ISSN 0280-5316
ISRN LUTFD2/TFRT--5776--SE

Radio Frequency Identification in Library

Emil Svärdh

Department of Automatic Control
Lund University
October 2006

Department of Automatic Control
Lund Institute of Technology
Box 118
SE-221 00 Lund Sweden

Document name
MASTER THESIS

Date of issue
October 2006

Document Number
ISRNLUTFD2/TFRT--5776--SE

Author(s)
Emil Svärdh

Supervisor
Patric Englund at Pöyry Forest Industry AB in Kristianstad.
Karl-Erik Årzén at Automatic Control in Lund

Sponsoring organization

Title and subtitle
Radio Frequency Identification in Library (RFID)
(Radio Frekvens Identifiering i bibliotek)

Abstract
In libraries today there is a need to faster and more efficient loan books without the need of clerks. Radio Frequency Identification is used to solve this problem. To make everything automatic and still safe a special system of doors will be used scanning all the angles. A light grid is used to determine when a person wants to leave the library. A connection to the libraries database is established enabling the system to determine whether the items are allowed to check out.

Keywords

Classification system and/or index terms (if any)

Supplementary bibliographical information

ISSN and key title
0280-5316

ISBN

Language
English

Number of pages
77

Recipient's notes

Security classification

The report may be ordered from the Department of Automatic Control or borrowed through:University Library, Box 3, SE-221 00 Lund, Sweden Fax +46 46 222 42 43

Preface

This master thesis has been conducted at Pöyry Forest Industry AB located in Kristianstad, Sweden. The thesis is one of the requirements for the Master of Science in Computer Science degree at Lund Institute of Technology (LTH) located in Lund. The quantity of the work is 20 weeks.

I would like to thank my supervisor at Pöyry Forest Industry AB, Patric Englund as well as Professor Karl-Erik Årzén at the Department of Automatic Control, LTH for all support and help during the project.

Emil Svärdh

Table of contents

1. Introduction

1.1 Problem description

This chapter describes the background and purpose of this master thesis.

1.1.1 Background

In Sweden the lending of books at libraries increases rapidly, this means an increased work effort for the existing personnel. Often there is not enough money to increase the staff in order with the increased lending.

A possible solution to this problem would be an automatic solution based on RFID technology in order to reduce the work for the staff. RFID is well known in the supply chain and could therefore be very applicable also in the library industry.

For example:

> A person picks up two books, puts them in a bag and heads for the exit. At the exit there are RFID antennas required for the person to pass and the following scenario occurs:
>
> 1. The system checks if the person has a valid library card, if not the person will not be able to lend the books.
> 2. If the library card is valid the books shall be registered as loaned by the person
> 3. When the loan has passed the doors will automatically be opened allowing the person to leave the library.

The most important feature in the software is that the user shall be able to choose the desired protocol for the communication between the client and the database.

1.1.2 Purpose

The purpose of this master thesis is to create a PC-based software solving the problem as well as choosing appropriate hardware and configure it to satisfy the requirements. The master thesis shall determine how to incorporate RFID technology into the library industry.

1.2 Pöyry Forest Industry AB

This master thesis has been conducted at Pöyry Forest Industry AB, a global engineering company with approximately 5600 employees in 42 different countries.

In Sweden there are 250 engineers divided between 14 offices ranging from Kiruna in the north to Lund in the south. The company has been active in Sweden since 1962 and offers customers services within processes, mechanics, construction, automation and industrial IT, electronics and instrument. Industries like the paper industry, packaging industry, provision industry, chemistry, energy and the engineering industry are common customers of the services.

In the summer of 2005 a company called Scancontrol was acquired with its 55 engineers specialized within automation. The automation section is the largest part of the company in the south of Sweden where Pöyry has offices in Helsingborg, Lund and Kristianstad.

Some of the companies Pöyry Forest Industry has worked with are:

- Ceba Foods Oatly, Landskrona
- Tetra Pak, Lund
- Frigoscandia distribution, Helsingborg
- Amcor Flexibles, Lund
- The Absolut Company, Åhus
- Baldwin Jimek AB, Arlöv
- Haldex, Shanghai, China

1.3 Outline

Chapter 2 discusses the background of RFID. It also discusses the components needed for a functional system as well as possible problems with the technology.

Chapter 3 discusses the Electronic Product Code, how to extract data from it and how it connected to the RFID technology.

Chapter 4 discusses the protocol used for communicating with the library database and the protocol used for communicating with the RFID interrogator. In this case the protocols are SIP2 [6] for the library database and STX / ETX for the interrogator.

Chapter 5 discusses the choice of hardware components to make a complete unit. The different components and its features are briefly discussed.

Chapter 6 discusses the software developed in this thesis. The software is used for synchronization between the hardware and the library database. Unified Modelling Language (UML) diagrams and a flowchart is presented as well as technical documentation of the system.

Chapter 7 discusses how the software and the hardware are interconnected. It also handles how to connect the antennas, interrogator and specification of for example the serial cable.

Chapter 8 discusses different problems encountered and how to solve them.

Chapter 9 discusses the complete system and draw conclusions of how the system works as well as future work to be done.

Chapter 10 presents all the references used for this thesis.

2. RFID

This master thesis is based on the technology Radio Frequency Identification (RFID) and how its features can be applied to the library industry. To explain RFID easy one might say identification of an item using radio waves.

2.1 Background

In 1935 a Scottish physicist named Sir Robert Alexander Watson-Watt discovered the radar which made it possible for airports to know when airplanes approached miles ahead [1]. This technology was widely used during world war two but the problem here was to identify whether the approaching airplane was hostile or friendly.

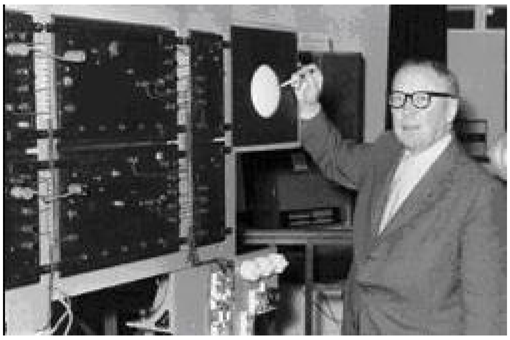


Figure 1 - Watson with the first radar [1]

The German forces solved this problem by letting the pilots roll their planes when returning to the air field, since this altered the responding radio signal and therefore the air field could determine whether the airplane was hostile or friendly. This was the first passive RFID system developed [1]. The first active RFID tag was developed at the same time by the British forces under the supervision of Watson-Watt, the system was called “Identification, Friend or Foe”. To do this they put a transmitter on every British airplane. When the transmitter received signals from the radar station on the ground, i.e. the air field, it began to broadcast a signal to notify the air field that the plane was friendly [1].

The development of RFID continued and during the 1960’s scientists from the United States, Japan and Europe presented papers on how to use RFID to remotely identify items. The commercial use of RFID began when companies launched so called anti-theft system which used radio waves. This system used a one bit tag which was set to zero if the item was paid for and set to one if it were not [1]. Then when leaving the store an alarm would sound if the bit was set to one which made it possible to determine whether the item was purchased or stolen.

Scientist soon discovered that RFID allowed much more features than just anti-theft equipment and in 1973 an entrepreneur named Charles Walton received a patent for the first passive transponder that was used to unlock a door not using a key. The technology used a card with an embedded transponder which responded to a reader placed near the door with a serial number and if the serial number matched the expected the door unlocked itself. Unfortunately this technology was very expensive and it was not until 1999 when the Auto-Id Center developed a cheap passive tag that the commercial used bloomed. The tag only consisted of a serial number and then data about the specific serial number was stored in a database which would be accessible over the Internet [1].

2.2 The technology

A typical RFID system includes the following components:

- Tags
- Antennas
- Interrogator

The components are connected together in the following way when using passive tags:

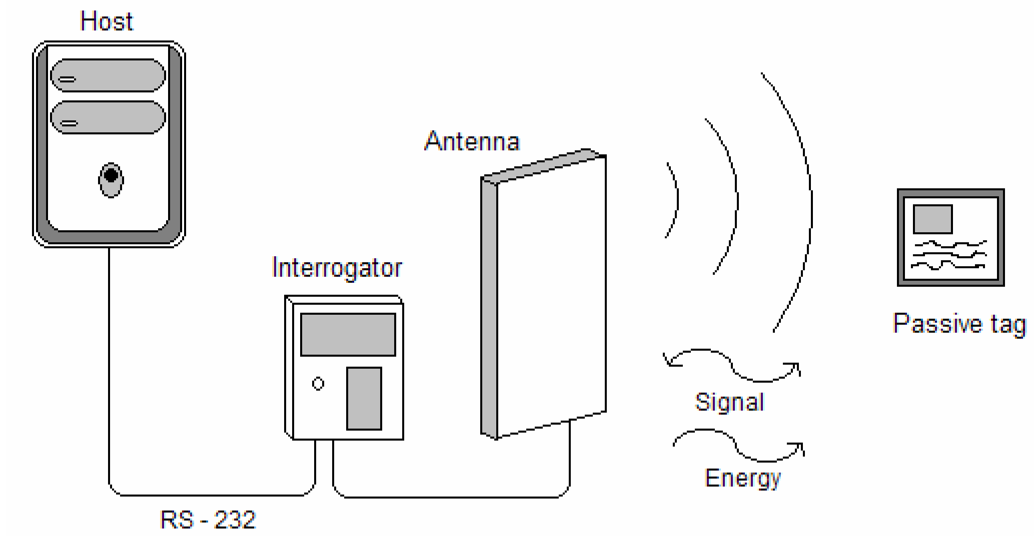


Figure 2 - Typical RFID system

As seen in figure 2 the interface between the interrogator and the host computer is RS-232, i.e. a serial cable. There are also interrogators available on the market using an Ethernet interface communicating with the host.

2.2.1 The tag

The tag is placed on the item wanted to identify, for example incorporated into a product, animal or a person. It consists of an antenna and a microchip with information stored about the tag, see figure 3. There are three types of tags; passive, semi-passive and active.

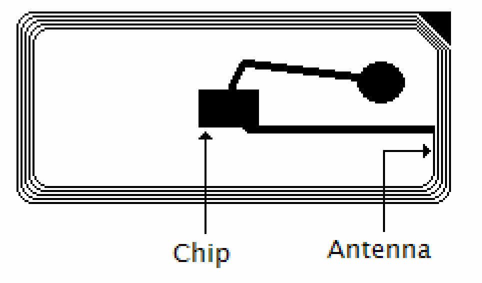


Figure 3 - RFID tag

The chip is very tiny and contains a unique serial number normally EPC standard [1], explained later in this thesis. The chip is constructed so that it, in front of an interrogator responds, with its unique id.

The passive tag is powered up by the magnetic field from the antennas and receives its energy from inductance [4]. The magnetic field provides enough energy for the tag to send a response to the interrogator, usually achieved by backscattering the signal from the interrogator. Backscattering means that the tag reflects the energy when responding, i.e. not creating any radio frequencies of its own [2].

The tags backscatter their data with three different methods depending on the choice of tag:

1. Amplitude Shift Key Modulation (ASK)
2. Phase Shift Key Modulation (PSK)
3. Frequency Shift Key Modulation (FSK)

Amplitude Shift Key Modulation means that the carrier frequency change amplitude depending on if it is a logic 0 or a logic 1. A typical output from an ASK modulation is shown in figure 4.

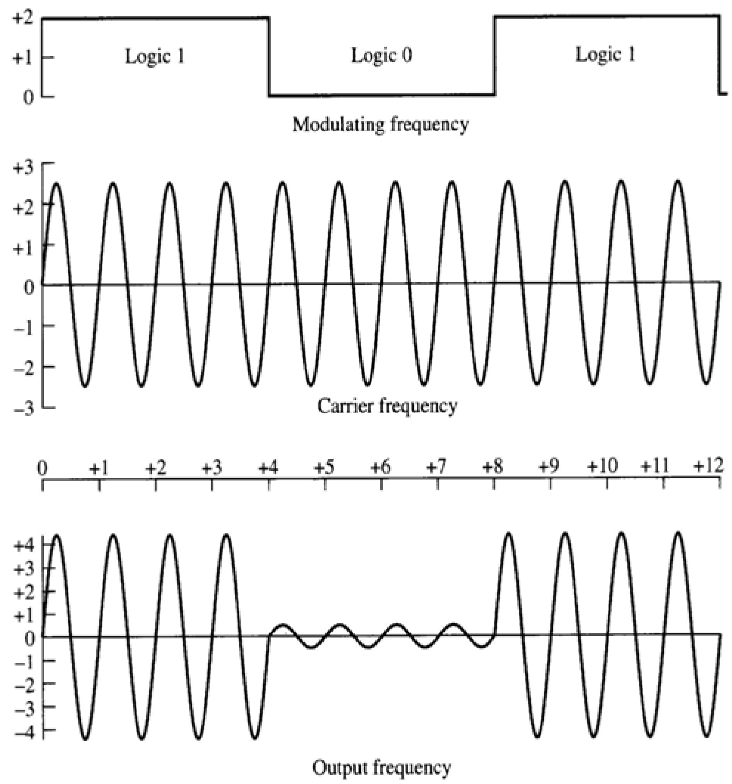


Figure 4 - Amplitude Shift Key Modulation [24]

Phase Shift Key Modulation means that the carrier frequency changes phase depending on the logic input. There different types of modulation methods, for example:

- Two-phase (2 PSK)
- Four-phase (4 PSK)
- Eight-phase (8 PSK)
- Sixteen-phase (16 PSK)

The modulators are so called Binary Phase Shift Key Modulators (BPSK). In figure 5 a two-phase modulations is shown where a logic 0 shifts the phase 180 degrees while a logic 1 does not change the carrier frequency phase.

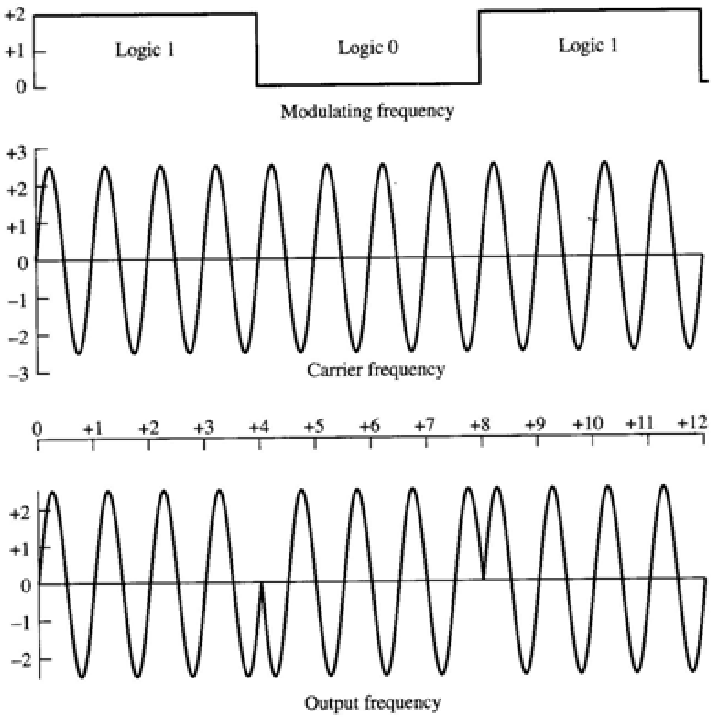


Figure 5 - Phase Shift Key Modulation [24]

Frequency Shift Key Modulation means that the carrier frequency is altered depending on if it is a logic 0 or a logic 1. A typical output from a FSK can be viewed in figure 6.

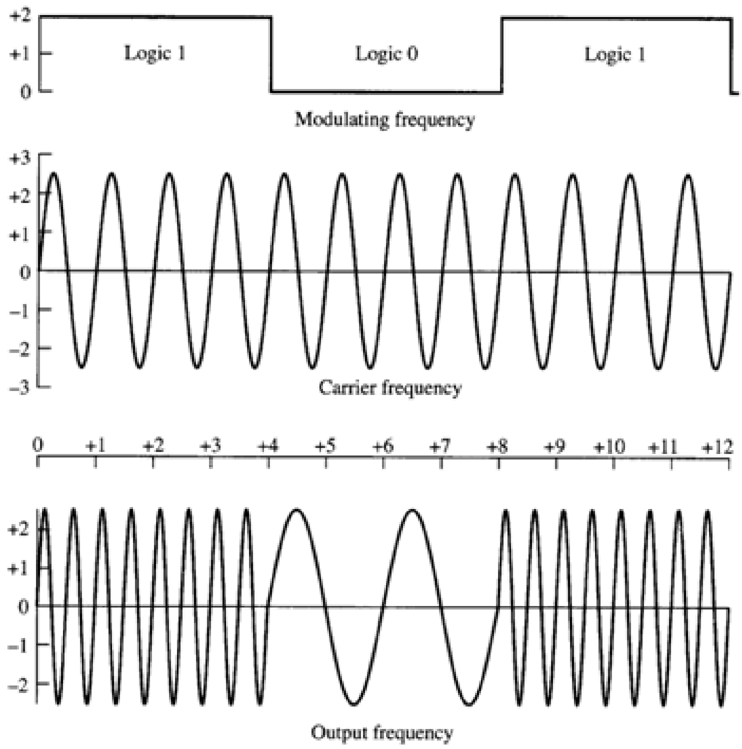


Figure 6 - Frequency Shift Key Modulation [24]

Since a passive tag does not require a battery it offers unlimited lifetime as well as being very small and thin [2]. The passive tag can store up to 64 kilobyte of data. These features are very applicable in library books as well as in library cards.

The semi-passive tag includes a small battery which is used to supply power for the tag. This makes the tag constantly powered up, minimizing the response time to the interrogator. Lifetime for a semi-passive tag is between 5 and 10 years [2].

The active tag includes a battery which is used to send replies to the interrogator. This tag is very useful in for example containers which are spread over a large area. This tag is the most reliable since it does not depend on the magnetic field from the antennas at all. Additionally it can store up to 8 megabyte of data [2][4].

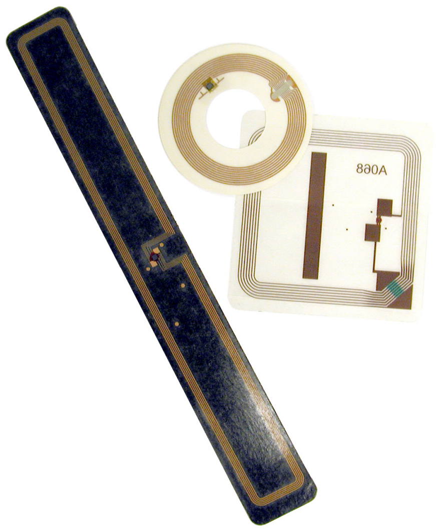


Figure 7 - Tags used in the library industry [13]

The data stored on the tags are binary and transmit there data like analog waves which then is interpreted by the interrogator [4]. The tag could either be pre-programmed by the factory or it can be programmed using the radio signals [3]. The programmable tags are very useful in application where one might want to change the information during time, for example in a supply chain. The pre-programmed tags are read only; this makes them useful in for example the library industry where all one want to know is the specific bar code.

