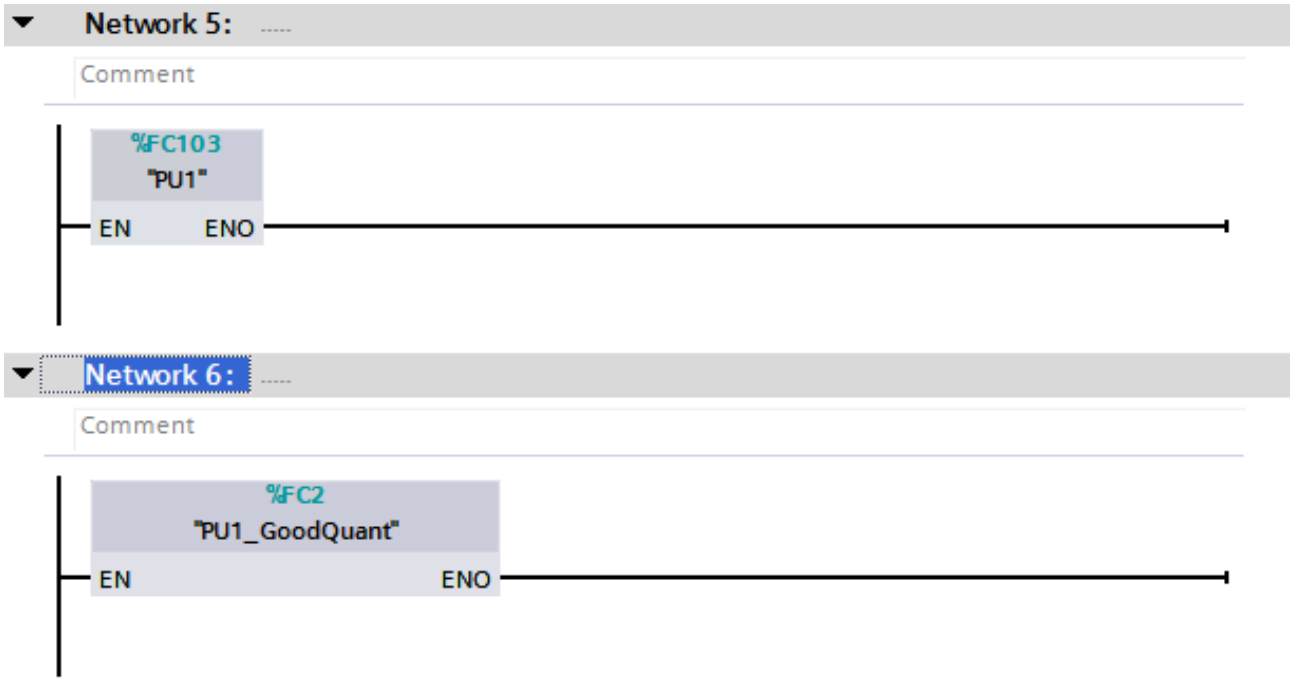
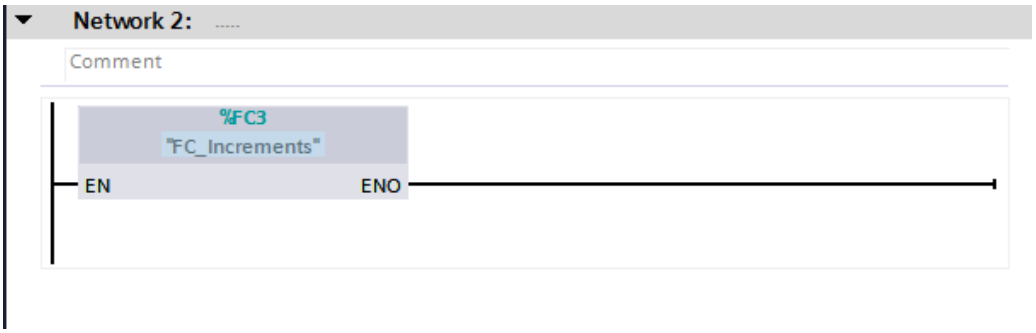


Fig. 1: Màquina d'estats que descriu el codi SCL programat al PLC.

