Aalborg University
Institute of Electronic Systems
Department of Control Engineering 9. Semester

TITLE:

Developing a Robust Protocol for Wireless Fire Alarms

ABSTRACT:

This project deals with the development of a communication protocol used to ensure a robust and reliable communication in a wireless fire alarm system.
Based on an analysis of the scenario consisting of wireless detectors communicating with gateways which are connected to a central unit via a cabled network, the communication technologies are chosen. The ZigBee protocol is chosen as the wireless communication standard, and the cabled network is simulated using RS232 connections.
An application layer protocol is developed on top of the ZigBee protocol, with high availability, reliability, robustness, and low power consumption as primary objectives.
A detector is designed and implemented using a TI-430F149 microcontroller and a gateway and central unit are merged and implemented on a Linux PC. Two Adcon Addlink 868 MHz ZigBee based radio modems are used as Wireless links.
The designed protocol is verified using UPPAAL to ensure that no live- or deadlocks exist, and the required specifications dictated by EN/DS standards are met.
The protocol software is implemented on a system consisting of one detector and one gateway/central unit.
Through testing it became evident, that the developed protocol can provide robust and reliable communication between detector and central unit, with an error rate of less then 10^{-9} errors/hour.

PROJECT PERIOD:
2nd of September, 2003 -
5th of January, 2004

PROJECT GROUP:
03gr937b

GROUP MEMBERS:
Morten Tofte Koch
Jesper Noer
Michael Pedersen

SUPERVISOR:
Anders Jørgensen

Copies printed: 6

Report Pages: 122

Appendix Pages: 20

Total number of Pages: 142

Aalborg University ♦ Fredrik Bajers Vej 7 ♦ 9220 Aalborg Ø
tlf: 98 15 85 22 ♦ fax: 98 15 97 57

Resumé

Dette projekt omhandler udvikling af en robust og pålidelig kommunikationsprotokol til brug i et trådløst brandalarmeringssystem.

Kommunikationsteknologien er valgt på baggrund af en analyse af scenariet, hvor trådløse detektorer kommunikerer via gateways, som er forbundet til en central enhed via et kablet netværk. ZigBee protokol standarden er valgt som til det trådløse netværk og det kablede netværk er simuleret med RS232 forbindelser.

En applikationsprotokol er udviklet oven på ZigBee protokollen, med parametre som høj tilgængelighed, pålidelighed, robusthed samt lavt strømforbrug som mål.

Detektoren er designet og implementeret ved hjælp af en TI-430F149 microcontroller. Gatewayen og den centrale enhed er implementeret på en Linux baseret PC. To Adcon Addlink 868 MHz ZigBee radiomoduler udgør den trådløse forbindelse.

Den designede protokol er verificeret ved hjælp af UPPAAL, for at sikre at systemet ikke indeholder live- eller deadlocks, samt at krav fra EN/DS standarder er opfyldte.

Protokol softwaren er implementeret på et system bestående af én detektor og én gateway/central enhed.

Systemtests har påvist, at den udviklede protokol giver en robust og pålidelig kommunikation mellem detektor og den centrale enhed, med en fejlrate på mindre end 10^{-9} fejl pr. time.

Preface

This report is written as ninth semester project documentation of the Distributed Application Engineering specialisation at Aalborg University's Department of Control Engineering.

Within the overall semester theme of Distributed Real Time Systems, the project group has chosen to work with Developing a Robust Protocol for Wireless Fire Alarms.

The report caters for the projects supervisor, the examiner, future students of this specialisation, and other people interested in the topic treated. Some knowledge of distributed systems and network terms and technology is required in order to gain full profit of the reports content.

Morten Tofte Koch

Jesper Noer

Michael Pedersen

How to read this Report

Bibliographic references are given in square brackets with a number where the number references to a bibliography on page 109. The bibliography is listed by the order of appearance. If no explicit author can be identified the company or organisation name is used as well as references to internet sites.

References within the report are given with both chapter, section and/or subsection number.

Figures and tables are identified by the chapter number and an incrementing number within each chapter. The same number scheme is used for equations and formulas.

In text references to software code such as variables, functions or commands in terminal session are indicated by using the `courier` font.

On the last page of this report is a nomenclature that explains the acronyms and terms used throughout this document.

Attached to this report is a CD-ROM that includes the source code and this report in PS- and PDF format. References to files on the CD-ROM are indicated by a CD-ROM logo and a description with complete path. The CD-ROM contains a self-executable menu from which the contents can be browsed.

Contents

Introduction

Chapter 1

In this chapter the formal purpose is described as defined by the ESN study board. An introduction to the initial problem is given, and the project scenario is established. Based on this scenario and the project scope, the problem definition is presented.

1.1 Purpose

The purpose of this project is to gain knowledge within the area of distributed real-time systems. The formal purpose is outlined in the theme description published by the study board [1]. According to this description the purpose of the semester is:

- *To provide knowledge and understanding of analysis and design methodologies of distributed real-time systems.*
- *To provide understanding of the importance of reliable behavior in dependable systems.*

The project is based on a general problem, where distributed real-time systems provide possible solutions. The problem is analysed with regards to communication needs and associated quality of service aspects. A number of communication technologies and architectures are compared to the needs identified and a technology is chosen for further treatment. A distributed real-time system is designed as a solution to the problem based on the selected technology and architecture. A number of elements, sufficient to demonstrate the virtues of the designed system, are implemented. Test is performed at module and system levels and acceptance testing is partly specified. At every level temporal behavior and reliability should be tested.

1.2 Initiating Problem

Ever since ancient history cultural values have been annihilated due to ravaging fires. Many people lost their lives trying to fight the flames, because the fire was not detected early enough and because of unefficient extinguish equipment.

Shortly after Samuel B. Morse invented the telegraph in 1844, scientists began trying to utilize the telegraph for reporting fires by means of fire alarm signal boxes connected to the nearest fire station. Such fire alarm systems were developed further as the communication infrastructure of society evolved from telegraph to telephone. Today fire alarms are widely used in factories, institutions, and other places where large groups of people gather e.g. train stations and shopping centers.

Conventional fire alarms consists of a number of detectors connected to a central unit which performs the alarm call. The connection between the detectors and the central unit consist of cables. The installation and cabling of a fire alarm system comes to approximately one third of the total alarm price [2].

Due to the cabling needed between detectors and the rest of the alarm system, there are situations where traditional fire alarms are unfit for use. In many buildings worthy of preservation it is unwanted to install a large cabled fire alarm system. This would be the case in e.g. a museum or a castle with a large ceiling fresco. Hence, a wireless system for detecting fires is necessary. A wireless system is also advantageous in rooms where drilling holes in walls or ceiling is unwanted. This could be the case in rooms containing material with high risk of explosion or on board ships.

Because of the limitations and drawbacks of conventional cabled fire alarms mentioned above, it is desirable to develop a wireless fire alarm (WFA). The WFA must provide reliable and robust communication with respect to current standards within automatic alarm systems and fire detection. The number of false alarms must not exceed those of conventional wired alarm systems, and interference from other wireless communication systems must not influence on the WFA.

To obtain a reliable alarm system single points of failure, where a single error can prevent the complete system from behaving correctly must be avoided. This could be done by introducing redundancy on the most critical parts of the alarm system, and/or by applying watchdog timers that can bring the WFA back to stable operation if unintended behavior arises.

The system software needed to implement the WFA must be tested thoroughly throughout the design process, in order to ensure that dead- and live locks does not occur among the various states the software may include. In combination with timers controlling that a given state is always reached before a certain time limit or deadline, a simulation of the timed automata in the WFA can be used to ensure stable and predictable software behavior.

A key issue when using wireless devices is the power consumption. In practical use, the lifetime of a battery powered fire detector should be at least several months - preferably around the lifetime known from the widely used smoke alarms. This means that the WFA must be constructed with extremely low power consumption in mind. By minimising the power consumption, the need of service and battery replacement is also minimised, thereby making the daily use of the system easier.

The usability of a fire alarm system is also an important issue. The user interface must be kept simple, and should only provide the necessary information about the system status. The user interface should also contain different levels of access, so that one group of users only have permission to operate the WFA, whereas another group can be authorized to perform a possible configuration of the WFA as well.

Being a safety critical system, fire detection systems and fire alarms must comply with several standards and precepts to be approved for use and installation. The DS/EN 54 [3] outlines the requirements of fire detection and fire alarm systems, and Precept 232 [4] from the Danish Institute of Fire and Security Technology defines the requirements of automatic fire alarms. A summary of these documents can be found in appendix A.

1.3 Project Scenario

The main purpose of this project is to develop a WFA for use in museums, factories or places where many people gather. As an example of such a place Institute 8 and 16 at Aalborg University is used. An overview of the department is shown on figure 1.1.

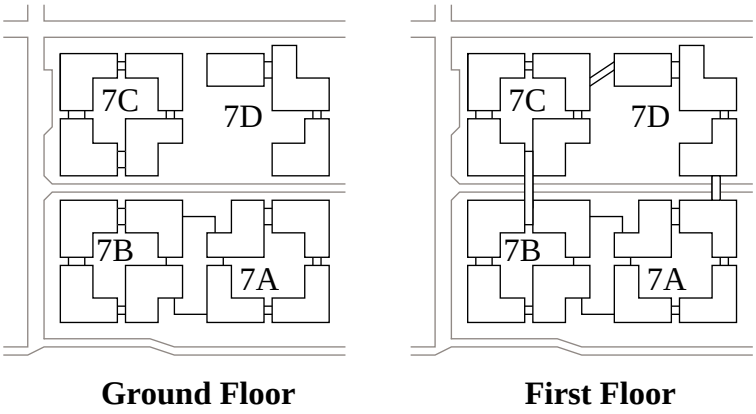

Figure 1.1: *An overview of institute 8 and 16 at Aalborg University on Fredrik Bajers Vej 7.*

A WFA typically consists of a number of zones each controlled by a gateway (GW). The gateway communicates wireless with a number of detectors, e.g. smoke-, temperature- or glass break detectors. Each detector contains a sensor and a wireless transceiver. A gateway is connected to a central unit (CU) which monitors all zones and sends an alarm to the fire department. The connection between GW and CU is often some wired bus or network, using either a star- or a ring topology. These topologies are illustrated on figure 1.2 and figure 1.3.

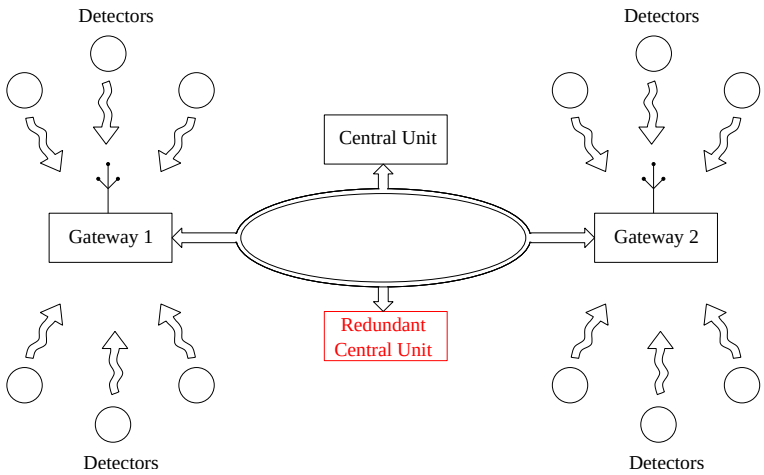

Figure 1.2: *The typical topology of a WFA when using a ring bus with redundant central units.*

Figure 1.2 also illustrates that redundancy could be introduced by adding a redundant central unit which can take over if the primary central unit fails. The redundant central unit could

contain another communication technology than the primary central unit, so that two ways of sending alarms to the fire department can be used. The ring bus connecting gateways and central units could be Can-bus, Profibus or other busses supporting long cable lengths.

Figure 1.3 shows a WFA where the connection between the gateways and the central unit is made using a star topology. This could involve different network standards such as RS485, Ethernet of PoE. Using the approach of figure 1.3 no redundant central unit is introduced. This implies that other means of preventing a single point of failure should be considered.

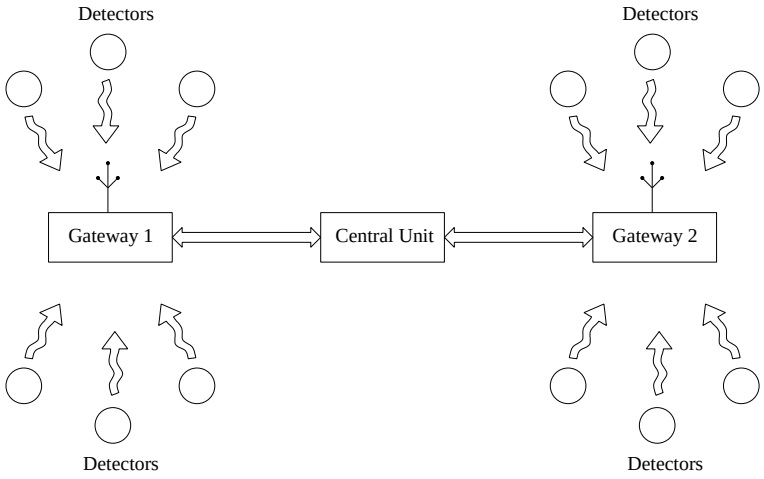

Figure 1.3: *The typical topology of a WFA when using a star topology where each gateway has a connection to the central unit.*

Applying the topology of figure 1.3 to the ground floor of the Control Department on figure 1.1 gives the project scenario shown on figure 1.4.

Each of the buildings 7A, 7B, 7C, and 7D each contain one zone, hence each are controlled by one gateway. The assembly hall between building 7A and 7B also contain one zone and gateway. In each zone a number of detectors are placed. The placement of the detectors does not take the actual ground plan of each building into account. The central unit is located at the caretakers office in the north west corner of building 7A.

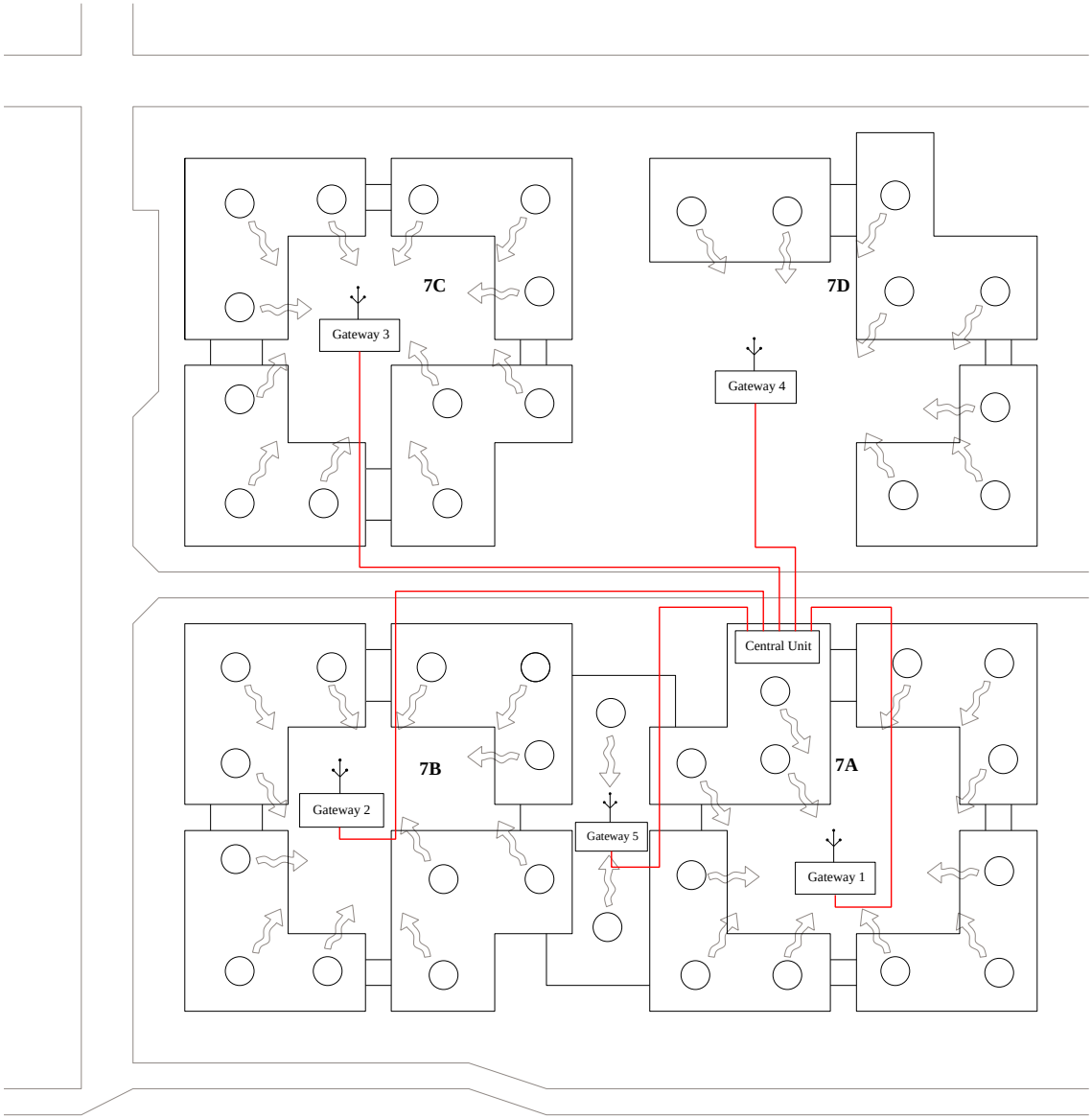

Figure 1.4: *The project scenario, where fire detectors communicate with the central unit through gateways.*

1.4 Scope

The various sensor technologies used in fire detectors embraces many complex engineering problems well outside the purpose of this project. Therefore, in this project, the sensors are treated as black boxes with well defined logical outputs.

Due to the financial limitations under which this project is carried out, only a small number of devices can be used. Because of this, a full scale implementation and test is not possible. Therefore tests are performed in the extent that the number of devices allow.

The limited period of time available for this project makes it impossible to construct a complete alarm system ready for production and sale. Instead a prototype of parts of the system can be analysed, designed, implemented and tested. The main focus of the project is to establish a robust wireless communication system to be used in a wireless fire alarm. In order to test whether such a system was successfully developed, a central unit is also necessary. The chosen network- or bus technology used for connecting the central unit and the gateways is less important, because the developed product is a prototype. In other words, whether a PoE or an RS485 connection is used, in a ring- or star topology, does not influence the outcome of this project.

The prototype developed in this project consists of the hardware devices shown on figure 1.5.

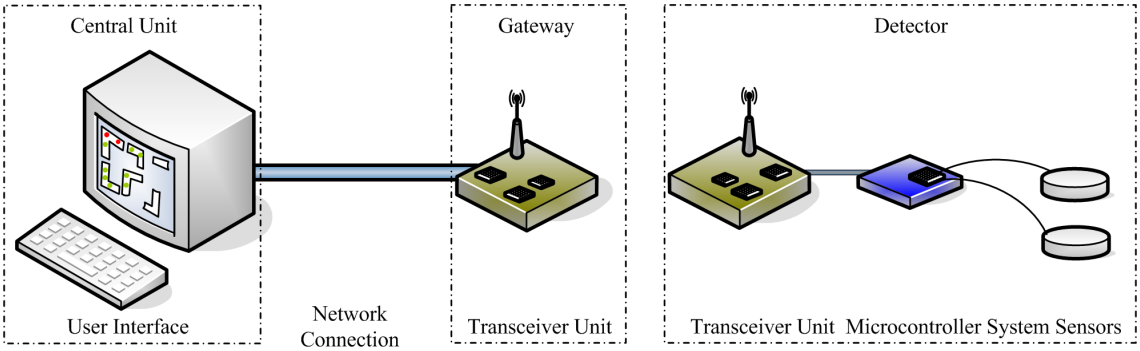

Figure 1.5: *Overview of the hardware used in the wireless fire alarm system.*

In order to demonstrate the implemented functionalities of the developed system, parts of the user interface is implemented. Since the project is a prototype, complete compliance with standards and regulations will not be guaranteed. This would not make it possible to develop the user interface on standard computers, since regulations demand that central units does not contain mechanical or magnetic devices such as hard drives. In a practical implementation of the system, this would not cause a problem, since electrical storage such as flash disks could be used.

Two transceiver units form the wireless network. One unit is used as a gateway while the other is a part of a detector. The detector also contain one or more sensors and a microcontroller system. This system carry out the fire detection and uses a specific network protocol for parsing messages back and forth between detector and central unit.

1.5 Problem Definition

Based on the previous introduction, and considering the scope of the project, the objective of this project is to develop a prototype of a wireless fire alarm to be installed in the project scenario described in section 1.3. The main issue is to develop a robust wireless communication system used between detectors and gateways. Furthermore, a user interface is designed to provide information about the status of the fire alarm and make it possible to configure the alarm system.

In consistency with the standards described in appendix A, the WFA should implement and fulfill the following functionality and requirements:

- Fire Alarm Condition (FAC):
  - Enter FAC max. 3 s after a sensor has detected fire.
  - Be able to receive signals from all zones.
  - A signal from one zone must not falsify reception from another zone.
- Fault Warning Condition (FWC):
  - Detect lost connection to a device within 100 s.
  - Provide means for transfer of battery status.
  - Provide means for transfer of other error messages.
- Disabled Condition (DC):
  - Each zone can be disabled and re-enabled independently.
- Test Condition (TC):
  - Each zone can be tested individually.
  - Test condition can only be entered or canceled by manual operation.
  - Zones in test condition must not prevent other zones from operating normal.

Considering the wireless fire alarm system to be developed in this project as an interface, the following input/outputs must be handled:

- Transfer of fire alarm conditions with an indication of which zone and detector initialised the alarm condition.
- Transfer of error conditions with an indication of which zone and device contains the error.
- Transfer disablement and re-enablement of a given zone.
- Transfer disablement and re-enablement of a given device.
- Transfer reseting of alarm indicators.

- Possibility to configure settings.

The requirements and features listed above form the basis of the acceptance test performed in chapter 8. Each of those features and requirements must be implemented in a robust modularised software. The main program flow and the interaction between hardware and software must be documented. The software must include means to prevent deadlocks and must not allow invalid data to cause errors in program execution.

Part I

Analysis

Part I contains an analysis of which topologies that can be used to establish the communication infrastructure needed to implement a Wireless Fire Alarm. The analysis takes its starting point in the project scenario established in the introduction. Two possible topologies are presented, and the most suited is chosen.

European standards and precepts outlines the requirements for a Wireless Fire Alarm system. These requirements includes a number of functional conditions in which the WFA can operate. Based on these conditions the necessary operations that must be supported by the protocol are analysed.

A variety of existing communication technologies and protocols are analysed, and based on parameters such as range, interference, and power consumption the technology to be used in this project is selected.

System Analysis

CHAPTER 2

This chapter contains an analysis of the system requirements of the wireless fire alarm. First the key aspects to be considered are described. Then two possible solutions than can be applied to the system topology are analysed. By evaluating these two solutions with regards to the key aspects, one solution is chosen. Hereafter the functional conditions and situations that must be handled are analysed.

2.1 Key Aspects

A wireless fire alarm is a safety critical device where unintended behaviour can have fatal consequences. The most effective way to prevent such behaviour is to keep the system as simple as possible. By doing so, the complexity and the necessary amount of software is reduced. The fewer lines of code - the lower risk of faults in the code.

Another way of limiting the system complexity is to concentrate system intelligence on as few devices as possible.

The WFA contains a wireless communication channel in which queueing can occur. One method to reduce the risk of queues arising is to maintain a strict schedule or pattern of the communication. This could be done by using a master/slave principle where all communication is requested from a master. This way the master can control that no queues are build up in the system.

A more detailed discussion of the aspects mentioned above can be found in appendix C.

An important property of a WFA, is that the power consumption must be very low. In order to become a commercial success the battery lifetime should be measured in years or at least months. Therefore steps should be taken to minimise the need of electrical power, when analysing system behaviour in terms of both hardware and software. In practice, this means that all battery powered devices must be in sleep mode as much as possible.

2.2 System Topology

The project scenario and the topology of a wireless fire alarm system is discussed in the introduction. As described, a number of GWs are used to forward signals from the wireless detectors to the CU. Each GW represents a zone. The GWs are connected to the CU in a star topology as shown in the project scenario on figure 1.4 on page 5.

Considering only a part of the scenario, the connection between the CU, two GWs and five detectors can be drawn as illustrated on figure 2.1. The GWs A and B each control a zone

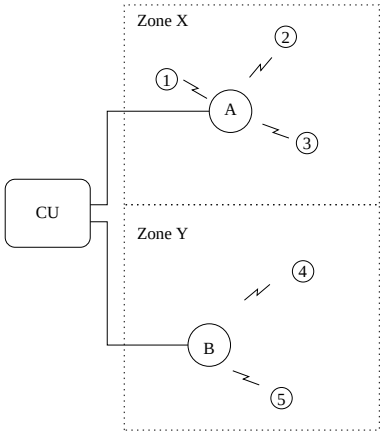

Figure 2.1: *One part of the WFA system, containing two zones controlled by GW A and GW B.*

of three and two detectors respectively. It is not possible for the CU to communicate with the detectors, if the GWs do not function correctly.

As the GWs are responsible of forwarding traffic back and forth between CU and detectors, the case of a GW or GW connection breaking down must be considered. In this case, two possible solutions exist. One is to let the zone continue to be fully functional even if the GW or GW connection breaks down. This is called the "re-routing" solution, since traffic has to be re-routed through other GWs and detectors in order for a detector within the zone to communicate with the CU. The other solution is to gracefully shut down a zone if such a breakdown occurs within the zone. This is called the "graceful shutdown" solution. These two solutions are analysed further in section 2.2.1 and 2.2.2 respectively.

2.2.1 Re-routing Solution

In case of a GW breakdown or the connection to a GW breaks a detector must send the alarm (in case of an alarm situation) to another GW, so this GW can forward the alarm to the CU. However, it may not be possible for a detector to reach an alternative GW due to limited wireless range. This means that the detectors must communicate with each other and forward an alarm received from another detector. This way, the detectors can form a wireless chain from the alerting detector to an alternative GW, which can then reach the CU. Such a system is illustrated on figure 2.2.

The detectors must be placed so they all have a secondary route to the CU. This is the case in the project scenario shown on figure 1.4 on page 5. The total area is approximately 100 m $\times$ 100 m, and with a communication range of approximately 30 m between the detectors, they are all able to reach at least one other detector, given that the range of the wireless communication is minimum 30 m.

The number of hops from a detector to its secondary GW would, in the project scenario be maximum five. The maximum number of hops may be discussed and specified during system

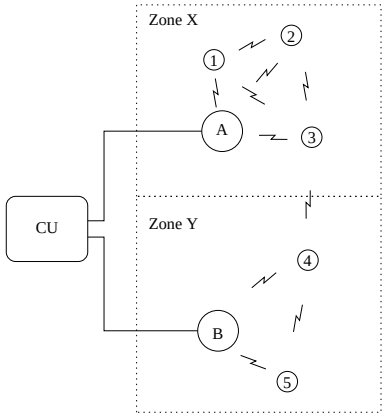

Figure 2.2: *A part of the WFA implemented using the re-routing solution. If gateway B breaks down, detector 5 can use detector 4 and 3 to reach gateway A.*

design phase, but would have to be taken into account when deciding detector positions during an installation of the WFA.

Handling the actual re-routing when a GW breaks down can be done in several ways. One solution is to let an alarming detector broadcast the alarm, if it has sent an alarm to the CU, but did not receive an acknowledgement message from the CU within a certain time. The detectors within range of the broadcasting detector receives the alarm and then starts to broadcast the alarm in order to reach a GW or another detector. In this case a lot of broadcasting is made which increases the possibility of queueing and interference issues in the wireless channel(s). The solution also requires that all detectors never enter sleep mode, because they always have to listen to the radio channel in case a detector needs help to forward its alarm. The fact that a detector does not enter sleep mode, increases the power consumption for the detectors severely and thereby decreases the battery life.

Another solution to the routing issue is to let the CU contain a routing table. The table must contain the shortest path to each detectors and at least one alternative path. In order to make the system redundant, the two paths must not go through the same GW. Except in case of a GW or a GW connection breakdown the detectors can enter sleep mode, because communication must follow a secondary route from the CU to the detectors. Therefore the CU must send a message to the detectors along the particular alternative route, telling them not to enter sleep mode. Naturally this may take some time, since some detectors along the route may already be in sleep mode.

Configuration

The configuration of the re-routing solution can not easily be done manually, because it is difficult to figure out which detectors are able to communicate with each other, because many parameters plays a role in wireless communication.

Therefore the configuration must be done automatically, which makes this system rather com-

plicated. The first configuration step is to make a table of the detectors connected directly to each GW. This table holds the primary routes, as shown in table 2.1.

Detector	GW Address	Detector Address
1	GW A	DET 1
2	GW A	DET 2
3	GW A	DET 3
4	GW B	DET 4
5	GW B	DET 5

Table 2.1: *A detector table holding the primary routes of the system on figure 2.2. The table states that detector 1, 2, and 3 are connected to GW A, and detector 4 and 5 are connected to GW B. Seen from the CU the primary route to detector 1 would be {GWA.DET1}.*

The primary routing table has to be expanded to include secondary routes to the detectors. The secondary routes are the routes going through other detectors. The complete routing table of the subsystem shown on figure 2.2 is shown in table 2.2.

Detector	GW Address	Detector Address	Detector Address	Detector Address
1	GW A	DET 1		
	GW B	DET 4	DET 3	DET 1
2	GW A	DET 2		
	GW B	DET 4	DET 3	DET 2
3	GW A	DET 3		
	GW B	DET 4	DET 3	
4	GW B	DET 4		
	GW A	DET 3	DET 4	
5	GW B	DET 5		
	GW A	DET 3	DET 4	DET 5

Table 2.2: *The complete routing table of the system shown on figure 2.2. Seen from the CU makes the secondary route to detector 1 would be {GWB.DET4.DET3.DET1}.*

The routing table can be made using the Dynamic Source Routing (DSR) protocol [5] and should only be done once, as the system is initially configured. This would normally be done after installation or in case of rebuilding.