2.2.2 The antennas

The antennas connected to the interrogator create an electromagnetic field, see figure 8, which is used to supply power for the semi-passive and the passive tag [4].

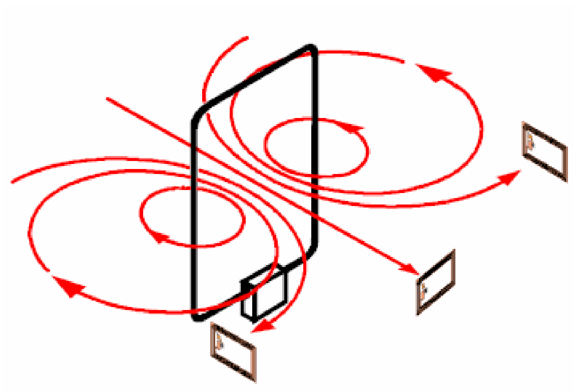


Figure 8 - Antenna magnetic field [23]

The reading distance of the antennas is limited by the antennas signal strength as well as the tags features. When installing the antennas one has to adjust the signal from the antenna so that other antennas will not be disturbed. Maximum distance using RFID is about 100 meters when using an active tag and the sight is free [4].

2.2.3 The interrogator

The interrogator is also called the reader. Antennas are connected to the interrogator and transmit a signal to all tags within reading distance, the signal sent from the tags in the area are then interpreted by the interrogator which determines the information contained in the tags [3]. The signal from the interrogator is transmitted several times per seconds which allows several reads every second.

The radio signal from the antennas use different frequency spans. The frequency is decided by the specific application and should depend on the components and the requirements of the application. There are three major categories of frequencies:

1. (LF) Low Frequency (< 135 KHz)
2. (HF) High Frequency (13.56 MHz)
3. (UHF) Ultra-High Frequency (868 – 915 MHz)

Low frequency interrogators are very cheap and have the ability to penetrate objects between the antenna and the tag. The negative side is that the reading distance is very small and the reading time is very long [5].

High frequency interrogators have the advantage of fast reading and long reading distance but the sight has to be fairly free [3]. As well as the low frequency interrogators they have a good penetration through metal and liquids [25].

Ultra-High frequency interrogators have a long reading distance. They are also fast and allow several reads every second. The interrogators are kind of cheap. The penetration of liquids are worse than interrogators operating at lower frequency spans.

2.3 Problems with RFID

In this chapter some of the identified problems with RFID will be presented.

2.3.1 Data privacy

When buying something in a store containing a bar code the product is virtually dead when leaving the store, i.e. no one can get information about the buying person. When instead buying an item containing a tag the data about the item will be broadcasted even after leaving the store, allowing anyone to pick up its data. The reason this is concerned to be a privacy issue is that a product can be tracked after the sale to the new owner [11].

There are in fact some possible solutions to this problem, either one disables the tag after the purchase or the tag could be a tear-away label [11].

2.3.2 Large amounts of data

Since RFID combined with EPC monitors every item independently there has to be some sort of algorithm that filters the data in order to just keep the essential. The unessential data also has to be deleted on a regular basis to keep the amount to a manageable level [11].

2.3.3 Standardization

The lack of a standard data form for RFID is one of the major issues since the cost of the implementation increases when each retailer use different formats. Because of the difference between all retailers format and the costs associated with RFID technology a company with a really good standard probably will not share it with the other companies since they then would loose their competitive advantage [11].

There are in fact industry organizations trying to coordinate the RFID technology into a standard, these are for example AutoID Labs and the Association for Automatic Identification and Mobility [11].

2.3.4 Issues with RFID

Some RFID tags emit Ultra High Frequency (UHF) waves, which operates in the 900 MHz span [12]. The UHF waves are absorbed by liquids, such as water, as well as being reflected by metals. This leads to faulty tag reads or even the tag being unscannable [11][12].

Solutions to this problem have been found and tested but still there is a five percent probability that the scan will be bad or the tag stays unscanned [11].

This is not a problem in this project since only high frequency tags are used.

2.3.5 Supplier retaliation

The RFID technology comes with strict implementation schedules causing numerous suppliers refusing to follow these because of the costs associated with them. A solution might be that the Department of Defence and Wal-Mart use RFID which will accelerate the commercial growth of RFID leading to cheaper hardware and therefore a possibility for the suppliers to switch to this technology [11].

3. EPC – Electronic Product Code

The Electronic Product Code (EPC) is the next generation of bar codes used for product identification [8]. EPC was developed by the Auto-ID Centre at the Massachusetts Institute of Technology [9].

EPC is designed in order for every desired item to get a unique product ID in the supply chain regardless of the items specification. The EPC standard is to divide numbers making it possible to find the identity of the manufacturer and the product type. It also has the ability to uniquely identify 268 million manufacturers with more than one million products each [10]. A typical EPC number has 96 bits and consist of the following [9]:

1. Header, information about length, type, structure, version as well as the generation of EPC.
2. Manager Number, the company's identity.
3. Object Class.
4. Serial Number, identifies the specific item of the object class.

Figure 9 - Description of the EPC numbering [8]

3.1 EPC and RFID

Every desired item in an RFID system is marked with a tag, the tag contains data about the object in this case the EPC for the specific item, see figure 10. This makes it possible to track all items sold with a tag containing the specific EPC number.

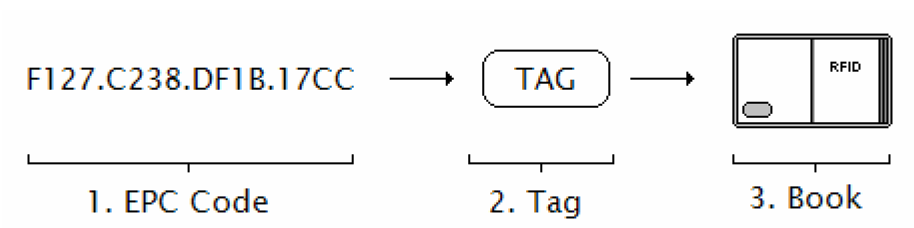


Figure 10 - Marking item with RFID and EPC

When a tag is scanned containing an EPC number the code will be sent over the network to an Object Name Server (ONS), the ONS then notifies where to find information about the item. After knowing where to find the information one is able to get the information using a language called Product Markup Language (PML) which is a standard developed to describe the item, see figure 11.

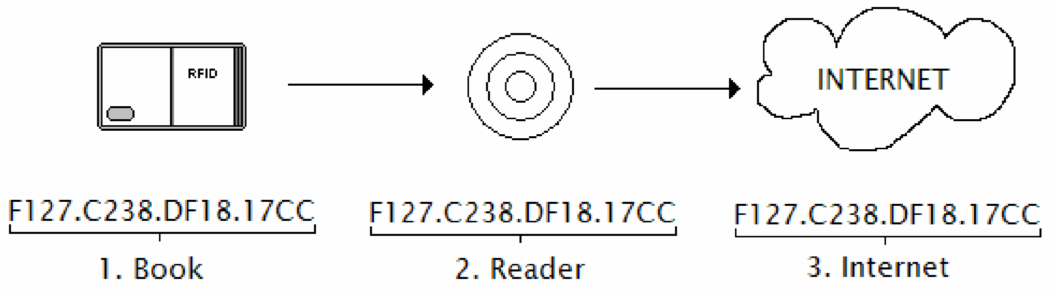


Figure 11 - Lookup EPC over the internet

4. Protocol

In this chapter the protocol used for communication with the main library database as well as with the RFID Interrogator is discussed. The protocol for the main database is 3M™ Standard Interchange Protocol V2.00 and for the interrogator STX/ETX.

4.1 3M™ Standard Interchange Protocol V2.00

The SIP2 protocol is a protocol where all the communication between the client and the server is initiated by the client. For every message the client sends to the server the server will respond with one message [6].

The communication messages are divided into pairs. For example if one wants to checkout a book the client sends a checkout message to the server and the server responds with a checkout response message. The message pairs are completely independent of each other so in the previously mentioned checkout message all the information will be passed on to perform the checkout [6].

To make sure that all messages sent between the client and the server are received correctly the SIP2 protocol incorporates a checksum as well as sequence numbers. If an error occurs there are predefined messages to ask either the server to resend or be asked by the server to resend the last message [6].

The connection between the client and the server using the SIP2 protocol is established using sockets over the TCP/IP protocol [6]. One has to establish a socket connection and the log on as described under “Scenario – Start-up”.

4.1.1 Message structure

The message always starts with a 2 digit number notifying the server or the client what type of message is coming, for example 99 for the “SC Status” message. After that different data follow depending on the type of message. At the end of each message there is always a sequence number, a checksum and finally “<CR>”.

An “SC Status” message has the following look:

99<status code><max print width><protocol version>

For status ‘0’, Print width ‘40’ and Protocol version 2.0 it becomes:

990<space>402.00

Then one has to add the sequence number and the calculated checksum:

AY<sequence number>AZ<four digit checksum><carriage return>

Assume sequence number 1 the entire message becomes:

990<space>402.00AY1AZFCB4<CR>

The checksum is calculated the following way:

ASCII Character	**Decimal Value**	
9	57	
9	57	
0	48	
<space>	32	
4	52	
0	48	
2	50	
.	46	
0	48	
0	48	
A	65	
Y	89	
1	49	
A	65	
Z	90	
	844	Decimal = 034C Hexadecimal

0000 0011 0100 1100 is the Binary equivalent of 0x034C

Then one has to invert all bits and add 1:

1111 1100 1011 0011 (All bits inverted)
+ 1
1111 1100 1011 0100 Binary checksum = FCB4 Hexadecimal checksum

4.1.2 Scenario – Start-up

The following sequence of messages, see figure 12, is used for the initiation of communication between the client and the server.

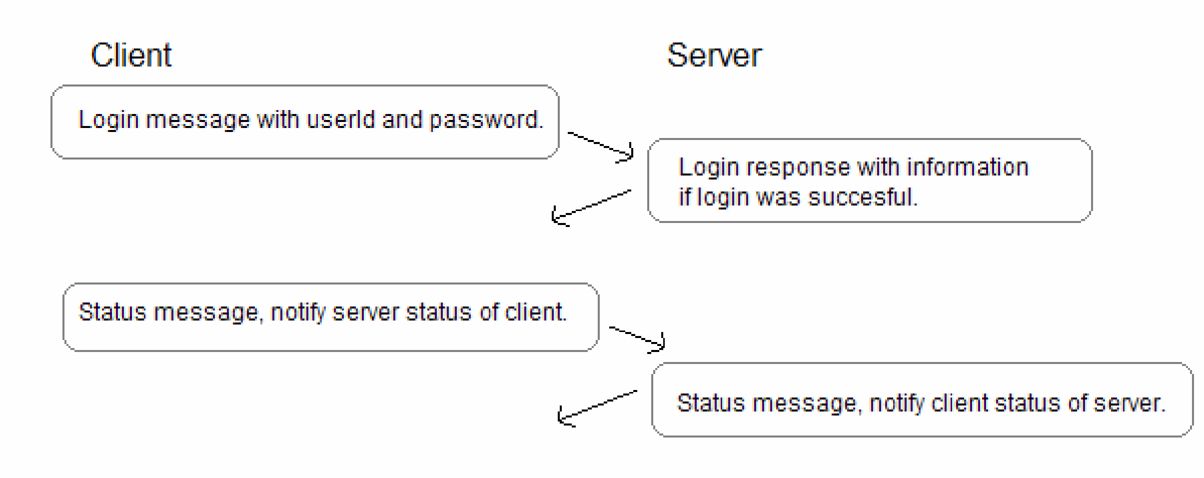


Figure 12 - Start-up scenario

4.1.3 Scenario – Item checkout

The following sequence of messages, see figure 13, is used when checking out an item from the library, i.e. the sequence used in this project.

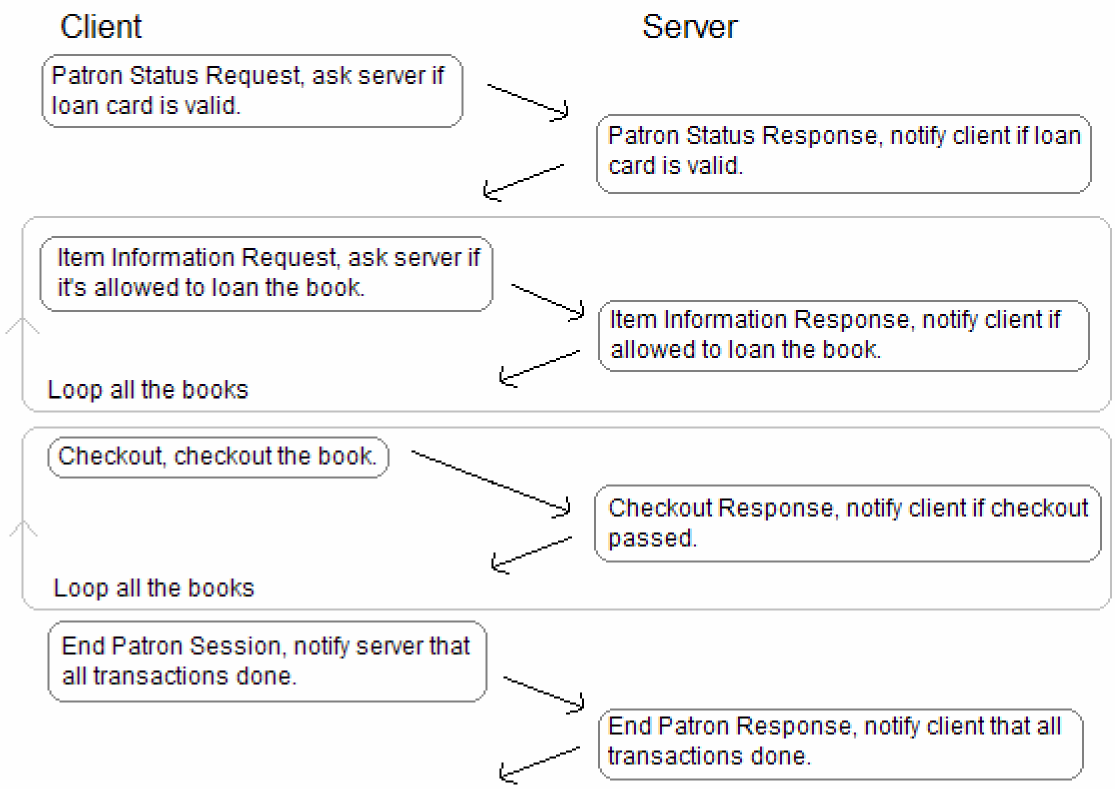


Figure 13 - Checkout scenario

As seen all the books are looped through and if any of them fail the item information the current session is aborted without any checkout, i.e. the lending sequence failed.

4.1.4 Scenario – Shutdown

The following sequence of messages, see figure 14, is used when shutting down the communication between the server and the client.

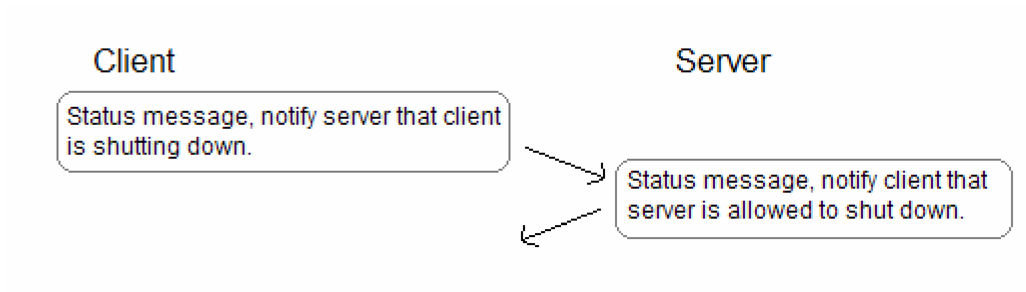


Figure 14 - Shutdown scenario

4.2 STX / ETX

The STX/ETX protocol is a standard protocol that uses STX characters in the beginning of each message and ETX characters at the end of each message. All characters sent between the client and the interrogator are transformed into hexadecimal character strings [7].

The principle here is the same as for the SIP2 protocol, i.e. the client sends a message to the interrogator and then the interrogator answers with one message. The main difference is that the interrogators response starts with ACK for acknowledge, NAK for no acknowledge or SYN for error. If an error occurs the interrogator also include a fault number in the error message allowing the client to determine whether it is a vital error or not [7].

To make sure all messages between the client and the interrogator are correct the protocol also includes a checksum for error detection [7].

Assume that one want to send the following string to the interrogator:

STX 1018 ETX

The sent string would have the following format:

0x02 “1018” 0x03 0x09

Where
- 0x02 stands for STX
- 1018 stands for 1018 (in this case the message “Flush Buffers”)
- 0x03 stands for ETX
- 0x09 is the calculated checksum

The checksum is calculated using an XOR-gate the following way:

Initial checksum	0000 0000
STX (0x02)	0000 0010
	0000 0010
1 (0x31)	0011 0001
	0011 0011
0 (0x30)	0011 0000
	0000 0011
1 (0x31)	0011 0001
	0011 0010
8 (0x38)	0011 1000
	0000 1010
ETX (0x03)	0000 0011
	0000 1001 = Binary checksum = 0x09 (Hexadecimal)

4.2.1 Scenario – Scanning for tags

The following sequence of messages, see figure 15, is sent between the client and the interrogator while scanning for tags and their data.

