Safety Critical System Designs

CSCM10 Report

Arran Jones

945187

Submitted to Swansea University in fulfilment

of the requirements for the Degree of Master of Science



Department of Computer Science

Swansea University

January 14, 2020

# Abstract

This report will give a brief outline of the project, present a chosen topic of the dissertation in detail and discuss published literature related to the topic.

The chosen topic for this report is the techniques used for industrial and safety critical control system design. In order to discuss these topics, we will first describe a safety critical system and the considerations required when designing such a system. The main body of the report will then describe some techniques used and a look at how these design techniques were used in a design consideration for industry.

# Table of Contents

[Abstract 2](#_Toc32803466)

[Table of Contents 3](#_Toc32803467)

[Chapter 1 Introduction 4](#_Toc32803468)

[1.1 Project outline 4](#_Toc32803469)

[1.2 Overview 4](#_Toc32803470)

[Chapter 2 Safety critical systems 5](#_Toc32803471)

[2.1 What is a safety critical system? 5](#_Toc32803472)

[2.2 Why is a different design system required for safety critical systems? 5](#_Toc32803473)

[Chapter 3 Safety critical system considerations 6](#_Toc32803474)

[3.1 Reliability 6](#_Toc32803475)

[3.2 Safety 6](#_Toc32803476)

[3.3 Single point failures 6](#_Toc32803477)

[Chapter 4 Safety critical system designs 7](#_Toc32803478)

[4.1 Hazard analysis 7](#_Toc32803479)

[4.2 Single channel protected designs 7](#_Toc32803480)

[4.3 Dual channel designs 8](#_Toc32803481)

[4.4 Homogeneous redundancy pattern 8](#_Toc32803482)

[4.5 Diverse redundancy pattern 8](#_Toc32803483)

[4.6 Monitor-actuator pattern 9](#_Toc32803484)

[4.7 Shutdown systems 9](#_Toc32803485)

[4.8 EPICS 10](#_Toc32803486)

[4.8.1 What are EPICS? 10](#_Toc32803487)

[4.8.2 History 10](#_Toc32803488)

[4.8.3 Standards 10](#_Toc32803489)

[4.8.4 Architecture 11](#_Toc32803490)

[4.8.5 EPICS in industry 12](#_Toc32803491)

[Chapter 5 Conclusion 13](#_Toc32803492)

[Bibliography 14](#_Toc32803493)

# Chapter 1 Introduction

## 1.1 Project outline

The dissertation project we have chosen is to design and develop a GUI desktop software for producing GCode like instructions from vector graphics or rasterized image data. These instructions will then move Lego Mindstorm motors to control a 2D pen plotter. This should all be done developed robustly using techniques from industrial and safety critical control system design.

## Overview

In this document we will discuss safety critical systems and the reasons they must be designed differently to other systems. We will discuss some industrial standards for safety critical system design and the reasons why these standards are must be different to standard design systems. Following this, we will discuss some of the techniques used and considerations that must be made in safety critical system design, including a more in-depth section on EPICS. Finally, we will discuss a paper about the use of EPICS in a design for a radiation therapy machine.

# Chapter 2 Safety critical systems

This section will describe what a safety critical system is and why different approaches must be used when designing such systems.

## 2.1 What is a safety critical system?

A safety critical system is a system which can lead to “loss of life, or to significant property or environmental damage” [1] when it fails. Safety critical systems are developed in many different industries, such as transport, factories and military purposes. These different domains may provide certification standards of their own that may be required when developing such systems, as the requirements may vary drastically depending on the kind of system being created.

Typically, safety critical systems are required to address the attributes, reliability, availability, maintainability and safety. These attributes are commonly referred to as RAMS, and to ensure a high level of quality in these attributes many techniques have been created in every area of the development lifecycle.

## 2.2 Why is a different design system required for safety critical systems?

Although safety critical systems are developed using far more scrutinous methods than other systems, it is impossible to unsure a system is free of failures. This is problematic as it is impossible to put a price on safety, but the more time spent on producing a system, the more expensive it gets. Difficulties such as this can lead to problems such as not enough or too much time spent on producing certain parts of a system, where the time could have been spent more effectively elsewhere.

# Chapter 3 Safety critical system considerations

This section will describe some of the considerations that must be made when designing a safety critical system.

## 3.1 Reliability

Reliability in system design is the “measure of the ‘up-time’ or ‘availability’ of a system” [2]. This implies that more reliable systems are ones which have less frequent and smaller failures. The commonly used measurement for reliability is the Mean Time Between Failure, or MTBF. This measurement gives a statistical value for the reliability of a system and provides a probability for failure which can be used to more efficiently deal with failures when they arise. This reduces down-time of systems, increasing its efficiency. Producing methods for increasing MTBF is also extremely important. One such approach for this is to redundancy, this is the process of creating a system that takes the original’s place if a fault should occur. This method is only possible if the two criteria for failure are independent of each other.