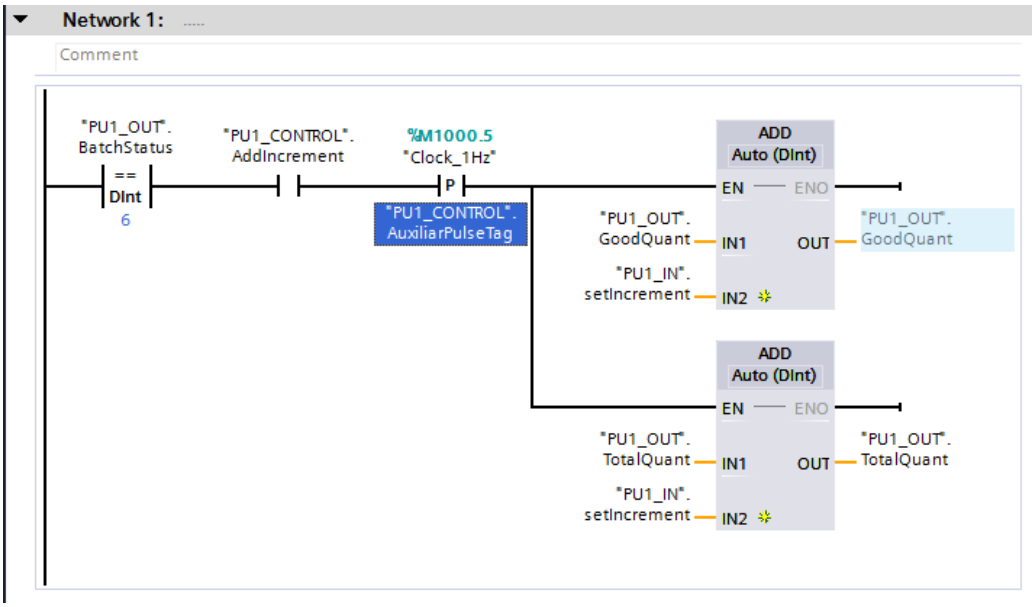
A.3 Logica PLC

A.3.1 Main





A.3.2 FC_INCREMENTS



A.3.3 PU1

```
IF "PU1_CONTROL".TransferBatchInfo THEN
    "PU1_CONTROL".TransferBatchInfo := 0;
    "PU1_OUT".BatchID := "PU1_IN".setBatchID;
    "PU1_OUT".MaterialNumber := "PU1_IN".setMaterialNumber;
    "PU1_OUT".OrderNumber := "PU1_IN".setOrderNumber;
    "PU1_OUT".Recipename := "PU1_IN".setRecipename;
    "PU1_OUT".MaterialDescription := "PU1_IN".setMaterialDescription;
    "PU1_OUT".OrderSize := "PU1_IN".setOrderSize;
    "PU1_CONTROL".Idle := 1;
    "PU1_CONTROL".AllowStateMachine :=1;
END_IF;

IF "PU1_CONTROL".SendErrorCode THEN
    "PU1_OUT".ErrorCode := "PU1_IN".setErrorCode;
    "PU1_CONTROL".SendErrorCode := 0;
END_IF;

CASE "PU1_OUT".BatchStatus OF
    3:
        "PU1_CONTROL".ShowAbort := 1;
        "PU1_CONTROL".ShowHold := 0;
```

Fig. 1: Codi d'exemple que mostra un estat i dona una idea general de com és aquesta part en contret. No s'inclou el codi a la seva totalitat ja que es massa extens.

A.4 Database

PU1_CONTROL (snapshot created: 1/11/2023 5:03:47 PM)

	Name	Data type	Start value	Retain	Accessible f...	Writa...	Visible in ...	Setpoint
1	Static							
2	RunBatch	Bool	false		☑	☑	☑	
3	CompleteBatch	Bool	false		☑	☑	☑	
4	SendErrorCode	Bool	false		☑	☑	☑	
5	HoldBatch	Bool	false		☑	☑	☑	
6	UnHoldBatch	Bool	false		☑	☑	☑	
7	AbortBatch	Bool	false		☑	☑	☑	
8	PauseBatch	Bool	false		☑	☑	☑	
9	StartBatch	Bool	false		☑	☑	☑	
10	StopBatch	Bool	false		☑	☑	☑	
11	ResumeBatch	Bool	false		☑	☑	☑	
12	ResetBatch	Bool	false		☑	☑	☑	

PU1_IN (snapshot created: 1/11/2023 5:03:47 PM)

	Name	Data type	Start value	Retain	Accessible f...	Writa...	Visible in ...	Setpoint
1	Static							
2	setIncrement	DInt	0		☑	☑	☑	
3	setMaterialDescription	String	''		☑	☑	☑	
4	setOrderSize	DInt	0		☑	☑	☑	
5	setBadQuant	DInt	0		☑	☑	☑	
6	setGoodQuant	DInt	0		☑	☑	☑	
7	setErrorCode	String	''		☑	☑	☑	
8	setBatchID	String	''		☑	☑	☑	
9	setOrderNumber	String	''		☑	☑	☑	
10	setMaterialNumber	String	''		☑	☑	☑	
11	setRecipename	String	''		☑	☑	☑	

