FASTory Assembly Line - An overview for the Introduction to Industrial Informatics course

Version: 03; 09-10-2023

Tampere University of Technology

Table of Contents

1 Introduction

The FASTory assembly line is used to assemble electronic devices. FASTory research environment is located in Tampere University of Technology, FAST-Lab., Finland. The FASTory line contains 10 identical workstations, which draw the main parts of three models of a mobile phone (WS 2-6 and 8-12). In addition, there is workstation, which is used to load raw materials (empty papers) and unload products (paper with mobile phone drawing) to/from pallets (WS 1). Finally, the twelfth workstation (WS7) is use for loading/ unloading pallets to the line. See Figure 1.

Figure 1-a: FASTory Line

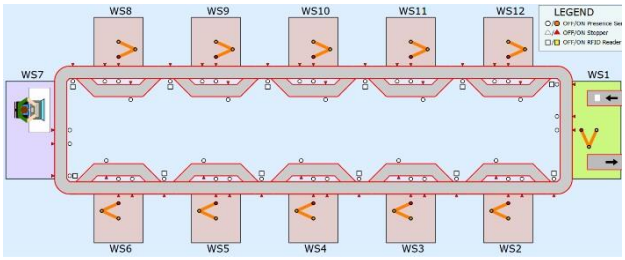


Figure 1-b: Fastory Line Layout

This document provides guidelines for the users of the FASTory line. It is divided into 7 sections; section 2 highlights the FASTory line before FAST-Lab exploitation. While section 3 presents the functionalities of the FASTory line. Section 4 lists the main components in the line. Then section 5 provides the RTUs and the APIs of the line.

2 Before FAST-Lab

Before being called the FASTory line, the line was used for assembling real mobile phones. The pallets were equipped with a special jig for holding the mobile phones during the production process. See Figure 2.

The line then was transformed for education and R&D (research and development) purposes. The modification covered the pallets, end-effectors and sensors.

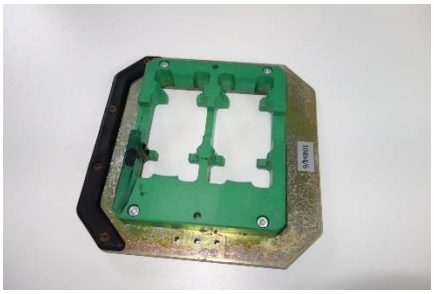


Figure 2: Old pallets

3 What the FASTory does?

The current FASTory assembly line is used to demonstrate the assembly of mobile phones by drawing mobile phone's main parts (frame, screen and keyboard) with different colors and different shapes. Figure 3 depicts a conceptual example of the FASTory final product.

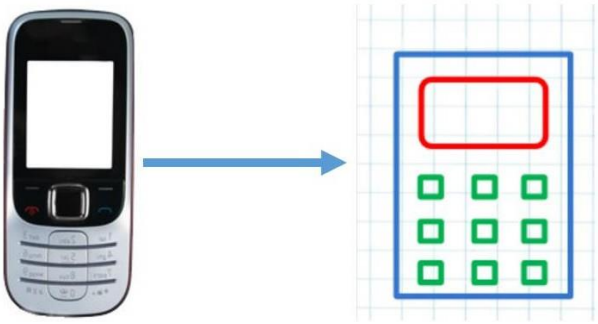


Figure 3: FASTory final product

With these combinations, the FASTory line is able to produce 729 (9 different screens x 9 different keyboards x 9 different frames) variant product. See Figure 4.

