SIEMENS

Application Example • 06/2016

SIMATIC RF650M Data Synchronization

SIMATIC RF650M / SIMATIC S7-1500 / TCP/IP via WLAN / Programming Aid

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 regarding the circuits shown, equipping and any eventuality. The Application Examples do not represent customer-specific solutions. They are only intended to provide support for typical applications. You are responsible for ensuring that the described products are used correctly. These Application Examples do not relieve you of the responsibility to use safe practices in application, installation, operation and maintenance. 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 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 damages for a breach of a substantial contractual obligation are, 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.

© Siemens AG 2016 All rights reserved

Security information

Siemens provides products and solutions with industrial security functions that support the secure operation of plants, systems, machines and networks.
In order to protect plants, systems, machines and networks against cyber threats, it is necessary to implement – and continuously maintain – a holistic, state-of-the-art industrial security concept. Siemens' products and solutions only form one element of such a concept.
Customer is responsible to prevent unauthorized access to its plants, systems, machines and networks. Systems, machines and components should only be connected to the enterprise network or the internet if and to the extent necessary and with appropriate security measures (e.g. use of firewalls and network segmentation) in place.
Additionally, Siemens' guidance on appropriate security measures should be taken into account. For more information about industrial security, please visit http://www.siemens.com/industrialsecurity.

Siemens' products and solutions undergo continuous development to make them more secure. Siemens strongly recommends to apply product updates as soon as available and to always use the latest product versions. Use of product versions that are no longer supported, and failure to apply latest updates may increase customer's exposure to cyber threats.
To stay informed about product updates, subscribe to the Siemens Industrial Security RSS Feed under http://www.siemens.com/industrialsecurity.

Table of Contents

© Siemens AG 2016 All rights reserved

1 Task

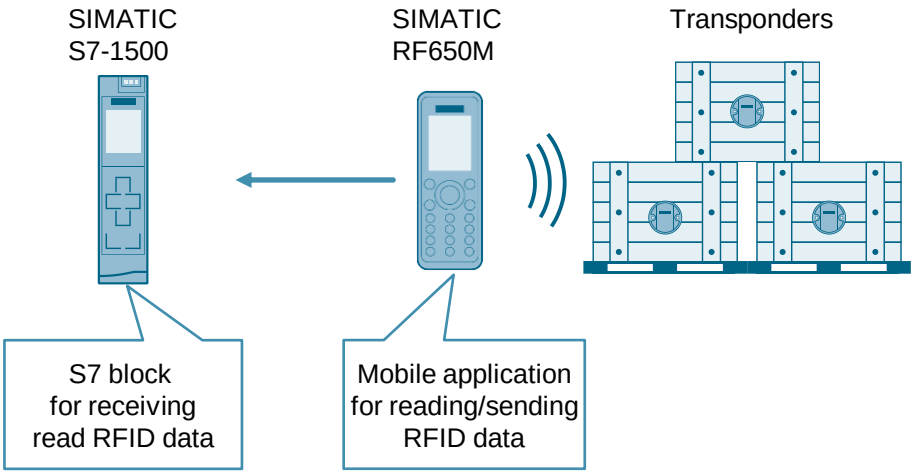
Introduction

The SIMATIC RF650M mobile UHF handheld reader was developed for mobile reading from and writing to RF600 transponders. Using the “RF650M” software supplied with the handheld reader, transponder data can be read and edited; however, it is not possible to continue processing the data remotely on an S7 controller.

Overview of the automation task

The figure below provides an overview of the automation task.

Figure 1-1



Before processing the data read – using the preinstalled “RF650M” software – can be continued, it is first necessary to save the data on the reader in a log file and then transfer it to a PC via USB with Windows Mobile Device Center. From there, there are several options to download the data to an S7 controller.
In addition to the USB interface, the mobile reader also features a WLAN interface. To simplify the procedure using USB/PC described above, the task is to directly synchronize the read transponder data with an S7 controller via a WLAN.
This involves creating an S7 block (as the server) for remotely receiving the transponder data. A second application (as the client) for the mobile reader is to run basic RFID read functions and automatically send the read data to the S7 block.

Objective of this application example

The application example shows what automated data transmission between the SIMATIC RF650M mobile handheld reader and an S7 controller can look like. In addition, the document contains tips and instructions that show you what needs to be considered if you want to extend the application example or create your own application for the mobile handheld reader.

© Siemens AG 2016 All rights reserved

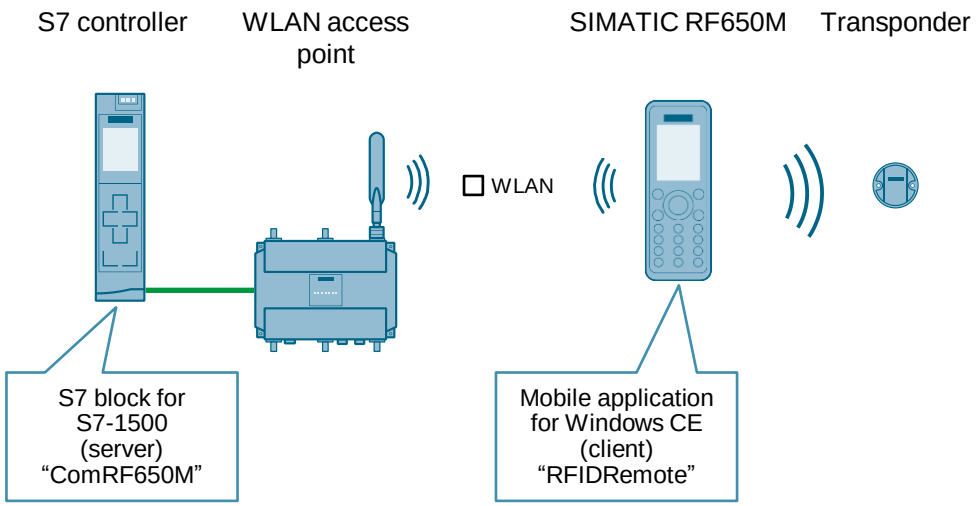
2 Solution

2.1 Overview

Diagrammatic representation

The diagrammatic representation below shows the most important components of the solution:

Figure 2-1



Advantages

The solution presented here offers the following advantages:

- You can automatically transfer transponder data from the mobile handheld reader to an S7-1500 via a WLAN.
- The automation solution saves you time and costs.
- The supplied source files of the applications allow you to extend the application example as required.

Scope

This application does not include a description of:

- Configuration of an access point
- Configuration of S7 controllers
- Object-oriented programming
- Programming with TIA Portal
- Principle of operation of RFID systems

In addition, basic knowledge of Visual Studio, C#, .NET, TCP/IP and WLAN is required if you want to extend the application.

