

# QuEST

## Quality Embedded Software Techniques

By

Eur. Ing. **Chris Hills** BSc, C. Eng., MIEE, FRGS

### **Part 0 The Introduction**

V0.9 21 December 2001

<http://quest.phaedsys.org/>



This paper (and subsequent versions) with any connected papers, source code and power point slides will be available at

<http://quest.phaedsys.org/>  
[QuEST@phaedsys.org](mailto:QuEST@phaedsys.org)

This paper will be developed further.

**The ART in Embedded Engineering  
comes through good  
Engineering discipline.**



## Contents

|     |   |    |
|-----|---|----|
| 0   | Introduction .....  | 7  |
| 1   | The Papers:- .....  | 9  |
| 1.1 | QuEST 1 Embedded C: Traps and Pitfalls .....                                    | 9  |
| 1.2 | QuEST 2 Debuggers.....  | 9  |
| 1.3 | QuEST 3 Advanced debug and test .....   | 9  |
| 1.4 | QuEST 4 C51 Primer In Development.....  | 9  |
| 2   | The Guides .....  | 11 |
| 2.1 | SCIL-Level.....   | 11 |
| 2.2 | The Tile Hill Embedded C Style guide (currently in 2 <sup>nd</sup> Draft) ..... | 11 |
| 2.3 | QuEST-C. Not available- In Development. ....                                    | 11 |
| 2.4 | QuEST Compliance Matrix .....   | 11 |



# Quality Embedded Software Techniques

## 0 Introduction

The **QuEST** is a series of papers based around the theme of **Quality Embedded Software Techniques**. It is not for a specific industry or specific type of work but for all embedded C. It is usually faster, more efficient and surprisingly a lot more fun when things work well.

This series will show you how to minimise the errors, the bugs and tiresome parts of software engineering (testing) and maximise the fun parts. With the right approach even testing becomes an interesting and fun challenge.

The **QuEST** series is aimed at the "smaller" end of the market in most senses. The 8/16 market and the smaller companies and sub contractors. The larger companies usually have money for tools and procedures in place. The smaller companies in the current economic climate usually have fewer tools and smaller budgets. For these companies time really is money. So if you can complete the job faster, with fewer bugs, at a higher quality it is more money in the pocket.

The **QuEST** series came about after I spent many<sup>1</sup> years in electronics, embedded software (8 to 32 bit), comms sw and hardware, specialist electronic production, avionics, installing ISO9000 a couple of times, joining the ISO-C and MISRA-C working groups and had a spell doing technical support. Being a member of ACCU also helped<sup>2</sup>.

The technical support role was an eye-opener! Many times the same basic questions were asked. The same myths were repeated to me many times over. It was this that initially prompted me to write one of the papers in this series and present it at a conference to try and dispel some of the myths and demonstrate software is Engineering.. The response to this paper along with re-writing and updating the Debuggers paper prompted this series as a homogeneous set.

The documents in this, expanding, series should lay the framework that will enable Engineers to do what they like best: Design and write code.

It can be done! I have worked on projects that were completed on time, in budget with no over time. I have also heard of several others. I looked for common themes. I also looked at the research and statistics available. These are usually of little use to working Engineers but can show the trends of what works and what doesn't.... if you can get past the hype and the management speak

By the way, I am an Engineer, a working one at that! Not a consultant as I fail the basic requirement to be a consultant... I can't play golf!

The series is designed to start at the point where the engineer has designed his system and the hardware it is to run on. Design methodologies are well outside the scope of this series (at present) and the hardware is usually covered by many specifications and standards. E.G. EMC, electrical, telephone etc and often industry specific requirements.

---

<sup>1</sup> "many" I worked it out and it is over 2 decades so I decided "many" as it was less depressing!

<sup>2</sup> I can trace it back even further to the influence of my father who designed aircraft engines in the 60's using computers when the computer's air conditioning plant had a building to itself and was the same size as the computer it cooled.

The whole thing seems to fall down when the software is required. This is where Quest comes in. Actually just before you start to write the code.

It looks at coding standards and coding style, code construction and all the usual traps and pitfalls of [embedded] e development. The "Embedded" is the important part as it does differ markedly from "normal" or desktop/mainframe development the series then follows on to the basics of debugging embedded software and systems before tackling the advanced testing and debugging methods.

The author reserves the right to make typos and mistakes. However I would be grateful if anyone finding any errors, typos, mistakes etc would let me know. I will endeavour to correct them. Also any new ideas and additions would be welcome. Full credit will be given.

Chris Hills  
21 December 2001  
<mailto:quest@phaedsys.org>

## **1 The Papers:-**

There follows short description of each paper. They were not originally written in the order shown. In fact one predates the series by several years! They are becoming more of a homogeneous set with each revision. The papers are informative and descriptive they are not designed to be a method or a set of rules. The Guides in the following section are designed along those lines.

### **1.1 QuEST 1 Embedded C: Traps and Pitfalls**

A look a basic embedded software development. How to apply Engineering discipline to a mess so that Engineers can let their art show through. It looks at some of the common myths, and removed some of the prejudices

[Web-Link](#)

### **1.2 QuEST 2 Debuggers**

Debuggers and myths everywhere but which type of debugger and which method is right for what application? Step by step through the history and myths monitors, simulators and ICE of all types to the real answers of what to use when. Also the more important: "why?." This should save you both time and money.

[Web-Link](#)

### **1.3 QuEST 3 Advanced debug and test .**

**In Development:- This paper will be presented at the Embedded Systems Show, Exel London in May 2002**

The Art and Engineering of testing. Why and how to reduce the bug hunting time, the frustration and increase the time spent out of the office. Automated unit test, system test, code coverage, timing and regression testing all in hard real time .

[Web-Link](#)

### **1.4 QuEST 4 C51 Primer In Development**

The starting point for Engineers developing on the 8051 family written by engineers with real experience. This paper is currently at about 140+ pages and will have circuits and software source available. It will be in first draft availability by the end of January 2002

[Web-Link](#)



## 2 **The Guides**

The guides are the part of **QuEST** that can be used as part of a project

### 2.1 **SCIL-Level**

A Simple guide to what type of tools to use at what level of project. Also what level does your project fall into? This is suitable for small companies and individual developers not already working to a rigorous system of industry required standard.

Click Link:- [Local-link-paper](#) [Local-link-slides](#) [Web-Link](#)

### 2.2 **The Tile Hill Embedded C Style guide** (currently in 2<sup>nd</sup> Draft)

The general-purpose style guide for embedded C. This guide links with the Scil level and the Embedded C Traps and Pitfalls paper. It was written after many requests from readers of the Embedded C Traps and Pitfalls paper because all the style guides people could find were either for Unix or 10 years out of date.

[Web-Link](#)

### 2.3 **QuEST-C**. Not available- In Development.

This will be a set of clear and simple rules for a sub set of the C-Language for safe use of C in general embedded systems. This will compliment the Style Guide.

[Web-Link](#)

### 2.4 **QuEST Compliance Matrix**

A compliance- Matrix for MISRA-C This is a template for a compliance matrix as described in section 5.3.2 of the MISRA-C guide.



[quest@phaedsys.org](mailto:quest@phaedsys.org)  
<http://quest.phaedsys.org/>