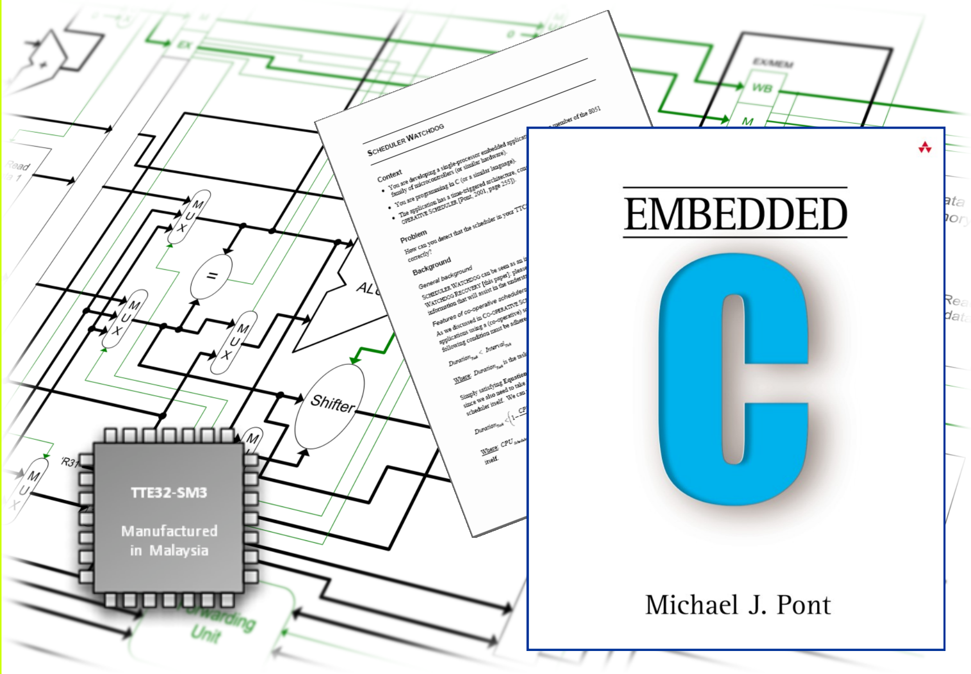
A new 5-day training course which explores design and implementation options for reliable embedded systems

Delivered in Leicester, 18 to 22 October 2010



[MSc Reliable Embedded Systems, Module EG7510 / A1]

Programming techniques for reliable embedded systems

TTE Systems Ltd

Cars, aircraft, medical equipment through to fridges, TVs, MP3 players and even (wet) razors are all good examples of modern systems which contain embedded processors. Overall, it is usually estimated that for every desktop computer chip sold, 100 processors are sold for embedded systems.

Using a practical mixture of informal seminars and “hands on” training sessions, this 5-day course has been designed to bring you up to date with the different design and implementation options which are now available for developers who wish to create **reliable** embedded systems.

This course can be take on a “stand alone” basis, or used as the first step towards a full MSc in Reliable Embedded Systems from the University of Leicester (UK).

TTE Systems

Programming techniques for reliable embedded systems

Introduction

Cars, aircraft, medical equipment through to fridges, TVs, MP3 players and even (wet) razors are all good examples of modern systems which contain embedded processors. Overall, it is usually estimated that for every desktop computer chip sold, 100 processors are sold for embedded systems.

Techniques for creating high-reliability embedded systems have focussed historically on safety-critical markets (e.g. the aerospace, medical and automotive industries), where system failures can have fatal consequences. These markets remain important, but embedded processors now also have an enormous impact in much broader areas of product development, including relatively simple consumer applications such as washing machines and set-top boxes. Manufacturers now wish to maximise the reliability of all such systems in order to reduce the cost of warranty repairs, minimise product recalls and ensure repeat orders.

Using a practical mixture of informal seminars and “hands on” training sessions, this 5-day course has been designed to bring you up to date with the different design and implementation options which are now available for developers who wish to create reliable embedded systems.

This course can be take on a “stand alone” basis, or used as the first step towards a full MSc in Reliable Embedded Systems from the University of Leicester (UK).

Pre-requisites

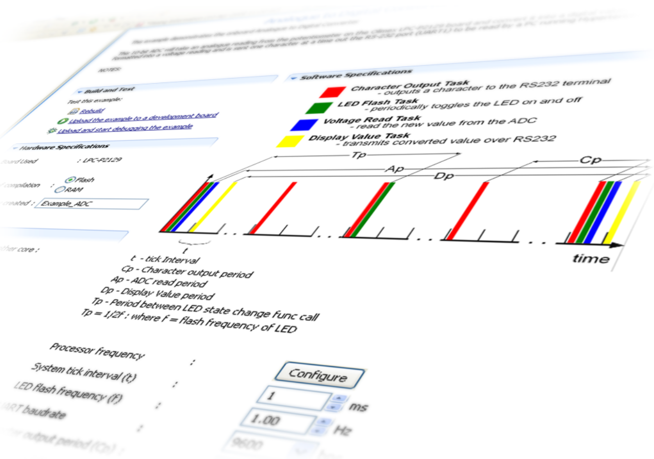
While some previous experience with embedded systems would be useful, this is not an essential pre-requisite for this exploratory course.