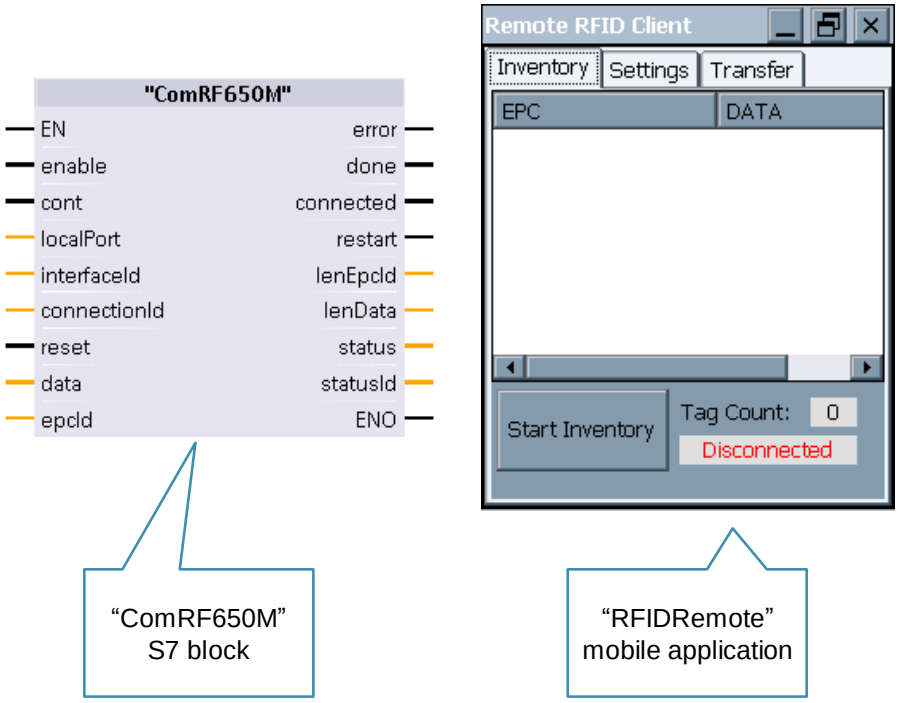
© Siemens AG 2016 All rights reserved

2.2 Description of the core functionality

Overview of the S7 block and the user interface of the mobile application

The “ComRF650M” S7 block receives the transponder data read with the “RFIDRemote” mobile application.
The S7 block and the mobile application establish a connection to each other via a WLAN and can synchronize manually or automatically.

Figure 2-2



Programming interfaces for the S7 block and the SIMATIC RF650M mobile handheld reader

The core component to create the program for the mobile handheld reader is the “NUR” API provided by Nordic ID. For more information, please refer to chapter “3 Basics of the NUR API”. In chapter “4 Principle of Operation”, you will learn how the NUR API is used in this application example.

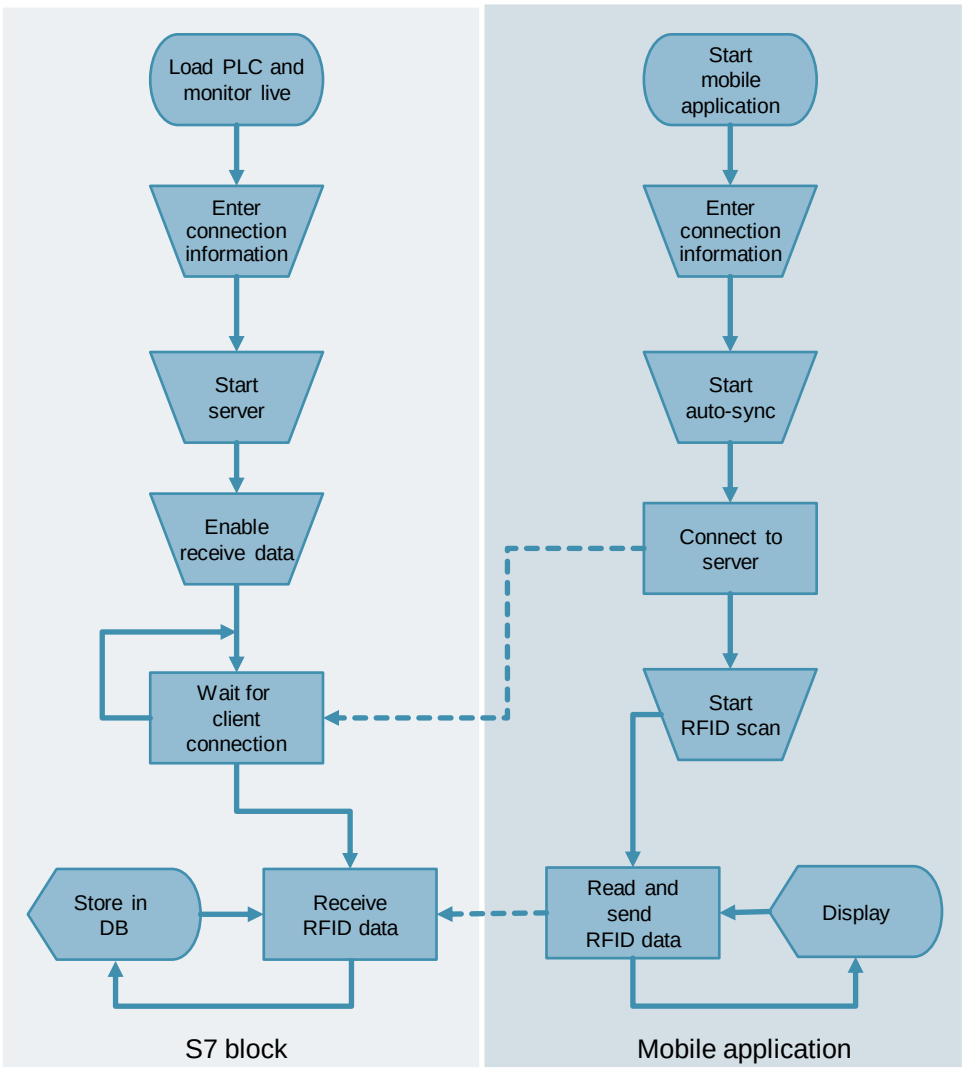
The S7 block uses only the “TRCV_C” standard block of TIA Portal and was created in SCL.

© Siemens AG 2016 All rights reserved

Sequence of the core functionality

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

Figure 2-3



© Siemens AG 2016 All rights reserved

2.3 Hardware and software components

2.3.1 Validity

This application example is valid for

- SIMATIC RF650M
- SIMATIC S7-1500
- Windows CE 5.0/6.0 (mobile application) or TIA Portal V13 SP1 (S7 block)

2.3.2 Components used

The application example was created with the following components:

Hardware components

Table 2-1

Component	No.	Article number	Note
SIMATIC RF650M	1	6GT2813-0CA00	
Docking station for SIMATIC RF650M	1	6GT2898-0BB00	
SIMATIC S7-1500 CPU 1511-1 PN	1	6ES7511-1AK00-0AB0	Alternatively, a different S7-1500 CPU can also be used.
RF640T	n	6GT2810-0DC00 (Europe) 6GT2810-2DC10 (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	No.	Article number	Note
Visual Studio 2005/2008	1		For the mobile application
Windows Mobile Device Center (ActiveSync)	1		
Morphic SDK	1		Obtain from Nordic ID \3\
NUR API	1		Obtain from Nordic ID \3\
STEP 7 TIA Portal Professional	1	6ES7822-1A.03-..	V13 SP1 or higher

© Siemens AG 2016 All rights reserved

Sample files and projects

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

Table 2-3

Component	Note
109479885_RF650M_RemotePLC1500_CODE_V10.zip	This zip file contains the projects and executable files of this application example.
109479885_RF650M_RemotePLC1500_DOKU_V10_en.pdf	This document.

