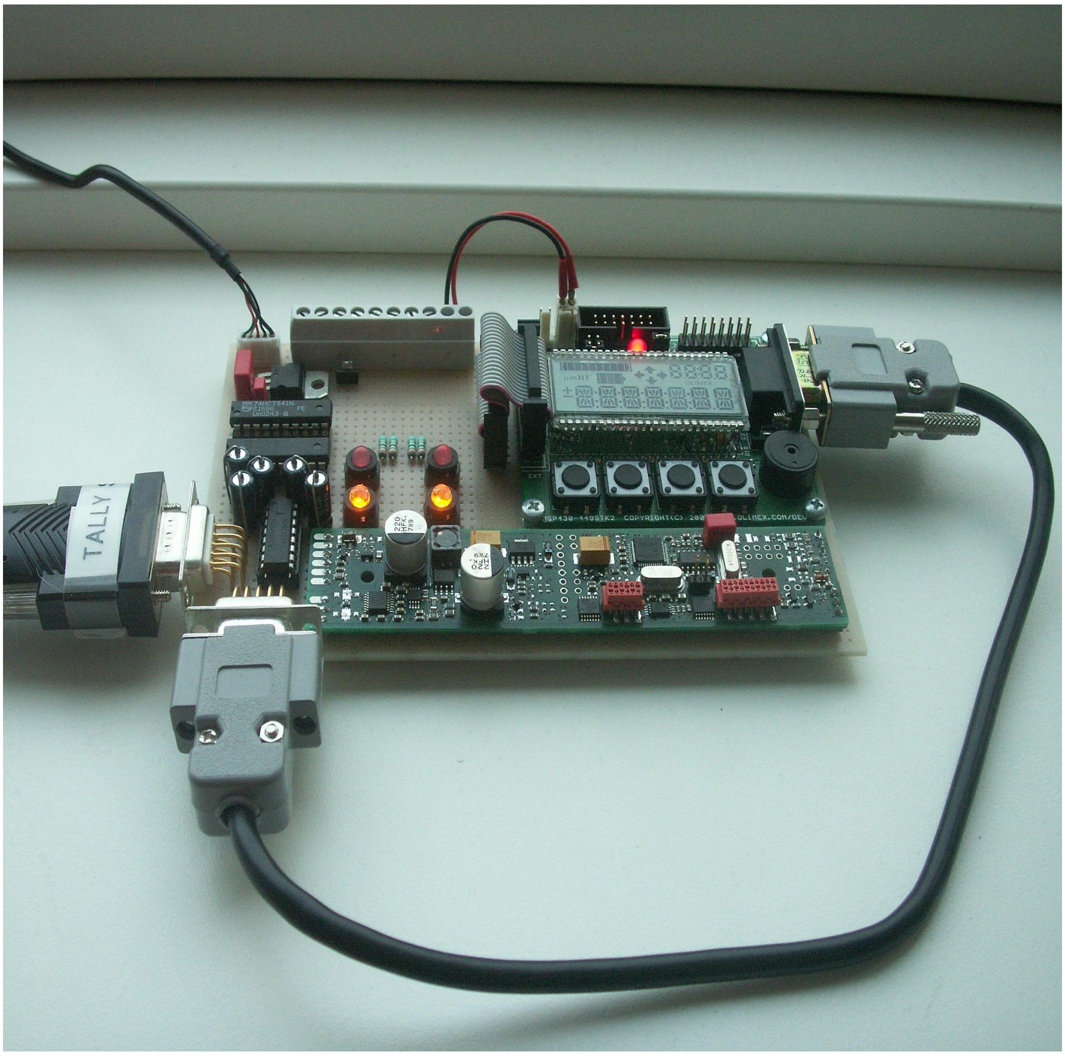
Development of the Tally Station

(for use in the existing offshore Tally System)



Written by: Admir Habota
Project period: October 2008 - January 2009
Due date: January 8th 2009
Supervisor at AAUE: Mr. Zhenyu Yang
Supervisor at ABB: Mr. Jan Sørensen

Titlepage

Title: Development of the Tally Station (for use in the existing offshore Tally System)
Theme: Thesis project
Written by: Admir Habota
Due date: January 8th 2008

Abstract:

The purpose of this project is to develop the new Tally Station with focus on embedding the components and possibly gaining the Ex certification. The Tally Station will be used in the existing Tally System which is implemented offshore.

The Tally System is implemented to keep an eye on the people-on-board count when people are moving around the platforms. Due to the lifeboat capacity limitations, the system is installed to count the number of people in any area at any time. If the number of people in an area has reached the limit of lifeboat capacity, the system will turn on red lights as a warning, if a new person tries to enter that area.
The Tally Station is part of the Tally System and its purpose is to register a person when passing by, and also inform a person if an area has reached its capacity limit.

Registration of people is done using RFID and the data is processed by the MSP430F449 microcontroller which serves as the CPU of the Tally Station. Communication with peripherals and the server is routed by the MCU, and the communication protocol has been developed for this purpose.

Using software developed by ABB it is possible to monitor entire Tally System and locate any individual at any time. This is useful as an extra precaution and when checking if all the people have been evacuated in case of emergency.

Supervisor at AAUE: Mr. Zhenyu Yang
Supervisor at ABB: Mr. Jan Sørensen

Preface

This report is written as documentation for Admir Habota's thesis project in cooperation with ABB Corporation in Esbjerg. The project is chosen based on student's interests which are primarily data communication and software development, and on ABB's proposal for the student thesis project.

The project is based on development of the Tally Station which is part of the Tally System used as a safety precaution offshore. One of the goals for the project besides development of a prototype is examining possibilities for gaining the Ex certification for the Tally Station. To avoid confusion the Tally Station will at times be referenced to as the Tally Station System or TSS throughout this report.

This report is intended for supervisors, examiner, fellow students and everybody with interest in hardware and software development based around microcontroller as a central unit of the system.

This report is divided into several parts. Project analysis part focuses on preliminary identification of eventual problems and needs for the system. It includes amongst others the Tally System description and requirements specification.
The design part focuses on hardware/software design of components which make out the system. This part includes description for amongst others MCU interfacing, peripherals and design of the communication protocol.
The implementation part describes how the designed components are implemented. This goes for both hardware and software.
The summary part is self explanatory as it contains the conclusion for the whole project period.
In the end of this report there is an appendix which contains drawings and various information relevant to the project. It contains amongst others explanation of various acronyms used throughout this report.

I would hereby like to thank ABB Corporation for giving me opportunity to work on this project, and entire ABB Esbjerg staff for supporting me throughout my project period. Special thanks to my supervisor Mr. Jan Sørensen.

Esbjerg, January 2008

Admir Habota

Contents

1 Introduction

Working offshore on oil platforms includes all kinds of risks, and there is a continuous need for improvement in safety precautions in order to limit loss of human lives in case of an accident. One of improvements in the safety area is implementation of the Tally System. As the name implies the Tally System counts number of people in an area at any time, and its purpose is to limit number of people in an area to the capacity of the lifeboats.

This project focuses on development of the new Tally Station which is part of the Tally System. People interact with the Tally System through the Tally Station. The purpose of the Tally Station is to register people when passing by and at the same time warn them if the area person is trying to enter has reached its capacity limit.
At the moment there is installed an old version of the Tally Station which needs improvement in several areas. Main improvements are production related cost reduction and several communication issues which include person identification and communication routing.

The new Tally Station is based on the old one, but has different overall structure, components and communication standards. Registration of people is done using RFID as in the old version. It is a reliable technology, and fullfills requirements for access control. There is no need for advanced access security as the interest is only to track number of people entering or leaving an area and not securing an area for unauthorized personnel.

To be installed offshore the Tally Station has to be Ex certified. Therefore it is also part of this project to analyze needs for Ex certification because it is one of the side goals to get the new Tally Station prototype certified. Some information about Ex certification is found in the appendix, but the primary goal is to develop the working prototype.

Part I

Project analysis

2 Problem description

As mentioned before The Tally System is used as a safety precaution and its purpose is to keep an eye on the people-on-board count when people are moving around the platforms. If the number of people in an area has reached the limit of lifeboat capacity, the system will turn on red lights as a warning, if a new person tries to enter that area.
Currently there is implemented an older version of the Tally Station offshore, and there is need for improvements in several areas such as cost reduction, functionality, etc. Old Tally Station can be seen on figures 2.2 and 2.3. The Tally Stations are placed at the crossings between two areas, and to get an idea this is shown on figure 2.1. The Tally Stations are marked with red ellipses.

Figure 2.1: Locations of the Tally Stations.

All equipment installed in hazardous areas offshore has to be Ex certified which means that the equipment is explosion-proof and fulfills some standardized requirements. This is also required for the Tally Station. The old Tally Station is made of several conventional components, and to gain Ex certification these components have been boxed in a very thick safe-like metal box which is Ex certified. The closed box is shown on figure 2.3.
Other components such as display and RFID antenna are Ex certified and are therefore located outside the box. Ex approved components are generally much more expensive

than regular components of the same type.

Figure 2.2: Old Tally Station implemented offshore.