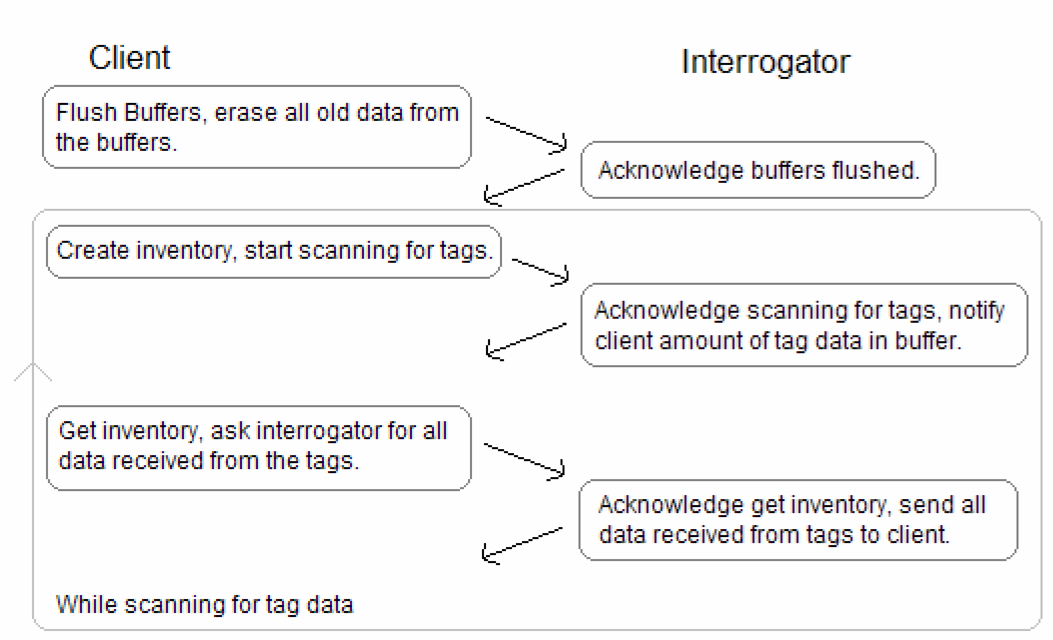


Figure 15 – Scanning for tag data scenario

5. Hardware

This chapter consists of a presentation of the hardware components used in the project as well as how they are interconnected.

5.1 Basic structure

The complete system has the following configuration:

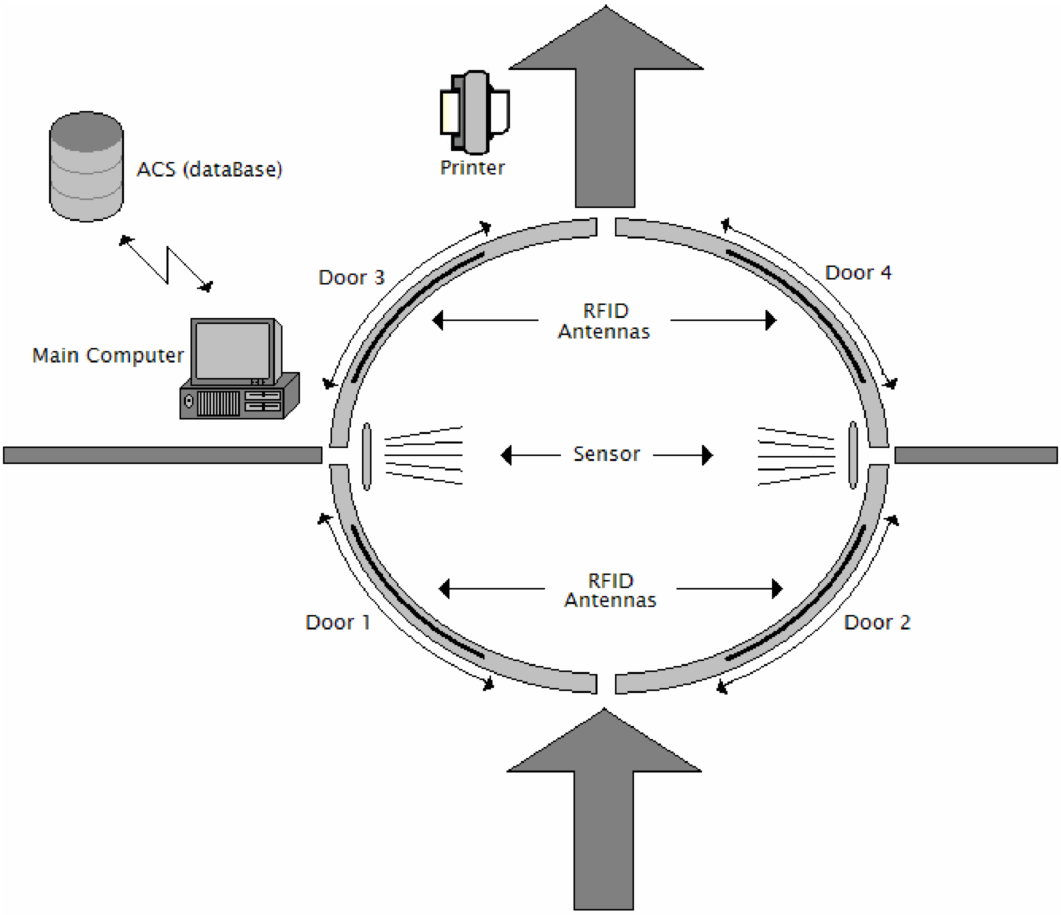


Figure 16 - Basic structure of the system

There are two more antennas in the design not shown in figure 16, one on the floor to scan from below and one in the roof scanning from the top.

The initial condition of the system is that door 3 and 4 are closed while door 1 and 2 are open. A typical sequence looks like this:

1. A person enters through door 1 or 2
2. The sensors notify the main computer that a person has entered.
3. The interrogator starts scanning for tag data using the antennas for the scanning.
4. Data about the scanned tags are sent to the main computer.

5. When door 1 and 2 are closed the interrogators stop the scan.
6. Received data from the interrogators are controlled by the main computer, the main computer communicates with the database (ACS) to determine whether the loan was successful or not.
7. In case of a successful lending sequence door 3 and 4 are opened.
8. A receipt is printed on the printer for the loaner.
9. System returns to the initial condition.

5.2 Components

The components needed in this project are:

1. PC
2. Network connection over TCP/IP
3. 4 doors with special design, read more further on in this thesis
4. Passive tags of model RFT331-1310 [15] for books and library card
5. 3 RFID Interrogators of model RFI341 [15]
6. 6 RFID Antennas of model RFA341 [15]
7. Printer with the possibility to print PostScript receipts
8. Light grid of model PLG3-270F431 [15]

5.3 Computer

The main computer is a standard PC. The requirements on the PC are the following:

1. Sun Java 1.5.0 [14] installed for running the software.
2. At least three serial ports for multiple uses of interrogators.
3. Ethernet card for communicating with the database.

The computer is the state machine in the system, starting the process when the light grids signals that a person has entered and ending the process when returning to initial condition.

Since the software is developed using Java which is platform independent there are no requirements due to the choice of operating system.

5.4 Interrogator

The interrogators used are RFI341 – Radio Frequency Interrogator [15], see figure 17:

Figure 17 - RFI341 [16]

RFI341 is a product from SICK Sensor Intelligence [15]. The interrogator is the unit that interprets the signals received by the antennas and transforms them into serial data sent to the computer.

The interrogator is an ISO-15693 [16] certified [15] read / write unit working at a frequency of 13.56 MHz. Connecting a single antenna to the interrogator a maximum reading distance of 1.2 meters can be achieved due to the high output and high input sensitivity of the interrogator [15].

The 13.56 MHz frequency waves are not more dangerous to humans than a standard car radio since they operate in the low end of the electromagnetic spectrum [21].

The interrogator can read a maximum of 50 tags per second and since the unit contains an internal splitter two antennas can be connected without interfering one another. There is also a possibility to use a photoelectric switch (light grid) for triggering scans [15]; in this project the PLG3-271F431 photoelectric switch was used.

The data interface communication with computer is RS-232 but can be changed to Ethernet via CDM420 with CMF400-3101 [15]. The protocol for the communication with the client is STX / ETX.

5.5 Antenna

The antennas used are RFA341 – Radio Frequency Antenna [15], see figure 18:

Figure 18 - RFA341 [16]

RFA341 is a product from SICK Sensor Intelligence [15]. The antenna is IP 65 which means that it is resistant to dust and flushing of water [15].

5.6 Tags

The tags used are RFT331-1310 – Radio Frequency Transponder [15], see figure 19:

Figure 19 - RFT331-1310 [18]

RFT331-1310 is a product from SICK Sensor Intelligence [15]. The tag is ISO-15693 [16] compatible and therefore accessible with the RFI341 interrogator.

The tag has a unique ID (EPC) that can be read when in front of an antenna connected to an interrogator. The tags are pre-programmed with a unique ID by the manufacturer and there is a possibility for the user to add additional data without interfering with the pre-programmed ID [18]. The tags are placed inside the books and on the library cards.

5.7 Light grid

The light grid used is a PLG3-270F431 – Pick2Light, Reflex light grid [17], see figure 20:

Figure 20 - PLG3-270F431 [17]

PLG3-270F431 is a product from SICK Sensor Intelligence [15]. The light grids are used for the detection of a person entering the RFID scan unit. The unit has IP class 54 which means it is dust protected and protected against sprinkled water [17].

The light grid is 270 mm high [17] and should therefore be enough to detect a person wanting to check out its items.

5.8 Doors

The doors have the following design, see figure 21:

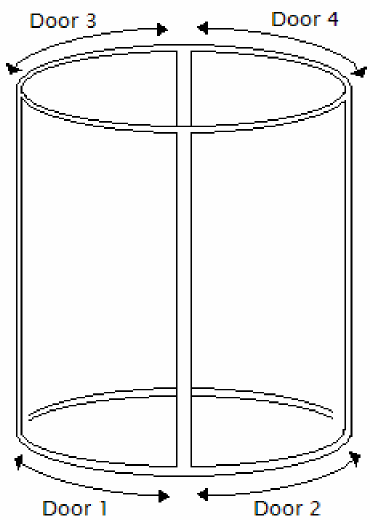


Figure 21 - Design of the doors

The main idea about the doors is four separate doors controlled by the main computer. There are two doors for entering the unit and two doors for leaving the unit.

A supplier of doors has not yet been chosen but the design has been determined as seen in the figure above.

5.9 Printer

The printer is connected to the computer via the LPT-port, the parallel port. The software then pushes a PostScript document onto the printer which prints a receipt on the following form (in Swedish), see figure 22.

Låntagare: John Doe

Bok: 'RFID Revisited' återlämnas senast: 20060718

Figure 22 - Receipt printed for loaner

6. Software

This chapter describes how the software has been developed and all the classes containing the source code.

The software has been developed using Java 1.5.0 [14].

6.1 Basic structure

The basic structure of the software is a state-machine. The state machine starts in the idle state waiting for a person wanting to leave the library and ends when the sequence is ended, i.e. the person is allowed to leave the library or not. There are three threads running simultaneously, one handling the database connection, one handling the RFID unit and one handling the doors.

The state-machine is event driven and starts acting when the light grid has triggered.

Data about the software

- Developed in Java 1.5.0 [14]
- 7 packages
- 36 classes
- 3015 lines of code
- 200 public methods
- 30 private methods

The connection to the main library database is TCP / IP where a socket connection is used to provide the communication. The client sends Strings to the database and the database respond with Strings.

The connection to the RFID Interrogator is a serial cable without handshake.

6.1.1 Flowchart

In this section a flowchart is shown for the different scenarios, see figure 23.

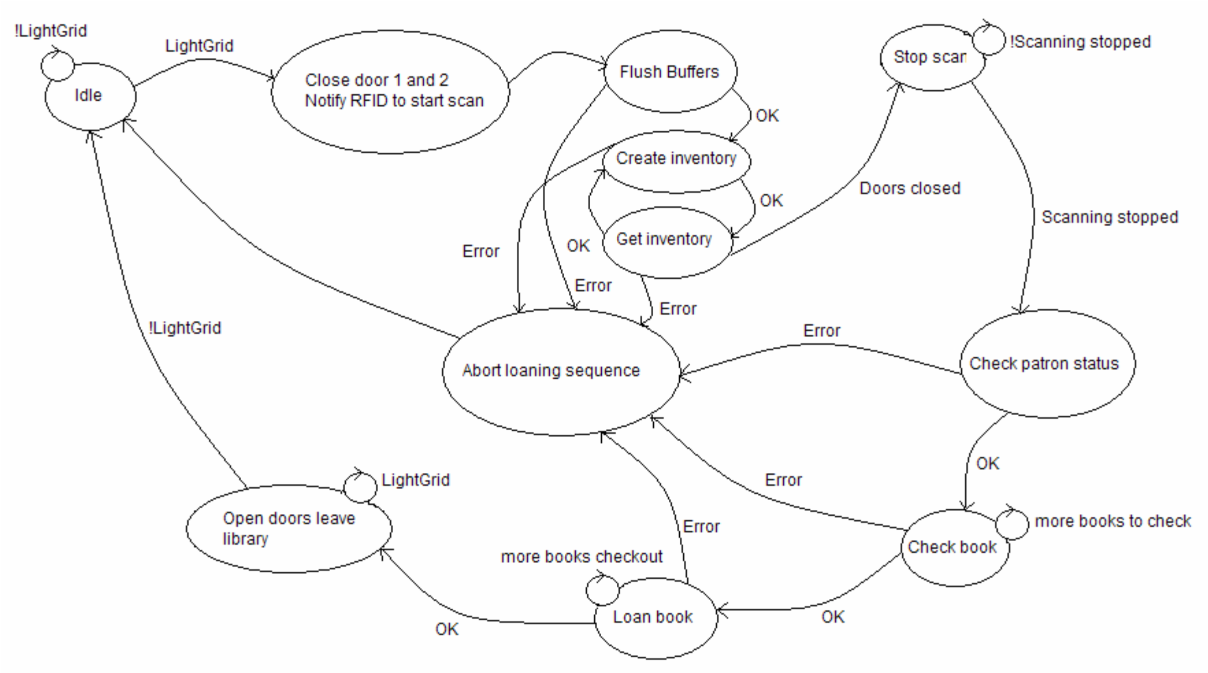


Figure 23 - Flowchart for the state-machine

The state-machine starts in the state idle, idle means that the doors to enter the unit are open while the doors leaving the unit are closed.

When state-machine is in state "idle" the only way to move in the graph is to activate the light grid which leads to the state "Close door 1 and 2, Notify RFID start scan". This state starts the RFID scan in the state "Flush Buffers".

"Flush Buffers" is a state where all the data present in the interrogators internal memory is cleaned so that no extra book of any kind is included in the checkout sequence.

The two following states from "Flush Buffers" are "Create Inventory" and "Get Inventory". These states handle the communication with the interrogator creating inventories and getting the data from read tags. In all of these states there is the possibility of an error, in case of one the state "Abort loaning sequence" is reached immediately aborting the sequence and going back to the idle state.

If no error occurs the state "Stop scan" is reached when the doors to enter the unit are closed. This state stops the scan and enters the state "Check patron status" which is a part of the database communication.

The "Check patron status" state controls if the library card is valid, if not the "Abort loaning sequence" state is reached. In case of a correct library card the "Check book" state is reached.

The “Check book” state checks all the books if one is allowed to loan the book, if it costs money to loan the book and if any of these occur the “Abort loaning sequence” state is reached, otherwise the “Loan book” state is reached. The reason that all the books are controlled if they are allowed to leave the library first is because in case of an unsuccessful checking sequence no books are checked out of the library. In this way both the library card and books are checked before checked out in the library database minimizing the risk of error.

If no error occurs during the “Check book” state the books are checked out in the “Loan book” state, the errors that can occur in this state are purely communication errors and in that case the state “Abort loaning sequence” is reached.

The final state if all other passed are the “Open doors leave library” state. In this state the doors to leave the unit are opened allowing the person loaning books to leave. When the person has left the library the light grid is no longer triggered and the state-machine returns to the idle state, i.e. closing doors to leave the unit and opening doors to enter the unit.

6.2 Unified Modelling Language

This is the structure of the software and its relationships to each other, see figure 24. Notice that relationships within the packages can be viewed under the technical documentation.

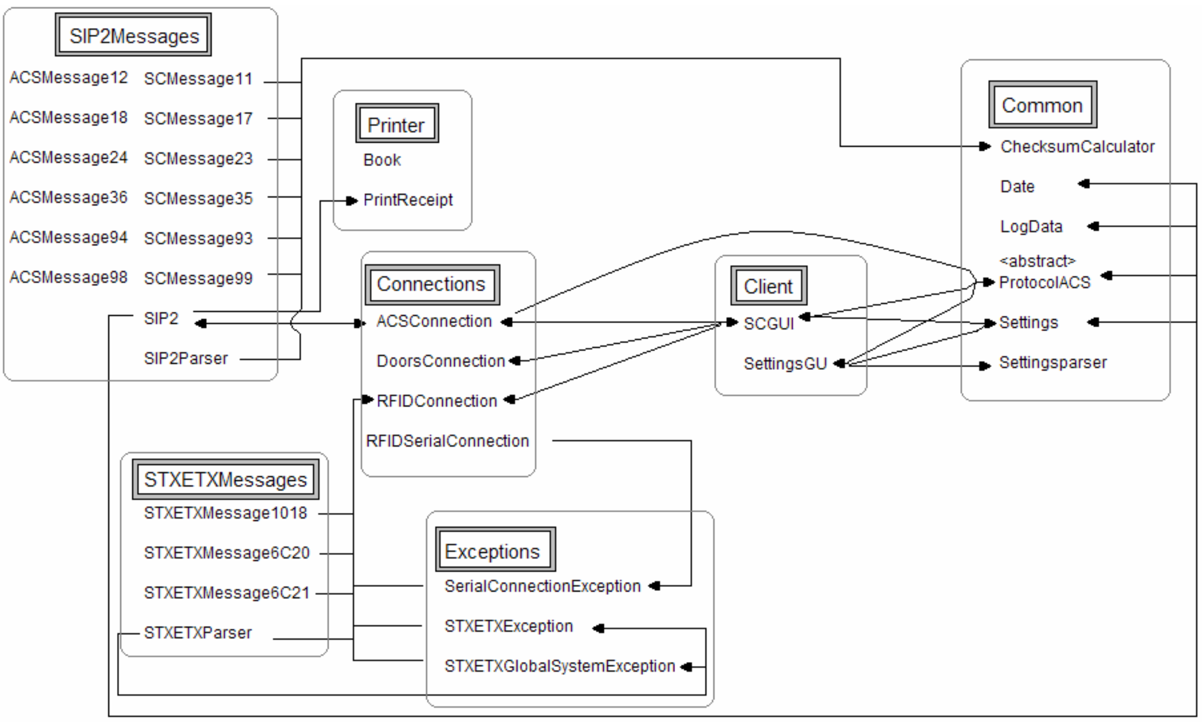


Figure 24 - UML diagram of the software

6.2.1 Relationship between UML and Flowchart

The first thing to do in the software is to connect to all the components, i.e. the doors, the interrogator and the database. The first screen displayed for the user is the settings screen (“SettingsGUI”), the previously used settings are stored in a text file and received by the “SettingsParser” and stored in the class “Settings”.

When connecting to the database the thread in “ACSConnection” is started and a login message “SCMessage93” is sent and an “ACSMessage94” is received. Then a status message “SCMessage99” is sent to notify the database of the client status, the received message is an “ACSMessage98” notifying the client of the database status, for example timeout time and maximum number of retries. All messages are synchronized via “SIP2” which send and receive all messages and using the “SIP2Parser” to check all messages that they are complete and correctly received, i.e. a correct checksum which is calculated by the “ChecksumCalculator”.

