Safety-critical Control in Mixed Criticality Embedded Systems

Master Thesis
Royal Institute of Technology
Stockholm, Sweden

Emil Hjelm
emilhje@kth.se

January 27, 2017

Examensarbete MMK2017:Z MDAZZZ Safety-critical control in mixed criticality embedded systems Emil Hjelm		
Approved: (datum)	Examiner: (examinator)	Supervisor: (handledare)
	Uppdragsgivare: Alten	Kontaktperson: Detlef Scholle

Abstract

Modern automotive systems contain a large number of Electronic Control Units, each controlling a specific system of a specific criticality level. To increase computational efficiency it is desired to combine multiple applications into fewer ECUs, this leads to mixed criticality embedded systems. The assurance of safety critical applications not being affected by non-critical applications on the same system is crucial.

Examensarbete MMK2017:Z MDAZZZ Säkerhetskritisk kontroll i blandkritiska inbyggda system Emil Hjelm		
Godkänt: (datum)	Examinator: (examinator)	Handledare: (handledare)
	Uppdragsgivare: Alten	Kontaktperson: Detlef Scholle

Sammanfattning

Denna del kommer att inehålla en sammanfattning av arbetet på svenska.

Preface

Credit where credit is due.

Emil Hjelm
Stockholm

Contents

Preface **iii**

Abbreviations **vii**

1 Introduction **1**
1.1 Background 1
1.1.1 Definition of safety-critical systems 2
1.1.2 Different levels of criticality 2
1.1.3 EMC^2 Mixed criticality embedded system 3
1.1.4 Platooning 4
1.2 Problem statement 4
1.3 Purpose 4
1.4 Goals 5
1.4.1 Team goal 5
1.4.2 Individual goal 5
1.4.3 Scope 5
1.5 Research design 6

2 State of the art **7**

3 System design **8**

4 Implementation **9**

5 Results **10**

6 Discussion **11**

7 Future work **12**

List of Figures

1.1 System overview of the MCS in place. [13] 3

List of Tables

Abbreviations

Abbreviation	Description
ECU	Electronic Control Unit
MCS	Mixed Criticality System
EMC^2	Embedded Multi-Core systems for Mixed Criticality applications in dynamic and changeable real-time environments
RTOS	Real-Time Operating System
SIL	Safety Integrity Level
ASIL	Automotive Safety Integrity Level
DAL	Development Assurance Level

Chapter 1

Introduction

This chapter will introduce the subject of mixed criticality embedded systems and the EU project "EMC^2" to the reader.

1.1 Background

Today, modern automotive systems contain a large number of Electronic Control Units (ECU)s, each controlling a subsystem of a specific criticality level such as safety-critical distance keeping system for platooning (see 1.1.4) or non-critical entertainment systems [3]. Having the ECUs isolated ensures that the numerous critical and non-critical applications do not interfere with each other, thus it is a simple task to certify an individual ECU. However, this approach leads to an inefficient use of system resources and expensive system implementation [2]. In order to lower the cost of the collective system and increase system efficiency (utilization), applications of different criticality levels can be integrated into a single multicore platform, leading to a Mixed Criticality System (MCS). However, this approach increases system complexity, and hinders the certification of safety-critical systems [13]. In order to facilitate the design, test, and certification of such systems, spatial and temporal partitioning can be used in the architecture of the system as described by [13].

Protecting the integrity of a component from the faults of another is desired in all systems hosting multiple applications. However, it is of higher significance if the different applications have different criticality levels. Without such protection all components on the same system would need to be engineered to the standards of the highest criticality level, potentially massively increasing development costs [2].

The EU project "Embedded Multi-Core systems for Mixed Criticality applications in dynamic and changeable real-time environments" (EMC^2) was created in order to "find solutions for dynamic adaptability in open systems, provide handling of mixed criticality applications under real-time conditions, scalability and utmost flexibility, full scale deployment and management of integrated tool chains, through the entire lifecycle" [4].

1.1.1 Definition of safety-critical systems

The term "safety-critical system" has many definitions, most quite similar. Most definitions relate to systems with the potential to harm humans if the system malfunctions. According to [8] it is defined as "A system in which any failure or design error has the potential to lead to loss of life." Further, [6] defines safety-critical systems as "A computer, electronic or electromechanical system whose failure may cause injury or death to human beings." A Wikipedia article, [12], defines a safety-critical system (or "life-critical system") as a system whose failure or malfunction may result in one (or more) of the following outcomes:

- death or serious injury to people
- loss or severe damage to equipment/property
- environmental harm

In this thesis, a safety-critical system will be defined as "a system whose failure may cause injury or death to human beings."

