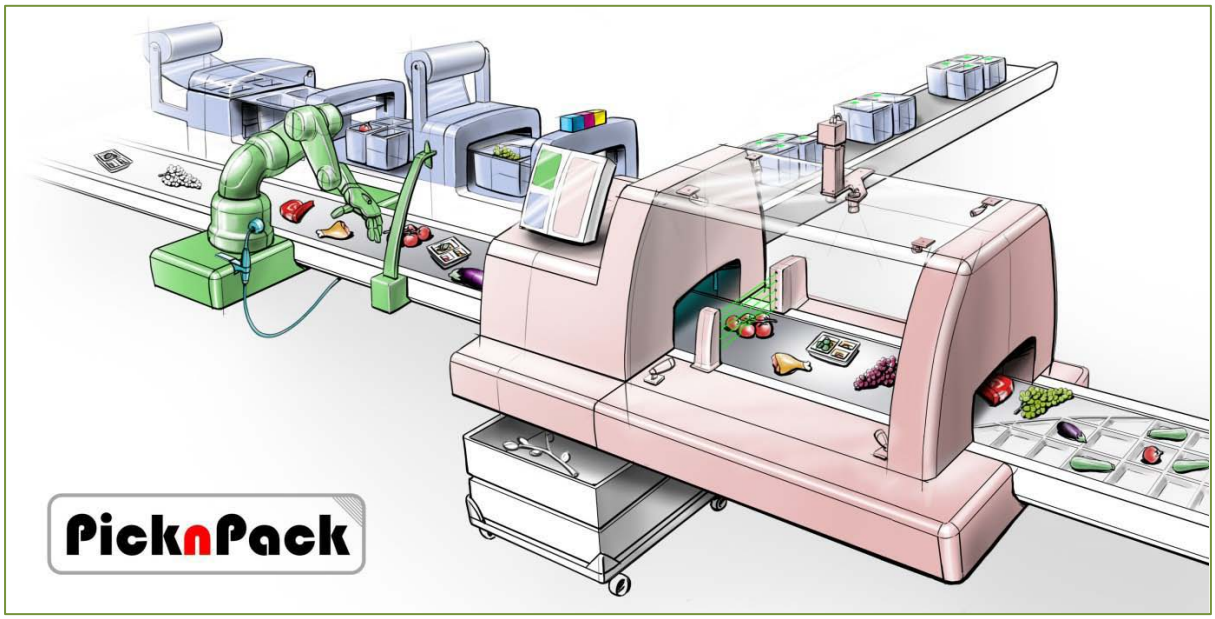
D3.8 – Report on Traceability Demonstration System

Zhipeng Wu (UM), Zhaozong Meng (UM) and John Gray (UM)
8/31/2015



Flexible robotic systems for automated adaptive packaging of fresh and processed food products

The research leading to these results has received funding from the European Union Seventh Framework Programme under grant agreement n° 311987.

Dissemination level		
PU	Public	X
PR	Restricted to other programme participants (including the EC Services)	
RE	Restricted to a group specified by the consortium (including the EC Services)	
CO	Confidential, only for members of the consortium (including the EC Services)	

Table of Contents

1. Introduction

This document outlines the progress of the traceability system design and implementation for WP3 on optimised traceability system, relating to the deliverable of:

- D3.8: Report on traceability demonstration system (M36)

The associated milestone is:

- MS7: Optimised and complete integrated traceability system (M36)

The RFID integrated traceability system described in D3.5 and D3.6 has implemented the software integration techniques such as ZeroMQ communication, JSON data format, and state machine. This report focuses on the traceability demonstration system.

The RFID system design, database design, RFID system connection with database, software integration technologies which have been implemented are discussed. The RFID system collects useful information and share the information with other modules in the line. In addition to RFID system, a camera is integrated to the system to assist the information traceability. Implemented functions are tested in laboratory, and on-line test with other modules depends on the progress of the individual modules, which will be carried out in the next step.

Further details on hardware system, database design, software integration, and functions of the traceability application are provided in the following sections.

2. Hardware System

Based on the regulations and requirements, this section gives key hardware designs and implementations of the traceability system: system architecture design, hardware modules, antenna placing in production line, and camera for barcode information tracing.

2.1 Hardware System Architecture

The system architecture is designed as shown in Figure 1. The whole system is working within a local area network. The RFID readers with antennas located in the production line can be managed by the traceability application, and information is collected and processed by the application. Product information with tag IDs are recoded in the database which can be accessed by the traceability application and the handheld reader application.

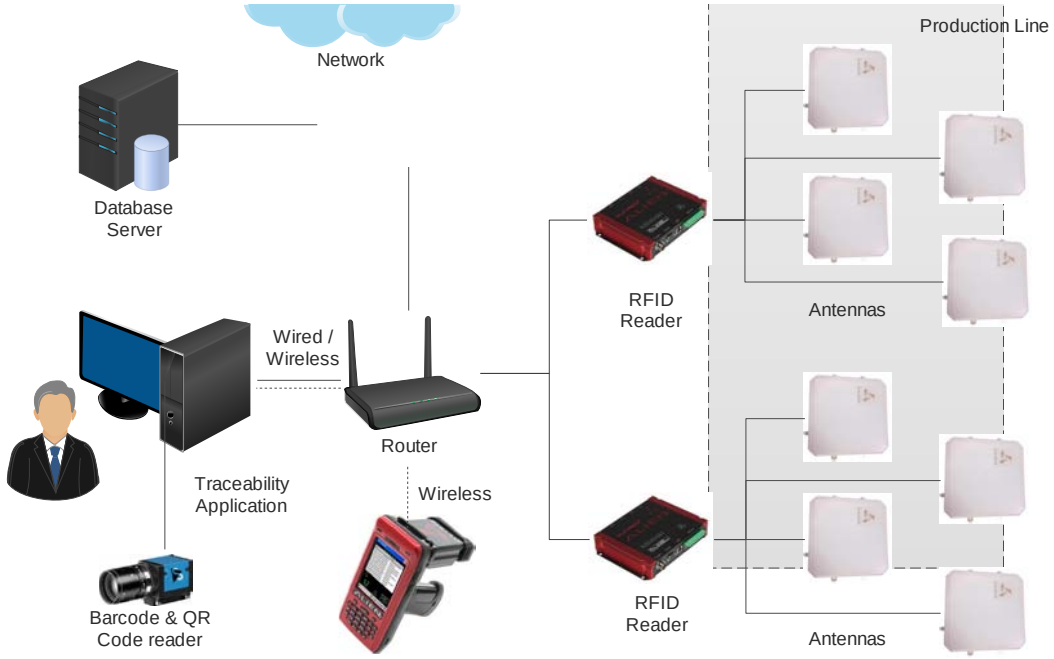


Figure 1 Hardware System

2.2 Hardware Modules

List of hardware:

(1) PC with traceability software
(2) Fixed RFID reader
(3) Handheld RFID reader
(4) RFID antennas
(5) RFID applicator
(6) Barcode and QR code reader (camera)
(7) Network infrastructure and devices, such as WiFi router

In the hardware modules, the RFID label applicator and the barcode reader are new devices, which are as shown in Figure 2 and Figure 3

Figure 2 RFID Applicator

Figure 3 Barcode and QR Code Reader

2.3 RFID Label Placing

As shown in Figure 4, the RFID antenna is installed in the production line to read the tags stick on the product objects. The antenna should be installed in a location where the radiation can cover all possible locations that products may appear when they go through.

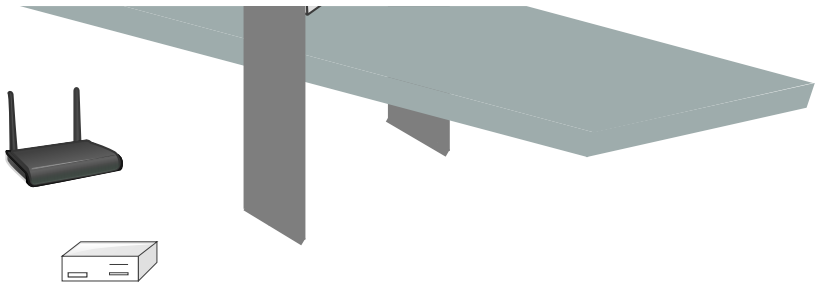


Figure 4 Antenna Placing

By placing the antennas in the appropriate location, all passing objects with tags can be recorded without human assistance.