Some previous programming experience in a high-level language (such as C, C++ or Java) is assumed at the start of the course. If you have not had this experience, suggestions for background reading can be provided.

Course objectives

After attending this course, participants should:

- Understand the need for reliability in modern embedded systems;
- Have a good knowledge of the three key hardware platforms used to create modern embedded systems: microcontrollers, field-programmable gate arrays and “embedded PCs”;
- Be able to write high-quality software for reliable embedded systems;
- Understand how to test and debug existing embedded systems;
- Understand the key concepts of “real-time” systems and the importance of timing behaviour when creating reliable embedded systems;
- Understand how to use MISRA C when creating reliable embedded systems;
- Understand how “design patterns” can be used to support the creation of reliable embedded systems.



Programming techniques for reliable embedded systems

Day 1: An introduction to "Embedded C"

On Day 1, the world of embedded systems is introduced using a series of short seminars interleaved with practical sessions which are designed to reinforce the key concepts. C is the standard programming language for embedded systems: working with a processor simulator and an 8-bit target, you will learn how to create your first programs in "Embedded C". Starting by flashing an LED, you will soon move on to learn how to create reliable code for working with switches and taking control of the real world. We'll end the session by discussing the use of the MISRA C programming guidelines and the ways in which the use of this "safe subset" of the C language can assist in the creation of reliable systems (there's a lot to take in: you'll be provided with your own printed copy of these guidelines to review at your leisure).

Day 2: Real-time "Embedded C"

On Day 2, we move on to look at the need for simple operating systems, and the concepts of "real time" constraints. We consider and discuss concepts such as worst-case execution time, task jitter and time-out mechanisms. By the end of the day, you'll have created and tested a complete set of code for your first realistic embedded systems.

Day 3: Exploring modern hardware platforms

In the first two days of this course, we focus on software development and you'll work with a simple processor simulator to test your code. On Day 3, we move into the 21st century: you'll by start creating code for a modern (32-bit) microcontroller with an ARM® processor core. After you've got your first system running, you'll learn about debugging (including the use of JTAG) and timing analysis.

Modern embedded systems use a wide range of platforms. To round off Day 3, we'll look at two further targets.

- In recent years, the Intel® Atom processor has become the basis of many "embedded PCs". We'll consider some of the challenges involved in creating reliable systems based on this type of platform. As you'll learn, your embedded PC will run happily without a "conventional" operating system (we won't be using Windows® or Linux: instead we'll be programming "at the bare metal" for maximum predictability and performance).
- We'll then go on to look at "Field programmable gate arrays" (FPGAs). FPGAs are already a popular implementation platform for developers of embedded systems (and some people believe that they will be the platform of choice for the majority of new systems within the next few years). We'll introduce you to some of the challenges and opportunities offered by this exciting new platform. As you'll see, FPGAs provide enormous flexibility, allowing you to create a custom processor target for every project.

Day 4: Working with multiple tasks

In Day 4, we begin to look at some more advanced topics, including issues associated with task design (including task pre-emption). We'll start by considering the popular rate-monotonic (RM) and earliest deadline first (EDF) task scheduling algorithms for single-processor systems. We'll go on to discuss some key design challenges for systems which involve task pre-emption and shared resources, including ways of avoiding "priority inversion".

Day 5: Working with distributed systems

On Day 5, we'll begin to consider some of the challenges involved in creating distributed embedded systems: that is, systems involving multiple processors connected in some form of network using (for example) a bus or star topology. We'll consider key design issues, including ways in which we can synchronise the timing of tasks that are running on different nodes. Our focus will be on systems which employ the popular Controller Area Network (CAN) protocol.

Learning goals

- To learn how to write reliable programs in “embedded C”;
- To learn how to create reliable, power-efficient and cost-effective embedded systems using microcontrollers;
- To learn how to use x86 processors (“embedded PCs) for prototyping and for low-volume production of very powerful embedded systems;
- To learn how to create very flexible embedded systems using field-programmable gate arrays (FPGAs).

Methodology

This course is taught through a carefully-planned combination of seminars and practical (laboratory) classes. Active involvement of all participants will be expected in both seminars and lab sessions. Problems will be set during seminars and in laboratory sessions. Case studies will be used extensively in the laboratory sessions.

Who should attend

The course is intended for **anyone** who is interested in working in the field of embedded systems. No previous experience of embedded systems is required, but some previous programming experience will be useful. The course is also an ideal “taster module” for the MSc in Reliable Embedded Systems.

Trainer biodata

Prof. Michael J. Pont holds a BSc (Electrical and Electronic Engineering) from the University of Glasgow and a PhD (Computer Science) from the University of Southampton.

Michael is Professor of Embedded Systems and Head of the Embedded Systems Laboratory at the University of Leicester: he is also CEO of TTE Systems Ltd.

Michael is author / co-author of more than 100 technical publications and author of three books (“Patterns for Time-Triggered Embedded Systems”, “Embedded C” and “Software Engineering with C++ and CASE Tools”).

Duration

5 days, 09.30 to 18.30

Price

£500 + VAT

Introductory price
(Please contact us for details)

Eligible for conversion to MSc
(please contact us for details)

Dates

18 to 22 October 2010

Venue

Leicester, UK

Contact

Dr Anjali Das
(a.das@tte-systems.com)