# Industrial Informatics

Masters in Electrical and Computer Engineering

2023/2024

## *Automation of a Production Line*

**WARNING**

The customer may need to make changes to the specifications up to 21 days before the delivery deadline.

Mário de Sousa 2024-02-09 ver. 1.1

## Change Log

| Version | Date | Changes |
| --- | --- | --- |
| 1.0 | 2024-02-09 | Initial published version |
| 1.1 | 2024-02-09 | Added High Level Architecture diagram/section (figure 3) |

# 1 Introduction

This document describes the work you will be doing throughout the Industrial Informatics course. In a simplified way, it consists of:

**Requirement 1**: automating the production of a flexible production line simulator;

**Requirement 2**: creating a Supervisory and Control interface to monitor the production line status and control the related automated control;

**Requirement 3:** development of a MES (Manufacturing Execution System) to monitor and manage the production on the plant floor;

**Requirement 4:** development of an ERP (Enterprise Resource Planning) module to receive orders from clients and schedule acquisition orders as well as production orders.

Typically the lowest level control (requirement 1) will use PLCs (Programmable Logic Controller) or equivalent equipment. The supervisory and control interface (requirement 2) will be based on a SCADA (Supervisory Control And Data Acquisition) solution. When implementing the higher level requirements (3 and 4) it is advisable to use higher level programming languages and tools (C++, Java, Python, C#, …). At these levels the use of a DBMS (Data-Base Management System) is also mandatory (PostgreSQL, MariaDB, ...).

A separate document describes the equipment in the production line and the interfaces provided to interact with it. The document you are currently reading describes the requirements that the control, monitoring, management and scheduling applications must comply with. To better understand these requirements we suggest you first peruse the document describing the the production line equipment.

# 2 Overview of the Flexible Production Line

The Flexible Production Line to be automated consists of 11 cells (Fig. 1 & Fig. 2):

- 2 automatic warehouses (W1, W2),
- 1 transfer cell with 3 conveyors to transfer workpieces from A2 to A1 (T),
- 6 machining cells with 2 machines each (C1 to C6),
- 1 loading cell with 4 loading docks (L).
- 1 unloading cell with 4 unloading docks (U).