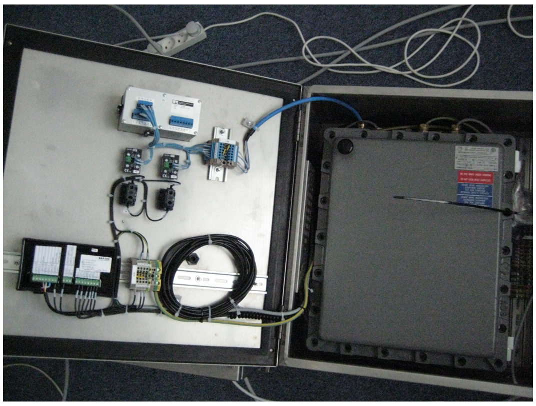


Figure 2.3: Old Tally Station inside view.

The new Tally Station is based on the old one, but has different structure and different components are used. An example is the use of microcontroller as the CPU, and RS485 serial communication instead of the Ethernet. This is further described in the design part of this report.

To cut the costs one of the goals is to make new version of the Tally Station smaller and more compact. This is in case that the new version also has to be boxed to gain Ex certificate.

As mentioned before there are some functionalities that need improvement. Currently the Tally Station is polling constantly to check if there is a card close to the RFID antenna, and also pooling to check if communication with the server is functional. Possibilities to implement these as the interrupt controlled sequences will be analyzed to avoid waste of CPU resources.

All the changes and new functionalities are described in the design and implementation parts which should give an idea how the new TSS is built.

3 Requirements specification

This chapter describes requirements for the Tally Station System (TSS). As mentioned before the primary goal is to develop a working prototype and this project is mainly focused on that goal. Secondary goal is to get the prototype Ex approved and that part will be on hold until the first goal is fully achieved. The new Tally Station should be implemented in the existing Tally System without major modifications or change of equipment such as servers, routers etc.

Requirements for the Tally Station:

CPU

The Tally Station should have a CPU which is responsible for data processing, communication routing etc. CPU should be able to communicate with the server when sending or receiving requests, and at the same time communicate with the RFID card reader. CPU should also be able to interface with other peripherals such as warning LEDs, user interaction buttons, etc.

Read the RFID card

The Tally Station should be able to read the RFID cards which are currently used offshore. It is much easier to use existing cards which all employees already posess, than to use new form of tags, or new identification method.

Communication with the server

The Tally Station should be able to communicate with the server. This includes sending RFID card data to the server, and receiving commands from the server. Distance from the server to a Tally Station must also be taken into consideration.

Display

The Tally Station should have a display for showing information to the users. The display can be used for various purposes but one of these should be to inform a person of the current number of people in an area the person is trying to enter.

User interaction

There has to be a way to enable users to interact with the system. This can be used for different purposes, but an example is if an area a person is trying to pass through is full, then a person cannot pass through that area and has to wait. By interacting with the system a user can "tell" the system that he is just passing through. The system should then wait some time for the same person to register elsewhere, or otherwise alarm the administrator.

Warning system

When a person is trying to enter an already filled area there has to be some kind of warning system to warn a person about this. There should also be a warning system to warn an administrator if there are for some reason too many people in an area, or if a person that should just be passing through has not registered elsewhere.

Communication watchdog

There should be implemented some kind of watchdog to monitor communication between the server and the Tally Station. If the communication cannot be established or becomes broken the watchdog should alert both the system administrator via terminal software and the Tally Station users by writing on the Tally Station's display.
Based on the system requirements figure 3.1 shows block diagram on how interaction between different blocks should look like. Later in the design part a components for each block are specified where a new block diagram is drawn based on the component interfacing.

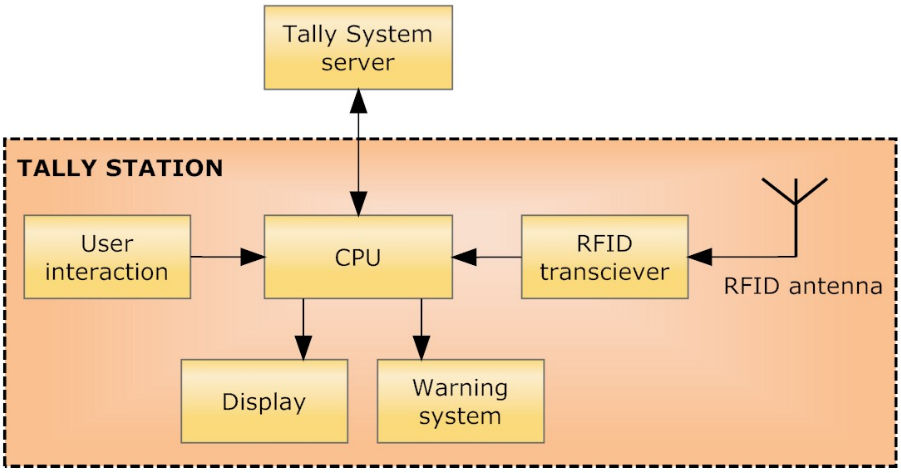


Figure 3.1: Block diagram of interaction between components based on system requirements.

4 Tally System description

References: [2] and [1]
To gain an overview of the Tally System it will be shortly described in this chapter.

Figure 4.1 shows overall structure of the Tally System which includes two Tally Stations and a single client PC.

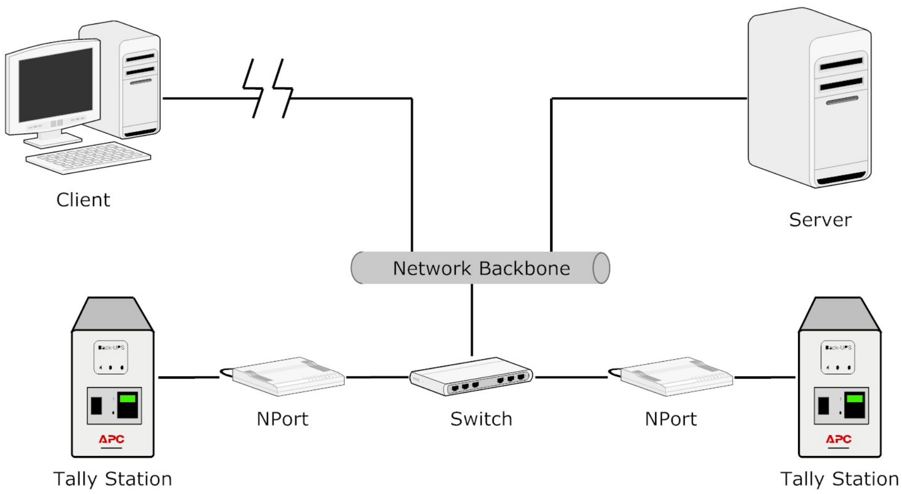


Figure 4.1: Structure of the Tally System.

The Tally System has a central server which holds information about all installed Tally Stations, areas, client computers and people working offshore. All information is stored in a SQL database on the server and can be accessed at any time. When a person is coming to work offshore he is created in the system, assigned a living cabin and a life-saving boat in case of an emergency. The person is also given a personal RFID card which is used as identification at Tally Stations. For system setup and creation of users ABB has developed a software application called TallySys Manager. This application is also used when creating/adding a Tally Station and/or a client computer in the Tally System.

A client computer is running a TallySys Client software application which is administrative user interface for the Tally System. The client computer is mainly used for surveillance of the Tally System, but can also be used to create users, locate users, access logs, etc. A client PC can be located either offshore or onshore and a user can access the server using a remote console application. There can exist multiple client computers in a single Tally System.

Another software application called TallySys Server is running on the server and handles communication between the server and the Tally Station.

As seen on Figure 4.1 Each Tally Station is connected to an NPort. NPort enables access to serial devices through the Ethernet. Each NPort is connected to a switch which enables a use of multiple Tally Stations which are connected to a single server.

Part II

Design

5 Identification of components

This chapter focuses on identification/selection of hardware components based on the requirements specification.

Components which must be selected include:

- CPU
- RFID reader and antenna
- Display
- User interaction components
- Warning system components

CPU

To save hardware building time due to the short project period the decision was made to use a development board with a microcontroller from MSP430 family. MSP430 family of microcontrollers is primarily chosen due to student's previous experiences, and there should be no problem finding a MCU that can achieve the system requirements. After analyzing several development boards with help from the supervisor at ABB the choice fell on MSP430-449STK2 development board from Olimex (www.olimex.com). This development board includes MSP430F449 microcontroller which is suited for use as the CPU of the Tally Station.

Some of the features include:

- 16-bit RISC architecture
- 60 KB flash memory
- 2 KB RAM
- two USART
- two 16-bit timers
- 12-bit A/D converter
- Integrated LCD Driver

There are also implemented other components on the development board which can be useful. These include a custom display from Olimex, RS232 interface with DE9 connector, four user buttons and a buzzer. The development board can be seen on figure 5.1.

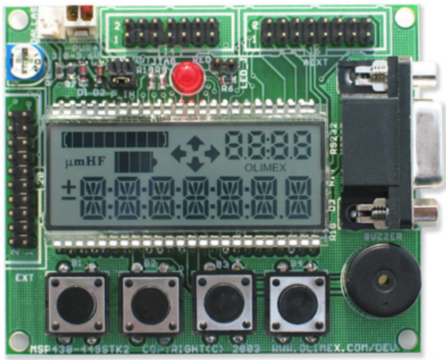


Figure 5.1: MSP430-449STK2 development board from Olimex.