## 3.2 Safety

Safety in system design is minimising the risk of an accident which can cause injury or damage. Reliability and safety should be independent of each other, although this is not always possible.

The most common cause of failure in safety for safety critical control systems is a failure in reliability, but this is not always the case. A system can be reliable but also unsafe. This is a common mistake made when designing safety critical designs as the development and design team may put more time into reliability design and forget about key features which may be necessary to ensure the system is safe for operation.

## 3.3 Single point failures

As stated earlier, a system should be safe when no faults are present, but for a system to be considered safe, no single point of failure (a single fault in one component of the software) should cause the safety of the system to be compromised. This is important to consider when designing each component of the system, as this means that each major component should be as independent from the other components as possible. If the components are not independent, then a single fault could cause more problems in other components, leading to a major fault somewhere in the system.

# Chapter 4 Safety critical system designs

In this section we will discuss some of the techniques used to design more reliable and safer systems.

## 4.1 Hazard analysis

Hazards are a condition required for safety to be compromised. If the design of the normally functioning system is complete, all possible hazards of the system should already have been dealt with. Knowing this, a simple stage for reducing the number of hazards a system which has encountered a failure would be a hazard analysis.

A hazard analysis document should be started at the requirements specification stage of a project, periodically reviewed and updated continuously as the project evolves. A hazard analysis document should include identification of the hazard, the causes of the hazard and the way the hazard will be dealt with.

## 4.2 Single channel protected designs

“A channel is a static path of data and control that takes some information and produces some output.” [2]

If a fault appears in any component of the channel, a channel wide failure will occur. Single channel protected designs are a technique which ensure a single channel is used for only a single process. This means that at most only a single process will fail throughout the whole system if only a single fault occurs at any one point in time. In many cases a single channel protected design may be difficult or impossible to implement as the system may require the output of one process to determine the action that should be taken in another component of the system.

## 4.3 Dual channel designs

Dual channel designs are named as such because control and monitoring are separated. In some cases, the control may be separated into several independent channels, but all channels will still be separate to the measurement channel. This is done so that a failure in the control channel can be detected by the monitoring channel and either alert a user, or preferably another system can take over and continue the systems functions correctly. If the fault is in the monitoring channel, then as long as the design was done correctly and the channels were kept independent, the system should continue to function correctly. If this happens, an alert can be sent to a user which can fix the issue (hopefully) before another fault occurs.

## 4.4 Homogeneous redundancy pattern

The homogeneous redundancy pattern is a technique used to reduce faults by using two separate identical channels which simultaneously compute the same values in order to provide redundancy with a small efficiency penalty, at the cost of more computational power. The result of these channels can then be used to decide whether the value is consistent or whether another system must be activated to determine the value that should be used.

This is a simple technique to implement as the two channels are identical, so very little extra design or implementation is required to build the system. This system however does not deal with systematic errors, such as software errors.

## 4.5 Diverse redundancy pattern

The diverse redundancy pattern is similar to the homogeneous redundancy pattern, except it uses multiple redundant channels which “are identical in semantics and interface, but differ in terms of implementation” [2].This has the advantage of the multiple implementations having different less likelihood of a similar systematic error.

This technique provides more protection than a homogenous redundancy pattern, but also requires far more designing and is much more effort to implement as each subsystem must be different to each other.

## 4.6 Monitor-actuator pattern

This technique is a lower cost variation of the diverse redundancy pattern which has a single primary channel which handles the normal control of the system. A monitoring channel then watches the operation and reads the results of the main system. If there is a fault in the main channel, the operation of this channel is brought to a halt and the operation is passed to a subchannel which brings the system to a safe state. As was stated for previous monitoring channels, so long as it is kept independent of other channels, a failure in this channel does not imply a fault elsewhere in the system.

## 4.7 Shutdown systems

The shutdown system is a component of the system dedicated to identifying major hazards and move the system into a “safe state” which in most cases terminates the whole system. This is analogous to the emergency stop button in a factory, except the system detects the hazard itself. This component is independent of every other component in the system but operates simultaneously. In order to do this, the shutdown system will require its own sensors. A subsystem may be used on the shutdown system to ensure the integrity of its decisions.[3] This is important to add redundancy and decreases the chance of the system unnecessarily entering a safe state, but also if there are any doubts the main system should take priority as safety is more important that runtime.

## 4.8 EPICS

### 4.8.1 What are EPICS?

EPICS stands for the “Experimental Physics and Industrial Control System”. EPICS is “an architecture for building scalable control systems” [4] and “a collection of code and documentation comprising of a software toolkit” [4]. Although originally EPICS required a commercial license it is now completely free under the “EPICS Open License”.