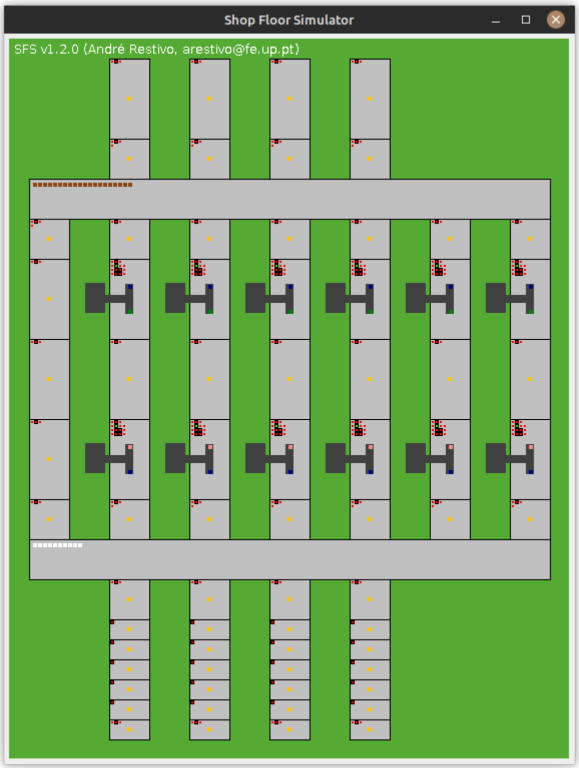


*Figure 1: Flexible Production Line – equipment layout.*

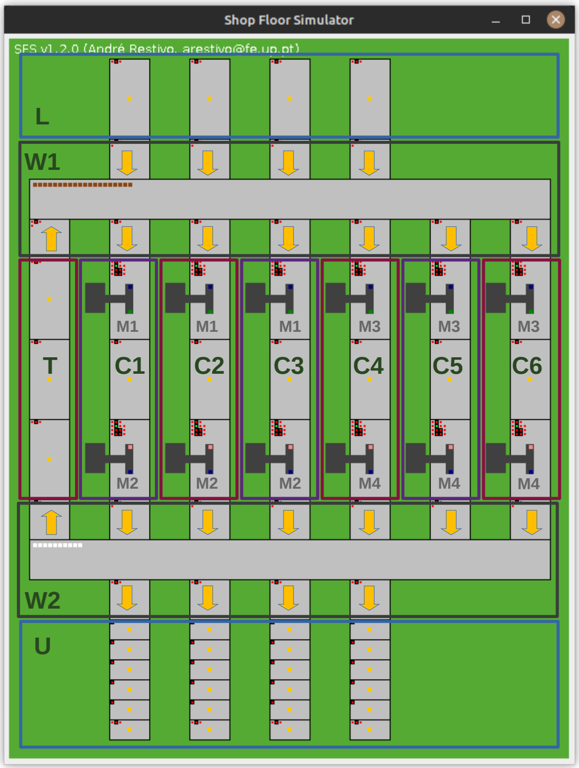


*Figure 2: Flexible Production Line – annotated equipment layout.*

The annotations in Figure 2 help in understanding the equipment capabilities.

- The large horizontal rectangles with brown and white dots/squares represent automatic warehouses (A1 and A2). The coloured dots represent the workpieces currently stored in that warehouse. The workpiece type is represented by the colour.
- All conveyors (grey or bluish grey rectangles) are bi-directional. The small yellow dots on the conveyors represent sensors that can detect the presence of a part/work-piece;
- Conveyors marked with an orange arrow can either insert (load) or remove (unload) work-pieces from the respective warehouse. Each of these stations is dedicated to either loading or unloading.
- The equipment in black represent machines, able to transform a work-piece from one type to another.
- Typically raw material (raw parts) will arrive on the top loading stations of A1. These parts will be unloaded onto one of the 6 processing cells where they be transformed into another part/workpiece type, and then stored in A2. Transformed workpieces are then unloaded into the loading docks. The transfer cell allows to transfer workpieces from A2 back to A1.

This production line will be simulated by the SFS software (Shop Floor Simulator) which, being developed in Java, is compatible with Windows, Linux and OSX. Access to the logic signals (sensors and actuators) is done through the Modbus/TCP protocol. The SFS simulator implements a Modbus/TCP server, in which the sensors (Boolean signals) are mapped to Input Discretes, and the actuators (Boolean signals) on to Coils. The addresses of each signal are available in a file io.csv (format comma separated values, which can be opened as a spreadsheet). This file is generated by the simulator itself with the command:

```
java -jar sfs.jar --csv io.csv
```

The description of the signals available to control each equipment (machine, conveyor, rotating conveyor, pusher and warehouse) is described in the accompanying document.

Each machine has a private warehouse of three tools, and can change the tool mounted on the machine from one of the three tools in the respective tool warehouse. Changing between any two tools takes 30 seconds.

The tools available in the private tool warehouse of each machine are listed in the following table:

| Machine | Tools |
|---|---|
| M1 | T1, T2, T3 |
| M2 | T1, T2, T3 |
| M3 | T1, T4, T5 |
| M4 | T1, T4, T6 |

*Table 1: Tools available on each machine*

## 2.1 The Production (Transformation) Process

The production line processes different types of parts/work-pieces (P1, P2, ...), represented by blocks with different colors. The color convention used is defined in the following table (same color convention as for electrical resistors):

| Piece | Colour |
|---|---|
| P1 | Brown |
| P2 | Red |
| P3 | Orange |
| P4 | Yellow |
| P5 | Green |
| P6 | Blue |
| P7 | Violet |
| P8 | Grey |
| P9 | White |

*Table 2: Mapping of work-piece type to colour .*

When processing a part of a certain type on one of the machines, using a given tool, for a certain period of time, this part changes its type.

