SIEMENS

Application example • 12/2015

# SIMATIC RF650M Remote

## SIMATIC RF650M / TCP/IP via WLAN / Programming help

https://support.industry.siemens.com/cs/ww/en/view/109479885

# Warranty and Liability

**Note**

The Application Examples are not binding and do not claim to be complete with regard to configuration, equipment or any contingencies. The Application Examples do not represent customer-specific solutions. They are only intended to provide support for typical applications. You are responsible for the correct operation of the described products. These Application Examples do not relieve you of the responsibility of safely and professionally using, installing, operating and servicing equipment. When using these Application Examples, you recognize that we cannot be made liable for any damage/claims beyond the liability clause described. We reserve the right to make changes to these Application Examples at any time and without prior notice. If there are any deviations between the recommendations provided in this Application Example and other Siemens publications – e.g. Catalogs – the contents of the other documents shall have priority.

We do not accept any liability for the information contained in this document.

Any claims against us – based on whatever legal reason – resulting from the use of the examples, information, programs, engineering and performance data etc., described in this Application Example shall be excluded. Such an exclusion shall not apply in the case of mandatory liability, e.g. under the German Product Liability Act (“Produkthaftungsgesetz”), in case of intent, gross negligence, or injury of life, body or health, guarantee for the quality of a product, fraudulent concealment of a deficiency or breach of fundamental contractual obligations (“wesentliche Vertragspflichten”). The compensation for damages due to a breach of a fundamental contractual obligation is, however, limited to the foreseeable damage, typical for the type of contract, except in the event of intent or gross negligence or injury to life, body or health. The above provisions do not imply a change of the burden of proof to your detriment.

Any form of duplication or distribution of these Application Examples or excerpts hereof is prohibited without the expressed consent of Siemens AG.



**Security information**

Siemens provides products and solutions with industrial security functions that support the secure operation of plants, solutions, machines, equipment and/or networks. They are important components in a holistic industrial security concept. With this in mind, Siemens’ products and solutions undergo continuous development. Siemens recommends strongly that you regularly check for product updates.

For the secure operation of Siemens products and solutions, it is necessary to take suitable preventive action (e.g. cell protection concept) and integrate each component into a holistic, state-of-the-art industrial security concept. Third-party products that may be in use should also be considered. For more information about industrial security, visit http://www.siemens.com/industrialsecurity.

To stay informed about product updates as they occur, sign up for a product-specific newsletter. For more information, visit https://support.industry.siemens.com.

# Table of Contents

# 1 Task

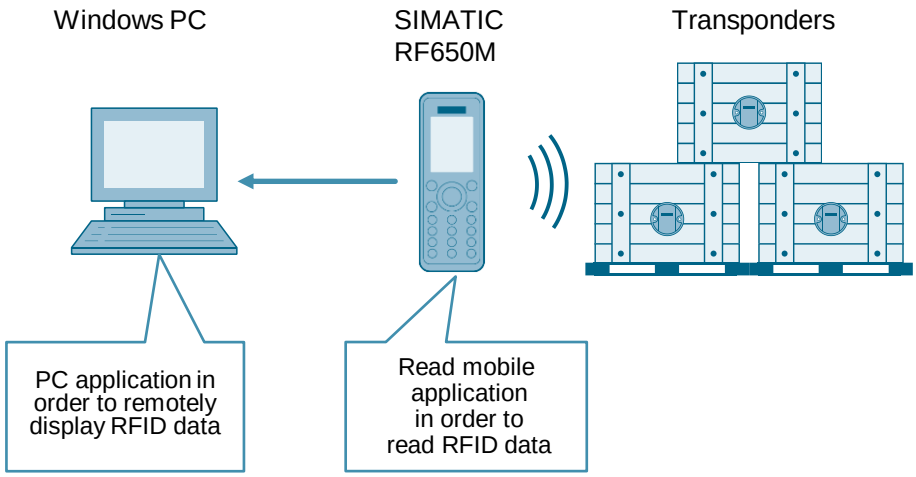
## Introduction

The SIMATIC RF650M mobile UHF handheld reader was developed for the mobile reading and writing of RF600 transponders. Transponder data can be read and edited via the “RF650M” software included in the delivery. However, it cannot be displayed remotely on a PC system.

## Overview of the automation task

The figure below provides an overview of the automation task.

Figure 1-1




Before the data - read via the preinstalled “RF650M” software - can be processed further, it has to be saved first via log file on the reader and then has to be transferred to a PC via USB with the Windows mobile device center.
In addition to the USB interface, the mobile reader also has a WLAN interface. In order to facilitate the just described procedure via USB, the read transponder data is to be transferred via WLAN to a remote PC.
For this purpose, a PC application (as server) is to be created for the remote display of the transponder data. A second application (as client) is to carry out rudimentary RFID read functions for the mobile reader and send the read data automatically to the PC application.

## Purpose of the application example

The application example demonstrates how an automated data transfer between the mobile SIMATIC RF650M handheld reader and a remote Window PC can look like. In addition the document includes many tips and instructions that show you what you have to observe, if you want to expand the application example or create an independent application for the mobile handheld reader.

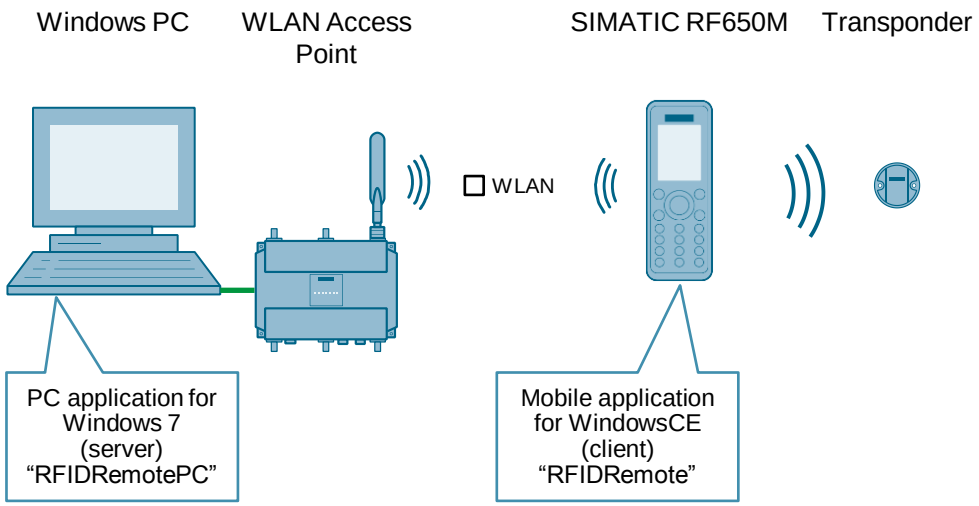
# 2 Solution

## 2.1 Overview

### Schematic layout

The figure below shows a schematic overview of the most important components of the solution:

Figure 2-1




### Advantages

The solution presented here offers the following advantages:

- You can transfer the transponder data from the handheld reader automatically to a PC via WLAN.
- The automation solution saves you time and costs.
- Through the included source files of the applications you can extend the application example as desired.