PU1_OUT (snapshot created: 1/11/2023 5:03:47 PM)

	Name	Data type	Start value	Retain	Accessible f...	Writa...	Visible in ...	Setpoint
1	Static							
2	YELLOW_TEST	Bool	false		☑	☑	☑	
3	TotalQuant	DInt	0		☑	☑	☑	
4	ErrorCode	String	''		☑	☑	☑	
5	BadQuant	DInt	0		☑	☑	☑	
6	GoodQuant	DInt	0		☑	☑	☑	
7	BatchStatusInfo	String	''		☑	☑	☑	
8	YellowStarting	Bool	false		☑	☑	☑	
9	YellowAbort	Bool	false		☑	☑	☑	
10	YellowHeld	Bool	false		☑	☑	☑	
11	YellowPause	Bool	false		☑	☑	☑	
12	YellowStop	Bool	false		☑	☑	☑	

Fig. 1: Dades d'entrada, sortida i control i els seus tipus, utilitzades en el projecte.

A.5 Dashboard

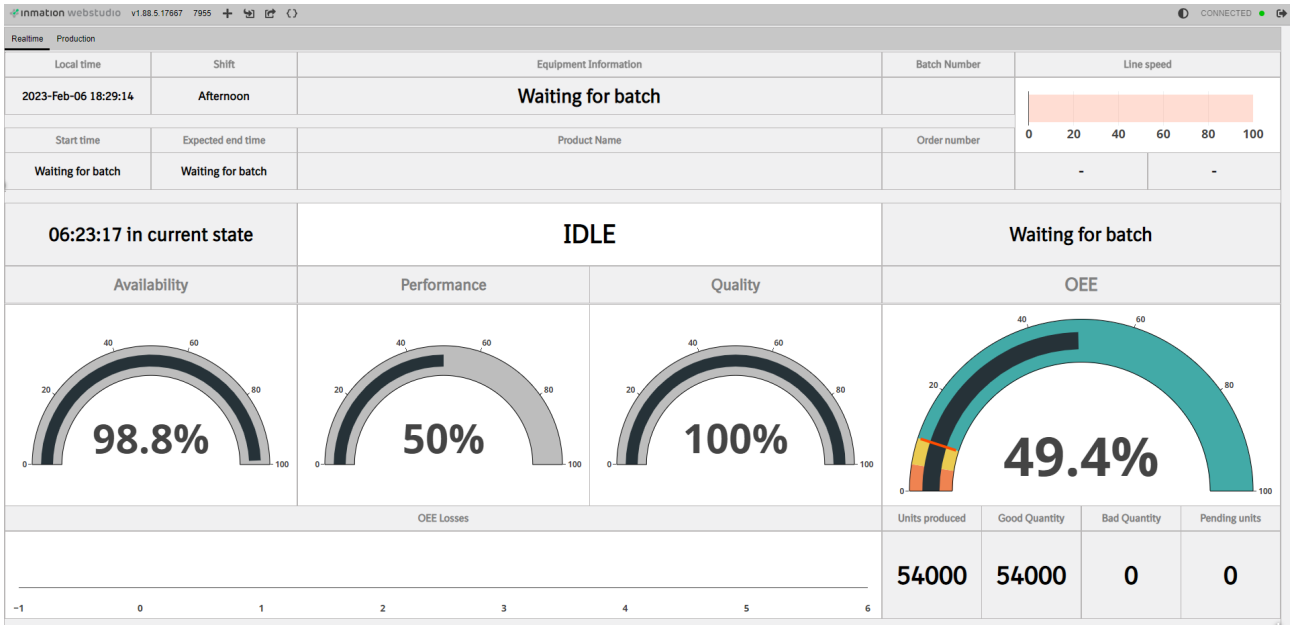


Fig. 1: Dashboard OEE Realtime, sobre el qual anirà dirigit el treball. Està elaborat pels diferents desenvolupadors de l'equip i mostra l'estat d'un batch a la línia de producció i es pot accedir mitjançant una direcció web. Entre les seves característiques principals esmentades al treball, es troba la possibilitat de mostrar rendiments depenent dels estats PAUSE o HOLD i les quantitats GOOD/BAD.