Handling Routing

When the CU loses contact with a GW, it looks in the routing table and finds out which detectors it is unable to communicate with through primary routes, and which detectors it must use to reach these detectors through the secondary route. The next time the CU communicates

with the detectors along the secondary route, it must instruct them to start forwarding messages and not enter sleep mode.

2.2.2 Graceful Shutdown Solution

An alternative to forwarding messages through other detectors when a GW or GW connection breaks down, is the "Graceful Shutdown" solution. Here the system is built as a master/slave configuration, where the CU is the master and the detectors are the slaves. If the CU can not reach a GW then it disables the zone controlled by that particular GW and alerts the alarm operator to indicate that an error is present. The other zones continue to operate normally.

A system using the "Graceful Shutdown" solution can remain the simple structure shown on figure 2.1.

Configuration

The system must be configured, and the CU needs a table of gateways and detectors similar to table 2.1. This configuration can be done manually. When all elements are entered in the table, the CU must configure each detector by sending a configuration message. The purpose of such a message is twofold. First of all, the detector must be told which GW it belongs to. Second, the configuration message is used to ensure that the routing table is entered correct, because if the detectors does not send back an acknowledgement message, the CU knows that an error has occurred. This means that the configuration is not done properly.

2.2.3 Selecting Solution

According to standard DS/EN 54 [3] (described in short in appendix A) there is no requirement stating that the WFA should be able to continue normal operation if a breakdown occurs. The standard states that the CU must enter fault warning condition if a zone error occur, and that the zone should remain disabled as long as the error is present. The remaining zones shall continue normal operation.

The re-routing solution requires that the detectors have to communicate with each other in case of a GW and/or GW connection breakdown. This severely increases the complexity of the communication software and the number of functions necessary, which is unwanted. The re-routing solution also requires more intelligent gateways and detectors than the graceful shutdown solution, which increases the complexity.

If the graceful shutdown solution is chosen, some scenarios may occur: If for instance a fire starts near a GW and destroys the cable between the GW and the CU before a sensor discovers the fire, only an error signal is sent to the CU and not a fire alarm signal. This fact is considered satisfactory, because the probability that a fire starts and an error at the WFA appears within very short time is very small.

Reliability is one of the most important issues for this WFA. As mentioned in section 2.1, the best way to make a system reliable is to minimise the amount of software and the complexity. As also mentioned, the standards does not require that a zone should remain alive in case of an error in the zone. Based on these arguments the solution chosen for this project is the "graceful

shutdown" solution. This means that a zone enters fault warning condition if an error occurs in a GW or on a GW connection.

In the "graceful shutdown" solution the GW only needs software to forward messages. No decisions needs to be taken by the GW, which makes the GW a dummy.

2.3 Functional Conditions

The WFA must meet the requirements given in standard DS/EN 54 [3]. The standard states that the WFA shall be capable of being in any combination of the following functional conditions:

- Fire Alarm Condition.
- Fault Warning Condition.
- Disabled Condition.
- Test Condition.

The WFA shall be capable of being simultaneously in any combination of these conditions. E.g. one zone is in disabled condition and another zone is in fault warning condition at the same time. If an error occurs in one zone it must not affect the alarm system for any other zone.

When the WFA is in none of the mentioned conditions, it runs in normal operating condition (quiescent condition). When a fire or an error is detected the corresponding zone must enter fire alarm or fault warning condition respectively. The disabled and test conditions must be activated manually from the CU.

To support the different functional conditions, a number of operations must be possible. These operations are analysed further in the following section.

2.4 Operations

An important issue in this WFA is that the power consumption in the detectors has to be very low, in order to increase the lifetime of the batteries. The different parts of a detector are shown on figure 2.3.

The power consumption is minimised by letting the radio link module in the detectors be in sleep mode as long time as possible.

By applying the "Graceful Shutdown" solution described previously, the communication flow can be kept almost entirely in a master/slave manner, meaning that a master (the CU) request information from the slave (a detector). The only exception is when a detector needs to send an alarm to the CU. Having entirely master/slave communication means that the scheduling of communication can be controlled centrally, thereby maximising the possibility of avoiding collisions and queueing.

If all communication where to be kept in a master/slave relationship then the CU would have to poll each detector and "ask" whether a fire had been detected. According to standard DS/EN 54

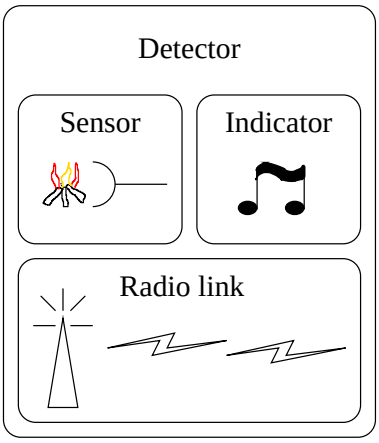

Figure 2.3: *The parts of a wireless fire detector.*

the WFA must enter alarm condition within 3 s after a fire is detected. This would imply that a detector could not sleep for more than maximum three seconds.

This means that a tradeoff exists between the advantage of having master/slave communication and battery lifetime.

According to standard DS/EN 54 the system shall enter fault warning condition within 100 s of the occurrence of a fault. While maintaining the ambition of master/slave communication, this implies that the CU must perform a status check at all GWs and detectors with an interval of maximum 100 s. If the alive check is performed in less than one second, the detector can be in sleep mode in more than 99% of the time.

Because the occurrence of a fire alarm is considered a very seldom event, a very high percentage of the network traffic is pure master/slave communication for checking gateways and detectors.

The number of devices used in a WFA depends on the size and architecture of the building(s) in which it is installed. The standard states that a single fire alarm system must not cover more than 10000 m^2, hence a system using the master/slave principle must not exceed a well defined number of detectors.

This means, that by using the master/slave principle for status requests and putting devices to sleep, a low power consumption and a communication channel with high availability is achieved.

Only the radio link part of a detector is allowed to enter sleep mode. The sensor part must always be alive to discover a fire situation. The sensor part must be able to wake up the radio part of the detector to report the alarm to the CU through a GW.

The GWs never enters sleep mode, because a GW has to be ready to receive alarm signals. The GWs are supplied with power from a power supply, so the power consumption is a less important issue in the GWs. Though the GWs must contain backup batteries in case of a power breakdown.

All together this requires that the WFA can handle the following situations:

- Configuration.
- Status Check.
- Alarm.
- Error.
- Disabled.
- Test.

These situations are described further in the following.

Configuration

In configuration mode, the CU is configured with all GWs and detectors in the system. To make a status check on the detectors, the CU must know how to reach each detector. The configuration is done manual, in order to keep communication software functions at a minimum.

Status Check

As mentioned above the CU must perform status check at all detectors with an interval of 100 s. It is done in normal operation by sending a status request to the detectors in the detector table. The detectors must not sleep when the CU is requesting the status, they must wake up themselves just in time, with the same interval of 100 s.

If the CU does not receive an acknowledgement message from the requested detector, or the message received contains an error message, it must enter fault warning condition. If the CU receives a message and the battery status is below a given limit, the CU must indicate the low battery status.

Alarm

If a sensor detects a fire, it must wake up the radio link at the detector and send an alarm message to the CU through the GW. Standard DS/EN 54 states that an alarm must be indicated at the CU within 3 s of the discovery of the alarm.

Error

If a detector discovers an error meaning that it is unable to report a fire alarm, it must set the error flag. The flag is included in the next status check message, and the CU must enter fault warning condition for the zone where the error is present.

Disabled

At the CU it must be possible to enter disabled condition for each zone independently. When the zone is disabled no fire alarms and/or errors must be indicated. The disabled condition is entered by setting a flag. The next time the CU needs to send a status check message to a device in that zone the flag is checked. If the flag value has changed since the last message sent to that device, a configuration message is sent to the detector instead of the status check message. The configuration message includes the disable flag.

Test

At the CU it must be possible to enter test condition for each zone independently. When the zone is in test condition, tests can be made on the system. The CU is not allowed to forward alarm signals from the test zone to the fire alarm routing equipment. The fire alarm routing equipment normally sends the alarm to the fire station, but not if the alarming zone is in test condition.

Communication Analysis

CHAPTER 3

The aim of the following section is to gain knowledge of existing wireless communication systems in terms of possibilities and weaknesses. By comparing these terms to the key aspect listed in section 2.1 on page 11, the technology to be used in this project is selected.

The use of wireless radio communication in European countries is an area of technologies restricted and controlled by the European Radiocommunications Committee (ERC). The radio frequency spectrum is categorised into distinct frequency bands. Each band is dedicated for specific purposes e.g. television, telephony, data, signalling, etc. which are strictly specified by the standards of ERC [6]. Using most radio bands requires a licence, which is obtained within the country where the radio transmitter/receiver is intended to be used. Few frequency bands does not require a specific licence for the product, instead certain recommendations and standards outlined by The European Telecommunications Standards Institute (ETSI) has to be fulfilled. [7]

Due to the extent of this project, it is most suitable to operate the fire alarm installation in a licence free radio band. This is a decision taken in order to be able to use the system in other European countries, without the need of a licence permission.

3.1 Communication Protocol

Communication between devices in the WFA has to follow a predefined scheme to avoid collisions between data frames. This scheme is the chosen communication protocol.

Developing protocols in data communication systems are divided into several abstraction layers from the level of the application down to the level of the physical transport medium. These layers could be developed for each individual product, as the WFA, resulting in a specific protocol stack only for this product.

Nowadays a full range of commercial radio products are available in the licence free band operating at different frequencies. These frequencies are defined in the Industrial, Scientific and Medical (ISM) radio bands which were originally reserved internationally for non-commercial use of RF electro magnetic purposes. [8]

As the ISM frequency band can be used for all purposes within wireless systems some radio bands are more intensively used for home automation installations such as garage ports, window openers, door bells, car alarm systems, computer mice and keyboards, and toys, etc. The protocols used in such installations often has a varied quality because currently no standardised protocols are used. Especially the 433 MHz band is heavily used in such commercial products causing the equipment to interference with each other, when used at the same time.

3.1.1 Home Automation

Choosing a standardised protocol in contrary to developing an application, a specific protocol has advantages in turns of the possibility of using the same protocol for more than one application.

Home automation systems is an area where a common standardised protocol is appropriate for the needs of communication between every piece of automated house equipment. Such installations could be used for sharing resources and expanding the purpose of the equipment by combining different functionalities into new possibilities. E.g. an integration of the fire alarm system together with a window/door opener system could reduce the damages in case of a fire. This is done by sharing the knowledge that closing windows and doors to the room where the fire is located would slow down the expansion of the fire because less oxygen is admitted to the fire.

Figure 3.1: *Example of a home automation system.*

The WFA could be integrated in a home automation system as the one shown on figure 3.1. Resources as gateways and central unit, described in section 1.3 on page 3, could be shared with other devices. For instance a burglar alarm system would operate in the same manner as a fire alarm. Temperature sensors, could use the gateways for routing temperature data to the heating installation, etc. The WFA contains resources that other home automation systems could take advantage of.

In order to make this possible, a standardised protocol should be used. By knowing which protocol to use, and perhaps a few details such as hardware addresses, other hardware vendors

can interface to the WFA.

3.1.2 Common Wireless Standards

License free frequency bands which are considered as possible solutions for this project are given in 433 MHz, 868 MHz (Europe) / 915 MHz (America), 2.4 GHz and 5.1 GHz bands. The first two are typically used for signalling and alarm systems, while the other two are used for data communication due to a higher data bandwidth.

The 5.1 GHz and 2.4 GHz frequency bands are covered by products as Wi-Fi (wireless fidelity) and Bluetooth products, referring to the IEEE 802.11b,g,a (WLAN) and IEEE 802.15.1 (Bluetooth) standards. These products are becoming more and more popular for broadband data communication purposes. The technology is characterised by high reliability, relatively ease of integration due to approved and well defined protocols together with low price because of huge mass production and already made electrical reference designs.

3.1.3 Protocols

Protocols aimed for control applications are typically available in the lower frequency spectrum. One of these wireless protocols is the RADIAN protocol, which comes in a 433 MHz and a 868 MHz version. RADIAN is mainly developed for the oil and gas industry where the objective is to transport values from one meter to another. RADIAN is not developed nor certified by the IEEE institute but is developed by a French firm *ITRON* in association with the oil industry. For home automation applications RADIAN is not a suitable protocol due to a very narrow functionality in the networking layers. [9]

A new protocol technology such as ZigBee (IEEE 802.15.4) is also aimed at the marked for control and signalling applications. ZigBee is characterised by low data rate and low power consumption and operate in either the 868/915 MHz or 2.4 GHz ISM band. ZigBee has a higher degree of possibilities within home automation compared to RADIAN.

GPRS over GSM is another option for transporting data by a radio link. GPRS requires a working GSM radio network provided by the telephone industry.

Table 3.1 summarises the characteristics of the mentioned technologies, in turns of battery life and bandwidth etc.

Choosing a usable communication protocol from table 3.1 for this project involves an evaluation of the communication needs and demands. Bandwidth, battery life and range as well as available hardware and software resources have an influence of which technology is chosen.

For this project one of the main objects for the radio link is to extend the time between replacing the battery of the detectors, while keeping a reliable, robust, and fault-tolerant connection between gateways and detectors. The amount of transmitted data packets is considered low because only status parameters need to be transmitted once in a while. The range between detectors and gateways in buildings vary within a limited area. The attenuation factor in building materials has influence on the penetration of radio waves. From the above assumptions the range of the transmitter has to be reasonable strong to cover at least 50 meters.

	ZigBee 802.15.4	Bluetooth 802.15.1	Wi-Fi 802.11b	GPRS/GSM 1XRTT/CDMA
Application Focus	Monitoring & Control	Cable Replacement	Web, Email	Voice/Data
System Resource	4 KB-32 KB	250 KB+	1 MB+	16 MB+
Battery Life(days)	100-1000+	1-7	0.1-5	1-7
Nodes Per Network	255/65K+	7	30	1000
Bandwidth(KB/s)	20-250	720	11000+	64-128
Range(meters)	1-75+	1-10+	1-100	1000+
Success Metrics	Low Power, Cost Effective	Cost, Conveniency	Speed, Flexibility	Reach, Quality

Table 3.1: *Various protocols for wireless applications [10].*

As the table indicates, the ZigBee protocol seems to be the most appropriate protocol that satisfy the needs for this project, mainly because of the low power consumption. Bluetooth technology is an approved standard suitable for many applications, but for this project the number of Bluetooth nodes per network might become a limitation as well as the short communication range. Even though the high covering range of Wi-Fi and GPRS are appropriate, it is considered as overkill for this WFA. Both technologies requires too much system resources and battery power which are factors considered as more important than the radio link transmitting range.

3.2 ZigBee Overview

The ZigBee protocol is founded by an alliance of members from some of the leading semiconductor manufacturers worldwide [11]. The alliance has defined a global standard for reliable, cost-effective, low power wireless applications. The lower layers of the ZigBee protocol are defined and specified in collaboration with the IEEE committee to ensure that the Media Access layer, Physical Layer, and Data Link Layer specifications are defined in accordance to common network standards. The specification of the upper protocol stack layers, namely the Network Layer and Application Interface layer are created by the semiconductor manufactures. This organisation method has ensured that ZigBee products are cheap to produce and easy to implement for the end user, because topics as these have been considered during the ZigBee protocol design process. In addition, the stack is small and simple in order to make it easier to include the protocol in micro electronics such as micro controllers. The ZigBee protocol comes in two variations, one version has a network coordinator included and the other version is without the network coordination. The protocol stack occupies approximately 32 KB and 4 KB memory, respectively. The outcome is a protocol based on and approved by the wireless PAN (Personal Area Network) IEEE task group 802.15.4.

The main object of the ZigBee technology is to suite a protocol for control applications, which do not require high data rates, but must have low power, low cost and ease of use for applications such as remote controls, home automation, etc. These applications often require high reliability, which ZigBee provides by using a bi-directional link to establish a connection-oriented data transport service.

Wireless links under IEEE 802.15.4 can operate in three frequency bands. Each frequency band is divided into a number of channels having a given data bandwidth as shown in table 3.2.

Frequency band	**868** MHz	**902-928** MHz	**2.4** GHz
Bandwidth	20 Kb/s	40 Kb/s	250 Kb/s
Channels	1	10	16

Table 3.2: *The bandwidth and number of channels used by ZigBee.*

For accessing the radio channel, ZigBee uses Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA). This means that ZigBee listens to the channel before sending data. If the radio channel is available it quickly sends the data, otherwise it waits a random period of time.

In order to preserve power, sleeping policy is of major concern in the ZigBee protocol. Instead of keeping the radio module awake all the time, ZigBee puts the radio to sleep and wakes it only a short time at a given duty cycle to check whether someone wants to communicate with this module. The wake up time is less than 15 ms, which results in less power consumption and fast connection to the ZigBee network.

The modulation method used for data communication is Frequency Shift Keying (FSK) . Further details on CSMA/CA and FSK is given in appendix B.

3.3 Choice of Radio Band

As mentioned in the previous sections it is possible to use the ZigBee protocol in different frequency bands.

When determining which band to use for the WFA, a criteria such as the environment in which the fire alarm system is installed should be taken into account. As depicted in the project scenario on figure 1.4 on page 5, the system is intended for monitoring larger buildings e.g. factories, offices, and campuses. As the market for wireless networks in offices is increasing, the probability of interference in the same radio band are likewise increasing. Hence the 2.4 GHz frequency is less suitable for operating the fire alarm detectors because this frequency is also used for the Wi-Fi and Bluetooth products.

Operating ZigBee in the 915 MHz band is only advisable in USA. In Europe and most Asian countries the radio band for GSM telephony lies closely to the ZigBee frequency, causing them to interference.

A number of possibilities for implementations of wireless systems in the ISM band which are not exposed for the same amount of interference and disturbance from other devices are left. ERC has designated some frequency bands which they recommend for social alarm systems. These are in the range of 403-404.5 MHz and 868-870 MHz. A project team from the Frequency Management Working Group has furthermore concluded that frequencies within 868-870 MHz is the most suitable band for alarm systems compared to other frequency bands [12]. Along with the fact that ZigBee does not operate in the 403-404.5 frequency band, only one possibility is left.

Based on the recommendations from ERC, the ZigBee protocol based on the 868 MHz radio band is chosen as the wireless communication standard used for the WFA.

When using the ZigBee protocol, only one channel is available in the 868 MHz band for communication. This consequence means that only one ZigBee device can transmit data at a time on the given network. This limitation does not effect the installation of the WFA appreciable because ZigBee takes advantages of the CSMA/CA system. Hence every device can exchange data, however it is not guaranteed exactly when the data can be delivered (See appendix B).

The data bandwidth of 20 Kb/s in the 868 MHz frequency band is considered to be enough for transmitting messages in the WFA. Anyhow the bandwidth might be too small if the system also should be used for other home automation applications.

3.4 Regulations of Alarm Systems in 868 MHz Frequency Band

Even though the chosen frequency is in the licence free band, the usage of the radio transceiver has to comply with certain regulations relating to the use of Short Range Devices (SRD). ERC Recommendation 70-03 sets out the general position on common spectrum allocations for SRD for countries within CEPT (Comité européen de Réglementation Postale) which are countries that complies to the guidelines from ETSI. Moreover the recommendation describes the maximum power levels, channel spacing and duty cycle. Table 3.3 covers annex 7 from ERC/REC 70-03 describing alarm systems in the 868 MHz frequency band.

Frequency Band	Power	Duty Cycle	Channel Spacing
868.600 - 868.700 MHz	10 mW	< 0.1 %	25 kHz
869.200 - 869.250 MHz	10 mW	< 0.1 %	25 kHz
869.250 - 869.300 MHz	10 mW	< 0.1 %	25 kHz
869.650 - 869.700 MHz	25 mW	< 10 %	25 kHz

Table 3.3: *Regulatory parameters for alarms in the 868 MHz radio band. [13]*

Other regulations are given in the European Standard for Electro magnetic compatibility and Radio spectrum Matters (ERM) EN 300 220 for SRD. The purpose of these regulations deal with testing environment and method. Noteworthy testing for wireless communication devices are values as maximum permissible frequency deviation in channel separation, adjacent channel power, spurious emissions etc. [14]

The choice of a radio module device for this project, has to comply with given directives. Hereby the task of getting the final WFA product approved by the authorities is much easier. If no modifications are done to the radio module device, the product does not have to be tested against EN 300 220 one more time.

Part II

Design

Part II contains the software and hardware design of the wireless fire alarm system. First the necessary hardware is chosen and described.
Based on the functional conditions and operations analysed in the previous part, the necessary services to be delivered by the protocol can be designed. A fixed schedule is outlined, and the necessary protocol frame is designed, in order to calculate transmission timing.

The communication system and the protocol is modelled as a timed automata using the UPPAAL tool. This model is used to verify the design and ensure that no live- or deadlocks are present. The UPPAAL model is converted to flowcharts, which can be used for implementing the protocol software.

A graphical user interface is designed according to the requirements and recommendations presented in the DS/EN 54 Standard and Precept 232.

System Design

CHAPTER 4

Based on the analysis of the communication requirements and the chosen protocol, the platform used to construct a prototype of the WFA is designed. After describing each device, the operating mode of the radio boards is chosen.

4.1 ZigBee Radio Boards

The communication analysis in chapter 3 concludes that the wireless communication between gateways and detectors is based on 868 MHz ZigBee technology. The lower levels, physical and MAC layers, of the ZigBee protocol stack were approved by the IEEE in fall 2003, hence the standard is very new. This implies that only a few hardware vendors have products ready for market. Most of the larger semiconductor manufacturers that are members of the ZigBee alliance have chipsets ready, but only a couple of vendors have developer kits with antennas and power supply's included. One company that has a complete kit ready is Adcon Telemetry. The Adcon Addlink 868 Demokit include two 868 MHz ZigBee radio boards, power supply's, batteries, and software. This kit is used to found the basis of the wireless communication system in the WFA.

The Addlink radio board contains an Infinion RF Chip, Antenna, and an eight bit Atmel ATMega8L microcontroller. The board has general purpose I/O available in terms of four user programmable I/O pins and two D/A converters. An RS232 port is also present, so that the board can be used as a radio modem compatible with standard AT or Hayes modem commands.

The available I/O and the microcontroller on the radio board makes it possible to implement detector- and gateway software directly on the radio boards. This way the radio boards appear as a stand-alone application. However, in order to use this possibility, a Software Developers Kit (SDK) is necessary. Without this, it is not possible to flash application software onto the board, without erasing the existing firmware controlling the radio communication. Details on hardware registers and addresses are also not available without purchasing an SDK.

Since the price of this SDK is approximately twice the price of the radio boards, obtaining an SDK is not possible within the financial boundaries of this project. Instead the radio boards are used as modems through the RS232 connection.

This choice has consequences on the hardware needed for both detector, gateway and central unit. These consequences are described in the following.

4.2 Detector Hardware

Interfacing the radio board through the RS232 and the fact that the detector software can not be embedded in the radio board microcontroller means that a dedicated microcontroller system

is needed as part of the detector. The microcontroller must have an USART or an RS232 port available in order to interface with the radio board.

Microcontroller

A Texas Instruments microcontroller unit (MCU) is chosen as the central processor of the detector. The MSP430F149 is a 16 bit MCU containing onboard memory, USART and I/O connections. This MCU is somewhat more powerful than necessary for the purpose, but because an SDK and evaluation board is already available, this MCU is found appropriate. To obtain an RS232 port a MAX3223CPP RS232 driver is connected to the USART pins of the MCU. The properties of the chosen MCU are similar to the Atmel MCU on the radio boards, so that software implemented on the Texas MCU could easily be ported to the Atmel processor if an SDK is available.

Peripherals

As illustrated on figure 2.3 on page 17 the detector contains a sensor and indicator, or actuator. Due to reasons explained in section 1.4 on page 6 the sensor part is treated as a black box. Therefore a conventional ionization smoke detector is used as sensor input to the detector.

As actuator a piezo buzzer is used. In addition to this coloured LEDs are added to indicate status. These LEDs are only intended to be used on a prototype to support development, debugging and test.

In order to set the hardware address of the detector a set of dip switches is also added. The address is needed to identify which detector the CU is communicating with.

4.3 Gateway Hardware

The gateway hardware only consist of a radio board. The fact that software can not be embedded in the gateway does not have a negative effect, since the gateway should be kept as transparent as possible, as described in section 2.2.3 on page 15. When using the radio board as a modem the data received on the wireless link is present on the RS232 connection, and vice versa.

Because the RS232 is the only cabled interface available on the radio board, this connection type is used as the cabled network between central unit and gateway. In a complete implementation of the WFA this would not be satisfactory due to the limited cable length of the RS232 standard. However, since the WFA is a prototype the bus type is not vital, as described in section 1.4 on page 6. Furthermore, the conversion between an RS232 connection and for instance an RS485 connection is somewhat straightforward, but considered irrelevant compared to the purpose of this project.

4.4 Central Unit Hardware

The central unit hardware is a standard PC running Linux. The PC must contain an RS232 compatible serial port in order to interface with the radio board. This type of hardware can only be used in a prototype, since an alarm system must not use any storage devices containing mechanical or magnetical parts according to the DS/EN 54 Standard [3].

4.5 Hardware Summary

The hardware constituting the WFA is shown on figure 4.1.

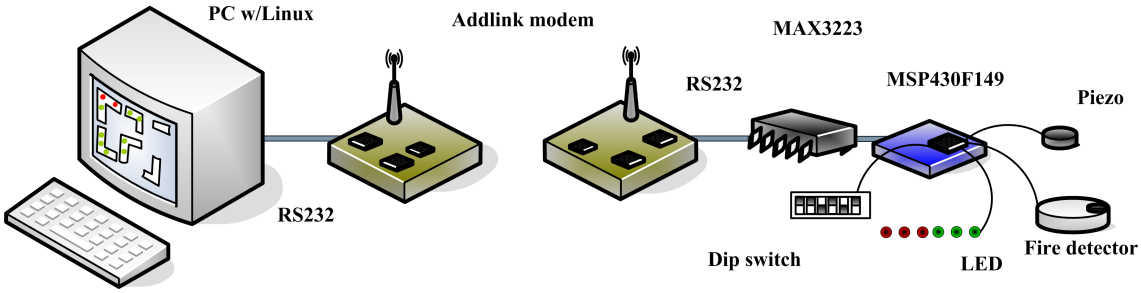

Figure 4.1: *The hardware constituting the WFA.*

The technical details relating to each hardware device is listed in table 4.1.

Detector:	
Radio Board:	Adcon Addlink, model 868addlink-NB-MC
Evaluation Board:	MSP-TS430PM64
MCU:	Texas Instruments, MSP430F149
Sensor:	Ionization Smoke Detector
Actuator:	Piezo Buzzer
Addressing:	Dip switches
Indicators:	LEDs
Gateway:	
Radio Board:	Adcon Addlink, model 868addlink-NB-MC
Central Unit:	
PC:	900 MHz AMD Duron
OS:	Knoppix Linux (Debian), kernel 2.4.22

Table 4.1: *Technical details on the hardware constituting the WFA.*

A simplified schematic of the detector is shown on figure 4.2. All detector devices except the ionizating smoke detector uses a 3.3 V power supply and is able to be powered by batteries for wireless operation. The ionizating smoke detector uses a 9 V battery as power supply.

4.6 Radio Board Operating Modes

The Addlink radio boards supports five different modes of operation depending on the desired level of security and functionality. These modes are:

- Transparent.

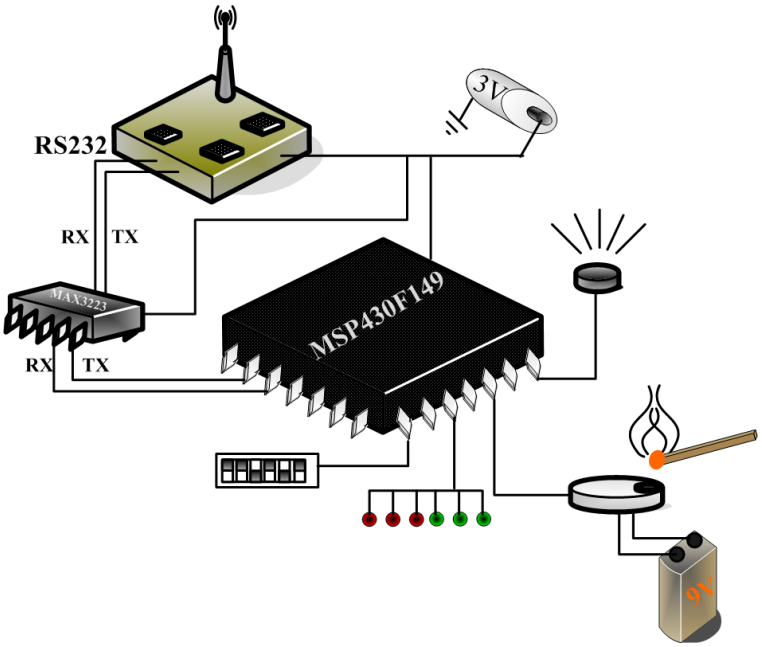

Figure 4.2: *Simplified hardware schematic of the detector.*

- Transparent Secured.
- Transparent Addressed Secured.
- I/O Copy Mode (Master and Slave).
- Demo Mode (Master and Slave).

Transparent

In transparent mode the radio link behaves like a wired serial link. Data received on the serial link is transmitted on the wireless link, and data received on the wireless link is transmitted on the serial link.

No flow control is present in transparent mode, and the behaviour of the radio modules is similar to the half duplex function of an RS485 cable.

Transparent Secured

In transparent secured mode a data flow control between the radio modules is added to the functionality of transparent mode. This way the risk of data loss is much lower. The receiving module checks each received frame and requests retransmission if necessary. The maximum number of retransmissions allowed can be configured in a register. The default value is two.

Transparent Addressed Secured

Transparent addressed secured adds addressing to the transparent secured mode. A device number is added in front of every data frame in order to identify the recipient.

I/O Copy Mode (Master and Slave)

I/O copy mode is a special purpose mode where the I/O pins of the master are sampled and the value is transmitted to the slaves. The slaves then copies the value to its output pins. After a 1 s break the slave samples its input pins and sends them to the master, which then copies the value to its output pins.

Demo Mode (Master and Slave)

In demo mode the boards communicate in a master/slave manner. The purpose is to demonstrate that the boards are working as intended. This is done by flashing various LEDs to indicate successful transmissions, lost packets etc.