RFID reader and antenna

There is need for an RFID card reader and antenna which can read cards currently used offshore. Currently used cards are of type Hitag1 and operate at 125kHz. A suited card reader and antenna is acquired from Scemtec (www.scemtec.com). The antenna is conventional type RFID antenna and is not Ex approved. Reader module supports most common tags and can operate at two frequencies 125kHz and 134.2kHz. It has RS232 interface which will be used for serial communication with the micro-controller. The RFID reader is shown on figure 5.2.

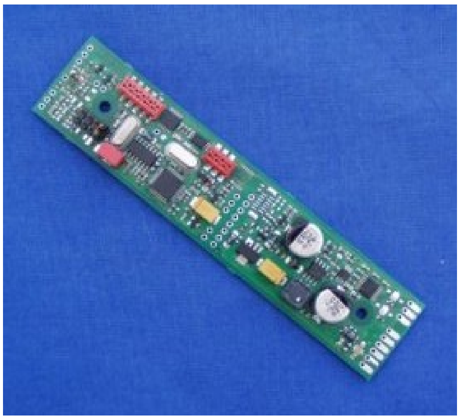


Figure 5.2: STK-5000 RFID reader/writer from Scemtec.

Display

As mentioned in the requirements specification there should be a display to show information to the users. Since the chosen development board has a display included, it is easy choice to use this display in a prototype development. The display can show up to seven characters at a time with 14 segments each. Although it is preferred to have a display with multiple lines which can show more than seven characters at a time, the included display should be enough for this project. The display has also some other features such as clock display, power level display etc., and these features might be used in this project.

User interaction components

Users should be able to interact with the Tally System through the Tally Station, and obvious choice here is to use buttons which are already implemented on the development board. Buttons will allow users to make various selections when using the system.

Warning system components

There has to be implemented a warning system to warn a person when entering an area. This can be for example if an area has reached its capacity limit, if there is an error when reading a card, etc. Since the used development board has the buzzer onboard this will be used along with some LEDs which will be implemented separately.

6 System design

This chapter describes a general system design based on the system requirements and identification of components. Figure 6.1 shows block diagram for the Tally Station System. The figure shows all system blocks and interaction between these. No block is drawn for the power supply and the reason is to make figure 6.1 more clear and understandable, but design for the power supply is described later in the hardware design chapter. Design for the individual blocks/components is also described later in chapters for hardware- and software design.

Figure 6.1 also shows where the components are located physically. As mentioned in previous chapter some components are located on the development board and this is shown as the green box. Components located inside the microcontroller are shown as the blue box. Components such as RFID reader and warning lamps are located on the print board along with other components, and the server is of course located at the remote location.

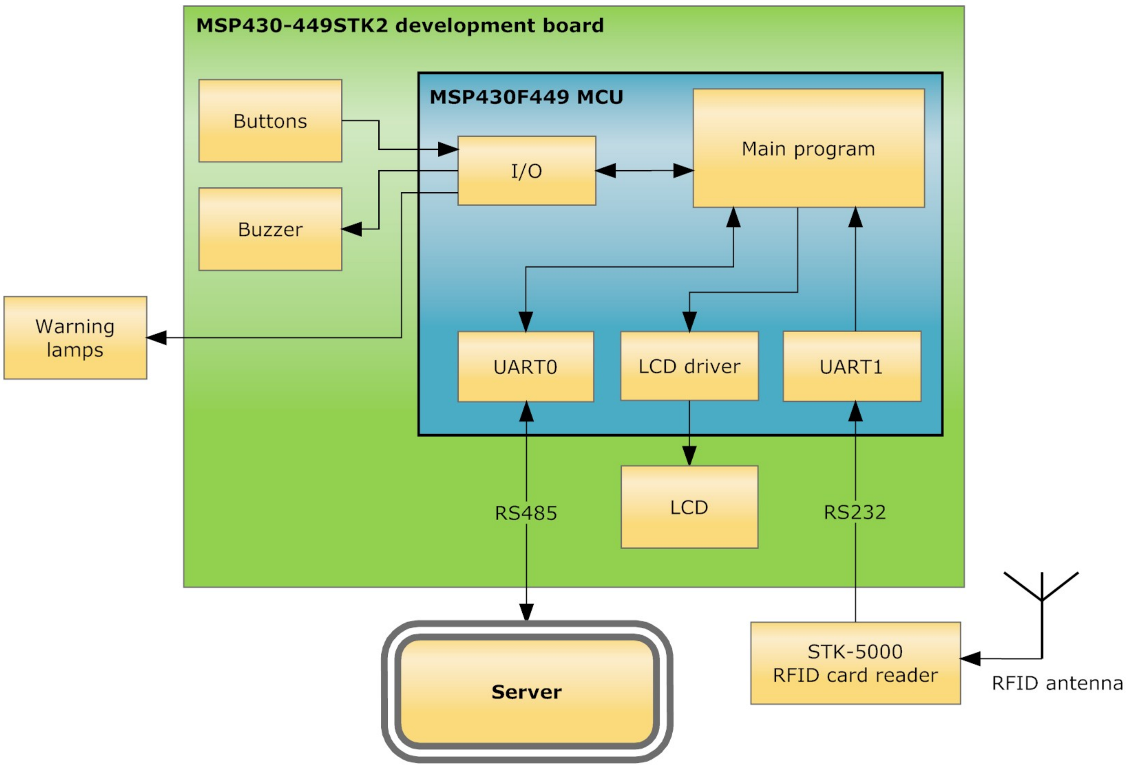


Figure 6.1: Block diagram for the Tally Station System.

7 Hardware

This chapter describes design for the hardware components in the Tally Station System. All important hardware components are described along with description of the interface to the component in question. Interface description is expressed as a table and contains information about input/output signals. Signal names from the interface description are also used throughout the report and on the project drawings.

7.1 Power supply

There is need for different voltage levels depending on the hardware components. RFID card reader and other logic components need +5V DC. The power supply signal for the RFID card reader should be steady with as little ripple as possible. Too much ripple in the supply signal can cause unexpected errors when reading card data.

Power supply for the microcontroller may not exceed +3.6V DC as this will damage the microcontroller. Therefore the microcontroller should be supplied by +3.3V DC which is enough for the microcontroller and other components implemented on the development board. Interface for the power supply is shown in table 7.1. It should be mentioned that there is no description for the input signal for the power supply as this will be described in the hardware implementation chapter.

Input/output	Signal type	Signal description	Signal name
Output	Analog	+5V DC	+5V
Output	Analog	+3.3V DC	+3.3V
Output	Analog	0V	GROUND

Table 7.1: Power supply interface.

7.2 LCD

As mentioned before the display is already implemented on the development board. The display is custom-made from Olimex and is already connected to the microcontroller.

Operating voltage of the LCD is between 2.7V and 3.6V which is fine as the microcontroller should be supplied by 3.3V and the same supply will be used for the display.

The display is software controlled and there are several different modes when driving the display. To avoid confusion an interface for the display will not be described here. Display functions are described in the software design and software implementation chapters.

7.3 User interaction buttons

There are four buttons implemented on the development board and these are used for user interaction with the Tally Station. It is possible that not all four buttons will be used in the prototype. Buttons are already connected to the microcontroller and the interface is shown in table 7.2.

Input/output	Signal type	Signal description	Signal name
Input	Analog	+3.3V DC	+3.3V
Output	Digital	Button 1	B1
Output	Digital	Button 2	B2
Output	Digital	Button 3	B3
Output	Digital	Button 4	B4

Table 7.2: User interaction buttons interface.

7.4 Buzzer

The buzzer is also implemented on the development board and is used as a warning system along with some LEDs. The buzzer should be driven by two digital outputs which need to be toggled to achieve sound. The interface for the buzzer is shown in table 7.3.

Input/output	Signal type	Signal description	Signal name
Output	Digital	Buzzer ON/OFF	BUZZER1
Output	Digital	Buzzer ON/OFF	BUZZER2

Table 7.3: Buzzer interface.

7.5 Warning LEDs

Four LEDs are used along with the buzzer to "tell" a person whether access is granted or denied for an area a person is trying to enter. As the Tally Station is placed between two areas, two LEDs are used for each area to signal whether access is granted or denied for the particular area. Red LED is lit if access is denied and yellow is lit when granting access. The LEDs should be controlled by the digital outputs from the microcontroller. Interface for the warning LEDs is shown in table 7.4.

Input/output	Signal type	Signal description	Signal name
Input	Digital	Left red LED	LED_LR
Input	Digital	Left yellow LED	LED_LY
Input	Digital	Right red LED	LED_RR
Input	Digital	Right yellow LED	LED_RY

Table 7.4: Warning LEDs interface.

7.6 RFID card reader

Reference: [6]
RFID card reader is from Scemtec and should be able to read already existing cards which are of type Hitag1 and operate at 125kHz. A conventional type RFID antenna has also been purchased from Scemtec which should be connected to the antenna port on the reader. Card reader needs steady +5V DC power supply to avoid errors or incorrect readings of card data. This should be accomplished with a voltage regulator, and this is described in the hardware implementation chapter.