2.4 Camera for Barcode and QR Code Tracing

Since the source goods batch, packaging job, and output containers all have their barcode, either obtained from previous trading partners or generated by the system, the barcode can be used to trace the information with a camera. In the hardware modules, the camera is a new integrated one.

The model of the camera is DFK 23UP1300 colour video CMOS sensor, resolution 1280x1024 (1.3MP), speed up to 95 frames per second. Model of Lens is DF6HA-1B. The camera is connected to computer with a USB cable.

The software module for camera video capture and barcode reading is developed and embedded to the traceability system. The barcode reading module supports 14 commonly used barcode format and QR code. A video capturing thread is running in background to detect barcode. The records in database linked to the detected barcode will be displayed in the search & query interface. The logic of the barcode reading and query is as shown in Figure 5.

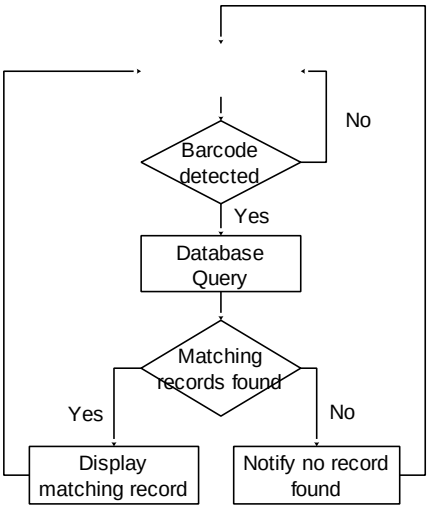


Figure 5 Logic of Barcode Information Traceability

3. Process Modelling and Database Design

The main task of the RFID traceability system is to information of interest and provide interface to share and query the information. The data structure of the information to collect links to the operation process. This section illustrates the process modelling and data structure design.

3.1 Process Modelling

The EU's General Food Law entered into force in 2002 and makes traceability compulsory for all food and feed businesses [1]. The traceability needs to cover "one step backward and one step forward" in the food chain, which is also called "one-up-one-down system".

The packer needs to collect data from previous and next trading partners, and also shares data with previous and next trading partners. The modelling of the food packaging process is as shown in Figure 6. The process can be briefly described with three steps:

(1) The source material from supplier is registered and then put in input containers, and all information is recorded and linked with RFIDs on input containers.
(2) In packaging line, source material is put into small packages, and then put into output containers. The RFID of small packages are linked with both input containers and output containers in the database.
(3) Output contains are set to be logistical units. Information of delivery and retailer market information are recorded.

Therefore, with the RFID and some unique IDs generated with the system, the source material information, product information, and logistic and market information are all

linked together. The data structure of the database is designed according to this relationship.

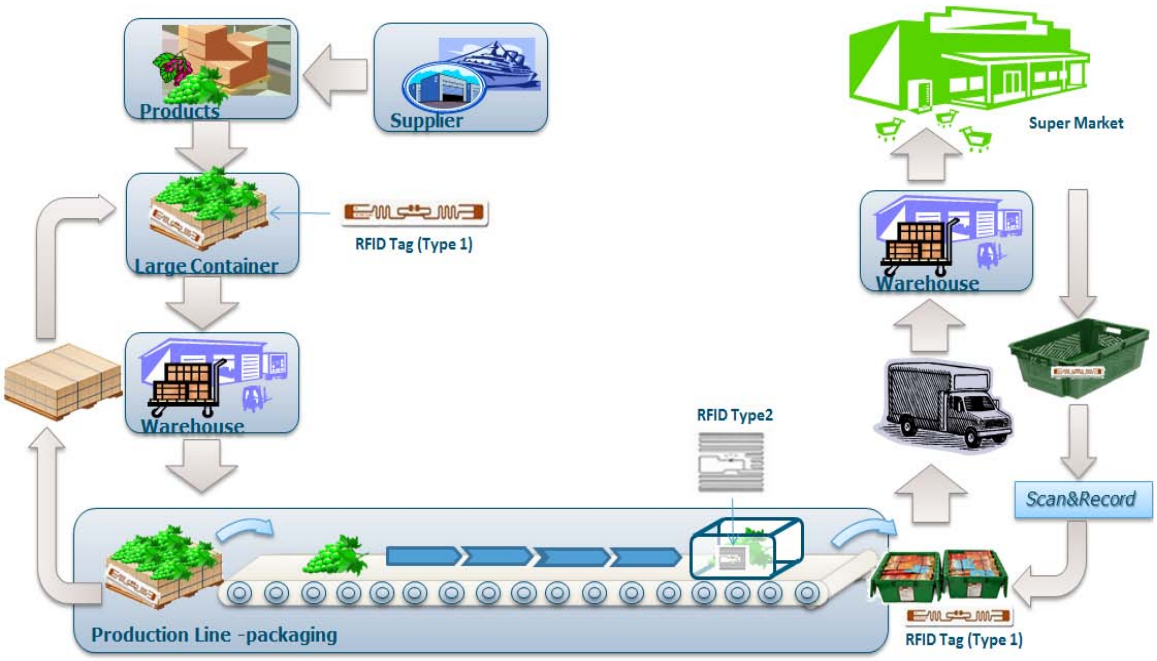


Figure 6 Food Packaging Process

3.2 Database Design

The data table structure is designed according to the process in section 3.1, and the database tables are listed as follows. The underling items are the foreign keys of the tables, which are linked to other tables.

- [Supplier]: Supplier ID/Location/Type/Contract period/Contact info/additional information…
- [Goods_batch]: Goods_Batch ID/Supplier ID/Content(e.g. grape)/ Quantity(volume)/ Arrived Time/Temperature Req./Humidity Req./Processed Before/Comments…..
- [Internal_container]: Internal_Container_ID/RFID ID/Type/InUse/Invalid
- [Internal_container_fill]: Internal_Container_Fill ID/Internal Container ID/ Goods Batch ID/Operator/Packaging Before/filled Time/empty Time
- [Plant_warehouse]: Warehouse ID/Location/Capacity/Remaining Capacity/ …
- [Plant_storage]: Plant_Storage ID/*Internal Container Fill ID*/Time enter/ Time leave
- [Food_packaging_job]: Packaging ID/ Small_Package_ID/ Product Name/Production line ID/Start Time/ Content Type/ *Internal container Fill ID*/Best -before Date/Operator/Process1 ID/ Process … ID/Process N ID/Finished Time/ *External_Container_Fill_ID*/…
- [Small_package]: Small_Package_ID/RFID ID/Registered Time/Type/…

- [Process_N]: Process_N ID/Time/Operator/weight (or other information)
- [External_container]: Container ID/RFID ID/Container Type/Dimensions/capacity/InUse/Invalid
- [External_container_fill]: External_Container_Fill_ID/Container ID/Content/Fill Time/Empty Time/ *Delivery ID*
- [Delivery]: Delivery ID/Vehicle ID/Operator/Departure Time/Arrive Time/ Production_Plant ID/Supermarket warehouse ID/Delivery Before
- [Super_market_warehouse]: Supermarket_Warehouse ID/ Location/ Capacity/ Remaining/….
- [Supermarket_warehouse_storage]: Supermarket_Warehouse_Storage ID/ External_Container_Fill ID/ Time enter/ Time leave/ Supermarket_Warehouse ID
- [Selling]: Selling ID/ External_Container_Fill ID/ Start Time/ End Time
- [External_container_return]: Return ID/External Container Fill ID/Send Time/Receive Time/Send Supermarket ID/Receive Plant ID
- [Supermarket]: Supermarket ID/ Location/ Other Information
- [Production_plant]: Production_Plant ID/ Location/ Other Information

The database is implemented with Microsoft SQL server. A software user interface is developed with Microsoft Visual C# 2010. The software application provides interface for users to manually input information in the database, collects RFID tags from RFID reader, and saves the data in database. It also provides interface for users to query product information from the database. The traceability software is illustrated in section 4.

4. Software Development and Software Integration

The traceability application is the intermediary between users, devices, and the database system. It provides user interfaces, data acquisition interfaces, and interface for data interaction with peer modules. This section briefly introduces the traceability application.

4.1 Traceability Application Development

4.1.1 Functionality

The function of the RFID tracking system is to record tag ID at different procedures of the production line process, and then provide all related information with the tag IDs. The processes in the information tracking with RFID are summarised as follows:

- New material subdivision
  With the RFID detected containers, the source material can be put into containers before packaging.
- Packaging

After providing the product information in 'Batch Setting' and 'Package Setting', user can start the system for packaging recording to create records in database.