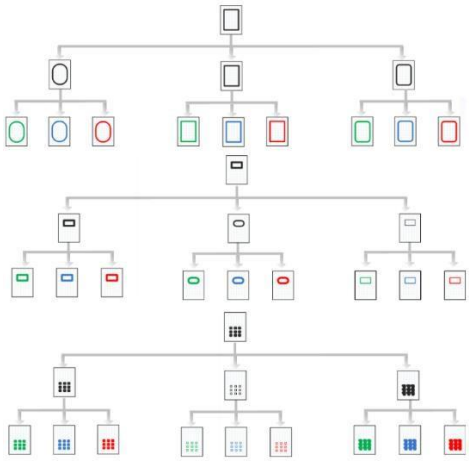


Figure 4-a: variations of FASTory line products

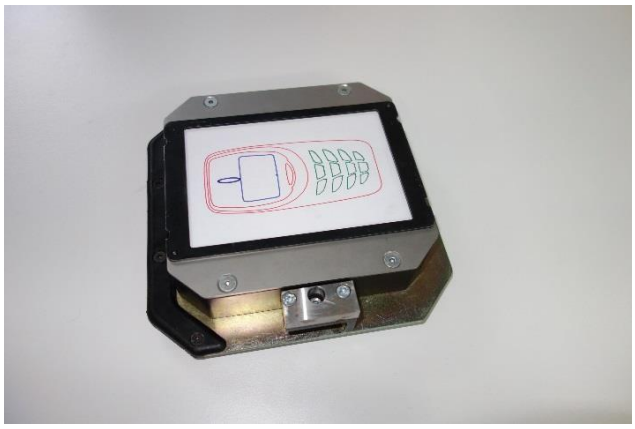


Figure 4-b: example of FASTory line products

4 FASTory Components

The FASTory line was used in many R&D projects including eSonia[1] and eScop[2] projects. Besides, the line has been continuously employed in FAST-Lab courses. Therefore, this section highlights the main components that the user could use in his/her study/research.

This section is divided into 6 subsections; pneumatic system, conveyors, robots, RFID readers, sensors and actuators.

4.1 Pneumatic System

The FASTory line requires an air pressure to work properly. The air pressure is used to actuate the track selector in the conveyor and the end effector in the robots. Thus, each work cell has a manual valve, pressure relief valve and solenoid valve as shown in Figure 5. In general, students are not expected to

[1] http://www.esonia.eu/

[2] http://www.escop-project.eu/

change any configuration in the pneumatic system. Therefore, if any problem detected in the line, students should report to the supervisor in the lab.

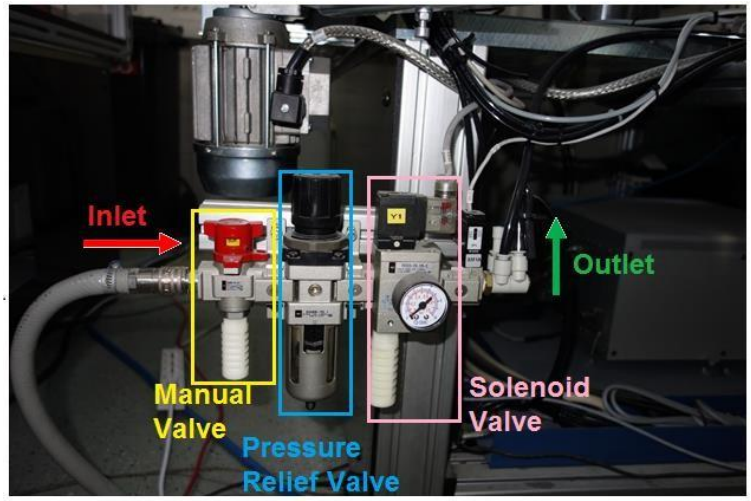


Figure 5: Pneumatic System

4.2 Conveyors

Each workstation in the FASTory line consist of two conveyors; main and bypass. The main conveyor is used if the pallet requires service from the work cell. Meanwhile, the bypass is used if the work cell is in busy state (another pallet(s) [max 2 pallets] are in the cell) to bypass the pallet to the next work cell. See Figure 6. Both Conveyors (Main and Bypass) are controlled by a single RTU.

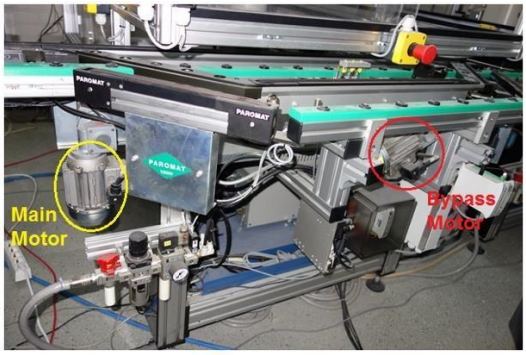


Figure 6-a: Conveyors motors

Figure 6-b: Conveyors direction

As shown in Figure 7, each conveyor in FASTory line is divided into different zones. There are five different types of zones. In each zone, there is one presence sensor that is used for detecting the pallet presence. In addition, there is one stopper to stop the pallet when the conveyor is transferring other pallets. An RFID reader is located in each zone 1 for each workstation. This reader reads the pallet's tags for identifying the entering pallet. Table 1 shows the functionality of these zones.

