SOFTWARE CONTROLLED FIRE DETECTION SYSTEMS

— DESIGN AND TESTING —

Heinz O. LUCK and Udo SCHLOSSAREK

Duisburg University

Electrical and Electronic Engineering

Dept. of Communication Technology

Bismarckstr. 81 ; D-4100 DUISBURG, FRG

(Received, January 18, 1988)

ABSTRACT

Software-controlled fire detecting installations are of increasingly growing importance and attraction. The paper discusses the advantages as the main reasons for this fact. In addition it gives an indication of the problems to be solved and the inherent failure sources. The main applicable design rules are derived from the principle that failure sources have to be avoided and where they are unavoidable their effects have to be minimized. The basic design approaches are discussed in this context. It is mentioned that in practice a combination of several aspects has to be applied for several reasons. Testing is necessary on the different levels of the design, development and application process. The contribution discusses the possible test philosophies for type testing in some detail and mentions a special test procedure for software-controlled fire detection systems which has been discussed in the European community and which is a compromise between a rigorous functional (dynamic) testing and a documentation check. It is likely to give suitable over-all indication of the system quality, it checks several key specifications and it keeps the testing expenses at an acceptable level.

It is shown that black-box testing is not possible.

1. INTRODUCTION

Software-controlled means of different kinds have been introduced into automatic fire detection systems during the recent years. They are going to gain increased importance in the next future and this fact gives rise to deal with design methods and testing possibilities for these techniques.

The main difference compared with conventional fire detecting systems which may already contain a considerable amount of electronics for signal and data processing is given by the fact that important parts of the system's performance is given by the execution of a bunch of software programs. The application of software in cooperation with programmable electronic and mechanical elements introduces a new quali-

ty in design and performance of the whole system that can be characterized by the following items^[1] :

- The over-all performance of the system is no longer uniquely predetermined by the physical design (hardware). It depends very heavily on the cooperation between hardware and software.
- In order to enable a proper program design, all possible branches of the system's performance (execution) have to be determined in advance. A very careful design procedure including the hardware and its characteristics is therefore required.
- Both parts hardware and software may with the same significance contribute to defects in the system or to failures in its performance.

Design methods and testing are of the same importance for both hardware and software. So as in the hardware case approved design methods should be used for the software that are likely to avoid systematic design errors.

- A very high grade of flexibility is introduced by the use of software solutions for system tasks. Changes in the system performance

can very easily be introduced by changes in the software programming without any change in the hardware at all.

In order to check how important software solutions in automatic fire detection systems may become in principle the following table gives a rough survey which main tasks of such system may be suitable for software solutions^[1,2].

Table : Accessibility of software solutions for functions in fire detecting systems

<p>Not accessible to software solutions :</p> <p>Measuring procedure of fire parameters</p> <p>Power supply</p>
<p>Not directly accessible but suitable for a software assisted solution:</p> <p>Display (optical and acoustic)</p> <p>Signal transmission to remote parts</p>
<p>Well accessible to software solutions :</p> <p>Automatic monitoring of the system</p> <p>Means making available checking possibilities</p>
<p>Perfectly accessible to software solutions:</p> <p>Signal and data processing</p> <p>Recording and storage of data</p> <p>Operation support</p>

It can be derived from the table that a considerable part of the whole detecting system may be realized by a software program running on a mini-computer or at least partly implemented on special custom VLSI-chips. The advantages associated with the use of software controlled electronics are:

- Reduction in volume, weight, expensive raw materials and energy consumption.
- High flexibility by easy software adaptability to different and changing environmental conditions.
- Easy handling of high quantities of measured and stored data associated with the possibility of reliable and quick execution of countermeasures in the alarm or in fault case.

- Considerable increase of storage capabilities for important data.
- Considerable increase in the possibility of automatic monitoring important parts of the system and identification of errors.