Selecting Mode

The operational mode used for the WFA is Transparent Secured. This mode has been chosen, because it is desirable to use as transparent a mode as possible, because the modes are not standardised by ZigBee and therefore compatibility with other vendors can not be guaranteed. However, it is desirable to utilize the retransmissions made possible when using Transparent Secured Mode instead of plain Transparent Mode. The Transparent Addressed Secured mode may seem the most suitable mode for the WFA system, but it was not possible to configure the boards to use this mode. The enclosed configuration utility returns an error when trying to write the registers needed to activate this mode.

Protocol Design

CHAPTER 5

This chapter describes the design of the application layer protocol for the WFA. First the necessary services found in the previous analysis are summarised, and their parameters are described. After this, sequence diagrams of each service are used to schedule the communication and timing parameters are calculated. A model of the WFA is build and verified in UPPAAL.

5.1 Services

As described in the analysis in section 2.3 on page 16, the WFA should be able to handle the situations:

- Configuration.
- Status Check.
- Alarm.
- Error.
- Disabled.
- Test.

Test is a mode where the system continue normal operation, but the central unit ensures that alarms generated in zones being in test mode are not forwarded to the fire department. *Error* is a situation where some error has occurred in the system. Any indication of errors in detectors can be included as a parameter when doing a status check. *Disabled* is a mode where a detector or a zone is disabled. The disabling of a detector or a zone can be handled by a configuration service.

By applying the simplifications mentioned above, the application layer protocol should contain the following services:

- Configuration.
- Status Check.
- Alarm.

These services are described further in the following three subsections.

5.1.1 Configuration

The configuration service must provide means for an operator to configure a detector. The configurable parameters include the address of the gateway through which the detector communicates with the CU and the setting of a number of flags. When a detector is configured it sends a reply back to the CU with the actual value of the configurable parameters, thereby making it possible for the CU to ensure that the configuration is done properly. The semantics of the configuration services are:

- `Config(adr,gwadr,flags)`
  Call from CU to configure a detector. The command sets up the gateway address and flags.
  - `adr`: Address of the detector being configured.
  - `gwadr`: Address of the assigned gateway through which the detector communicates with the central unit.
  - `flags`: Value of the configuring flags. The flag byte contains the following bits:
    * Disable.
    * Error.
    * Buzzer.
- `ConfigReply(srcadr,gwadr,flags)`
  Confirmation of a received `Config()` command. The message confirms the actual value of the parameters configured using `Config()`.
  - `srcadr`: Address of the detector sending the reply.
  - `gwadr`: Address of the assigned gateway.
  - `flags`: Value of the flags.
    * Disable.
    * Error.
    * Buzzer.

Sequence Diagram

The configuration services are visualised in the sequence diagram of figure 5.1.

5.1.2 Status Check

The status check service must ensure that the WFA is operating correct, and that the communication between devices is working. The service provides means for checking the status of both gateways and detectors. When the CU requests the status of a gateway, the gateway responds by sending a reply. The status response from a detector contains more valuable information. Upon request from the CU, a detector returns a response with its battery status and an error type. The error type is used to indicate if the detector has found some internal error, e.g. a suspicious value from a sensor. The semantics of the status check services are:

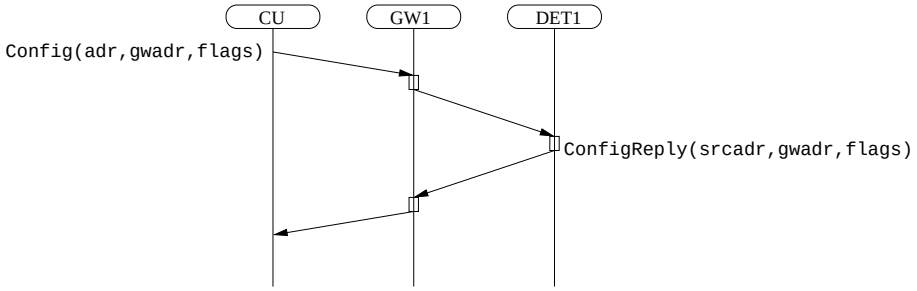

Figure 5.1: *Sequence diagram when the WFA is in configuration mode.*

- `GwStatus(gwadr)`
  Tests that communication between the CU and a gateway is working properly.
  - `gwadr`: The address of the gateway being tested.
- `GwStatusReply(gwadr)`
  Reply which confirms that the connection between GW and CU is present.
  - `gwadr`: The address of the gateway confirming the connection.
- `Status(adr)`
  Tests that communication between the CU and a detector is working properly.
  - `adr`: The address of the detector being tested.
- `StatusReply(srcadr,batstatus)`
  Returns some status parameters of a detector to the CU, thereby ensuring that communication is present, and that the detector works as intended.
  - `srcadr`: Address of the detector sending the reply.
  - `batstatus`: The battery level of the detector.
  - `flags`: Value of the flags.
    * Disable.
    * Error.
    * Buzzer.

Sequence Diagrams

The status check services are visualised in the sequence diagram of figure 5.2 and 5.3.

5.1.3 Alarm

The alarm service is used when a detector has been triggered by a sensor indicating that a fire has been discovered. The detector sends a message to the CU indicating which detector has observed the alarm and the alarm type. If a detector with both smoke- and temperature sensors sends an alarm, the alarm type is used to indicate if smoke or temperature caused the alarm.

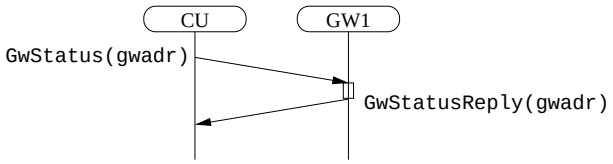

Figure 5.2: *Sequence diagram when the WFA is testing communication between gateway and central unit.*

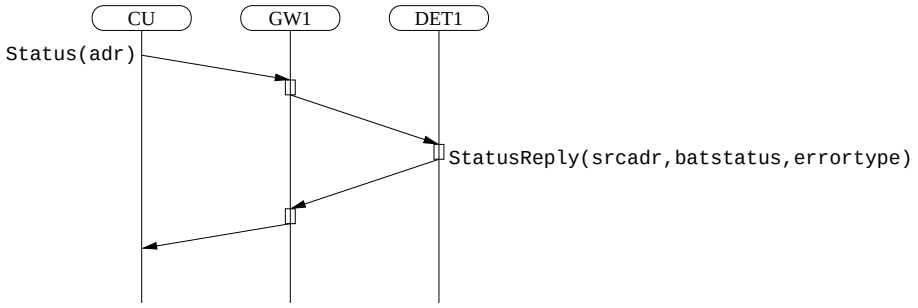

Figure 5.3: *Sequence diagram when the WFA is testing communication between central unit and a detector.*

- `Alarm(srcadr,alarmtype)`
  Transfer of an alarm from a detector to the central unit.
  - `srcadr`: Address of the detector sending the alarm.
  - `alarmtype`: Value indicating which type of alarm it is. This could be e.g. "smoke" or "temperature" if more than one type of sensor is included in a detector.
- `AlarmReply(adr,alarmtype)`
  Confirmation message from CU back to the detector which sent the `Alarm()`.
  - `adr`: Address of the detector that sent the `Alarm()`.
  - `alarmtype`: Value indicating the alarm type.
- `AlarmStop(adr)`
  Disables an alarm, and puts a detector back in normal operation mode.
  - `adr`: Address of the detector that sent the `Alarm()`.
- `AlarmStopReply(srcadr)`
  Confirmation message from the detector back to CU.
  - `srcadr`: Address of the detector sending the reply.

Sequence Diagram

The alarm services are visualised in the sequence diagram of figure 5.4.

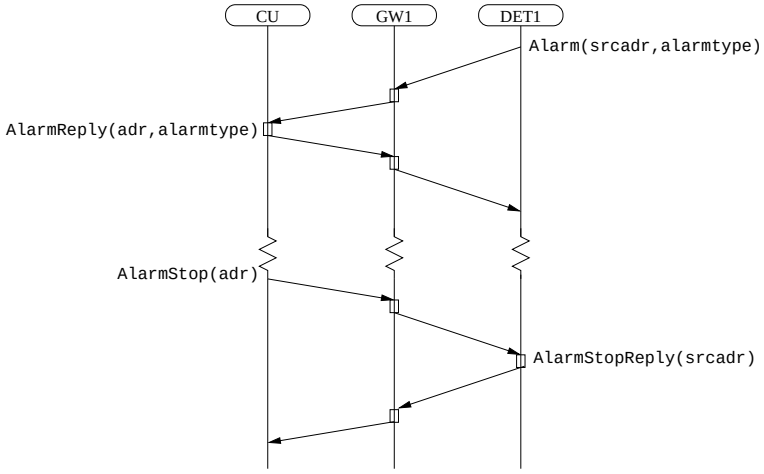

Figure 5.4: *Sequence diagram when the WFA is in alarm mode.*

5.2 Scheduling

When the fire alarm is operating in normal mode, i.e. no alarms are present, the central unit sends status requests to each gateway and detector in a fixed pattern, so that after the status exchange is finished, a detector can sleep for a fixed period of time T_{Sleep}. The status check of a particular detector is repeated every T_{Cycle}. A sequence diagram of the normal operation sequence is shown in figure 5.5. The vertical black full-drawn lines represents devices being idle, and the vertical dotted lines represents devices being in sleep mode.

T_{Cycle} at the CU is 100 s. During the 100 s CU must perform status check at all detectors in the WFA. T_{Sleep} at the detectors is 100 s minus the operation time in the detector. The operation time is very low, which makes T_{Cycle} become very close to 100 s.

When CU has received a reply message it waits before sending a new status check message. CU must wait as long as it takes an alarm message to reach CU. These timing issues are described in section 5.3.

If a user want to reconfigure a zone or a detector (e.g. disable or enable), a manual operation is done at the user interface. When this happens, CU sends a configuration message to the affected detector(s) when it is scheduled to communicate with the detector, instead of a status check message.

If CU does not receive a status reply message within expected time, CU presumes that the detector is dead. The next time CU is scheduled to send a status check message to that detector, it sends a config message instead. This is done to try to make the detector operate normal again.

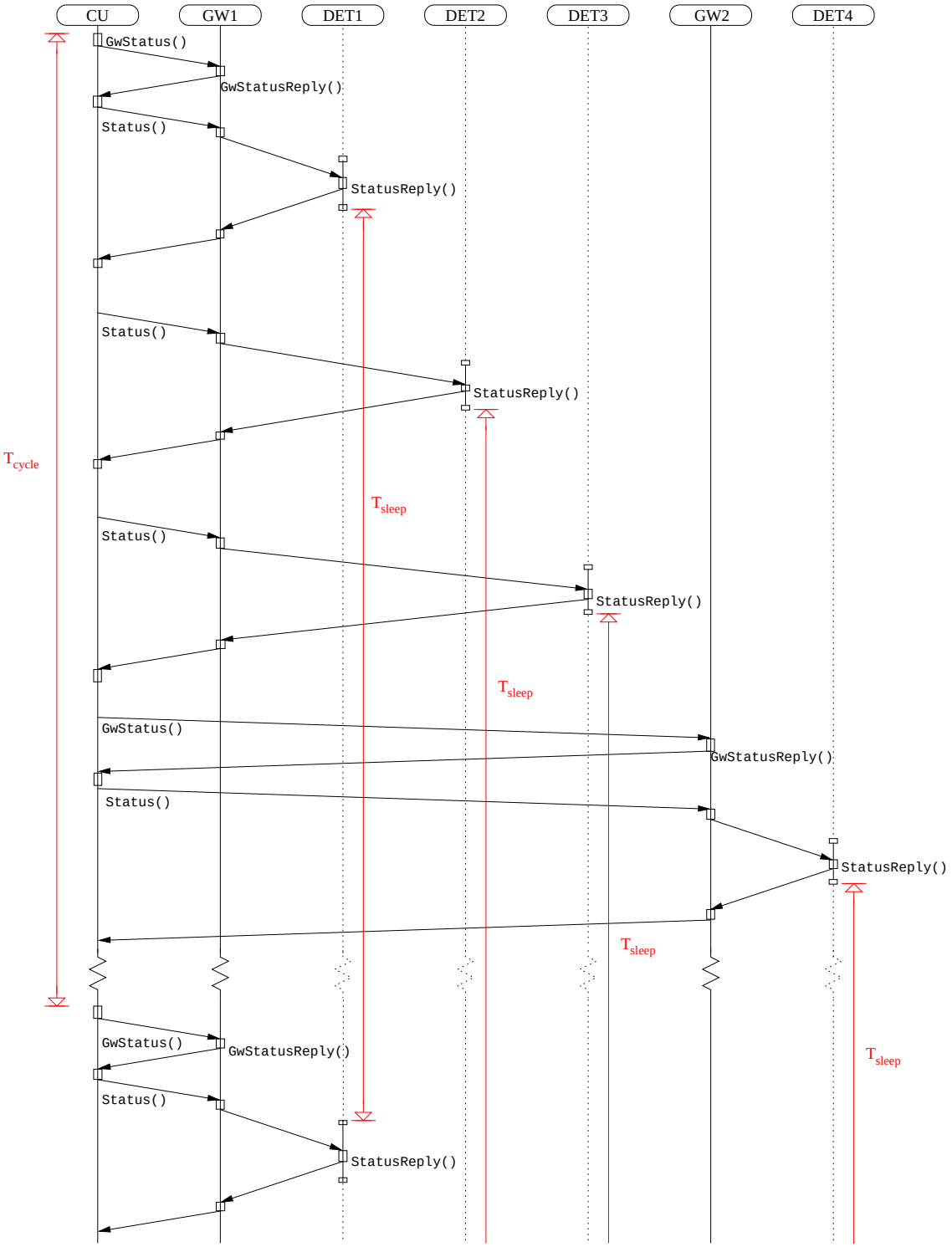

Figure 5.5: *Sequence diagram when the WFA is in normal operation. Black full-drawn lines represents devices being idle, and dotted lines represents devices being in sleep mode. Red arrows indicates time intervals and black arrows represents ongoing communication.*

In some cases special considerations must be made. These cases are described in the following.

5.2.1 Lost Contact to a Detector

The system is in normal operation the most of the time, since it is assumed that alarms and faults occur seldom. However the system must be able to handle all situations. If for instance a detector does not wake up or the radio link is jammed by another radio communication, the gateway retransmits the frame twice (see figure 5.6), but the detector does not reply. The CU has started a timer (Timeout) when it sends the message to the detector. If CU does not receive a reply message before the timer runs out, it presumes that the detector is dead.

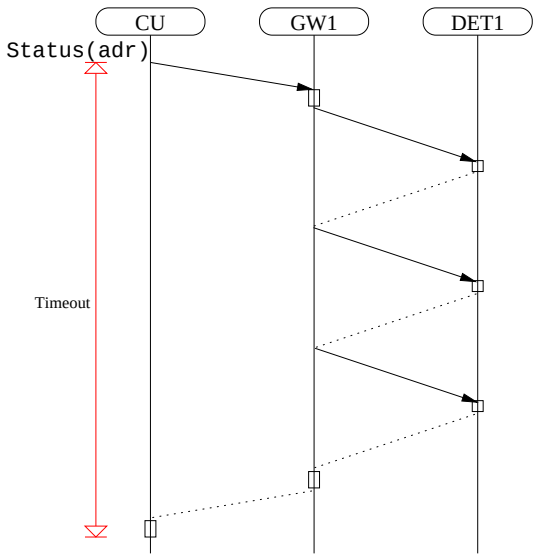

Figure 5.6: *A gateway retransmits a frame twice, but the detector does not reply.*

Central unit must enter fault warning condition for the specific zone when the timer runs out. The CU continues to follow the schedule, meaning that the next time the CU is scheduled to communicate with the missing detector, CU sends a configuration message to the detector as. If the detector responds, the CU leaves fault warning condition and enters normal operation (if no other zone faults have occurred meanwhile).

5.2.2 System Power Up Configuration

When the system (or a zone) has been installed, it must be configured for the first time. No scheduling is made before the first configuration message is received, meaning that the detectors can not enter sleep mode, because they do not know when to wake up. The CU sends a configuring message to all detectors and they enter sleep mode. When the CU has received a config reply message from all detectors it enters normal operation condition.

If the CU does not receive reply from all detectors, it continues to send out the configuration message until it has got reply from all detectors. The CU must not enter alarm condition before the system is well configured. The power up configuration is shown in figure 5.7.

The full-drawn vertical lines represents the devices being in operation mode, and the dotted lines represents the devices being in sleep mode.

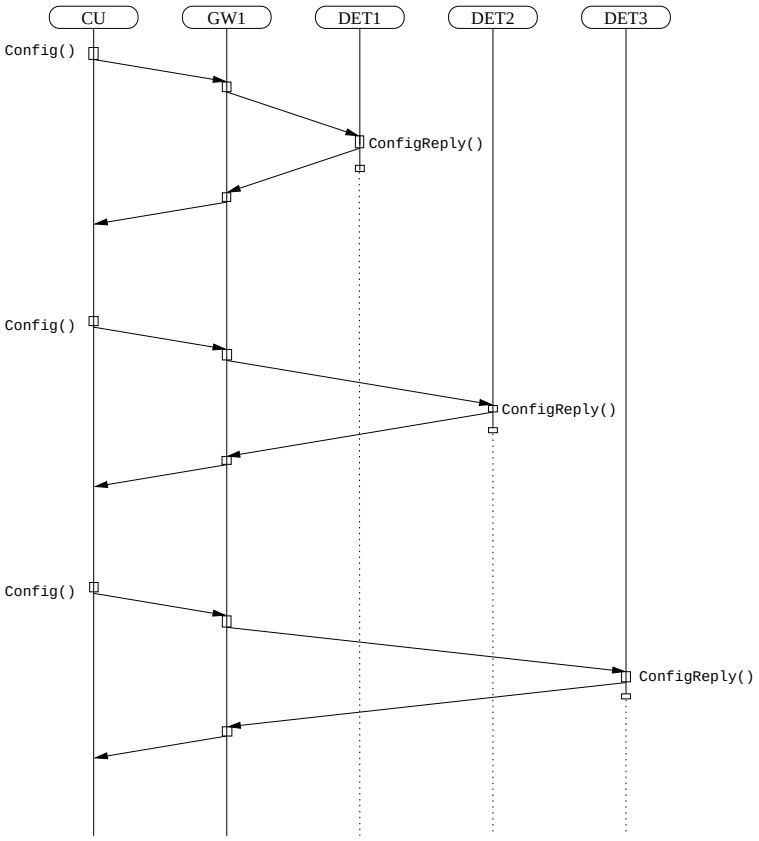

Figure 5.7: *The first configuration message to the detectors. A detector enters sleep mode after it has sent a reply message. When the power up sequence is finished, the time scheduling is started.*

5.2.3 Disabling/Enabling of a Zone

It must be possible to disable and enable each zone independently. When a zone is set to be disabled, all detectors in the zone must be disabled. This is controlled by the CU. A detector is disabled when it receives a configuration message from CU with the disabled flag set.

When a detector is disabled, it may not start its audible warning equipment and it may not send an alarm message to the CU. After disabling, the detector enters sleeps mode, and the scheduled status check continues to run, as if the detector is not disabled.

A detector is disabled until it receives a configuration message where the disabled flag is not set. The enabling configuration message is sent according to the time schedule instead of a status check message. This means that a detector can remain disabled up to 100 s after the enabling is activated at the CU. This is due to the trade off between power consumption and status/config message frequency.

The disabling of a zone is shown on figure 5.8.

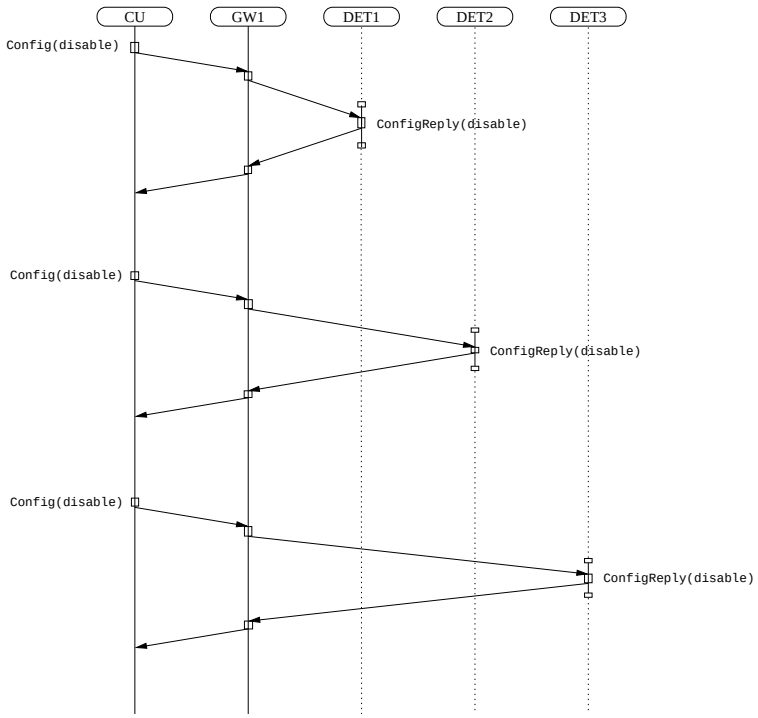

Figure 5.8: *Disabling of a zone requires disablement of all detectors in the given zone. The `config(disable)` message is sent instead of a status check message.*

The alarm handling requires special attention during the disabling and enabling of a zone.

In the time between starting the disablement of a zone and until all detectors have received the disabling message, a detector, which has not been disabled at that time, can send an alarm message to the CU. If CU receives an alarm message during the disablement of a zone, it must not enter fire alarm condition. Instead it sends back an alarm stop message to the detector, and it stops transmitting alarm messages.

In the time between starting to enable a zone until all detectors have received the enabling message, a detector, which has been enabled at the time, can send an alarm message to the CU. If CU receives an alarm message during the enabling of a zone, it must enter fire alarm condition, since the zone is meant to be enabled.

5.3 Timing Considerations

The timing aspects of the execution of the protocol communication sequences are determined by the wireless transmission speed and the frame length of transmitted data.

5.3.1 Frame Definition

The data frame format for the WFA can be designed either as a variable frame length or by defining a constant frame length for the transmitted data messages.

Using a variable frame length has several advantages. It can reduce the need of bandwidth when transferring short data messages, which reduces the transmission time. Variable frame length gives the opportunity to expand the protocol for including more functionality. The drawback is the need of a method to distinguish when a frame starts or stops, and not knowing how long each frame is. In time critical systems, it is often important to know the exact transmission time of each frame in advance. Using variable frame size the timing aspects can be difficult to guarantee, because the transmission time depend on the frame length.

A solution with constant frame size solves the timing considerations. For the WFA, the developed fire alarm protocol has a number of well defined static messages. Using a constant frame size may result in overhead for some messages but makes computation of overall data transfer time possible.

The protocol communication is carried out using serial RS232 communication. The standard of RS232 uses either 5, 6, 7 or, 8 bit ASCII bytes for data exchange. Bits indicates when a byte start and stop. This project uses 8 bit ASCII, one start/stop bit and no control bits (8N1).

From the above considerations, the frame format and size can be defined in the following. Figure 5.9 shows the chosen frame structure.

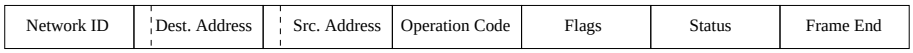

Figure 5.9: *Structure of the protocol frame, with a total size of 7 bytes.*

Each of the protocol fields are included when a frame is transmitted, even though some of the fields might not contain any information. The nine fields of figure 5.9 are described below:

- **Network ID: (8 bit)**
  Determines which fire alarm system the unit belongs to. This is to make sure that no conflicts happens with another similar WFA in the area.
- **Destination Address: (7 bit)**
  A unique destination address for the message within the selected zone.
- **Source Address: (7 bit)**
  A unique source address for the device where the message comes from.
- **Operation Code: (8 bit)**
  Small code which indicate the given service of the message.
- **Flags: (8 bit)**
  Field which contain more than one option. Is used to check on/off parameters.

- **Status: (8 bit)**
  A value for reading or writing status parameters, such as the battery level, wireless signal strength, etc.
- **Frame End: (8 bit)**
  Confirming the frame is ended by setting a special command containing new line.

Both the destination and the source addresses contains extra bit, determining whether the frame is intended for the CU or a detector. Applying the extra bit makes both the destination and the source fields occupying 8 its each.

Using the serial transmission method, each 8 bit character is encapsulated between a logical start and stop bit. This method makes each byte occupying 10 bit. The total space for a single frame becomes a total of 70 bit.

Also a special GW status frame is introduced. This frame is for checking whether the gateway is alive or not. The special gateway checking frame can be seen on figure 5.10.

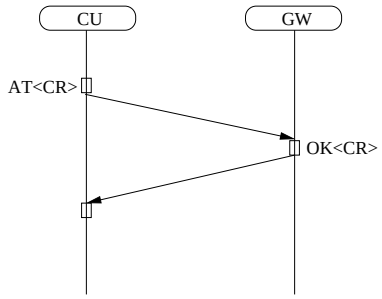

Figure 5.10: *Two frames for checking a gateway.*

As shown on figure 5.10, the frame consist of a Hayes AT commando. The reply is also shown in the figure. If the GW is ready the reply is OK. Both frames are terminated with a carriage return character as specified by the radio modem hardware.

5.3.2 Transmission Speed

The other timing aspect for the WFA is the broadcasting speed. From the hardware overview in section 4.5 on page 31 three links are shown where data messages are transferred:

- Central Unit to Gateway, using cabled serial link.
- Gateway to Detector, using wireless link.
- Detector to micro controller, using cabled serial link.

Having the radio modem devices use their default serial communication speed gives a baud rate for the cabled link of 19.2 kbit/s and the wireless link is specified to 10.0 kbit/s.

For each beginning of a new frame, the radio modem has an additional service time of 25 ms to stabilise the wireless link by sending a preamble.

Because wireless radio links are susceptible to interference, transmitting data can easily be erroneous. The ZigBee protocol retransmit frames if the content are detected to be wrong. The default setting of the maximum numbers of retransmitting tries are two. The timing calculation has to take care of a worst case situation were the maximum number of retransmissions is used every time.

5.3.3 Transmission Timing

Calculating the transmission time depends on what type of message is transmitted. Moreover the following calculated times are only theoretical, it does not take the protocol computation time into account, namely the propagation delay, which is carried out in every device. The propagation delay is expected to be negligible compared to the transmission time.

Four types of data transmissions are possible in the system:

- Transferring messages from CU to detector.
- Transferring messages from a detector to CU.
- Checking connection to GW.
- Reply message from GW.

Considering the transmission from CU to a detector and its reply gives the transmission times $T_{\text{Trans-bus}}$, $T_{\text{Trans-wireless}}$ as depicted on figure 5.11.

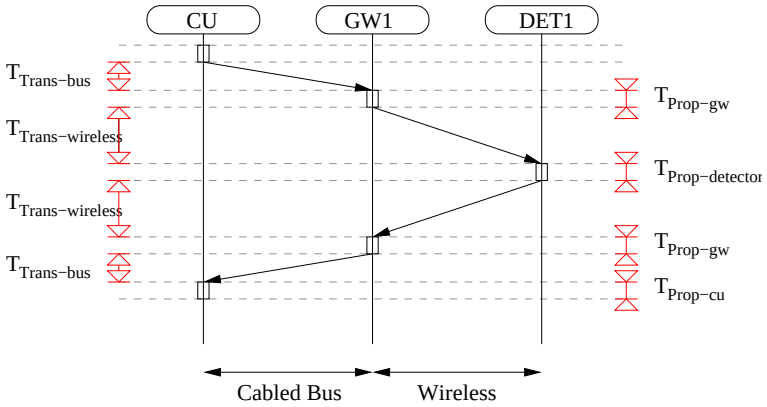

Figure 5.11: *Timing parameters to be considered when scheduling the communication.*

$$T_{\text{Trans-bus}} = \frac{\text{frame length}}{\text{baudrate cabled}} \Rightarrow \frac{70 \text{ bit}}{19200 \text{ bit/sec}} = 3.6 \text{ ms} \tag{5.1}$$

$$\begin{aligned} T_{\text{Trans-wireless}} &= \frac{\text{frame length}}{\text{baudrate wireless}} \cdot 3 \text{ transmissions} + \text{ preamble} \\ &\Rightarrow \frac{70 \text{ bit}}{10000 \text{ bit/sec}} \cdot 3 + 25 \text{ ms} = 46 \text{ ms} \end{aligned} \tag{5.2}$$

As discussed previously, the propagation delays and propagation delays for $T_{\text{Prop-gw}}$, $T_{\text{Prop-detector}}$, and $T_{\text{Prop-cu}}$ are not considered.

The total transmission time for delivering a data message from the CU to the detector, and receiving a reply, as shown on figure 5.11 gives:

$$\text{Transmission time} = T_{\text{Trans-bus}} \cdot 2 + T_{\text{Trans-wireless}} \cdot 2 = 99.3 \text{ ms} \tag{5.3}$$

The times needed for checking the gateway status are given as:

$$T_{\text{GW-request}} = \frac{\text{AT commando}}{\text{baudrate cabled}} = \frac{30 \text{ bit}}{19200 \text{ bit/sec}} = 1.5 \text{ ms} \tag{5.4}$$

$$T_{\text{GW-reply}} = \frac{\text{AT reply}}{\text{baudrate cabled}} = \frac{30 \text{ bit}}{19200 \text{ bit/sec}} = 1.5 \text{ ms} \tag{5.5}$$

Which gives a 3 ms gateway checking time.

5.4 System Proportion

From the timing calculations in section 5.3.3 an estimate of the maximum number of detectors in the WFA can be determined.

The DS/EN 54 standard [3] requires that any malfunctions in the WFA system are reported within 100 seconds to the CU. By this requirement, every detector has to be checked once within this time period.

Then the total number of devices in a system can not exceed $\frac{100 \text{ s}}{99.3 \text{ ms / device}} \approx 1000$ devices.

To ensure that an alarm message from a detector can reach the CU within a given time, the CU has to schedule a short break between a status check for the next detector.