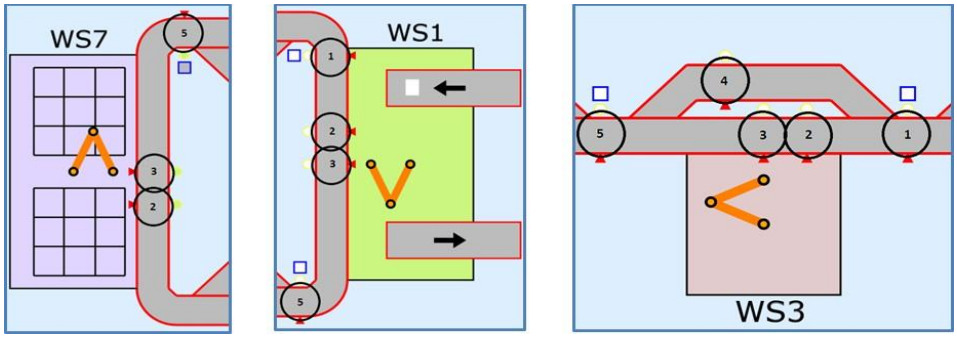


Figure 7: FASTory conveyor zones (NB! WS7 was recently turned into manual station – see Figure 1-b).

Table 1: Zones functionality of the conveyors in the FASTory line

Zone ID	Zone Functionality	Availability in FASTory
Z1	Work station entrance. This zone reads the ID of the transferred pallet from previous workstation. In this zone the decision could be taken for move the pallet to the work station or to pass it to the next work station	This zone is available in all work stations except ws7
Z2	Internal buffer for the workstations. This zone holds extra pallet in the line if Z3 is occupied. An exception in WS1 where the Z2 is used for loading and unloading the papers.	This zone is available in all work stations
Z3	Productions zone. This zone is used to accomplish the task of the workstation (drawing).	This zone is available in all work stations
Z4	This zone is used for bypass conveyor. There are two functionalities; first, one is to decide which pallet will be transferred to zone 5 (from 3 to 5 or from 4 to 5). The other functionality is to hold pallet if zone 5 is occupied.	2, 3, 4, 5, 6, 8, 9, 10, 11 and 12
Z5	Exit zone where the pallet reach the end of work station	This zone is available in all work stations

4.3 Robots

The factory line uses SONY SCARA robots for production. Each robot is represented as an RTU in the line. The robot has 4 DOF(X, Y, Z and R_z). The robot uses custom-made end-effector for grapping the pen. In this manner, a custom-made jig holds three different pens allowing the robot to pick the needed one. See Figure 8. The robots are programed with specific tasks. Students do not need to change the robot programs. Nevertheless, the logic of the tasks execution is located in the dedicated RTU, which could be modified by students if needed. The RTU communicates with the robot using RS-232 serialcommunication.

Figure 8-a: Sony SCARA robot

Figure 8-b: pens holder

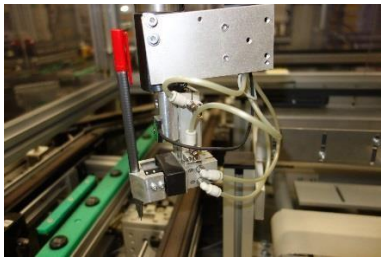


Figure 8-c: end-effector

4.4 RFID Readers

In order to identify the pallets in the line, The FASTory line is equipped with RFID reader (in each Z1) that can read the attached tags beneath the pallets. The reading mechanism uses the RS-232 serial communication by the conveyor RTU.

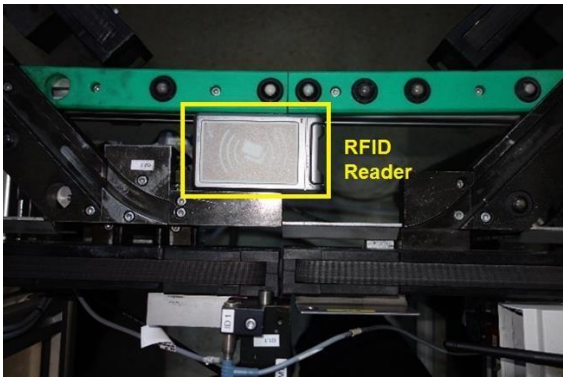


Figure 9-a: RFID reader

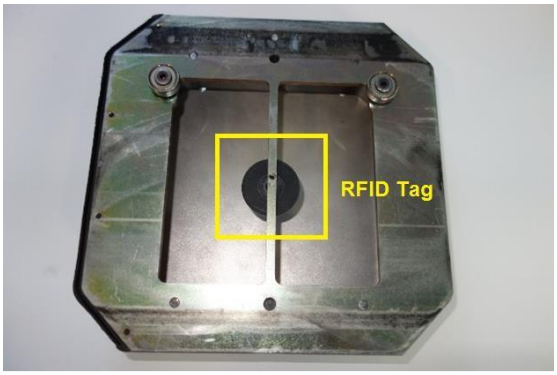


Figure 9-b: RFID tag

4.5 Sensors

As highlighted in 4.2, each zone in the conveyor has a presence magnetic proximity sensor to detect the presence of pallets. Figure 10 shows the sensors in Z1 and Z5.