- Logistic unit
  User can select detected available containers and register selected containers as a logistic unit.
- Delivery
  User can select a customer and sending place to create a database record of dispatching a logistic unit.
- Scan and Query
  With unique RFID tag IDs, all related information stored in database can be retrieved with the traceability application or a handheld reader.

In all the above steps, the RFID tag ID is unique information to associate the different process and track the objects automatically in the production line.

4.1.2 Flow Chart

With RFID devices implemented in the traceability system, the production line process is then assisted with the RFID modules and product information is recorded in database automatically. The process model is designed as shown in Figure 7. The components with RFID icons are the processes enhanced with RFID tracing, and those without RFID icons are operated by human only.

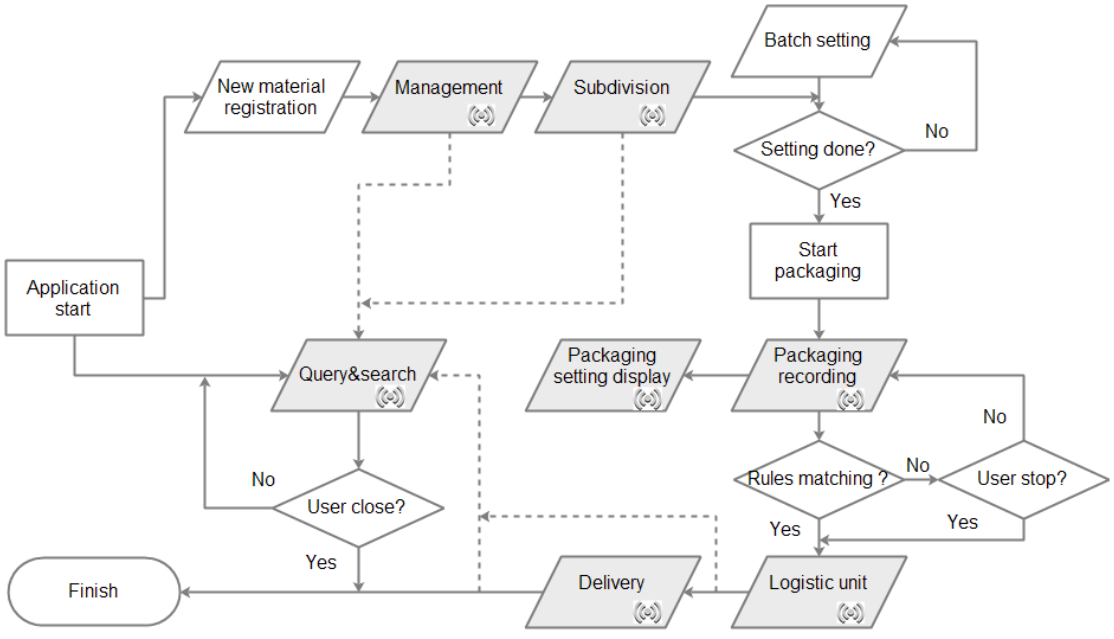


Figure 7 Flow Chart of the RFID Traceability Application

When new material for packing arrives, users need to manually register the materials with the system. The containers are also registered with RFID tags in the management process. Then, after the batch setting, the packing job can be started by clicking the 'Start' button.

The RFID detected available container can be registered as logistic units by clicking 'Register' button. Users can then select a valid customer and a sending place to create a record of dispatching a logistic unit by clicking 'Send' in delivery window.

All through the process, user can monitor the information by query & search function. The records of registered containers in management, containers for subdivision, packaging jobs, logistic units, and sent delivery units can be searched and checked by object IDs or tag IDs.

4.2 Software Integration

The traceability system is not a separate module. It needs to listen to the line controller and speaks to the peer modules for data communication. All devices need to follow the same protocols and standards for interoperable interaction. The standards to follow are Life Cycle State Machine (LCSM), ZeroMQ communication [2-4], Javascript Object Notation (JSON) [5, 6] data format, and spotlight protocol [7].

4.2.1 Finite State Machine

To integrate with the line, a finite state machine is designed. The design of the FSM [8] is based on its events and activities of the traceability application. For the traceability system of current version, it is separated into five states: Configure, Ready/Waiting, Running, Pause, and Interrupt:

- Configure: the system initiates and configures hardware devices, such as ZMQ connection, RFID reader initiation and configuration, packaging setting, job batch setting, input/output container registration, etc.
- Ready/Waiting: waiting when hardware devices and settings are ready
- Running: working
- Pause: stopped because of container empty, needs user to restart
- Interrupt: stopped because container removed, auto-start when container is found

The events in traceability system can be classified into four types:

- Device and setting state event
- Line command event
- User UI command event (start, pause, stop)
- RFID detected container event (full, empty, lost, or found)

The states transfer of the state machine is triggered by these events. In the state machine, the states are set with transitions "e[g]/a", where

- e=event
- [g] = guard condition
- a=action

Diagram of the state machine is as shown in Figure 8.

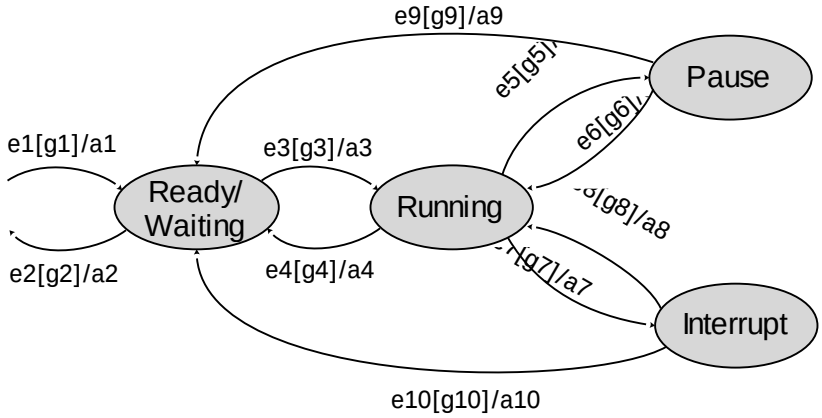


Figure 8 FSM of the Traceability System

The events are monitored by Window UI, and TCP socket from RFID reader, and ZMQ message from the line. In the Microsoft Visual C# environment, since the application has already been complicated, the handling of multi-thread communication is important for the performance of the traceability application.

4.2.2 ZMQ Interfaces for Inter-module Communication

The ZMQ based message interface for inter-module communication is as shown in Figure 8.

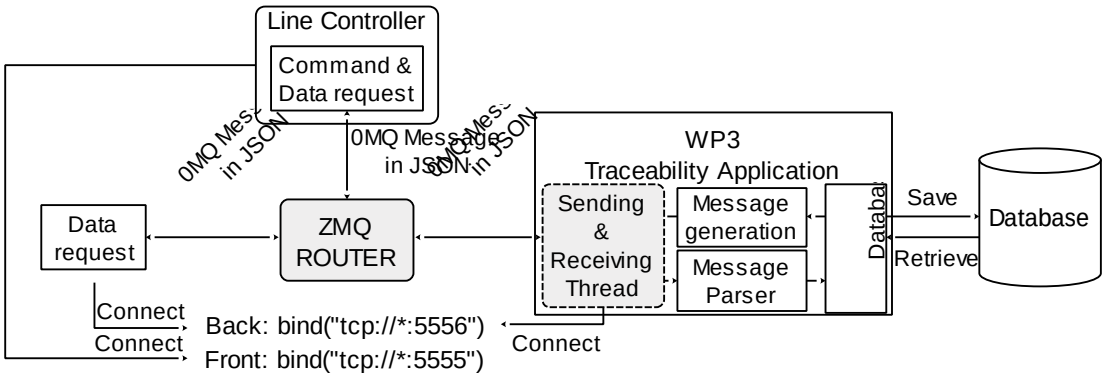


Figure 9 ZMQ Interface for Data Request

As shown in Figure 9, the traceability application connects to a TCP ports as a ZMQ DEALER, and modules requesting data are also connect to a ZMQ ROUTER. The modules send ZMQ message in JSON to the traceability application. The traceability application receives the message, parses the JSON message, executes the request, wrap the data in JSON, and then reply the message to the requester. On the other hand, the traceability system can also request data from the other modules in the line. The line controller works as a ZMQ Client.

In the data interface, the ZMQ ROUTER plays a very important role. It forwards module X's request to the traceability application and the forwards traceability application's reply back

to module X. It also forwards traceability applications data request to module X and forward back module X's reply back. In addition, it forwards traceability application's broadcasting information to all connected modules. Figure 10 illustrates the inter-module communication between traceability application and other modules.