The card reader has RS232 interface which should be connected to the already implemented DE9 (also known as D-sub9) connector on the MSP430-449STK2 development board. The DE9 connector is connected to microcontroller's UART1 port. More information is available on a connection diagram TSS-001 found in appendix. Interface for the RFID card reader is shown in table 7.5.

Input/output	Signal type	Signal description	Signal name
Input	Analog	+5V DC	+5V
Input	Analog	0V	GROUND
Input	RS232	Serial communication	RFID_RX
Output	RS232	Serial communication	RFID_TX

Table 7.5: RFID card reader interface.

7.7 Communication

Reference: [3] and [4]
As mentioned before the communication between microcontroller and RFID reader is using RS232 standard. RS232 driver circuit is already implemented on the development board, so there is no hardware design needed for this part of the communication.

The second part is communication between the MCU and the server, and to fulfill system requirements other communication standard has to be used because the server can be far from the Tally Station. The distance can exceed 100 meters so neither RS232 serial communication nor Ethernet can be used. RS232 can have maximum cable length of around 15 meters and Ethernet has reach of around 100 meters. It should be mentioned here that cable length for RS232 can be larger if running at lower baud rates.

After some analysis it was determined that RS485 standard can be used as it can deliver signal as far as 1200 meters from the transmitter. RS485 has also other advantages over RS232 but these will not be mentioned here as the RS485 standard is only chosen because of the signal distance.

To achieve the RS485 communication the initial idea was to implement the RS485 driver on the Tally Station side and use a RS232-to-RS485 converter on the server side, but this would make debugging impossible as a converter is not available at this time. Therefore it has been decided to use the RS232 communication in the prototype to make it possible to debug the system.

There is still possibility of achieving the RS485 communication by installing a RS232-to-RS485 converter on both the server side and the Tally Station side.

To achieve RS232 communication there has to be implemented an RS232 driver which should be connected to microcontroller's UART0 port. There should also be installed some buffers to ensure that appropriate voltage levels are obtained, as it can be dangerous to deliver more than 3.6V to the microcontroller. Block diagram for communication is shown on figure 7.1.

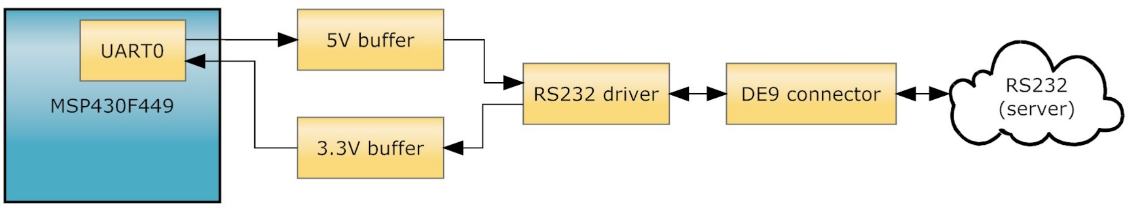


Figure 7.1: Block diagram of RS232 communication between the Tally Station and the server.

7.8 Microcontroller

As mentioned before the microcontroller is the CPU of the Tally Station. It should be able to interact with other peripherals and at the same time communicate with the server and RFID card reader.

Table 7.6 shows an overview of used ports on the microcontroller. Signals such as power supply, ground and LCD connections are not displayed here. Instead refer to drawing TSS-001 in appendix.

Port	Pin no.	Signal type and description	Signal name
1.0	87	Digital output	BUZZER1
1.2	85	Digital output	BUZZER2
1.3	84	Digital output	LED_STATUS
2.4	75	Serial communication output	UART0_TX
2.5	74	Serial communication input	UART0_RX
3.0	71	Digital output	LED_LR
3.1	70	Digital output	LED_LY
3.2	69	Digital output	LED_RR
3.3	68	Digital output	LED_RY
3.4	67	Digital input	B1
3.5	66	Digital input	B2
3.6	65	Digital input	B3
3.7	64	Digital input	B4
4.0	63	Serial communication output	UART1_TX
4.1	62	Serial communication input	UART1_RX

Table 7.6: Overview of the connections on the microcontroller.

8 Software

This chapter describes design for software in the TSS including communication protocol which is used for communication between the server and the Tally Station. Software for most components is described in this chapter. Software for other components such as control of the warning LEDs and the buzzer is not described here as this simply involves sending a digital signal to needed ports and this should be very simple to implement.

8.1 Main program

The main program should be implemented as a constantly running loop which should be interrupted when interrupt routines are invoked. Interrupt routines should invoke when receiving data from either the server or the RFID card reader. Data processing is done within the interrupt routines and when an interrupt routine has finished running, the program should return inside the loop again.

All MCU's modules should be initialized prior to entering the loop, which means that all initialization routines should be run first.

8.2 Communication watchdog

There should be implemented a watchdog to monitor communication status between the Tally Station and the server. If there is a communication failure both system administrator and the Tally Station users should be noticed. Terminal software/server software should be responsible for notifying the administrator while the Tally Station's software should notify the users of the TSS.

The plan is to make software flexible with two watchdog options. First option is also option which should be used in the prototype. Communication status should be checked prior to running the main program and in the main program the communication status should only be checked when sending data to the server. If the Tally Station does not receive an acknowledgment from the server then this would indicate communication failure.
Second option is to constantly check for communication status inside the main program. Unfortunately the only way to implement the constant communication watchdog is to poll the server about communication status. If the server for some reason does not respond then this would indicate a communication error. A user should be able to see communication status on the Tally Station's display.

8.3 LCD

LCD is controlled by the software via MCU's built-in display driver. To be able to write on the LCD, a setup of the display driver is necessary. To be able to display all segments on the LCD it has to be driven in 4-mux mode. 4-mux mode simply makes it possible to control four LCD segments via single pin on the microcontroller. More about this will be described in the software implementation chapter.

The display should be able to show various information to the users and this is specified on figure 8.1. Figure shows which segments should be used for displaying information about TSS. There are other segments on the display which will not be used and are therefore not shown on figure 8.1.

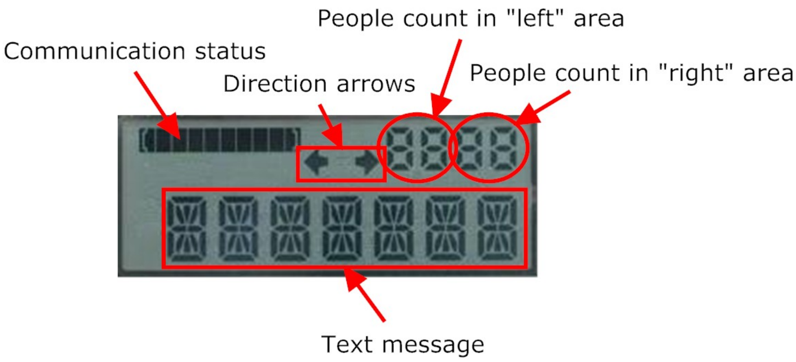


Figure 8.1: Overview of the LCD segments used when displaying information to users.

8.4 RFID reader

The Tally Station should be able to receive RFID card ID from the STK-5000 RFID reader module. Card ID is an eight character string which is unique to every card. This ID is used for user identification in the Tally System. The cards which are currently used offshore are of type Hitag1 and can contain up to 256 bytes of data, but for this project the only interest is a card's ID. To read a card's ID a request must be sent by the software from the microcontroller to the reader module. This is done by using the reader's communication protocol.

RFID reader has its own communication protocol which is implemented in the firmware by the manufacturer. The protocol is named STXETX protocol and includes a predefined set of commands which can be sent to the RFID reader. After receiving a command the RFID reader responds with a proper answer. Request and response are always sent as a string embedded within the STX and ETX control characters and a checksum follows at the end. Checksum is sent as a hexadecimal value and is calculated by XOR-ing all characters in the string including STX and ETX control characters. STX and ETX are included in the ASCII character set and have hexadecimal values of 02h and 03h respectively. More about this is described in the software implementation chapter.

According to STXETX protocol specification it is not possible to implement card reading as an interrupt controlled sequence, which is quite disappointing, and which means that a request for card reading must be sent continuously from the MCU. This means that reading of RFID cards should be implemented as polling.

8.5 Interrupt routines

Interrupt routines should be invoked when receiving data via microcontroller's UART ports. The routines should be invoked using the UART interrupt vectors. Each of the MCU's UART ports has an interrupt vector for transmitting and receiving respectively. An interrupt vector is a memory address of the interrupt handler and this is executed when an interrupt occurs. There should be implemented two interrupt routines.

Interrupt routines should process all communication and when finished the program should return inside the main loop. Communication processing includes executing enquiries, sending acknowledgments and acting upon peripherals accordingly. When receiving data, software buffers should be used to save entire communication string before processing it. Buffers should be implemented as character arrays and should be able to store entire communication strings as defined in the communication protocols.

Table 8.1 shows an overview of interrupts that should be implemented along with their respective buffers.

Interrupt vector	Software buffer	Used when receiving data from...
UART1RX_VECTOR	rx1_buffer	RFID reader
UART0RX_VECTOR	rx0_buffer	Server

Table 8.1: Overview of interrupts that should be implemented.