### Topics not covered in this application

This application does not include a description of

- the configuration of an access point
- object-oriented programming
- functionality of RFID systems

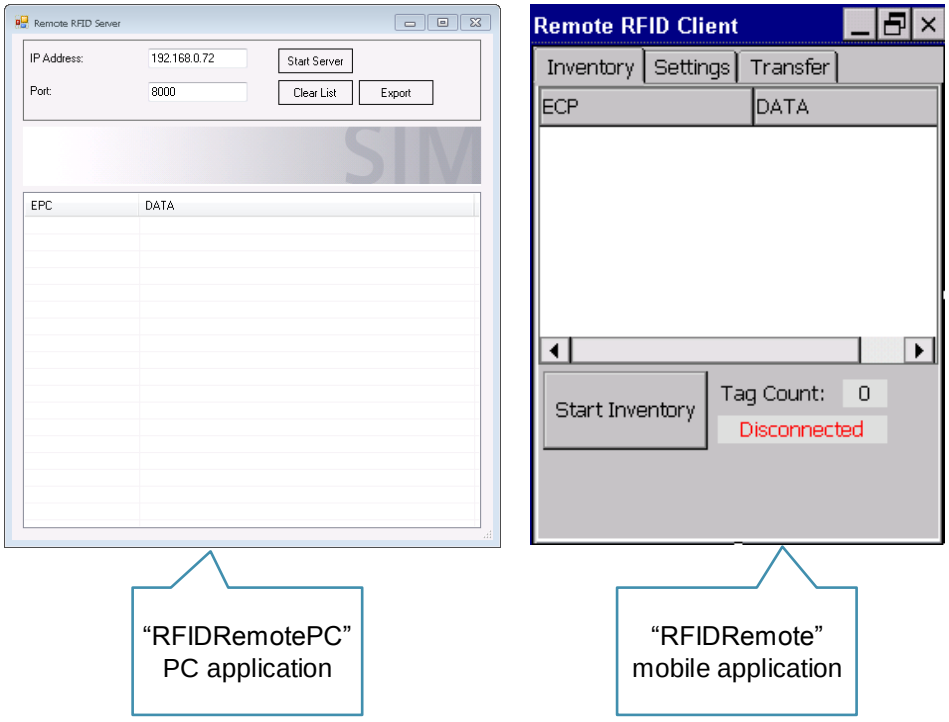
In addition, basic knowledge of Visual Studio, C#, .NET, TCP/IP and WLAN are assumed.

## 2.2 Description of the core functionality

### Overview and description of the user interface

The “RFIDRemotePC” PC application shows the transponder data read with the mobile “RFIDRemote” application.
The two applications establish a connection to each other via this WLAN interface and they can synchronize manually or automatically.

Figure 2-2




### Programming interfaces for PC and the mobile SIMATIC RF650M handheld reader

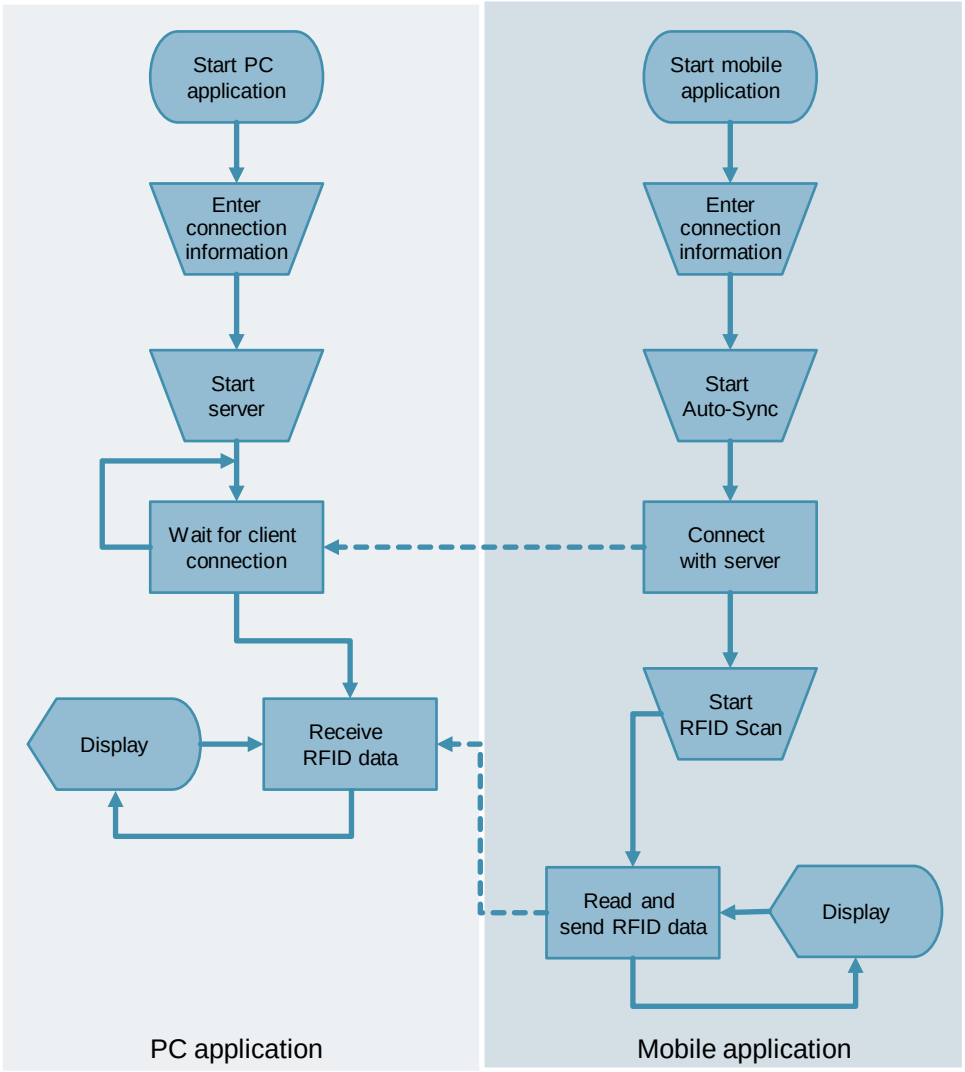
The core component for creating a program for the mobile handheld reader is Nordic ID NUR API. More information can be found in chapter “3 Basics on NUR API”. In chapter “4 Mode of Operation” you will find out how the NUR API is used in this application example.

In contrast to the mobile application, the PC application only accesses Visual Studio and .NET tools.

### Sequence of the core functionality

The figure below shows the core functionality of the application example for automatic synchronization:

Figure 2-3



# 2.3 Hardware and software components

## 2.3.1 Validity

This application example is valid for

- SIMATIC RF650M
- Windows CE 5.0/6.0 (mobile application) or Windows 7 (PC application)

## 2.3.2 Components used

The application example was created with the following components:

### Hardware components

Table 2-1

| Component | Qty | Article number | Note |
|---|---|---|---|
| SIMATIC RF650M | 1 | 6GT2813-0CA00 | |
| Docking station for SIMATIC RF650M | 1 | 6GT2898-0BB00 | |
| RF640T | n | 6GT2810-0DC00 (Europe)<br>6GT2810-0DC10 (USA) | Alternatively, any RF600 transponder can be used |
| SCALANCE W788-1 RJ45 | 1 | 6GK5788-1FC00-0AA0 | Alternatively, a different WLAN access point can also be used |

### Software components

Table 2-2

| Component | Qty | Article number | Note |
|---|---|---|---|
| Visual Studio 2005/2008 | 1 | | For the mobile application |
| Visual Studio 2010 | 1 | | For the PC application |
| Windows mobile device center (ActiveSync) | 1 | | Has been integrated in the system since Windows Vista |
| Morphic SDK | 1 | | Available from Nordic ID \3\ |
| NUR API | 1 | | Available from Nordic ID \3\ |

### Example files and projects

The following list includes all files and projects that are used in this example.