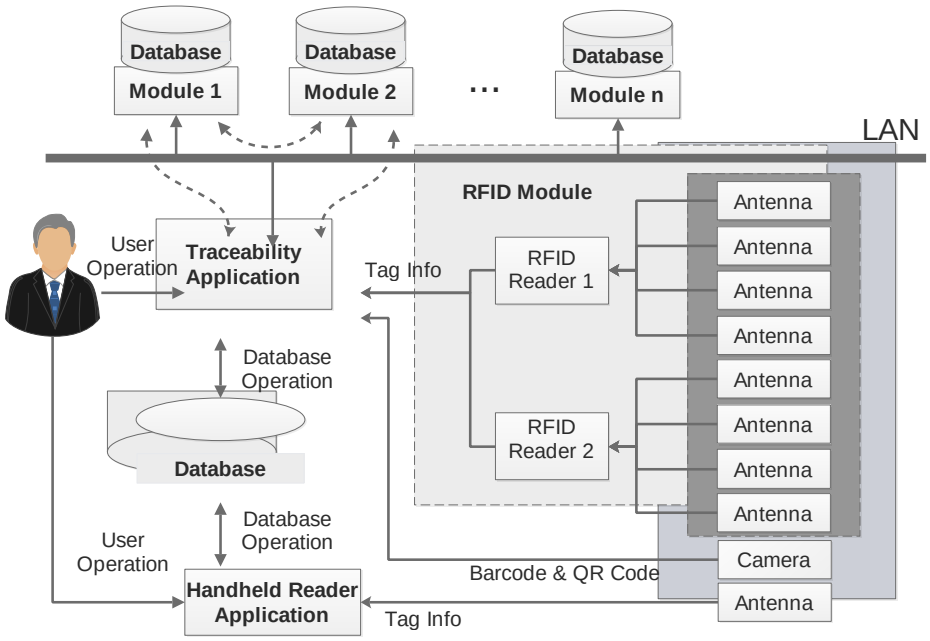


Figure 10 Data Flow of the Traceability System

4.2.3 Zyre Interfaces for Inter-module Communication

In addition to ZMQ interface, the Zyre interface is also implemented and tested, which will be integrated into the traceability application. Zyre is an open and standard way to connect applications without the cost of complicated technologies. A cheap, simple, universal messaging system that spoke the most widely used, and best understood protocol: plain old HTTP [9].

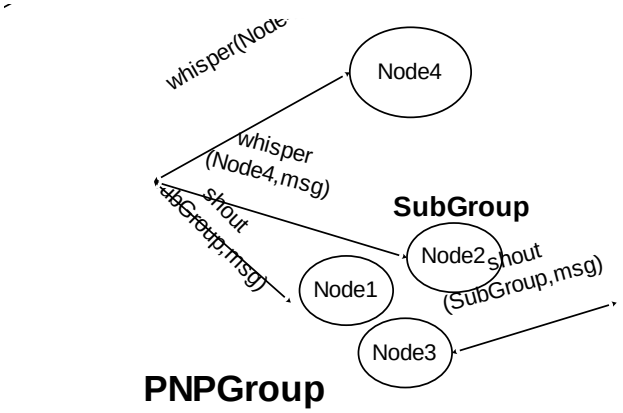


Figure 11 Topology of Zyre Connection

Different from ZMQ connection, the Zyre connection is more convenient to use. The main functions of Zyre are whisper and shout. Whisper is used for a node to talk to a specific node in the network, and shout is for a node to broadcast messages to all nodes in a group. The topology of the devices in Zyre connection is as shown in Figure 11.

The Zyre in Microsoft Visual C# is learned from work package 2 by visiting on 17 September 2015. The dynamic link library (DLL) file wrapping the functions [10] are used in our applications. This will be integrated to traceability application for inter-module communication with the modules in the packaging line. The testing of Zyre can be found in section 5.2.

5. Work Up to Date

This section summarise the functions implemented so far, including the software development and integration, and provides some test results.

5.1 Functions Implemented

The main functions implemented so far are summarised as follows:

- RFID hardware in place
- Traceability application
- Handheld RFID query application
- Database connection of both traceability application and handheld reader query application
- Inter-module communication interface with ZeroMQ
- JSON message generation and parsing for inter-module communication

The traceability application and handheld RFID reader application can fulfil the tasks of information traceability. The ZeroMQ interface can receive messages from line and other modules and also send messages to other modules for data exchange. JSON wrapper can generate JSON message which can be understood by other modules, and JSON parser can understand messages from other modules.

5.2 Test and Results

This section gives some selected test results of the traceability application development and integration technologies.

Figure 12 gives the main interface of the traceability application. Top left are the buttons to launch the interface of all functions of the software, such as new material registration, subdivision, packaging setting, logistic units setting, delivery, management, batch setting, scan&Query, RFID reader setting, and camera setting. On the bottom are events messages

and communication messages. In the middle are Barcode and QR code generated for packaging products. On the right is the packaging process information.