So it is not surprising that these modern means are increasingly used for three different goals:

- Improvements of the handling and operating characteristics of the system in the cases of alarm or faults (s. c. handling intelligence).
- Improvements of the automatic monitoring which result in higher reliability, increased availability and a better guidance for the maintenance procedure (s. c. monitoring intelligence).
- Increase in the detection capability, that is a

better capability to distinguish between the FIRE and the NON FIRE case (s. c. detection intelligence or efficiency).

In this context the false alarm rate is expected to be considerably reduced.

2. INVOLVED PROBLEMS AND FAILURE SOURCES

Several different failure sources have to be taken into account in any technical system. Fire detection systems are mainly affected by the following items :

- (1) Errors and faults which are due to physical or chemical breakdowns of system elements and which may be transient.
- (2) Draft, design and manufacturing errors involved in even more complex structures.
- (3) Operating and maintenance errors which are the more likely to occur the more complex the system becomes.
- (4) Errors due to shortcomings in the detection capability of the fire detection system. False alarms due to detector deception are the main result.

The first kind of failure can be reduced to an acceptable amount by the use of reliable system elements and by designing the whole system to be insensitive against environmental conditions including electromagnetic interference. In the following this paper will not deal with this aspect. But it is handled in more detail elsewhere^[4]. The second two are man-made. The efforts to reduce them govern the methods for design and testing almost essentially. The following paragraphs will discuss these aspects. The forth includes the question of the design of suitable detection algorithms. It gains very much from the application of software controlled electronics and work is going on in this field^[2,5,6]. But the point is beyond the scope of this article. It is not discussed in the following. The introduction of software solutions into automatic fire detection installations adds the following special features to the problem :

- The software itself is an additional failure

source of special kind. This is a well known fact but not avoidable^[1].

- Signal and data processing in software-controlled systems is mainly executed via an integrated microprocessor. This means a high degree of centralization in the alarm and fault signalization path. The result is a relatively easy monitoring of the system's performance on the one hand but it increases the weight of errors and breakdowns in this central element on the other.
- Software-controlled electronics are in a special sense sensitive to transient hardware failures, electromagnetic interferences and erroneous input data, because the program flow may be affected in an uncontrollable way.

3. DESIGN STRATEGIES FOR COMBINED HARDWARE AND SOFTWARE DETECTION SYSTEMS

Design strategies for fire detection systems are not different from those for other purpose systems. They are aimed to minimize the number of failure sources or to avoid significant failure consequences, i. e. to increase reliability and availability. Two basic strategies are possible. The *perfectionist approach* is the attempt to avoid all possible failure sources, while the *fault tolerant approach* is based on the view that failures and breakdowns in technical components are unavoidable matters of principle and therefore measures have to be taken to minimize their effects. In the following the main measures under both approaches are listed and brief comments are given where necessary^[10,11] :

3.1 The perfectionist approach:

3.1.1 Measures against failures due to physical and/or chemical influences:

Burn-In for system components, quality checks throughout the whole manufacturing period, long term testing under various environmental conditions, hardware design with

maximum possible signal-to-noise ratios, screening, potential isolation are very widely used tools. The software itself is not directly affected by component breakdowns or external interferences. All software failures are created during the design procedure including corrections made afterwards.

3.1.2 Measures against design errors :

Hardware : Responsible attention has to be paid to common hardware design rules, e. g. limited energy release or countermeasures against overheating, use of standardized components, careful observation of component specifications etc.

Software : The software design has to be based on unique, consistent and complete definitions of the task and the structure of the detection system associated with a careful documentation. The program structure should be divided into modules that execute special functions with clearly defined interfaces to other program or system parts including the parameters to be transferred^[7]. Structured programming must be used and a unique and easily understandable documentation is mandatory, i. e. the documentation has to facilitate verification which is achieved by top-down-design and Nassi-Shneiderman flowchart techniques^[7,8]. Problem oriented programming languages should be used that are completely and unambiguously defined and should be applied with a thoroughly tested compiler. Any retrospective attempts to optimize memory space or execution time or to alter the structure of the program should be avoided.