When connecting to the interrogator the thread in “RFIDConnection” is started and the serial connection is initialized in the “RFIDSerialConnection”. When connected an “STXETXMessage1018” is sent flushing the buffers of the interrogator.

When connecting to the doors the thread in “DoorsConnection” is started.

In the first state “Idle” all the three threads are asleep and the doors thread are waiting for the light grid to trigger. When the light grid has triggered the class “DoorsConnection” notifies “SCGUI” that someone has entered. “SCGUI” notifies the class “RFIDConnection” to start the scan. The scan is started by sending the message “STXETXMessage1018” which flushes the buffers, i.e. cleans the buffers.

To start the inventory an “STXETXMessage6C20” is sent and received checking if an error occurs. The scanned inventory is received by sending and receiving an “STXETXMessage6C21” containing all the scanned data. All the received messages from the interrogator are parsed by the “STXETXParser” and the checksum are controlled by the “ChecksumCalculator”.

In case of an error in the serial communication there are three different possible exceptions, “SerialConnectionException” if an error in the communication, “STXETXException” if a known error occurs in the interrogator for example an incorrect message and “STXETXGlobalSystemException” if a global system exception occurs, if this exception occurs the client closes the connection to the interrogator and the loaning sequence is aborted.

When the doors are closed the “DoorsConnection” notifies “SCGUI” that the doors are closed which notifies the “RFIDConnection” to stop the scan. The “RFIDConnection” now check if any library card has been scanned and if any books have been scanned.

If no error occur during the scan the check patron state is reached. In this state the “ACSConnection” sends an “SCMessage23” with the patron id requesting information about it and a “ACSMessage24” is received with the information. If the library card is valid the check books state is reached.

When checking all the books for each book an “SCMessage17” is sent and an “ACSMessage18” is received checking if one is allowed to loan the book or not. In case one is allowed to loan all books the loan books state is reached.

When loaning all the books for each books an “SCMessage11” is sent and an “ACSMessage12” is received checking out the books from the library. When all books are looped through an “SCMessage35” is sent ending the current session and an “ACSMessage34” is received acknowledging the end of the current session.

When the checkout has been accomplished the books are stored as “Book” in the class “PrintReceipt” which pushes a PostScript document to the printer. In this way a receipt is printed. The books id and the patron id are also stored in a text file with the class “LogData”.

In case of no error the open doors leave library state is reached which notifies the “DoorsConnection” to open the doors to leave the library. When the light grid is not triggered any more the system returns to idle, i.e. all threads sleep.

6.2.2 DataBase Interface

The abstract class “ProtocolACS” is an interface for the communication between the client and the database. The class has following look:

```
public abstract class ProtocolACS {

	//Shutdown of client
	public static final String SHUTDOWN = "SHUTDOWN";

	//Different protocols
	public static final int SIP2 = 1;

	//Required functions by protocol
	public abstract void loginSequence();
	public abstract String disconnect();
	public abstract String getResendMessage();
	public abstract void parseMessage(String message);
	public abstract boolean checkCorrectMessage(String message);
	public abstract void loanBooks(String patronId,
	                               String[] itemId);
	public abstract void setSettings(Settings settings);
}
```

The String SHUTDOWN is a String used by the class “ACSConnection” to notify it that the connection is about to shut down. Adding an extra protocol one has to add an extra static int containing a unique id and a name of the protocol.

The abstract methods are the following:

void loginSequence()
This method is the first method called by the "ACSConnection". The class handles the login sequence to the database. This method has to send the login message to the database.

String disconnect()
This method should return a String containing the message used to log out from the database.

String getResendMessage()
This method should return a String containing a message sent to the database when a faulty message has been received, i.e. a message notifying the database to resend the last message.

void parseMessage(String message)
This method is the synchronization between all the messages. Every message received by the "ACSConnection" is sent to this method. This method then interprets the message and determines the next message and notifies the "ACSConnection" which is the next message to be transmitted if any.

boolean checkCorrectMessage(String message)
This method should return true if the message in the String "message" is correct and false otherwise.

void loanBooks(String patronId, String[] itemId)
This method should start the loaning sequence with the scanned patronId and the vector containing the books Id.

void setSettings(Settings settings)
This method is used to set the settings entered at the start-up of the software. For example login name and password.

6.3 Technical documentation

The technical documentation contains information about all source code packages and their purpose.

6.3.1 Package – Client

The package named Client contains three different files that are responsible for the graphical user interface and the synchronization between the doors, RFID unit and the database.

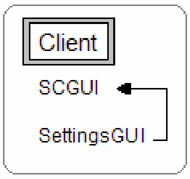


Figure 25 - Relationships within the package Client

SCGUI
This file contains the source code for the graphical user interface shown when running the system.

SettingsGUI
This file contains the source code for the graphical user interface shown when starting up the software. It handles all the settings needed to perform before starting the main interface. It handles for example the password required for the main database as well as saving all the entered settings besides the password for easier start-up.

StartSC
The main program file, used for starting the application.

6.3.2 Package – Common

The package Common contains files used by many different classes and therefore there are many *static* classes for easy access.

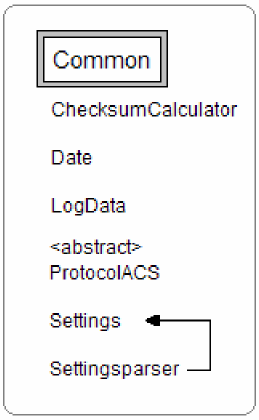


Figure 26 - Relationships within the package Common

ChecksumCalculator
This class handles all the calculations of checksum for the database protocol as well as for the interrogators protocol. All methods in the class are static and the check if the received message had the correct checksum can be supported.

Date
This class handles the current date and when asked it returns the current date on the specific form required by the communications protocol.

LogData

This class handles the logging of data, every time a person is allowed to check out an item through the RFID system the library card ID and the books ID:s are logged in files. The file the data is logged in is named by the current date for example 20060819.txt for the 19th of august 2006. The data are stored in an easy way allowing the user to look at the data in Microsoft Excel.

ProtocolACS
This class is an abstract class and acts as an interface for the desired communication protocol to the database. This allows future users using a different protocol to use the same source code for the connection to the database and the synchronization only changing the source code for the specific protocol.

Settings
This class handles all the settings entered at start-up and allows other class to read them when needed.

SettingsParser
This class handles the reading of the settings-file and creates an instance of the class Settings allowing the user to save the desired settings not needing to re-enter them at the next start-up sequence.

6.3.3 Package – Connections

The package Connections handles all the different connections, i.e. for the database which is a TCP/IP connection and for the interrogator which is a RS-232 connection.

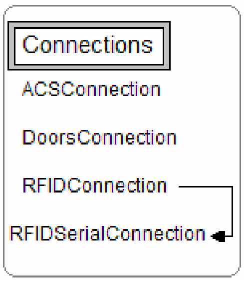


Figure 27 - Relationships within the package Connections

ACSConnection
This class handles the communication between the client and the main database. The connection is TCP/IP and therefore it opens a socket between the client and the database to operate on. The class also handles settings decided by the database such as time before retry if didn't receive expected message as well as handling maximum number of retries before client closes the connection.

DoorsConnection
This class handles the connection to the doors. The class notifies the doors when to open and when to close.

RFIDConnection

This class handles the communication between the client and the interrogator. The connection is RS-232 and there for a serial cable is used between the client and the interrogator. The class handles all messages sent and received according to the STX / ETX protocol and are therefore a bit more complicated to change than the database protocol.

RFIDSerialConnection
This class handles the actual communication between the client and the interrogator but doesn't handle the received messages which are done by the previously described class RFIDConnection. The class has functions like open a connection, close a connection, read and write on the serial port, in this case default COM1.

6.3.4 Package – Exceptions

The package Exceptions contains classes with exceptions used for notifying the software errors as easy as possible.

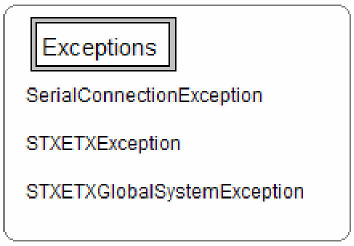


Figure 28 - Relationships within the package Exceptions

SerialConnectionException
This class contains an Exception which if thrown when an error occurs in the class RFIDSerialConnection and is catched by the class RFIDConnection which determines the appropriate action.

STXETXException
This class contains an Exception which is thrown when an error occurs on the interrogator. The exception is thrown by the class STXETXParser which is a *static* class.

STXETXGlobalSystemException
This class contains an Exception which is thrown when a global system exception occurs on the interrogator. Global system exception means severe errors on the interrogator and it is therefore important to disconnect the interrogator when this error occurs.

6.3.5 Package – Printer

The package Printer contains classes used for the printing feature; the classes create postscript documents and push them onto the printer port.

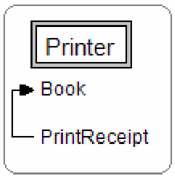


Figure 29 - Relationships within the package Printer

Book
This class contains information about a book. It handles the name of the book as the due date of the book.

PrintReceipt
This class handles all the printing as well as adding books to the current loaning list. When asked the class creates a postscript document with information about the loaner and all the books with their respective due date.

6.3.6 Package – SIP2Messages

The package SIP2Messages contains classes which handle the protocol SIP2 and all its different messages passed between the client and the database. As mentioned before the client always starts the conversation with the database and then waits for an answer, each sent message from the client has to be responded by the database. All classes starting with *SC* are messages sent to the database and all classes starting with *ACS* are messages received from the database.

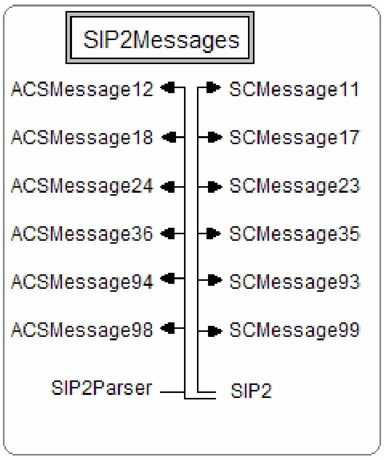


Figure 30 - Relationships within the package SIP2Messages

SCMessage93 (Login)
Client send SCMessage93 → Database respond with ACSMessage94
This class handles the login message sent from the client to the database allowing the user to define password, username and the optional location ID.

ACSMessage94 (Login response)
This class handles the message received from the database when the login message was sent. The message tells the client whether the login was successful or not.

SCMessage99 (SC Status)
Client send SCMessage99 → Database respond with ACSMessage98
This class handles the message sent from the client to the database notifying the database if the client is going on-line or offline as well as telling the protocol version used.

ACSMessage98 (ACS Status)
This class handles the message received from the database when the SC Status message was sent. The message notifies the client different options supported by the database as well as the maximum number of retries and the time out time.

SCMessage23 (Patron Status Request)
Client send SCMessage23 → Database respond with ACSMessage24
This class handles the message sent from the client to the database asking for the status of a specific library card.

ACSMessage24 (Patron Status Response)
This class handles the message received from the database when the Patron Status Request was sent. The message notifies the client the name of the library card owner and if it is a valid library card.

SCMessage17 (Item information)
Client send SCMessage17 → Database respond with ACSMessage18
This class handles the message sent from the client to the database asking about information of a specific item.

ACSMessage18 (Item information response)
This class handles the message received from the database when the Item information was sent. The message notifies the client if one's allowed to loan the book or not.

SCMessage11 (Checkout)
Client send SCMessage11 → Database respond with ACSMessage12
This class handles the message sent from the client to the database notifying the database to check out the desired item.

ACSMessage12 (Checkout response)
This class handles the message received from the database when Checkout was sent. The message notifies the client whether the checkout was successful or not.

SCMessage35 (End patron session)
Client send SCMessage35 → Database respond with ACSMessage36
This class handles the message sent from the client to the database notifying the database that the current session is over.

ACSMessage36 (End session response)

This class handles the message received from the database when End patron session was sent. The message notifies the client whether the session was successfully ended or not.

SIP2
This class handles the synchronization between the messages in a loaning sequence. The class also handles all received messages and notifies appropriate classes of the required setting received from the database. It also notifies the state machine if the loan was successful or not.

SIP2Parser
This class handles the parsing of the messages, determining what type of message and creating instances of ACS messages used by the SIP2 class to access vital information from the database.

6.3.7 Package – STXETXMessages

The package STXETXMessages contains classes which handle the STX / ETX protocol used for the communication between the client and the interrogator. The classes are so called multifunction since they are used both for creating messages to send and to read the response from the interrogator.

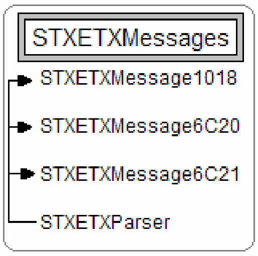


Figure 31 - Relationships within the package STXETXMessages

STXETXMessage1018
This class handles the function Flush buffer which is used to clean all current data in the interrogators buffer. It also handles the response from the interrogator notifying the client that buffers has been flushed.

STXETXMessage6C20
This class handles the function Create inventory which is used to start a new inventory, i.e. start scanning for tags. It also handles the response from the interrogator notifying the client that a new inventory has been created.

STXETXMessage6C21
This class handles the function Get inventory which is used to get the data about the tags that have been scanned. It also handles the response from the interrogator notifying the client about the scanned data.

STXETXParser
This class is used to parse messages from the interrogator and notify the RFID connection if any error occurred and how essential it is.

7. Setup

This chapter describes how to connect the software and the hardware. It includes for example the type of cables needed and the power consumption.

7.1 RFID

The interface of the RFI341 [18] looks as figure 32:

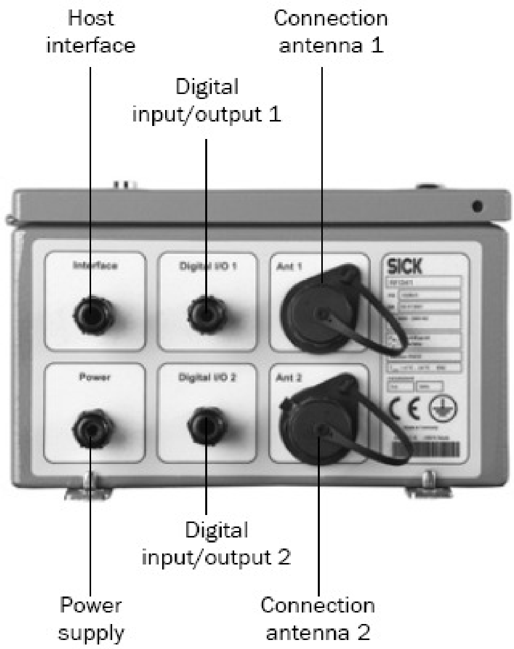


Figure 32 - Interface for RFI341 [18]

The interrogator is as mentioned before controlled by RS-232, serial cable, interface [18]. The protocol used is described in the chapter “Protocols”. The serial cable should have the following set-up, se figure 33.

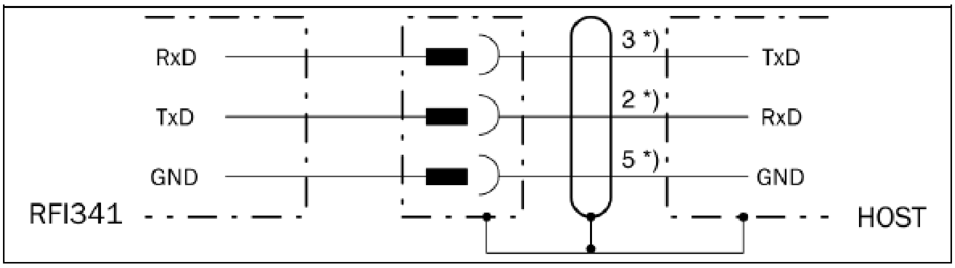


Figure 33 - Setup for the RS-232 communication [18]

As seen in the figure above no handshake of any kind is used, so this is the same set-up as a simple null modem.

Connecting the light grid or so called photoelectric reflex switch to the interrogator is done in the following way, see figure 34:

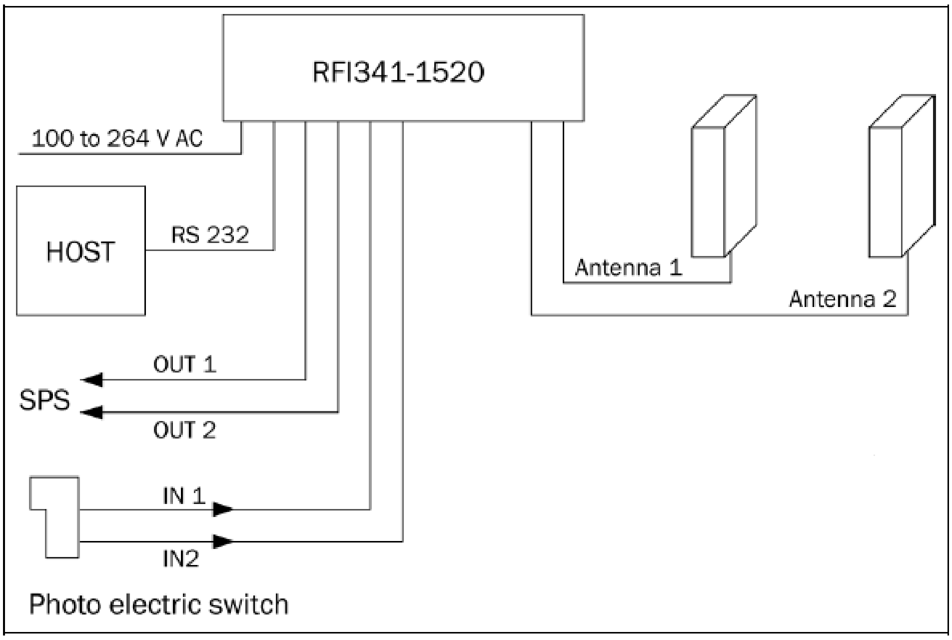


Figure 34 - Connecting the RFI341 [18]

As seen in the figure above the antennas are connected to the interrogator and they should be of model RFA3x1 in this case RFA341 [16] is used. The light grid is connected to the inputs used for triggering the tag scan.

The connection to the host, in this case the developed software, is performed via a RS-232 interface as seen in the figure above.

The light grid is connected to the Digital Input 1 with its M12, 4-pin connection which has the following setup, see figure 35.

Figure 35 - Light grid connection setup [20]

The light grid switches output PNP which means that the current flows from the emitter to the collector which is compatible with the RFI341 [19] which has the following setup for the digital inputs, see figure 36.