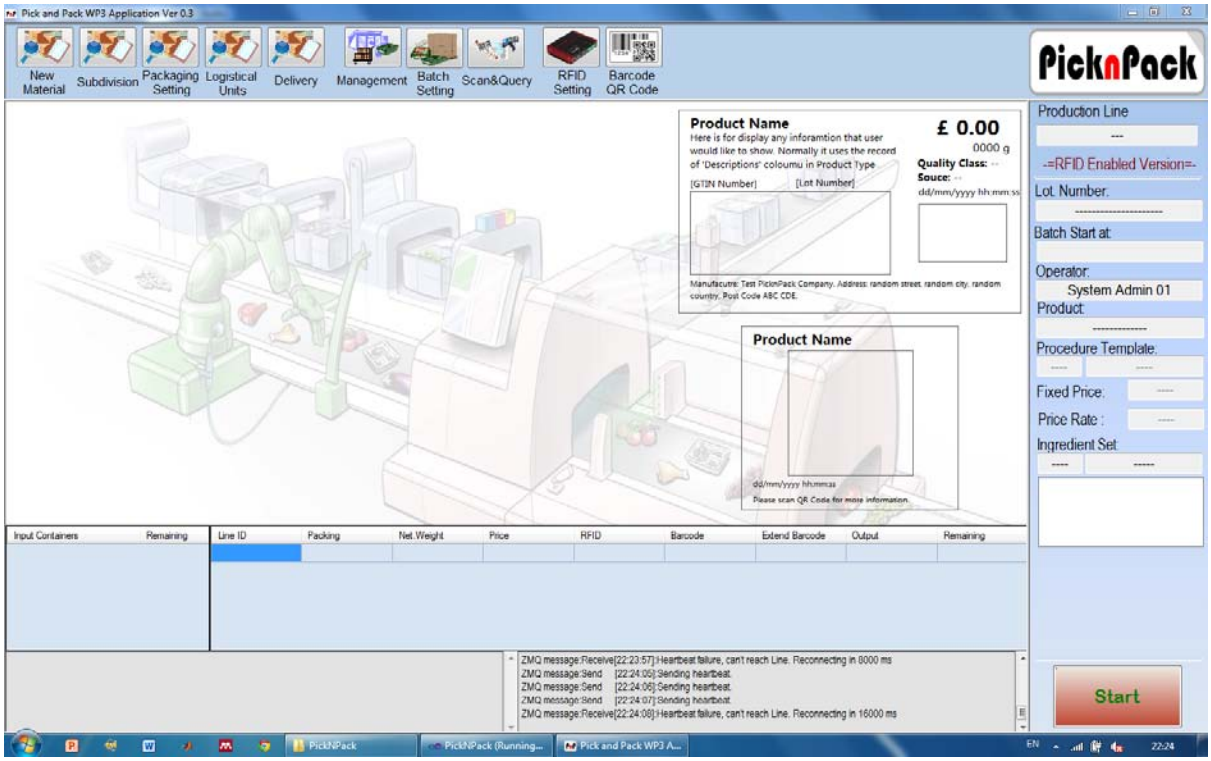


Figure 12 Main Interface of Traceability Application

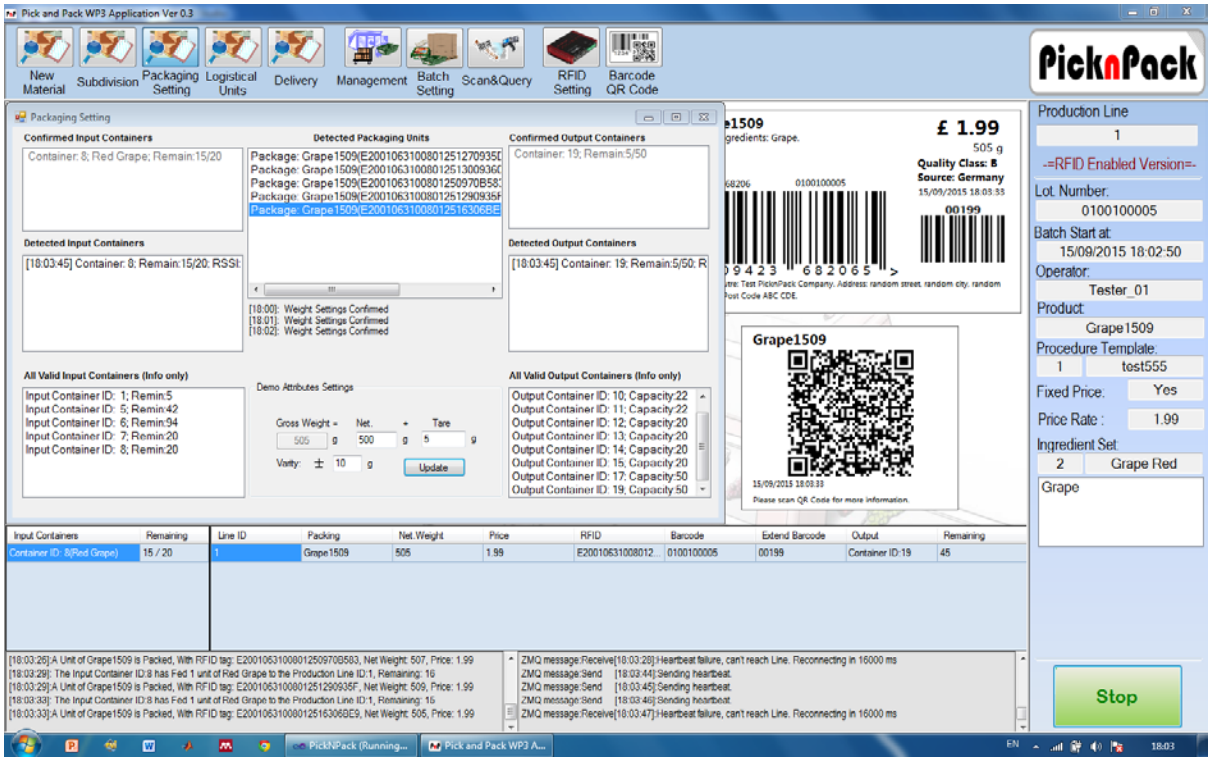


Figure 13 Traceability Application in Operation

When the traceability system is running, the available input container and output container are detected, and the small packages pass the packaging are is recorded. Therefore, the RFID on small packages are linked to input containers and output containers, which are also linked to source material information, and logistic and retailer information. The system in operation is as shown in Figure 13.

Figure 14 gives the Search & Query page, where user can search the required information with RFID, Barcode, and category of objects such as new material, input container, small package, output container, logistic units, etc. User can also click RFID tags detected on the bottom to query the information linked to it. In addition, user can launch the camera to query the information with Barcode and QR Code, as shown in Figure 15. All related information in data base can be displayed on the user interface.

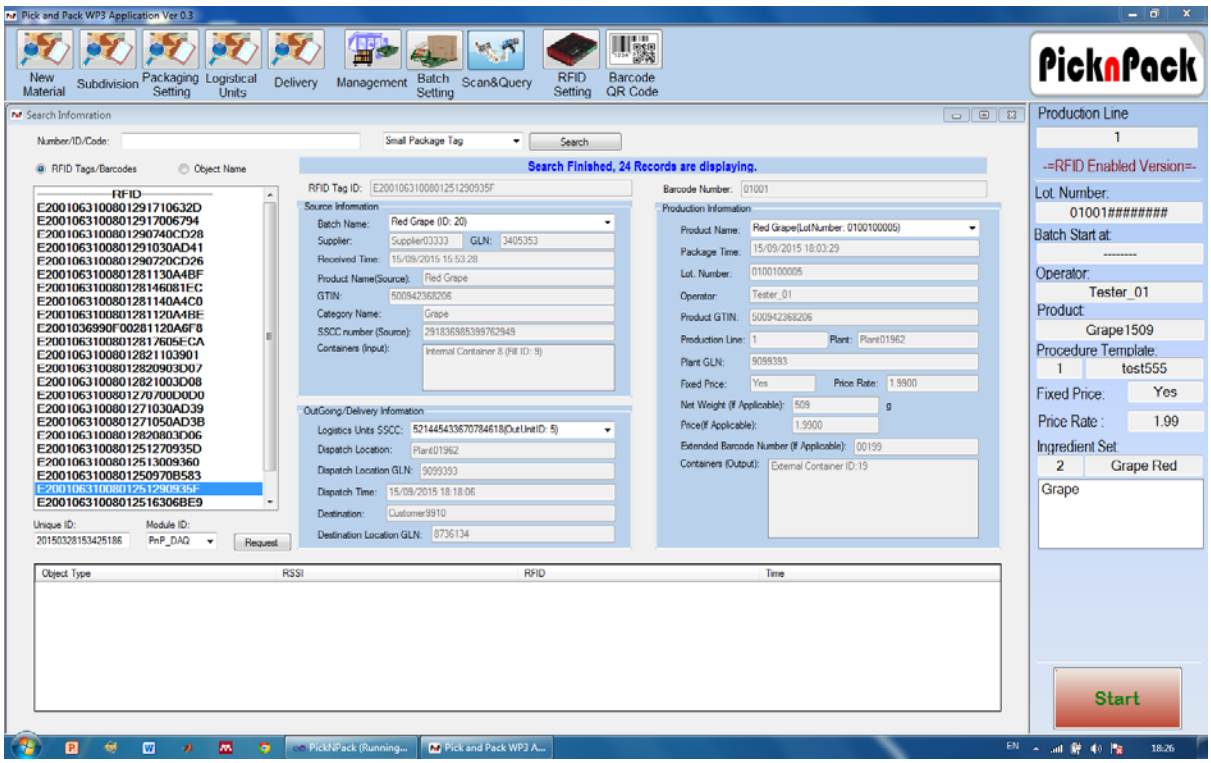


Figure 14 Query Interface

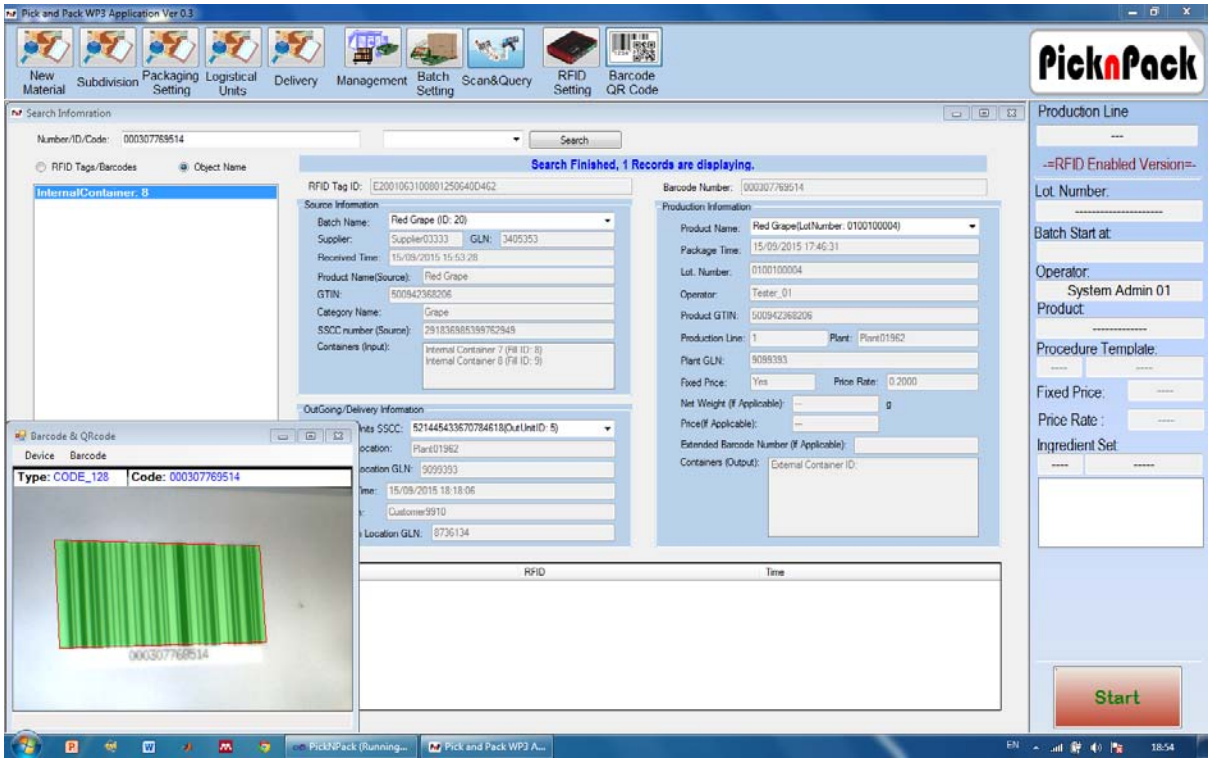


Figure 15 Barcode Decoding with USB Camera

The traceability system can share the information in database. That means other modules can request data from RFID traceability system as shown in Figure 16. Figure 17 shows the interface that other module receives data requested from the traceability application.

