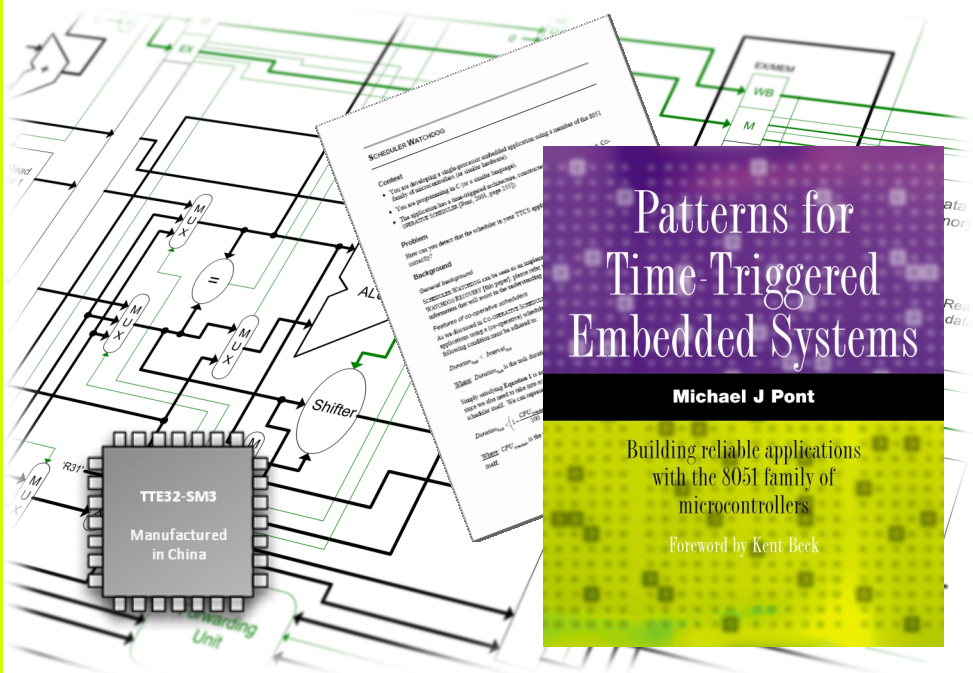


A 5-day training course, delivered to your staff,  
on your company site, on dates of your choosing



# Design patterns for reliable embedded systems

TTE Systems Ltd

First introduced in the 1990s, design patterns have become an established tool for companies that wish to improve the quality of the products they produce.

**Put simply, design patterns allow organisations to capture the expertise of their best designers and make this available to other development teams within their organisation.**

To obtain maximum benefit from this approach, companies aim to produce a “pattern language”: such a language is produced by a well-proven series of steps involving “mining”, “shepherding” and “workshopping”.

**By the end of this 5-day course, participants will be in a position to create their first complete pattern language for reliable embedded systems.**

**TTE** Systems

# Design patterns 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. This course considers ways in which **design patterns** can be used to help achieve this goal. First introduced in the 1990s, design patterns have become an important way for many companies to improve the quality of the products they produce. **Put simply, design patterns allow organisations to capture the expertise of their best designers and make this available to other development teams within their organisation.**

To obtain maximum benefit from this approach, companies aim to produce a “pattern language”: such a language is produced by a well-proven series of steps involving “mining”, “shepherding” and “workshopping”.

By the end of this 5-day course, participants will be in a position to create their first complete pattern language for reliable embedded systems.

After the course, you will have up to **two months** to register for the University of Leicester’s new part-time **MSc in Reliable Embedded Systems** if you wish to do so (subject to submitting a completed application and receiving the offer of a place on the programme from the University). **If you join the MSc programme during this period, the fee for your short course will be refunded in full.**

## Pre-requisites

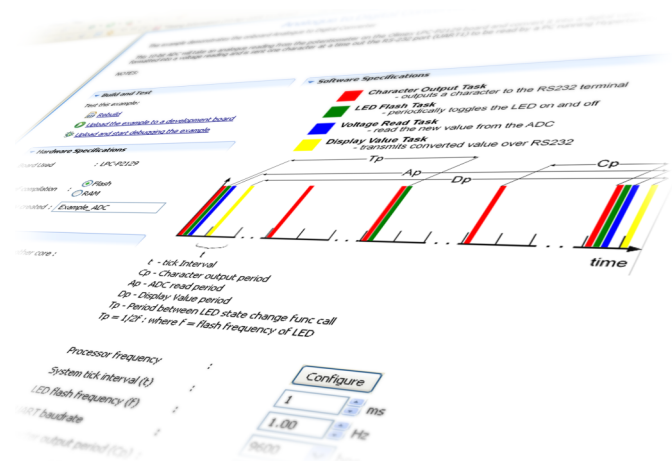
No prior knowledge of design patterns is assumed.