The transfer time of an alarm message sent from the detector to the CU is calculated by adding the results of equation 5.1 and 5.2, which gives equation 5.6:

$$\begin{aligned} T_{\text{Alarm-transfer}} &= 1 \cdot T_{\text{Trans-wireless}} + 1 \cdot T_{\text{Trans-wire}} \\ T_{\text{Alarm-transfer}} &= 46 \text{ ms} + 3.6 \text{ ms} = 49.6 \text{ ms} \end{aligned} \tag{5.6}$$

The CU then has to wait this $T_{Channel\text{-}idle}$ time, which reduces the number of devices in the system. The scheduling of an alarm message can be seen on figure 5.12.

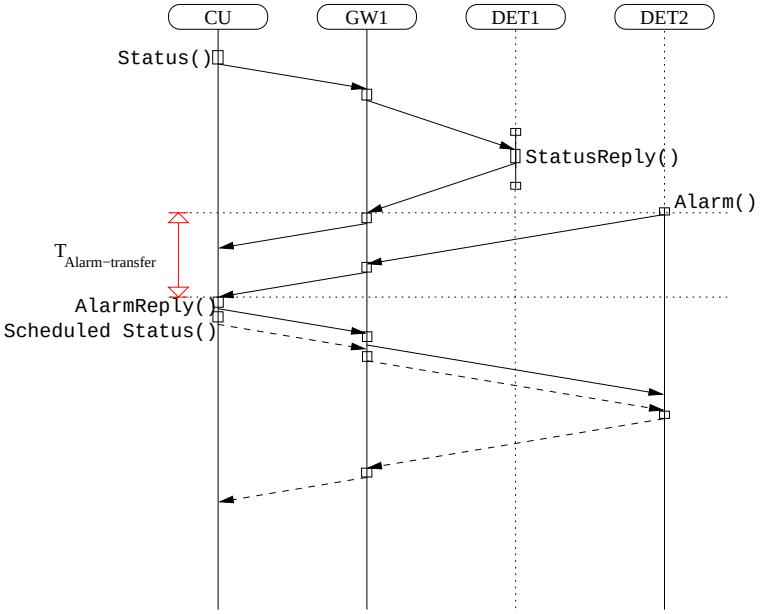

Figure 5.12: *The time to transfer an alarm from a detector to CU is the Alarm transfer time. CU must be idle in a given time, so an alarm message will be transferred.*

To ensure that alarms can be received at CU, the radio channel is left idle for a period between each status exchange. The length of the period must be equal to the time needed to transfer an alarm message from a detector to CU. This time is labelled $T_{Alarm\text{-}transfer}$ on figure 5.12. The time needed for a status exchange and an alarm transfer is calculated by adding equation 5.3 and 5.6 and subtracting equation 5.1 which becomes equation 5.7.

$$99.3 \text{ ms} + 49.6 \text{ ms} - 3.6 = 145.3 \text{ ms} \tag{5.7}$$

This result implies that each detector requires ≈ 150 ms of available radio channel. This fact reduces the number of detectors to 670 as shown in equation 5.8.

$$\frac{100 \text{ s}}{150 \text{ ms/detector}} \approx 670 \text{ detectors} \tag{5.8}$$

The DS/EN 54 standard [3] states that the covering area of a single fire alarm system must not cover more than 10000 m^2. A zone area may not exceed a given size, according to the number of rooms in the zone. If there is only one room the zone floor area can be up to 1600 m^2 and if there is more than 15 rooms the zone floor area must not exceed 400 m^2.

Table 5.1 shows the needed number of detectors in one fire alarm system.

Number of Rooms	Maximum Floor Area	Maximum Number of Zones	Area for each Detector	Number of Detectors
15	400 m^2	25	15 m^2	375
1	1600 m^2	6	15 m^2	640

Table 5.1: *The maximum number of zones and detectors in one fire alarm system.*

The maximum number of detectors needed is 640 detectors. This fact states that this WFA has no limitations according to maximum number of detectors in a system.

5.5 Timed Automata

This section describes the timed automata for the WFA. A model for the timed automata is built, simulated and verified in UPPAAL which is an integrated tool environment for modelling, validation and verification of real-time systems [15].

The model contains six processes: CU, RS-232-TX, RS-232-RX, GW, Wireless and Det, representing the central unit, RS-232 full duplex, gateway, wireless link and detector. The system is simulated with one CU, one RS-232 link, one GW, one wireless link and a number of detectors. This represents one zone in the WFA. The CU shall be seen as a part of the complete central unit, namely the part belonging to the specific zone. The complete central unit will have multiple CU processes running in parallel, one for each zone.

The central unit and the detectors can communicate using two different paths through RS-232, GW and the wireless link. One path is reserved for alarms and alarm replies. This path starts and ends at the detector (see figure 5.4). The UPPAAL channels using this path are all named with an "a_" in the beginning. The other path is used for messages sent from the central unit. These are `Status()`, `Config()`, and `AlarmStop()` messages. The reply messages from the detectors are also using this path.

The GW status check is not included in the UPPAAL model, since the time used to make this check is considered very low. Additionally the GW status check is only performed once every 100 s, and is thereby neglected.

In the UPPAAL model a number of global arrays are used to describe a condition or the purpose of a synchronisation. For instance `dead[d]:=1` says that the detector with address `d` is dead and `status[d]:=1` means that a status message is sent to the detector with address `d`.

The processes are described in the following.

5.5.1 Central Unit

The model of the central unit has six states. These are: Idle, Receiving, Alarm Reply, Alarm, Alarm Stop, and Alarm Stop Reply. The initial state is Idle. The model is explained from the UPPAAL model shown on figure 5.13.

Idle: The model is made with one global timer. Each detector is set to wake up with a given interval, and the CU is set to send messages with the same interval. In the idle state the CU

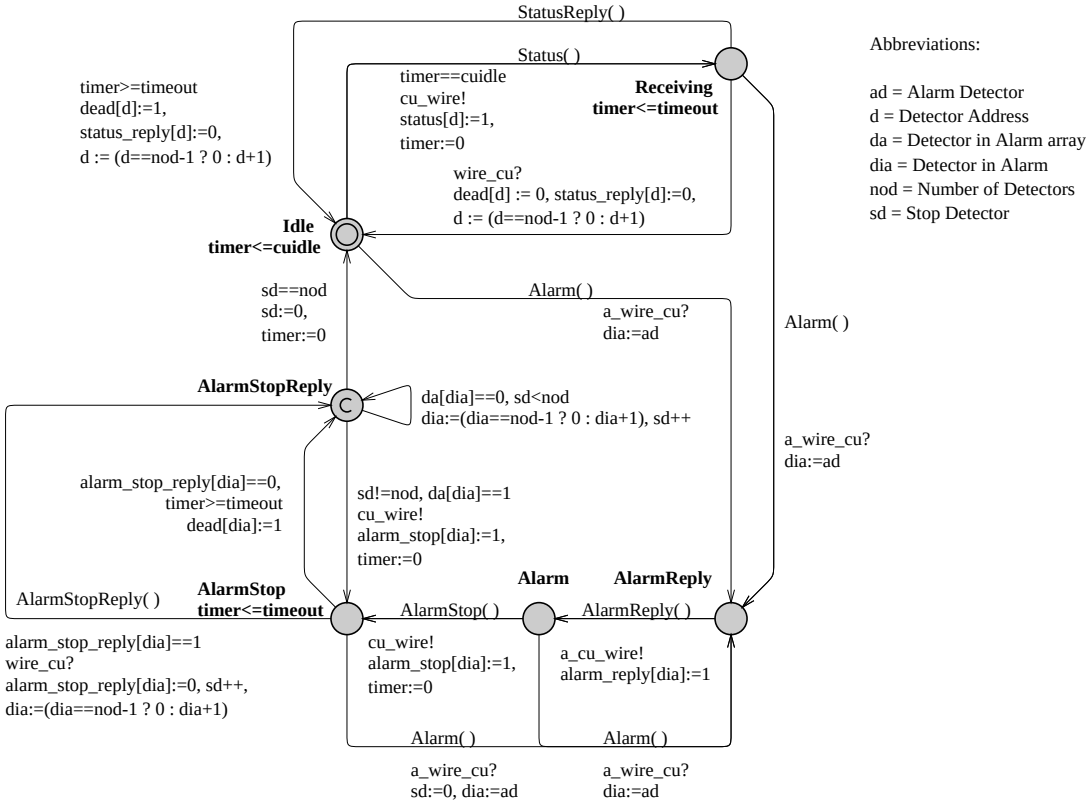

Figure 5.13: *The* UPPAAL *model of the central unit. The messages to be send/received are shown on the figure (e.g. Status()).*

waits until the first detector wakes up. This is indicated with `timer==cuidle` where `cuidle` indicates the time where CU is allowed to sleep, which is $\frac{100s}{\text{Number of detectors}}$.

d is a global variable and indicates the detector address. d is initialised to 0, which corresponds to the first detector. An invariant in the Idle state, makes sure the CU leaves the Idle state after `cuidle` s. The timer is reset when CU sends a message. CU leaves the Idle state when it receives an alarm message or when it is time to send a status or a config message to a detector. The timing in the WFA is the same whether a status or a config messages are send. Therefore only the status messages are used for the UPPAAL model.

After the CU has sent a status or a config message, it enters the Receiving state.

Receiving: In the receiving state the CU waits for a `StatusReply()` message from the detector. If the CU does not receive a reply message before the timeout, it states that the detector is dead and it must enter fault warning condition. The CU must stay in fault warning condition as long as any element in `dead[d]` is set.

Before the CU enters the Idle state again, the detector address is increased by one or set to 0 if this is the last detector and the timer is reset.

Alarm Reply: If the CU receives an alarm it enters the Alarm Reply state and sends a fire alarm signal to the fire department. CU sends back a reply message to the detector in alarm. The address of the detector in alarm is set in the `ad` variable, and is saved in the `dia` variable when the CU receives the alarm message.

Alarm: In the Alarm state the CU stays until it receives a new alarm, or the user want to stop the alarm. If a new alarm message is received, CU goes to the Alarm Reply state and sends an alarm reply message to the new detector in alarm.

Alarm Stop: When the user wants to stop the alarm, the CU sends an `AlarmStop()` message to the detectors who are in the alarm state. These messages are not time scheduled, they are sent out one by one. When an `AlarmStop()` message is sent, the CU enters the Alarm Stop state, where it waits for an `AlarmStopReply()` message before it goes to the Alarm Stop Reply state. If it does not receive the `AlarmStopReply()` message before the timeout, it labels the detector dead, and enters the Alarm Stop Reply state.

If a new alarm message is received in the Alarm Stop state the CU enters the alarm state again, and the user has to press the stop alarm button again to stop the alarm.

Alarm Stop Reply: In the Alarm Stop Reply state the CU finds out if one or more detectors are in Alarm state and thereby needs to be stopped. If not all detectors has been stopped, CU sends a new `AlarmStop()` message to the next detector in Alarm state. The `da[]` array contains all detectors in Alarm state, and an `AlarmStop()` message is sent to the detector addresses which are in Alarm state.

If a detector does not send back a reply message, a new `AlarmStop()` message is sent to that detector. This continues until CU receives the reply message.

When all detectors have been stopped, the CU enters the Idle state.

5.5.2 TX and RX

The model of the RS-232 link is divided into two processes namely TX and RX. This is done because the RS-232 link is full duplex. Each process has two states, which are idle and forward message. The initial state is idle. The UPPAAL model is shown in figure 5.14 and on figure 5.15.

Idle: When TX or RX are in idle state, they are ready to receive a message to forward. The processes can go to the forwarding state when they receive a message to forward on one of the two paths.

Forwarding: When RX or TX receives a message they resets their timers `rxt` and `txt`. In the forwarding state they must stay as long time it takes to transmit the frame in the cable. That time is given in the variable `wtt` (wired transmission time).

5.5.3 Gateway

The model of the gateway has three states. These are: Idle, forward to CU, and forward to detector. The initial state is idle. The model is explained from the UPPAAL model shown on figure 5.16.

The GW is very simple since it does not make any decisions, it is only used to forward messages.

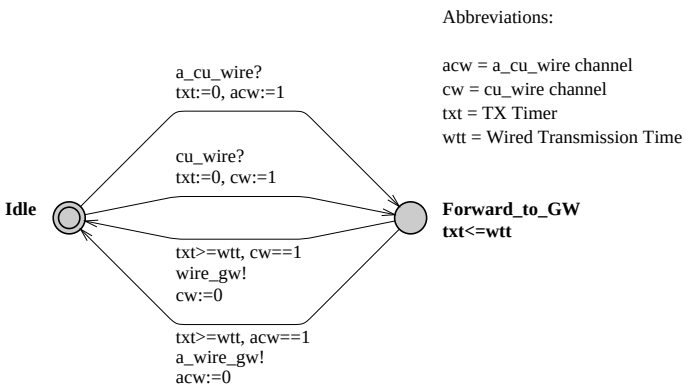

Figure 5.14: *The* UPPAAL *model of the TX part of the RS-232 link.*

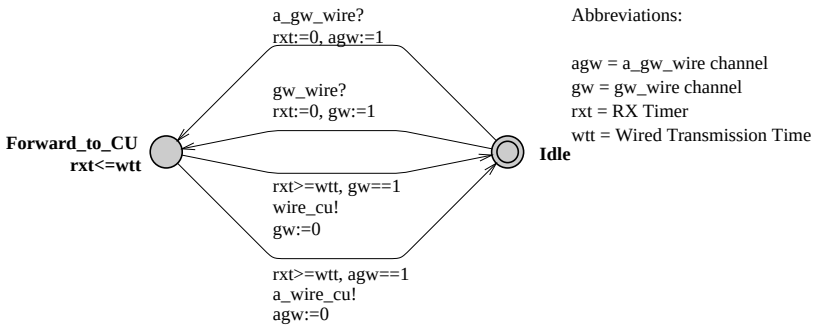

Figure 5.15: *The* UPPAAL *model of the RX part of the RS-232 link.*

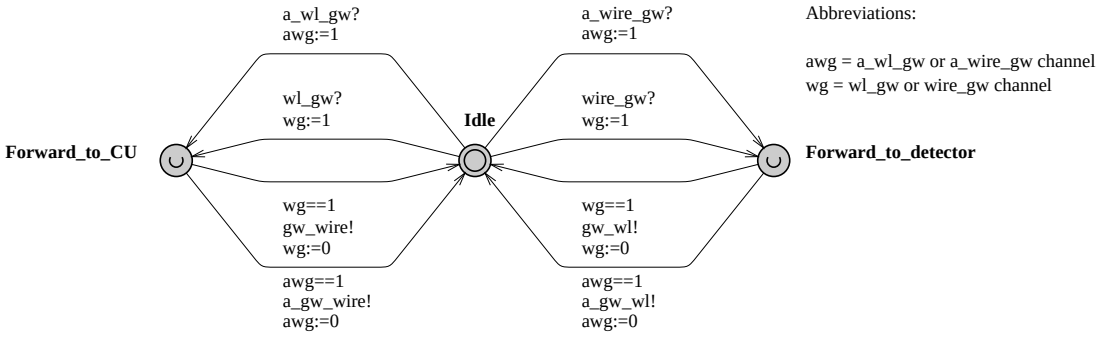

Figure 5.16: *The* UPPAAL *model of the gateway.*

Idle: When the GW is in the idle state, it waits for a message from the RS-232 cable or the wireless link.

Forward to detector: When the GW receives a message from the RS-232 cable to forward to the wireless link, it sends the message and goes back to the idle state.

Forward to CU: When the GW receives a message from the wireless link to forward to the RS-232 cable, it sends the message and goes back to the idle state.

The forwarding states are urgent, since it is assumed that no time passes in the GW.

5.5.4 Wireless

The wireless link is modelled in only one process, since it is a simplex communication method.

The model of the wireless link has three states. These are: Idle, forwarding to GW, and forwarding to Detector. The model is explained from the UPPAAL model shown on figure 5.17.

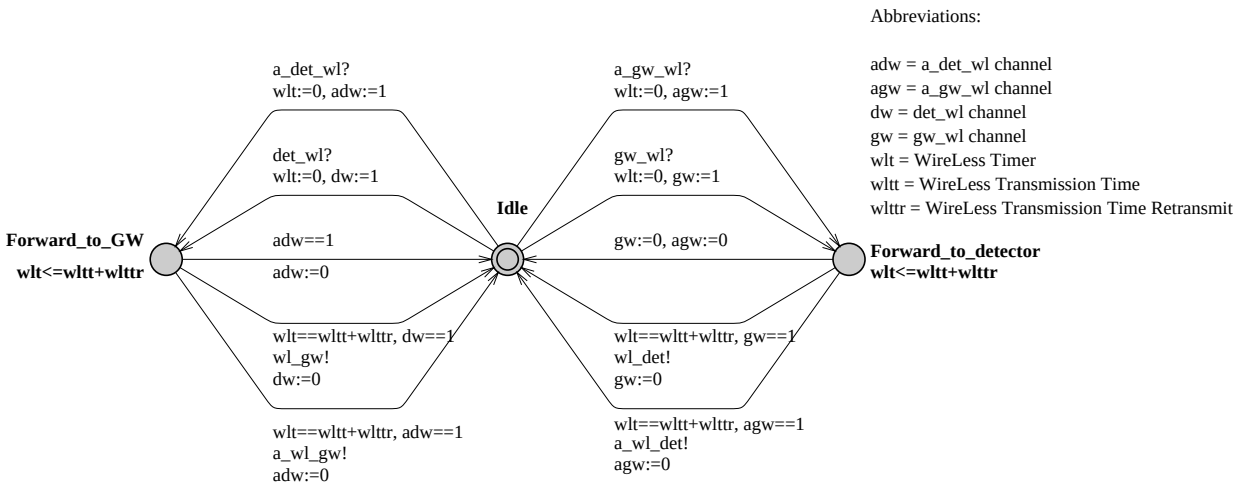

Figure 5.17: *The* UPPAAL *model of the wireless link.*

Idle: In the idle state the wireless link is waiting for messages to forward from a detector or the GW.

Forward to GW: When the wireless link receives a message from a detector it resets the timer `wlt` and stays in the state in `wltt` (wireless transmission time) + maybe the `wlttr` (wireless transmission time retransmit). If the message is an alarm message, the detector might try to send the message while another message is to be sent through the air. If this is the case, the message can not be sent, and wireless enters the idle state again. It is then up to the detector to retransmit the alarm message.

Forward to Detector: When the wireless link is forwarding to a detector the timer is reset and the message can be forwarded. But if the detector can not be reached, e.g. it is out of range or the battery level is too low, the frame can be lost. Then the wireless link goes back to the idle state, and it is now up to CU to handle this situation.

5.5.5 Detector

The model of the detector has seven states. These are: Idle, status, sleep, Local Alarm, Send Alarm, Alarm, and Local Alarm Stop. The initial state is Idle. The model is explained from the UPPAAL model shown on figure 5.18.

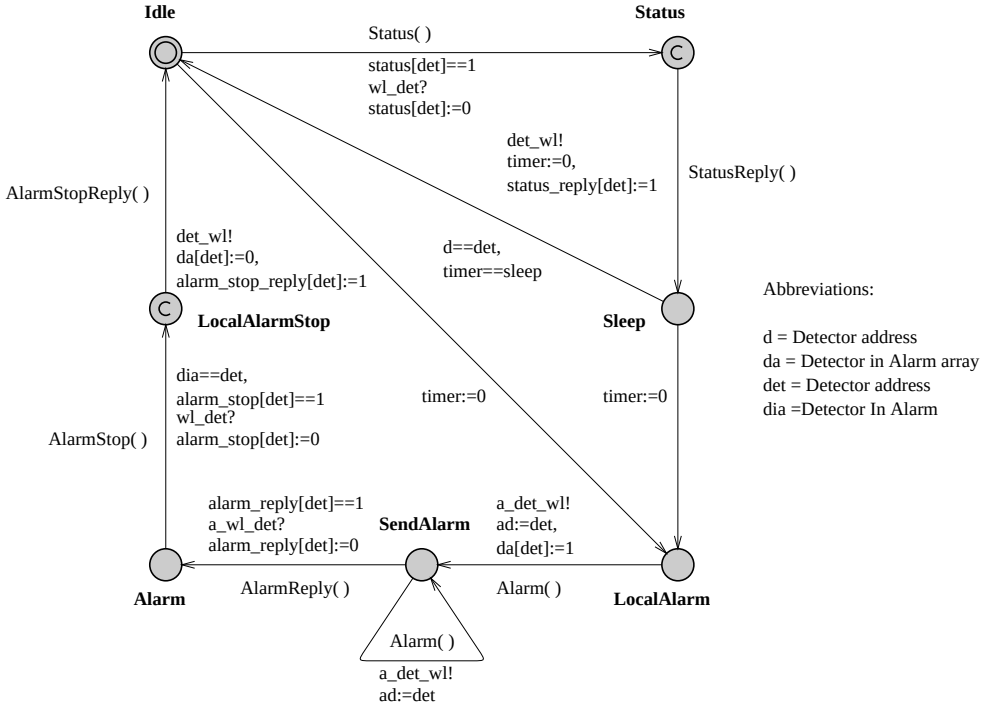

Figure 5.18: *The UPPAAL model of a detector. The messages to be send/received are shown on the figure (e.g. Status()).*

Idle: In the idle state the detector is ready to receive a message from the CU, or it can enter the Local Alarm state if the sensor detects a fire.

Status: When the detector receives a status or config message it enters the status state. This state is committed, meaning that the detector immediately sends back a reply message to the CU. Then the timer is reset, and the detector enters sleep mode.

Sleep: The detector is in sleep mode until it wakes up after the given time or until an alarm occurs. If it is woken up by the alarm it goes to the Local Alarm state, otherwise it goes to the Idle state.

Local Alarm: The detector must enter the Local Alarm state if a fire is detected. The detector is only able to enter the alarm state if it is not disabled and if no error has occurred in the detector. This is not implemented in this UPPAAL model since this model is primarily made for timing considerations.

The detector sends the alarm through the urgent channel `a_det_wl!`, and enters the Send Alarm state.

Send Alarm: In the Send Alarm state the detector waits for an `AlarmReply()` message from the CU. If it does not receive the alarm reply message before a timer runs out, it sends the alarm message once again. It continues to send alarm messages until it receives a reply message. When the reply is received, the detector enters the Alarm state.

Alarm: In the alarm state the detector starts its audible warning equipment if that is enabled, and it stays in the alarm state until it receives an `AlarmStop()` message. It does not enter the sleep state, since it must be ready to silence when it receives the `AlarmStop()` message.

Local Alarm Stop: The detector enters the Local Alarm Stop state when it receives a `AlarmStop()` message. This state is committed, meaning that it immediately sends back an `AlarmStopReply()` message and enters the Idle state.

5.5.6 Verifying in Uppaal

UPPAAL has a verifier, to verify that the UPPAAL model is behaving as desired. An important verification is the deadlock check, verified by the `A[] not deadlock` command. UPPAAL has verified that no deadlocks are discovered for the model of the WFA with up to five detectors. If more than five detectors is put into the system, the verifier requires to many CPU and memory resources from the simulation computer and the verification crashes.

The CU is not allowed to be in the Idle state when one or more detectors are in Alarm state. This can be verified by applying the following command:

```
A[] not (Central_Unit.Idle and (Detector0.Alarm or Detector1.Alarm))
```

This property is also satisfied with five detectors.

The time scheduling and timing considerations can also be verified using the UPPAAL verifier. In section 5.4 the maximum number of detectors for the system is calculated to 670. Two detectors in the UPPAAL model can each represent 670/2=335 detectors by decreasing the CU idle time. The CU idle time is:

$$\frac{100 \text{ s}}{\text{Number of detectors}} \Rightarrow \frac{100 \text{ s}}{670} \approx 150 \text{ ms} \tag{5.9}$$

The `cuidle` = 150 ms is inserted in the UPPAAL model and the verifier can tell if the system is able to make the status check at all detectors without running out of time. This is verified by this command:

```
E<> (Central_Unit.Idle and Detector0.Sleep and Detector1.Sleep)
```

This property is satisfied for the system. It can also be verified that CU is able to receive an alarm message within 3 s. The following command tells if the alarm message can be received at the CU, and the alarm reply is sent back in 200 ms.

```
E<> (Central_Unit.Alarm and Detector0.Sleep and Detector1.Alarm and
Detector1.timer <= 200)
```

This property is also satisfied and states that an alarm is able to reach CU within 200 ms for 670 detectors.

All properties are satisfied, and the UPPAAL model is used for the implementation of the wireless fire alarm system.

Uppaal Model on CD-ROM:
The UPPAAL model can be found in `uppaal/model/` on the enclosed CD-ROM.

5.5.7 Flowcharts

The UPPAAL model is the foundation to the implementation of the WFA. Here follows two flowcharts made from the UPPAAL model. The flowcharts describes the central unit and a detector respectively, since it is these two components which are implemented.

The flowcharts are shown on figure 5.19 and in figure 5.20. Rounded boxes are added to the flowcharts to indicate the relation from the UPPAAL models to the flowcharts. From these flowcharts the software implementation is made.

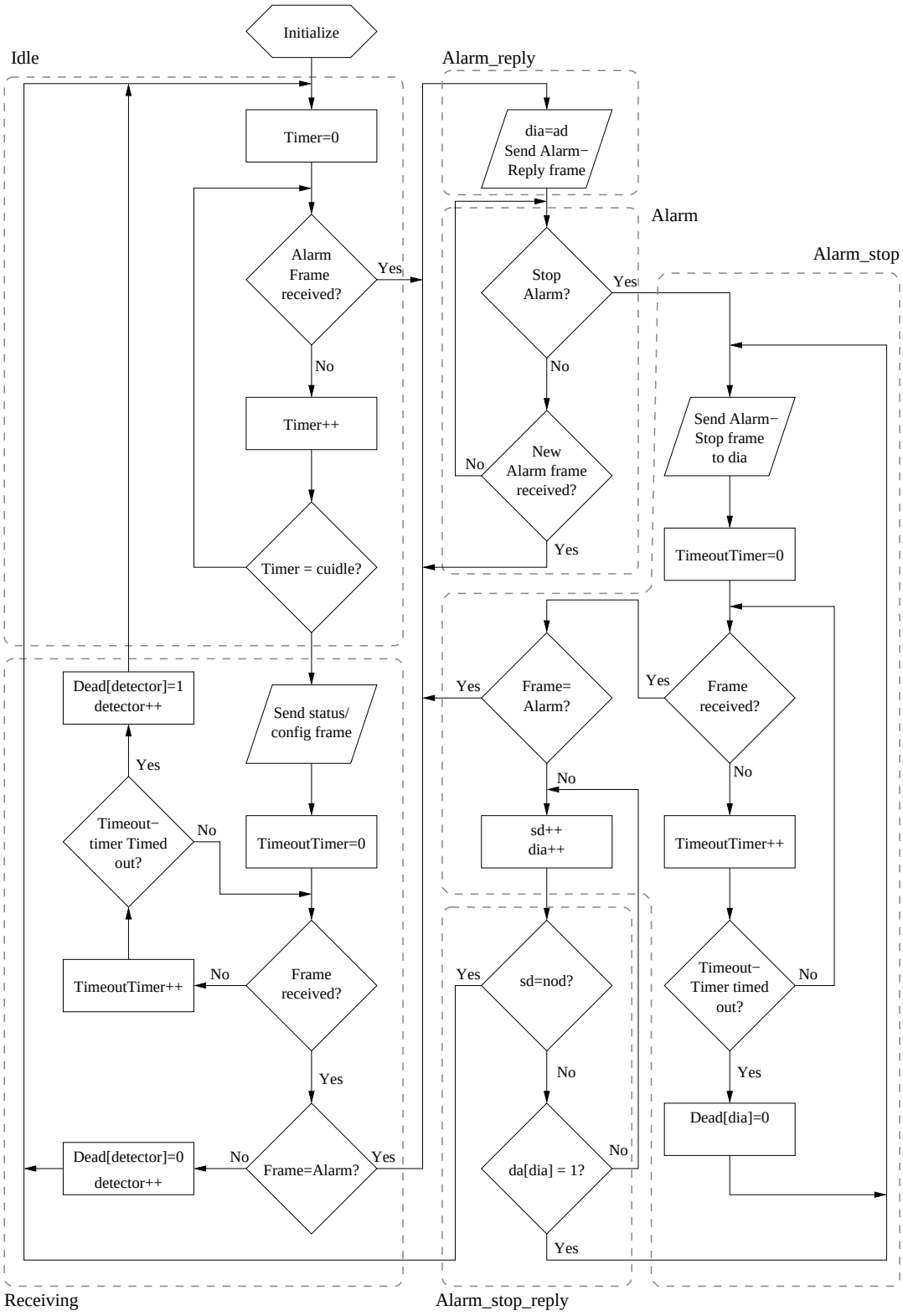

Figure 5.19: *The central unit flowchart made from the* UPPAAL *model on figure 5.13.*

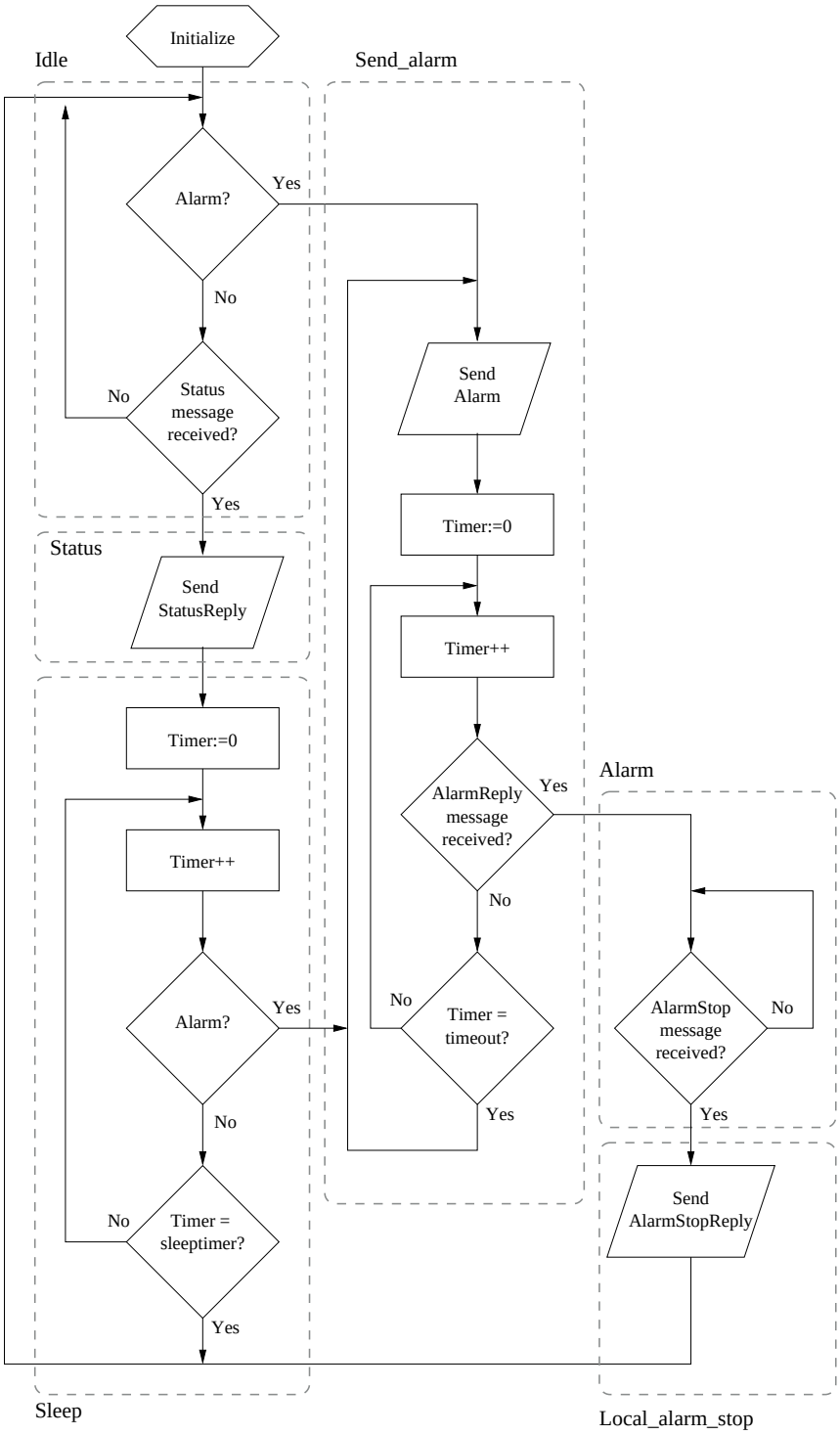

Figure 5.20: *The detector flowchart made from the* UPPAAL *model on figure 5.18.*

User Interface Design

CHAPTER 6

This chapter describes the design of the user interface used to control and monitor the operation of the fire alarm, as well as alerting people when a fire alarm is detected. The inputs and outputs are outlined, and the requirements from standards and recommendations are described. Based on a conceptual model and some interface metaphors, the control panel and the display windows are designed.