© Siemens AG 2016 All rights reserved

3 Basics of the NUR API

NUR API – general

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

To work with the SDKs and the API, the following requirements must be met:

- Visual Studio 2005/2008
- .NET Compact Framework 2 SP1
- USB interface on the programmer
- Handheld terminal docking station
- Microsoft Windows Mobile Device Center (ActiveSync)

© Siemens AG 2016 All rights reserved

Programming with the NUR API

The following step-by-step instructions show you how to proceed if you want to create an RFID application for the mobile handheld reader:

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

```
using NurApiDotNet;
```

6. Now you can use the NUR API like a class within the namespace. Please see the following sample 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();
```

NUR API – basic functions and program flows

The following list explains the most important basics of programming with the NUR API:

- Before you can access the RF650M's integrated RFID read head via the NUR API, you have to connect the API to the reader using the following method:

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

- After reading, transponder data will be stored in the NUR API's TagStorage. The following code allows you to create an image of the memory area and access the read data:

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

- The NUR API intensively uses events. If you start, for example, an RFID read, the API fires an event (InventoryStreamEvent) as soon as a transponder has been read. This mechanism ensures that you are promptly informed of new transponder data in the TagStorage.

Note For the complete NUR API documentation, please visit the Nordic ID website (\3\).

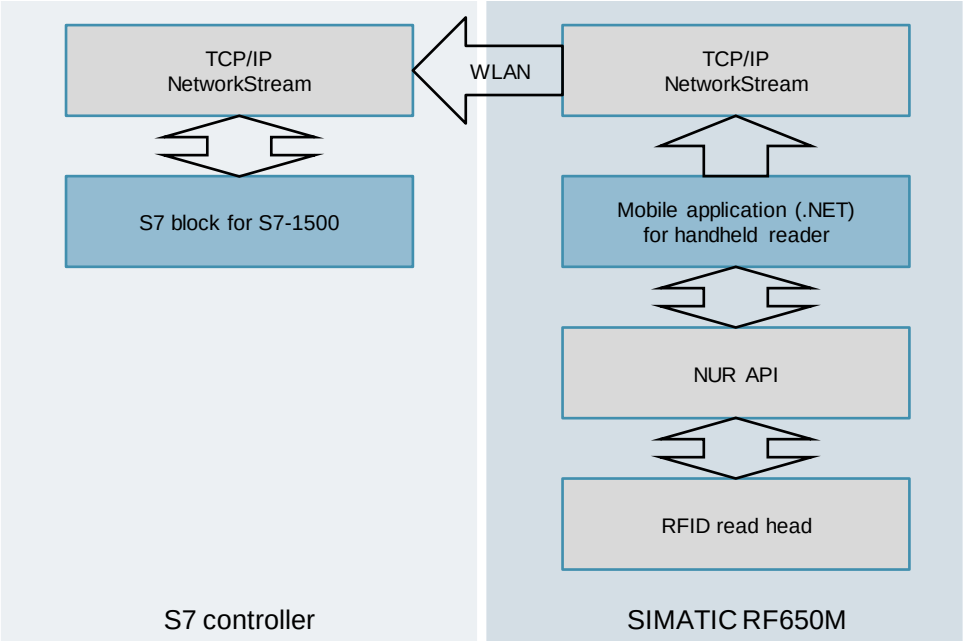
© Siemens AG 2016 All rights reserved

4 Principle of Operation

4.1 Complete overview

The following figure gives you 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 (C#). The mobile handheld reader with the Windows CE operating system communicates with the integrated RFID read head via the “NUR API” provided by Nordic ID. TCP/IP via WLAN is used as the interface to the associated S7 block.

Note In addition to the NUR API, this application example uses another API called “MHL”. This API does not contribute to any core functionality. It is only used as a simple means to generate a sound when transponders are read. The MHL API is not discussed in greater detail in this application example.

© Siemens AG 2016 All rights reserved

4.2 Functionality of the “RFIDRemote” mobile application

The mobile application for the RF650M handheld reader supports basic RFID functions. It reads transponder data and sends it to the S7 controller. The application interacts with the RFID read head via predefined members of the NUR API class. The user interface and communication with the S7 controller are implemented with .NET standard functions.

4.2.1 Structure of the mobile application

The mobile application for the handheld reader consists of five “regions”. Please see the following figure:

Figure 4-2

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

        //Form main()
        public RFIDForm()...

        //Misc region
        /// <summary>
        /// Miscellaneous event handles
        /// </summary>
        MiscRegion                            (2)

        //Inventory region
        /// <summary>
        /// Methods handling inventories
        /// </summary>
        InventoryRegion                       (3)

        //Settings region
        /// <summary>
        /// Methods handling reader settings
        /// </summary>
        SettingsRegion                        (4)

        //Transfer region
        /// <summary>
        /// Methods handling data transfer to PC application
        /// </summary>
        TransferRegion                        (5)
    }
}
```

© Siemens AG 2016 All rights reserved

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 event handling of NUR API ConnectedEvents
3.	Methods required to perform RFID inventories
4.	Methods for RFID and program settings the user can make in the application example
5.	Methods required for data transfer to the PC application

4.2.2 List of the most important methods

The following list provides an overview of the most important methods of the mobile application. For in-depth information, please see the comments in the supplied source files.

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

  Allows you to make the following settings:
  - 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 application and the PC application.
- **connectionWatchDogTimer_Tick**
  A timer starts as soon as a TCP connection has been established. At each timer tick, this method sends a 1-byte connection monitoring packet to the server.

© Siemens AG 2016 All rights reserved

- **reconnectTimer_Tick**
  If sending the WatchDogTimer packet fails, the connection to the server is interrupted. Another timer is started that uses the “connectTcp” method to initiate a new connection establishment process at each timer tick.
- **sendTagStorage**
  This method requests the TagStorage from the RFID module and sends it to the server via NetworkStream. The following NUR method is called in this method:
  - GetTagStorage()
- **synchronisebutton_Click**
  In manual mode, this button click method triggers the sending of the TagStorage to the server.
- **autoTransferCheckBox_CheckStateChanged**
  This check box check method starts / stops the automated TagStorage transfer to the server.

© Siemens AG 2016 All rights reserved

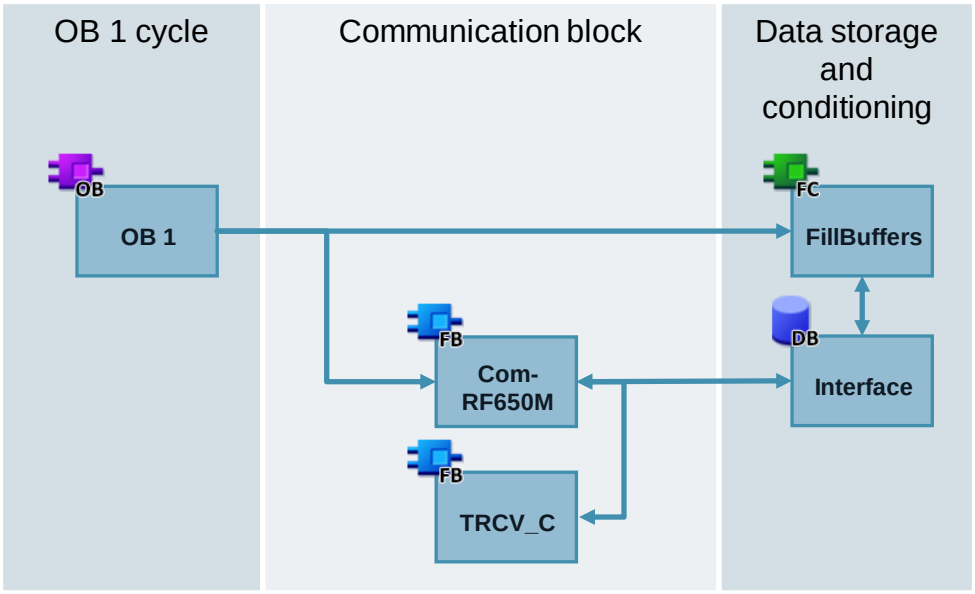
4.3 Principle of operation of the S7 program

The S7 controller receives read transponder data from the handheld reader's mobile application and stores it in byte arrays.