The transformations of parts that may occur depend on the type of tool, and are described in the following table:

| Starting Piece | Produced Piece | Tool | Processing Time |
|---|---|---|---|
| P1 | P3 | T1 | 45 s |
| P3 | P4 | T2 | 15 s |
| P3 | P4 | T3 | 25 s |
| P4 | P5 | T4 | 25 s |
| P4 | P6 | T2 | 25 s |
| P4 | P7 | T3 | 15 s |
| P2 | P8 | T1 | 45 s |
| P8 | P7 | T6 | 15 s |
| P8 | P9 | T5 | 45 s |

*Table 3: Work-piece Transformations*

If you try to perform operations that are not found in the table above (for example: processing P8 with tool T3), the part will be destroyed and turn black. Similarly, if you perform an operation described in the table above, but for a longer period of time than indicated, the part will also become destroyed. Black work-pieces should never occur in your project as machines never break down. Trying to load a black work-piece in a warehouse will result in the simulator crashing.

If you start to perform a transformation operation but stop before the necessary time has elapsed the transformation will remain incomplete, leaving the piece with the two colors (of the original piece, and of the final piece) in the correct proportion. The transformation can be completed at a later time. Work-pieces that are partially transformed may be moved out of the machines but cannot be placed in a warehouse, i.e. they must remain on the conveyors.

In order to produce a final piece it is necessary to take into account the processing sequence of the raw materials. This sequence results from the successive combination of raw material and tool, which may generate by-products for further processing. For example, if you want to produce a P4 piece from a P1, it will be necessary to apply T1 to the P1 resulting in a by-product of the type P3, followed by T2 resulting in P4.

## 2.2 Warehouses

The warehouses in cell W1 and W2 can store any work-piece type (P1 to P9). Any loading or unloading operation executed by the warehouse takes 1 seconds to complete. Although the simulator does not enforce it, you should consider the warehouse as having a limited capacity of 32 work-pieces

The production line starts with empty automated warehouses (no work-pieces).

## 2.3 The Loading Process

Cell L is used for loading raw material. The two conveyors on the left are used to load P1 work-pieces, whereas the other two conveyors on the right are reserved for P2 work-pieces. Both P1 and P2 work-pieces are considered raw material that the production line consumes, and cannot create.

The loading stations function as normal conveyors on which pieces will be manually deposited. These work-pieces will be transferred to the warehouse in cell W1.

## 2.4 The Unloading Process

Cell U is used for unloading work-pieces through the four unloading bays/docks.

Each unloading station consists of a slider. The slider is an inclined ramp fitting a maximum of 6 workpiece. A workpiece loaded onto the top of the slider will slide down by itself until it hits the bottom or the next workpiece. The bottom-most position is actually a conveyor that should normally be stopped / not moving. Unloading the slider can be simulated by actuating the bottom most conveyor and letting all the workpieces fall to the floor (where they will simply evaporate).

# 3 Enterprise Management (ERP – Requirement 4)

## 3.1 Client Orders

The company's ERP software must accept orders sent to it by its clients. These orders will be encoded in XML files, and will be sent to port 24680 of the UDP/IP protocol. Each file may include one or more client orders, coded according to the following rules:

– **Client Order:**

**<Client NameId="name"/>**
**<Order Number="nnn" WorkPiece="Px" Quantity="XX" DueDate="DD"**
**LatePen="LP" EarlyPen="EP">**
**</Order>**

**name** – Name identifying the client
**nnn** – order number
**Px** – work-piece type being ordered
**XX** – quantity being ordered
**DD** – Due date (integer)
**LP** – penalty (value in €) incurred for each day of delay in completing the order
**EP** – penalty (value in €) incurred for each day of advance in completing the order

The ordered work-pieces must be delivered on the due-date. They must not be delivered early, nor delivered late. The stated penalties apply when the due-dates are not satisfied.

Clients will only order workpieces of types P5, P6, P7, and P9. Workpieces of type P3, P4 and P8 are all considered intermediate products that are not sold by the company.

## 3.2 Plans

Your ERP software is expected to generate a Master Production Schedule (MPS) into the future, **if possible** trying to maximize enterprise profit. Usually this means satisfying all client orders so as to avoid penalties, of minimizing acquisition costs of raw material, and of minimizing overall production costs. This MPS should define for each day in the future:

- the amount of work-pieces of each type you expect to have by the end of that day
- the capacity used in each warehouse