Table 2-3

| Component | Note |
|---|---|
| 109479885_RFIDRemote_CODE_V10.zip | This zip file contains the source files and the executable files of this application example |
| 109479885_RFIDRemote_DOKU_V10.pdf | This document. |

# 3 Basics on NUR API

### The NUR API in general

The “NUR” API (programming interface) serves as interface between the WindowsCE operating system and the integrated RFID read head of the mobile SIMATIC RF650M handheld reader (device: Morphic by Nordic ID).
The API supports you in creating high-performance RFID applications for the handheld terminal with very little programming effort.
On request, Nordic ID (\3\) provides different free and well documented software development kits (SDKs) for each Nordic ID device.

The following requirements must be fulfilled in order to work with the SDKs and the API:

- Visual Studio 2005/2008
- .NET Compact Framework 2 SP1
- USB interface on the programming device
- Docking station of the handheld terminal
- Microsoft Windows mobile device center (ActiveSync)



### Programming with the NUR API

The step-by-step instruction below shows you how to proceed, if you want to create a RFID application for the mobile handheld reader:

1. Obtain all required software packages from Nordic ID:
   - Morphic SDK (for SIMATIC RF650M)
   - NurDistribution (contains the API DLL and the corresponding documentation)
2. Install the SDK.
3. Create a Visual Studio project with the following project type:
   Visual C#/C++ > Intelligent device > Windows CE 5.0 or 6.0 > device application.
4. Integrate the NUR API DLL (NurApiDotNetWCE.dll) in your project as reference.
5. Integrate the using directive for the API in your project files:

```
using NurApiDotNet;
```

6. You can now use the NUR API as a class within the namespace. Please note the following example code:

```
//Declaration of NUR-API class
NurApi myNurApi;

//Creating a new instance of NUR-API
myNurApi = new NurApi();

//Connecting the integrated reader to the .NET-application using
the NUR instance
myNurApi.ConnectIntegratedReader();

//Starting an RFID Inventory via NUR-API using the NUR instance
myNurApi.StartInventoryStream();
```

### Basic functions and program sequences of NUR API

The list below explains the most important basics for programming with NUR API:

- Before you can address the integrated RFID read head of the RF650M via the NUR API, you have to connect the API with the reader via the following method:

```
myNurApi.ConnectIntegratedReader();
```

- Transponder data is stored in the NUR API after reading in TagStorage. You can create an image of the memory area with the following code and access the read data:

```
NurApi.TagStorage myTagStorage = myNurApi.GetTagStorage();
```

- The NUR API now increasingly works with events. If you, for example, start a RFID read, the API fires an event (InventoryStreamEvent), as soon as a transponder has been read. Due to this mechanism you know time-discrete that the new transponder data is available in TagStorage.

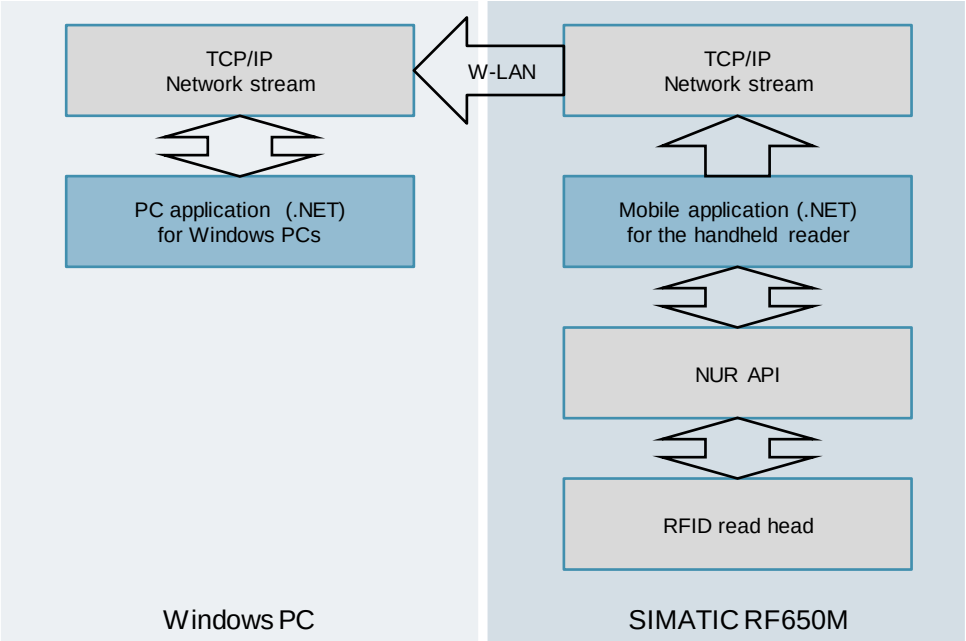
**Note** The complete documentation of NUR API can be found on the Nordic ID manufacturer's page (\3\).

# 4 Mode of Operation

## 4.1 General overview

The following figure gives a rough overview of how the most important components of this application example interact:

Figure 4-1




The mobile application for the handheld reader was created in .NET Compact (C#). The mobile handheld reader with WindowsCE as operating system communicates with the dedicated integrated RFID read head via the “NUR API” provided by Nordic ID.
TCP/IP via WLAN serves as interface for the appropriate PC application that was also created in .NET (C#),.

**Note** In addition to the NUR API, this application example uses another API called “MHL”. This API does not add to any core functionality. It only serves to easily create a tone when the transponders are read. The MHL API is not discussed in detail in this application example.

## 4.2 Functionality of the mobile “RFIDRemote” application

The mobile application for the RF650M handheld reader supports rudimentary RFID functions. It reads out transponder data and sends it to the PC application. The interaction of the application with the RFID read head is via predefined members of the NUR API class. The interface and the communication for the PC application are realized with .NET tools.

### Structural configuration of the mobile application

The mobile application for the handheld reader consists of five “regions”. Please have a look at the following figure:

Figure 4-2

The following table describes the five regions of the mobile application.

Table 4-1

| No. | Includes |
|---|---|
| 1. | Declarations that can be accessed from the entire project |
| 2. | Basic methods for the event handling of NUR API ConnectedEvents |
| 3. | Required methods in order to execute RFID inventories |
| 4. | Methods for RFID and program settings that the user can make in the application example |
| 5. | Methods that are required for the data transfer of the PC application |

**List of the most important methods**

The list below provides an overview of the most important methods of the mobile application. For more detailed information, please note the comments in the source files included.