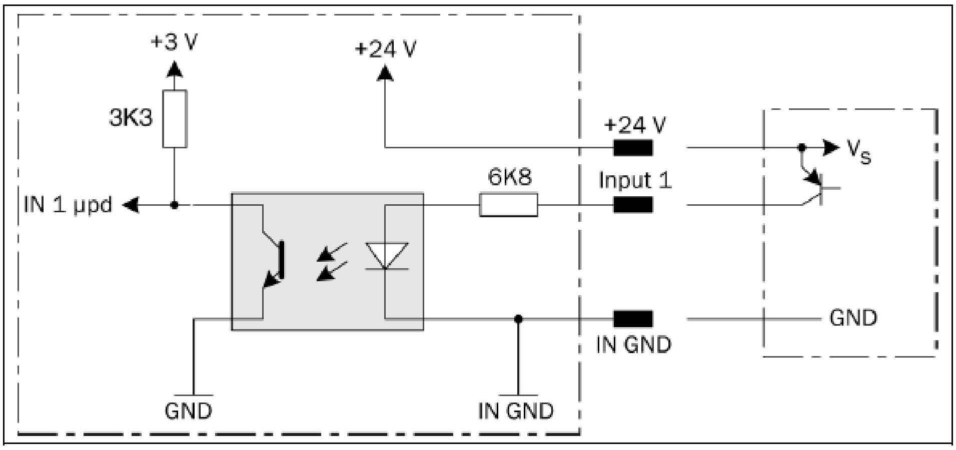


Figure 36 - Digital input on RFI341 [19]

As seen in the figure above the current flow from the emitter to the collector.

The tags are placed on the inside of the book, see figure 37, together with a bar code (EPC bar code) allowing the user to check out the book with the RFID unit or at a clerk.

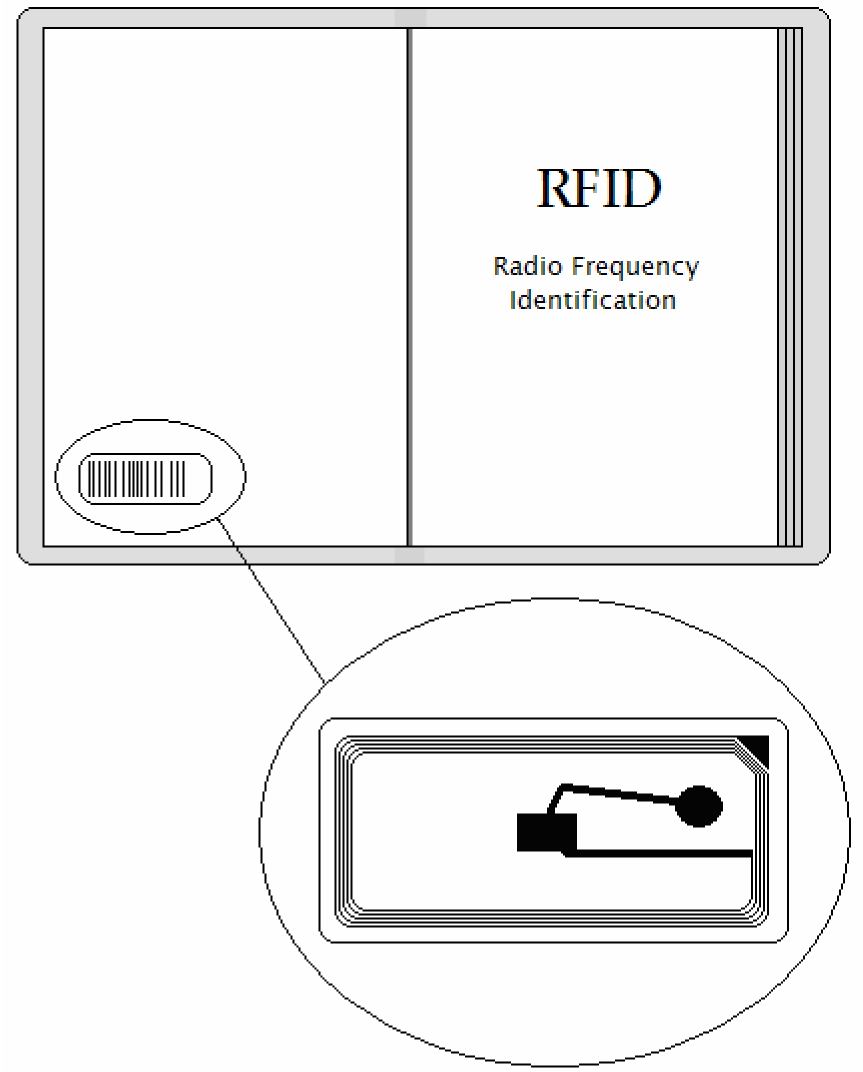


Figure 37 - Book with tag including barcode

The library cards are also equipped with tags as well as a bar code. The library card are the same size as credit cards.

The tags are pre-programmed with the same unique ID as the bar code, the only difference is that in the programmable section of the tag a character will be added to notify the client whether it is a book or it is a library card. The following syntaxes are used:

- P <unique ID> for the library card
- B <unique ID> for the book

8. Analysis

This chapter discusses the analysis of special problem areas using RFID technology. It is also discussed if and how these problems can be solved.

8.1 Angle between antenna and tag

One of the major difficulties is that the tag is best scanned at an angle of 90^{o} towards the antenna so the behaviour of the tags has to be investigated as they differ from this angle.

8.1.1 Theory

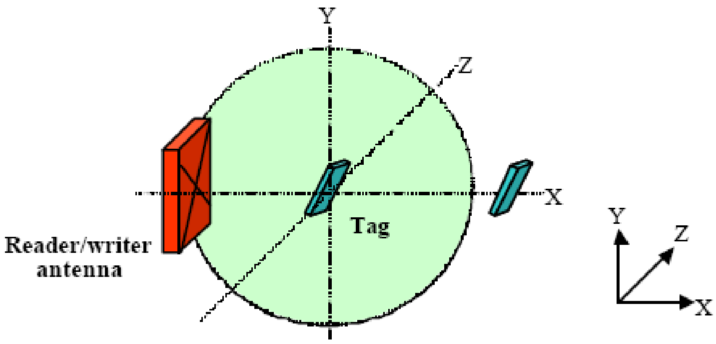


Figure 38 - RFID scan angle [22]

In theory it should be enough to have sliding doors scanning around the loaner, see figure 39, for the tags and in that way detecting all the present tags.

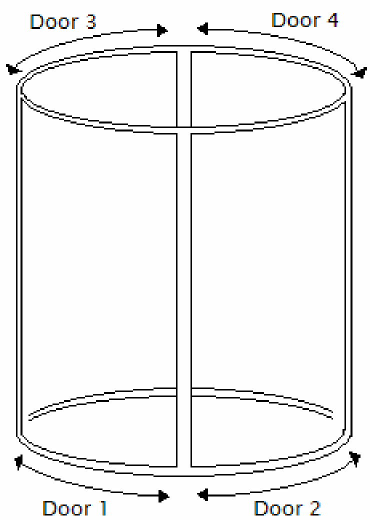


Figure 39 - Door design

The doors have one antenna each incorporated into the door. The angle of scan corresponding to each door is the following, see figure 40.

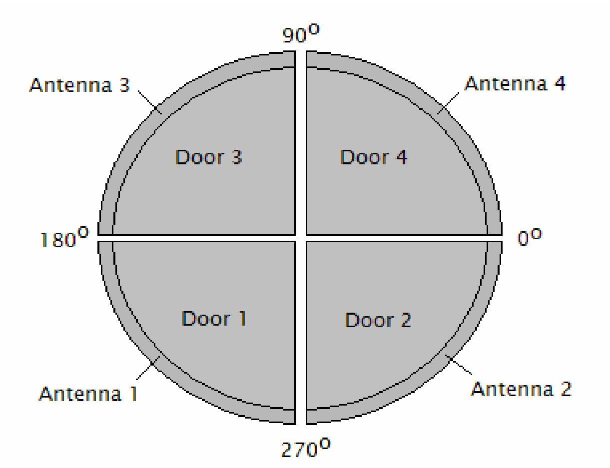


Figure 40 - Scan-angle of antenna 1 to 4 seen from above

As seen in the figure above all angles around the loaner are scanned by four independent antennas. The next problem is to scan from above and under the loaner this was solved by adding two antennas, one at the top and one at the bottom scanning like this, see figure 41.

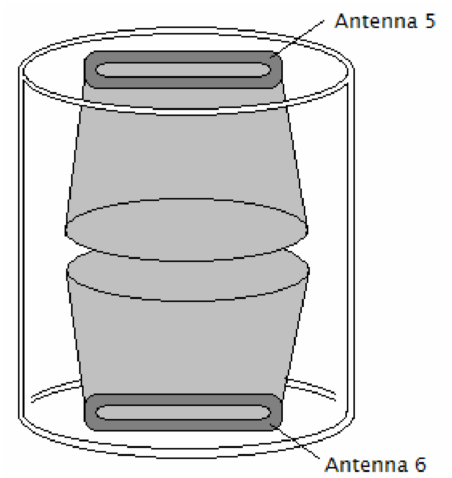


Figure 41 - Scan-angle of antenna 5 and 6 seen from the side

Another but more expensive approach to the top and bottom scan is to let door 1 and door 2 scan as before. The difference here is to rotate the magnetic field of the antennas in door 3 and door 4. This is done by introducing a splitter between the antennas and then make one of the coaxial cables longer than the other.

In this case one want to phase shift one antenna 90°. To do that one has to calculate the wave length:

$$\lambda = \frac{c}{f} = \frac{300000000}{13560000} = \frac{300}{13.56} = 22.12m$$

To phase shift this now one has to know the velocity factor of the coaxial cable in this case denoted by y.

$$Extension = \frac{22.12}{4}y = 5.53y$$

The setup of the system will now be the following, see figure 42.

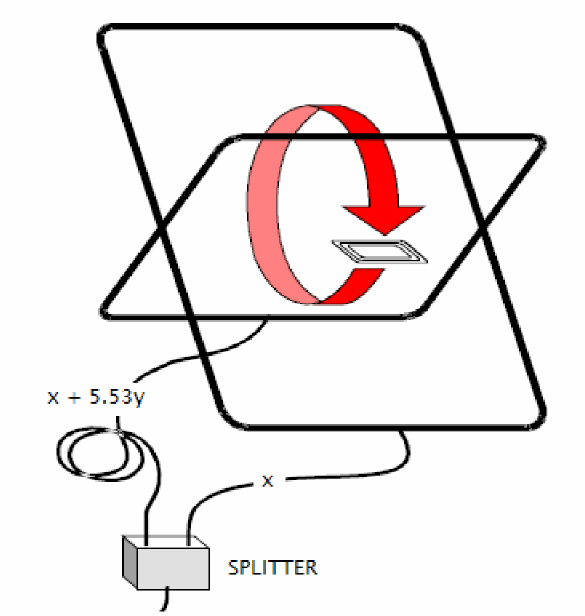


Figure 42 - Top-bottom scan with phase shift [23]

A way to set up the splitter and still keep the resistance at 50 Ohm as required [16] is in figure 43.

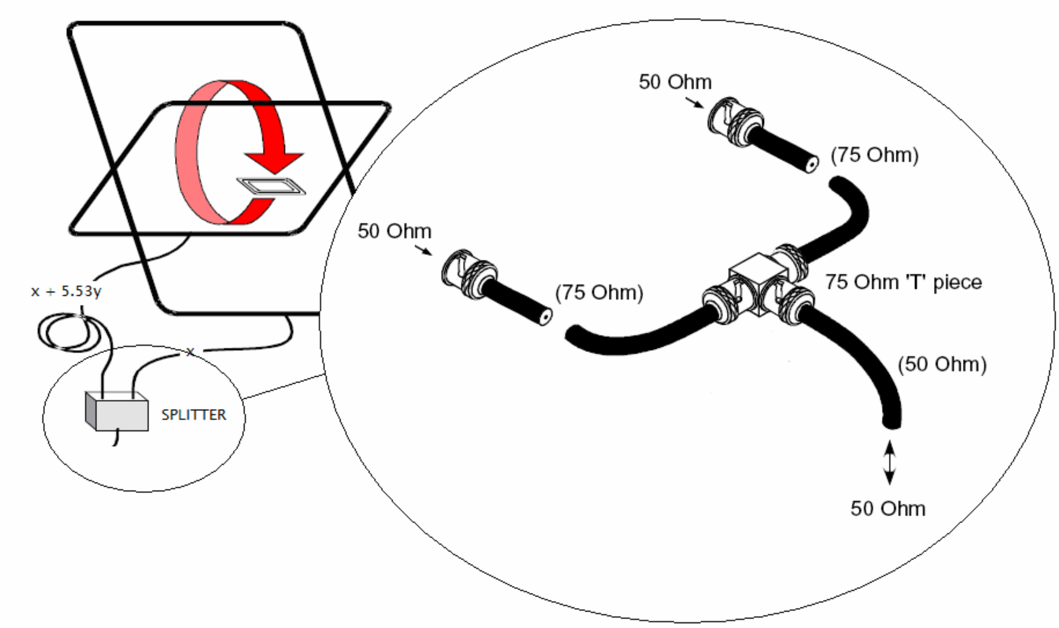


Figure 43 - Coaxial cable splitter [23]

The critical angle in this setup is when the angle of the tag between the doors antennas and the top or bottom antennas are 45°. If this angle result in a successful read it is functional for all other angles since they are closer to 90° in angle to at least one of the antennas.

The antennas create a magnetic field which has the following look, see figure 44:

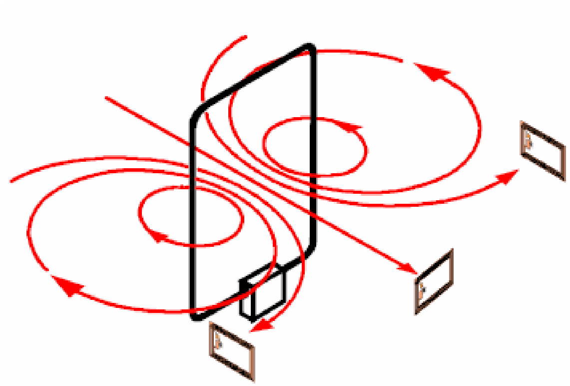


Figure 44 - Antenna magnetic field [23]

There should not be a problem with the angle between the antenna and the tag. This since the doors scan around the tags and therefore cover all angles.

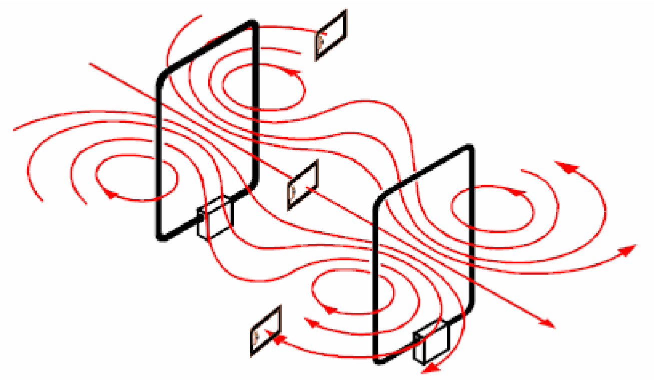


Figure 45 - Dual antenna magnetic field [23]

Figure 45 shows that introducing two antennas, which has been done in this case, creates an even bigger and more powerful magnetic field minimizing the risk of errors.

Experiments show that the angle between the tag and the antenna is important, but still a successful read is conducted at an angle of 45° between the tag and the antenna [22]. Test set-up see figure 46.

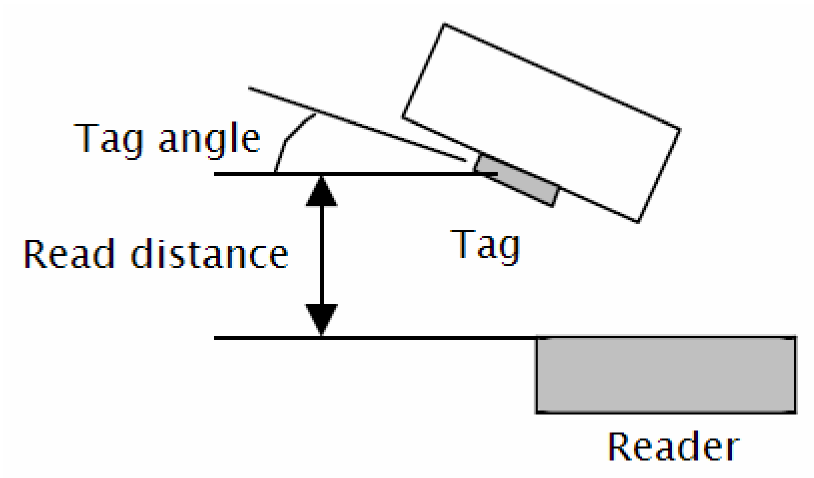


Figure 46 - Angle between tag and reader

The reading distance is impacted by the angle and a study [22] gave the following result of reading distance compared to angle.

Top angle	0^{o}	30^{o}	60^{o}	90^{o}
Read distance	45 cm	40 cm	35 cm	30 cm

Note that the maximum distance of read for the tested system is 45 cm compared to 120 cm which is the maximum read distance for the intended system in this thesis.

An analysis of this shows that the read distance is linear compared to the angle between the tag and the antenna.

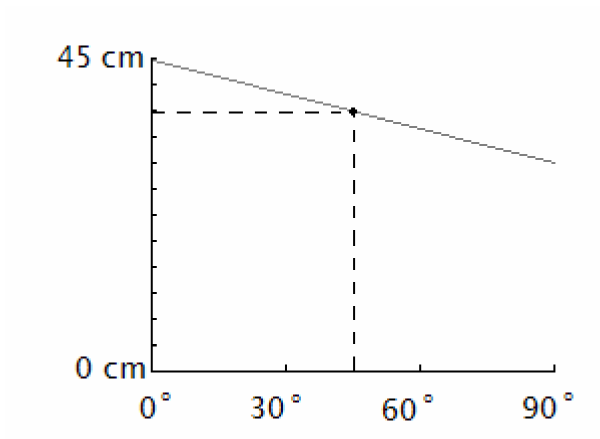


Figure 47 – Relationship between angle and read distance

As seen figure 47 at an angle of 45^{o} the read distance is approximately 37.5 cm. The read distance decreases about 16 % due to the change of angle. Since the maximum read distance in this system is 45 cm the corresponding loss of read distance in a system with maximum 120 cm would be 20 cm. Still the read distance should be 100 cm which in this case is sufficient.

