

IoT@VET Training Module Index

GT-ARC, Germany

This document summarizes the 10 training modules for the IOT@VET project in the domain of **smart factories**. These modules are designed as an **interconnected set of modules**. They are designed such that the individual modules developed by different students or student groups in a class will collectively realize a “smart factory” IoT scenario, as depicted in Figure 1.

In our smart factory scenario, there is a warehouse, where different *parts* are delivered and stored. These parts have specific temperature and humidity requirements for long-term storage. Therefore, sensors are deployed in the warehouse to monitor the environment.

The production facilities in the smart factory include conveyor belts and robotic systems for production. It is also supported by automated systems to pick up and deliver the required parts from the warehouses into the production facilities. For simplification, each product in our scenario is a combination of certain parts, as shown in Figure 1.

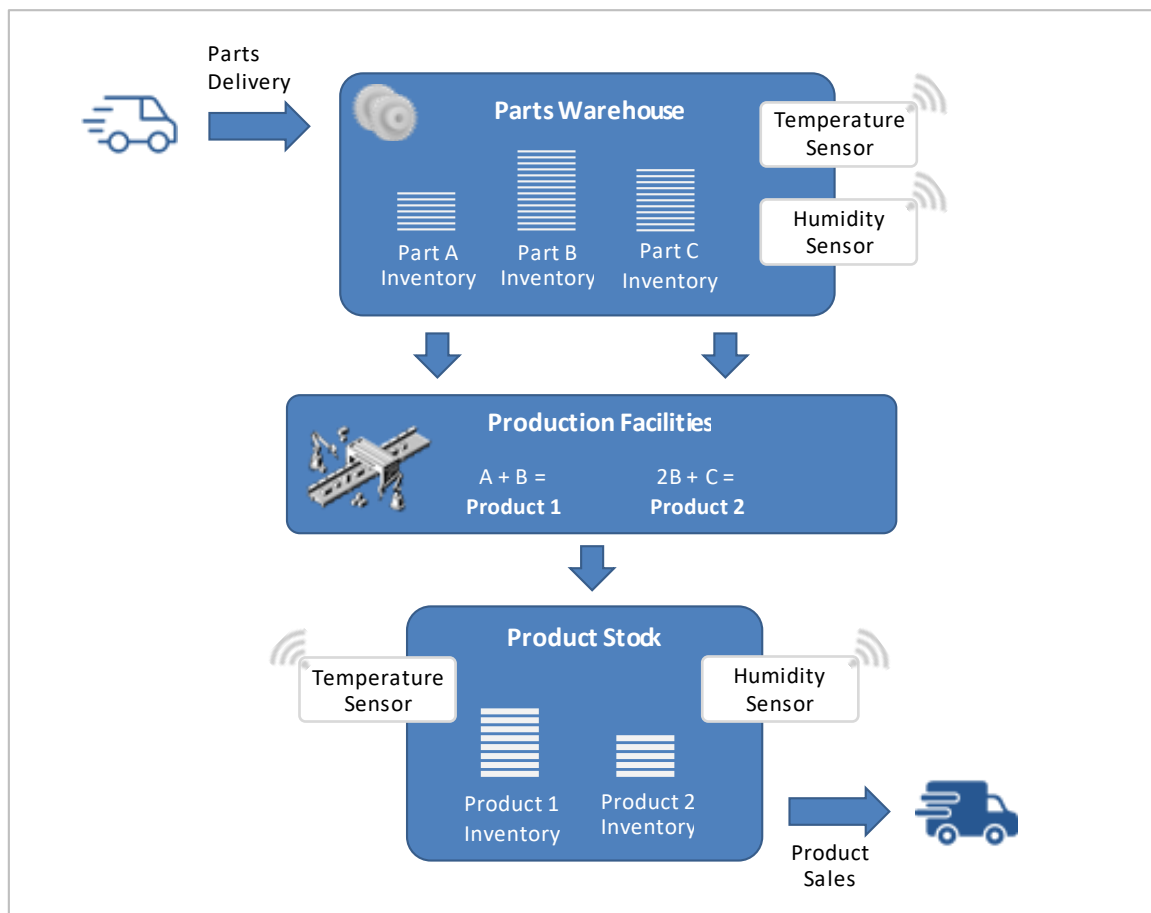


Figure 1. Smart factory scenario with inventory tracking and environment monitoring supported by IoT sensors

The product stock serves as a warehouse for finished products and facilitates their automated delivery based on sales orders. As in the parts warehouses, the product stock is also equipped with temperature and humidity sensors to ensure proper storage conditions for the products.