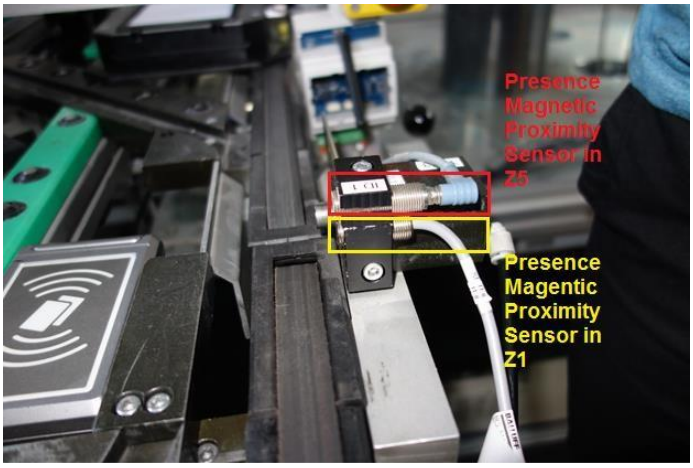


Figure 10: The magnetic proximity sensors in Z1/Z5

4.6 Actuators

In the FASTory line, there are two pneumatic actuators; stoppers and path selector. Figure 11 shows both the stoppers and the selectors. The stopper stops the pallet in each zone. Meanwhile, the path selector switches between main and bypass conveyors.

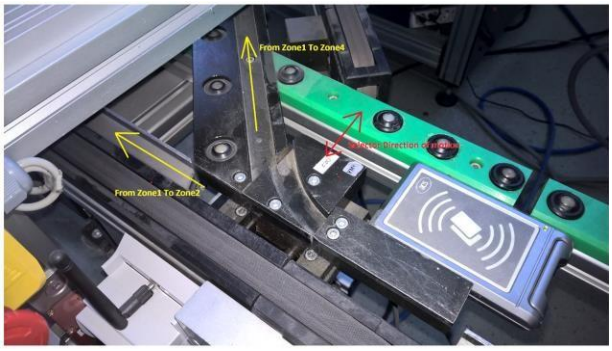


Figure 11-a: Path selector in the conveyor

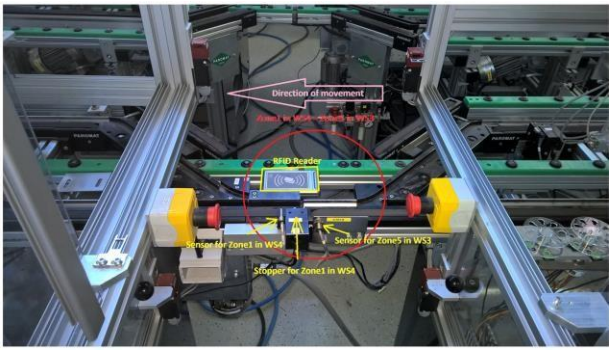
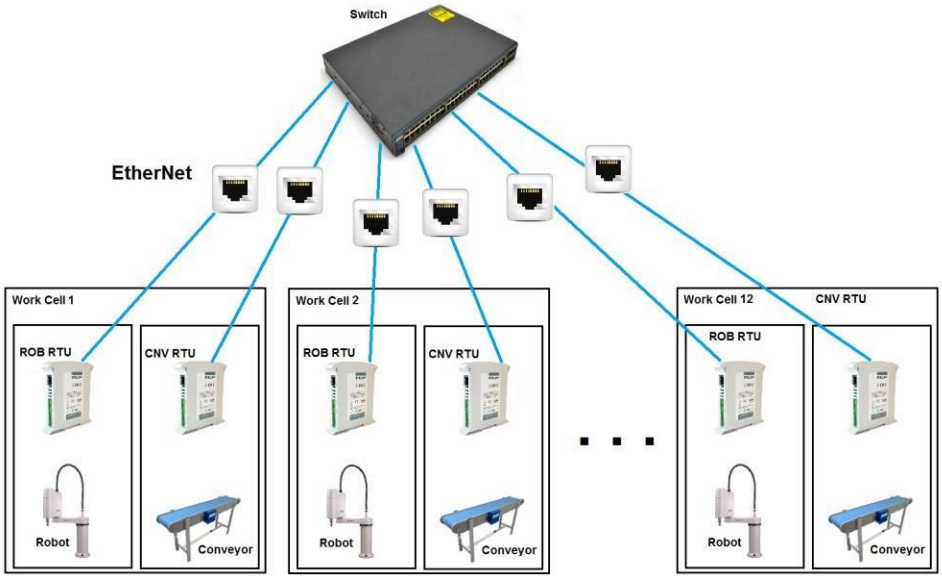