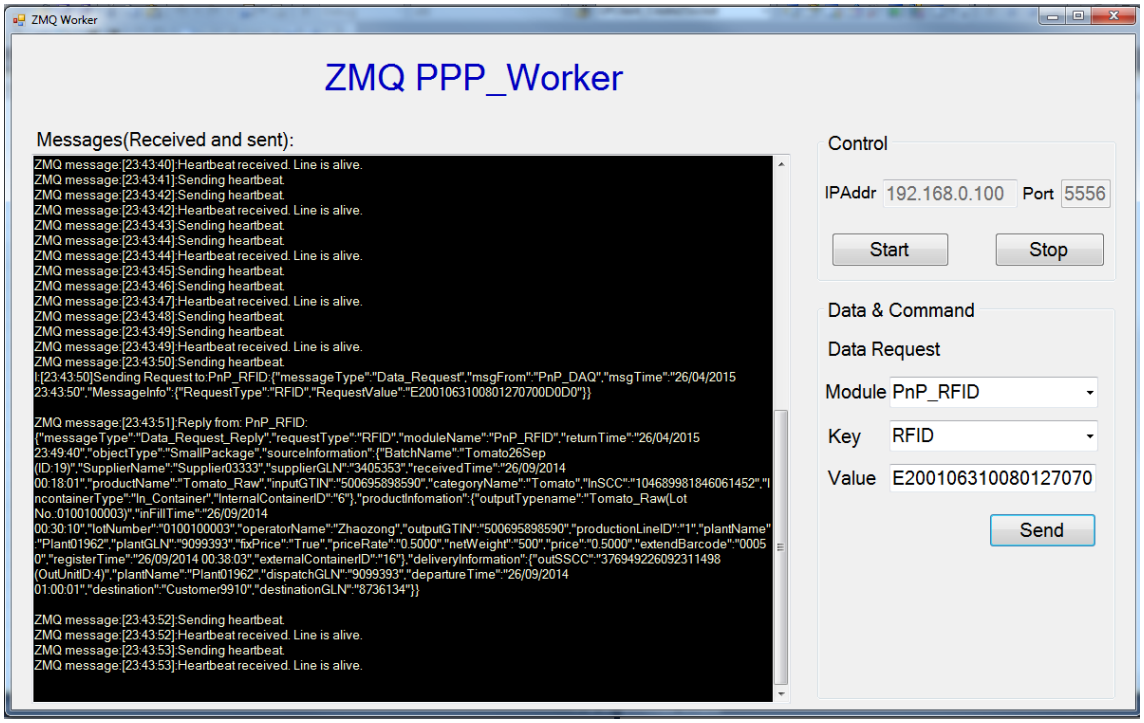


Figure 16 Other Module Request Information from Traceability Application

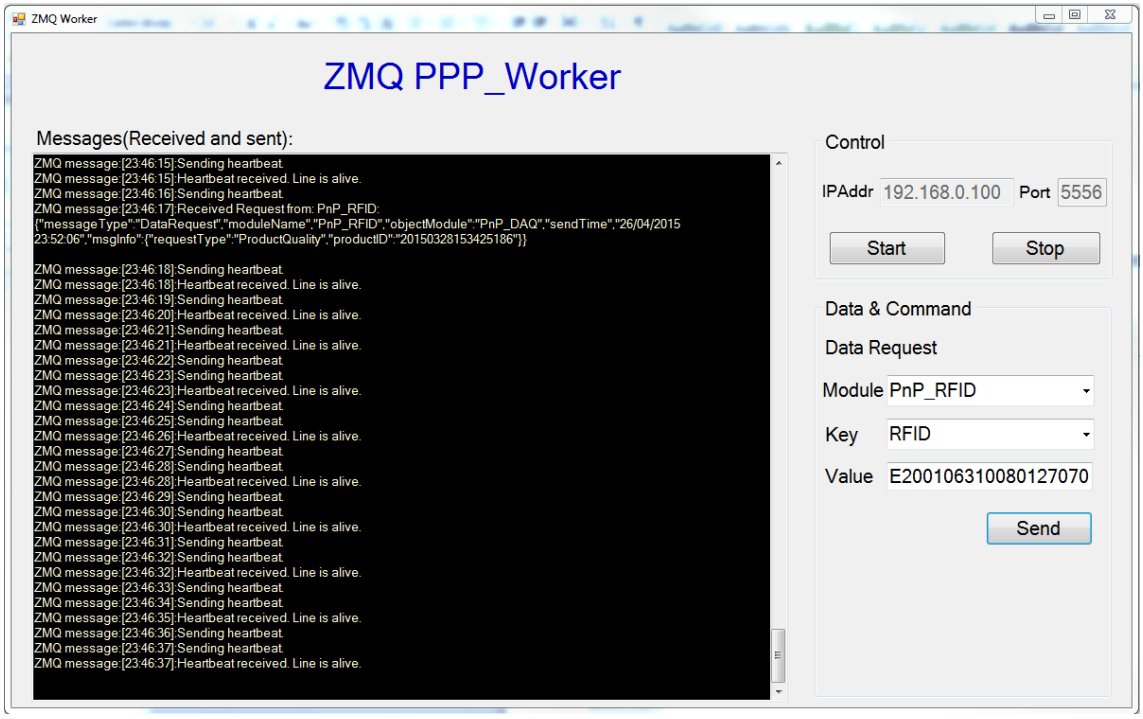


Figure 17 Other Module Receives Data Request from Traceability Application

The communication between traceability system and the other modules are through ZMQ as shown in Figure 9. The ZMQ router receives messages and then forwards these messages to the object clients. Figure 18 shows how message is received and sent between traceability application (PnP_RFID) and another module (PnP_DAQ).

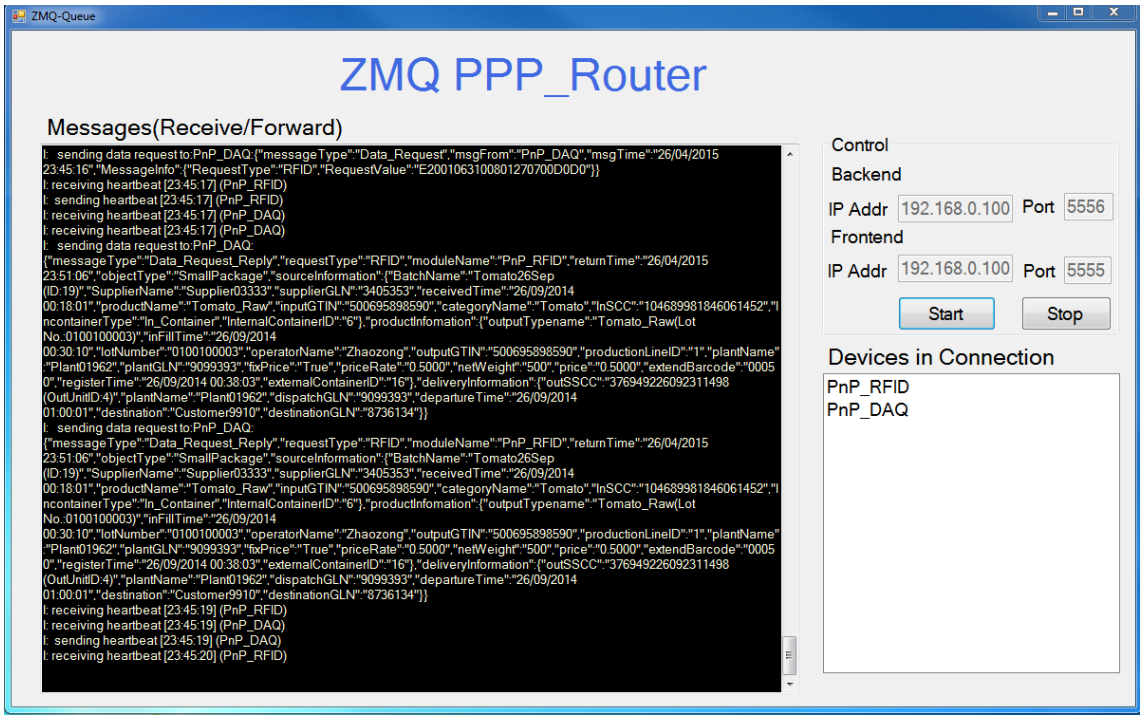


Figure 18 ROUTER Forwarding Messages between Traceability Application and Other Modules