8.6 Communication protocol

The communication protocol is used for communication between the server and the Tally Station. Protocol is based on the two way communication which means that both the server and the Tally Station can send requests to each other. After one of the parts sends a request, the other part should respond either by acknowledging a request, meaning that the request is processed, or by responding that a request cannot be processed.

The focus is on making a simple but functional communication protocol and the inspiration was taken from the STXETX protocol which is implemented in the RFID reader's firmware. The communication protocol is a thirteen-byte string in which different commands are sent between the server and the Tally Station. All commands are predefined and should be recognized in the software. The protocol is defined as shown on figure 8.2.

Byte 0 … Byte 12

STX	Option	Command	8 Data bytes								ETX	Checksum

Figure 8.2: Communication protocol used by the server and the Tally Station.

Every communication string is encapsulated by STX and ETX characters. These stand for "start of text" and "end of text" and are standard ASCII characters with hexadecimal values of 02h and 03h respectively. These characters should always be included in the communication string regardless of other bytes. The last byte is always the checksum for the communication string which is described in the following. The other bytes between STX and ETX characters may vary depending on different commands that are implemented and are therefore described in the software implementation chapter.

Checksum for the communication string is calculated by using XOR operation on hexadecimal values for all bytes including STX and ETX characters. Following shows an example of checksum calculation. Assuming that a communication string is defined as following:

STX	ENQ	C	4	2	C	D	8	2	4	5	ETX	Checksum

Figure 8.3: Example of a communication string when sending RFID card ID to the server.

Then the calculation of the checksum would look like following:

Character	Hex value	Previous checksum value	Checksum value
			00h
STX	02h	00h	02h
ENQ	05h	02h	07h
C	43h	07h	44h
4	34h	44h	70h
.	.	.	.
.	.	.	.
ETX	03h	4Eh	**4Dh**

Table 8.2: Example of the checksum calculation.

Value 4Dh is final checksum value and should be sent as the last byte in the communication string.

Part III

Implementation

9 Hardware

This chapter describes implementation of the hardware components which are described previously in the hardware design chapter. All of the implemented components are described to provide good understanding of the Tally Station's build and functionality. The connection diagram for all hardware can be seen on drawing TSS-001 in the appendix.

Here comes an overview of print board where all hardware components are placed. Microcontroller development board and RFID reader module were also shown on figures 5.1 and 5.2 respectively. Print board layout is shown on figure 9.1.

It should be mentioned that RFID antenna is attached on the other side of the print board and is therefore not visible.

Print board's size is 160mm x 100mm.

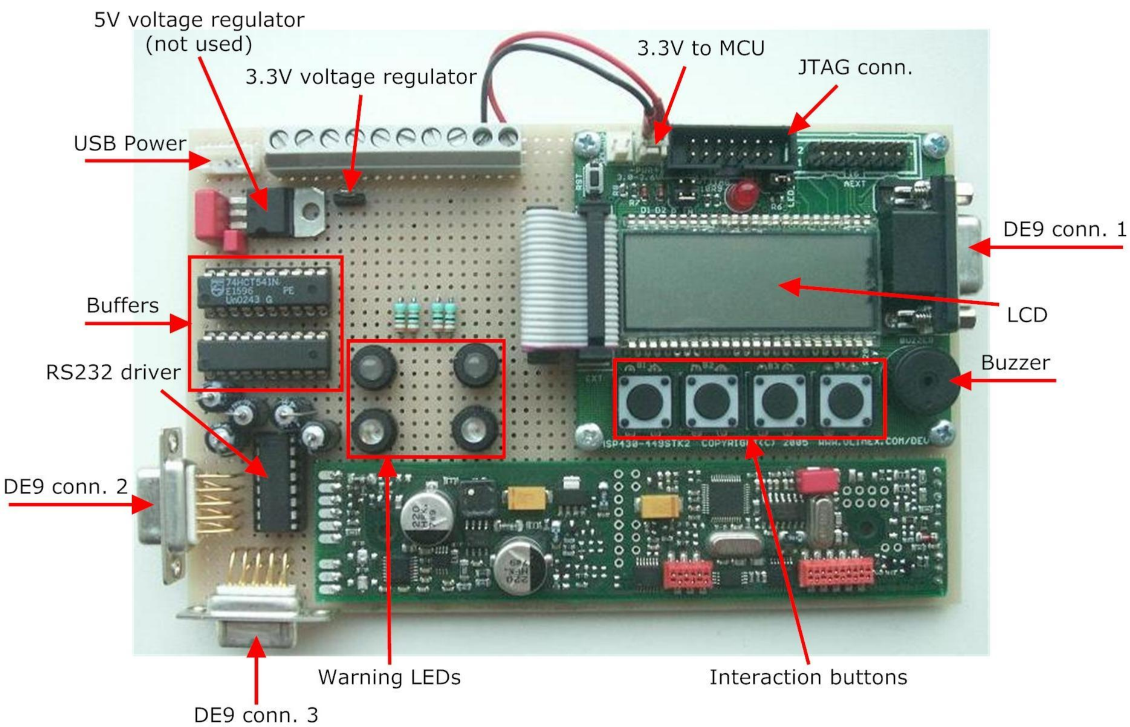


Figure 9.1: Placement of hardware components.

9.1 Power supply

As mentioned in the design part the power supply should deliver two different voltages to the Tally Station. +3.3V is needed for the microcontroller and +5V is needed for other components such as RFID reader and other logic components. The main focus at the implementation stage was to avoid laboratory power supply unit, and due to the short project period building of the power supply from scratch was not even considered.

After some consideration the decision was made to use a computer's USB port as a power supply for the prototype. The USB port delivers +5V DC and a maximum current of 500mA which should be enough for the TSS. To get +3.3V for the microcontroller a voltage regulator is used. Diagram for the power supply block is seen on figure 9.2.

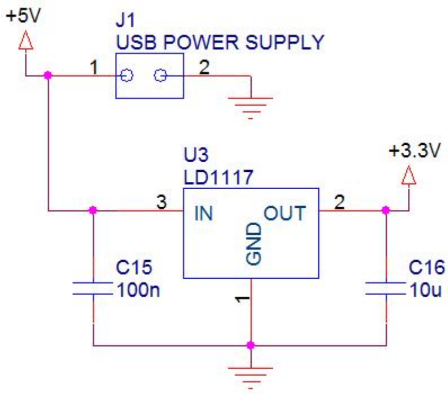


Figure 9.2: Power supply diagram.

9.2 LCD

The Display was already implemented on the development board and here comes a short description of the implementation. The display has 48 pins where 46 are connected to the microcontroller's LCD controller pins. Two of the pins have no functionality. This can be seen on drawing TSS-001 in the appendix.
Each pin drives four display segments together with four COM lines. Segment assignments can be seen in the LCD's datasheet which is included on the enclosed CD-ROM. There are numerous options for driving a display and these are selected by software. Therefore all display functions are described in the software design and implementation chapters.

9.3 User interaction buttons

As mentioned earlier in the design chapter four buttons are implemented on the development board. These are simple push buttons which are connected to the microcontroller's digital input pins. The MCU's pins must be chosen for input by the software. The input signals to the microcontroller are high when buttons are not pushed as seen on figure 9.3.

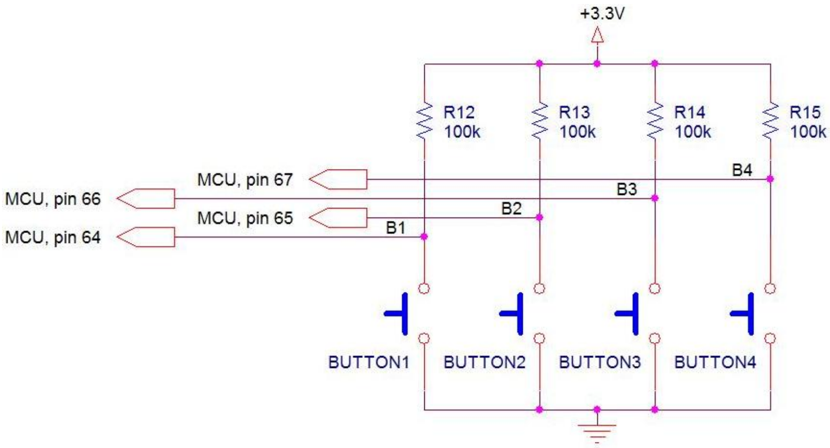


Figure 9.3: User interaction buttons diagram.

9.4 Buzzer

The buzzer is only used when warning a user that an area has reached its capacity limit. It is always turned on together with one of the red LEDs depending on which area a person is trying to enter.
The buzzer is connected to two digital outputs on the microcontroller and to get the sound these outputs have to be toggled continuously. The buzzer is controlled by the software and further information is available in software implementation chapter.

9.5 Warning LEDs

The Tally Station has four LEDs which are used as signals whether access to an area is granted or denied. Two LEDs are used for each area, red LED for access denied signal and yellow LED for access granted signal. The LEDs are controlled by the software and they signal a person shortly after a card is read and processed.

LEDs are connected to microcontroller's digital output pins and this can be seen on the connection diagram TSS-001 in the appendix.

9.6 RFID card reader

The RFID reader's power supply should be between +4V and +12V which are absolute maximums. Therefore it is fine to use +5V power supply from computer's USB port. Only important thing is that the power supply signal is without ripple or else a reader can behave strangely according to the datasheet for the reader module.
After some research on the Internet it was determined that the power supply from

computer's USB port is without ripple. It should be mentioned that this was not tested in praxis, but the module currently works fine with USB power supply.