Figure 11-b: The stopper in Z1

5 FASTory RTUs

The FASTory is equipped with INICO S1000 Remote Terminal Units (RTUs). INICO S1000 is a programmable RTU device, which offers process control capabilities, as well as a Web-based Human-Machine Interface (HMI), and it supports REST and DPWS Web Services. Each Robot and Conveyor is controlled by an RTU. The RTUs are connected to the FASTory local network as appears in 12.



WS	RTU	IP Address
1	ROB	192.168.1.11
	CNV	192.168.1.12
2	ROB	192.168.2.1
	CNV	192.168.2.2
3	ROB	192.168.3.1
	CNV	192.168.3.2
4	ROB	192.168.4.1
	CNV	192.168.4.2
5	ROB	192.168.5.1
	CNV	192.168.5.2
6	ROB	192.168.6.1
	CNV	192.168.6.2
7	ROB	N/A
	CNV	192.168.7.2
8	ROB	192.168.8.1
	CNV	192.168.8.2
9	ROB	192.168.9.1
	CNV	192.168.9.2
10	ROB	192.168.10.1
	CNV	192.168.10.2
11	ROB	192.168.11.1
	CNV	192.168.11.2
12	ROB	192.168.12.1
	CNV	192.168.12.2

Figure 12: FASTory network configuration

6 How to run FASTory Line?

1. Connected to the FASTory line network
   IP address: 192.168.x.y (x and y must not be one of the defined ones for the RTUs)
   Subnet Mask: 255.255.0.0
   How to do it?
   - Windows : https://www.youtube.com/watch?v=F89Mb_u5UEQ
   - MAC: https://www.youtube.com/watch?v=h3s-We2YdQc
   - Linux: https://www.youtube.com/watch?v=Wpiw6HPJrr8
2. Turn the work cell ON by turning the main in the front panel switch clockwise. Then, press the push button in the front panel. At this moment, both the Robot and the conveyor RTUs are turned one, the Conveyor can be used but the robot needs to be turned ON by turning the switch in the side panel clockwise. See 13.

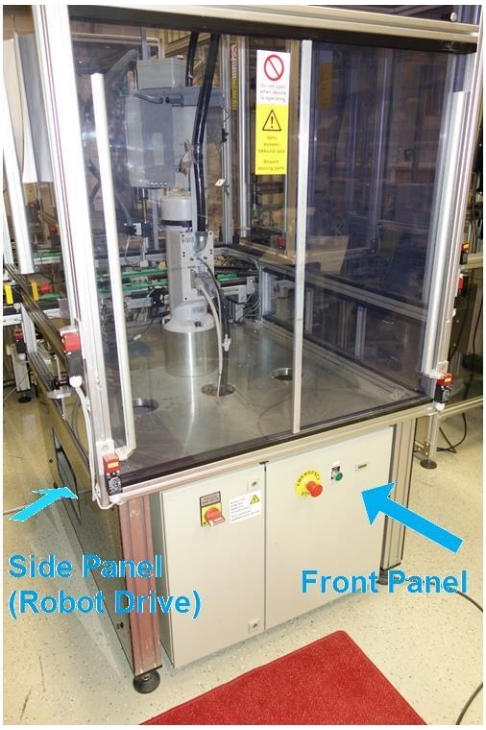


Figure 13-a: FASTory work cell

Figure 13-b: Front panel

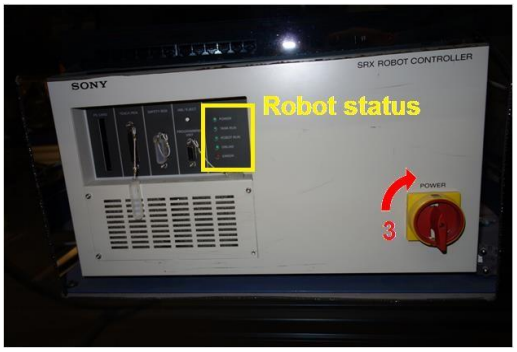


Figure 13-c: Side panel

Once the robot is turned on, the status LEDs will be changed. In the case of error, report the error to the FAST-Lab personnel so maintenance can take a place.