6.1 Inputs and Outputs

The user interface needed to operate the WFA contains a number of inputs and outputs. The outputs are used to provide the users with information from the system. Most of this information is displayed on a graphical display, but light emitting indicators are also used to indicate which functional condition the alarm system is in. Additionally audible indicators are used to alert the users of errors or alarms. The input is a touch sensitive layer on top of the display, so that the provided information can be navigated by touching the display with a finger.

This leads to the following list of inputs and outputs:

- Inputs:
  - Touch sensitive layer on top of the display.
- Outputs:
  - Graphical Display (GD).
  - Light Emitting Indicators (LEI).
  - Audible Indicators (AI).

The DS/EN 54 Standard [3] and Precept 232 [4] contains a number of formal requirements to the control and indicating equipment used in fire alarms. These requirements are described in the following section.

6.2 Formal Requirements

The formal requirements to the indicating equipment clearly states details on the use and type of indications. Details of which colors to use for what type of indications, and which type of indications that must be shown are clarified by the standards. It is also outlined how the accessibility of the system must be controlled, and which events must lead to an indication on either display, LEI or AI.

6.2.1 Access Levels

Four levels of access to the fire alarm must exist. Level 1 is the most accessible level and level 4 is the least accessible. The use of each access level is listed in table 6.1.

Each access level higher than level 1 must have its own access procedure. This could be entering a password or inserting a key or a card.

Access level:	Users:
1	Normal users
2	Safety personnel
3	System technician or service mechanic
4	System engineer or developer

Table 6.1: *Access levels and their users.*

6.2.2 Mandatory indications

The control panel of the central unit must indicate each of the four functional conditions, as well as quiescent or normal operation:

- Quiescent Condition (QC).
- Fire Alarm Condition (FAC).
- Fault Warning Condition (FWC).
- Disabled Condition (DC).
- Test Condition (TC).

Each condition has its own display window, and it is possible to switch between them. The required indications in each condition are described further in the following.

Quiescent Condition

When the WFA is in QC the display may show what ever information the designer may want. The only requirement is that it must not be possible to confuse the QC indications with any of the other condition indications.

Fire Alarm Condition

When the WFA is in FAC, a LEI must be flashing as a general indicator showing that *some* zone is in FAC. The color of the indicator must be red, and the flashing frequency must be 1 Hz with minimum on/off periods of 0.25 s. Specific information on which zone(s) is in FAC must be shown on the GD. When the WFA enters FAC the display window showing alarms must be raised. The display must only show information related to the present alarm(s), but display windows of other conditions can be shown by navigating buttons on the control panel.

If another display window is selected the control panel must return to the FAC display window after 15-30 s of inactivity.

An audible indication is also mandatory when the WFA is in FAC. The audio level must be 60 dB (A) for alarm situations. The audible indication can be silenced manually in access level 1 and 2.

Fault Warning Condition

The FWC indication must also include a LEI as a general indicator showing that *some* zone is in FWC. This indicator must be yellow and flash with a frequency of 0.2 Hz with minimum on/off periods of 0.25 s. The details on which zone it is, and what may have caused the error can be shown on the GD, but is not a required indication.

The audible indication may be the same as in FAC, but in case the two conditions appear simultaneously the FAC must have the highest priority. The audio level must be 50 dB (A) for fault warning situations. The audible indication can be silenced manually in access level 1 and 2.

Disabled Condition

A general indicator must indicate that *some* zone is in DC. This indicator must be a yellow LEI. The detailed information on which zone is disabled may be shown on the GD. The indication must be shown within 2 s after the disabling has been performed.

No audible indication is used to indicate this condition.

Test Condition

A yellow LEI must indicate that *some* zone is in TC. The GD must show which zone(s) it is. No audible indication is used, when a zone is in TC.

6.2.3 Additional Indications

Beside indications for the five functional conditions described in the previous sections, a number of other indications are necessary.

A green LEI must indicate that a proper power supply is connected, and that the WFA is turned on. If the main power supply is lost, the central unit must run an audible indication for at least one hour, to warn people that the WFA does not function properly.

As an optional information, the DS/EN 54 Standard mentions statistical information such as the number of alarms and errors that have occurred. As a general requirement, the standards dictate that no additional information shown on the GD may confuse the user, or conflict with more important information, such as alarms or errors.

The indications of alarms or errors may under no circumstances be falsified by multiple reception of alarm- or error signals from devices in the WFA system.

6.3 Control Panel Design

The physical items used to construct the control panel is a graphical LCD Display of size 320 by 240 pixels. The display must have touch screen support, so that no buttons are needed on the control panel. Along the sides of the display, light emitting diodes are used as general indicators (LEI's) of the functional conditions described in section 6.2.2. The control panel is shown on figure 6.1.

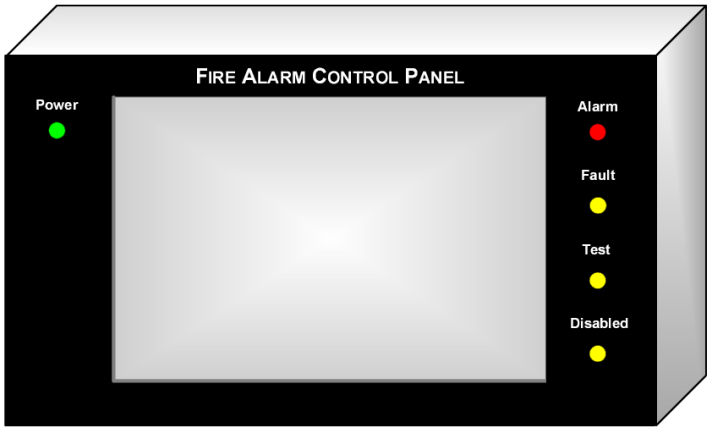

Figure 6.1: *The control panel placed on the central unit.*

6.3.1 Conceptual Model

The conceptual model is a description of the system, in terms of a set of ideas of what it should do, behave and look like. The model is developed by considering the most common types of activities that users are likely to be doing, when interacting with the system. [16] For this particular system these activities are:

1. Exploring or browsing information.
2. Confirming or responding to indications.
3. Navigating and manipulating settings.

These activities are described further in the following.

Exploring or browsing information

During this activity the user is browsing information on the system status. This mostly occurs when the system is in quiescent condition, and a user wants to see how the system has behaved the last period of time. But the activity is also used during faults or alarms, when information provided on a display is scanned by the user - or when browsing the menus prior to doing some configuration of the system, like for instance putting a zone in test mode.

Confirming or responding to indications

The second activity, *Confirming or responding to indications*, occurs when the system state changes from quiescent, or some other non-critical state, to a more severe state that requires the attention of the user. This would be the case when going from quiescent to alarm condition. In this case the user needs to take action, and perhaps evacuate the building. If the change of state were from quiescent to fault warning condition, the user action would be to recognize the fault or error and then call someone to repair the system. In both cases the user could respond by pressing a button, so that the audible indication is silenced, while the visual indication remains active.

Navigating and manipulating settings

The possibility of changing some configuration of the system is done using the third activity, *Navigating and manipulating settings*. This activity is used when some setting needs to be changed, e.g. when a user wants to disable a zone during a repair. In this case the user needs to navigate the menus to find the setting in question, and then manipulate the value of the setting by the press of a button.

In terms of access levels, the first activity is mainly used in access level one and two, where activity two is primarily level two. Activity three is not allowed in access level one.

6.3.2 Interface Metaphors

To support the conceptual model, some metaphors are used to link the visualization of the activities in the conceptual models to some physical objects that the user is familiar with.

The primary metaphor used is to let the display look like index cards know from card files. Each of the functional condition indications have its own display window represented by a virtual index card. The display window is then changed by navigating to the next window, or index card, by operating the sheet markers on top of the index cards.

Another metaphor used is to represent each zone by a graphical representation of the zones ground plane. This way the user can relate information on a zones state to the physical rooms and walls.

6.3.3 Display Design

The display design that corresponds to the five functional conditions described in section 6.2.2 are shown in the following subsections. Except for a configuration display, all displays are shown as they appear in access level one.

Disabled and Quiescent Condition

The QC display is shown when the WFA is working properly, with no alarms, no errors and no tests. This display is shown on figure 6.2.

The left side of the display shows a map of all the zones connected to the WFA system. Each zone has a "LED" indicates the status of the zone. The right part of the display contains

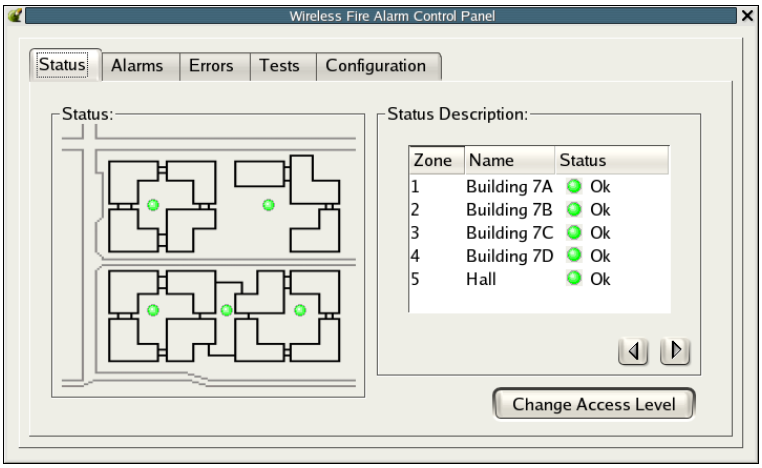

Figure 6.2: *The display when the unit is in quiescent condition.*

a textual description of the system status. If this description is more than 6 lines it can be scrolled using the arrow keys below the description. The top of the display contains the index card metaphors as described in section 6.3.2. The QC display is also used to display zones being in DC. In this case the "LED" is yellow and the textual description states "Disabled".

Alarm Condition

When an alarm is generated the display changes to AC mode, meaning that the alarm "index card" is risen. This display is shown in figure 6.3, where an alarm has been detected in zone 3.

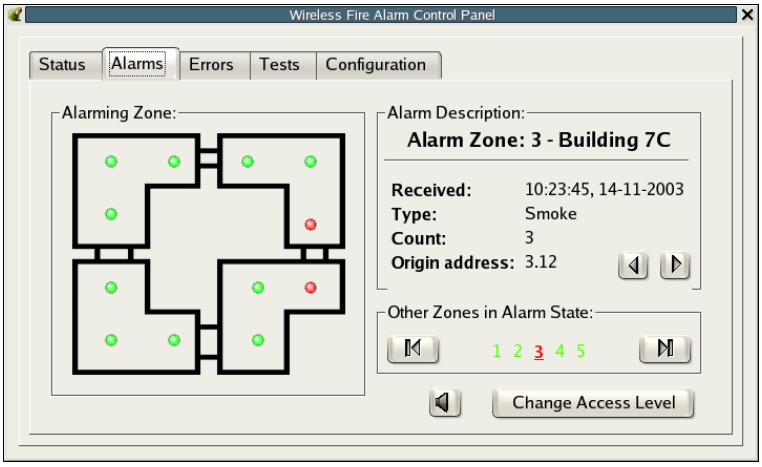

Figure 6.3: *The display when an alarm has occurred in zone 3.*

Now the left side of the display shows a map of the alarming zone. Each detector that has reported an alarm is colored red. The box in the upper right side of the display shows a description of the alarm. The description shows the time of the first reception, what type it is, and the address of the reporting detector, along with the number of alarms reported. The arrow

keys in the bottom of the upper box are used to navigate the alarms reported in that particular zone. Another way of navigating is to press the alarming "LEDs" on the map in the left side.

The box in the bottom right side is used if more than one zone is in alarm condition. Then the arrow keys are used to change between the alarming zones. Then zones in alarm state have a red number, and the zone number currently shown is underlined. Below this box buttons used for changing access level and silencing the audible indication are placed. The audible indication is restarted each time a new zone enters the alarm condition.

Fault Warning Condition

The structure of the FWC display is identical to the AC display. Figure 6.4 shows the FWC display when an error has occurred in zone 3.

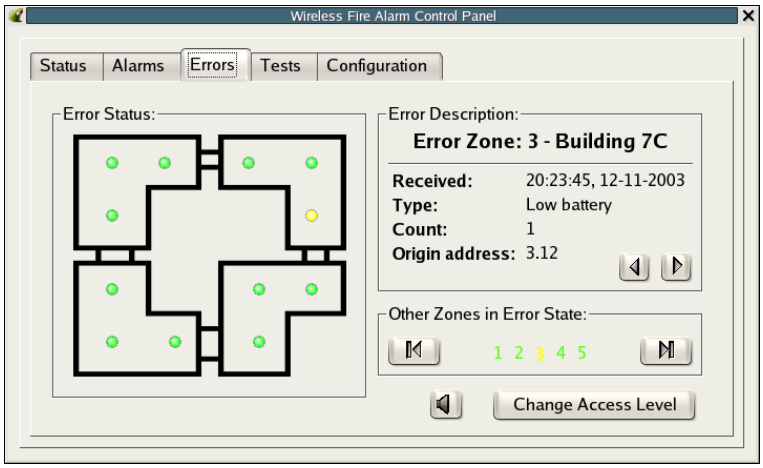

Figure 6.4: *The display when an error has occurred in zone 3.*

Test Condition

When the system is in TC the display structure also follows the structure of AC and FWC. This display is shown in figure 6.5.

One thing to notice is that on the map all detectors are yellow, which is the color used to indicate test condition or errors. The DS/EN 54 Standard states that test mode can only be entered zone-wise, meaning that all detectors in a zone must be in test mode simultaneously. When in test mode, the number of alarms and errors are registered, and displayed on the right half display. At the bottom of the right display side, two buttons used for changing access level and silencing the audible indication are placed. The audible indication is restarted each time a new zone enters the fault warning condition.

Configuration

A few settings can be configured from the control panel when the access level is three or four. To do this, the configuration display is used. This display is shown on figure 6.6. The display makes it possible to enable and disable zones, and to enable or disable test mode for a zone.

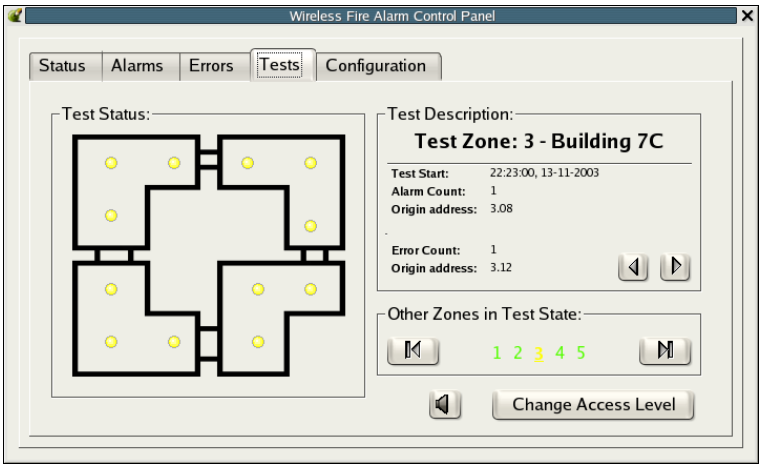

Figure 6.5: *The display when zone 3 is in test mode.*

The number of the zone being configured is entered using a keypad on the right part of the display.

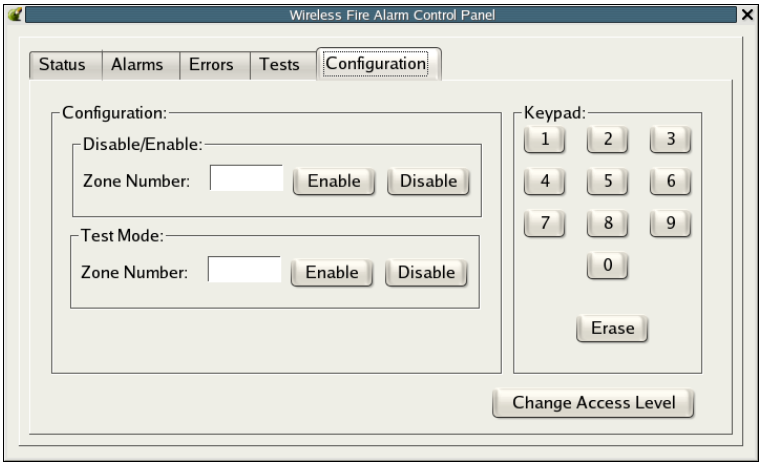

Figure 6.6: *The display when the unit is in configuration mode.*

Part III

Implementation

Part III contains the implementation of the protocol software design in the previous part.
The protocol software is implemented as a prototype for a wireless fire alarm systems consisting of two wireless transceivers. The interfacing with these transceivers affect the possibilities of implementing the full functionality of the designed protocol.

The system is divided into detector and central unit subsystems, and each of these subsystems contains a number of software modules. The dynamic flow of these modules is described, and the communication between the modules is explained.

Implementation

CHAPTER 7

This chapter describes the implementation of the software needed to test the designed protocol. The implementation is a prototype limited to operate one detector, in one zone, controlled by a single gateway and central unit. A number of limitations influence the implemented solution. After a description of these, the procedure used to perform module testing during implementation is described. The implementation of the protocol is explained, and the necessary variables and data structures are defined. The system is divided into two subsystems and the implementation of the processes within these subsystems are described.

7.1 Limitations

The implementation of the protocol and system software designed during this project is limited by a number of factors. Despite having stated the opposite, the company that produced the radio boards used in this project, did not supply the source code of the firmware running the boards. If this code had been available, the software used for the detector and gateway could have been implemented in the microcontroller included on each radio board. However, without the source code it is not possible to develop the software needed to implement the application protocol on top of the radio protocol. This, and a few other incompatibility issues regarding the radio boards, severely limits the possibilities of implementing the protocol software the way it is designed for.

Instead of embedding the protocol software in the radio board microcontroller, a separate microcontroller system based on a Texas Instruments MSP-430 MCU is designed. This system and one of the radio boards constitutes a detector. The second radio board is used as gateway. Since gateway software can not be embedded on this board either, the gateway is considered completely transparent. The forwarding of messages from the CU to the detector is left entirely to the firmware of the radio board.

When transmitting data through the radio boards, no information on whether the transmission succeeded or failed is available. According to the manufacturer of the boards, configuration should be possible by using AT commands on the serial connection. The documentation provided with the modules states that this can be done. However, tests have shown that this is not the case. An enclosed Windows utility is able to read and write a selection of the status- and control registers, but the utility uses some non-documented comport settings and commands to access the registers. This means that it is not possible for the protocol software to perform configuration of the boards, or read status registers within the boards. As a consequence of this, it is not possible to perform a gateway status check as designed in section 5.3.1 on page 44.

Another issue is the number of devices available. Since the number of radio modules is only two, a full scale implementation of a WFA system is not possible. A full scale implementation

would most likely imply some sort of Object Oriented Programming, since the system includes some obvious class candidates like detectors and zones. However, since the implementation is limited to a prototype with only two devices, the necessary amount of code can be reduced by considering a single detector and a single gateway. A system of this size can be implemented using a sequential programming language such as C, since no reuse of code can be achieved by implementing it Object Oriented.

Because the educational purpose of this project is targeted at achieving knowledge within distributed real-time systems and fault tolerant design, the graphical user interface is considered less important. The user interface designed in chapter 6 does not fulfill its potential, with the amount of information available due to the reduced number of devices. Therefore a simplified textual user interface is implemented instead. The purpose of this user interface is to make it possible to demonstrate the systems potential, and to support the testing of the implemented software.

7.2 Implementation Method

The International Electrotechnical Commission has published the DS/EN 61508-3 standard on software requirements within safety related systems [17]. This standard recommends a certain development lifecycle called the V-model. The model is very similar to a software development method called SPU [18].

Figure 7.1 shows an adapted version of the V-model adjusted for use in this project.

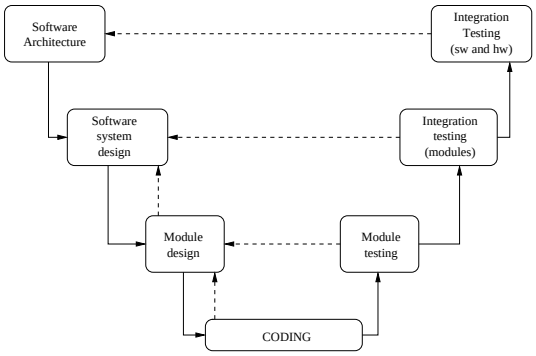

Figure 7.1: *The V-model adjusted to be used in this project. The dashed arrows represent verifications, and the solid arrows represents outputs.*

The original V-model contains one ekstra level on top of those illustrated on figure 7.1. This layer contains a validation test, where a security certificate is obtained by testing the product against a software safety requirements specification. Due to the prototype nature of this project, this layer has been omitted. Furthermore, the standard suggests that the software architecture layer and the software system design layer can be merged in small systems. This has also been done during this project.

7.3 Protocol Implementation

The communication interfaces used to communicate with the radio boards are RS232 connections with a baudrate of 19200, eight data bits and one stop bit. No flow control or control signals are used.

Every message transmitted from one device to another, uses eight bit codes giving a total of 256 possible ASCII signs. Some of these codes have special purposes, and can not be used to transfer data. Such codes are used for control purposes. If they are used in the developed protocol frame it can cause malfunctions.

One way of avoiding the malfunctions would be to use escape characters, meaning that an extra ASCII character is put in front of each character with control purpose. In this project, it is not possible to use this solution, because it will give the frame a variable length.

Therefore it is chosen only to use ASCII codes which can not effect the functionality for transmitting data frames. This means that only ASCII characters with a decimal value of 48 or more may be used.

7.3.1 Protocol Frame

The necessary protocol fields in a protocol frame is described in section 5.3.1 on page 44. These fields include:

- Network ID
- Destination Address
- Source Address
- Operation Code
- Flags
- Status
- Frame End

The complete frame can be seen on figure 7.2.

Figure 7.2: *The complete frame.*

Network ID

The network ID is used to check that the frame is intended for the particular network. The Network ID of the implemented prototype is chosen to be the value `0b01110111`, corresponding to the ASCII character "w".

Destination Address

The destination address is an eight bit field consisting of two subfields, a seven bit address of the destination host, and a single bit flag indicating whether the address is a gateway or a detector. The flag is set to "1" if the host is a gateway. The flag is the most significant bit of the 8 bit field, and the seven bit address are the least significant bits.

Source Address

The source address is constructed similar to the destination address. Again a single bit indicating whether the source host is a gateway or a detector is the most significant bit, and the lowest seven bit are the actual address of the source host.

In this prototype implementation the gateway address is set to `0b1000110` (an F) and the detector is `0b1000101` (an E).

Operation Code

The operation code is used to identify which of the messages described in section 5.1 on page 35 is being sent. The operation codes used in this implementation are:

Name:	Binary Value:	ASCII Char:
Config	`0b01100001`	"a"
ConfigReply	`0b01100010`	"b"
Status	`0b01100011`	"c"
StatusReply	`0b01100100`	"d"
Alarm	`0b01100101`	"e"
AlarmReply	`0b01100110`	"f"
AlarmStop	`0b01100111`	"g"
AlarmStopReply	`0b01101000`	"h"

Flags

A number of flags are used to set options and receive status. These flags are represented as an eight bit value, where each bit is used as a flag. The flags used can be seen on figure 7.3.

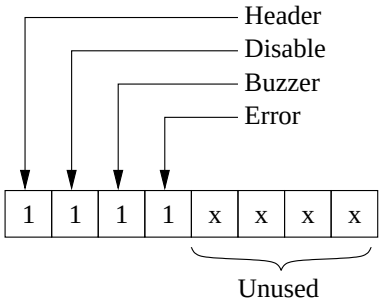

Figure 7.3: *The flags used in the implementation.*

The most significant bit is labeled header. This bit is always set to "1" because otherwise the value of the flags could be NULL, which is an illegal ASCII value.

Status
Status is also an eight bit value used when a value needs to be transferred. In this implementation it is used to transfer and indication of the battery level in the detector.

Frame End
A frame end is used to indicate to the receiver that the frame is complete. In this implementation a newline is used for this purpose.

7.4 Subsystems

The system is divided into subsystems prior to the implementation of the system software. Because the gateway software is included in the central unit, the WFA only contains two subsystems:

1. Detector.
2. Central Unit.

The implementation of each subsystem is divided into modules, in order to determine if each module works as intended prior to testing the complete subsystem software. The two subsystems and their respective modules are described further in the next two sections.

7.5 Detector

Software for the detector subsystem is embedded into the microcontroller MSP430F149. No operation system is required for the microcontroller in this project due to the relative limited extent of the detectors functionality. Interfacing input and output capability for serial communication and smoke detection devices is implemented using the microcontrollers hardware registers.

The software is implemented in C and compiled with C-SPY using IAR C-SPY processor descriptor for MSP430. The detector software is constructed in a Windows environment using the IAR Embedded Workbench IDE.

The implemented software constitute of three parts:

- Serial communication.
- Alarm timer.
- Detector main.

The functionality within these parts are specified from the UPPAAL model. Values in the detector are exchanged as global variables and every detector function is dedicated in separate software routines.

The `serial communication` part is constructed as a driver for the microcontrollers USART. It takes care of receiving and transmitting bytes and stores received data in a circular buffer if the system is occupied by other processes when receiving data.

Detecting fire is carried out by a smoke detector which pulls one pin low on the microcontroller. Once every 600 ms the `alarm timer` part is checking this pin, to investigate whether it is high or low. The routine is enabled by the use of an interrupt vector, which makes the detection routine run like a timed POSIX thread.

The flowchart of the routine is illustrated on figure 7.4. The routine is responsible for checking the smoke sensor, sending alarm frames, checking for received alarm reply and retransmit the alarm if no reply is received. The routine also starts the alarm buzzer. When an alarm is detected and confirmed by the CU, the `alarm` shown on figure 7.4 is set. This is a lock that prevents the detector from transmitting the same alarm more than once. The `alarm` is unlocked when the CU sends an `AlarmStop` frame.

`Detector main` is the part of the detector software which execute each of the detector functions listed in table 7.1.

Function	Description
`InitialConfig()`	Initiate and configure the detector, after receiving the first config frame from CU.
`ConfigReply()`	Replying a config frame by sending relevant detector flags.
`GetFrame()`	Receives and verify a frame form the serial buffer.
`DetectorFault()`	Make indication if a failure occurs.
`SendAlarm()`	Sending an alarm message to detector.
`AlarmStopReply()`	Reply indicates an alarm stop.
`SendStatus()`	Reply to a status check message.
`RxAwake()`	Idle state, detector is ready to receive status and config frames.

Table 7.1: *Functions implemented in* `detector main`.

When the detector unit is reset, it initialises all global variables, reads its detector address, initialises the serial driver and returns to the `InitialConfig` function. As can be seen in the flowchart of figure 7.5, the detector is pending on a config frame from CU before the detector is activated in the wireless fire alarm system.

The idle state of the detector is the `RxAwake` function. `RxAwake` analyses the received frame and depending of the frame type it activates other functions to take care of the frame, which is illustrated on figure 7.5.

Source Code on CD-ROM:
The source code of the detector software can be found in `sourcecode/detector/` on the enclosed CD-ROM.