A prior knowledge of a high-level programming language (such as C, C++ or Java) is assumed.

## Course objectives

After attending this course, participants should:

- Understand what a design pattern is
- Understand how to “mine” for design patterns in their organisation
- Understand how to act as a “shepherd” as patterns are prepared for a workshop
- Understand how to run a pattern workshop
- Understand how to create a complete “pattern language”



## Course contents

### 1. Introduction and motivation

This seminar introduces the course. To illustrate how design patterns can be used to support the rapid development of reliable embedded systems, the origins and evolution of the “PTTES” collection is discussed. Key concepts are introduced.

### 2. The work of Christopher Alexander

This seminar focuses on the landmark work of Christopher Alexander who introduced the concept of patterns. Examples of Alexander’s patterns are presented and discussed.

### 3. Patterns everywhere

Seminar 3 introduces the concept of software design patterns that has evolved from Alexander’s original ideas. The influential work of the ‘Gang of Four’ (GoF) is discussed. Various pattern ‘forms’ (GoF, POSA, etc.) are presented and the main elements of a pattern (problem, solution, context, forces, etc) are examined. Some examples of patterns that have been successfully applied to various other fields such as teaching and organisational management are also presented.

### 4. Patterns for fault-tolerant software

Seminar 4 considers patterns which have been developed to support the design and implementation of fault-tolerant systems. The focus is on the Hammer pattern collection.

### 5. Second-generation pattern languages for embedded systems

Seminar 5 considers the challenges of creating software patterns for embedded systems which (a) contain useful low-level details, and (b) can be adapted for use with different hardware targets. The seminar discusses how second-generation pattern languages have been developed which address these issues, and discusses practical techniques for creating such languages.

### 6. Identifying and documenting good patterns

In Seminar 6, John Vlissides’ seven habits of successful pattern writers are discussed. The seminar also explores the process of “mining” for patterns in an organisation.

### 7. Refining your patterns: The “Shepherding” process

Seminar 7 considers how patterns are ‘shepherded’, in preparation for a pattern workshop. The seminar considers the skills required to act as a good shepherd and the shepherding process. A pattern language for shepherding is also presented

### 8. Refining your patterns: Running a “Writers’ Workshop”

Seminar 8 demonstrates how to run a pattern workshop. Tips for holding a successful pattern workshop will be presented. A pattern language for writers’ workshops is also discussed.

### 9. Assessing the effectiveness of your pattern language

There have been many “silver bullets” in the software world, techniques which have “guaranteed” that your future projects will always be completed on time and on budget. Instead of making rash claims, Seminar 9 explores cost-effective and practical ways in which you can obtain **real data** to determine whether **your** pattern language will be effective in **your** organisation.

### 10. The future of “pattern-based software engineering” for embedded systems

Seminar 10 considers what Kurian has called “pattern-based software engineering” (PBSE). Tool support for PBSE is considered. In addition, possible applications of patterns in support of the **maintenance** of embedded systems is considered. We wrap up the course by reviewing the strengths and weaknesses of pattern-based design approaches for the developers of reliable embedded systems.

## Learning goals

- To learn how to create design patterns for use in reliable embedded systems
- To learn how to act as a pattern shepherd
- To learn how to run a pattern workshop

## 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 engineers working in the area of embedded systems who wish to improve the reliability of the systems which they create. The course is also suitable for technical managers who wish to ensure that their teams produce systems which are reliable.

## Trainer biodata

**Dr Anjali Das** holds a BSc (Electronics) and an MSc (Electronics Science) from the University of Delhi, India, an MSc (Information and Communications Engineering) from the University of Leicester, UK, and a PhD (Communications) from the University of Kent, UK.

Having worked previously as a Lecturer at the University of Delhi, Anjali is now Technical Training Manager at TTE Systems Ltd.

**Dr 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").

Michael's book (PTTES) on patterns for embedded systems is considered to be one of the most complete pattern languages yet published. Michael's more recent work on patterns has been selected for the Master Track at the EuroPloP pattern conference. Michael also serves on the Editorial Board for the Springer-Verlag journal "Transactions on Pattern Languages of Programming".

## Duration

5 days, typically 08.00 to 18.00

## Price guide

£15,000 for 10 people, including  
All course notes, books and software licences.

**Eligible for conversion to MSc**

## Dates

Dates of your choosing

## Venue

On your site

## Contact

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