3. Go to the dedicated INICO page in the web browser using the defined IP addresses for the conveyor(192.168.z.x) and the robot(192.168.z.y) where x,y and z can be seen in Figure 12. After that, the user can change the mode (config/run) of the RTU. See Figure 14. The RTUs are already programed and it is not expected from students to change the program. You may though ensure that the controllers are in the RUN mode already. By introducing IP address in the web browser, you may in general case just check that the RTU is reachable.

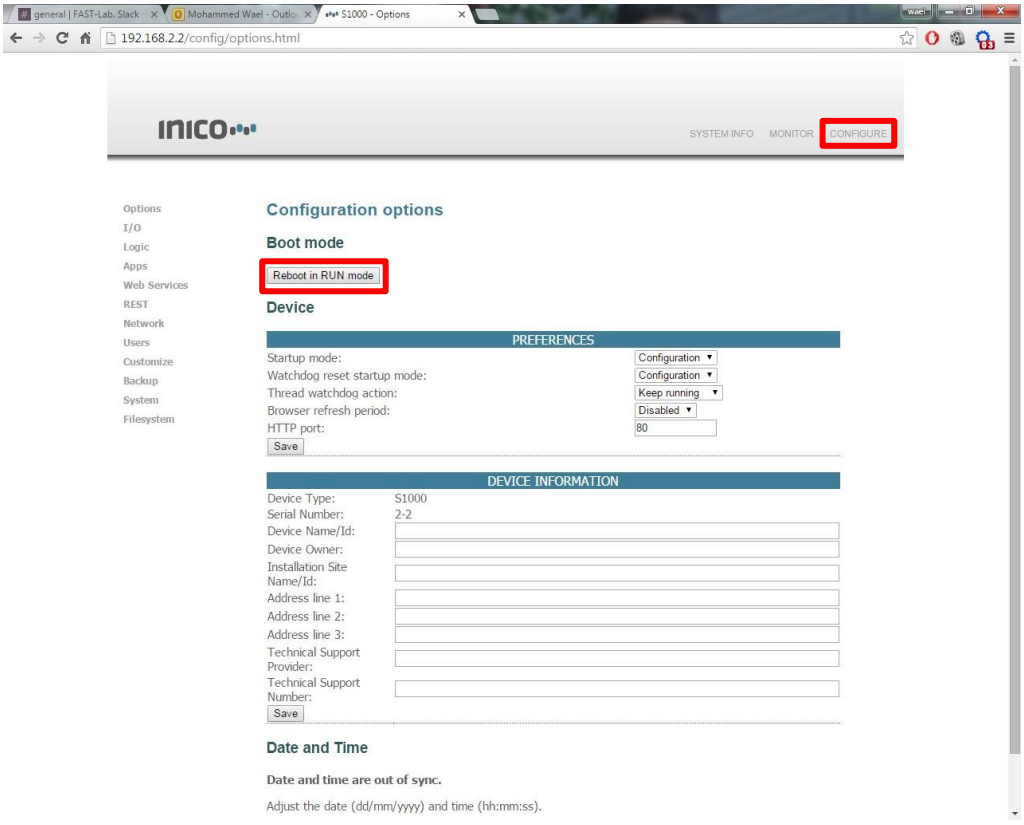


Figure 14: Configuration page in the RTU

4. Calibrate the robot by invoking the following service using, for example, Advanced REST Client:

```
POST: {HostIP}/rest/services/Calibrate
Headers: Content-Type: application/json
Body: {}
```

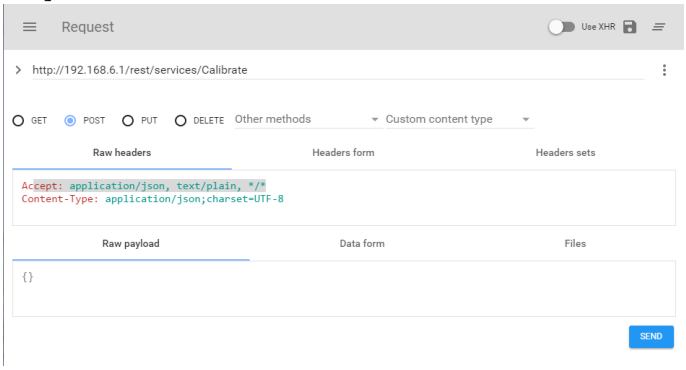


Figure 15: example of calibrating ROB6

After invoking the calibration service as shown in Figure 15, the user should wait until the robot finishes (around 3 min). During calibration, you may observe robot moving, unless robot was not already in the calibration position as the controller was turned on. In this manner, the calibration procedure is needed just once the robot drive is turned on and it is needed for each robot separately.