- **startInventoryButton_Click**
  This button-click method starts/stops the RFID inventory runs whilst taking into account the inventory settings transferred with the “configureInventory” method. The following NUR methods are accessed with this method:
  - StartInventoryStream() – starts the RFID readings.
  - ClearTags() – deletes the entries in the TagStorage.
- **nur_InventoryStreamEvent**
  This method is executed as soon as an InventoryStreamEvent of the NUR API is fired. Within the method the TagStorage is requested by the RFID module and the “updateTagList” method is accessed in order to display the data from the TagStorage. The following NUR method is called with this method:
  - GetTagStorage() – creates an image of the TagStorage of the mobile handheld reader.
- **updateTagList**
  This method writes the data from the TagStorage in a ListView (System.Windows.Forms.ListView) in order to provide it visually for the user.
- **configureInventory**
  This method configures the inventories. The following NUR method is called with this method:
  - InventoryRead() – configures the RFID readings.

  The following settings can be made with it:
  - Inventory type (ECP; EPC + data)
  - Selection of the memory bank for the data to be read
  - Start address for the data to be read
  - Length of the data to be read
- **connectTcp**
  This method establishes a TCP connection between the mobile and the PC application.
- **connectionWatchDogTimer_Tick**
  As soon as a TCP connection has been established, a timer starts. For each time tick, a package of 1 byte is transmitted to the server via this method in order to monitor the connection.

- **reconnectTimer_Tick**
  If the sending of the WatchDogTimer package fails, the connection to the server is interrupted. Now another timer is started that starts a new connection buildup, at every timer tick via the “connectTcp” method.

  **sendTagStorage**
  The method requests the TagStorage from the RFID module and sends it via NetworkStream to the server. The following NUR method is called with this method:

  - GetTagStorage()

- **synchronisebutton_Click**
  This button-click method triggers the sending of the TagStorage to the server in manual mode.
- **autoTransferCheckBox_CheckStateChanged**
  This checkbox-check method starts/stops the automated transfer of the TagStorages to the server.

## 4.3 Functionality of the “RFIDRemotePC” application

The PC application receives the read transponder data from the mobile application of the handheld reader and visually prepares it.

The application consists of two basic classes. The main class from the type Windows.Forms includes all important methods and the user interface. Another class “RFIDRemoteDataHandle” encapsulates all functions on the communication with the client.

### Structural configuration of the PC application

The main class of the PC application consisting of 3 “regions”. Please have a look at the following figure:

Figure 4-3

```
namespace RFIDRemotePC
{
    public partial class RFIDPCForm : Form
    {
        //Global declarations
        /// <summary>
        /// Declaration of global properties
        /// </summary>
        GlobalDeclarations                    (1)

        //Form main()
        public RFIDPCForm()...

        //Transfer region
        /// <summary>
        /// Methods handling data transfer from hand terminal
        /// </summary>
        TcpTransferRegion                     (2)

        //Processing data region
        /// <summary>
        /// Methods processing data from hand terminal
        /// </summary>
        ProcessingDataRegion                  (3)
    }
}
```

The table below describes the 3 regions of the main class of the PC application:

Table 4-2

| No. | Description |
|---|---|
| 1. | Includes all declarations that can be accessed from the entire project |
| 2. | Includes all methods that are required for the client for establishing a connection |
| 3. | Includes all methods that are required for the further processing of the data transferred. |

### List of the most important methods

The list below provides an overview of the most important methods of the main class of the PC application. For more detailed information, please note the comments in the source files included.

- **startListener**
  This method starts the TCP listener in order to respond to incoming connection requests.
- **beginToAcceptClient**
  This method starts a separate thread in order not to block the interface. “handleNewTcpClient” is called as call-back function.
- **handleNewTcpClient**
  This call-back method accepts connection requests and provides the network stream to the connection partner. Then “beginToAcceptClient” is accessed again.
- **updateListView**
  This event-triggered method passes the received data on to a ListView in order to provide the data visually to the user.
- **resetListView**
  This event-triggered method deletes the contents of the ListView.
- **exportButton_Click**
  This button-click method opens an export dialog for the contents of the ListView and exports them.

### Structural configuration of the communication class within the PC application

The communication class of the PC application consists of 2 “regions”, please note the following figure:

Figure 4-4

```
namespace RFIDRemotePC
{
    class RFIDRemoteDataHandle
    {
        //Declarations region
        /// <summary>
        /// Declarations of this communication class
        /// </summary>
        DeclarationsRegion                    (1)

        //Receiving data region
        /// <summary>
        /// Methods receiving/handling data from hand terminal
        /// </summary>
        ReceivingDataRegoin                   (2)
    }
}
```

The following table describes the 2 regions of the communication class of the PC application:

Table 4-3

| No. | Description |
|---|---|
| 1. | Includes the declarations for the class-wide access of tags |
| 2. | Includes all necessary methods of the communication class |

**List of the most important methods**

The following list gives an overview of the most important methods of the communication class of the PC application. For more detailed information, please note the comments in the source files included.

- **readDataTransfer**
  This method starts the reading from the NetworkStream in a separate thread in order not to block the interface. “handleStreamDataCallBack” is called as call-back function.
- **handleStreamDataCallBack**
  The data read from the NetworkStream is transferred to this call-back method. The data is decoded in the method and provided to the application as strings. Then “readDataTransfer” is called again.
- **onReceivedDataEvent**
  This method fires the “NewDataReceivedEvent” event, as soon as new data is read from the mobile application from the NetworkStream.

## 4.4 Functioning of the data transfer

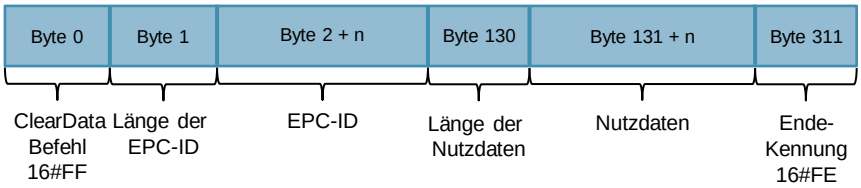
The data transfer between the two applications is exclusively unidirectional and asynchronous from the mobile handheld reader to the PC application. The expanded TcpListener .NET classes (as server) and TcpClient serve as software basis of the TCP communication that are each based on Systems.Net.Sockets.

The transfer is done based on data packages with predefined size. Here, each read transponder unit (EPC + user data) forms an individual data unit (corresponds to a data package). When several read transponder units are available, they are sent one after another as separate data units.

**Structure of the data packages (byte arrays)**

Read transponder data is read out from the TagStorage of NUR API and written serialized in a byte array with the following structure:

Figure 4-5



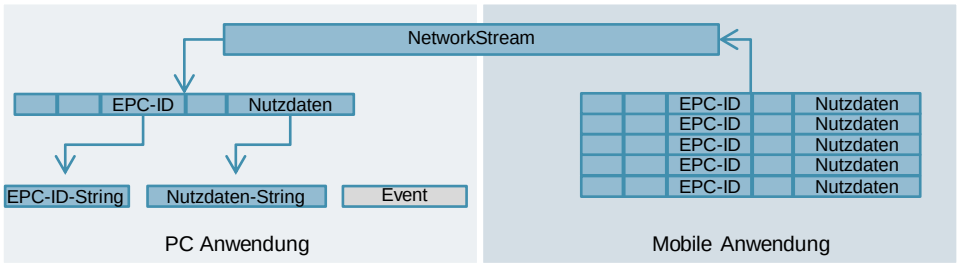

**Note** The ClearData command (16#FF) is only sent with the first transponder unit of the read list. In the units that follow, the byte has the value 16#0.