8.1.2 Experimental setup

The experiment has been conducted using the following components:

- RFI341 (Interrogator)
- 2 x RFA331 (Antennas)
- RFT331 (Tags)

The antennas have a maximum reading distance of 55 cm which is a bit less than desired but still applicable for the testing procedure. SICK [15] has other antennas that are more applicable with longer reading distance which will be used when setting up the entire system.

The software used for the experiments was a demonstration software for the RFI341 developed by SICK [15]. The reason this was used is because the software supports all features available for the interrogator. The software developed in this thesis does not support all features since the complete system does not require for instance writing to tags.

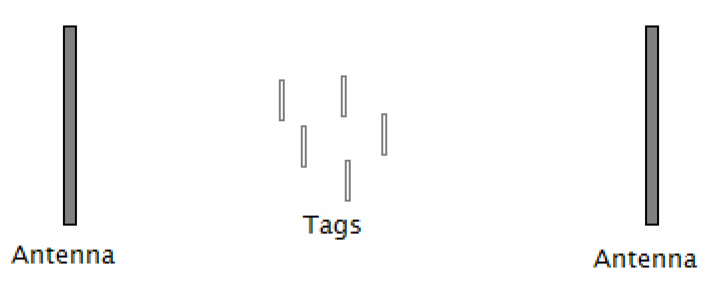


Figure 48 - Experimental setup

There is one meter between the antennas and the tags are placed at a distance of 50 cm from each antenna as seen in figure 48.

8.1.3 Result

Experiments show that the best reading result is received when the antennas are facing each other. Badly positioned antennas could cause an unbalanced electromagnetic field that is out of phase, see figure 49.

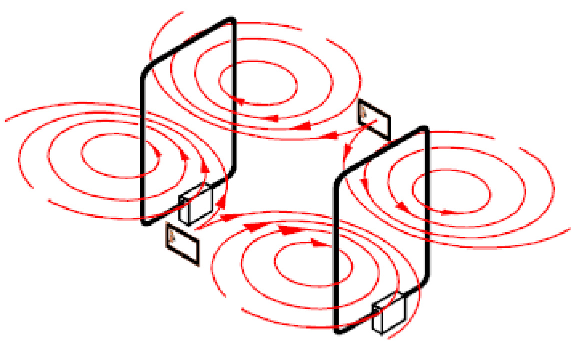


Figure 49 - Antennas out of phase [23]

To handle this problem the main idea is to let the antennas scan during different intervals of the scan cycle. In other words antenna 1 starts the scan and reads its data, antenna 1 stop the scan and allows antenna 2 to scan and so on.

Different test has been conducted to check how the tag behaves in different angles towards the antenna. The following variation of angles has been tested, see figure 50.

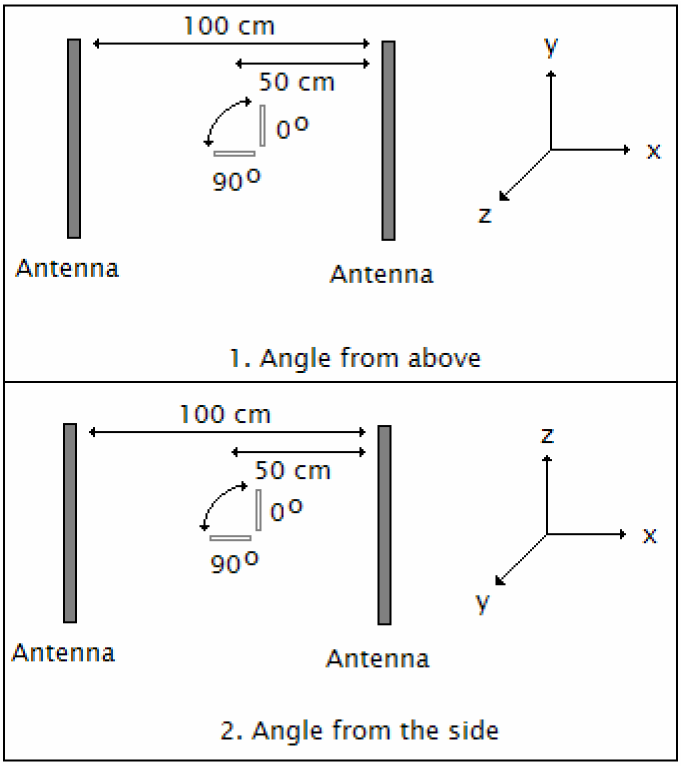


Figure 50 - Tested angles

And the result was as follows in table 1:

Angle from above (1)	Angle from the side (2)	Successful read
0 degrees	0 degrees	YES
15 degrees	0 degrees	YES
30 degrees	0 degrees	YES
45 degrees	0 degrees	YES
60 degrees	0 degrees	YES
90 degrees	0 degrees	NO
0 degrees	15 degrees	YES
0 degrees	30 degrees	YES
0 degrees	45 degrees	YES
0 degrees	60 degrees	YES
0 degrees	90 degrees	NO
15 degrees	15 degrees	YES
30 degrees	30 degrees	YES
45 degrees	45 degrees	YES
50 degrees	50 degrees	YES

Table 1 – Test result RFID

As determined before the critical angle of the system is when the angle from the side is 45 degrees. The scan at this angle has been successful and still allows the angle to be 50 degrees from above and 50 degrees from the side. This is a very satisfying result and the conclusion is drawn that with other antennas it is possible to introduce automatic loaning at libraries using the RFID technology.

8.2 Attaching the tags to books

This experiment is to determine whether the previous experiment also works when attaching the tags to books.

8.2.1 Experimental setup

The setup of the components is the same as in the previous experiment. The books have been attached with tags on the inside of the book, see figure 51.

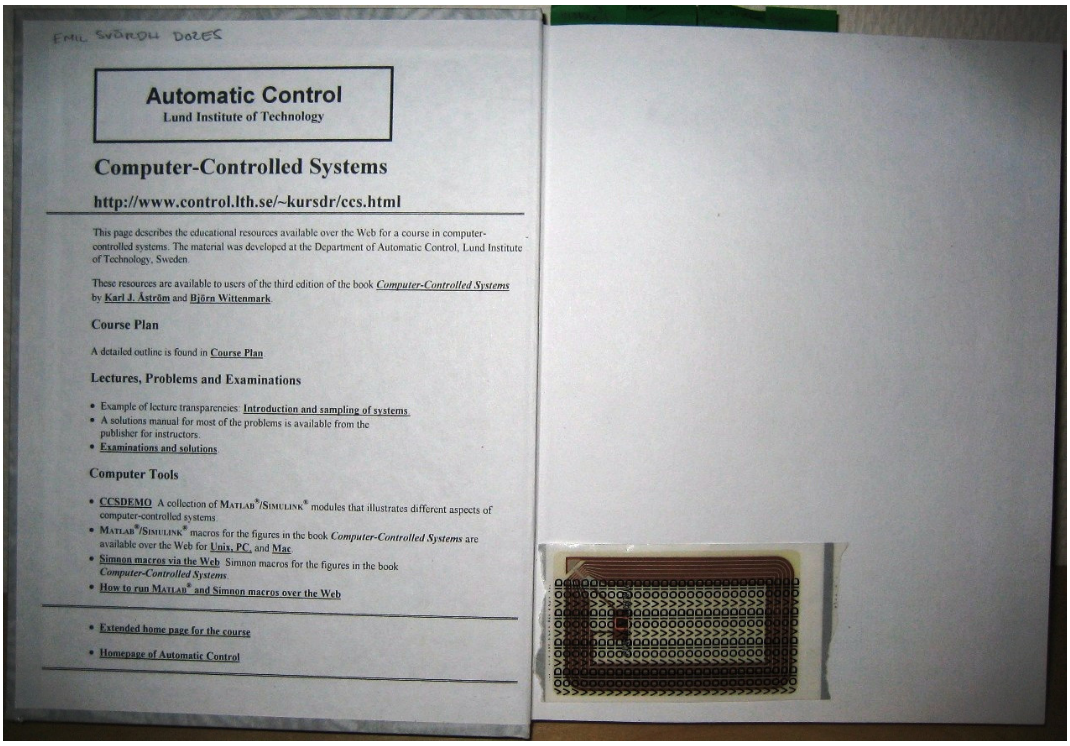


Figure 51 - Book including tag

In the experiment four books have been used and tried in different angles as well as put closely together since that is the worst case scenario, see figure 52.

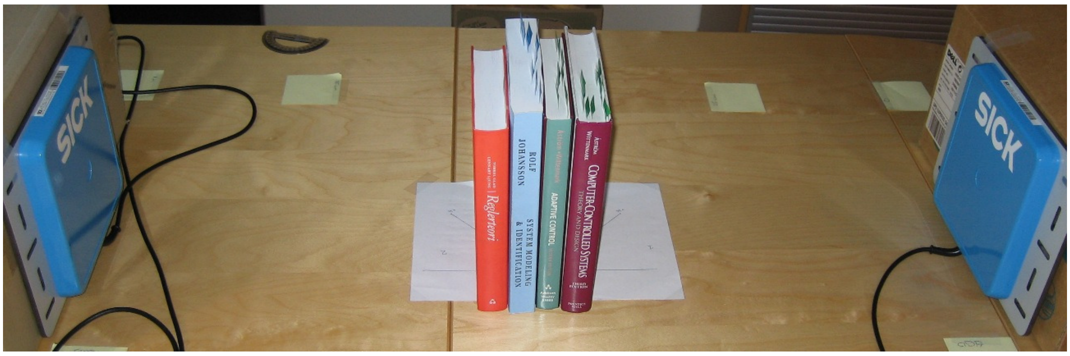


Figure 52 - Books between antennas

8.2.2 Result

Experiments show that there is no problem reading the tags when inside a book. The same test angles were used as in the previous experiment and the results were as follows, see table 2:

Angle from above (1)	Angle from the side (2)	Successful read
0 degrees	0 degrees	YES
15 degrees	0 degrees	YES
30 degrees	0 degrees	YES
45 degrees	0 degrees	YES
60 degrees	0 degrees	YES
90 degrees	0 degrees	NO
0 degrees	15 degrees	YES
0 degrees	30 degrees	YES
0 degrees	45 degrees	YES
0 degrees	60 degrees	YES
0 degrees	90 degrees	NO
15 degrees	15 degrees	YES
30 degrees	30 degrees	YES
45 degrees	45 degrees	YES
50 degrees	50 degrees	YES

Table 2 - Test result with books

With the same reasoning as in the previous experiment the critical angle is when the angle is 45 degrees from the side. This angle gave a satisfying result and therefore it can be concluded that the system works. Test has also been conducted when placing the books inside a bag and this gave the same result, i.e. the read result is not influenced by obstacles like bags.

8.3 Testing the developed software

The developed software works very well and no errors at all have been found. The loaning sequence is as described in the chapter "Software" and all different scenarios behave as expected.

For instance if one test with four books the sequence starts with that the light grid is triggered and the doors start to close. Then when doors are closed the RFID scan is ended and the following scenarios can occur. Figure 53, 54 and 55 show four different scenarios and what happens in each step at each component.

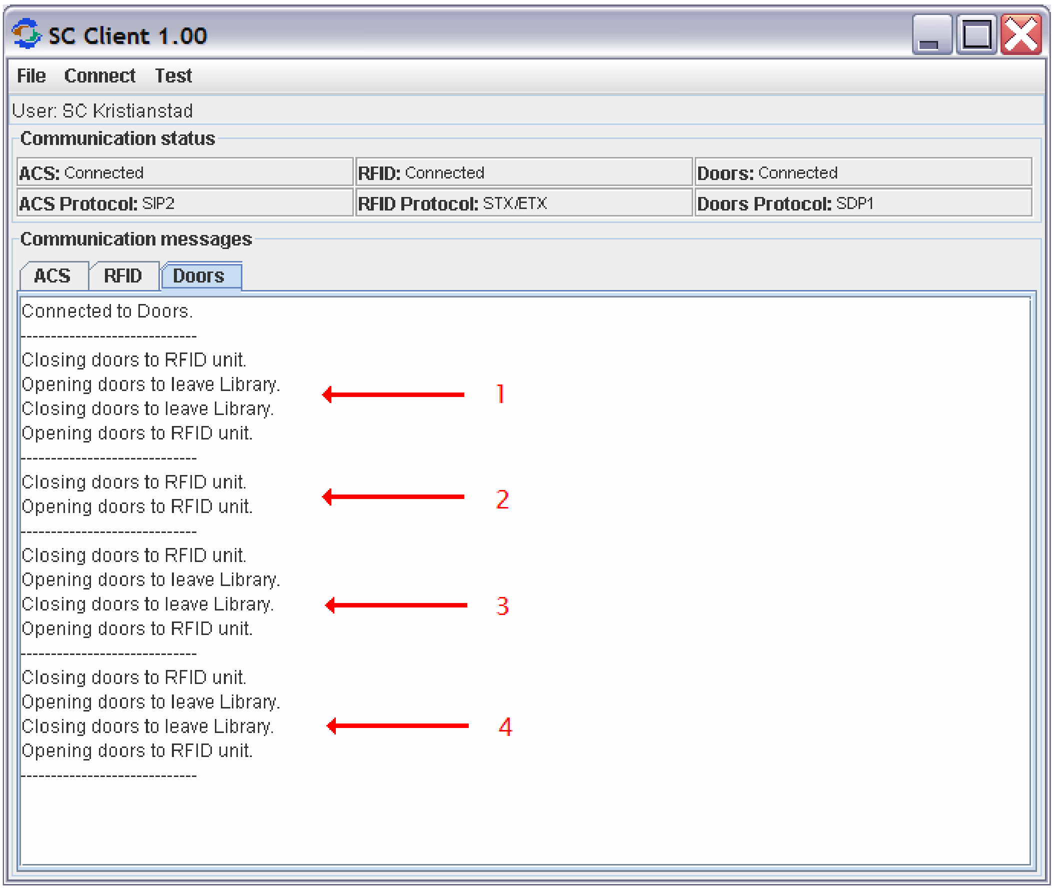


Figure 53 - Door sequences

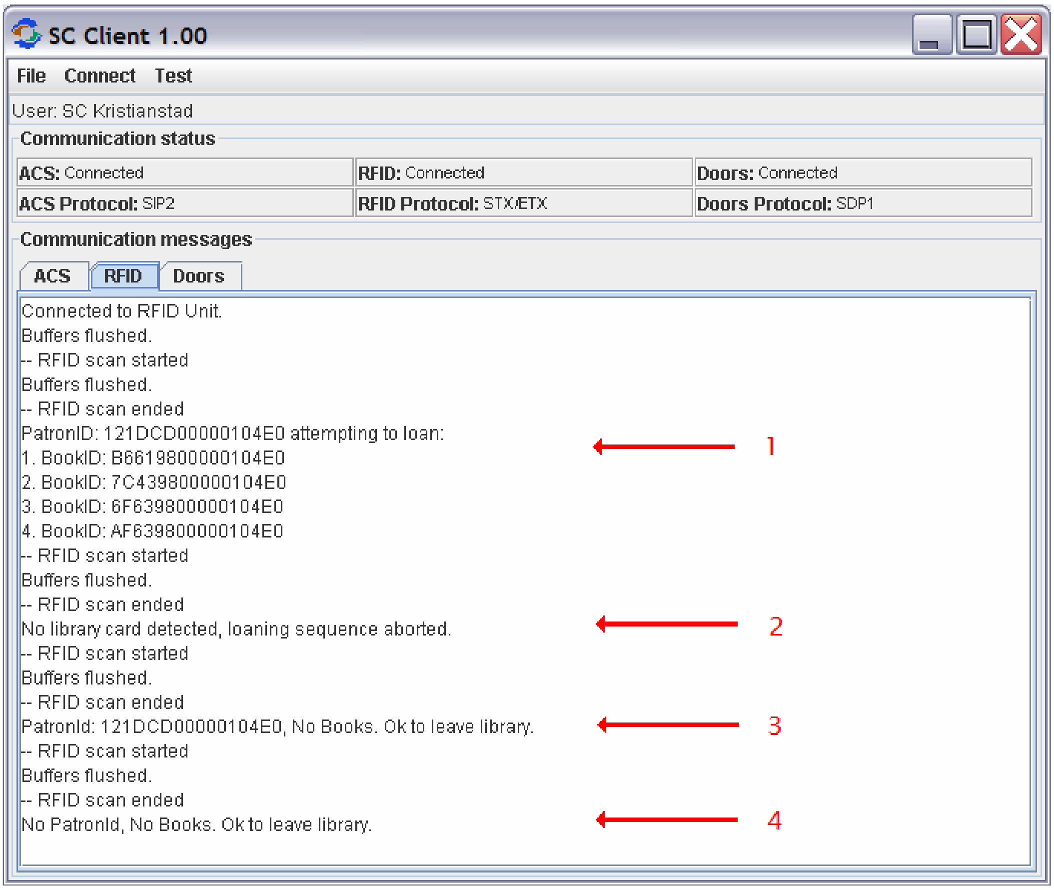


Figure 54 - RFID sequences

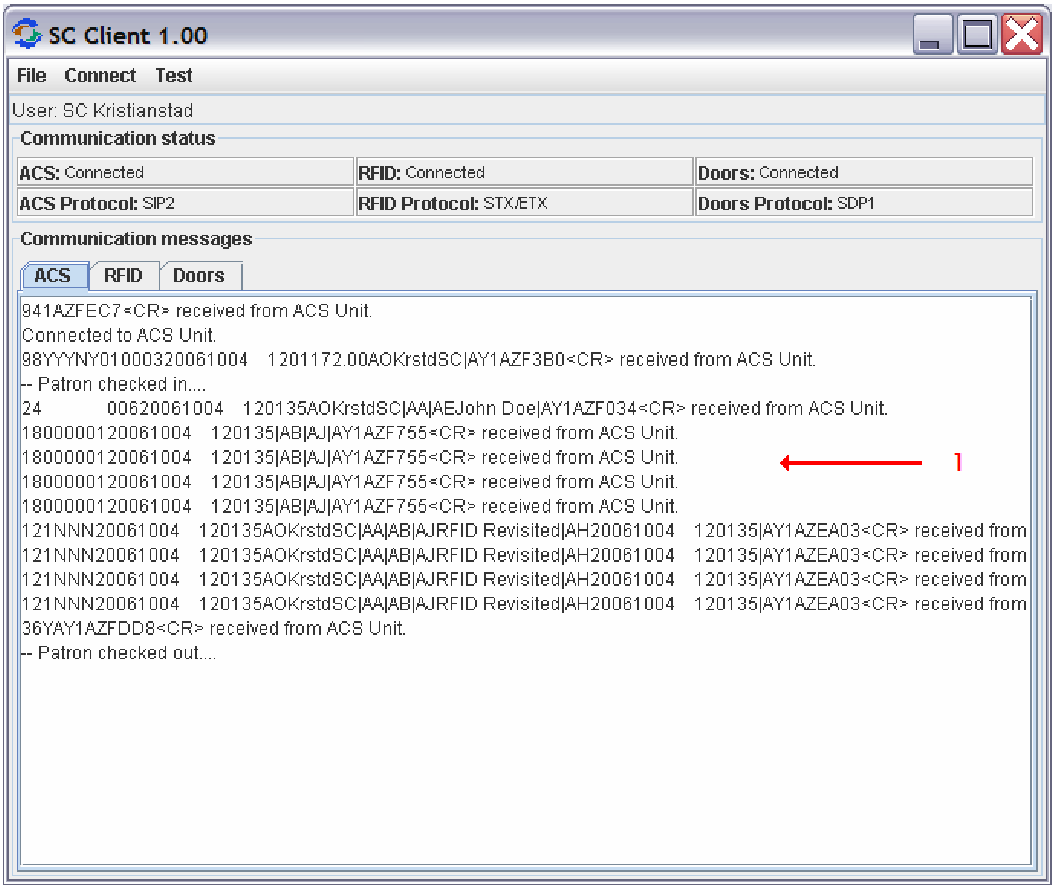


Figure 55 - ACS sequences

8.3.1 Scenario 1 – Successful loaning sequence

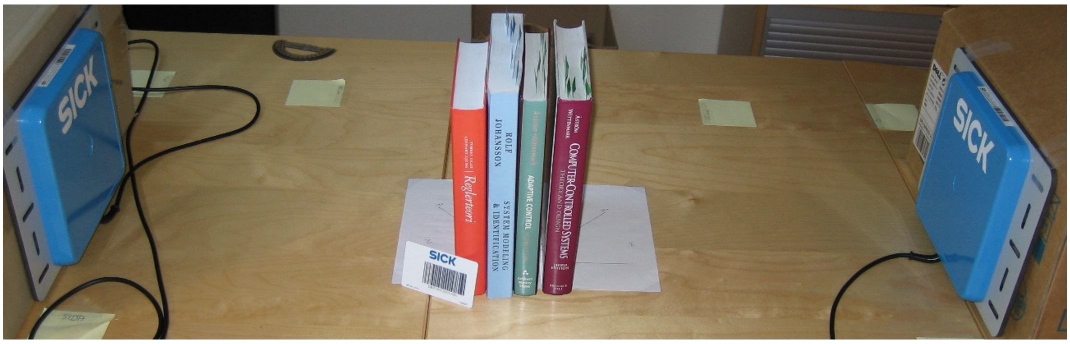


Figure 56 - Books and library card

When the doors has closed the RFID scan has ended and as seen in figure 54 one patron (library card) and four books has been identified, these are now checked by the ACS, see scenario 1 figure 55, to determine if one is allowed to loan them or not. In this case it was clear to loan the books and as seen in figure 53 under scenario 1 the doors open allowing the person to leave the library and then the doors return to the initial state.