1.1.2 Different levels of criticality

Different names of levels of criticality are typically Safety Integrity Level (SIL), Automotive Safety Integrity Level (ASIL) and Development Assurance Level (DAL). The IEC 61508 standard [9] defines four different levels and the ISO 26262 standard [10] and the DO-178C standard [7] define five different levels each. These levels range from low to no hazard up to life-threatening or fatal in the event of a malfunction requiring the highest level of assurance that the dependent safety goals are sufficient and have been achieved.

In this report the number of criticality levels will be restricted to two: "safety-critical" and "non-critical". This is due to the constraints presented in section 1.1.3 and 1.4.3.

1.1.3 EMC² Mixed criticality embedded system

The current MCS is implemented on a Xilinx Zynq-7000. It employs two operating systems to handle applications of different criticality. A Linux General Purpose Operating System (GPOS) for non-safety critical applications and a Real-Time Operating System (RTOS) for safety-critical applications. An overview of the system can be seen in Figure 1.1.

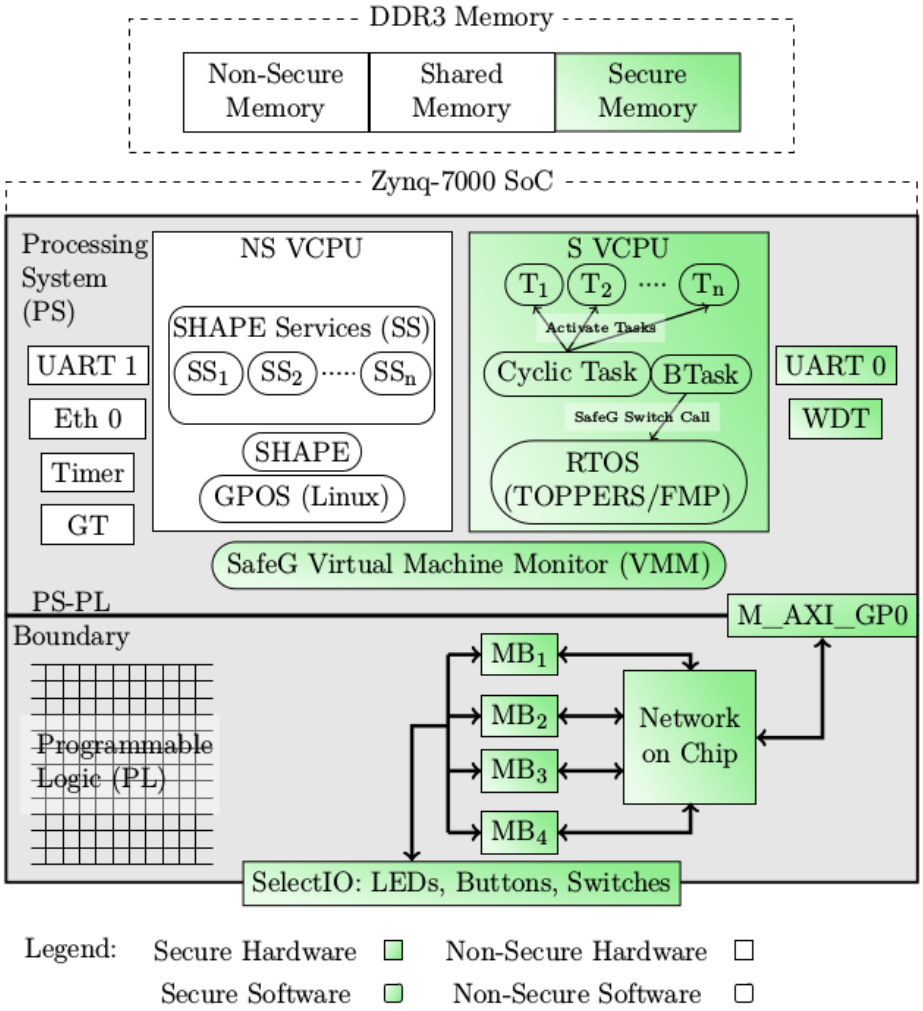

Figure 1.1: System overview of the MCS in place. [13]

For more detailed information, see the report by [13].

1.1.4 Platooning

"The platooning concept can be defined as a collection of vehicles that travel together, actively coordinated in formation. Some expected advantages of platooning include increased fuel and traffic efficiency, safety and driver comfort" [1].

1.2 Problem statement

A distance keeping control algorithm for platooning will be implemented on the embedded system described in 1.1.3. A demonstrator will be constructed in the form of a RC car capable of following a vehicle in front of it at a certain distance. It should be verified that no matter the computational load and eventual crashes of the Linux based non-critical system, the distance keeping algorithm on the RTOS never crashes.

This problem leads to the research question:

- How well can a safety-critical control system perform when implemented on a mixed criticality system using virtualization?

alternatively:

- How, in a disciplined way, to reconcile the conflicting requirements of partitioning for safety assurance and sharing for efficient resource usage? [2]

alternatively:

- Is virtualization an efficient approach when trying to reconcile the conflicting requirements of partitioning for safety assurance and sharing for efficient resource usage when implementing a safety-critical control system?