Based on the MPS you must generate a production plan as well as a purchasing plan. The production plan defines what orders to send to the MES on each day, while the purchasing plan defines when to place orders for purchasing raw material (P1, P2) from your suppliers. You have available only 3 suppliers described in the following table:

| Supplier | Piece | Minimum order | Price per piece | Delivery time |
|---|---|---|---|---|
| SupplierA | P1 | 16 | € 30 | 4 days |
| SupplierA | P2 | 16 | € 10 | 4 days |
| SupplierB | P1 | 8 | € 45 | 2 days |
| SupplierB | P2 | 8 | € 15 | 2 days |
| SupplierC | P1 | 4 | € 55 | 1 day |
| SupplierC | P2 | 4 | € 18 | 1 day |

*Table 4: Raw material suppliers*

The ERP must provide a user interface that shows the resulting plans (MRP, Purchasing and Production plans), as well as the current date. The plan should also include a calculation of the costs incurred in satisfying each client order.

## 3.3 Unit Cost

The cost (per work-piece) for each order should be determined using the following formula:

$$Tc = Rc + Pc + Dc$$

Tc – Total Cost
Rc – Raw Material Cost (price of the raw material used to produce that piece)
Pc – Production Cost (Cost to Produce the piece)
Dc – Depreciation Cost (Cost of money invested in the piece)

$$Dc = Rc \times (Dd - Ad) \times 1\%$$

Ad – Arrival Date – date the raw material arrived at the production line
Dd – Dispatch Date – date final work-piece leaves the production line (unloaded on cell E)

$$Pc = € 1 \times Pt$$

Pt – Total Production time (in seconds).

The production time (Pt) is the amount of time a work-piece spends inside a machine, whether the machine is busy doing a transformation or not (e.g. the machine may be waiting for the output conveyor to become free, or changing a tool, while the workpiece is loaded).

Notice too that the work-pieces produced for the same client order may have different values of Total Cost. The raw material cost may be different (ordered from different manufacturers), as may the production cost (time spent in the machines) and the Depreciation Cost (time spent in the manufacturing line). The ERP should therefore show the average cost for each piece in the client's order.

**NOTE**

So as to speed up the simulation, we will consider each day to be equivalent to 60 seconds of simulation time.

**NOTE**

You **DO NOT** **need** to implement optimization algorithms at the ERP level that try to reduce total cost to the enterprise. You only need to calculate the total cost of each client order.

However, a competition might be held between student groups. This competition will be won by the group that completes all orders with the minimum cost. To win this competition you **may** need to implement an optimization algorithm.

# 4 Managing the Plant floor (MES – Requirement 3)

## 4.1 MES Orders

The Manufacturing Execution System (MES) receives production orders form the ERP and manages the execution of these orders on the plant floor. It is up to you to define how the ERP and the MES interact, both logically (i.e. what information to send/receive and when) and technologically (i.e. how that information is exchanged, e.g. communication protocol, …).

Typically the ERP will emit production and/or delivery orders once a day. The MES must also provide the ERP with sufficient information so the ERP can calculate the real production cost.

It is expected that the MES will prioritize the orders to be executed in such a way as to minimize production costs, while guaranteeing that the warehouse capacity is never exceeded.

The MES should also choose and indicate the unloading dock to use for each delivery order. Orders with more than 6 work-pieces must use more than one dock. Delivery orders should be placed on the unloading docks in the first part of the day (first 45 s of each 60 s day). The user will manually remove these work-pieces from the unloading dock on the remaining part of the day.

It is expected that the ERP will never ask to deliver more than 24 workpieces per day.

The MES must provide the PLC with sufficient information so that the PLC can control the manufacturing equipment. It is up to you to define when and what information to exchange between the MES and the PLC. However, the client mandates that you use the **OPC-UA** communication protocol to exchange this data.

## 4.2 User Interface

The MES should provide a user interface through which the user can monitor the current production order list, as well as the status of each order received from the ERP. The status includes the number of pieces already  produces, the number of pending pieces, the total production time of the order, etc. This information will also depend on the execution ordering algorithm used by the MES (e.g. the list of production orders in decreasing priority). The user interface should also provide enough information for the user to determine the status of that algorithm.

## 4.3 Statistics:

To allow users to monitor the production process, the MES must provide the following information:

- **Machine Statistics**

  A list with the statistics of the operation of each machine. For each machine, provide:
  - total operating time,
  - total
  - number of operated work-pieces (total and for each type).