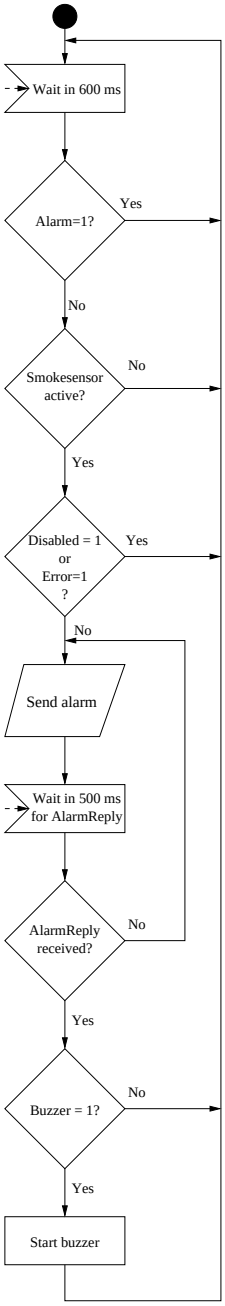

Figure 7.4: *Timer interrupted software routine for fire detection.*

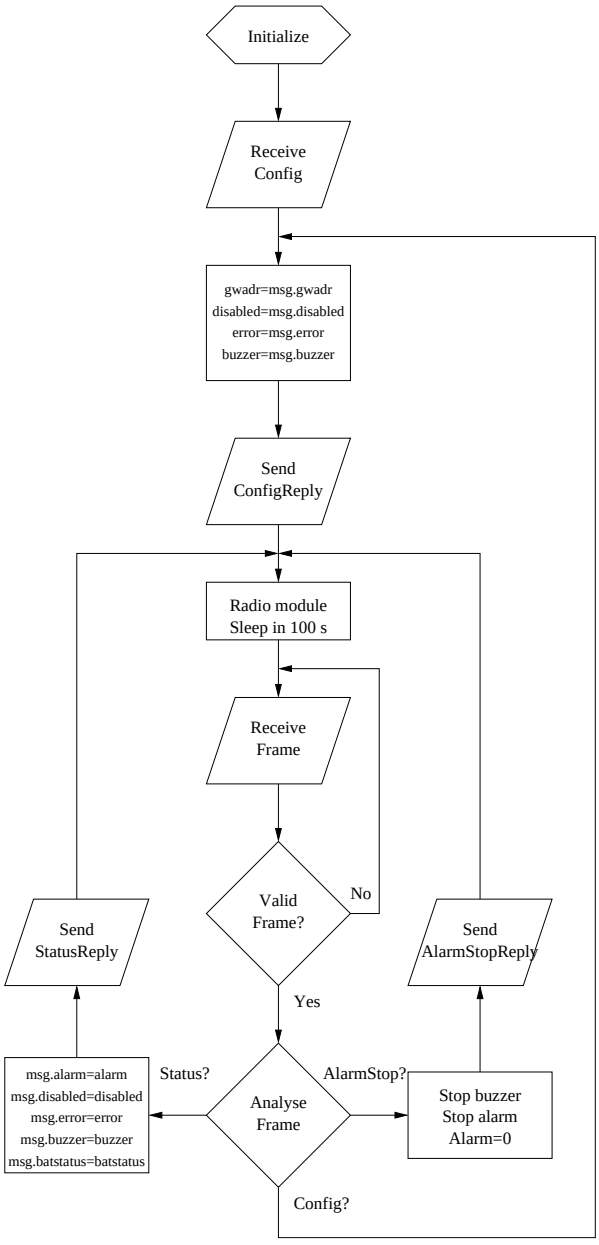

Figure 7.5: *Flowchart of the detectors main routine.*

7.6 Central Unit

The central unit software is implemented in C, and runs on a Linux based PC. The Linux distribution is a Knoppix Linux, which is a Debian based operating system without support for real-time. It is assessed that a real-time operating system is not necessary, to fulfill the requirements dictated in the DS/EN 54 Standard [3] and Precept 232 [4].

The software is compiled using GCC, and requires standard C libraries glibc, as well as POSIX threads. The program relies on an RS232 port properly installed in `/dev/ttyS0` and a PC-speaker available as `/dev/tty0`. Both settings should be standard on every available Linux distribution.

7.6.1 Data Structures

The central unit software uses a number of data structures to maintain the status and states of the detector and zone parameters.

The data structure that regards the detector is shown in the following:

```
90  struct detectorVariables{
91    short int configOk;           /* Initial configuration has been done */
92    short int disabled;           /* Detector is disabled */
93    short int dead;               /* Detector is dead - didn't send a reply */
94    short int alarm;              /* Detector is in alarm condition */
95    short int configReply;        /* A configReply is received */
96    short int statusReply;        /* A statusReply is received */
97    short int alarmReply;         /* An alarmReply is sent */
98    short int alarmStopReply;     /* An alarmStopReply was received */
99    short int config;             /* Config must be sent instead of
100                                    status */
101   short int zone;               /* The zone number to which the
102                                    detector belongs (corresponds to gw number) */
103   short int det;                /* The detector number */
104   short int errorreset;         /* User has requested to reset an
105                                    error */
106   short int buzzer;             /* Detector may use its buzzer */
107   int batstatus;                /* The battery level of the detector */
108   short int error;              /* Error in detector */
109   short int call911;            /* The CU may alert the fire
110                                    department */
111 }det = {0};
```

Except for the `batstatus`, `det`, and `zone` these struct elements are all flags that can be set (value 1) or unset (value 0). Details on when these flags are used is provided during the dynamical description of the Central Unit software, in section 7.6.2.

The data structure containing the zone parameters is shown in the following:

```
113 struct zoneVariables{
114   short int nr;                 /* The zone number */
115   short int disable;            /* Start disabling all detectors in
116                                    the zone */
117   short int alarm;              /* The zone is in alarm condition */
118   short int stopAlarm;          /* The user has requested to stop an
119                                    alarm */
120   short int test;               /* The zone is in test condition */
121   short int dead;               /* The zone is in dead condition */
122   short int error;              /* Error in zone */
123 }zone = {0};
```

The first variable is the zone number, and the next is a flag indicating that the user has requested that the zone is disabled. The rest of the variables are all flags that indicate whether a condition is active or not.

Both the zone- and the detector structure are critical regions being accessed by a number of threads and functions. Therefore means for protecting the contents of these regions must be used. For this purpose Mutex locks are used. A Mutex is a flag that is set, when a thread gains access to a critical region. While that thread has the Mutex lock, no other thread can obtain it. When the thread is finished using the critical region, it releases the Mutex again. This way only one process at the time can access a critical region.

Source Code on CD-ROM:
The data structures are defined in the file `defines.h`, which can be found in `sourcecode/cu/` on the enclosed CD-ROM.

7.6.2 Dynamical Description

Based on the protocol design in section 5.1 on page 35 and the UPPAAL model of the communication system described in section 5.5 on page 49, a dynamical description of the program flow can be outlined. It is very important to maintain a strict relation to the design verified by UPPAAL, so that errors and possible live- or deadlocks are not introduced.

The central unit contains six threads each responsible of a separate part of the functionality.

Two threads are used to maintain housekeeping information by receiving inputs from the user, through a keyboard, and updating the user interface on a display.

Since the reception of messages on the serial port must always be possible, a separate thread is dedicated to this task. By doing this, it is avoided that the reception of data can be a blocking point in the program flow. The possibility of two other threads reading from the serial port simultaneously is also eliminated.

Sending and receiving status- and configuration messages takes place according to a fixed schedule, as described in section 5.2 on page 39. Because the schedule is fixed, status- and configuration messages can be handled by a single thread that maintains the fixed schedule detector communication.

The reception of `Alarm` messages and sending of `AlarmStop` messages does not happen in a fixed schedule. Therefor these activities require separate threads, so that they can be serviced independent of the other threads in the system.

In total, this gives the threads listed below:

- `TuiThread.`
- `KbdThread.`
- `ReadComportThread.`
- `DetectorThread.`
- `AlarmThread.`
- `AlarmStopThread.`

When the central unit software is started, the main loop creates the threads. Two threads, `TuiThread` and `KbdThread`, are started immediately, whereas the other threads must be signalled, before they start. `TuiThread` sends start signals to `ReadComportThread` and `DetectorThread`. `AlarmThread` is started when an `Alarm` message is received by the `ReadComportThread`. `AlarmStopThread` is started when a user request to stop an alarm through the `KbdThread`.

The inter-process communication used to transfer signals between the threads is based on the POSIX signals and conditions available in libpthread.

The purpose and process flow of each of the six threads are described in the following.

TuiThread

The purpose of the `TuiThread` is to read values from the detector- and zone data structures and present the values on the display. The thread also checks which functional condition the WFA is in, and display it on the display, as well as starting audible warning if necessary. When started the first time, the thread starts the `ReadComportThread` and `DetectorThread`. The complete process flow of the `TuiThread` is shown in figure 7.6.

An example of the textual user interface is shown on figure 7.7.

Beside the display shown on figure 7.7 a more detailed display can be shown. This display contains technical information on packet loss, time outs and transmission statistics needed by technicians or engineers. This display corresponds to setting the system in access level four. The display is accessed by pressing the hidden key “4”.

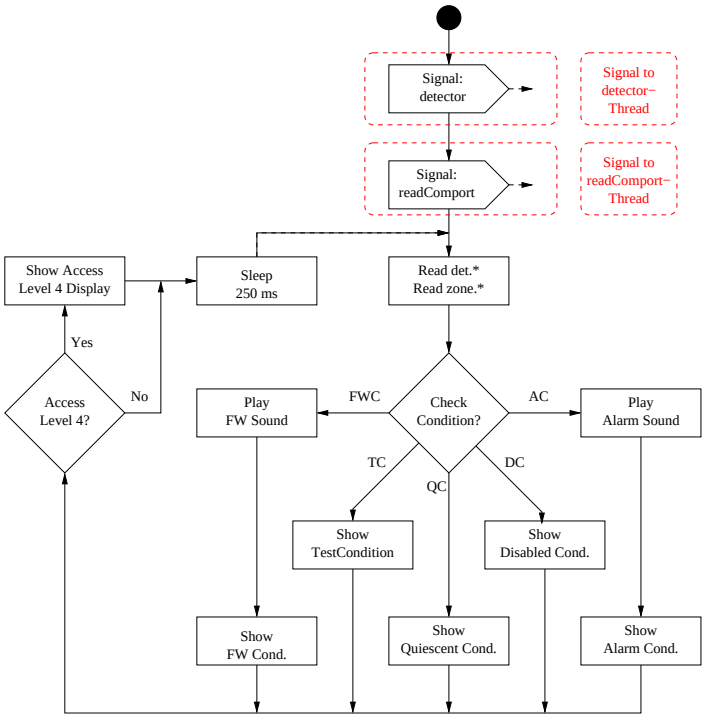

Figure 7.6: *Process flow of the* `TuiThread`.

```
*******************************************************
*          Wireless Fire Alarm Control Panel          *
*      12.34.42 on Wednesday, 17. December 2003       *
*******************************************************
*                 Quiescent Condition                 *
*******************************************************
*      Zone Status:       *     Detector Status:      *
*******************************************************
*                         *                           *
*  Zone Number:   1       *  Det. adr:       1000101  *
*  Zone Alarm:    0       *  Det. Alarm:           0  *
*  Zone Test:     0       *  Det. Bat.status:     80  *
*  Zone Disabled: 0       *  Det. Disabled:        0  *
*                         *  Det. Error:           0  *
*                         *  Det. Dead:            0  *
*                         *  Det. Buzzer:          1  *
*                         *                           *
*******************************************************
*                   Select Option:                    *
*******************************************************
*                                                     *
*  Stop Alarm:    (s)          Test Mode:     (t)     *
*  Error Reset:   (r)          Disable:       (d)     *
*                                                     *
*******************************************************
```

Figure 7.7: *The Textual User Interface during quiescent condition.*

KbdThread

The KbdThread scans the keyboard and responds according to the input received. The possible inputs, and the corresponding actions are:

Char:	Action:
"s"	Stop current alarm.
"r"	Reset current error.
"d"	Disable/Enable zone.
"t"	Enable/Disable test condition for a zone.
"q"	Quit program.

As described in the previous section, an extra option is present. By pressing "4" the advanced engineering display is shown. The process flow of the KbdThread is shown in figure 7.8.

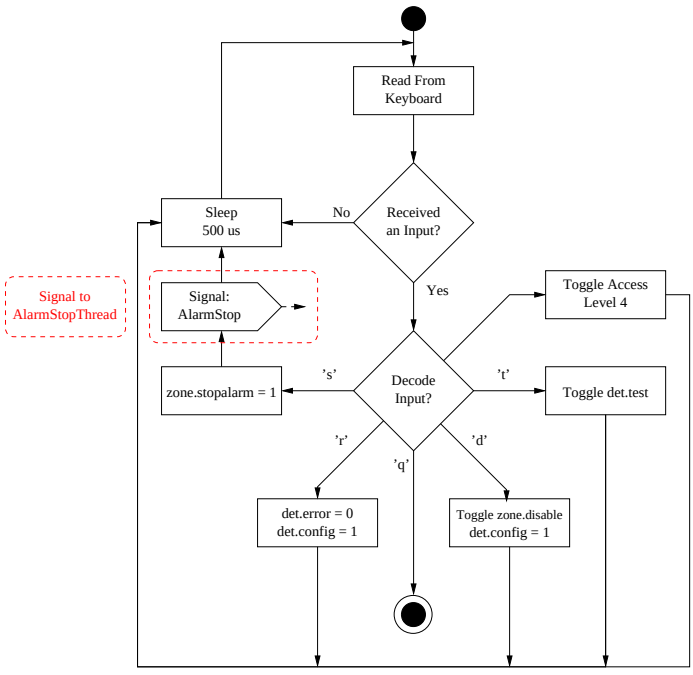

Figure 7.8: *Process flow of the KbdThread.*

ReadComportThread

This thread reads data from the serial port every 50 μs and decodes each valid frame. Depending on the Operation Code, relevant parameters are copied to the data structures, and signals are sent to the threads responsible of handling each message type.

The process flow is shown on figure 7.9.

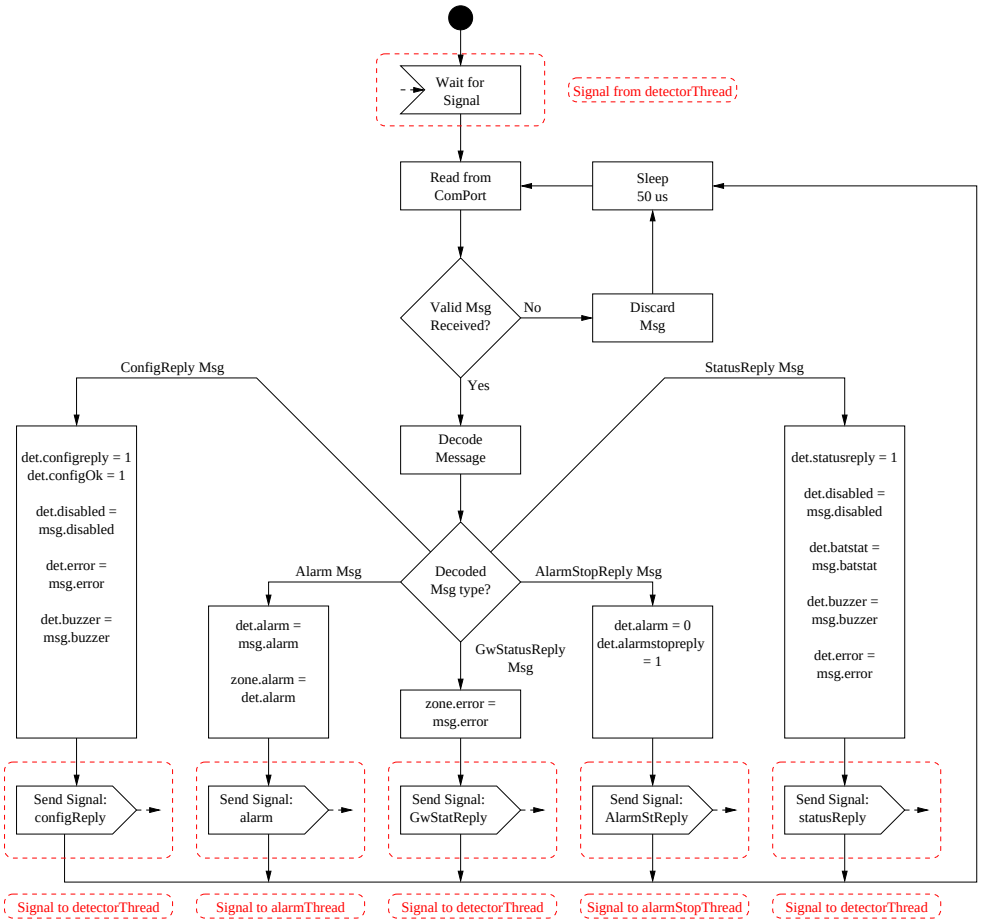

Figure 7.9: *Process flow of the* `ReadComportThread`.

DetectorThread

The `DetectorThread` controls all fixed schedule communication. The process flow is shown on figure 7.10. The central issue is whether the thread should send a configuration- or a status message. This is determined from the `det.config` variable in the detector data structure. If this parameter is "1" a configuration message is sent, otherwise a status message is sent. The first check shown as the top most diamond of figure 7.10 does not play an actual role in this prototype implementation, because a `GwStatusCheck` can not be performed, as explained in section 7.1 on page 69.

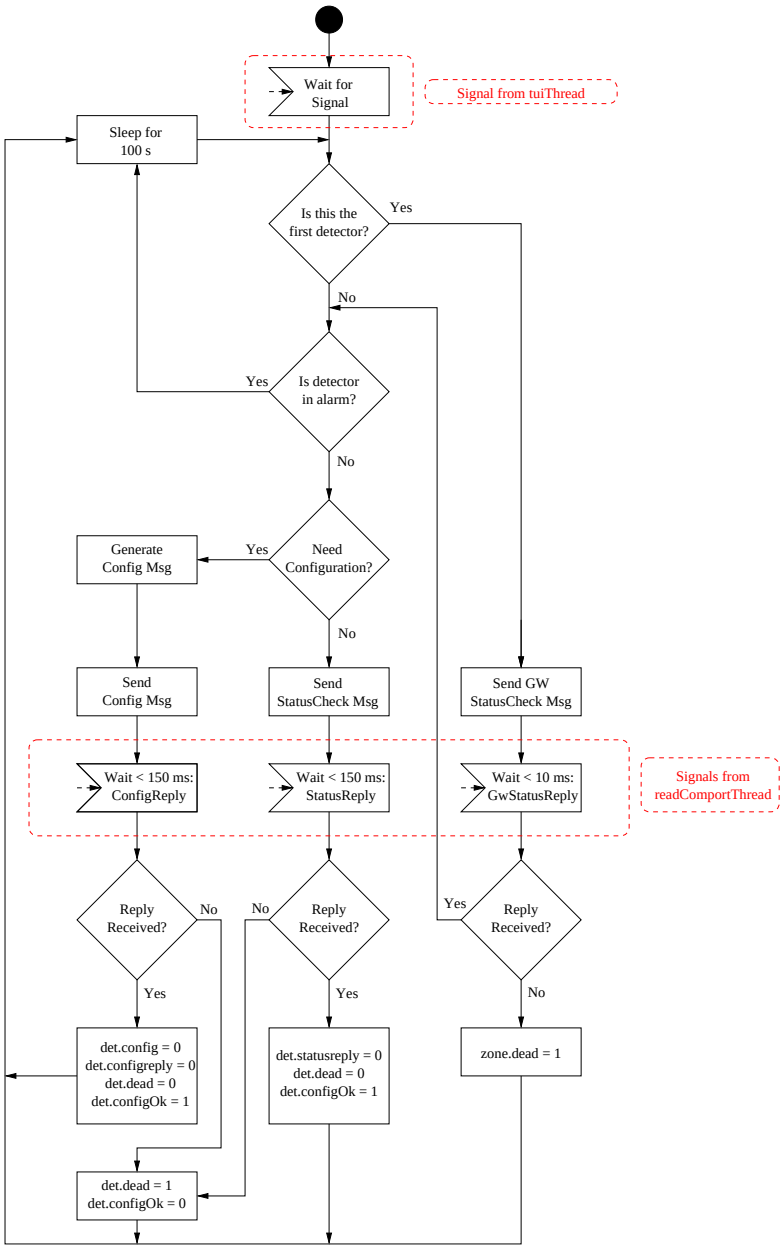

Figure 7.10: *Process flow of the* `DetectorThread`.

AlarmThread

The `AlarmThread` is signalled by the `ReadComportThread` when a message with the alarm Operation Code is received. When this happens, the `AlarmThread` performs the processes illustrated in figure 7.11. If the detector that sent the alarm is not already in alarm state, an alarm reply message is sent to the alarming detector.

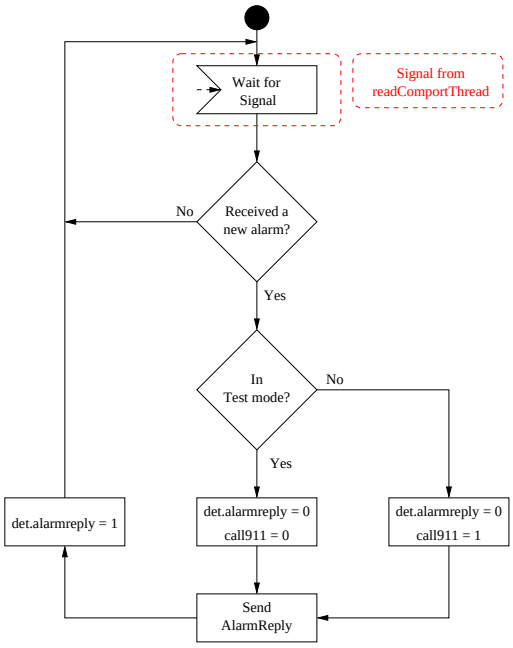

Figure 7.11: *Process flow of the `AlarmThread`.*

AlarmStopThread

The `AlarmStopThread` is trigged by a user input to the `KbdThread`. When a user requests to stop an alarm, the `AlarmStopThread` generates an alarm stop message and sends it to the detector. After having sent the message, the thread waits for `ReadComportThread` to signal that an `AlarmStopReply` has been received. If this reply is not received within 150 ms the detector is registered as dead. The complete flow of the `AlarmStopThread` is shown on figure 7.12.

Source Code on CD-ROM:
The threads are implemented in the file `main.c` and `serial.c`, which can be found in `sourcecode/cu/` on the enclosed CD-ROM.

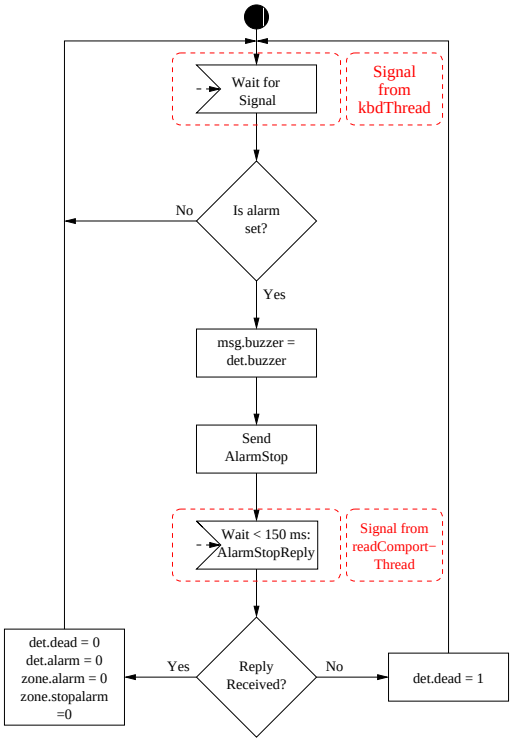

Figure 7.12: *Process flow of the* `AlarmStopThread`.

Part IV

Test & Conclusion

Part IV contains the tests performed on the implemented system and the conclusion. The system is tested up against requirements from standards and prescepts and the problem definition in section 1.5 on page 7.

Based on the test results and the knowledge gained throughout this project period a conclusion is written. The conclusions summarizes and comments on the obtained results and put the project into perspective. As a final remark future improvements or research within the projects area is suggested.

System Tests

CHAPTER 8

The purpose of this chapter is to validate the system through a series of tests. The WFA prototype is evaluated against definitions stated in the DS/EN 61508-3 Standard [17]. The results of the tests are used as a quality indicator of the protocol developed in this project. The first part of this chapter, describes functionality tests for the system. The second part describes performance tests. This includes statistical treatment and calculation of how many test there has to be carried out to obtain a given level of significance. Finally, tests of range and power consumption are described along with a discussion of the test results.

8.1 Functionality Test

Different functions of the system is tested to determine whether the functionality is present or absent. An acceptance test evaluates whether the specified requirements from the problem definition in section 1.5 on page 7 are met.

8.1.1 Standardised Tests

The objective of the standardised tests is to evaluate the operation of the equipment. The test result has to comply with the specifications given in the DS/EN 54 Standard. The following topics are an extract of the test which has to be carried out before a fire alarm system can be approved.

Fire Alarm Condition: (1)
a) Initiate and reset a fire alarm from a zone.
b) Check indications are correct and outputs on CU and detector are correct.

Fault Warning Condition: (2)
- Initiate and reset fault warnings corresponding at least to:
a) Loss of one of the power sources.
b) Short circuit in a detection unit.
c) Interruption in a detection unit.
d) Interruption in a transmission path to GW and CU.

Disabled Condition: (3)
a) Disable and restore one zone.
b) Disable and restore one transmission path between CU and GW or GW and detector.

All of the above tests have to be carried out using one device, afterwards two devices have to be connected, there after three, etc. In this project only one specimen can be carried out, due to the

limited number of devices. The devices have to pass every test at different temperature, relative humidity, and air pressure. But testing the device at these different environmental conditions are considered irrelevant for this project.

Table 8.1 shows the results of the tests specified above.

Test type	Result	Remark
1.a	✓	CU receives an Alarm message when initiating an alarm.
		CU can reset the alarm, using the user interface.
1.b	✓	Detector indicating alarm sending until alarm reply.
		User interface shows an alarm is received.
		Buzzer can be turned on at the detector.
		Alarm signal is given at CU.
2.a	✓/÷	Detector and GW may lose the power source.
		CU can not resist lose of power.
2.b	!	Not tested.
2.c	✓	Yes.
2.d	✓	Yes.
3.a	✓	Indication on the user interface.
		Smoke is not detected on the detector.
		Restoring brings the system back to normal condition.
3.b	÷	System returns to fault warning condition.

Table 8.1: *Extract of the DS/EN 54 Standard test specifications.*

8.1.2 Acceptance Test

The purpose of the acceptance test is to ensure that the wireless fire alarm system is designed and implemented according to the specifications given in the problem definition, section 1.5.

From the user interface it is possible to get an overview of the system performance. Setting a zone in test mode and disabling a zone is possible, as well as stopping an alarm. If an error occurs in a detector, it is possible to reset the error. All these functionalities respond correctly when the user interacts.

When a sufficient amount of smoke is present near the detector, the detector starts transmitting alarm messages to CU. This is indicated on the user interface. Quickly after, both the buzzer on the detector and the CU are signalling fire, by the use of sound, unless the zone is disabled or in test mode.

The alarm can be called of at the CU, afterwards the detector is put back into normal condition. By using visual light emitting indications, it is possible to view the condition of the detector.

Investigation of how the fire alarm system has behaved in the past can be accessed from a log file located at the CU. Events such as alarming, configuration, errors, and system start are quoted with a time stamp in the log file.

The detector is wireless and equipped with batteries, which causes unlimited possibilities of placement of the detector. Installations can be done places where drilling holes in walls or

ceiling for cables is unwanted or even impossible. Such an installation can be carried out within an indoor environment for a range up to 20 meters from a gateway.

The requirements given in the problem definition are compared with the developed hardware and software prototype in table 8.2. Both the specified requirements, the design phase **(D)**, UPPAAL simulation **(S)**, prototype implementation **(I)**, and the succeeded test results **(T)** are summarised in the below table.

Requirement	D	S	I	T
Enter FAC max. 3 s after a sensor has detected fire.	✓	✓	✓	✓
Be able to receive signals from all zones.	✓	÷	-	-
A signal from one zone must not falsify reception from another zone.	✓	÷	-	-
Detect lost connection to a device within 100 s.	✓	✓	✓	✓
Provide means for transfer of other error messages.	✓	÷	✓	✓
Each zone can be disabled and re-enabled independently.	✓	÷	-	-
Each zone can be tested individually.	✓	÷	-	✓
Test condition can only be entered or cancelled by manual operation.	✓	÷	✓	✓
Zones in test condition must not prevent other zones from operating normal.	✓	÷	✓	-

Table 8.2: *Comparing the gained results to the specified requirements. (✓) means consistent, (-) means not possible to determine, (÷) means not implemented or wrong.*

The simulation is focused on ensuring that timing requirements are met, and that no live- or deadlocks occur. The remaining functionality is not included in the simulation model to reduce the computation time needed by UPPAAL. Whether requirements that require more than one detector and one gateway/zone are met, can not be determined since it is not possible to test them. Therefore the results of table 8.2 is satisfactory and expected.

8.2 Performance Test

To document the operation of the WFA a number of measurements are carried out. The tests are performed with radio modules operating in modes for retransmission (transparent secure mode) and without retransmission (transparent mode), respectively. The following tests are made:

- Transmission time.
- Status transmission.
- Alarm transmission.
- Performance test.
- Range.
- Power consumption.

These tests are described in the following sections.

8.2.1 Transmission Time

The purpose of the transmission time test is to measure the time needed to transfer a message over a given media. The test is performed both on wired and wireless link.

To perform transmission tests, additional modification of the CU has to be added. The parallel port of the CU is used to give an output signal when sending and receiving a data frame. Timing parameters can be measured from the signals using a digital oscilloscope. The setup is shown on figure 8.1.

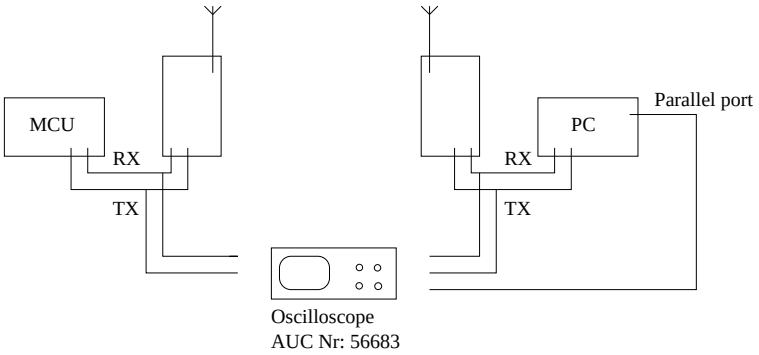

Figure 8.1: *Setup for measuring timing parameters.*

The following series of tests determines the time to transmit a frame from CU to detector and detector to CU. It is given that no other units are using the selected media.