The module's current consumption is 80mA when reading/writing data, and 30mA when in standby. These values are fine as it is possible to draw up to 500mA from the USB power supply.

To be able to communicate with the microcontroller via RS232 and for debugging purpose a DE9 connector is connected to the RFID reader's RS232 interface. RFID antenna is also connected to the appropriate pins, and thereby reading of RFID cards is possible.

Diagram for implementation of the RFID reader is seen on figure 9.4.

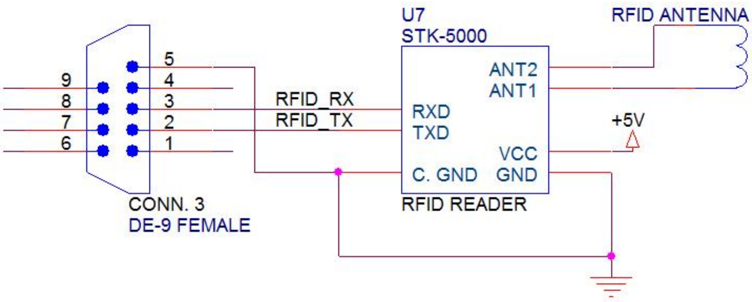


Figure 9.4: RFID card reader diagram.

Software provided by the manufacturer makes it possible to test the reader module's functionality by sending and receiving commands which are implemented in reader's communication protocol. These commands are also used when communicating with the reader through software. This is further described in software design and implementation chapters.

9.7 Communication

There are two communication channels in the TSS. Channel one is communication between the MCU and the server, and uses MCU's UART0 port. Channel two is communication between the MCU and the RFID reader, and uses MCU's UART1 port. Both channels are using RS232 serial communication.

As mentioned in the design chapter the communication with the server should be RS485 but due to debugging reasons RS232 is chosen for the prototype. Although it is still possible to use converters to achieve RS485 communication.

To be able to communicate using RS232 the drivers are implemented for both channels.

Driver for channel one has been implemented as described in hardware design chapter and the diagram is shown on figure 9.5. This driver is integrated circuit of type

HIN232, whereas driver for channel two is made by Olimex from conventional components. HIN232 has onboard voltage converters which convert voltages between TTL and RS232. It needs +5V and draws a maximum current of 10mA.

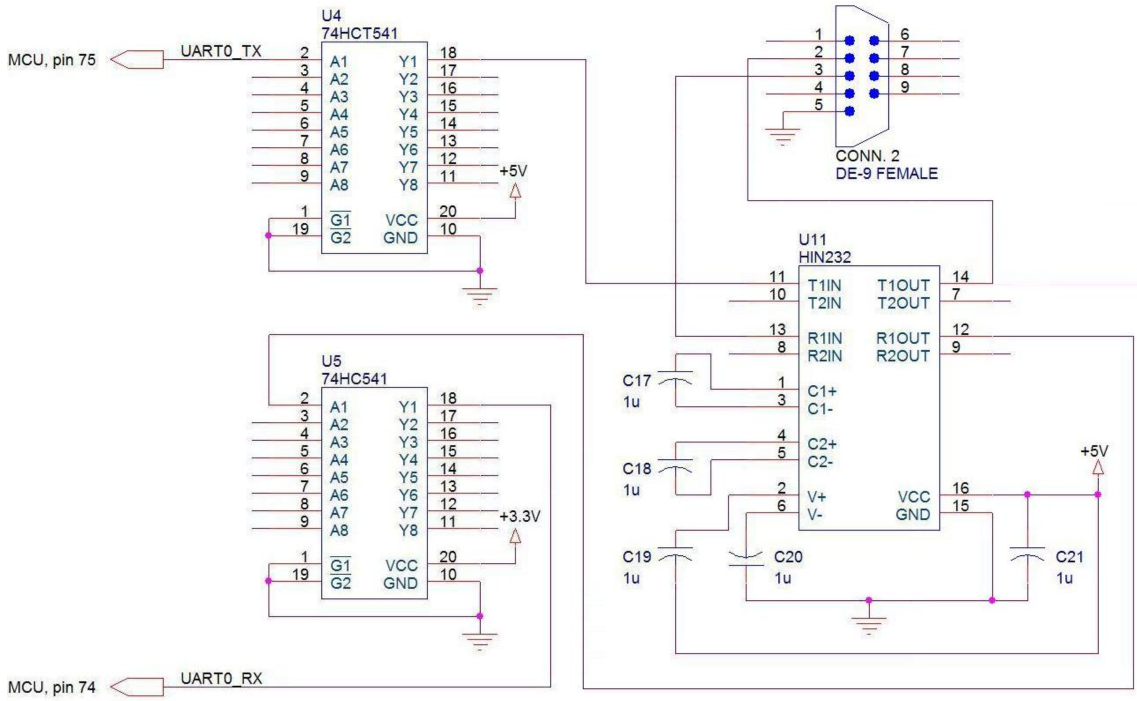


Figure 9.5: RS232 driver for communication with the server.

Driver for channel two has been implemented on the development board by the manufacturer and is shown on figure 9.6.

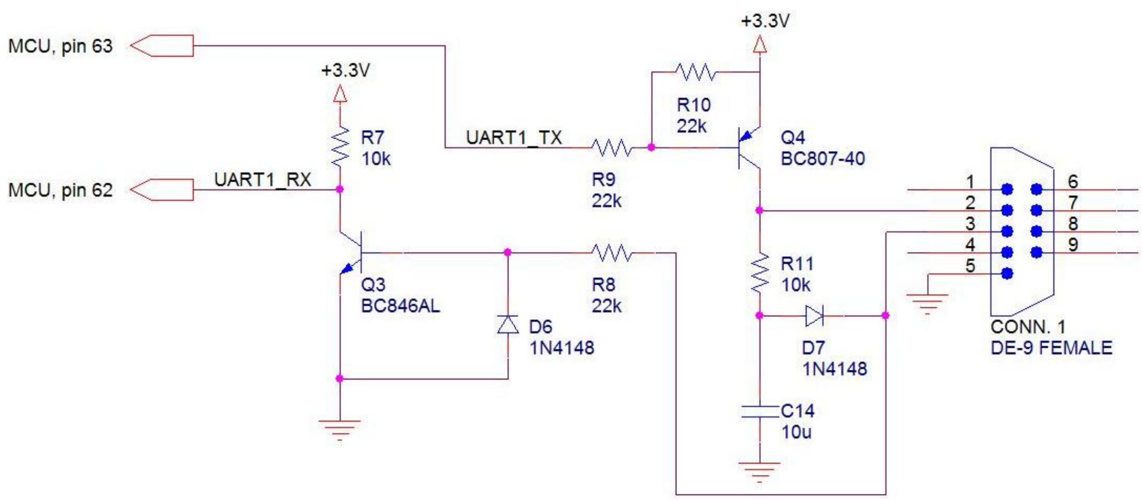


Figure 9.6: RS232 driver for communication with the RFID reader.

9.8 Microcontroller

There is not much to write about hardware microcontroller implementation as it was installed by Olimex on the development board used for the project. Peripherals like the interaction buttons, the buzzer and display were also implemented by the manufacturer. The same goes for DE9 connector 1 and RS232 driver which use UART1 port.

Other components are connected to the appropriate pins on the microcontroller and all information on MCU implementation is seen on drawing TSS-001 found in appendix.

10 Software

This chapter describes software implementation as described previously in the software design chapter. Software is implemented using C programming language. Testing and debugging is done using Embedded Workbench software from IAR systems.

10.1 Main program

The main program is implemented as an "endless" loop which is only stopped when an interrupt is invoked. Flow diagram is shown on figure 10.1.