In addition to ZMQ, the Zyre interface is also implemented and tested. Figure 19 shows the traceability application receives JSON message in Zyre, and replies JSON message to the node talks to it. Figure 20 shows other node receives whisper message from traceability system, and Figure 21 shows all nodes in group module1 receive shouted messages from traceability system.

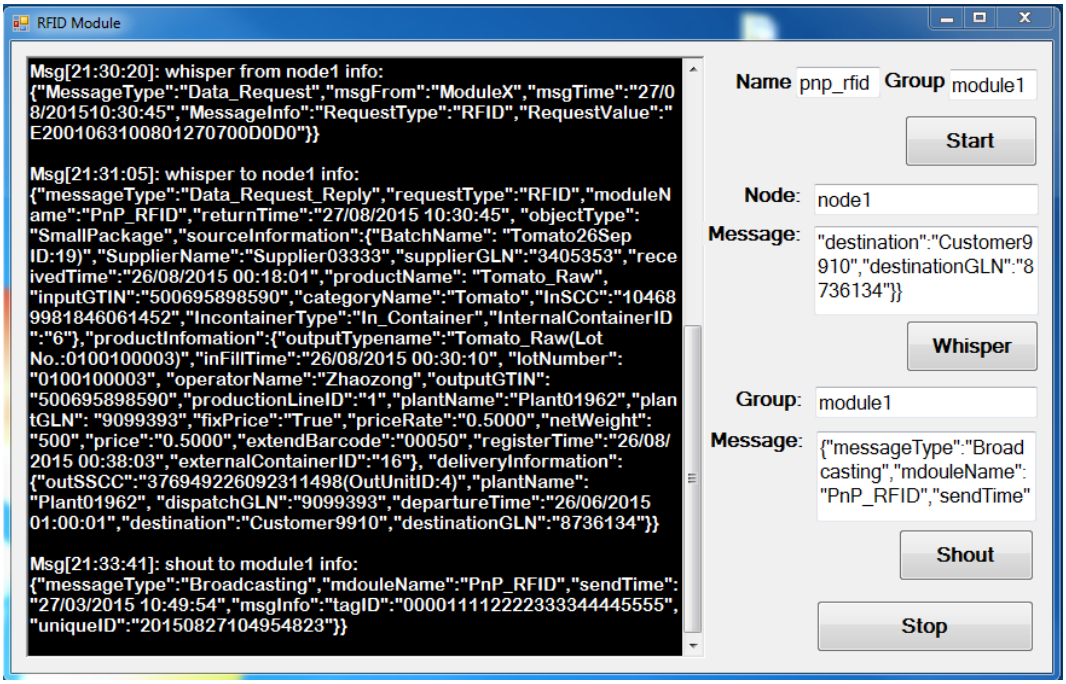


Figure 19 Traceability System Whisper and Shout to Other Nodes

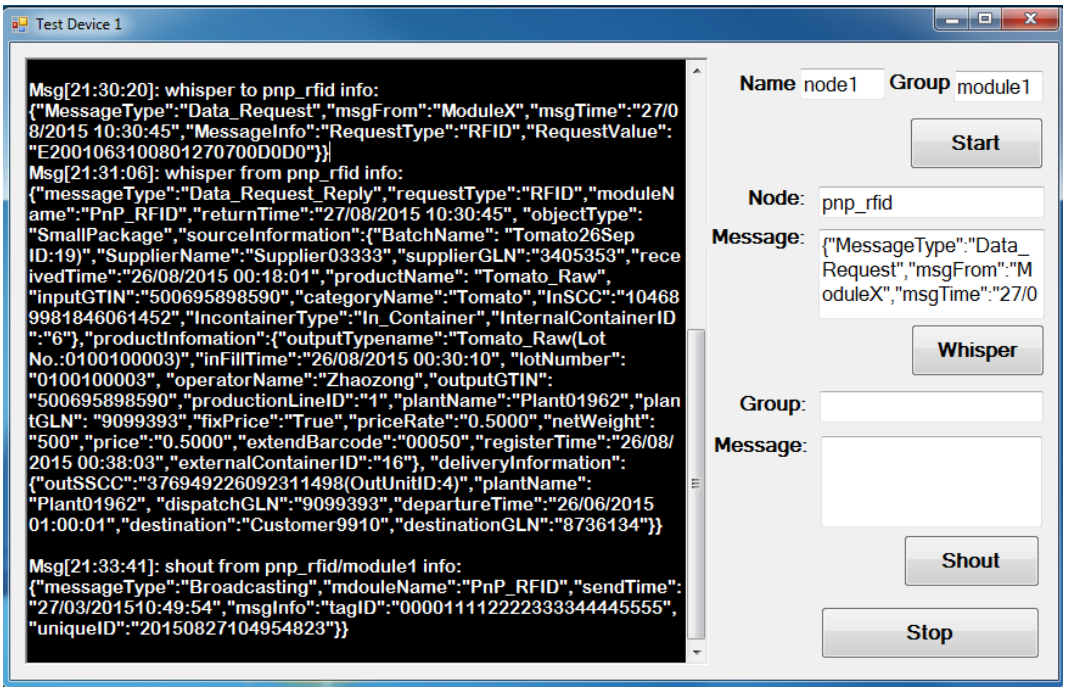


Figure 20 Other Node Receive Whisper Message from Traceability System

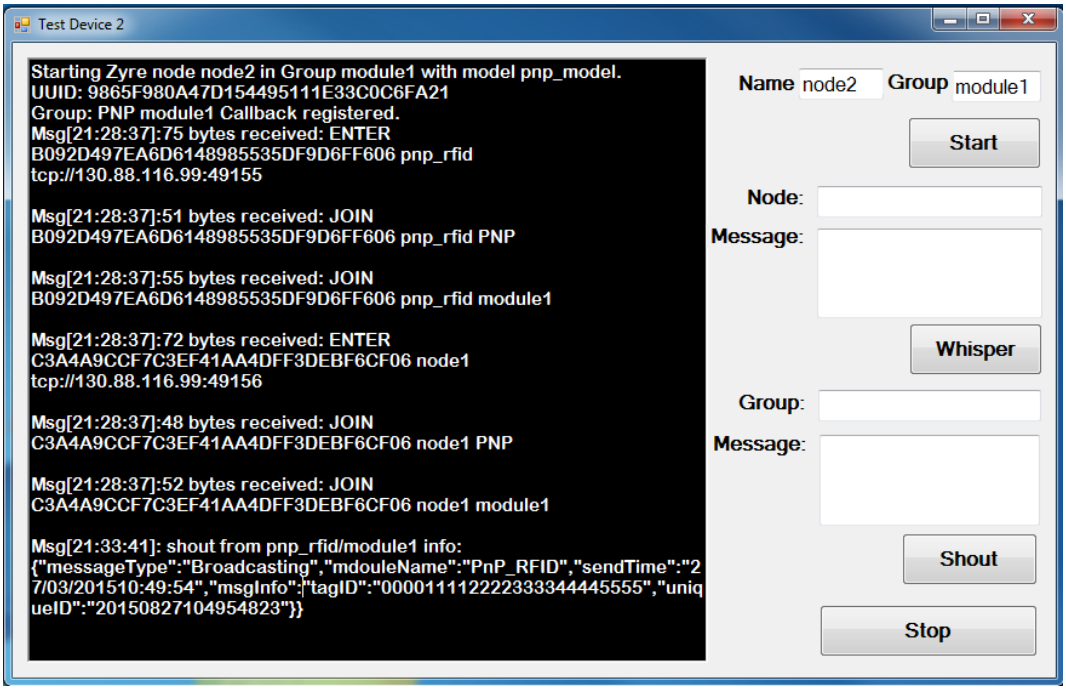
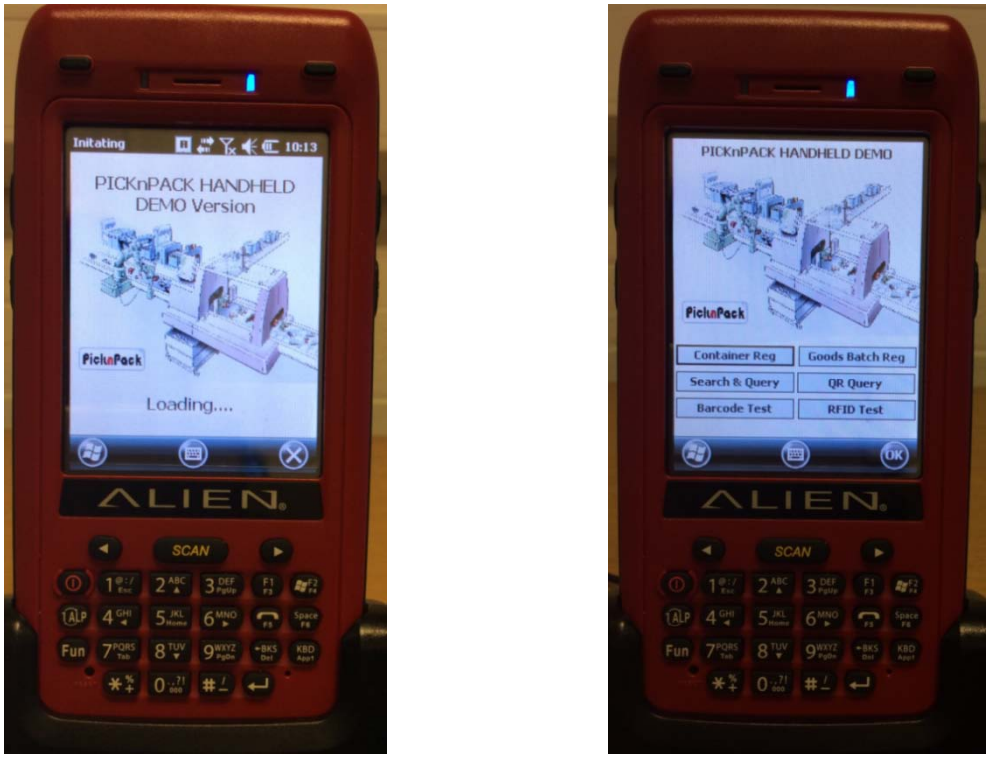


Figure 21 Other Node Receive Shouted Message from Traceability System

In addition to the traceability application based on PC, a handheld RFID reader can be used to trace the product information. The handheld RFID reader can detect the RFID tag, query the database, and display the matching records to the user on its interface. The start page and main page is given in Figure 22, and search and query page is given in Figure 23.



(1) System Booting (2) Main Interface

Figure 22 Start Page and Main Interface of Handheld Reader Application

(1) Incoming Goods Tracking

(2) Small Package Tracking

Figure 23 Information Tracing with Handheld RFID Reader

In addition, the handheld RFID reader can also decode the QR code generated by the traceability application, which is as shown in Figure 24.

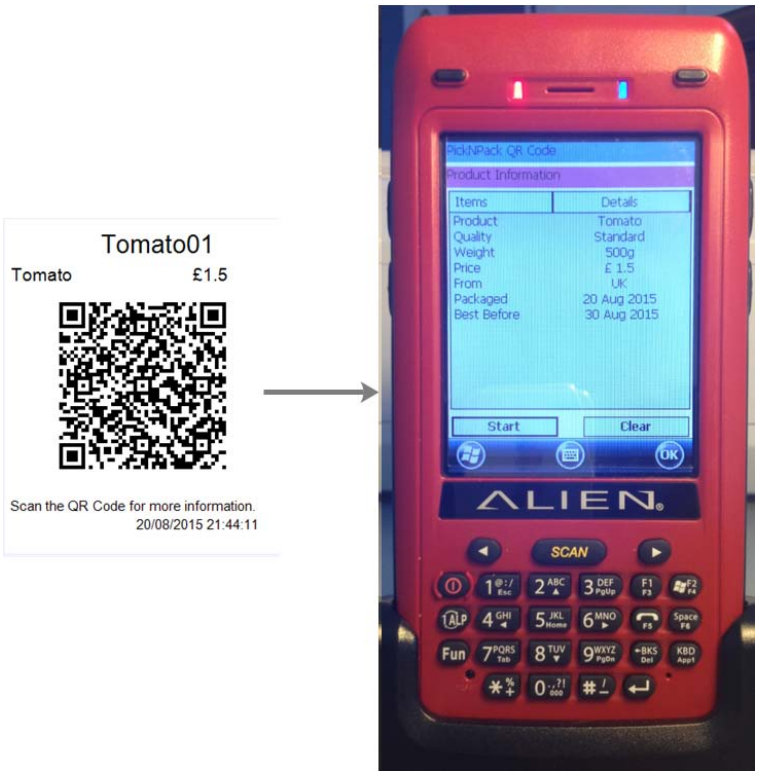


Figure 24 QR Code Decoding with Handheld Reader Application

All the above interfaces are some selected typical ones, which are cut in the test of the system.

5.3 Traceability System under Testing

The hardware modules and software interface under in-lab testing is as shown in Figure 25.

Figure 25 In-lab Test Bench of the RFID Traceability System

6 Summary

This report summarises the design and development of the traceability system for demonstration, including (1) hardware system, (2) process modelling and database design, and (3) software development and software integration.