Cable

The transmission speed on the cable is 19200 baud. It is possible to measure the time between the first and the last bit of the frame by using the oscilloscope. The measured transmission time is given in table 8.3.

From	To	Transmission time
CU	Detector	3.6 ms
Detector	CU	3.6 ms

Table 8.3: *Transmission time of one frame over a serial cable.*

Wireless With ZigBee Retransmission

As specified in chapter 4 on page 29, the wireless link should use a transportation method with capability of retransmitting lost frames. It is possible to configure the maximum number of attempts the radio module can use to retransmit a lost frame. Table 8.4 shows measurements of the transmission times using the radio modules, configured at different maximum allowed numbers of retransmissions.

Table 8.4 shows a significant difference between the transmission times. When transferring data frames from CU to detector the times are constant at 40 ms. But transferring from detector to CU with the same frame, the time varies extremely much up to approximately 10 times

Retransmission Possibilities	CU → Detector	Detector → CU
0	40 ms	57 ms
1	40 ms	57 ms
2	40 ms	130 ms
3	40 ms	200 ms
4	40 ms	272 ms
5	40 ms	350 ms

Table 8.4: *Transmission times with different number of retransmissions allowed.*

the transmission time the other way. It is interesting that the transmitting time is increased as more retransmissions are possible. In every transmission the frame is received successfully, any indication of a retransmission can not be determined using the available equipment.

The same test performed with the radio boards swapped showed the same result, so the phenomenon is not related to one specific radio board.

From the results of table 8.4 it can be concluded that transmission times using the transparent secure mode are not reliable.

Wireless Without ZigBee Retransmission
Setting the radio modules in transparent mode, which is without the retransmission possibility the transmission time is much more constant. Doing several test transmissions the timing is constant in both directions at 22.9 ms for transmitting one frame. It is proven that the transmission time of transparent mode is much more reliable than transparent secured mode, hence transparent mode is used in the following tests and for the final implementation.

Propagation Delay
Propagation delay is the processing time inside the detector and the CU. The delays are measured as the time from the unit receives an `Alarm` / `AlarmStop` frame until the time when the unit transmits a reply. The propagation delay includes all necessary processing.

- Propagation delay for CU: 30.8 ms.
- Propagation delay for Detector: 650 μs.

8.2.2 Status Transmission

It is important to measure the time for a transmission of a status check frame. The DS/EN 54 Standard specifies that every detector is checked at least once in 100 seconds. The transfer time of a status check has influence on the number of detectors it is possible to have in the WFA.

From the frame transmission time measurement it is possible to calculate the total transfer time of a status check.

The time is calculated from when the status check frame is constructed by the CU until a status reply frame is received and processed at the CU. The measurement is carried out using transparent mode. The time of a status checking can be seen in table 8.5.

Transmission method	Transmission time [ms]
RS232	3.6
Wireless	22.9
RS232	3.6
Propagation delay, detector	0.65
RS232	3.6
Wireless	22.9
RS232	3.6
Propagation delay, CU	30.8
Total	90.85

Table 8.5: *Total transfer time of a status check.*

As specified in figure 5.12 on page 48, the protocol has to make space for an alarm frame to be transmitted. Table 8.6 gives the timing parameters of sending an alarm frame from detector to CU.

Transmission method	Transmission time [ms]
Propagation delay, detector	0.65
RS232	3.6
Wireless	22.9
RS232	3.6
Propagation delay, CU	30.8
Total	41.75

Table 8.6: *Transfer time of an alarm frame.*

From table 8.5 and table 8.6 the total numbers of detectors in a network can be found using equation 8.1.

$$\frac{100\text{ s}}{\text{Status frame} + \text{alarm frame}} = \frac{100\text{ s}}{90.85\text{ ms} + 41.75\text{ ms}} \approx 750\text{ detectors} \qquad (8.1)$$

To determine whether it is possible to have 750 detectors in a network and still be able to handle alarm transmissions, alarm transmission tests are performed in section 8.2.3.

8.2.3 Alarm Transmission

The main concern of the WFA system is to ensure that every detected fire is reported to the CU within 3 seconds. The alarm transmission test measures the time from when the alarm is generated until the alarm is verified at the CU. While the alarm transmission test is carried out the CU sends status check frames. To simulate a system with more detectors the frequency of how often the CU sends status frames is adjusted. Under normal operation each detector receives one status frame within 100 seconds, but when simulating more detectors, the same detector receives more than one status frame within the 100 seconds.

Two test series are made. The first test is performed as the system is under normal condition with a single detector. The other test stresses the system by simulating more detectors. Each test is performed with 100 samples, to ensure a sufficient statistical basis.

Low Load

The low load is the test where the wireless link only is occupied with one status check each 100 seconds. The histogram on figure 8.2 shows the time of an alarm transmission, performed 100 times.

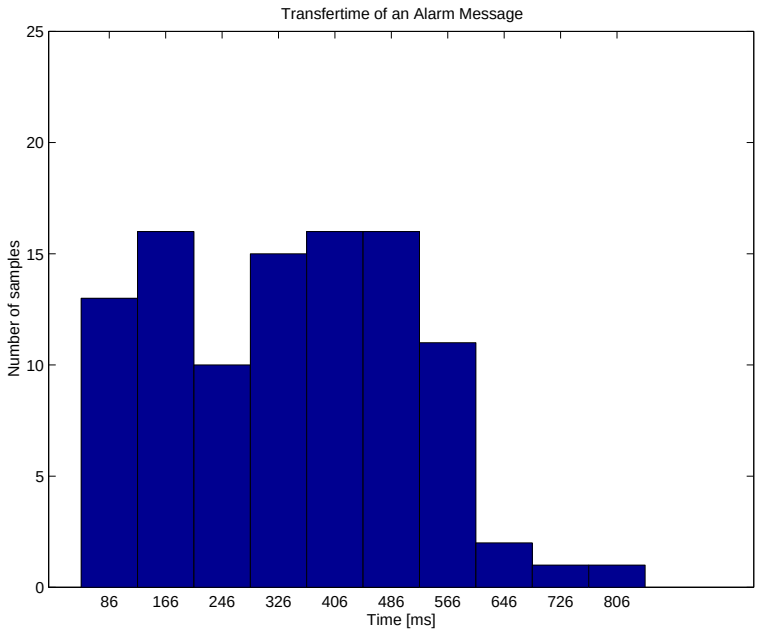

Figure 8.2: *Histogram of measured time for alarm transmissions with status checks every 100 seconds, corresponding to a system with one detector.*

Table 8.7 show the statistical values of the low load alarm transmission test.

Mean $\bar{x}$	Minimum	Maximum	Variance s^2	Sample size
337.8 ms	46 ms	846 ms	29407	100

Table 8.7: *Results of the low load test using wireless link.*

High Load

The high bandwidth load is the test where more detectors are simulated.

Testing the system with 750 detectors, which is a 133 ms delay between each status check, can not be completed using the radio modules. After a short period of time, the CU status check routines times out and alarm messages are not sent from the detector. To inspect where the bottleneck is in the system a test is carried out using the cable connection with 750 detectors.

Table 8.8 and figure 8.3 shows the results of the high load alarm transmission test with cabled connection and simulation of 750 detectors simulated.

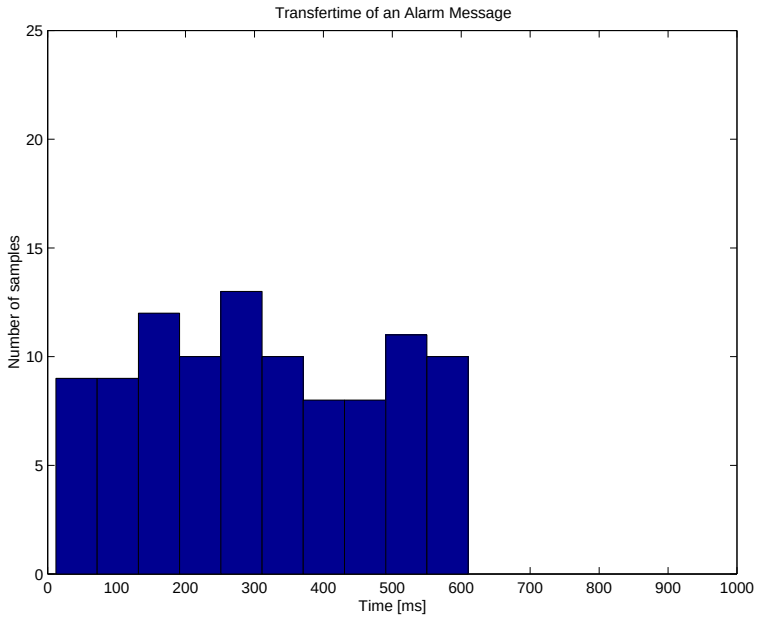

Figure 8.3: *Histogram of measured time for alarm transmissions with status checks every 133* ms*, corresponding to a system with 750 detectors.*

Mean $\bar{x}$	Minimum	Maximum	Variance s^2	Sample size
308.0 ms	12 ms	610 ms	30314	100

Table 8.8: *Results of the high load test, with 750 simulated detectors using a cabled connection.*

The results from table 8.8 shows no signs of delayed alarm transfers compared to the low bandwidth load test. This test indicates congestion in the wireless link.

Reducing the number of detectors in the network to simulate 250 detectors, which is a delay of 400 ms between status checks is also tested. The radio modules are used as links. The results are shown on figure 8.4 and in table 8.9

It is worth noticing that the mean transfer time is higher under higher load when comparing table 8.9 with the result of the low load test in table 8.7. This indicates that a detector has

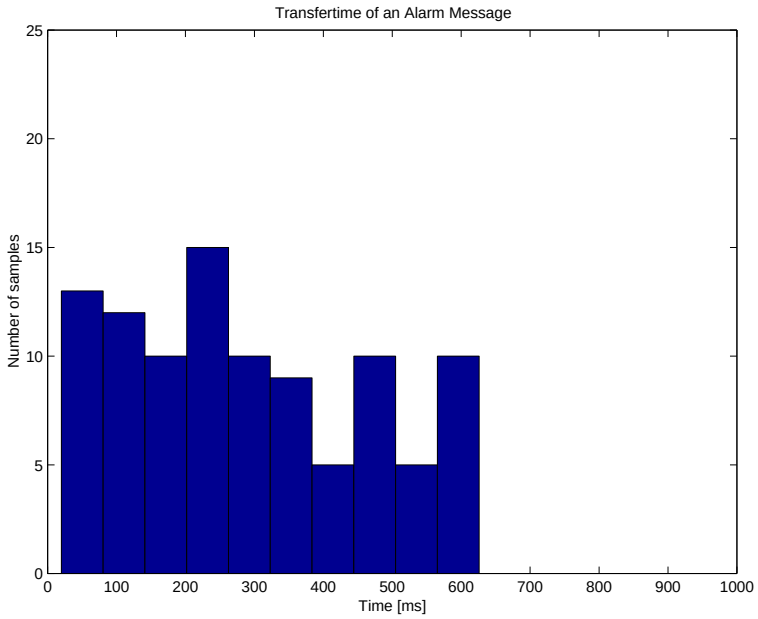

Figure 8.4: *Histogram of measured time for alarm transmissions with status checks every 400* ms*, corresponding to a system with 250 detectors.*

Mean $\bar{x}$	Minimum	Maximum	Variance s^2	Sample size
413.0 ms	62 ms	982 ms	43951	100

Table 8.9: *Statistical values of high bandwidth load, with 250 simulated detectors.*

to wait a longer time before the radio channel is free to transfer an alarm frame when the delay between status checks is low. The variance of the transfer time varies more in table 8.9 than in table 8.7. This is expected because the radio modules transfer protocol waits a random period of time before it checks whether the channel is free or not before transmitting the alarm message. This behavior is described in appendix B.2.

Using 250 detectors in the system gives a satisfying result because the transfer time is still well within the recommendation of 3 seconds given in the DS/EN 54 Standard.

8.2.4 Determining Necessary Test Period

Reliability of a system is defined as the probability that the system will perform properly within a specified period of time [19]. The system has to be error free within a fixed time of a test period. The length of this period can be determined using statistical methods, when knowing the desired level of significance.

Random failures occur for most systems. To model the reliability of a system, it is assumed that system failures can happen independent of the time when the system was started. In other words: failures can happen at any time. This assumption does also apply for the system implemented in this project. In a safety critical system it is important, that the average time between failures is minimised.

Exponential Probability Distribution

A mathematical tool to determine the time before failure is to use the exponential probability distribution. This is possible because the failure rate model of a continuous operating system leads to an exponential distribution. [19]

The exponential probability density function (pdf) is shown in equation 8.2.

$$f(x) = \frac{1}{\mu} e^{-x/\mu} \quad \text{for} \quad x \geq 0, \mu > 0 \tag{8.2}$$

The area under the graph of equation 8.2 corresponds to an interval that provides the probability that the random variable assumes a value in that interval.

To compute the exponential probabilities within a fixed interval of a probability distribution it is desirable to use equation 8.3.

$$P(x \leq x_0) = 1 - e^{-x_0/\mu} \tag{8.3}$$

Equation 8.3 shows the cumulative probabilities of a fixed probability range, where x_0 denotes a specific value of a range. [20]

To comply with the DS/EN 61508-3 Standard, two types of failure rates are given:

Low Demand Mode
An average probability of 10^{-5} failures per hour. These types of failures are only design parameters such as display and indicator failures.

High Demand Mode and Continues Mode
An average probability of 10^{-9} failures per hour. These failures are classified as dangerous types, such as failed fire detection and performance of system operation.

For the above two probability values, it is possible to calculate the minimum testing time for the low demand mode and high demand mode, respectively.

To ensure the stability of the systems functionality in low demand mode, following equation 8.4 gives the testing period.

$$\begin{aligned} P(x \leq x_0) &= 1 - e^{-x_0/\mu} \\ P\left(1 - 10^{-5}\right) &= 1 - e^{-x_0/\mu} \\ -x_0/\mu &= -11.5 \\ x_0 &= 11.5 \cdot \mu \end{aligned} \tag{8.4}$$

Using equation 8.4 guarantee that testing in 11.5 hours with non failures gives a 1 hour certainty of 99.999%.

In high demand mode the testing time is given in equation 8.5.

$$\begin{aligned} P\left(1 - 10^{-9}\right) &= 1 - e^{-x_0/\mu} \\ x_0 &= 20.7 \cdot \mu \end{aligned} \tag{8.5}$$

8.2.5 System Test

A final test of the system is a system test. The system is operating in a normal environment for a long period of time, it is verified that the system is working properly by monitoring how the system is performing.

The length of this test period have been 310 hours during which no signs of errors did occur. Using equation 8.5 a period of 15 hours without errors can be guaranteed with a certainty of 99.9999999%.

8.2.6 Range Test

The transmitting range of the radio modules is specified to 75 meters indoor. Two tests are carried out to measure the range of the detector indoor where walls and doors are blocking the sight.

The first test uses the mode with possibilities of retransmission, while the other test is without.

With ZigBee Retransmission
The range is measured to about 25 meters. At this distance the CU starts to timeout very frequently making it impossible to use the detector at a farther range.

Without ZigBee Retransmission
Measuring the range without the possibility to retransmit, shows a shorter reliable communication range. When the detector passes about 20 meters the received frames are starting to be corrupt. Within 20 meters from the gateway the system is performing stable.

These results are not satisfactory. This is caused by the strange behaviour experienced when allowing the radio boards to retransmit. Packet loss is expected when operating at long range without retransmissions enabled, and since the retransmission mode seems unstable, the result of the range test using retransmissions is not much better than without retransmissions.

However, testing the modules with the built-in "Demo Mode", described in section 4.6 on page 31, a range of approximately 60 meters is obtained. This confirms that the range problem is caused by the retransmission problems, and not a general ZigBee problem.

8.2.7 Power Consumption

By using a special output connector on the radio modules it is possible to measure the power consumption. In this project it is only interesting to know the consumption of the radio unit, because this unit is the only part of the detector prototype which is going to be used in a final product. All processing of the detector is carried out by the microcontroller inside the radio board. The power consumption at a supply voltage of 3.3 V are measured to be:

- Sleep mode = 19 μA.
- Receiving mode = 24 mA.
- Transmitting mode = 28 mA.

When using the microcontroller in the radio modems for processing the fire alarm protocol and detection of smoke, it is expected that the module will obtain a power consumption which is a little higher. Current measurements of the modems internal microcontroller can not be completed.

Using the battery supplied with the radio boards, it is possible to calculate the runtime of the detector, before a replacement of the battery is necessary. The battery has a capacity of 1 Ah. It is expected that the detector only is in receiving / transmitting mode 0.2% of the time. For the rest of the 99.8% time, the radio is in sleep mode. The power consumption is calculated in equation 8.6 and 8.7.

$$\frac{0.2}{100} \cdot 28 \text{ mA} = 56 \ \mu\text{Ah} \tag{8.6}$$

$$\frac{99.8}{100} \cdot 19 \ \mu\text{A} = 18.9 \ \mu\text{Ah} \tag{8.7}$$

By using the results of equation 8.6 and 8.7, the lifetime of the battery is calculated in equation 8.8.

$$\frac{1 \text{ Ah}}{56 \ \mu\text{Ah} + 18.9 \ \mu\text{Ah}} \approx 13350 \text{ hours} \approx 556 \text{ days} \approx 18.5 \text{ months} \tag{8.8}$$

This result is satisfactory.

8.3 Test Conclusion

The results of the tests show that the wireless fire alarm system is working as expected, but some measurements indicate behavior which is not satisfying. The selected radio modem operational mode does not function stable. When allowing the modem to retransmit lost frames the transmission time is not the same in both directions, and the modem seems unstable. This result is clearly not satisfactory.

The functionality tests has proven that the WFA complies to selected standardised tests. Moreover the acceptance test verifies that the required specifications given in the problem definition are met according to the design, simulation, and implementation. However, the limited number of devices makes it impossible to test all of these specifications.

Time measurements show that the measured transmission times are approximately identical to the transmission times calculated during the designed process.

The test simulating 750 detectors using the single implemented detector did not succeed when using the radio boards. When using a serial cable the test succeeded. This does not imply that the WFA is not able to handle 750 detectors, since the stress on the detector and radio modules is sufficiently higher when one detector needs to simulate 750 detectors. Using the radio modules as links between central unit and detector the number of successfully simulated detectors is 250.

Measurements of the power consumption shows a relative low power consumption in sleep and operation mode. This gives a calculated operation time of 18.5 months, which is satisfactory.

The range that the detectors can communicate to a gateway is measured to 20 meters. This result is lower than expected, which is caused by the retransmission problem mentioned earlier.

The system is tested to work without errors for 15 hours with a certainty of 99.9999999%.

Test Data on CD-ROM:
The data files of the test can be found in `test/data/` on the enclosed CD-ROM.

Conclusion

CHAPTER 9

9.1 Summary

Within the overall semester theme of "Distributed Real Time Systems" a robust protocol for a wireless fire alarm is developed.

9.1.1 Analysis

Based on an analysis of wireless communication technologies parameters such as range, power consumption, interference, bandwidth, possible number of nodes, and the advantage of a standardised protocol available are compared, and 868 MHz ZigBee is chosen as the most suitable standard. ZigBee is designed for applications with a low bandwidth need and extremely low power consumption. ZigBee devices features a range of 75-300 meters and a maximum number of 65000 devices in a network.

The physical- and MAC layers of the ZigBee protocol are approved by IEEE in the 802.15.4 specification during fall 2003. The rest of the layers are not fully specified yet and details on the preliminary versions are only available to members of the ZigBee alliance. One of these members, Adcon Telemetry, has produced an evaluation kit with two ZigBee compliant radio modules, power supply's, and software available. These modules are selected as the wireless links used in the wireless fire alarm.

The topology of the wireless fire alarm fulfils the DS/EN 54 Standard: *"Fire detection and fire alarm system"* and Precept 232: *"Automatic Fire Alarm Systems"* from the Danish Institute of Fire and Security Technology. The alarm system is divided into zones each containing a number of detectors and a single gateway. The detectors communicate wireless with the gateways. Each gateway is connected to a central unit through a cabled network. Because the development of the communication protocol is the most important issue, the communication technology used for the cabled network between central unit and gateway is not essential. This network is simulated using an RS232 connection because this is the connection available on the radio modules. The transmission speed is 19200 baud. The central unit is responsible for alerting the fire department and also contains a user interface.

The standards and precepts dictates that the fire alarm must be able to simultaneously be in any combination of the functional conditions alarm, fault warning, quiescent, test, and disabled. It is also defined that an alarm must be received at the central unit within 3 s after it is signalled by a detector. Any error or fault must be indicated on the central unit within 100 s. These requirements are fulfilled by letting the central unit request a status message from each detector and gateway once every 100 s. The central unit then receives a reply with information on any errors that may be present in the device, as well as housekeeping information such as battery status and various device settings.

The protocol is developed with a number of objectives in mind. The primary aspect is to achieve a reliable and robust communication while keeping the power consumption as low as possible. This is achieved by concentrating the intelligence on as few devices as possible, which is accomplished by keeping the gateways completely transparent. By designing the protocol in such a way that the periodical request of status messages is controlled by the central unit, a master/slave relation is achieved. This method minimises the possibility of collisions and queues building up in the communication system. Another advantage is that the radio part of the detectors can be put to sleep for a fixed period of 100 s when the exchange of status messages is finished. By letting the master request the status the slave is automatically synchronized with the master every time a status transfer is finished. This way the possibility of the master- and slave clocks drifting apart is eliminated. Alarm messages are not requested by the central unit. Doing this would imply that the detector could not sleep for more than 3 s, which severely increases the power consumption. Instead an alarm message is sent spontaneous from the detector to the central unit. By designing this way increases the complexity of the model of the system, but it does not compromise the advantages achieved by the master/slave relation, because the transfer of alarm messages is considered as a seldom event. The scheduling of status messages is designed in such a way, that the wireless channel is left idle long enough between each status check for an alarm message to be transfered.

Another objective applied to ensure reliability and robustness is to keep things simple. This principle is expressed in the way a gateway breakdown is handled. If the connection to a gateway is lost the zone is gracefully shut down and put into fault warning condition. An analysis of different methods for re-routing signals so that a zone without gateway connection can continue normal operation concludes that the increasing complexity inflicted to the protocol is undesired. This is conform to the DS/EN 54 Standard, since no requirement states that an alarm system must continue to operate if a defect device is present.

Because the upper layers of the ZigBee protocol stack are not available, the developed protocol is an application layer protocol used on top of the ZigBee radio boards. No feedback information on whether a transmission succeeded of failed is available, and therefore the radio boards are considered completely transparent. The necessary transfer of acknowledge messages is therefore handled by the application layer protocol.

9.1.2 Design

Based on an analysis of the functional conditions required by the DS/EN 54 Standard and the selected topology the protocol design is made. The protocol includes means for performing status check, configuration, sending alarms, and stopping alarms. Each of these services contains a reply message to confirm that the message is received and understood. This reply must be received by the central unit within a given time. The status check is performed every 100 s unless a user has changed a setting interacting the user interface on the central unit. It this is the case a configuration message is sent instead. This means that it can take up to 100 s until a configuration change is put into effect in a detector. This is an acceptable trade off between conserving battery life time and responding immediately to configurations. But since a change of configuration is considered as a seldom event, the battery life time is prioritised higher.

Alarms are transferred immediately and a zone remains in alarm state until the alarm is stopped manually from the central unit. When this happen an alarm stop message is immediately sent to the alarming detectors. While a zone is in alarm condition status checks are not sent, and the detectors do not enter sleep mode. This is done because the alarm condition has the highest priority according to the DS/EN 54 Standard, and because the detectors must not enter sleep mode, which would prevent them from receiving an alarm stop message.

The configuration service makes it possible to disable zones, reset errors, and put a zone in test condition.

All messages consist of a 70 bit frame. The fixed frame length is used because it makes it easier to determine start and stop of frames and to calculate transmission times. The transfer time of a sequence containing one request message and a reply message is calculated to 99 ms. Maintaining a schedule where the channel is left idle for the transfer of an alarm message makes it possible to have 670 detectors in a system. By looking in the standards at the guidelines for placing detectors, it is calculated that the maximum number of detectors in a system is 640. Thereby it can be concluded that this WFA design has no limitations according to the number of detectors. Based on the calculated transfer time the central unit accepts waiting for a reply message in maximum 150 ms.

The protocol is designed and verified using the UPPAAL tool. UPPAAL is a tool for modelling, validating, and verifying real-time systems. The UPPAAL model has verified that the protocol contains no deadlocks, and that an alarm message is able to be transmitted from a detector to central unit within 200 ms, no matter when the alarm occurs. This is verified in a simulation with 670 detectors.

A graphical user interface is designed according to the requirements of the DS/EN 54 Standard. The standard dictates the use of colours, frequencies of flashing indicators, and a mandatory use of separate display windows for each functional condition.

The radio board contains an 8 bit microcontroller and some general purpose I/O which is sufficient to run the software needed for the detector. However, the software can not be flashed into the microcontroller without a software developers kit provided by the vendor. Due to the financial limitations of this project such a kit is not available. Therefore an additional microcontroller system is designed to implement the detector. This system contains a 16 bit microcontroller from Texas Instruments and an RS232 driver used to communicate with the radio board. A ionizating smoke detector is used as sensor input and a piezo buzzer as audible indicator. The detector address is set using dipswitches and the power supply is delivered by a 3.3 V battery.

9.1.3 Implementation

To test the developed protocol a prototype of the wireless fire alarm system is implemented.

Because only two radio modules are available, the implemented prototype only supports one gateway and one detector. Since it is not possible to embed software in the microcontroller on the radio boards the implementation of a separate gateway is not possible. Therefore the gateway- and central unit software is merged. This is not expected to cause problems, because

the radio board supports Hayes or AT modem commands which can be used to perform status check on the gateway and put the radio part in sleep mode. However, it is not possible to force the radio board into Hayes- or AT mode without using a software utility enclosed in the evaluation kit. Measurements have shown that the utility uses some non-documented comport- and flow control settings to change the mode. Therefore it is not possible to implement status checks on the gateway nor put the radio modules in sleep mode.

The central unit is implemented on a PC running Knoppix Linux, which is a Debian based Linux distribution running kernel version 2.4.22. The software is implemented as a multi-threaded program written in C. Due to the number of devices present and the purpose of the project, the graphical user interface is not implemented. Instead a textual user interface is implemented to demonstrate the functionality provided by the protocol.

The detector microcontroller system is programmed in C using a software developers Kit for the Texas Instruments microcontroller.

9.1.4 Test

The radio boards can operate in different modes. The protocol is designed for the "Transparent Secured Mode" which is a mode that automatically retransmit packets up to a defined number of times if the receiver of a packet detects it to be incorrect. However, through testing it became evident that the boards adopt an unstable behavior in this mode. Measurements shows that the transfer time needed to transfer a frame across the wireless link is not the same in both directions. In one direction the transmission time is constant 40 ms while the transmission time in the opposite direction varies from 57 ms to 350 ms when adjusting the maximum allowed number of retransmissions from 0 to 5.

As a consequence of this result, the radio boards are configured to use "Transparent Mode" instead. In this mode the boards do not retransmit lost or erroneous frames. The "Transparent Mode" is used throughout the rest of the tests. The transmission time across the wireless link is then measured to be 22.9 ms in both directions for a single frame.

The complete transmission time of a request- and reply frame from central unit to detector and back again is also measured. Including processing in the detector and central unit this time is measured to be 90.85 ms. This result is satisfactory compared to the 99 ms calculated with two retransmissions, but without data processing at both ends.

The transfer of an alarm message is measured to be 41.75 ms. The result is approximately half the time needed to transfer a complete request- and reply sequence. This is expected, since the alarm acknowledge sent from the central unit back to the detector is not included in the measurement. The reason for this is that the schedule is designed in such a way that an alarm message can pass the wireless channel between each status exchange.

Knowing the transmission time of a request- and reply sequence and the transmission time of an alarm message the maximum number of detectors can be calculated to 750 detectors.

Although the prototype is only implemented for one detector, a performance test to determine the maximum number of detectors is performed. By reducing the delay between status requests being sent from the central unit, the detector is forced to simulate several detectors. While

running the test at this increased pace, alarms are generated. The time between the alarm generation and the point where the alarm is recognised by the central unit is measured, and must be below 3 s. Testing with 750 detectors is not possible. After a short time the central unit times out because status replys are not received within the deadline of 150 ms and the detector stops sending alarms. Replacing the wireless link with a serial cable leads to a successful measurement. All 100 generated alarms are received within the time limit, with a mean time of 308 ms. This result clearly indicates congestion in the wireless link. However, it can not be concluded, that the designed system is not capable of handling 750 detectors since the test is performed under artificial conditions. By letting a single detector simulate 750 detector the period between receiving status requests is only 133 ms. This stresses the detector and particularly the radio board in the detector significantly more than it is designed for.

The same test is performed using the radio boards with 1 detector and 250 detectors (simulated by the single implemented detector). Both results are successful with mean alarm reception times of 337.8 ms and 413 ms respectively. The mean time measured with many detectors indicates that the radio modules have to wait longer to obtain channel access.

The DS/EN 61508 Standard requires that a safety-critical system such as the wireless fire alarm is tested to ensure an error rate of 10^{-9} errors/hour. This software is tested without errors for 310 hours meaning that 15 hours of operation without failure can be guaranteed with a certainty of 99.9999999%.

By measuring the power consumption during transmission and when the radio board is put into sleep mode using the enclosed software utility the battery life for the 1 Ah battery supplied with the board is calculated to 18.5 months, which is considered satisfactory.

A series of functionality tests shows that the necessary functionality that can be tested with one detector works as intended.

By considering the results of the tests executed, it can be concluded that a robust wireless fire alarm protocol has been designed, implemented, and tested. The designed fire alarm system features low power consumption, high reliability, and compliance with the most critical requirements regarding timing in current standards. Furthermore it is concluded that ZigBee is well suited as the communication technology of a wireless fire alarm system.

With a few modifications the protocol and the wireless fire alarm system can become completely compliant to the DS/EN 54 Standard and Precept 232 and ready production maturation.