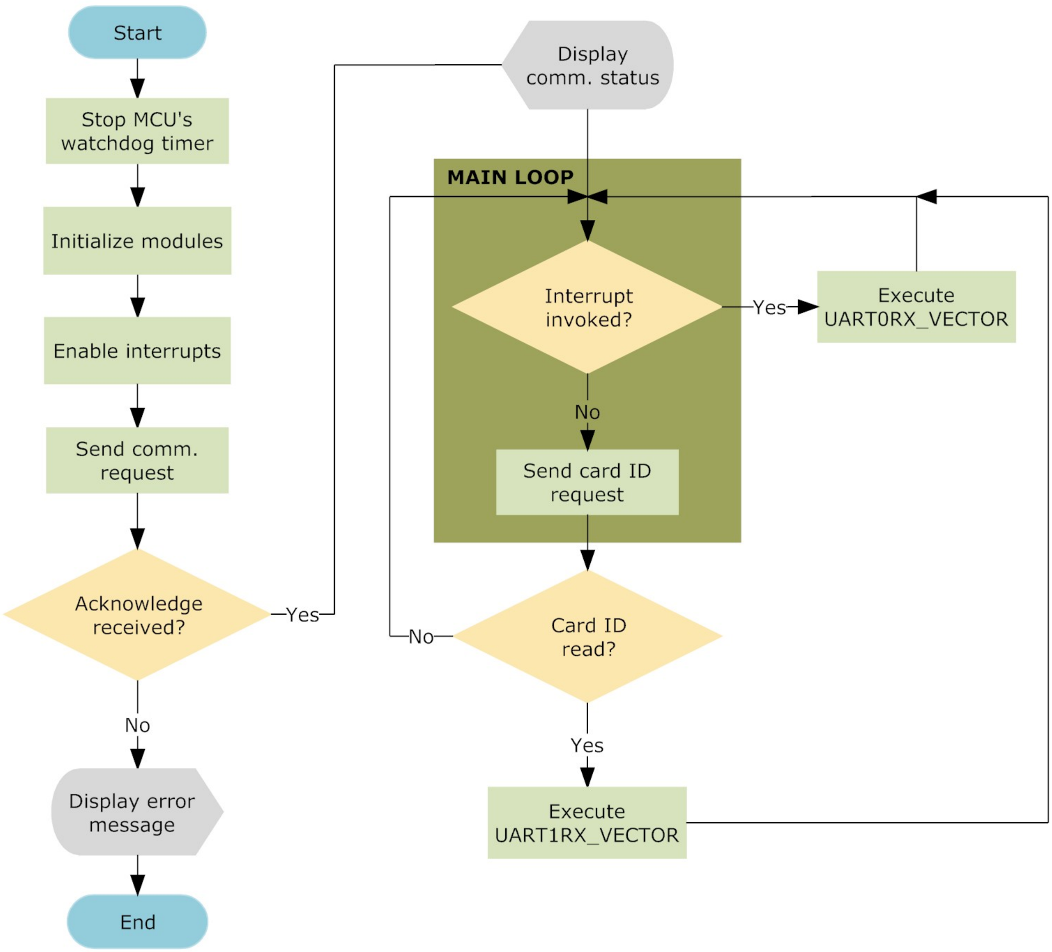


Figure 10.1: Flow diagram of the main program.

Prior to entering the loop initialization of MCU's modules is done. To make diagram simpler all initialization routines are drawn as "Initialize modules" block.

When checking the communication status with the server a request is sent according to the communication protocol standard. Possible commands are described in the communication protocol implementation section. If no acknowledgement is received from the server an error message is written on the display and the main program is stopped. Tally station has to be reset to run properly again.

If a communication with the server is existing, the communication status is displayed and the main loop is executed.

10.2 Communication watchdog

The communication watchdog is implemented as described in the software design chapter. Prior to entering the main loop a command string is sent to the server via communication protocol enquiring about the communication status. If the server responds the communication status bar is displayed in the Tally Station's LCD. If there is no response from the server the communication status bar is not shown but an error message is displayed instead. Types of error messages are described in appendix B.

After the initial check of the communication status the main loop is executed. Hereafter the communication status is only checked when sending enquiries to the server, and thereafter the status on the Tally Station's LCD is updated accordingly.

10.3 LCD

As mentioned in the software design chapter the LCD is driven in the 4-mux mode. In the 4-mux mode, each MCU segment pin drives four LCD segments and all four common lines, COM0, COM1, COM2, and COM3 are used. An example of the 4-mux operation mode is shown on figure 10.2. Mux mode is selected in the initialization routine for the LCD. Other important things in the initialization routine include setting all MCU segment pins to output direction, and setting the correct frequency for common and segment lines. Choosing the right frequency for LCD is important to avoid flicker or other potential problems with the display.

The LCD controller uses the f_{LCD} signal to generate the timing for common and segment lines. MCU's Basic Timer1 is used to generate the f_{LCD} signal using ACLK which has a standard frequency of 32768Hz. This frequency has to be divided to get a suitable frequency for the LCD.
The proper f_{LCD} frequency depends on the LCD's frame frequency (f_{Frame}) which is found in the LCD's datasheet, and the multiplex rate (mux mode). f_{LCD} is calculated by the following formula:

$$f_{LCD} = 2 \cdot mux \cdot f_{Frame} \tag{10.1}$$

f_{Frame} should be between 30Hz and 100Hz, and since LCD is driven in 4-mux mode it is possible to determine minimum and maximum values for f_{LCD}.

$$f_{LCD-min} = 2 \cdot 4 \cdot 30 = 240Hz \tag{10.2}$$

$$f_{LCD-max} = 2 \cdot 4 \cdot 100 = 800Hz \tag{10.3}$$

This means that it should be safe to set f_{LCD} to any frequency between 240Hz and 800Hz. This presents two choices for LCD frequency of 512Hz and 256Hz, when dividing the ACLK by 64 or 128 respectively. LCD controller frequency is set at 512Hz as it lies in the mid-range between the allowed frequencies.

MCU's LCD controller functions can be difficult to understand and describe, but here comes a short and hopefully understandable description on how LCD controller works. LCD controller has 20 memory registers of eight bits each. Each memory register controls two segment pins which means that in 4-mux mode each memory register controls eight LCD segments. The segments are turned on by setting the appropriate bits in the desired memory register.

Following is an example for controlling a single seven segment digit. This example is taken from the actual project. The digit used in the example is the last digit in the upper right corner of the LCD.

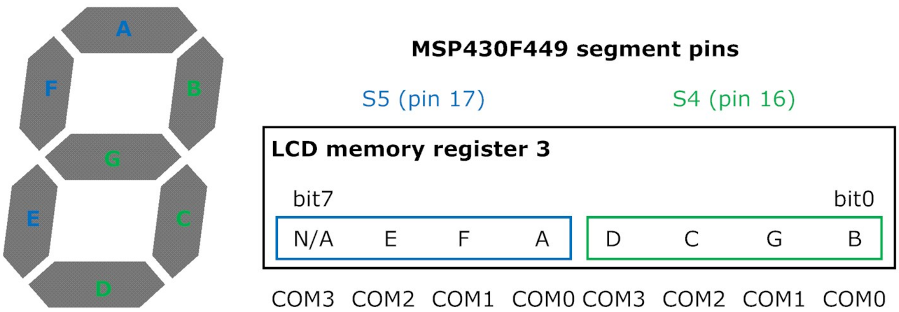


Figure 10.2: Example of controlling a seven segment digit using MCU's LCD controller.

In this case all seven segments are controlled by LCD memory register three which controls segment output pins S4 and S5. For example if a number three should be written on the display the hexadecimal value of 1Fh is written to LCD memory register three.

After determining the functionality and limitations of the LCD controller, several software functions are implemented for writing to different areas of the LCD as described in the software design chapter. Functions are not described in depth but merely mentioned and their functionality is shortly described. It should be mentioned that not all functions are used, but are implemented in case there arises a need for them.

The implemented functions include:

- *void LCD_init()*
  Initialization subroutine for the LCD controller

- *void LCD_writeLetter (unsigned int position, char letter)*
  Write a single letter in the 14-segment field
- *void LCD_writeWord(const char *word)*
  Write a word in the 14-segment field
- *void LCD_writeNumber(int long number)*
  Write a number in the 14-segment field
- *void LCD_writeClockDigit(unsigned int position, unsigned int digit)*
  Write a single digit in the 7-segment field
- *void LCD_writeClockNumber(unsigned int value, int paddingZeroes)*
  Write a number in the 7-segment field
- *void LCD_setClockColon(int on)*
  Set a colon between digits in the 7-segment field
- *void LCD_clearClock()*
  Clear the 7-segment field
- *void LCD_allOn()*
  Turns on all segments on the LCD
- *void LCD_clear()*
  Clear the display

10.4 RFID reader

To be able to communicate with the RFID reader the UART1 port's setup must match the setup of the serial interface of the RFID reader. Default setup for RFID reader is used and the UART1 is set-up accordingly. The setup is following:

- Baud rate: 9600 bps
- Data bits: 8
- Parity: none
- Stop bit: 1

As mentioned in the software design chapter the reader has to be polled when reading a card ID. This is done from the main loop by sending the command for reading the card ID. All commands are defined as four characters enclosed by STX and ETX characters with the checksum at the end. When requesting a card ID a following command is sent to the reader:

02h 6200 03h 05h

6200 is the command for requesting the ID from an RFID tag of type Hitag1. 02h, 03h and 05h are hexadecimal values for STX, ETX and checksum respectively. All commands and other protocol information is available on the enclosed CD-ROM in the STXETX protocol specification documentation.

If a card is in range of the reader than a card's ID is returned in the following string:

06h 02h 620042CD8245 03h 09h

The first character returned from the reader indicates whether a request is acknowledged or if an error occurred. In this case first character returned is ACK which is ASCII character and has a hexadecimal value of 06h. This indicates that the card is read and its ID is included in the string. Card ID is the eight characters following the issued command which is 6200. In this case the card ID is 42CD8245.

If there is no card near the reader the following string is returned from the reader indicating an error.

16h 02h 620010 03h 12h

In this case the first character is SYN which has a hexadecimal value of 16h. This indicates that an error occurred while trying to read a card ID, in this case because no card is present near the reader.

When checking if a card has been read a first character returned from the RFID reader is always checked, as it indicates if a return string is an acknowledgment or an error. If the string indicates an error than the string is ignored. If the string indicates a successful read and returns a card ID than this string is processed and card ID is sent to the server for user check. Communication processing is described in the next section which concerns interrupt routines.

Based on the size of return strings from the RFID reader the rx1_buffer's size is set to 16 bytes.