EPICS was designed to be with the goals of robustness and easy scalability while maintaining the ability for incremental upgrades. It is currently being used in over 50 large institutions and many commercial companies worldwide.

### 4.8.2 History

In 1988 Bob Dalesio, Jeff Hill and others in their team developed a system called “Ground Test Accelerator Controls System” (GTACS) at Los Alamos National Library. The following year, Marty Kraimer joined the team from Argonne National Library where his previous work on the Advanced Photon Source control system was used to adapt the GTACS system. The resulting system was renamed EPICS and presented in 1991.

### 4.8.3 Standards

The EPICS system “relies on standards at every level” [4]. Achieving this allows for EPICS to be independent of the hardware and allows for incremental upgrades both within and across layers. This also has the advantage of maintaining a inverse proportionality of cost to performance without a software penalty.

### 4.8.4 Architecture

EPICS is a fully distributed system where no central device or software is required at any layer. There are 8 layers required for the EPICS architecture, three physical and five software.

The first physical layer is the front-end layer called the “Input/Output Controller”, which is connected to a back-end layer on a workstation via a network layer.  
 The first software layer is called the “Client layer”. This is usually run in the workstation (back-end) physical layer. This layer provides communication between a user and the system, including components such as control screens and alerts.

The second software layer is called the “Channel Access layer” and its purpose is to connect clients to servers; this is done while hiding the network from both ends. All data is sent with timestamps, this allows any user to access the correct data which will also carry validation information to help ensure data is not lost or modified during transmission. Channels in EPICS are not as described earlier for other design techniques of safety critical systems, they are paths to attributes of a record in the database.

The third software layer is called the “Server layer”. The main server in this layer is the channel access server. This runs of the CPU in every Input/Output Controller (front-end physical layer). It’s core purpose it to prevent direct access between the client layer and the database layer.

The fourth software layer is the “Database layer”. This layer is the core component of an EPICS system. This layer uses host tools to describe the database as records. These records can perform many tasks and can take around 50 different types. These records are how the channel access layer communicates between the client and the server.

The fifth software layer is the “Device Driver layer”. This is a large collection of drivers written in C, for many different hardware devices, which have been collected and added to by many EPICS partners.

These layers have created a very modular and robust system which is continuing to be adopted by industries and is also continuing to evolve. Another defining factor why EPICS is achieving popularity is due to its good documentation and support on their website.[6]

### 4.8.5 EPICS in industry

One case found of EPICS being used is in a paper by Jonathan Jacky titled “EPICS-BASED CONTROL SYSTEM FOR A RADIATION THERAPY MACHINE” [7], which comments on their use of EPICS for non-therapy functions of their cyclotron and how they wish to try implementing EPICS into their therapy control too.

Later in the paper, they describe the safety analysis of the system, and comment that almost 30 years of operations have never ended in a mishap. They also comment on how they would like the new system to eliminate some hazards but removes other hazards from the system’s control, such as a safe state where the beam and motions can be manually turned off. These considerations are important for maintaining safety and preventing harm to patients and practitioners which may be using the machine.

At the time of writing the paper the project had not concluded, and hence we can only draw from their initial conclusions. According to the conclusion, the system seems as trustworthy as their previous platform, despite their fears, while also providing some advantages. Although it is stated that further testing will be required.

# Chapter 5 Conclusion

In this paper we have discussed some of the more common techniques in critical system design. It is worth noting that every safety critical system is different and may have very different requirements. This means that the techniques required to develop some systems compared to others are vastly different and there is no one system that suits every situation. In most cases, implementing more techniques will improve safety and reliability; but this isn’t always possible as cost and computation time may be an issue. Hence, more extensive testing or more time refining the design may be a more favourable approach. Factors such as these should not reduce the safety of a system, and a safe state should always be implemented in case of a catastrophic failure.

# 

# Bibliography

[1] Pietrantuono, Roberto & Russo, Stefano. (2013). Introduction to Safety Critical Systems. 10.1007/978-88-470-2772-5\_2.

[2] Douglass, Bruce. (1998). Safety-Critical Systems Design.

[3] Architecture of safety-critical systems (2005). https://www.embedded.com/architecture-of-safety-critical-systems/

[4] Lewis, S.A. (2000). Overview of the Experimental Physics and Industrial Control System: EPICS.

[5] EPICS (2020). https://en.wikipedia.org/wiki/EPICS

[6] EPICS. https://epics-controls.org/

[7] Jacky, J. (2013) October. EPICS-based control system for a radiation therapy machine. In *International Conference on Accelerator and Large Experimental Physics Control Systems*