



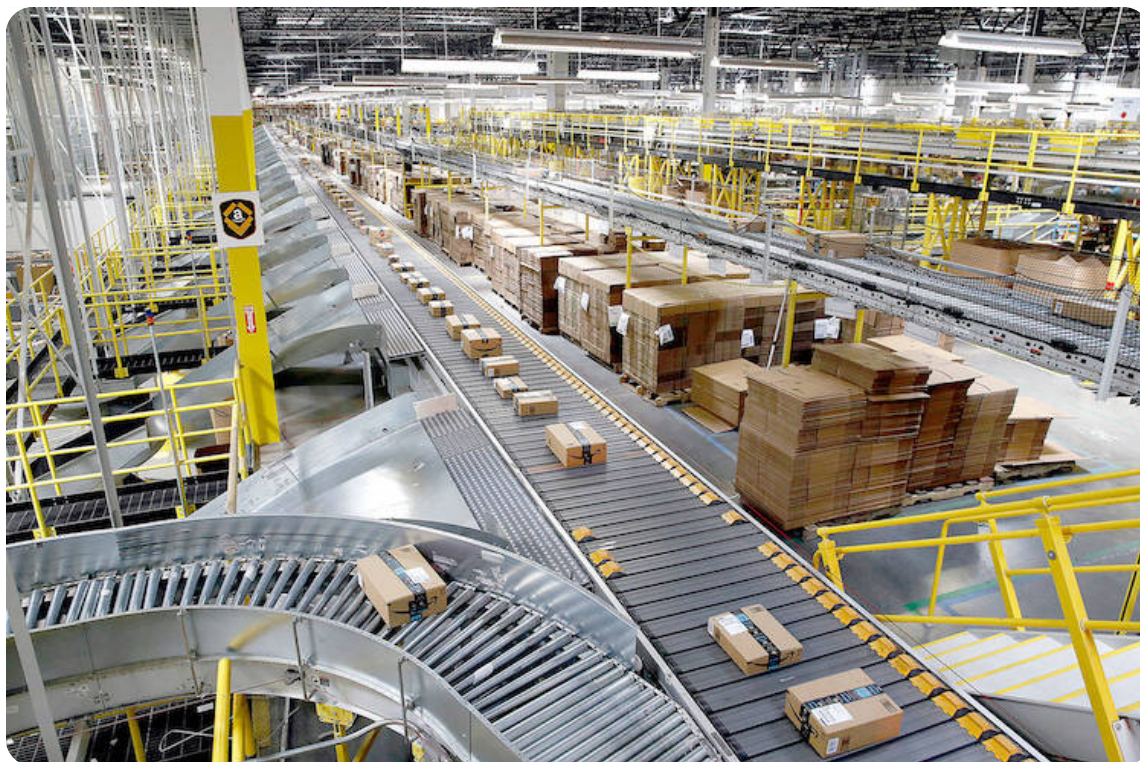
# THE CONVEYOR

TIGHT & TOUGH LOGISTICS



# THE CONVEYOR

Your customer has an old sorting conveyor that he wants to integrate into his recent installations.



The machine controller being obsolete, a new version has been developed by a service provider. It makes it possible to control all the components of the machine and to add network communication functions.

However, the service provider abruptly quit the project before being able to finalize the firmware.



The client will make available the meager elements it has to allow you to identify the makeup of the new controller



As physical access to the machine is not possible, you will be able to perform your tests remotely with the assistance of an on site technician

## Conveyor information

The conveyor receives packages directly after their manufacture and packaging by the production lines. It must direct them to one of the three storage warehouses and feed the information back to the WMS in order to track the stock.

When a package arrives on the line, the conveyor must:

- ✓ bring the package to an NFC reader ;
- ✓ read the product reference on the card ;
- ✓ query the list of WMS products to obtain its destination and other information ;
- ✓ create a stock movement in the WMS to the destination warehouse ;
- ✓ direct the package to the appropriate exit.

## Information on Warehouse Management System

The customer uses Dolibarr with *stock* module. It is possible to communicate with dolibarr using its API REST.

The system operates in a network and records all parcel movements **at each sorting step**. Data is stored on NFC tags, allowing information about the product to be available.



A user is specifically dedicated to the conveyor, the access token will be provided to you

When a package arrives on the conveyor, its tag contains an identifier that corresponds to the *product reference* in dolibarr. It allows you to find the destination warehouse.

Dolibarr shall contain at least all the products that must go on the conveyor. Each product record in the WMS contains multiple informations: default warehouse, surface, weight, etc... Three *warehouses*, A, B and C, are configured, corresponding to the three possible outputs of the conveyor.

The conveyor must then create a *stock movement* in the WMS.

## Other information

The sorter is part of a complete logistics line, the customer insists on the ability of the machine to quickly restore operation in the event of a service interruption.

The controller must be able to return to its operating state in the event of an outage (power failure, manual interruption, loss of connection, etc.)

It is essential to be able to quickly configure the controller in order to modify its dependencies (Wi-Fi, API, etc.) without needing to recompile the code.

## Deliveries

In order to validate the progress of the project, the client asks to follow the following steps:

### ✓ POC roadmap:

The customer wishes to document the composition of the controller.  
Submit an intermediary report with at least :

- Identify the controller elements ;
- Create a documentation describing the modules (software components to be used);
- Create a sequence diagram;
- Present what you are about to do with them (october).

### ✓ Testing sessions:

You'll have access to the machine, **for a limited time**, to carry out some test on:

- Handling of the controller (optional - 30 min - november) ;
- WMS communication tests (optional - 30 min - december) ;
- Firmware test and debug (mandatory - 45 min - january).

### ✓ Final delivery:

- A release note and the binary ;
- Pouvoir installer facilement le firmware avec une autre machine ;
- Associated documentation.



It is highly recommended to use PlatformIO.



The firmware **must** be written in C/C++.  
UiFlow (Sketch) and micropython are **not** allowed.

{EPITECH}