3.1.3 Measures against operating and maintenance errors :

Hardware: No operating controls and input facilities should be available from outside the installation.

Software: No possibility to change the program or important parameters by the use and the maintenance personnel should be accepted.

3.2 The fault tolerant approach :

Here the measures to be taken are not distinguished as uniquely associated to hardware or software. This strategy takes into account unavoidable failures (transient and not transient) that may affect the performance of both hardware and software. The approach is to recognize these incorrect performances and to minimize their effects.

3.2.1 Measures against failures due to physical and/or chemical influences :

A commonly used measure is to introduce redundancy in the hardware structure. This is a powerful tool against single faults or transient performance breakdowns in limited parts of the system but it results in considerably increased technical efforts. A graduated system to introduce redundancy measures has been proposed^[1]. Another common measure is to give the system a fail-safe behaviour. That means the system is designed to go to the safest possible state if a failure in some subsystem occurs. In the case of a fire detection system this would cause some problems, because the safest possible state is from the point of view of public safety the alarm state. This would cause an increase in the false alarm rate which of course cannot be tolerated. So in this case the fail-safe-state only can be the indication of a fault signal. On the other hand this measure cannot be applied e. g. for the output elements without some redundancy. A careful function and program flow monitoring in important parts of the system is a further possibility. Examples are:

- the cyclic monitoring of the correct function of the microprocessor, the availability of memories and input units (detector heads) ;
- the use of watch-dog techniques^[2] ;
- reasonableness checks for input data between subsystems (e. g. detectors and control and indicating units) or for input data in the operating or maintenance procedure ;
- execution control for cycles in the program

flow, e. g. by limiting the number of loop repetitions or the execution time.

3.2.2 Measures against design errors :

Hidden design errors are the more likely to occur the more sophisticated the software is going to become. Design errors in the software can be reduced by using redundancy techniques in combination with diversity. This means that important software modules are doubled or even multiplied and used to solve the same sub-problem in parallel action and the results are compared. It is essential in this context that the software modules are designed using different methods in order to ensure that the same result is achieved using different approaches (e. g. different algorithms, programming languages, compilers etc.). If not, there may be a design error somewhere. And again fail-safe techniques and program flow monitoring are suitable tools to detect hidden design errors. For example, the unused memory space in the microprocessor circuit should be loaded with well defined orders or data to bring the system into a safe, in other words well defined and recognized, state if some unwanted jump in the program flow is going to occur. In addition, the same items as mentioned under 3.1.2 are applied here.

3.2.3 Measures against operating and maintenance errors :

Under this strategy incorrect operation and incorrect input data have to be taken into account. The number of indication and operation components has to be restricted to the lowest possible level in order to avoid confusion in the alarm or fault case. Automatic checking for reasonableness of input data from the operating panel is mandatory and in the case of a display output an easily understandable user's guidance is necessary. The introduction of different levels of accessibility to the fire detection installation

- for the user and the public:

Indication of alarm and fault situations. Silencing of acoustic alarm signals.

- for the fire brigade and the maintenance personnel:

Communication with the system by easy to handle technical means, functional checks, maintenance exchange of components.

- for the installer and the manufacturer (personnel with special education):

Installation and changing operational data.

- for the manufacturer :

Detection system design, changing program modules, redesign of the system configuration.

has to be established.

In the practical case, a combination of both approaches (strategies) is used to achieve a list of suitable and meaningful technical requirements (specifications) which softwarecontrolled fire detection installations are expected to meet. From these requirements list test procedures can be derived^[8].

4. BASIC TEST PROCEDURES

Testing is necessary and is done on several different levels with different purposes:

- Test procedures are applied during the design and development phase to subsystems and prototypes in order to ensure the success of the design steps. This testing is fairly individual and uses simulation methods as well as physical testing.
- Quality testing is used to control the regularity and the observance of tolerances in the manufacturing process.
- Type testing is applied by independent testing facilities which may be governmental or private, i. e. insurance organisations, research institutes etc., to produce a background for approval mainly on a national basis, but in the next future for an international acceptance too, e. g. in the European Community from 1992 on.