- **Unloaded Work-pieces**

  For each unloading dock, provide:
  - the number of unloaded work-pieces (total and by type).

The interface through which the monitoring is carried out is not specified, so you are free to implement it as you see fit. If you wish, you can extend the communication protocol so that they are made available in XML format on the UDP port.

# 5 Controlling the Production Equipment (PLC – Requirement 1)

For requirement 1 (section §1) you will need to develop the software to control the equipment that is part of the cells of the production line.

You are free to use this equipment as you see fit as long as the simulator manages to produce work-pieces with the correct colour. Most notably you are free to store each work-piece on whatever warehouse you think most appropriate, or even keep it on a conveyor until needed. Work-pieces can be transported in the production line (using the conveyors) in any direction.

Please note that the simulator will have very slight timing differences depending on the computer it is running. We strongly suggest you do **not** rely on timing delays when controlling the manufacturing plant, with the sole exception when timing the work-piece transformations inside a machine. The final evaluation may occur on a computer with slight timing differences, and your code must still work reliably.

# 6 Supervisory and Control Interface (SCADA – Requirement 2)

The supervisory and control interface must be implemented using a SCADA based solution. The SCADA interface must monitor **one** of the production cells (C1 to C6), and allow the user to, at a glance, determine the current status of the cell. This includes the status of each conveyor (running or stopped) and machine (operating, stopped, current tool, list of tools in the machine's tool warehouse, etc.), and current time and date. Through this interface the operator should be able to change **each** of the machines in the cell to one of the following operating modes:

- automatic
- manual
- maintenance

In automatic mode the machine is completely controlled by the PLC. When in manual mode the operator can manually control the machine status (operating/stopped, change tool), through buttons on the SCADA interface. When in maintenance mode the machine is considered to be switched off and unavailable for production – during this time any worn-out tools may be replaced, or other maintenance made on the machine.

For the above the conveyor associated with the machine is not considered part of the machine – even in manual and maintenance modes the conveyor is still fully controlled by the PLC. However, the manual mode should have a button to allow the operator to indicate that the operation on the workpiece has been finished, allowing the PLC to remove the workpiece present on the conveyor under the machine.

The SCADA must also maintain an historical list of events, and show any alarms that may have gone off. For the moment only one alarm type is defined – an alarm is set whenever a workpiece remains in the machine (i.e. on the conveyor) for longer than 10 s after having completed processing.

Communication between SCADA and the PLC should use the Modbus/TCP protocol.

# 7 Persistence

The information managed by the ERP and MES must be persistent, i.e. it should not be lost if the ERP and/or MES have to be turned off. This requirement must be achieved using a database managed by a database management system (e.x .: MariaDB, MySQL, PostgreSQL, ...).

The information to be placed in the database should be such as to allow the ERP and MES to be disconnected and resumed later without loss of information regarding pending orders, or statistics produced up to that time. During the period in which the ERP and/or MES is disconnected it is acceptable to stop accepting new orders from the ERP or client.

With MES turned off, production operations may also be affected, and the level at which they are affected will depend on the way in which the architecture of the production system is organized.

# 8 High Level Architecture

The previous requirements embody a high level architecture that is better represented in the following diagram.

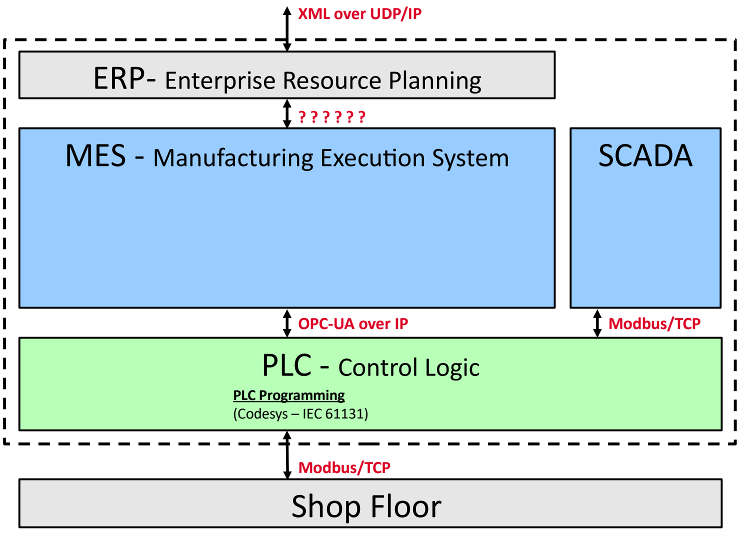


*Figure 3: High Level Architecture*

# 9 Expected Outcomes

This project should be carried out by groups of 5 students.