4.3.1 Program overview

The following figure shows the call hierarchy of the S7 user program of this programming aid:

Figure 4-3



The "ComRF650M" block is called in OB 1 of the user program. Internally, it uses the "TRCV_C" system block to implement the TCP server functionality and receive data from the mobile application.
The "FillBuffers" block copies the received EPC IDs and user data to two array structures and ensures that the data of the mobile application is synchronized with the data of the S7 controller.
The "Interface" data block contains all variables for the block interfaces of the user program. It additionally contains the array structures for the EPC IDs and user data.

© Siemens AG 2016 All rights reserved

4.3.2 Program details on the “ComRF650M” block

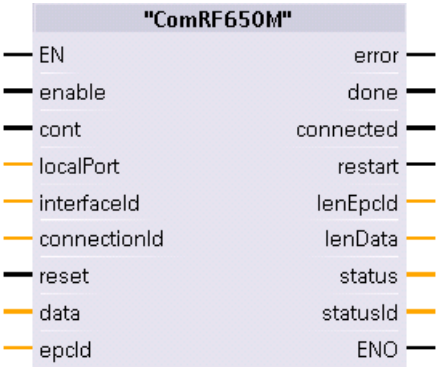
The “ComRF650M” block performs the server functionality of this application example. The integrated S7 block, “TRCV_C”, is responsible for connecting, receiving data and disconnecting.

The received data packets (see “4.4 Principle of operation of the data transfer”) are broken down, divided into EPC IDs and user data and provided to the user for one cycle via the “epcId” and “data” block interfaces. In addition, the lengths of the received EPC IDs and user data are provided via the “lenEpcId” and “lenData” block interfaces.

When another synchronization (first data record of the transponder list) between the mobile application and the S7 block is started, the “restart” output will be set for one cycle.

The following figure shows the “ComRF650M” S7 block:

Figure 4-4



Block interface

Input parameters:

Table 4-2

Name	Data type	Description
enable	Bool	Enable input for receiving data
cont	Bool	Enable input for connecting
localPort	UInt	Port address of the programmed connection
interfaceId	HW_ANY	Hardware identifier of the local Ethernet interface
connectionId	CONN_ OUC	Connection ID of the programmed connection (allowed values: W#16#0001 to W#16#0FFF)
reset	Bool	Optional: Reset block-internal buffers (when block behavior incorrect)
data	Variant	Pointer to the receive area for transponder data (maximum length: 182 bytes)
epcId	Variant	Pointer to the receive area for EPC IDs (maximum length: 128 bytes)

© Siemens AG 2016 All rights reserved

Output parameters:

Table 4-3

Name	Data type	Description
error	Bool	• 0: No error • 1: Error while connecting, receiving data or disconnecting
done	Bool	• 0: Job not started or still running • 1: Job completed successfully
connected	Bool	Connection to mobile application established
restart	Bool	Data synchronization restart (corresponds to the “ClearData” command of the data packets, see “4.4 Principle of operation of the data transfer”)
lenEpcId	Byte	Length of the received EPC IDs
lenData	Byte	Length of the received user data
status	Word	Job status
statusId	UInt	Error source of the status value

Status and status ID

The status shows the status of the job. The following user-defined status values exist:

Table 4-4

Status (W#16#...)	Description
7001	Data successfully received
7002	Data synchronization restart (only for automatic synchronization)
7006	Receiving data
8D12	Bad data received
8D13	Received data cannot be buffered
8D14	epcID parameter: Bad receive area
8D21	Length of the received data is outside the area
8D24	data parameter: Bad receive area
8D31	EPC ID length is 0
8D32	Length of the received EPC ID is outside the area

The status ID indicates the error source to see what the status values mean. The following error sources are evaluated.

Table 4-5

Status ID	Description
1	User-defined status value of the ComRF650M block
2	MOVE_BLK_VARIANT (EPC ID)
3	MOVE_BLK_VARIANT (data)
4	TRCV_C
5	MOVE_BLK_VARIANT (EPC ID initialization)
6	MOVE_BLK_VARIANT (data initialization)
7	MOVE_BLK_VARIANT (EPC ID reset)

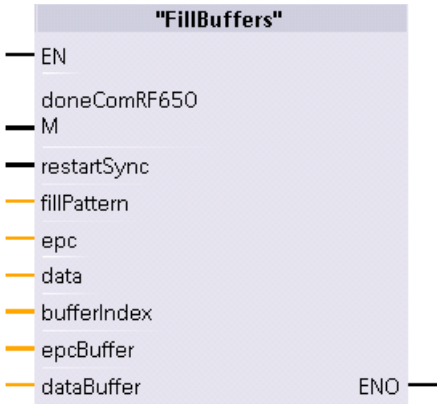
© Siemens AG 2016 All rights reserved

Status ID	Description
8	MOVE_BLK_VARIANT (data reset)

4.3.3 Data storage using the “FillBuffers” function

The “FillBuffers” FC reads the EPC IDs and user data from the receive area of the “ComRF650M” communication block and writes them to two-dimensional buffers of the “Interface” data block. This permanently saves the individual items of transponder data – that are in the “ComRF650M” receive buffers for only one cycle – until the next synchronization start.

Figure 4-5



Block interface

Input parameters

Table 4-6

Name	Data type	Description
doneComRF650M	Bool	Successful job of the “ComRF650M” block (ComRF650M interface: done)
restartSync	Bool	Data synchronization restart (ComRF650M interface: restart)
fillPattern	Any	Reference to a byte for initializing the data buffers (preferably 16#00) after a data synchronization restart
epc	Any	Receive area of the “ComRF650M” block (ComRF650M interface: epcId)
data	Any	Receive area of the “ComRF650M” block (ComRF650M interface: data)
bufferIndex	Int	Index variable for indexing the two-dimensional buffers
epcBuffer	Array[0..X,0..127] of Byte	Two-dimensional buffer for EPC IDs (the first dimension stands for the number of possible EPC IDs, the second one for the actual EPC IDs)
dataBuffer	Array[0..X,0..181] of Byte	Two-dimensional buffer for user data (the first dimension stands for the number of possible user data items, the second one for the actual user data)

© Siemens AG 2016 All rights reserved

4.4 Principle of operation of the data transfer

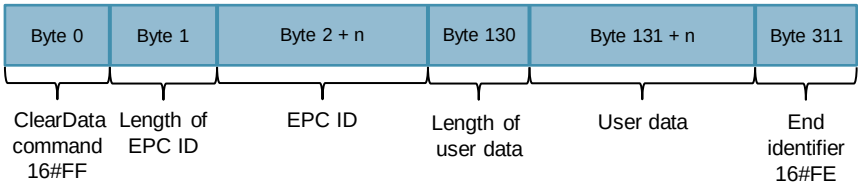
Data transfer between the two applications is always unidirectional and asynchronous from the mobile handheld reader to the S7 controller. The advanced TcpClient .NET class based on Systems.Net.Sockets and the “TRCV_C” S7 block are used as a software basis for TCP communication.

The data is transferred using data packets of a predefined size: Each read transponder unit (EPC + user data) forms a single data unit (corresponds to a data packet). If there are multiple read transponder units, they are sent one after the other as separate data units.