For the last item it is important to have a commonly agreed test procedure which has to be based on an agreed number of technical re-

quirements which may be derived from the tasks a fire detection system is expected to fulfill. This contribution will deal with this last aspect only.

Fire detection installations are mainly systems of limited size and costs. This fact restricts the possibility to apply one of the design approaches discussed in the previous chapter in a rigorous manner. Consequently, detection equipment and practical installations result, that may not meet all desirable requirements. In practice, mainly a combination of items from both approaches is applied which is considerably different between the manufacturers. So the task of the associated testing procedures is to ensure that at least some agreed minimum requirements are met.

The most consequent approach would be to require a complete functional test, i. e. an arbitrarily selected unit (complete system) is tested applying all relevant fire and non-fire conditions including the changes in the environments and the significant possible errors in the input data. It is easily shown that this is impossible^[1]. It can also be shown that sufficiently complete dynamic testing restricted to the software part of the system is already likely to produce unacceptable test expenses. In addition the possible effort to be put into any meaningful test procedure is limited to an amount that is suitable compared to the main system costs. Therefore the following basic test philosophy for software-controlled fire detection systems has been proposed and is widely accepted in the European area:

- (1) Automatic fire detection systems are tested according to the agreed European standard EN 54, parts 7 and 9^[9]. This test procedure includes a full scale fire test series (functional tests) and several environmental tests (quality tests).

These tests are not at all complete in the sense discussed above; e. g. they do not take into account failure sources that are associated with the application of highly sophisticated electronics (microprocessors) and the software components.

Therefore the test procedure is amended by the following three paragraphs.

- (2) The hardware used for the design of the electronic part of the system is checked mainly by using the documentation according to the following aspects:
 - Choice of proper electronic components,
 - Concept and structure of the electronic circuitry (e. g. compliance with specifications concerning automatic monitoring of the processor, watch-dog, proper data storage etc.)
- (3) The software is checked by using the documentation whether its structure and its design meet the requirements for modern, consistent and clear software design:
 - Modules definition of the tasks, interface definitions, parameters,
 - Structured programming and clearly understandable documentation,
 - Programming languages properly defined,
 - Self monitoring aspects,
 - Addresses in the data memory space indicated,
 - Fail-safe behaviour etc.

As the last two items are based on checks using the documentation a new aspect concerning the system documentation becomes important and introduces some requirements concerning the documentation itself in order to make the detecting system testable under this testing philosophy.

- (4) The documentation is checked whether
 - it is in line with the system presented for testing,
 - it gives a unique description of the tasks and specifications including modules and sub-systems,
 - it is clear and understandable,
 - it contains the name, the codelisting, the parameter definitions for all modules,
 - it provides listing of used memory space etc.
- (5) The system is checked according to its operation facilities which have to meet the safety

requirement such as

- signalisation standards,
- accessibility levels,
- easy (e. g. uncoded) input facilities for safety relevant communication to the system, etc.

A first attempt has been made to derive a testing procedure for practical use based on this testing philosophy^[8]. Again it has to be mentioned that it does not represent a complete test and the testing results do not cover all system properties including possible hidden failure sources. But it gives a good over-all indication of the system quality, checks several key specifications and keeps the testing expenses at an acceptable level.

Nevertheless there are some basic drawbacks with the testing procedure discussed above. Software-controlled electronics will enable future fire detection systems to apply far more sophisticated and hopefully more efficient detection algorithms (e. g. multiparameter detectors or conditional detection schemes). Item (1) in the above testing procedure will no longer be applicable to fire detectors using these techniques. It will have to be revised to a considerable amount. Furthermore new communication networks, e. g. ISDN (Integrated Services Digital Network) which will be introduced in the next future by the national authorities will enable properly designed fire detecting systems to transmit their alarm and fault signal of any kind with high speed and flexibility to every suitable remote place. If these future possibilities will be used the fire detecting system will contain transmission and communication subsystems that are matched to these switched public networks. No items in the above testing scheme deal with specifications and technical requirements concerning the new upcoming facility. Specifications and technical requirements taking into account this new technique in combination with automatic fire safety are not known to the author so far. They are not covered by the testing method mentioned above.