1.3 Purpose

Reducing the amount of computers in automotive systems would have many effects. Manufacturing costs would decrease and with fewer physical components maintenance costs would also decrease. However, the system complexity would increase and thereby increasing time and cost to design the system. To combat this the EMC^2 project aims at creating platforms for easier development of MCS.
The EMC^2 project lists several goals [11]:

- Reduce the cost of the system design by 15%
- Reduce the effort and time required for re-validation and re-certification of systems after making changes by 15%
- Manage a complexity increase of 25% with 10% effort reduction
- Achieve cross-sectorial reusability of Embedded Systems devices and architecture platforms that will be developed using the ARTEMIS JU results.

1.4 Goals

In this project there are both team goals and individual goals that do not always necessarily align with each other.

1.4.1 Team goal

The team consists of five master thesis students. The students areas of work are: control theory and system modeling, data aggregation, safety-critical communication in MCS, line following and system testing and finally safety-critical control in MCS. Together the team will build a vehicle capable of following a vehicle ahead of it while keeping inside road markers.

1.4.2 Individual goal

Verify quantitatively the performance of safety-critical distance keeping controller. Evaluate different task scheduling methods in the specific system. Solve the problems described in 1.2.

1.4.3 Scope

The work of this thesis and the implementation on the demonstrator will build upon the work of [13].
The embedded computer is constrained to the Xilinx Zynq-7000 [1].
The thesis is produced at Alten AB.

[1] `https://www.xilinx.com/products/silicon-devices/soc/zynq-7000.html`

1.5 Research design

The plan is to make an confirmatory investigation using qualitative data/operations. This is a mixed methods approach where the data is gathered using simulations of the environment [5].

The position of the demonstrator will be read by a separate sensor of the same type as the one on the demonstrator. The performance of the control system and the embedded controller will be measured and compared with the same system without any non-critical computational load. This will be done for several scheduling methods, as well as for a simulation of the same system. The measures regarding control system performance will consist of

- Response time
- Overshoot
- Settling time

and the measures regarding embedded controller performance will consist of

- Missed deadlines
- CPU utilization

Chapter 2

State of the art

This chapter will go through relevant articles and already known knowledge on the subject.

Chapter 3

System design

This chapter will derive the design of the controller to be implemented.

Chapter 4

Implementation

This chapter will describe the implementation of the control system in the demonstrator.

Chapter 5

Results

This chapter will present results from the demonstrator to the reader.

Chapter 6

Discussion

Discussion about the results produced by the thesis.

Chapter 7

Future work

This chapter will contain thoughts and ideas for future work building on this thesis.
Facilitate for more than two different criticality levels.

Bibliography

[1] Carl Bergenhem, Henrik Pettersson, Erik Coelingh, Cristofer Englund, Steven Shladover, and Sadayuki Tsugawa. Overview of platooning systems. In *Proceedings of the 19th ITS World Congress*, Vienna, Austria, October 2012.

[2] Alan Burns and Robert I. Davis. Mixed criticality systems - a review. Available online: `https://www-users.cs.york.ac.uk/burns/review.pdf`, July 2016.

[3] EMC2 Consortium. Emc2 - ll1 - automotive applications, January 2017. `http://www.artemis-emc2.eu/?id=23`.

[4] EMC2 Consortium. Emc2 public webpage, January 2017. `https://www.artemis-emc2.eu`.

[5] John W. Creswell. *Research design: Qualitative, quantitative, and mixed methods approaches*. SAGE Publications, Inc., 2009.

[6] Dictionary.com. safety-critical system, January 2017. `http://www.dictionary.com/browse/safety-critical-system`.

[7] Software considerations in airborne systems and equipment certification. Standard, Radio Technical Commission for Aeronautics, Washington, DC, USA, December 2011.

[8] Encyclopedia.com. safety-critical system, January 2017. `http://www.encyclopedia.com/computing/dictionaries-thesauruses-pictures-and-press-releases/safety-critical-system`.

[9] Functional safety of electrical/electronic/programmable electronic safety-related systems – parts 1 to 7. Standard, International Electrotechnical Commission, Geneva, CH, April 2010.

[10] Road vehicles – functional safety – part 9: Automotive safety integrity level (asil)-oriented and safety-oriented analyses. Standard, International Organization for Standardization, Geneva, CH, November 2011.

[11] Werner Weber. A platform project on embedded microcontrollers in applications of mobility, industry and the internet of things, Mars 2015. https://artemis-ia.eu/publication/download/1131.pdf.

[12] Wikipedia.com. Life-critical system, January 2017. https://en.wikipedia.org/wiki/Life-critical_system.

[13] Youssef Zaki. An embedded multi-core platform for mixed-criticality systems: Study and analysis of virtualization techniques. Master's thesis, KTH, School of Information and Communication Technology (ICT), 2016.