Structure of the data packets (byte arrays)

Read transponder data is read from the TagStorage of the NUR API and, on a serialized basis, written to a byte array with the following structure:

Figure 4-6



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

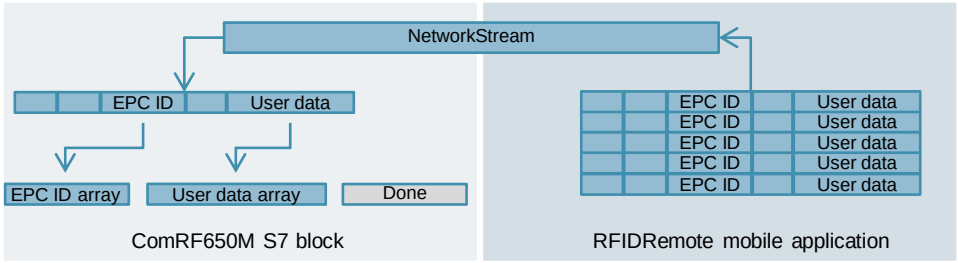
Note With the defined packet size of 312 bytes, a maximum of 180 bytes of user data can be transferred from the USER memory of the transponders. This ensures that all Siemens transponders can be used, except the RF622L and RF622T. The user bank of each of these two transponders has a user data size of 3224 bytes.

© Siemens AG 2016 All rights reserved

Transferring the data packet

When the data packet has been created by the mobile application, it is written to the NetworkStream of the TCP connection. The S7 block reads the data packets from this NetworkStream. Then the ECP IDs and the user data are available to the user in two byte arrays.

Figure 4-7

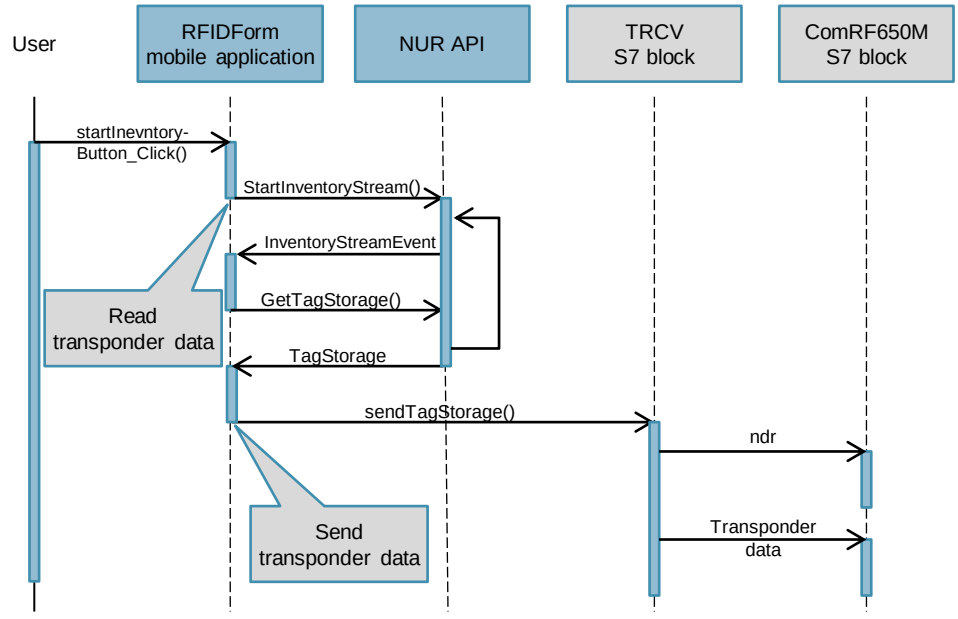


Note In addition to the data packets, the mobile application cyclically writes 1-byte data packets to the NetworkStream. This is for connection monitoring purposes.

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 4-8



© Siemens AG 2016 All rights reserved

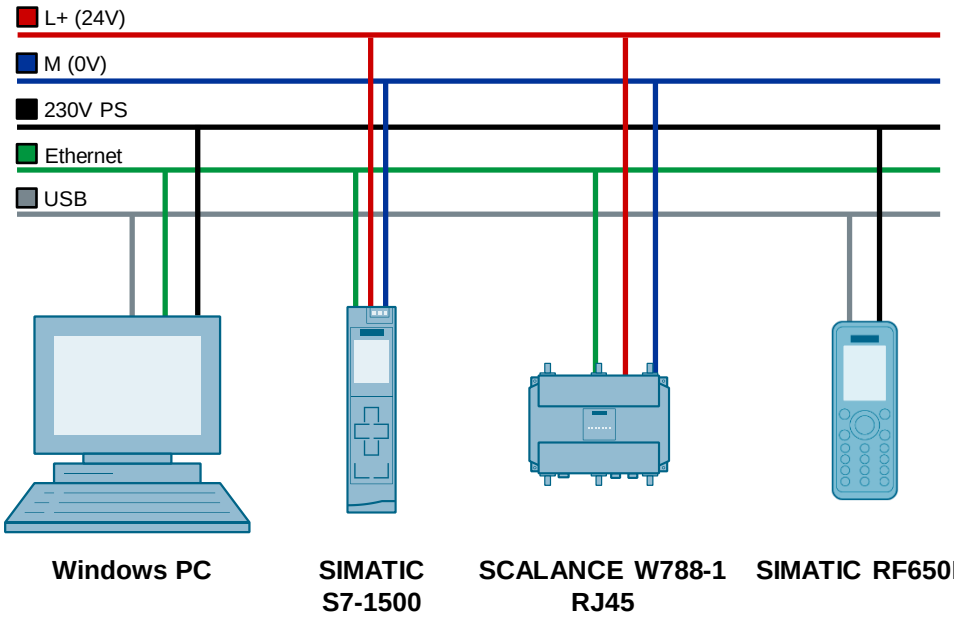
5 Installation and Startup

This chapter provides a more detailed explanation of the steps necessary to start up the application example.

5.1 Installing the hardware

The figure below shows the hardware configuration of the application.

Figure 5-1



Follow the below step-by-step instructions to install the hardware of the application example:

1. Connect the access point to a suitable power supply.
2. Connect the S7 CPU to a suitable power supply.
3. Connect your programmer / PC to the access point via Ethernet.
4. Connect the S7 CPU to the access point via Ethernet.
5. Connect the docking station of the RF650M to a suitable power supply.
6. Connect the docking station to the programmer / PC via USB.
7. Plug the RF650M into the docking station.

Note Always follow the installation guidelines and operating instructions for the SIMATIC S7-1500 \7\, the SIMATIC RF650M \5\ and the SCALANCE W788-1 RJ45 \6\.

© Siemens AG 2016 All rights reserved

5.2 Installing the software

This chapter describes the steps for installing the development environments in order to extend the application example or view the code.

1. Install Visual Studio 2005
2. Install Morphic SDK (by Nordic ID)
3. Install Windows Mobile Device Center
4. Install .NET Compact Framework 2 SP1
5. Install TIA Portal V13 SP1

5.3 Startup

This chapter describes the steps necessary to start up the application example.

5.3.1 Providing the mobile application on the SIMATIC RF650M

The following list explains step by step what you have to do to start the mobile application on the mobile handheld reader.

1. The executable application, “RFIDRemote”, is available on the HTML page \2\ from which you downloaded this document. Save the **109479885_RFIDRemotePLC300_CODE_V10.zip** archive to your hard drive.

2. The ZIP archive contains the executable file for the mobile application and the TIA project for the S7 CPU. Extract the ZIP archive.

3. Start Windows Mobile Device Center.

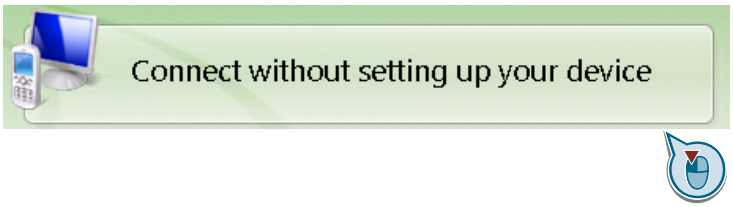
Note Windows Mobile Device Center opens automatically when you connect the SIMATIC RF650M to your PC using USB.