**Note** With the package size specified as 312, a maximum of 180 bytes of user data can be transferred from the USER memory of the transponders. Thus, all Siemens transponders apart from the RF622L and RF622T can be fully used. These two transponders have a user data size of the user bank of 3224 bytes each.

**Transmitting the data package**

The data package is written in the NetworkStream of the TCP connection once it has been formed by the mobile application. The PC application reads the data package from this NetworkStream and deserializes the byte array again. The user then has a string each in the “RFIDRemoteDataHandle” communication class for the ECP ID and the user data. In addition, an event is fired so that the user can pick up the data before new data is transferred and the old one is overwritten.

Figure 4-6

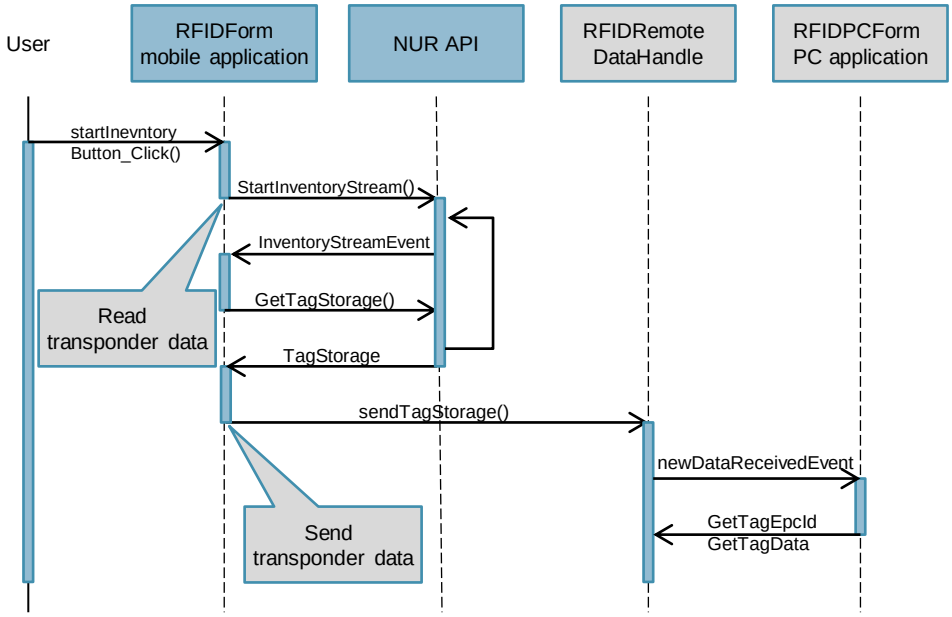



**Note** In addition to the data packages, 1 byte-sized data packages are cyclically written into the NetworkStream from the mobile application. This is used to monitor the connection.

### Sequence of the data transfer

The following sequence diagram shows the sequence from “Read transponder data” to “Send transponder data”. The connection was established before this sequence.

Figure 4-7



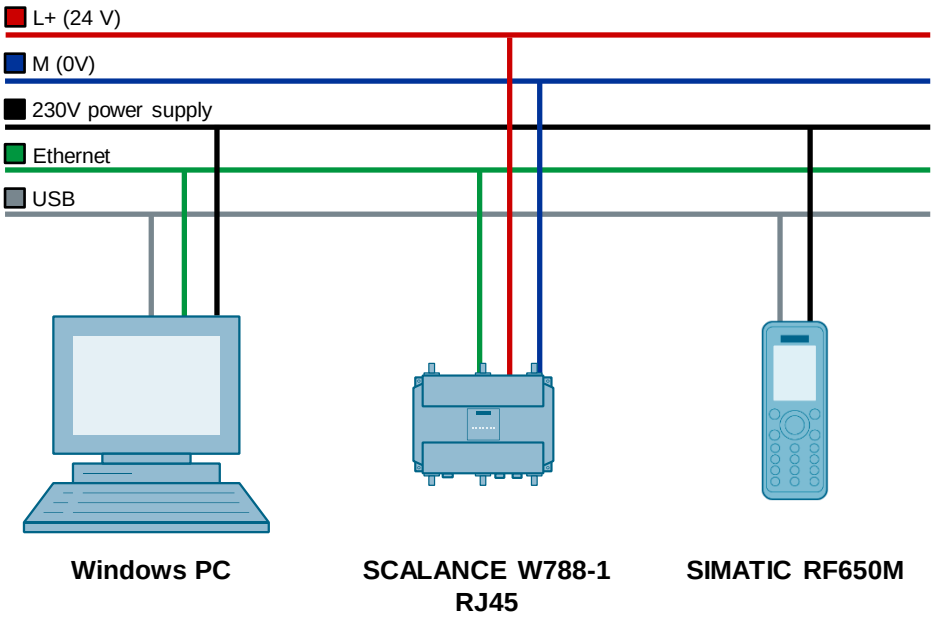
# 5 Installation and Commissioning

This chapter explains the required steps for commissioning the application example.

## 5.1 Installing the hardware

The figure below shows the hardware configuration of the application.

Figure 5-1




Carry out the hardware setup of the application example whilst taking the following step-by-step instruction into account:

1. Connect the access point to a suitable power supply.
2. Connect your programming device/your PC with the access point via Ethernet.
3. Connect the docking station of the RF650M to a suitable power supply.
4. Connect the docking station with the programming device/PC via USB.
5. Insert the RF650M into the docking station.

**Note** The setup guidelines for the SIMATIC RF650R \5\ and the SCALANCE W788-1 RJ45 \6\ always have to be observed.

## 5.2 Installing the software (download)

This chapter describes the steps for the installation of the development environments in order to expand the application example or to view the code.

1. Install Visual Studio 2005
2. Install Visual Studio 2010
3. Install the Morphic SDK (by Nordic ID)
4. Install .NET Compact Framework 2 SP1

## 5.3 Commissioning

This chapter describes the steps required to commission the application example.

**Providing the mobile application on the SIMATIC RF650M**

The list below explains step-by-step what you have to do in order to start the mobile application on the mobile handheld reader.

1. The executable applications “RFIDRemotePC” and “RFIDRemote” can be found on the HTML page (\1\) from which you have downloaded this document. Save the ZIP archive **109479885_RFIDRemote_CODE_V10.zip** on your hard disk.

2. The executable files for the mobile application and for the PC application are in the ZIP archive. Unzip the ZIP archive:

3. Start the Windows mobile device center.

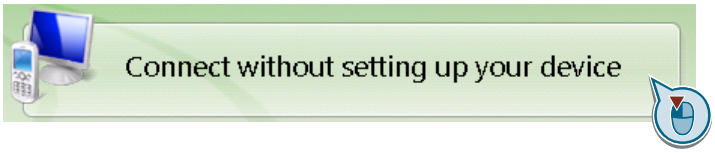
**Note** The Windows mobile device center opens automatically as soon as you connect the SIMATIC RF650M via USB with your PC. The Windows mobile device center is installed when it is first connected. Follow the instructions on your monitor.