9.2 Future Improvements

In order to obtain a product ready for production, the detector software should be implemented in the microcontroller on the radio board.

For the fire alarm system to be useful in a practical installation another cabled network between central unit and gateways should be implemented.

In order to perform a full scale test of the system more detectors should be added. It would be advantageous to redesign the central unit using an object oriented tool such as UML and do an object oriented implementation using C++. To increase the number of devices a more

precise schedule is needed. This could be obtained by implementing the central unit software on a real-time operating system such as RTLinux or using a real-time kernel patch such as RTAI.

By obtaining full knowledge to the upper layers of the ZigBee protocol stack feedback could be used by the application layer protocol, so that the reply messages could be omitted. This requires that information on whether a packet transfer succeeded or not can be obtained from the application layer protocol. If this is possible the amount of traffic can be halved providing a potential of twice as many devices in a system.

9.3 Project Evaluation

During the development of the wireless fire alarm protocol knowledge and experience is gained with the areas of:

- Analysing existing technology based on well defined criteria.
- Protocol Design & Implementation.
- Complying with standards and precepts during the analysis, design, and implementation phases of a product development cycle.
- Incorporating new and partly specified technology in product development.
- Using UPPAAL to design and verify distributed systems.

Through the development process a number of factors have been causing problems. Obtaining standards from IEEE and The Danish Institute of Fire and Security Technology have been a very time consuming task, and has caused delays in the analysis and design phases. The technical documentation provided by Adcon Telemetry on the ZigBee radio modules and the software on them have been inadequate, which has caused unexpected difficulties during the implementation phase. Furthermore the radio modules does not appear to function correct in "Transparent Secured Mode", which implies that retransmissions can not be used. This causes trouble when the distance between detector and gateway approaches the limit of range.

As described in the introduction on page 1, according to the formal semester description outlined by the study board, the purpose of project is:

- *To provide knowledge and understanding of analysis and design methodologies of distributed real-time systems.*
- *To provide understanding of the importance of reliable behavior in dependable systems.*

Based on this purpose, the project group is satisfied with the knowledge gained, and the results obtained through the process of elaborating this project.

Bibliography

[1] ESN Theme description of 9th semester of Distributed Application Engineering, 2003. http://esn.auc.dk/Studieordning_PDF/specialer/4_06_distributed_application.htm.

[2] Tommy Brandi Krog.
Teknik bremser trådløs brandsikring, 2003.
http://www.ing.dk/apps/pbcs.dll/article?AID=/20030718/BYGGERI/107180031.

[3] Dansk Standard.
DS/EN 54: Branddetektorer og -alarmsystemer, 2000.

[4] Dansk Brandteknisk Institut.
Forskrift 232: Forskrift for automatiske brandalarmanlæg, 1996.

[5] David B. Johnson, David A. Maltz, and Josh Broch.
The Dynamic Source Routing Protocol for Multi-Hop Wireless Ad Hoc Networks.
Addison-Wesley, 2001.

[6] European radiocommunications committee, 2003.
http://www.ero.dk/.

[7] The european telecommunications standards institute, 2003.
http://www.etsi.org/.

[8] Wikipedia, 2003.
http://www.wikipedia.org/wiki/ISM_band.

[9] Radian standard radio protocol, 2003.
http://www.radian.com.

[10] Helicomm inc., 2003.
http://www.helicomm.com/sol_zigbee.htm.

[11] ZigBee Alliance, 2003.
http://www.zigbee.org.

[12] European Radiocommunications Committee.
Erc decision of 30 june 1997 on the harmonised frequency band to be designated for social alarm systems, erc/dec/(97)06, 1997.

[13] European Radiocommunications Committee.
Erc recommendation 70-03, relating to the use of short range devices (srd), 2002.

[14] European Standard.
En 300 220-1 v1.2.1 (1997-11), 1997.

[15] The UPPAAL Team, 2003.
http://www.uppaal.com.

[16] Preece, Rogers, and Sharp.
Interaction Design - beyond human-computer interaction.
John Wiley and Sons, 2002.

[17] Dansk Standard.
DS/EN 61508-3: Funktionel sikkerhed for elektriske/elektroniske/programmerbare sikkerhedsrelaterede systemer - del 3: Krav til software, 2002.

[18] Stephen Biering-Sørensen & Finn Overgaard Hansen & Susanne Klim & Preben Thalund Madsen.
Håndbog i Struktureret Programudvikling.
Ingeniøren|Bøger, 2000.

[19] Elmer E. Lewis.
Introduction To Reliability Engineering.
John Wiley and Sons, 1996.

[20] Williams Anderson, Sweeney.
Statistics for Business and Economics, 8th edition.
South-Western, 2002.

[21] Simon Haykin.
Communication systems, 4th ed., 2001.

[22] IEEE Computer Society.
Part 15.4: Wireless medium access control (mac) and physical layer (phy) specifications for low-rate wireless personal area networks (lr-wpans), 1. Oct 2003.

[23] Fsk: Signals and demodulation, 2003.
http://www.wj.com/pdf/technotes/FSK_signals_demod.pdf.

[24] Alan Burns and Andy Wellings.
Real-Time Systems and Programming Languages, Third Edition.
Addison Wesley, 2001.

Part V

Appendices

Fire Detection and Fire Alarm Systems

Appendix A

This appendix gives a short resume of standard DS/EN 54 [3] and Precept 232 [4]. Only the parts relevant for this project is included in the appendix.

If a fire alarm system is to be accepted by the authorities, it must fulfill the requirements given in standard DS/EN 54. The standard contains 11 parts, and 6 of them are requirements for different fire detectors. These parts are not relevant for this project. The only relevant part for this project is DS/EN54-2 *Control and indicating equipment*.

The Danish Institute of Fire and Security Technology has made a precept about automatic fire alarm plants Precept 232. It is divided into 9 parts, where the most relevant parts for this project are: *000-019 The plant* and *080 Wireless Equipment*. Part *080 Wireless Equipment* is not available for the project group and is thereby not explained in this appendix.

Figure A.1 shows the components forming a fire detection and fire alarm system.

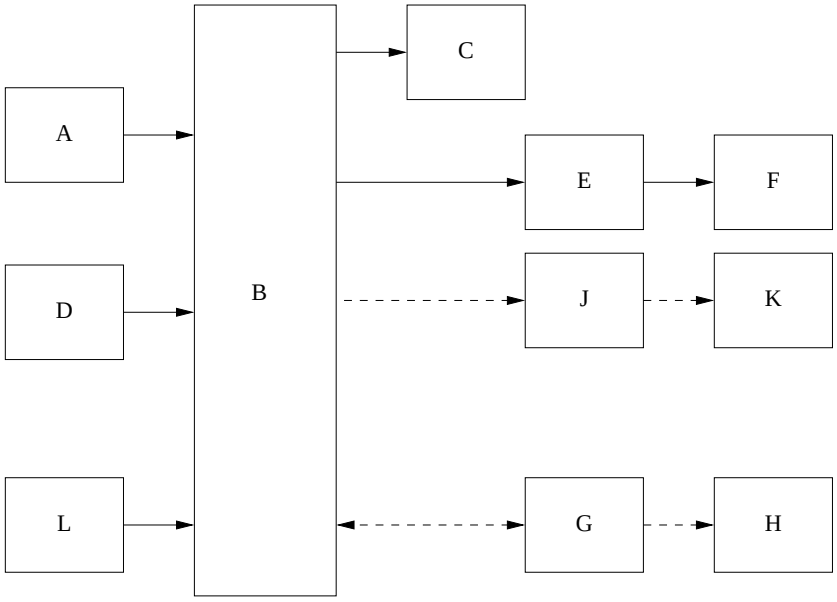

Figure A.1: *The components forming a fire detection and fire alarm system.*

The components will be referred to as:

A Fire detector(s).

B Control and indicating equipment (c.i.e.).

C Fire alarm device(s).

D Manual call point(s).

E Fire alarm routing equipment.

F Fire alarm receiving station.

G Control for automatic fire protection equipment (optional)(can be placed in B).

H Automatic fire protection equipment (optional).

J Fault warning routing equipment (optional)(can be placed in E).

K Fault warning receiving station (optional)

Other elements such as an operating panel, control functions and monitoring system can be placed on the figure as well, but they can also be a part of B.

The monitoring area for one system must, in normal circumstances, not exceed 10.000 m^2. The system may be divided into zones. One zone can have one or more sensors and alarm devices. A zone area may not exceed a given value, according to the number of rooms in the zone. If there is only one room, the zone floor area can be up to 1.600 m^2, if there is more than 15 rooms the zone floor area must not exceed 400m^2. A detailed specification can be found in [4, part 010]. A zone may be on one floor only.

A.1 Conditions

The c.i.e. shall be capable of unambiguously indicating any combination of the following functional conditions:

- Fire Alarm Condition.
- Fault Warning Condition.
- Disabled Condition.
- Test Condition.

The c.i.e. shall also be capable of being simultaneously in any combination of the conditions. E.g. one zone is in disabled condition and another zone is in fault warning condition at the same time. If an error occurs in a zone it must not affect the alarm system for any other zone.

A.1.1 Fire Alarm Condition

The c.i.e. (component B) shall enter the fire alarm condition when fire alarm signals are received from a sensor. The c.i.e. shall be able to receive signals from all zones and a signal from one zone shall not falsify receiving and indication of a signal from another zone.

The time from a signal is received, from a sensor (component A) or a manual call point (component D), until the c.i.e. enters the fire alarm condition, must not exceed 10 s. The zones in alarm shall be visibly indicated by means of a separate light emitting indicator for each zone and/or a alphanumeric display. The audible alarm shall resound for each new zone in alarm and it shall not be silenced automatically.

The c.i.e. shall be capable of being reset from the fire alarm condition. After a reset operation the c.i.e. shall be re-established within 20 s.

The c.i.e. shall action all mandatory outputs within 3 s of the indication of a fire alarm condition if the alarm is activated by a sensor (component A). If the alarm is activated by a manual call point (component D), the requirement is 10 s.

The c.i.e. may have provision for automatic transmission of fire alarm signals to fire alarm routing equipment (component E). This indication shall remain until the fire alarm condition is manually reset.

A.1.2 Fault Warning Condition

The c.i.e. shall enter fault warning condition when signals are received which are interpreted as a fault. It shall be capable of simultaneously recognising the faults listed below, unless this is prevented by:

- the presence of fire alarm signals from the same zone and/or
- the disablement of the corresponding zone or function and/or
- the testing of a corresponding zone or function.

The c.i.e. shall enter the fault warning condition within 100 s of the occurrence of the fault or the reception of a fault signal. The following faults shall be indicated:

- a short circuit or interruption in a detection circuit in a point.
- the removal of a point (lost contact).
- a short circuit or an interruption in a transmission path to a power supply.
- other power supply faults.
- any fault which is capable of affecting a mandatory function.

A system fault is a fault that prevents the given requirements from being fulfilled. This can occur in case of use of software in the c.i.e. A system fault shall be visibly and audible indicated on the control panel.

The system may manually or automatically be reset if a fault no longer is recognised. The system shall be re-established within 20 s after a reset operation.

The c.i.e. shall have an output which signals all faults specified above. The output signal shall be given if the c.i.e. is de-energised.

A.1.3 Disabled Condition

The c.i.e. shall be in disabled condition while a disablement exists.

The following shall be capable of being independently disabled and re-enabled:

- each zone.
- output signals and/or transmission paths to fire alarm devices. (Component C)
- output signals and/or transmission paths to fire alarm routing equipment. (Component E)
- output signals and/or transmission paths to fault warning routing equipment. (Component J)
- output signals and/or transmission paths to control for automatic fire protection equipment. (Component G)
- each addressable point (optional requirement). It shall be possible to identify all the disablements by manual interrogation. (Component A and D)

The disabled condition shall be indicated visibly by a separate light emitting indicator and an indication for each disablement. The indication shall appear within 2 s of the manual disablement.

A.1.4 Test Condition

The c.i.e. may have provision for testing the processing and indication of fire alarm signals from zones. This may inhibit the requirements during the fire alarm condition which corresponds to that zone. In this case at least the following shall apply:

- The c.i.e. shall be in test condition while one or more zones are under test.
- A test state shall only be entered or cancelled by a manual operation.
- It shall be possible to test the operation of each zone individually.
- Zones in the test state shall not prevent the mandatory indications and outputs from zones not in the test state.

- Signals from a zone under test shall not lead to the operation of the outputs to fire alarm devices, fire alarm routing equipment, controls for automatic fire protection equipment or fault warning routing equipment.

The test condition shall be indicated visibly by a separate indicator for each zone.

A.2 Input/Output Interface

The c.i.e. may have provision for a standardised input/output interface. In this case at least the following shall apply.

The interface shall be capable of transmitting at least the occurrence of:

- the fire alarm condition.
- each zone in alarm.
- the transmission of output signals to fire alarm routing equipment.
- the transmission of output signals to fire protection equipment.
- the fault warning condition.
- each zone fault.
- the disablement and re-enablement of each zone.
- the disablement and re-enablement of the output to fire alarm devices.
- the disablement and re-enablement of the output to fire alarm routing equipment.

The interface shall be capable of receiving at least the following information and of activating the corresponding functions of the c.i.e.:

- silencing of the audible indication.
- the reset of the fire alarm condition.
- silencing and re-sounding of fire alarm devices.
- the disablement and re-enablement of zones.
- the disablement and re-enablement of output signals to fire alarm devices.
- the disablement and re-enablement of output signals to fire alarm routing equipment.

A.3 Software Requirements

The manufacturer shall prepare documentation which gives an overview of the software design. This documentation shall be in sufficient detail for the design to be inspected for compliance with standard DS/EN 54, and shall comprise at least the following:

- A brief description of the main program flow. See details in [3, part 2, page 28].
- A description of which areas of memory are used for the various purposes.
- A description of how the software interacts with the hardware.
- A description of each module of the program. See details in [3, part 2, page 29].
- The source code listing, including all global and local variables, constants and labels used, and sufficient comment for the program flow to be recognised.
- Details of any software tools used in the preparation of the program (tools, compilers etc.)

In order to ensure reliability of the c.i.e. the following requirements for software design shall apply:

- The software shall have a modular structure.
- The design of the interfaces for manually and automatically generated data shall not permit invalid data to cause an error in the Program execution.
- measures shall be included in the program to prevent the occurrence of a deadlock in the system.

A lot of demands about what memory to use is given in the standard, but this will not be taken into account in this project, since this is a prototype. If the product is to be produced, this has to be taken into account.

Frequency Modulation

Appendix B

Architectures for transporting data from one device to another in wireless systems are organised in the same way as in wired systems. Namely in layers as the OSI model specifies. The following description focuses on the wireless aspects of the layers for the communication systems in this project.

Two important OSI layers are the Physical (PHY) and the Media Access Control (MAC) layer. The PHY consist of low level control mechanism for the Radio Frequency (RF) module and is responsible for broadcasting messages between terminals on a given medium. The MAC layer gives access to the PHY from the layer above (data link layer) by handling and maintaining the connectivity of data frames between the network devices. These data frames consist of all kinds of services, having different needs of handling and control mechanisms. The MAC layer is responsible of providing these mechanisms.

B.1 Physical Layer

Delivering data messages from one terminal to another in a communication system requires a reliable transportation system. The transmitting terminal modifies the data message into a form suitable for transmission over the specified medium. The receiver has to decode the modified message to get the full content. In wireless systems radio waves are modulated in accordance with the message signal. Different methods for modulation are used depending on the purpose of the data communication, available bandwidth, range between terminals, noise level and the surroundings.

Modulation is classified either as continuous-wave modulation or pulse modulation. In continuous-wave modulation (CW) the data message is modulated along a sinusoidal carrier. A method for transferring messages is to vary the amplitude of the sinusoidal carrier in accordance to the message, this method is amplitude modulation (AM). By varying the angle of the carrier or the frequency is another method, called phase modulation (PM) and frequency modulation (FM).

In pulse modulation, the carrier consists of a periodic sequence of discrete rectangular pulses. The pulse modulation it self can be analogue (PAM) or digital (PDM). Analogue pulses are varied in amplitude, duration, or position in accordance to the message depending on the methodology. The typical form of digital pulse modulation is known as pulse-code modulation (PCM). PCM transfers only the binary values as one or zero. Analogue messages has to be quantised before transmitted and analogue decoded when received. [21]

B.2 Media Access Control Layer

The ZigBee protocol uses several services to maintain the wireless network. These services relies in the MAC layer. Examples are:

- Keeping the synchronisation of broadcasted network beacons.
- Supporting security.
- Employing of the CSMA/CA mechanism for channel access (Carrier Sense Multiple Access with Collision Avoidance).

In this project the ZigBee protocol is configured to use a peer-to-peer network model when transferring data between ZigBee devices. Hereby there are no central network coordinator to maintain any hierarchy of the use of the network. All privileges are coordinated by CSMA/CA.

When a CSMA/CA node wants to broadcast a data packet, it has to ensure that the network is not occupied by another user, otherwise collisions would occur. The method is described on figure B.1 where a packet is pending to be transmitted.

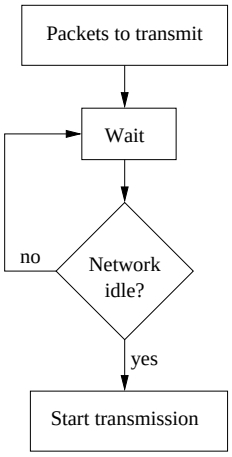

Figure B.1: *Simplified diagram of the CSMA/CA method.*

Figure B.1 indicates that the first step is to wait a random period of time before the node checks whether the network is idle before transmitting the data. If the network is occupied, the node has to wait another random time, before the network is checked again. The waiting time is called the *backoff time*, because it indicates the time when the node may not use the network. The time is calculated as an exponent relative to other parameters in the CSMA/CA algorithm. The backoff time is increased as the number of retries grows until a time-out occurs.

When the network becomes idle CSMA/CA sends a Request to Send (RTS) message to the destination node. If the node receives a Clear to Send (CTS) message from the other node, it continues its data transmission. When the data packets has been received, the receiving node sends an acknowledgement packet. This structure is depicted on figure B.2.

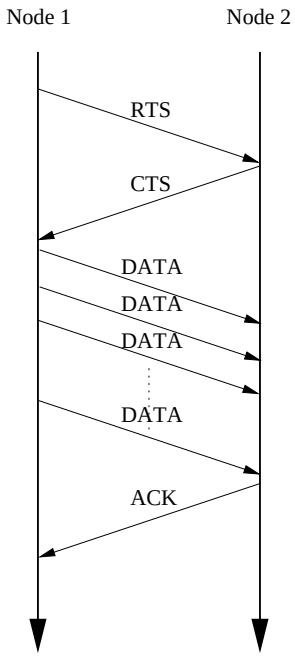

Figure B.2: *Transmission of data from one node to another.*

Common problems arise when using the CSMA/CA method. In real-time perspective, no timing can be guaranteed because the channel access is pending on a random waiting time. As more devices are attached to the network, the waiting time before the data transmission can be started grows. The *hidden terminal* phenomenon arises if two devices start transmission at the same time, because they are too fare apart from each other to detect that the communication channel is occupied by the other part. [22]

B.3 Frequency Shift Keying

The Adcon Addlink radio modules used in this project uses a modulation method called Frequency Shift Keying (FSK). This type of modulation transmits data by shifting the frequencies of a continuous carrier in a binary manner to one or the other of two discrete frequencies. One frequency is designated as the *mark* frequency and the other as the *space* frequency. The mark and space corresponds to binary one and zero respectively. Figure B.3 shows how the two frequencies corresponds to the binary data sequence.

When dealing with FSK modulation, more than one frequency is used for the data transfer transaction. The bandwidth between the two frequencies is stated as the *shift* which is the frequency difference between the mark and the space frequency. Shifts are usually in the range of 50 to 1000 Hz depending on the base frequency used.

The length (in time) of a mark or a space element is stated in terms of the transmitting speed. Shorter length gives higher speed. Taking the inverse of the length is equal to the transmitting

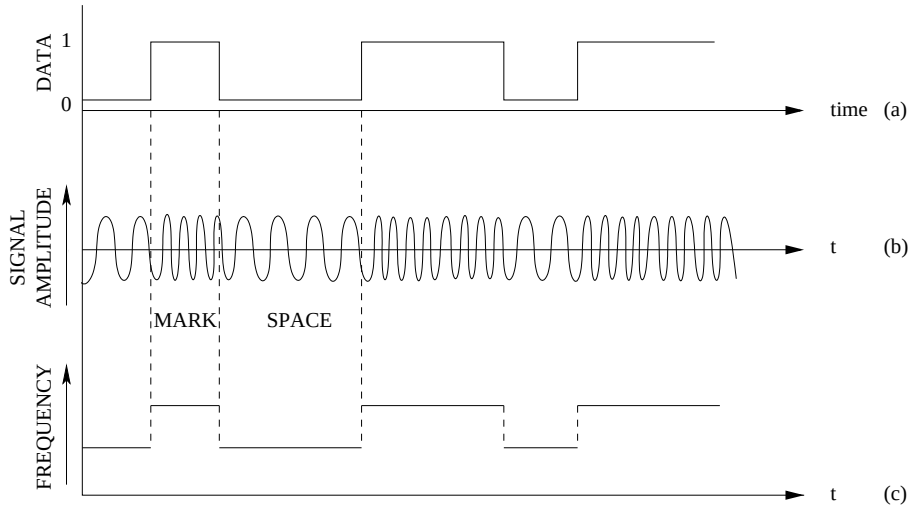

Figure B.3: *FSK modulation method. (a) the data sequence, (b) modulated continuous signal, (c) relation between frequency.*

baud rate which in FSK modulation is called the *keying* speed.

Different coding schemes for FSK exist. Some schemes are developed to gain higher reliability while others are for speeding up the date rate. Instead of using different frequencies, two different phases could also be used. The speed of FSK is increased if more than two frequencies are applied to the modulation, this method is called Frequency Division Multiplex (FDM), but details are omitted in this project. [23]

Analysis and Design of Fault-Tolerant Systems

Appendix C

This appendix describes important parameters and methods to consider during analysis and design of fault-tolerant systems. The appendix is based on the contents of the course "Analysis and Design of Fault-tolerant Distributed Real-time Systems" given by Associated Professor Roozbeh Izadi-Zamanabadi and Professor Ole Brun Madsen at Aalborg University, fall 2003.

When developing safety-critical devices such as fire alarms or industrial robots a major objective is to obtain a reliable and safe design. In the past, the normal way of achieving this was to add hardware redundancy, but in the recent years research within active fault-tolerant design has produced methods for improving the stability and availability, while minimising cost.

For safety critical systems the following aspects have to be considered:

- Availability.
- Maintainability.
- Reliability.
- Safety.

In the following sections possible considerations to be taken within the topic of these aspects are described.

C.1 Availability

Availability is the probability of some device performing satisfactorily at a given time. Within computer terminology availability is often referred to as uptime. For safety critical devices a very high availability is necessary. One way of achieving a high system availability is to introduce redundancy on critical devices. The availability depends on many factors including software stability, load, and the reliability of the infrastructure in which a device is operating.

C.2 Maintainability

Maintainability covers the probability that a device can be restored to a specified condition when maintenance is performed by trained personnel following prescribed procedures using prescribed

resources and equipment. To maintain a high maintainability in some device or system, the documentation needs to be very thorough and adequate. Otherwise a technician or service mechanic will not be able to fully repair the system. Also a strict policy should be applied to the use of spare parts. Only original, or absolutely identical spare parts should be used.

C.3 Reliability

Reliability can be defined as the probability of a device performing its required function in a specified manner within a given time period, and under specified and assumed conditions.

Reliability and safety are closely related, but not identical. While using a selection of techniques may improve the reliability of a device or system, it does not necessarily improve safety as well.

In fact reliability and safety overlaps each other. Accidents and damage may occur without component failure, but component failure can also happen without causing accidents or damage.

C.4 Safety

A device, or a piece of software, is said to be safe if it is impossible (or at least highly unlikely) that the software could ever produce an output that would cause a catastrophic event for a system to which it belongs or to which it interfaces. A catastrophic event could be the loss of life, personal injury or perhaps environmental damage due to discharge of chemicals.

In general, safety and reliability is achieved by a combination of:

- Fault Prevention.
- Fault Tolerance.
- Fault Detection and Diagnosis.
- Autonomous Supervision and Protection.

Fault Prevention is a combination of fault avoidance and fault removal. These two activities has to be accomplished during the design and test of a product. *Fault Tolerance* is the ability of a system or device to tolerate a number of faults during its lifetime, and still maintain the intended operation. By applying *Fault Detection and Diagnosis* a system or device can respond intelligent to faults occurring and diagnose which actions should be taken. A system or device featuring *Autonomous Supervision and Protection* contains means of determining if the present state corresponds to the intended behaviour, and take actions if this is not the case.

The necessity of each of these elements depends on the application, but for a safety-related system all aspects of reliability, availability, maintainability, and safety must be considered. These four properties are relevant to the responsibility of the manufacturer, as well as to the acceptability of a customer.

Various techniques for achieving effective safety-critical systems exist. One of such techniques is Hazard Analysis. In short, the processes in the technique is to identify possible hazards and rank

them according to hazard level, which is a combination of severity, and likelihood or frequency of occurrence.

Since the hardware used in this project is mostly developed by external vendors, the main objectives within safety and reliability analysis lies in the protocol and the software used to implement it.

C.5 Software Fault-tolerance

Software systems generally provide some sort of services - either to users or to other pieces of software or hardware to which an interface is present. Two general failure classes within software can be identified:

- Value Failure.
- Time Failure.

When a *Value Failure* occurs the outcome of the software is simply wrong. It could be a mathematical algorithm ending up with a wrong result. A *Time Failure* occurs when the software - or service delivers the result in wrong time. This could be either too early or too late - or maybe even infinitely late. Failures that combine these two classed are called arbitrary.

As within hardware, redundancy can also be used in software. This can be achieved by letting a number of people develop a piece of software, based on the same specification but without interaction during the development process. These software pieces execute concurrently, and the correct result is chosen by comparing and voting between the results of the individual software pieces. [24]

C.6 Practical Approaches

When developing protocol software for a safety-critical system such as a wireless fire alarm several steps can be taken to avoid value-, time-, and arbitrary failures.

The analysis of requirements and demands from external organisations such as local authorities, standards and precepts is a very important part of the analysing phase of the development process, when developing safety related systems. When the system contains both software and hardware, it is important to perform a thorough investigation of reliability issues prior to selecting a specific piece of hardware to build the system on. Well designed software can not compensate for unstable hardware. When the system to be developed is a Wireless Fire Alarm, then the hardware analysis should form the basis of a well-founded choice of which communication standard and technology to use. Important properties would be reliability, low possibility of interference and a stable, acceptable range.

After having analysed requirements and selected a hardware platform to use, the design of software can begin. During the design phase it is important to discover as many pitfalls as possible,

because they are easier and cheaper to correct in the design phase than during implementation. Therefore simulation and verification should be a part of the design process.

Several simulation tools are available to test state machines for dead- and live-locks. This can be used to ensure that the behaviour of a designed protocol performs as intended. One example of such tools is UPPAAL, but tool-boxes for Matlab can also do similar simulations. UPPAAL integrates tools for modelling, validation and verification of real-time systems modelled as networks of timed automata. This combination of tools makes it possible to do a complete model of the communication process and step through a simulation while knowing exactly which state each device in the communication system is in.

One well known principle when designing software, is that minimising the amount of code, often leads to more reliable software. Keeping things simple leads to a more robust product, because error possibilities are easier spotted during development and coding.

When designing a communication system the reliability can be improved by maintaining a fixed schedule in the communication. If messages need to arrive at one host at a fixed time, then it can easily be determined if it succeeded or not. This way time failures can be detected, and steps can be taken. One way of obtaining a fixed schedule is to use master/slave communication. Here the master requests all communication, meaning that no spontaneous communication arises from the slave to the master.

After having verified that a valid protocol and software design have been obtained, the next step is to implement it. In order to maintain the robustness and reliability obtained through simulations and verification it is very important that the implementation follows the simulation model closely. Otherwise new pitfalls may be introduced.

The last step in the development process is to test and verify that specifications and requirements have been met. This is a very time consuming task that should include both standardised test cases as well as some amount of creative testing. The test phase may also include external audit in order to receive some sort of product approval from the authorities.

Nomenclature

The nomenclature is sorted alphabetically for symbols and letters respectively.

AI	:	Audible Indication.
c.i.e.	:	Control and Indicating Equipment.
Can-bus	:	Controller Area Network bus.
CEPT	:	Comité européen de Réglementation Postale.
CSMA/CA	:	Carrier Sense Multiple Access with Collision Avoidance.
CU	:	Central Unit.
CW	:	Continuous-Wave modulation.
DC	:	Disabled Condition - the state the WFA is in when it is disabled.
DET	:	Detector.
DSR	:	Dynamic Source Protocol.
ERC	:	European Radiocommunications Committee.
ERM	:	Electro magnetic compatibility and Radio spectrum Matters .
ETSI	:	The European Telecommunications Standards Institute.
FAC	:	Fire Alarm Condition - the state the WFA is in when an alarm has been detected.
FDM	:	Frequency Division Multiplex.
FSK	:	Frequency Shift Keying.
FWC	:	Fault Warning Condition - the state the WFA is in when an error or fault has been detected.
GCC	:	GNU C Compiler.
GD	:	Graphical Display, the display on the control panel of the CU.
GW	:	Gateway.
ISM	:	Industrial, Scientific and Medical radio bands.

LEI : Light Emitting Indicator, a general indicator on the control panel, indicating that some functional condition is present.

MAC : Media Access Control layer .

MCU : Microcontroller Unit.

Mutex : A flag used to lock critical regions in software.

PAM : Pulse Analogue Modulation.

PCM : Pulse-Code modulation.

PDM : Pulse Digital Modulation.

PHY : Physical layer in the OSI model.

PoE : Power Over Ethernet.

QC : Quiescent Condition - when the WFA is in normal condition, enabled with no errors or alarms present.

RF : Radio Frequency.

SDK : Software Developers Kit.

SPU : Struktureret Programudvikling - a structured method for developing software.

SRD : Short Range Devices.

TC : Test Condition - the state the WFA is in when it is running in test mode, meaning that alarms are not sent on to the fire station.

WFA : The wireless fire alarm developed in this project.

CD-ROM

Appendix E