Finally, the smart factory has an automated inventory tracking system to monitor the amount of available parts and products in real time. Whenever there is a new delivery of certain parts, the production of a product, or sales and delivery of a product, the inventories are updated accordingly.

The first two training modules in this collection are common for all students and aim to set up the common development environment for the remaining lab sessions. For the remaining modules, it is suggested that the teachers/instructors assign the modules to different students (or groups), such that the overall scenario described above is realized as the result of **teamwork** with the inclusion of all students.

Module Descriptions

- 1. Docker as a Virtualization Platform for Cloud and Edge:** This module aims to introduce one of the virtualization concepts, *containers*, as an enabler of cloud and edge computing. It introduces *Docker* as a container technology and provides hands-on experience on the creation and management of containerized applications in the Docker environment, focusing on Node-RED as an example application running in Docker.
 - Getting started with Docker
 - Installing an application as a Docker container
 - Running & configuring a container/app from an existing local image
- 2. Node-RED as an IoT Application Development Tool:** This module aims to provide basic understanding of Node-RED as a *visual flow programming* tool that can support IoT application design and development.
 - Installing and running Node-RED (as a docker container)
 - “Nodes” and “Flows” in Node-RED
 - Creating your first flow / application in Node-RED
- 3. Implementing the Parts Warehouse Sensors:** This module implements simulated temperature and humidity sensors to keep track of the environmental conditions in the parts warehouse.
 - Using the *inject* and *function* nodes in Node-RED to create virtual sensors
 - Generating random numbers to emulate sensor data
 - Using an *MQTT-out* node to send the sensor data to a remote MQTT Broker
- 4. Implementing the Product Stock Sensors:** Similar to the previous module, this module implements a pair of temperature and humidity sensors, but instead of creating them from scratch, we will see how to create a copy of the warehouse sensors and edit them easily for the product stock.
 - Exporting and importing flows in Node-RED
 - Editing the properties of existing flows and nodes

-
- 5. Monitoring and visualization of sensor data:** This module implements a visual dashboard to monitor the sensor data from the parts warehouse and the product stock.
- Reading sensor data by subscribing to an MQTT topic
 - Plotting sensor data by using *node-red-dashboard* nodes
 - Understanding and adapting the user interface (UI) layout in Node-RED
- 6. Implementing an Inventory Service API – GET method:** This module introduces how to implement an inventory database service in Node-RED and to develop a Web API for reading the current inventory/stock amount for any given part or product (e.g., “retrieve the number of part A units available in the warehouse”).
- Using “flow variables” in Node-RED to store values persistently across different nodes
 - Learning to implement a web service with a REST API
 - Implementing a GET method for external software modules to read the inventory/stock states
- 7. Implementing an Inventory Service API – POST method:** This module continues with the implementation of an inventory database service by developing a Web API for updating the inventory/stock for parts and products (e.g., “add two more units of Product 1 to the stock”).
- Implementing a POST method for external software modules to update the inventory/stock states
 - Observing the context data and variable values to test the application functionality
- 8. Simulation of Parts Delivery and Product Sales:** This module implements a web-based UI to simulate the delivery of parts and sale of products. It utilizes the inventory/stock API of the web service from the last two modules.
- Implementing a client that uses the POST method of the web service API
 - Using forms in Node-RED to retrieve and process user input
- 9. Simulation of Production:** This module implements a web-based UI to simulate the production in the smart factory, resulting in an increase in product stock values and decrease in the inventory of the associated parts. It also relies on the inventory/stock web service.
- Implementing a client that uses a combination of the GET and POST methods of the web service API
 - Checking for sufficient amount of the required parts for production
- 10. Dashboard for Monitoring Inventory/Stock status:** This module implements a visual dashboard to monitor the number of available parts in the warehouse and products in the stock, utilizing the inventory/stock API of the web service from the last two modules.
- Creating web UI with multiple data series in a single plot
 - Using bar charts for data visualization