4. When your RF650M is inserted in the docking station and it is connected via USB with the PC, the Windows mobile device center will show you that your PC is connected with the device (“connected”).

   Figure 5-2


5. Click on “Connect without setting up your device”.

   Figure 5-5

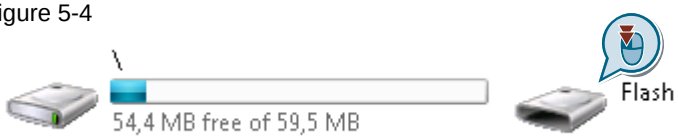


6. Now click “File Management” > “Browse the contents of your device”

Figure 5-3


7. Then your explorer will open and you will be in the file system of the handheld reader. Navigate to the flash memory of the “Flash” device and open it by double clicking it.

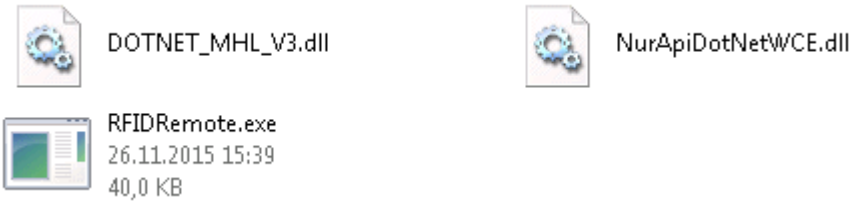
Figure 5-4




8. Create a folder and give it a name, for example, “RFIDRemote”.

9. Copy the files “RFIDRemote.exe”, “DOTNET_MHL_V3.dll” and “NurApiDotNetWCE.dll” in the just created folder. You can find these files in the “RFIDRemote_EXE” folder that you have previously extracted.

Figure -55



### WLAN connection buildup of the SIMATIC RF650M with an access point

The list below explains step-by-step what you have to do in order to connect the mobile handheld reader via WLAN with an access point.

1. Setup your access point. The mobile handheld reader supports WLAN according to the standards IEEE 802.11 a/b/g/n. If you are using the same access point as in this application example, you can find an explanation of how to set it up in the operating instructions \4\.

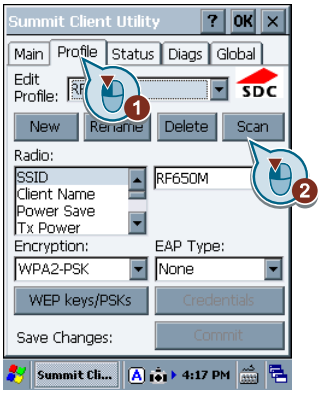
**Note** The access point of this application example supports you in the setup with a wizard. In order to be able to follow the commissioning instructions of this chapter, you have to make sure to assign a clearly defined SSID for your WLAN network. Furthermore, provide the network with a WPA2 key in order to guarantee the security of your network.

2. Pull the mobile handheld reader from the docking station and establish a WLAN connection to the access point. Proceed as follows:

3. Start the preinstalled “Summit Client Utility” application on the mobile handheld reader. You can find this application in “Start” > “Programs” > “Summit” > “SCU” on the mobile handheld reader.

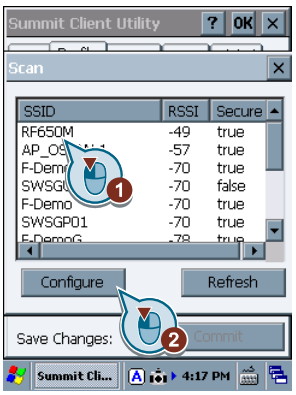
4. Go to the “Profile” tab and then click “Scan” in order to search for WLAN networks within reach.

Figure 5-6



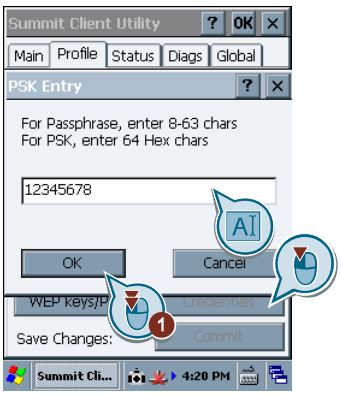

5. Select the SSID that you have configured in your access point and click “configure”.

Figure 5-7



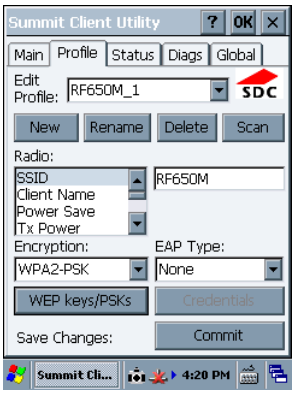

6. Enter the WLAN key specified by you and confirm with “OK”.

Figure 5-8



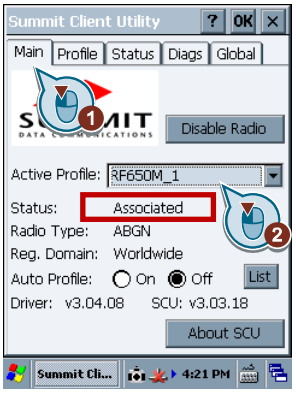

7. Confirm with “Commit”.

Figure 5-9



8. Go to the “Main” tab and select the just created connection profile from the “Active Profile” drop-down menu.
   If you have previously entered all the data correctly, the handheld reader will connect with the WLAN network. “Associated” is now displayed in “Status”.