10.5 Interrupt routines

The interrupt routines are responsible for processing the received communication strings. One interrupt routine is implemented for each communication channel which is invoked as soon as a complete character is received in the MCU's receive buffers. To explain shortly the MCU has one receive buffer for each UART port which contains last received character from the receive shift register. These characters must be saved one at a time to be able to receive a complete communication string. This is why in this project a software buffers are used.

Each interrupt routine has a software buffer where the characters are saved prior to processing.

In case when receiving data from the RFID reader the entire string is not always saved due to polling. The first character that is received is checked, and if this is a SYN character the string is ignored. If the first character is the ACK character then the string is saved and processed, which usually involves forwarding card ID to the server for validation. The size of the software buffer must be able to accommodate all characters in the string which is why the size of the rx1_buffer is as mentioned in the previous chapter set to 16 bytes.

In case when receiving data from the server the string is validated as the characters are received. The focus is on STX and ETX characters as these should always be included in the communication strings. After the complete string is received the checksum is validated. In case of a checksum error another string is sent back to the server with a message that the previous string is not acknowledged. The rx_buffer should be able to hold all characters in a single communication string as described in the designed communication protocol, which is why the size of rx_buffer is set at 13 bytes.

10.6 Communication protocol

Communication protocol is implemented as described in the software design chapter. The protocol structure was described earlier in the design chapter along with the checksum calculation method.
In this section bytes between the STX and ETX characters will be described along with different command possibilities.

Byte 1 in the communication string is the option byte and indicates the type of the communication string. It can contain values shown in table 10.1. All of the option bytes shown in the table are acceptable by both the server and the Tally Station.

Option (ASCII)	HEX value	Description
ENQ	05h	The string is an enquiry/request
ACK	06h	The string is an acknowledgement/answer to a request
NAK	15h	The previously requested command is not acknowledged

Table 10.1: Overview of possibilities for the option byte in the communication string.

Byte 2 is the command byte and "tells" the receiving part what to do if sent in an enquiry string, or "tells" the receiving part what was done if sent in an acknowledgement string. Table 10.2 shows a list of commands that are implemented.

Command (ASCII)	HEX value	Description	Sender
B	42h	Turn on/off the buzzer	Server
C	43h	Check the card ID	T. Station
c	63h	Check the card ID (when passing through)	T. Station
D	44h	Write message on the Tally Station's display	Server
(	28h	Turn on/off "left" red LED	Server
$<$	3Ch	Turn on/off "left" yellow LED	Server
)	29h	Turn on/off "right" red LED	Server
$>$	3Eh	Turn on/off "right" yellow LED	Server

Table 10.2: Overview of possibilities for the command byte in the communication string.

Data bytes are used as an additional data associated with the command byte. There are three cases when data bytes are sent in a communication string, otherwise these are set as NULL character (00h).

First case is when the Tally station is sending a card ID to the server for processing. All eight data bytes are filled as the card ID is eight characters long.

Another case is when the server is sending an acknowledgement for a processed card ID and the server "tells" the Tally Station which LED should be turned on. Only one data byte is used here and the character sent is indicating which LED should be turned on. The characters for LED control are mentioned in the table 10.2.

The third case is when the server is sending a command to the Tally Station for writing a message on the LCD. Characters that should be written on the display are sent as the data bytes. Only seven data bytes are used as this is the maximum number of characters that can be displayed on the LCD. The eighth and final data byte is sent as NULL character.

Part IV

Summary

11 Conclusion

The goal for this project was to develop the functional Tally Station and achieve the system requirements. The reason for choosing this kind of project is mainly due to my interest in microcontrollers, software development and especially wireless communication. These are also areas of work I would prefer to work with in the future.

I think that this project achieves most of the system requirements, although entire software is not fully implemented and tested. Part of the software is only tested in simulation mode in the IAR Embedded Workbench. The reason for this is that in the last stage of the project period I accidentally burned the security fuse on the microcontroller. Once the fuse has been burned the microcontroller can no longer be flashed. The purpose of security fuse is that it should only be burned after a complete program is implemented, and this should prevent future changes.

Other thing that is not implemented is user interaction with the Tally Station. The reason for this is prioritizing, as the first priority was to develop a functional TSS that can receive RFID card ID and communicate with the server.

There were plans to do a final test and describe it in the report but this was not possible as the MCU cannot be used. I also had plans to test the Tally Station in simulated environment with other ABB software but this was dropped due to lack of time, and of course due to accident with the microcontroller.

I enjoyed working on this project but wished at times that there was more time to make several improvements in the system. I could especially use more time in the beginning stage as I would like to do more research concerning potential technologies that could be used in such project. For example it would be nice if the users did not have to take their RFID tags out of the pocket but just stand in front of the Tally Station to be registered. This probably needs other type of RFID antenna, or maybe even another tag standard. This is just one of the things that could be improved if more time was available. One of the things that consumed a lot of time was understanding the LCD controller and the functionality of the implemented LCD. This was mainly because I never used this type of LCD controller nor the LCD with so many segments.

Generally I am satisfied with the accomplishment considering that it is my first solo project. I think that things can get quite stressful when a single person has to focus on all project elements. I think that in some stages of the project I was working on several things at the same time, and was not sure where to start. But as mentioned in the previous I am satisfied with the overall project.

Part V

Appendix

A Connection diagram

B LCD messages

Implemented LCD messages are described here. LCD messages are shown on the LCD's 14-segment text field to inform the users about current TSS status.

LCD message	Description
TSS ON	The Tally Station is fully functional
COMM E	Error in establishing the communication with the server
USER E	User is not reckognized or card reading error
AREA F	Area a person is trying to enter is full

Table B.1: Overview of the implemented error messages.

C Ex certification

References: [5] and [7]
As mentioned in the analysis part the Tally Station has to be Ex certified to be installed offshore. This is because the areas where the Tally Station will be installed are considered as hazardous areas, due to vapors and gasses. All electrical equipment installed in these areas has to be explosion-proof which means that it must be isolated as electrical equipment can emit sparks and ignite surrounding gasses and vapors. Electrical equipment that is Ex certified must bear a label that shows a level of protection.

There are several different zones and classifications of hazardous areas and the equipment has to be certified accordingly. In the following is a short description of various zones, gas groups, and temperature classification that are used to mark the Ex certified equipment.

Safe area

A living area such as a house would be classed as safe area because the only risk of a release of explosive or flammable gas is for example use of a spray can. The only explosive or flammable liquid would be paint and perhaps some domestic cleaners. These are classed as very low risk of causing an explosion and are more of a fire risk.

Zone 2 area

In this case a gas or vapor would only be present under abnormal conditions, for example if there exist leaks and such. Unwanted substances should only be present under 10 hours per year or 0 – 0.1% of the time.

Zone 1 area

In this area special or classified electrical equipment must be used. It is expected that the gas or vapor will be present for long periods of time. This is defined as 10 – 1000 hours per year or 0.1 – 10% of the time.

Zone 0 area

This is the worst scenario as gas or vapor is present over 1000 hours per year or more than 10% of the time. It is very rare that a zone 0 area will be in the open.

Each gas or vapor comes under a certain gas group. There are four gas groups which are described in the following.

Group I includes all underground coal mining gasses and vapors.

Group IIA includes industrial methane, propane, petrol, majority of industrial gasses and similar.

Group IIB includes ethylene, coke oven gas and other industrial gases.

Group IIC includes hydrogen, acetylene, carbon disulphide and similar.

Another important consideration is the temperature classification of the electrical equipment. The surface temperature of the electrical equipment may not rise beyond the auto-ignition temperature of the gas or vapor that exists in the area where the equipment is installed.
All of the above mentioned must be taken into consideration when certifying the equipment for use in hazardous areas. It tends to be an extensive process and is probably one of the reasons why the Ex certified equipment costs more than similar equipment that is not Ex approved.

D Acronyms

Here are descriptions for some of the acronyms used throughout the report.

DE9 - This is connector type with 9 pins used for serial communication. It has other names such as Dsub9 and DB9.

IC - Short for Integrated Circuit.

LCD - Liquid Crystal Display which is most used type of display today.

MCU - Stands for Micro Controller Unit. This is simply short for microcontroller.

RFID - Stands for Radio Frequency Identification. This is a wireless communication form using radio waves.

UART - Universal Asynchronous Receiver/Transmitter is a hardware that translates data between parallel and serial forms.

USB - Universal Serial Bus is today's most used standard for interfacing devices to a computer.

Bibliography

[1] ABB.
Gorm Tally System Maintenance Instructions.
ABB, 2005.

[2] ABB.
TallySys Client User guide.
ABB, 2005.

[3] Lammert Bies.
rs232 specifications and standard.
http://www.lammertbies.nl/comm/info/RS-232_specs.html.

[4] Lammert Bies.
rs485 serial information.
http://www.lammertbies.nl/comm/info/RS-485.html.

[5] IECEx.
about the iecex.
http://www.iecex.com/about.htm.

[6] Scemtec.
scemtec transponder technology gmbh.
http://www.scemtec.com/stt/en/index.htm.

[7] Wikipedia.
electrical equipment in hazardous areas.
http://en.wikipedia.org/wiki/Electrical_Equipment_in_Hazardous_Areas.