4. When your RF650M is in the docking station connected to the PC via USB, Windows Mobile Device Center shows that your PC is connected to the device (“Connected”).

   Figure 5-2

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

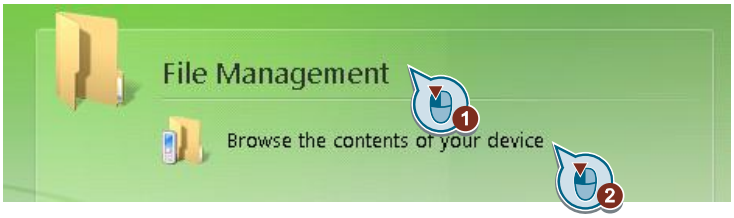
   Figure 5-3



© Siemens AG 2016 All rights reserved

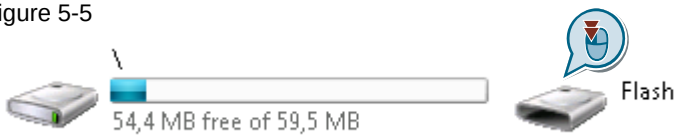
6. Now click “File Management” > “Browse the contents of your device”.

Figure 5-4



7. Then your explorer opens and you are in the handheld reader’s file system. Navigate to the “flash” memory of the device and double-click to open it.

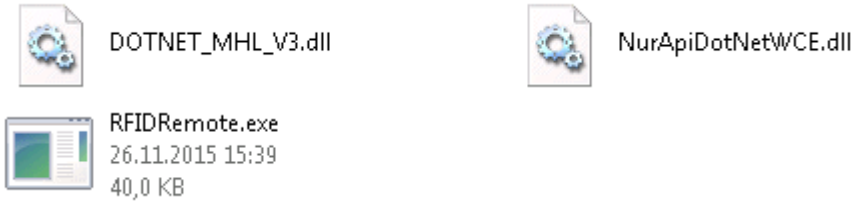
Figure 5-5



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

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

Figure 5-6



5.3.2 Connecting the SIMATIC RF650M to an access point via a WLAN

The following list explains step by step what you have to do to connect the mobile handheld reader to an access point via a WLAN.

1. Set up your access point. The mobile handheld reader supports a WLAN according to the IEEE 802.11 a/b/g/n standards. If you are using the access point used in this application example, the Operating Instructions, \6\, describe how to set it up.

Note The access point of this application example supports you in setting it up with a wizard. In order to follow the startup description of this chapter, make sure to assign a visible SSID to your WLAN. In addition, provide the network with a WPA2 key to ensure the security of your network. The mobile handheld reader communicates via the 2.4 GHz band.

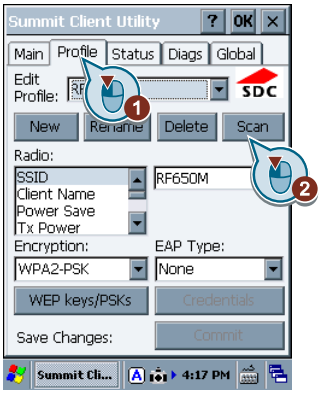
2. Remove the mobile handheld reader from the docking station and establish a WLAN connection to the access point. To do so, proceed as follows:

© Siemens AG 2016 All rights reserved

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

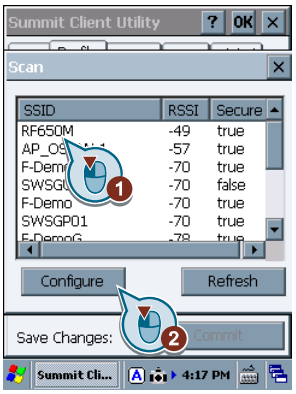
4. Go to the “Profile” tab and then click “Scan” to search for WLANs in range.

   Figure 5-7



5. Select the SSID you have configured in your access point and click “Configure”. In the following message box, select “Yes” to confirm.

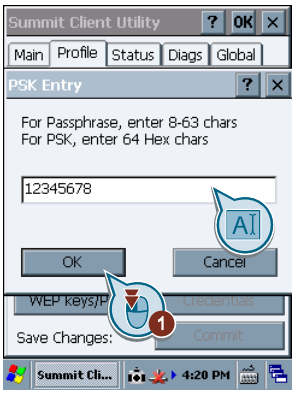
   Figure 5-8



© Siemens AG 2016 All rights reserved

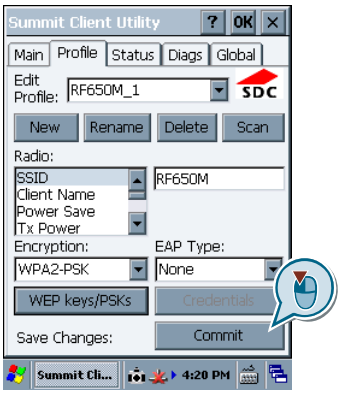
6. Enter the WLAN key you have defined and select “OK” to confirm.

Figure 5-9



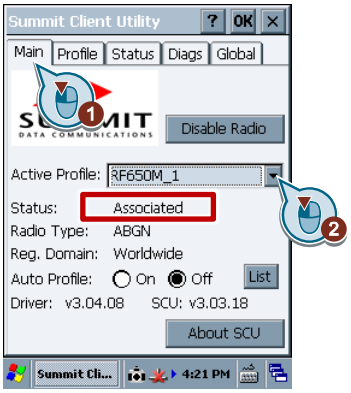
7. Select “Commit” to confirm.

Figure 5-10



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

Figure 5-11



© Siemens AG 2016 All rights reserved

5.3.3 TIA project

The following list explains step by step what you have to do to start the TIA project and download it to your S7 CPU.

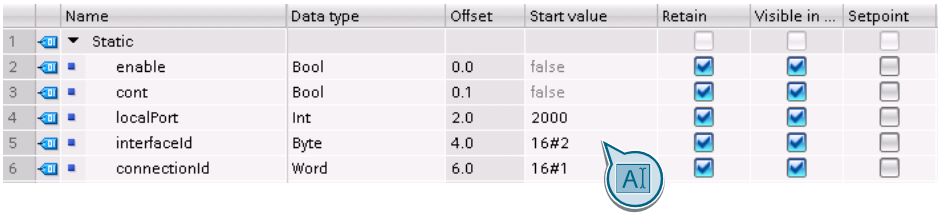
1. The TIA project, “RFIDRemotePLC1500”, is available on the HTML page \2\ from which you downloaded this document. Save the **109479885_RFIDRemotePLC1500_CODE_V10.zip** archive to your hard drive.

2. The ZIP archive contains the executable file for the mobile application and the TIA project for the S7 CPU. Extract the ZIP archive.

3. Open the “RFIDRemotePLC1500” folder contained in the extracted ZIP archive and start the TIA project included in it.

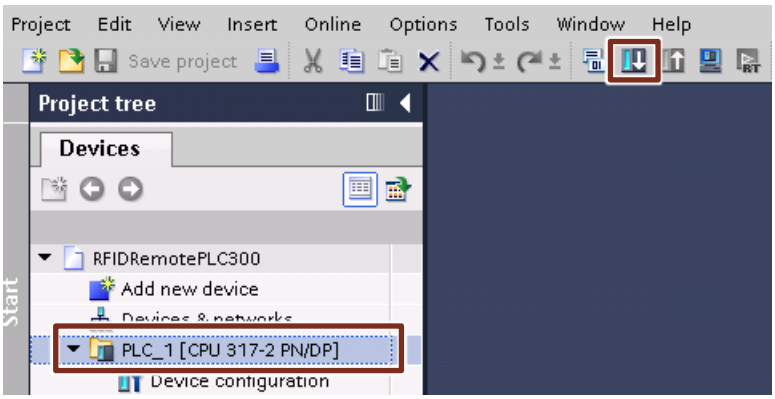
4. In the Project tree, navigate to the “PLC_1” > “Program blocks” folder and double-click to open the “Interface” data block.