   Figure 5-10



# 6 Operating the Application

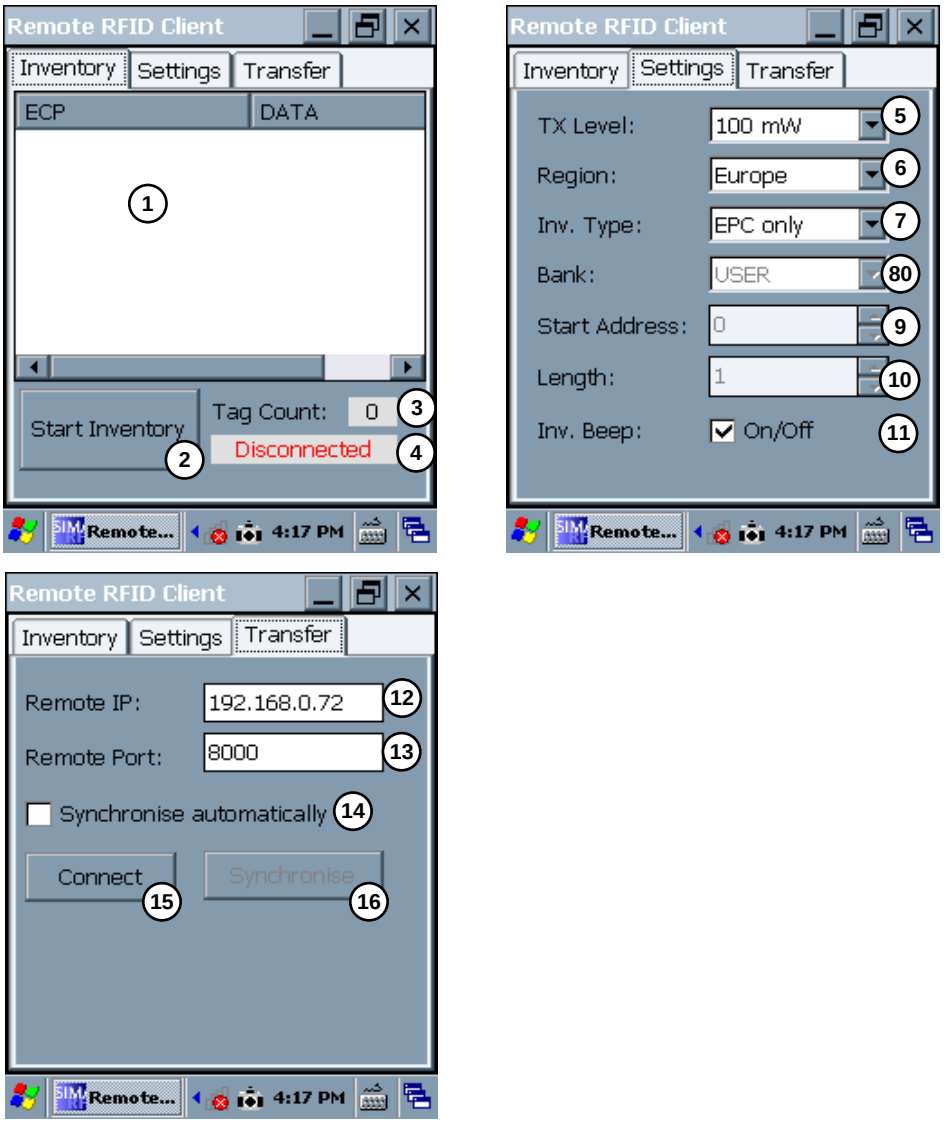
## 6.1 Overview and description of the user interfaces

The following two chapters explain the interfaces of the mobile and of the PC application.

### Interfaces of the mobile application

The interface of the mobile application of this application example consists of 3 tabs. In the first tab “Inventory”, the reading of the transponders is triggered and the read results are displayed.
In the second tab “Settings”, different settings for the RFID read head of the handheld reader and the application can be made.
The third tab “Transfer” provides the connection settings of the remote PC application.

Figure 6-1



## 6.1 Overview and description of the user interfaces

The table below explains the different buttons of the mobile application.

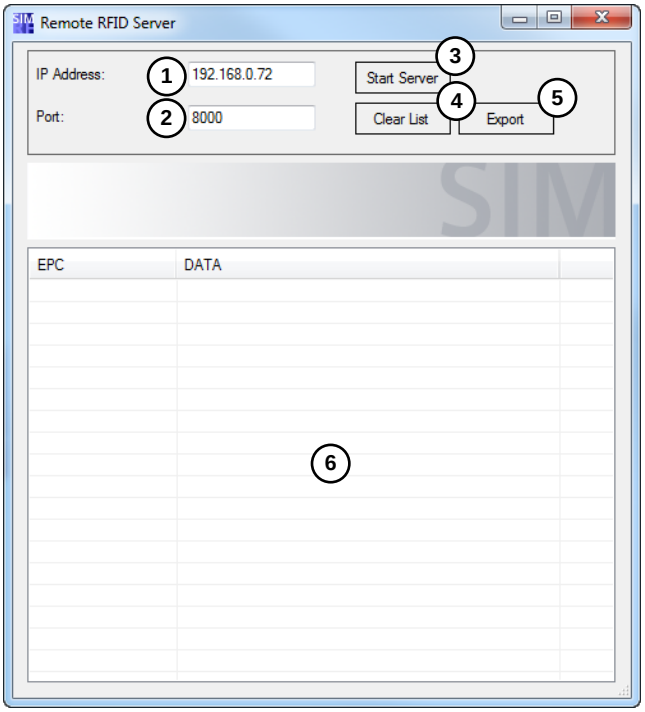
Table 6-1

| No. | Description |
|---|---|
| 1. | List of read results |
| 2. | Button to start/stop the RFID reading |
| 3. | Number of read transponders |
| 4. | Connection status of the remote PC application |
| 5. | Transmission power of the read head |
| 6. | Region (radio profile) |
| 7. | Inventory type (EPC; EPC + data) |
| 8. | Selection of memory area of the data (PWD; EPC; TID; USER) |
| 9. | Start address of the selected memory area |
| 10. | Length of the data to be read from the selected memory area in WORD<br>Note: No more than 90 WORD are transmitted to the PC application. |
| 11. | Check box to switch a signal tone for read transponders on/off. |
| 12. | IP address field for the remote PC application |
| 13. | Port field of the remote PC application |
| 14. | Check box for automatic data synchronization between mobile and PC application |
| 15. | Button to connect/disconnect a connection manually to the remote PC application |
| 16. | Button for the manual data synchronization between mobile and PC application |

### Interface of the PC application

The interface of the PC application contains a list for the remote display of the read transponders of the mobile application. In addition, you can set connection parameters there, start/stop the server, empty the list or export the received data.

Figure 6-2




The table below explains the different buttons of the PC application.

Table 6-2

| No. | Description |
|---|---|
| 1. | IP address field for the PC application |
| 2. | PC field for the PC application |
| 3. | Button to start/stop the server |
| 4. | Button to delete the transponder list |
| 5. | Button to export the transponder list |
| 6. | Transponder list |

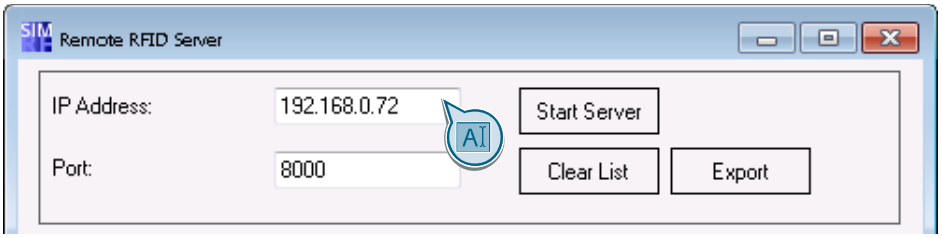
## 6.2 Operating the application example

The table below describes step-by-step how you can operate this application example.

1. Start the PC application by double clicking the “RFIDRemotePC.exe” file. This EXE file can be found in the “RFIDRemotePC” folder that was extracted by you.

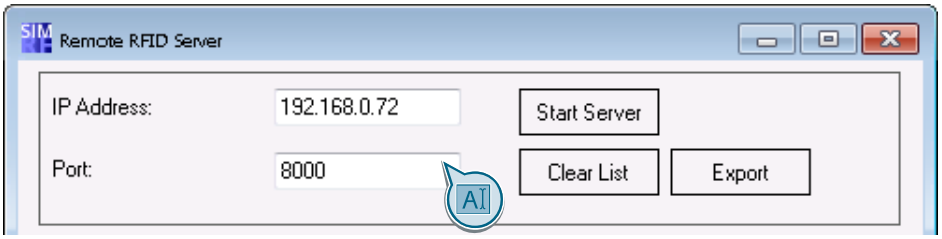
2. Enter the IP address of the network interface of your PC into the IP address field via which the mobile application is to be connected.