5. Now, the FASTory line is ready to be used.

7 FASTory Line APIs

The FASTory line is capable of supporting both RESTful and SOAP (DPWS) services. Nevertheless, as a current implementation, RESTful web services are the only services that is used for interacting with the line. In this context, with RESTful web services, the user is able to invoke services (operation /query) on the RTU level. Besides, the user might get a feedback of the line by subscribing the specific events in the RTU level. This section provides a guideline of the services in the FASTory line.

7.1 Services

In order to read the services description hosted by the RTUs, the user should request the following RESTful service:

```
GET: {hostIP}/rest/services
```

Then, using the web links, the user can read the services as children of the RTU services. The *self link* in each service is the one that the user can request. The http method (post, get) is defined by the class attribute; if the class is operation, then the user should use POST method, while query class represents the get method.

For operations, the user could use the following body if (s)he will wait for confirmation message about the finished operation:

```
{"destUrl":"http://example.com"}
```

In this manner, the RTU will send an empty POST request to the provided *destUrl* informing about the finishing of the operation. As an example, a user has an application, which invokes TransZone12 operation in workstation 6. The request should be like:

```
POST: http://192.168.6.2/rest/serives/TransZone12

Body:

{"destUrl": http://192.168.123.123:8080/events"}
```

If the request is correct, then the RTU will respond to this request with 202. After that, the user's application will receive a POST request on http://192.168.123.123:8080/events with empty body. Figure shows the sequence diagram of invoking operation in the FASTory line. There should be the Application having 8080 port opened with implemented branch `/events` and waiting for receiving the responses. After response arrives, it can be processed by the Application.

The branch `/events` can be implemented with different name by the developer's will, even with empty branch `/`.

All FASTory RTUs services can be found in Table 2

Figure 16: RESTful service description

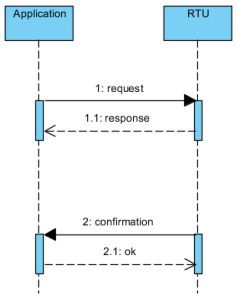


Figure 17: Sequence diagram of invoking services in the FASTory RTUs

Table 2: RESTful API for the hosted services in the FASTory line

RTU	Service Name	Description	Method	body	response
CNV	http://192.168.X.Y/rest/services/TransZone12	Move pallet from zone 1 to zone 2	POST	{"destUrl" : ""}	202 if accepted and 404
	http://192.168.X.Y/rest/services/TransZone23	Move pallet from zone 2 to zone 3	POST	{"destUrl" : ""}	

	http://192.168.X.Y/rest/services/TransZone35	Move pallet from zone 3 to zone 5	POST	{“destUrl” : “”}	if forbidden

	http://192.168.X.Y/rest/services/TransZone14	Move pallet from zone 1 to zone 4	POST	{“destUrl” : “”}	
	http://192.168.X.Y/rest/services/TransZone45	Move pallet from zone 4 to zone 5	POST	{“destUrl” : “”}	
	http://192.168.X.Y/rest/services/Z1	Get zone 1 state	POST	{}	PalletID if there is pallet, -1 if zone is empty
	http://192.168.X.Y/rest/services/Z2	Get zone 2 state	POST	{}	
	http://192.168.X.Y/rest/services/Z3	Get zone 3 state	POST	{}	
	http://192.168.X.Y/rest/services/Z4	Get zone 4 state	POST	{}	
	http://192.168.X.Y/rest/services/Z5	Get zone 5 state	POST	{}	
ROB	http://192.168.X.Y/rest/services/Draw1	Draw recipe 1	POST	{“destUrl” : “”}	202 if accepted and 404 if forbidden
	http://192.168.X.Y/rest/services/Draw2	Draw recipe 2	POST	{“destUrl” : “”}	
	http://192.168.X.Y/rest/services/Draw3	Draw recipe 3	POST	{“destUrl” : “”}	
	http://192.168.X.Y/rest/services/Draw4	Draw recipe 4	POST	{“destUrl” : “”}	
	http://192.168.X.Y/rest/services/Draw5	Draw recipe 5	POST	{“destUrl” : “”}	
	http://192.168.X.Y/rest/services/Draw6	Draw recipe 6	POST	{“destUrl” : “”}	
	http://192.168.X.Y/rest/services/Draw7	Draw recipe 7	POST	{“destUrl” : “”}	
	http://192.168.X.Y/rest/services/Draw8	Draw recipe 8	POST	{“destUrl” : “”}	
	http://192.168.X.Y/rest/services/Draw9	Draw recipe 9	POST	{“destUrl” : “”}	