The functions for integration are implemented with to the required protocols and standards in the current version of the traceability application. In order to achieve interoperability, ZeroMQ and JSON are successfully implemented. Further test and optimisation will be carried out in order to interact with other modules in the next step. The future work for the traceability system is suggested as follows:

(1) Fit the traceability system to the line for demonstration, including hardware and software
(2) Design the data sharing application interfaces using the standards, according to the interfaces other modules provide
(3) In line test and evaluation of performance of the traceability system
(4) Optimise the integrated traceability system
(5) Since ZeroMQ will be changed to Zyre, the updated functions will integrated to the traceability application.

The points listed above are the essential tasks to be achieved in the future work.

Reference

[1] European Communities (2007). Food Traceability, http://ec.europa.eu/food/safety/docs/gfl_req_factsheet_traceability_2007_en.pdf, accessed: 28 August 2015

[2] Hintjens, P. (2013). *ZeroMQ: Messaging for Many Applications*. Sebastopol, USA: O'Reilly Media, Inc.

[3] Hintjens, P. (2013). *ØMQ – The Guide,* http://zguide.zeromq.org/page:all, Accessed: 28 August 2015

[4] Ezzadeen, M. (2012). ZeroMQ via C#: Introduction, http://www.codeproject.com/Articles/488207/ZeroMQ-via-Csharp-Introduction, Accessed: 28 August 2015

[5] Ecma International (2013). Standard ECMA-404 – The JSON Data Interchange Format (1st Edition), http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-404.pdf, Accessed: 28 August 2015

[6] Newtonsoft (2015). JSON .NET – Popular high-performance JSON framework for .NET, http://www.newtonsoft.com/json, Accessed: 28 August 2015

[7] Philips, J., Bruyninckx, H. (2015). Pick-n-Pack WP2 Software Workshop Handouts, WP2 Workshop Pick-n-Pack meeting, Leuven, 3 March 2015

[8] Wagner, F., Schmuki, R., Wagner, T., Wolstenholme, P. (2006) Modeling Software with Finite State Machine: A Practical Approach, USA: Auerbach Publications

[9] Zyre by Wikidot.com (2015) introducing Zyre, http://zyre-v1.wikidot.com/blog:introducing-zyre, Accessed: 20 August 2015

[10] PicknPack Gitlab (2015) Pick-n-Pack/PnP-dll, https://gitlab.mech.kuleuven.be/rob-picknpack/pnp-dll/tree/bcfd4be42763c4683cae371f4f4f203173b14ec4,Accessed: 20 August 2015