   Figure -63



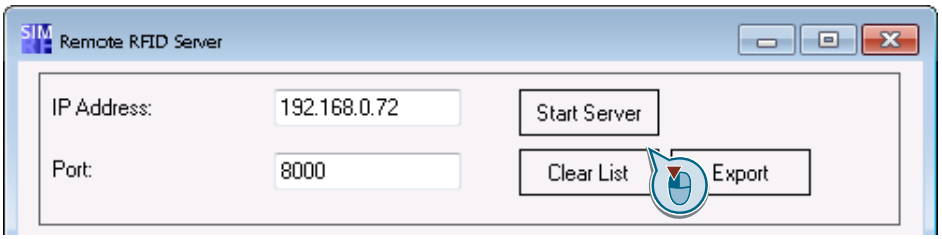

3. Enter a suitable port (observe individual firewall settings) into the port field (ports are only permissible from 0 to 65535) via which the mobile application is to connect.

   Figure 6-4




4. Click the “Start Server” button. If the IP and port values are correct, the server will start and the text of the button will change to “Stop Server”.

   Figure 6-5

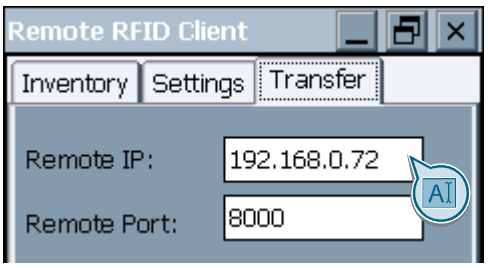



5. Now start the “RFIDRemote” application on the handheld reader by double clicking the “RFIDRemote.exe EXE file that you have previously copied. You will find the file by double clicking “My Device” on the handheld reader from the desktop and then navigating to the “Flash” folder. This is where you will find the “RFIDRemote” folder created by you that includes the EXE file.

6. Go to the “Transfer” tab.

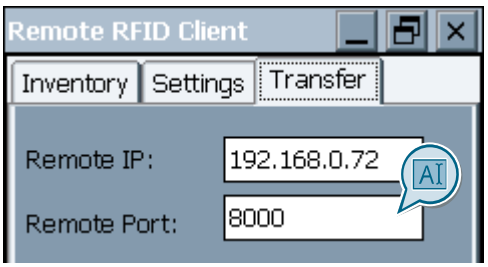
7. Enter the IP address of the remote PC into the IP address field that you have previously assigned in the PC application.

Figure 6-6



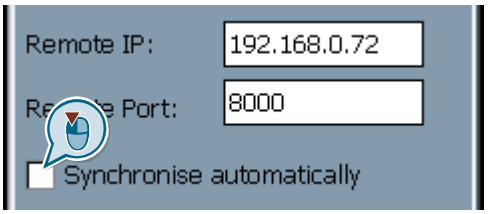

8. Enter the port of the remote PC application into the port field that you have previously assigned in the PC application.

Figure 6-7




9. Enable the “Synchronise automatically” check box. The mobile application now automatically establishes a connection to the PC application with the connection information specified by you. The transponder data is automatically synchronized. If the connection is disconnected, the mobile application automatically tries to establish new connection.

Figure 6-8




**Note** If you do not enable the control box, you have to manually establish a connection via the “Connect” button and manually synchronize the transponder data via the “Synchronise” button.

10. Go the “Inventory” tab to check whether the two applications are connected.

11. The mobile application is successfully connected with the PC application when it says “Connected” in the connection status field.

Figure 6-9

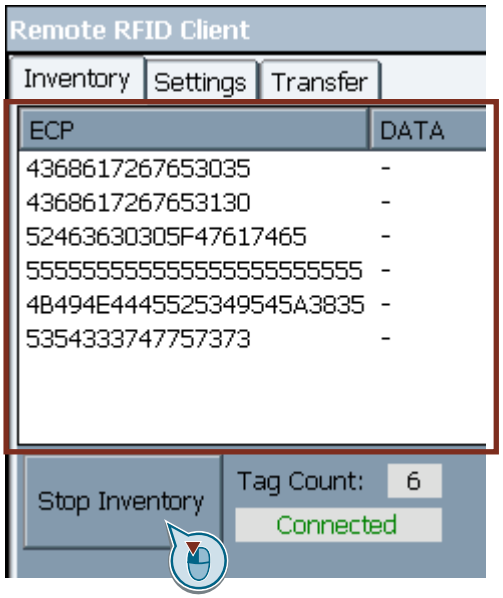
**Note** Check the entered connection information or check the WLAN connection between the handheld reader and the access point if the connection could not be established.

12. Go to the “Settings” tab and make all the required settings.

13. Go to the “Inventory” tab.

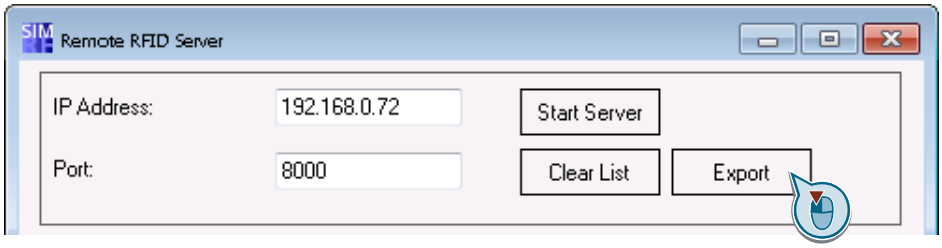
14. Start a reading via the “Start Inventory” button. As long as you do not click this button again, the handheld reader will carry out non-stop inventories. Each read transponder will be shown to you on the handheld reader and will automatically be transferred to the PC application and will also be displayed there.

    Figure 6-10




15. If required, you can export the read transponders via the PC application either as .xls, .txt or .csv export.

    Figure 6-11



# 7 Links & Literature

Table 7-1

| | Topic | Title |
|---|---|---|
| \1\ | Siemens Industry Online Support | https://support.industry.siemens.com |
| \2\ | Download page of the entry | https://support.industry.siemens.com/cs/ww/en/view/109479885 |
| \3\ | Manufacturer's website of the mobile reader | http://international.nordicid.com/ |
| \4\ | SIMATIC NET Industrial Wireless LAN SCALANCE W780/W740 to IEEE 802.11n Web Based Management | https://support.industry.siemens.com/cs/ww/en/view/79689667 |
| \5\ | SIMATIC Ident RFID systems SIMATIC RF650M mobile reader Operating Instructions | https://support.industry.siemens.com/cs/ww/en/view/109475735 |
| \6\ | SIMATIC NET Industrial Wireless LAN SCALANCE W788-x / W748-1 | https://support.industry.siemens.com/cs/ww/en/view/79692166 |



# 8 History

Table 8-1

| Version | Date | Modifications |
|---|---|---|
| V1.0 | 12/2015 | First version |
| | | |
| | | |