In this case there is also a receipt printed allowing the loaner to see which books that has been loaned and the due date for them.

8.3.2 Scenario 2 – Books but no library card

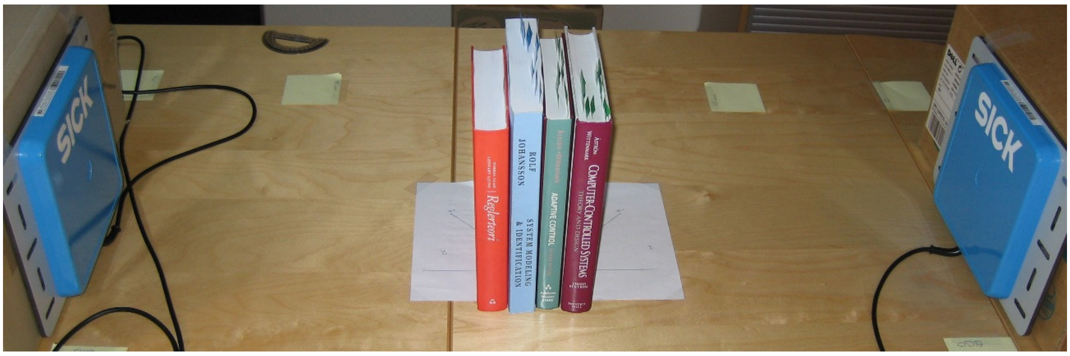


Figure 57 - Books but no library card

When the doors have closed the RFID scan has ended and as seen in figure 54 under scenario 2 books has been found but no library card. The doors then open again, see scenario 2 figure 53, allowing the person to enter the library again but not leave it.

8.3.3 Scenario 3 – Library card but no books

Figure 58 - Library card and no books

When the doors has closed the RFID scan has ended and as seen in figure 54 under scenario 3 a library card has been found but no books. Since it is legitimate to leave the library the person is let out of the library and the doors return to the initial state, as seen in figure 53 under scenario 3.

8.3.4 Scenario 4 – No library card and no books

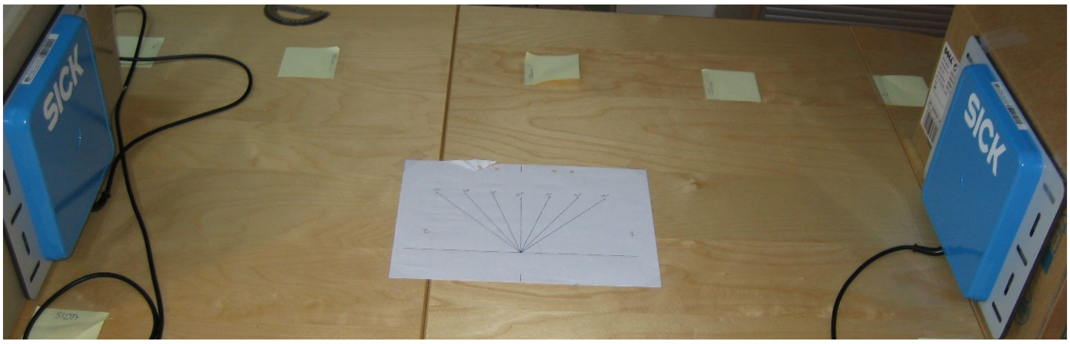


Figure 59 - No books and no library card

When the doors has closed the RFID scan has ended and as seen in figure 54 under scenario 4 no library card and no books has been found. The person is now let out of the library and the doors return to the initial state, as seen in figure 53 under scenario 4.

9. Conclusions

The developed system is a fully functional model of how to integrate RFID into the library industry.

There is a complete setup of components to use as well as how to interconnect them. A software has been developed communicating with the main library database using the SIP2 [6] protocol and communicating with the interrogator using STX / ETX protocol. The software is prepared to handle different protocols communicating with the database. This has been achieved by creating an interface for the protocol and one just have to implement the intended functions and it works with any desired protocol (described under "Software").

It has also been proven that the design of the doors and the placement of the antennas should be sufficient to scan all tags regardless of the angle between the tag and antenna. This since the critical angle is 45^{o} and the antennas still support reading distances up to 1 meter which should be sufficient.

Still there is a lot of testing to be conducted before this project is complete. For example one has to build a complete 1 to 1 scale model of the system and test it. One also has to test the best scan approach, using the phase shift method for top-bottom scanning or check if it is sufficient to scan with the two antennas at the top and bottom.

The described difficulties with the scan angle has been solved and proved that the intended hardware is sufficient to solve this problem and scan all tags. The only thing necessary is to order custom made antennas to fit the height of the doors but that should not be a problem.

To summarize all of this one can say that it is a fully functional model of how to incorporate RFID into the library industry, all tests done has proven this. The software has been implemented and tested as well as choosing and configuring the hardware components. Now the conclusion is drawn that the purpose of this thesis (described under "Introduction") has been accomplished.

10. References

10.1 Internet

[1] The history of RFID, RFIDJournal
http://www.rfidjournal.com/article/articleview/1338/1/129 (2006-08-14)

[2] Wireless Technology: What is RFID?, Pacific Northwest National Laboratory
http://www.pnl.gov/news/back/wirelessoverview.stm (2006-08-14)

[3] RFID: Solutions without a Barrier, California Software Labs
http://www.cswl.com/whitepapers/rfid-technology.html (2006-08-14)

[4] Glossary of RFID Terms, RFIDJournal
http://www.rfidjournal.com/article/articelview/208 (2006-08-14)

[5] Brave New World of Radio Frequency Identification, Lai Ying Cheung & Joshua Lee & Yuan Horng
http://www.dsta.gov.sg/DSTA_Horizons/2005/08.htm (2006-08-14)

[8] Electronic Product Code, EPCGlobal US
http://www.epcglobalus.org/Network/Electronic%20Product%20Code.html (2006-08-15)

[9] Frequently Asked Questions, EPCGlobal Inc.
http://epcglobalinc.org/about/faqs.html (2006-08-15)

[12] RFID Vendors Unite to Promote UHF for Items, Beth Bacheldor
http://www.rfidjournal.com/article/articleview/2412/1/1/ (2006-08-16)

[13] Radio Frequency Identification
http://en.wikipedia.org/wiki/rfid (2006-08-16)

[14] Java Technology
http://java.sun.com (2006-08-18)

[15] SICK Sensor Intelligence
http://www.sick.se (2006-08-18)

[16] ISO Standards, IHS / Global
http://global.ihs.com (2006-08-18)

[17] Automation Light Grids, SICK Sensor Intelligence
http://ecatalog.sick.com/Products/ProductFinder/product.aspx?finder=Produktfinder&pid=18570&lang=en (2006-08-18)

[18] 13,56 MHz - ISO 15693 compatible transponder - RFT331-1310, SICK

Sensor Intelligence
http://www.sick.com/home/factory/catalogues/auto/rfid/rftxxx/en.html (2006-08-22)

[21] Frequently Asked Questions, RFIDJournal
http://www.rfidjournal.com/article/articleview/207 (2006-08-29)

[22] Technical report on RFID tag study, Mr Tan Jin Soon
http://www.itsc.org.sg/synthesis/2000/itsc-synthesis2000-jinsoon-rfid-tag-study.pdf (2006-09-18)

[24] ICT Technologies
http://cbdd.wsu.edu/kewlcontent/cdoutput/TR502/ (2006-10-13)

[25] Understanding RFID Frequencies, Bob Scher
http://rfidusa.com/superstore/pdf/Understanding_RFID_Frequencies.pdf (2006-10-13)

10.2 Papers

[6] SIP2 Interface Developer's Guide, 3M Library Systems

[7] RFI341 Radio Frequency Interrogator Telegram Listing, SICK Sensor Intelligence

[10] Auto-ID Across the Value Chain: From Dramatic Potential to Greater Efficiency & Profit (2002), Ajit Kambil & Jeffrey D. Brooks

[11] The use of Radio Frequency Identification as a Replacement for traditional Barcoding (2004), Gaurav Dargan, Brian Johnson, Mukunthan Panchalingam, Chris Stratis

[16] RFI341 Radio Frequency Interrogator Product Information, SICK Sensor Intelligence

[19] RFI341 Radio Frequency Interrogator Operation Instructions, SICK Sensor Intelligence

[20] Pick2Light Technical description, SICK Sensor Intelligence

[23] HF Antenna Design Notes, Texas Instruments

Appendix A – Software user manual

SC Client 1.00

User manual

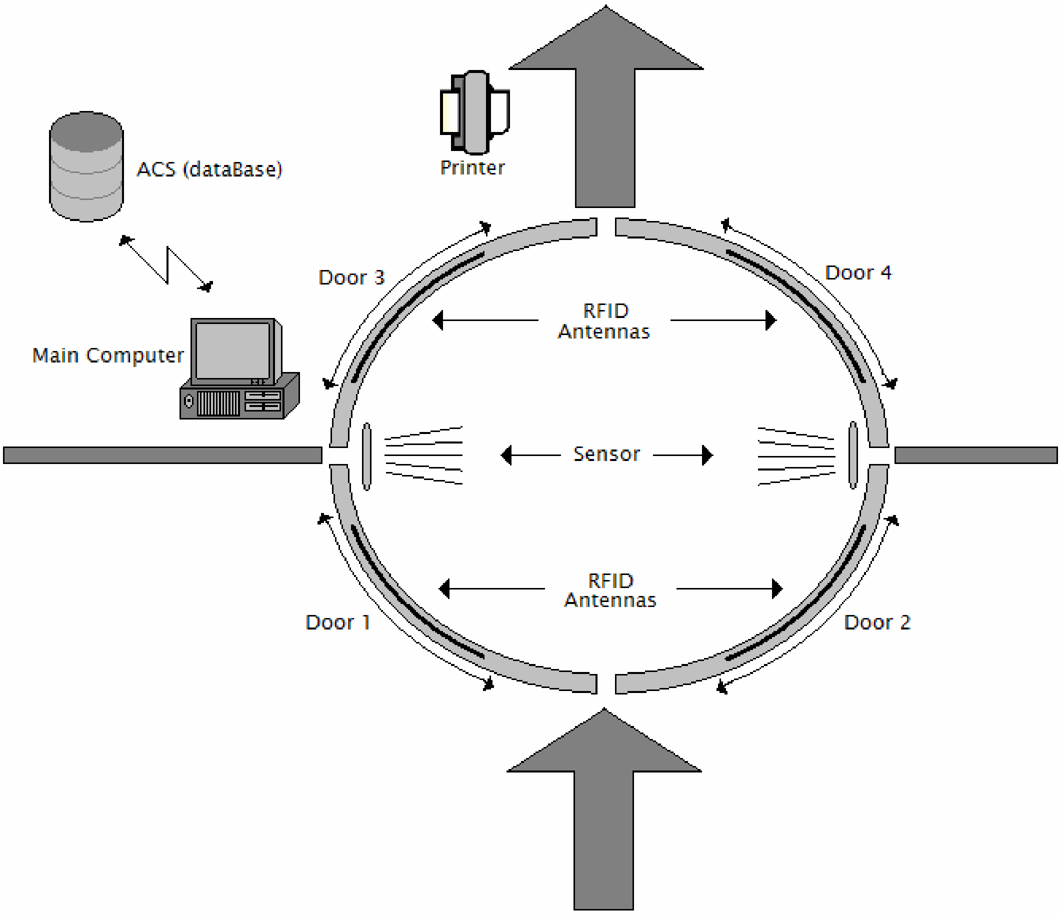
Version 1.0

Date: 2006-08-22

1 Contents

2 The system

The system has the following design:



A typical sequence loaning a book is the following (observer that door 1 and 2 are initially open while door 3 and 4 initially are closed):

1. A person walks through door 1 and 2 triggering the sensors which notifies the client that someone wants to leave the library.
2. Door 1 and 2 starts to close.
3. The interrogator starts scanning for tag data and sends the data to the client.
4. The scanning is stopped when door 1 and 2 are closed.
5. The received data from the tags are checked by the main computer determining if the checkout was successful or not.
6. In case of a successful checkout door 3 and 4 opens allowing the person to leave the library, in the opposite case door 1 and 2 opens again for the person to re-enter the library.
7. A receipt is printed containing data about the loaned books.
8. The system returns to its initial condition.

To easily follow the sequence of actions look under the flaps shown in the interface, more detailed description later in this document.

3 The client

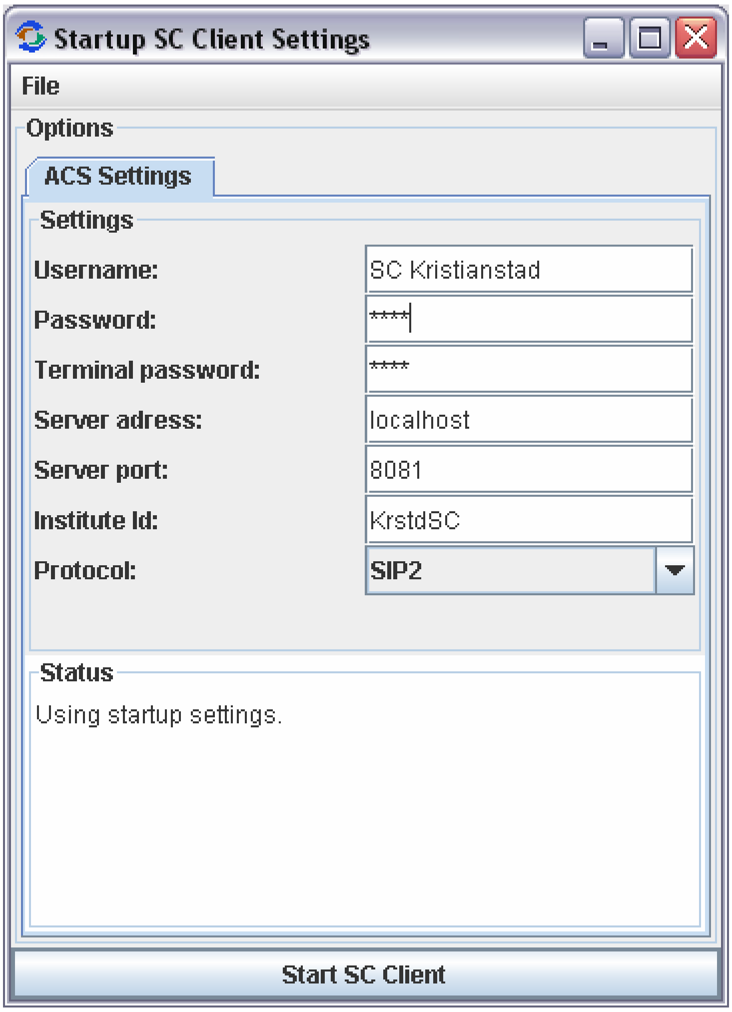
Below there is a description of how to start the system as well as how to configure it for a specific library.

3.1 Settings in the system

Before starting the system it is important to check that the RFID interrogator is turned on and the doors are powered up.

When the system starts it's time to type in all the necessary parameters needed to establish a connection between the client and the main library database. The database keeps track of all the books.