5. In the “localPort” variable, enter the port address via which you want the ComRF650M block to accept a connection. In the “interfaceId” variable, enter the network interface to be used (see “4.3.2 Program details on the “ComRF650M” block”). In the “connectionId” variable, enter a free connection ID.
   Figure 5-12



	Name	Data type	Offset	Start value	Retain	Visible in ...	Setpoint
1	Static						
2	enable	Bool	0.0	false	☑	☑	☐
3	cont	Bool	0.1	false	☑	☑	☐
4	localPort	Int	2.0	2000	☑	☑	☐
5	interfaceId	Byte	4.0	16#2	☑	☑	☐
6	connectionId	Word	6.0	16#1	☑	☑	☐

6. In the Project tree, select the S7 CPU and download the project to the controller.
   Figure 5-13



© Siemens AG 2016 All rights reserved

6 Operation of the Application

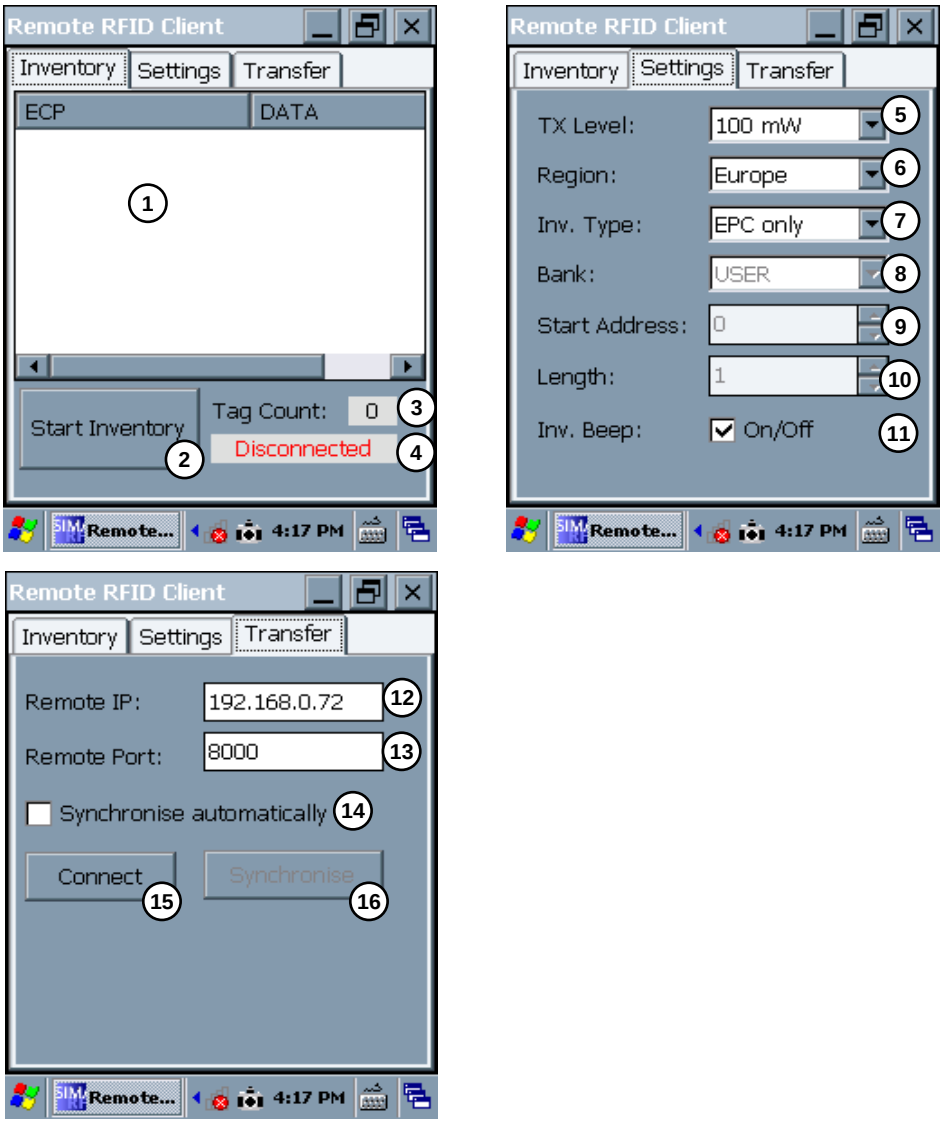
6.1 Overview and description of the user interface

The following chapter explains the user interfaces for the mobile application.

User interface for the mobile application

The user interface for the mobile application of this application example consists of 3 tabs. The first tab, “Inventory”, triggers reading the transponders and displays the read results.
The second tab, “Settings”, allows you to make various settings regarding the handheld reader’s RFID read head and the application.
The third tab, “Transfer”, provides connection settings for the S7 controller.

Figure 6-1



© Siemens AG 2016 All rights reserved

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

Table 6-1

No.	Description
1.	List of read results
2.	Button to start/stop RFID reads
3.	Number of read transponders
4.	Connection status to the remote PC application
5.	Read head's transmitter power
6.	Region (wireless profile)
7.	Inventory type (EPC; EPC+data)
8.	Select the 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 Note: More than 90 WORDs will not be transferred to the PC application.
11.	Check box to enable/disable a beep for read transponders
12.	IP address field for the remote PLC
13.	Port field for the remote PC application
14.	Check box for automatic data synchronization between mobile application and PC application
15.	Button to manually establish/terminate a connection to the remote PC application
16.	Button for manual data synchronization between mobile application and PC application

© Siemens AG 2016 All rights reserved

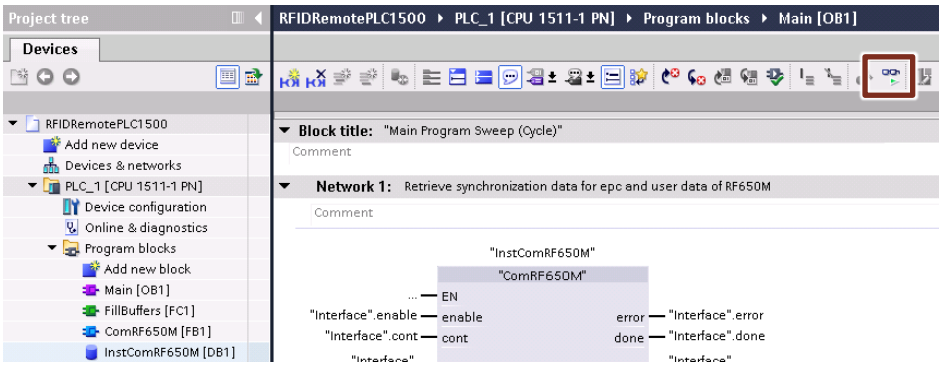
6.2 How to use the application example

The following table describes step by step how to use this application example.

1. In the Project tree of the TIA project, navigate to the “PLC_1” > “Program blocks” folder and double-click to open the “Main” organization block.