All the operating logic behind the control of the production line must be analyzed, designed and implemented by the students, who must identify, in addition to the components described above, all the components that will be necessary for the operation of the line. It is recommended that you start first by not considering the possible optimization of the use of available resources. However, the optimization of resources will not fail to be valued when evaluating the project.

**The expected results of the project are:**

1. **Final Report** (35% of the project grade), to be delivered by **24 May 2024.**
2. **Demonstration** and discussion of the application (65% of the project grade)
    a. the SCADA interface, in the week of **1st to 5th of Arpil 2024**.
    b. final demonstration (ERP. MES, ...) in the week of **20 to 24 of May 2024**.

**<u>Distribution of effort between group elements</u>**:

The project grades will be attributed to the group. The report should indicate how the group wants this grade to be distributed to each member of the group. This distribution must be made as a percentage in such a way that the sum of the percentages is 100.

| **final individual grade calculation** | |
|---|---|
| Final Group Grade: G (0..20) | A = Max[0; SIN(pi/2 * (Px/P - 1) / (1/P - 1 ))] |
| Weight for each group member: P1, P2, P3, P4, P5 | B = Min[0; (Px/P - 1)] |
| P = Average(P1, P2, P3, P4, P5) | C = (20-G)*G/20 * A |
| Individual Grade Gx = G + C + D | D = G * B |
| **NOTE**: The formulas expect values for P be between [0,1] (e.g., 0.25 corresponds to 25%) | |

**<u>Final Report</u>**:

The report should be written in arial font with size 11pt and spacing between lines of 1.5 lines, and a single column, with a 2cm margin around A4 pages. The page limit does not include the space used up by images, diagrams, and tables.

The report should also follow the following structure:

- (5%) **Introduction** (up to 0.5 pages of text)

Summary of the approach followed with identification of the problem, technologies used, some special characteristics of your solution, some results that you want to highlight and structure of the report. For this to fit in half a page, you will have to be succinct. Focus in providing a quick idea of what you did and what you will report on next.

- (25%) **High Level Architecture** (up to 2 pages of text, excluding images)

Description of the high level architecture proposed and implemented in your application. You should probably use UML diagrams (class, object, sequence, activity, etc.). The text should highlight the responsibilities of the main components of the architecture and refer to the respective diagrams, and describe how these components interact (when and what data/events are exchanged, etc.).

- (20%) **Comparison with Common Design Patterns** (up to 1.5 pages of text)

Critical comparison of the architecture used in the project with the standards (design patterns) proposed by the ISA 95 standard, RAMI 4.0, and the article “Service Granularity in Industrial Automation and Control Systems”.

- (25%) **Scheduling and Optimisation** (up to 2 pages of text)

Describe the scheduling algorithms you used when deciding which parts to produce, at what time, and on which machine. You should mention the algorithms used at all three levels: ERP, MES, and PLC. Justify your choice of algorithms.

- (20%) **Testing and Results** (up to 1.5 pages of text)

Describe the tests performed, in particular what complex situations were tested. Performance can also be quantified (e.g., order XXXX took xx seconds to be processed; sequence of orders X, Y and Z took yy seconds ...). Critical appreciation of these results.

- (5%) **Conclusions** (up to 1.0 pages of text)

Summary of what was said in the report, highlighting some results considered more relevant.

# 10 Competition

We are looking into the possibility of launching a competition at the end of the semester that will be won by the group that manages to complete a certain sequence of client orders for the lowest cost. The group that wins the competition will have a bonus of (t.b.d.) in the project grade. The group in second place will have a bonus of (t.b.d.).

The sequence of client orders to be used in the competition would be delivered after the project's delivery deadline. The sequence will be made available in the form of a script so that orders are launched at predetermined times (and not all at once at the start of the program).

Groups interested in competing must submit the parameters (total cost incurred). The winning groups will also have to make a live demonstration in a public session.