The first window shown is the following:

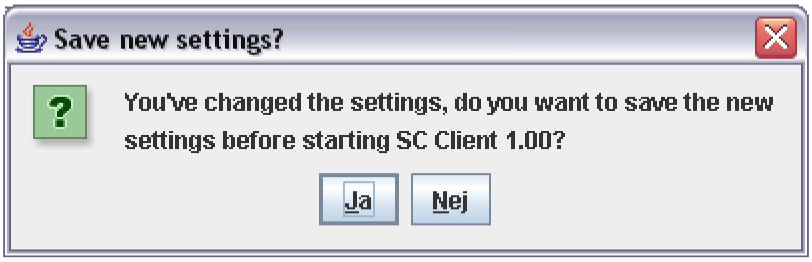


Here one is suppose to type in the following:

1. Username – Username for the library database
2. Password – Password for the library database
3. Terminal password – Password for the current unit
4. Server address – Address to the library database (for example 127.0.0.1)

5. Server port – The port used by the library database
6. Institute Id – Id for the specific library
7. Protocol – The protocol used for communicating with the library database

The settings entered in this window are saved, i.e. one doesn't have to re-enter them at each start-up of the system. To save the setting one just has to click on File → Save. If one change the settings without clicking on Save the following window appear:



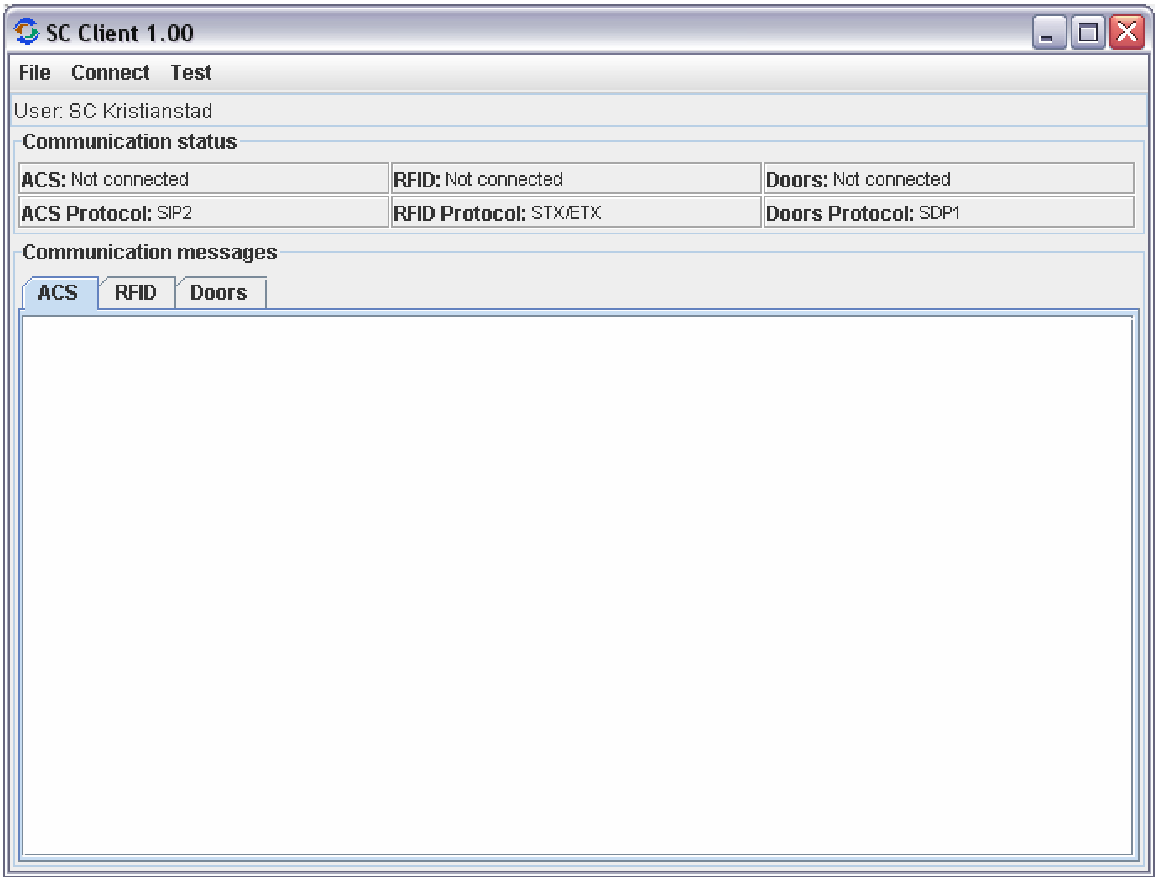
The user is now allowed to choose if the settings should be saved or not.

Observer that the password to the library database isn't saved due to safety reasons.

Now that all the settings have been entered it's time to connect all units to the client, read more about this in the following chapter.

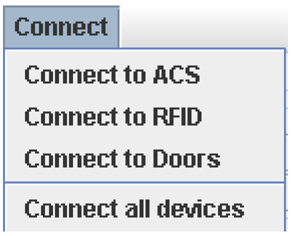
3.2 Start-up of the client

Now it's time to connect all units to the system which is done the following way. First the following frame on the screen is presented.



Under the label "Communication status" one can see that none of the units have been connected to the client.

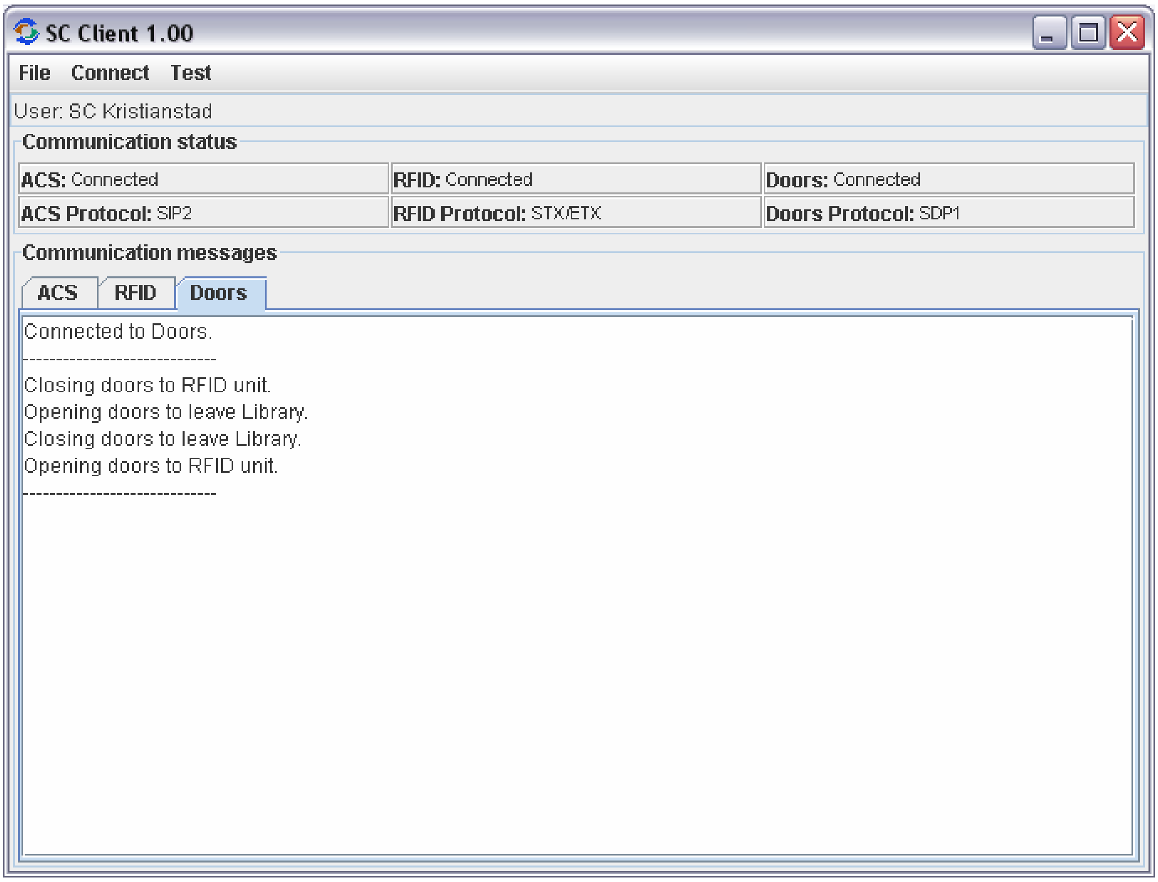
By clicking in the menu connect, see picture below, one can connect to each device individually or to all of them at once.



Clicking on "Connect all devices" connects to all devices needed to run the system.

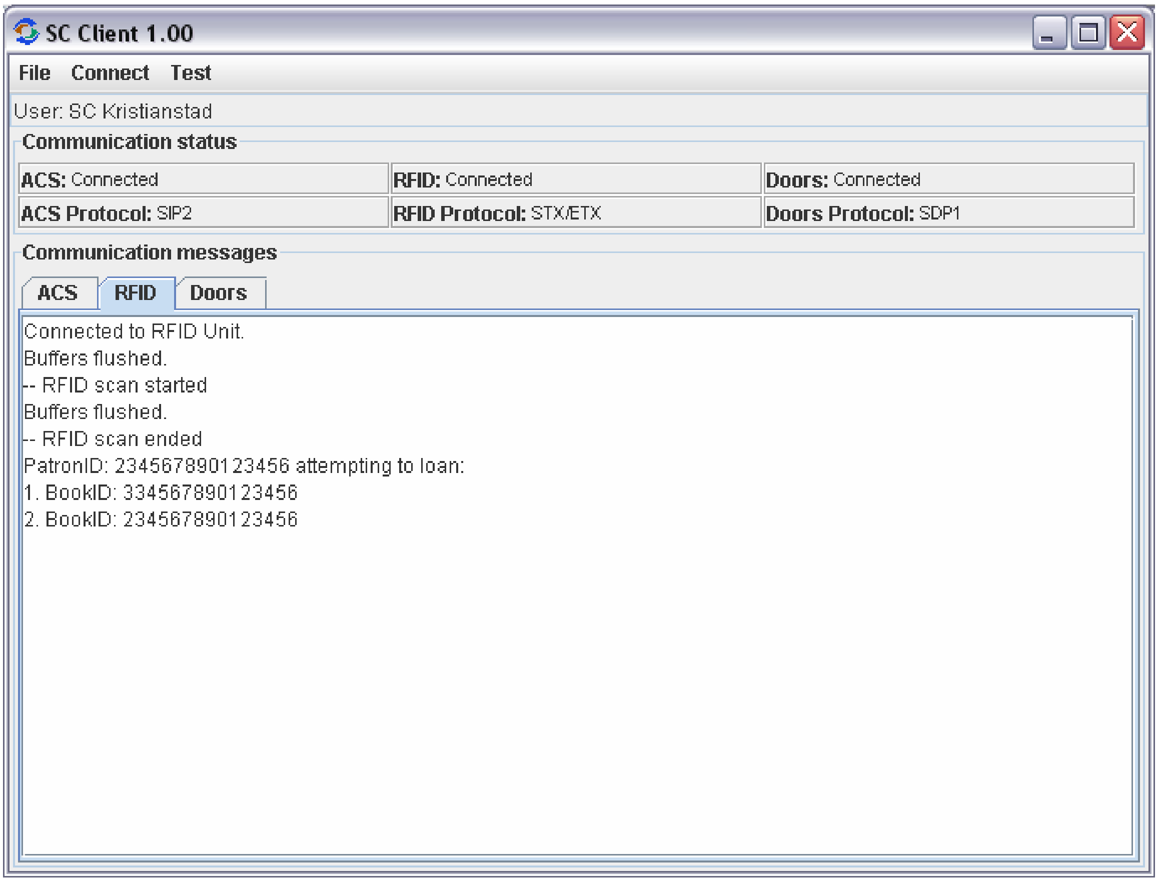
When all the units have been connected it's typed "Connected" after all the units presented in the "Communication status" field. In case of not all units connected check under the respective flap for the error message and then consult this manual to determine the fault.

When a correct loaning sequence has been conducted it looks the following. First shown is how the doors work.



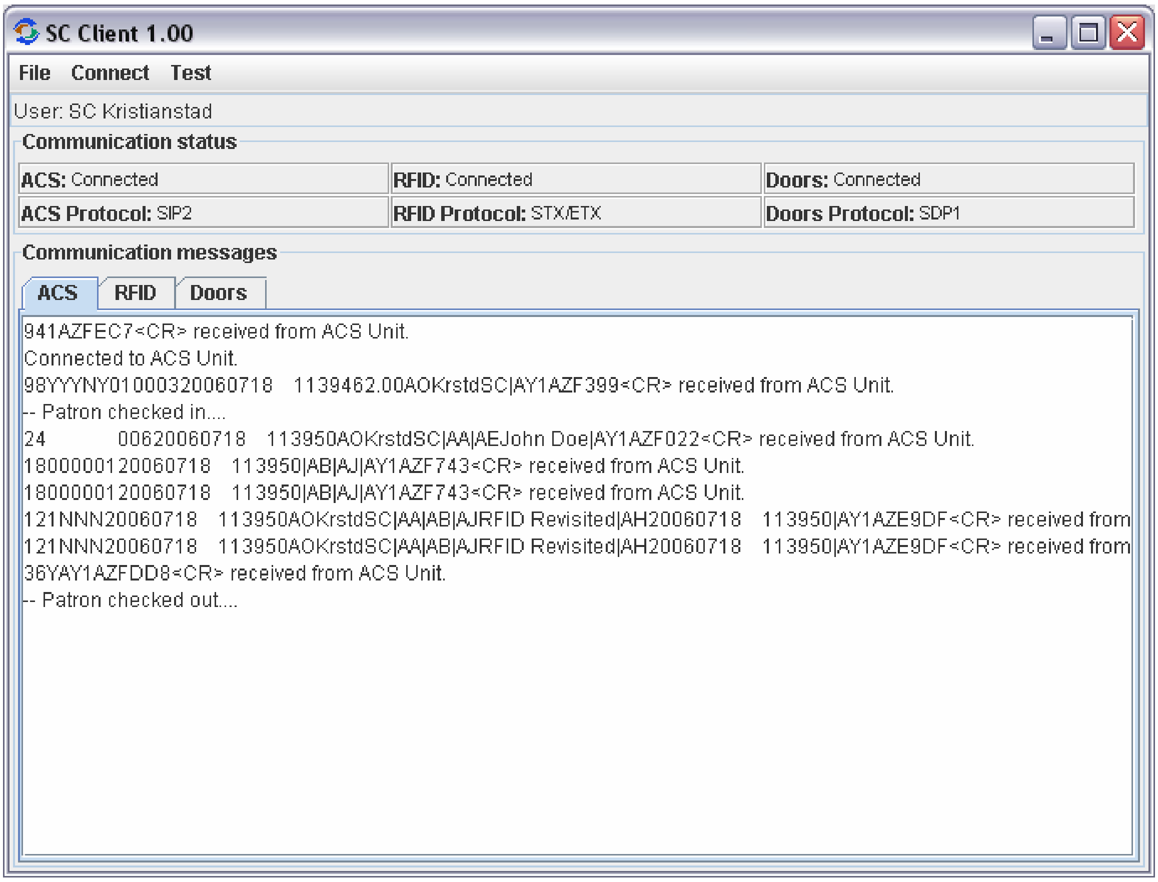
First one can read that the doors where successfully connected to the client and then one can follow how it operates in real time.

For the RFID unit it looks the following way:



Here one can read that the unit was successfully connected to the client as well as when the scan started and ended. The identification number of the library card and the books are also shown.

Finally one can also follow the communication between the client and the library database under the flap ACS which has the following form.



Here one can read that one's connected to the library database and a sequence of messages between the client and the library database. In the example above the protocol SIP2[1] is used.

[1] Standard Interchange Protocol Version 2.00, 3M Library systems™

4 Logging of data

When a loan pass through this unit data about the loan is stored. The log-files are named by the date of the loan, for example 20060617.txt corresponds to events logged on the 17th of June.

The log-files can be found under the following path:

- \SCApplication\logdata\<date>.txt

The format on the files looks as below:

```
Date;Patron ID;Book ID:s -->;
20060718 09:54:07;234567890123456;334567890123456;234567890123456;
```

As seen the fields are divided by semicolon (;) making it possible to import this file to for example Microsoft Excel[2]. In this way one can easily present data about the use of the unit on an easily read format for example the following:

Date	**Patron ID**	**Book ID:s -->**	
20060718 10:02:23	234567890123456	458535254523155	845628514235534
20060718 10:03:59	259845161523154	456845354862852	
20060718 10:04:19	156545631251351	456312351535532	245453155312211
20060718 10:09:52	368123531523513	512515155310544	821562351232537
20060718 10:12:30	864153812124523	864528658413231	

First column corresponds to the date and time of the loan, second column corresponds to the library card Id and the other columns correspond to the Ids of the loaned books.

[2] Microsoft Corporation, http://www.microsoft.com

5 Error messages

Below different possible error messages are shown for the respective flap.

5.1 ACS

5.1.1 ERROR 001: Unable to establish socket connection

The client could not connect to the library database, control that the library database is powered up and the network connection is up and running.

5.1.2 ERROR 002: Unknown host

The client could not connect to the desired port, check that entered address and port to the library database is correct.

5.1.3 ERROR 003: Retried out

Communication with library database broken, tried to resend the message without answer. Control that the library database is powered up and the network connection is up and running.

5.1.4 ERROR 004: Wrong password or username

The entered username or password was incorrect, restart the program and enter correct data.

5.1.5 ERROR 005: Connection broken

Connection to the library database broken, try to reconnect. If unable to reconnect control that the library database is powered up and the network connection is up and running.

5.2 RFID

5.2.1 ERROR 001: Unable to open port. Might already be in use

Couldn’t open serial port com1. Control that no other applications use the specific port.

5.2.2 ERROR 002: Port does not exist

Serial port com1 does not exist, control that the interrogator is connected to com1.

5.2.3 ERROR 003: Unable to configure serial port connection

Couldn’t configure the serial port, pleased restart application and retry.

5.2.4 ERROR 004: Unable to open connection

The application could not connect to the interrogator. Control all settings, restart the application and retry to connect.

5.2.5 ERROR 005: Multiple library cards detected

Multiple library cards detected by the scan, loaning sequence aborted.

5.2.6 ERROR 006: Write error 6C20 / 6C21

Couldn’t start a scan with the interrogator.

5.2.7 ERROR 007: Read error 6C20 / 6C21

Couldn’t read scanned tags from the interrogator.

5.2.8 ERROR 008: Unable to flush buffers 1018

Couldn’t flush the buffer on the interrogator, could result in faulty check out. Interrogator I shut down.

5.2.9 ERROR 009: Connection broken

Couldn’t communicate with the interrogator, try to reconnect.

Appendix B – Terminology

RFID
Radio Frequency Identification, making it possible to identify items from distance.

Tag
A transponder containing data about the item. The tag is placed on the item one want to identify.

Interrogator
The master of the RFID system which sends questions through the antenna to the tags and then translates the response received by the antenna. The interrogator is also called the RFID reader.

SIP2
Standard Interchange Protocol V2.00 which is the protocol used for communicating with the main library database.

STX/ETX
Start – of – text / End – of – text which is the protocol used to communicate with the RFID interrogator.

EPC
Electronic Product Code which is the next generation of bar codes.

UHF
Ultra High Frequency, used by the RFID tags operating at frequencies between 866 MHz and 960 MHz.

UML
Unified Modelling Language, shows relations between different classes in the source code.

ACS
Automated Circulation System, the database containing data about all books and persons with library cards.