7.2 Events

Another feature of the RTUs in the FASTory line is the event notification capability. Similarly, to the services, the user can read all possible notification in each RTU via the following request:

```
GET: {hostIP}/rest/events
```

As shown in FF, all notification events are listed as JSON. The user **subscribes** to an event as follows:

```
POST: {hostIP}/rest/events/{eventID}/notifs
```

```
Body: {“destUrl” : “http://example.com”}
```

Once the Application of the user subscribes to an event, the application will receive notification as POST request with a JSON body. As show below, the notification contains the event Id, the RTU Id and the payload. This payload contains the information about the event.

Figure 18: Figure 16: RESTful service description

```
{
        Id: "{EventID}",
        senderID: "{RTUID}",
        lastEmit: "",
        payload:
        {
        }
}
```

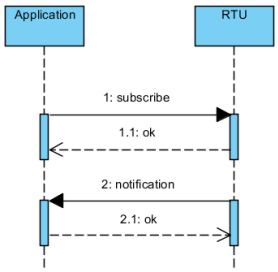


Figure 19: sequence diagram of event notification in the FASTory RTU

As an example, an application needs to subscribe to Z2_Changed event in CNV8.

```
POST: http://192.168.8.2/rest/events/Z2_Changed/notifs

Body:

{"destUrl" : "http://192.168.123.123:8080/events"}
```

Notice: the IP address and Port in the `destUrl` should be the same as your flask server's configuration.
Then once a pallet reach or leave Z2 in the CNV8, the application will receive a notification informing about the change of the zone status. All possible events in the FASTory line are listed in Table 3.
Make sure that the event is subscribed to at least once by (debug) sending a GET request to the same URL:

```
GET: http://192.168.8.2/rest/events/Z2_Changed/notifs
Response Body:
{
    "id": "notifs",
    "links": {
        "self": "/rest/events/Z2_Changed/notifs",
        "info": "/rest/events/Z2_Changed/notifs/info"
    },
    "class": "notifs",
    "children": {
        "urn:uuid:0fdfe8e6-3f46-4aeb-5503-0050c2899386": {
            "id": "urn:uuid:0fdfe8e6-3f46-4aeb-5503-0050c2899386",
            "links": {
                "self":
"/rest/events/Z2_Changed/notifs/urn:uuid:0fdfe8e6-3f46-4aeb-5503-
0050c2899386",
                "info":
"/rest/events/Z2_Changed/notifs/urn:uuid:0fdfe8e6-3f46-4aeb-5503-
0050c2899386/info"
            },
            "class": "eventNotif"
        }
    }
}
```

The number of `urn:uuid` in `children` tells how many messages will be fired from the RTU once the event `Z2_Changed` is triggered.
If subscription for each event is more than one, subscription deletion (unsubscribe) can be made by send DELETE request to the same URL.
If there is none, or one but being delepted after an event triggered without any readings on the registered server, then the registration was not correct.
Get the list of possible events by sending:

```
GET: http://192.168.8.2/rest/events
```

Table 3: all event that can be triggered in the FASTory line RTUs

Event name	Event description	Event body
Z1_Changed	Triggers once zone 1 state changes	

Z2_Changed	Triggers once zone 2 state changes	{"Id":"","senderId":"","lastEmit","","payload":{"PalletID":""}}
Z3_Changed	Triggers once zone 3 state changes	
Z4_Changed	Triggers once zone 4 state changes	
Z5_Changed	Triggers once zone 5 state changes	
PenChangeStarted	Triggers once the operation of changing pen is started	{"id": "PenChangeStarted", "senderID": "","payload": { "PenColor":""}}
PenChangeEnded	Triggers once the operation of changing pen is ended	{"id": "PenChangeStarted", "senderID": "","payload": { "PenColor":""}}
DrawStartExecution	Triggers once the robot begins drawing each recipe	{"id": "DrawStartExecution","senderID": "", "payload":{"Recipe" : "","PenColor":””} }
DrawEndExecution	Triggers once the robot finishes drawing each recipe	{"id": "DrawStartExecution","senderID": "", "payload":{"Recipe" : "","PenColor":””} }