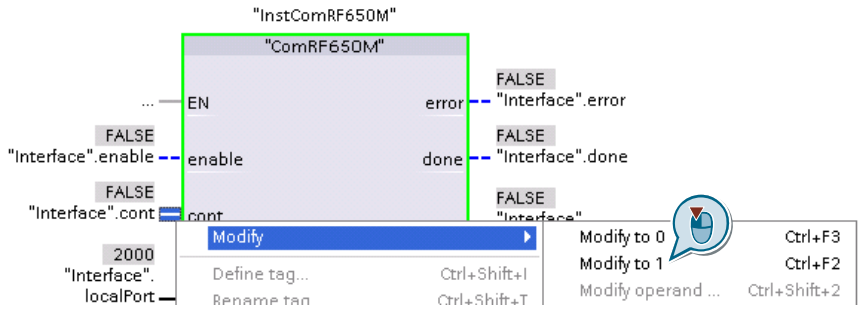
2. Click “Monitoring on/off” to connect the organization block online to the S7 controller.

   Figure 6-2

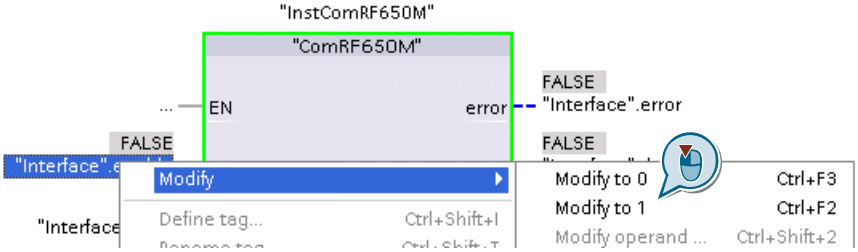


3. Right-click the “cont” parameter interface and in the context menu, navigate to “Modify”. Click “Modify to 1”. This starts the TCP server.

   Figure 6-3



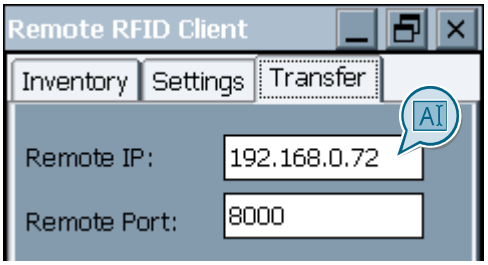
4. Right-click the “enable” parameter interface and in the context menu, navigate to “Modify”. Click “Modify to 1”. This starts ‘receive data’.
   Figure 6-4



© Siemens AG 2016 All rights reserved

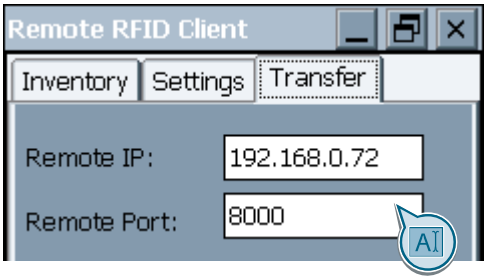
5. Now double-click the “RFIDRemote.exe” file copied in one of the previous steps to start the “RFIDRemote” application on the handheld reader. To find this file, double-click “My Device” from the handheld reader’s desktop and then navigate to the “Flash” folder. This is where you will find the “RFIDRemote” folder you have created. This folder contains the EXE file.
6. Go to the “Transfer” tab.
7. In the IP address field, enter the IP address of the S7 controller.

Figure 6-5



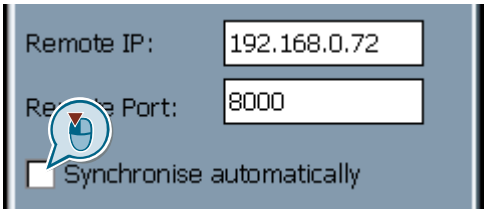
8. In the port field, enter the port of the S7 controller.

Figure 6-6



9. Check the “Synchronise automatically” check box. Now the mobile application automatically establishes a connection to the S7 controller. The read transponder data is automatically synchronized. In the event of a disconnect, the mobile application automatically attempts to establish a new connection.

Figure 6-7



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

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

© Siemens AG 2016 All rights reserved

11. The mobile application has successfully connected to the S7 controller when the connection status field displays “Connected”.

Figure 6-8

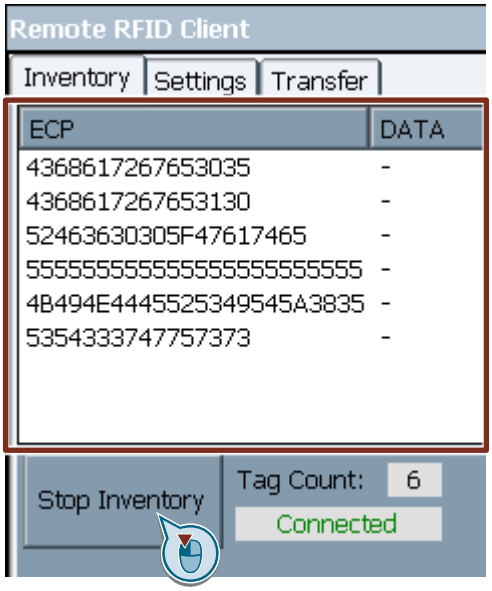
Note If the connection could not be established, check the connection information you have entered or check the WLAN connection between the handheld reader and the access point.

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

13. Go to the “Inventory” tab.

14. Use the “Start Inventory” button to start a read operation. As long as you do not use this button again, the handheld reader will continuously perform inventories. Each read transponder is displayed to you on the handheld reader and automatically transferred to the S7 controller.

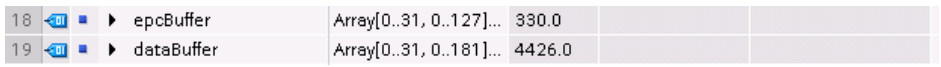
Figure 6-9



15. The read transponder data can be found in the “Interface” data block.

16. In the Project tree of the TIA project, navigate to the “PLC_1” > “Program blocks” folder and double-click to open the “Interface” data block.

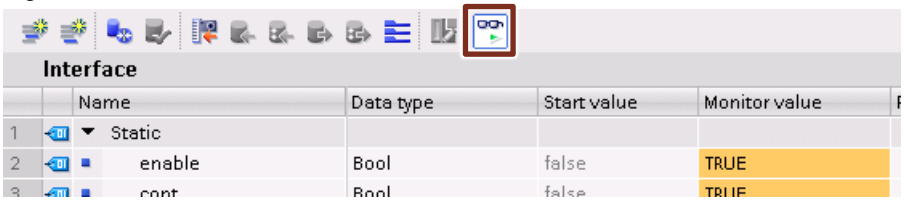
Figure 6-10



© Siemens AG 2016 All rights reserved

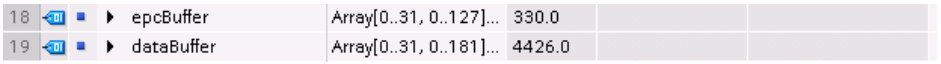
17. Click “Monitoring on/off” to connect the data block live to the S7 controller.

Figure 6-11



18. In the “epcBuffer” array structure, you will find the transferred EPC IDs. In the “dataBuffer” array structure, you will find the transferred user data. In this application example, the arrays hold 32 data records.

Figure 6-12



© Siemens AG 2016 All rights reserved

7 Links & Literature

Table 7-1

	Topic
\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\	Mobile reader manufacturer website 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
\7\	SIMATIC S7-1500, ET 200MP Automation System https://support.industry.siemens.com/cs/ww/en/view/59191792

8 History

Table 8-1

Version	Date	Modifications
V1.0	06/2016	First version

© Siemens AG 2016 All rights reserved