In the past the so-called black-box testing philosophy has been proposed for the test of fire safety equipment and especially for automatic fire detection systems. The pure black box approach does not provide any information for the testing facility about the working principle (e. g. smoke or heat detection), the task solution details (e. g. hardware or software), the design methods used or any technical component applied. Black-box testing would be a rigorous way towards the very attractive principle that only the result of the solution for a technical task (reliable fire detection in this case) is important and not the technical details of the solution. It can be shown that this approach cannot be applied in the fire detection case for the following reasons^[1].

- The black-box testing principle for fire detection systems is based upon conditions which are likely to occur in the vicinity of the detector heads in any possible fire case and on the conditions given in the non-fire case. For reliable testing these conditions have to be modelled by test fires representing all possible fire situations and experiments representing all suitable non-fire cases, which is obviously impossible. Or they have to be defined in terms of measurable physical values which the different fire detectors are able to measure (temperature, smoke "density" in terms of an ionization chamber device etc.) The latter requires the knowledge about the detectors' working principles in order to put up suitable experimental arrangements. And this contradicts the black-box principle.
- This testing approach does not take into account failure sources which have nothing to do with the result of the technical solution but with technical details applied or components used. Software solution failure sources, for example, are given by this technical tool and have to be taken into account as such. Black-box testing will not be able to help reducing them.

So, the above mentioned testing procedure also

being a compromise is likely to give suitable results towards more reliable fire detection installations which are equipped with software-controlled electronics.

REFERENCES

1. H. Luck, "Softwaregesteuerte Brandmeldeanlagen-brand schutztechnischer Fortschritt?", ZVers Wiss, Bd. 75 (1986) S.233-246
2. H. Luck, Th. Brockmann, U. Schlossarek, "New Technologies In Automatic Detection And Suppression" in NEW TECHNOLOGY TO REDUCE FIRE LOSSES & COSTS (S. J. Grayson and D. A. Smith ed.), Elsevier, London 1986, p.211-218
3. Th. Brockmann, U. Schlossarek, "Softwaregesteuerte Brandmeldeanlagen", Proc. 7. Intern. Brandschutz-Seminar (7. IBS86) Wien 1986, Bd. II, S.77-87, VFDB
4. E. Unger, "Problems of the Operating Reliability in Control and Indicating Equipment", Fire Safety Journal, 6 (1983), p.241-250
5. H. Luck, "Correlation Filters for Automatic Fire Detection Systems", FIRE SAFETY SCIENCE (C. E. Grant and P. J. Pagui ed.), Proc. of 1st intern. Symp., Springer-Verlag, Berlin 1985, p.748-758
6. J. Harwood, B. Hume, "The Application of Microprocessors to Automatic Fire Detection Systems" see [2]. p.235-242
7. Europ. Workshop on Industrial Computer Systems, "Safety Related Computers", Verlag TÜV Rheinland, Köln 1985.
8. U. Schlossarek, "Softwaregesteuerte Gefahrenmeldeanlagen : Anforderungs-und Prüfkriterien", Universität Duisburg-NT 1987, Herausg. : Verband der Sachversicherer e. V. Köln VdS 2173, 5/87
9. European Standard EN 54 "Components for automatic fire detection systems", Part 7: Point type smoke detectors; detectors using scattered light, transmitted light or ionization; Part 9: Fire sensitivity test; CEN, Brüssel 1982
10. "Directive for testing of software quality in control and surveillance systems", ELEKTRONIKCENTRALEN, Horsholm Denmark 1983
11. PES "Programmable Electronic Systems in Safety Related Applications", British Health and Safety Executive (Library and Information Services, 1 Chepstow Plac Westbourne Grave, London) 1987.