

# 目标机环境下的嵌入式软件单元测试探讨

刘少雷

陕西烽火电子股份有限公司 陕西宝鸡 721006

**摘要:** 在嵌入式系统的开发中, 由于环境的不同, 宿主所进行的单元测试脚本与目标机器的结果不相符, 从而导致了测试的精确性。通过对某一特定的工程实例, 提出了一种基于对象的单元测试环境, 可以有效地解决这一问题, 提高单元测试的可靠性。在宿主环境下进行单元测试是非常方便的, 因此, 在软件开发的时候, 大部分的单元测试都是在这样的环境中完成的。不过, 因为使用了主机执行单元测试所使用的编译连接环境与实际的嵌入式系统运行环境有很大的不同, 造成实际操作结果与现实不符, 影响了单元试验的可靠性。为保证测试函数的输出与产品代码中的功能保持一致, 那么就必须在目标机上进行单元测试。同时, 利用实际工程中的单元测试脚本, 对宿主单元和靶机单元的测试结果进行了对比。嵌入式软件是基于嵌入式系统的, 具有很好的实时性。开发人员很难在写了一段代码之后立刻就能高效地执行这个程序, 这会造成更多的bug。软件单元测试是软件开发的基础工作, 是软件验证的重要工具。对软件进行一系列的单元测试, 可以有效改善软件的质量、减少软件的缺陷、降低软件的危险性、发现软件存在的潜在问题、增强代码的规范性、稳定性和可靠性。针对嵌入式软件的特性, 结合多年的测试实践, 参照有关标准, 给出了一系列的测试程序, 包括测试工具、测试内容、测试过程标准化等要求。嵌入式系统的测试环境包括了对对象和宿主两个方面。但是, 在单元测试阶段, 嵌入式软件则可以在宿主环境中进行, 除非是在目标环境中进行。

**关键词:** 目标机环境; 嵌入式软件; 单元测试

## Discussion on embedded software unit testing in target machine environment

Shaolei Liu

Shaanxi Fenghuo Electronics Co., Ltd. Baoji, Shaanxi 721006

**Abstract:** Embedded software is based on an embedded system and has good real-time performance. It's hard for developers to write a piece of code and immediately execute it efficiently, which leads to more bugs. Software unit testing is the basic work of software development and an important tool of software verification. A series of unit tests on software can effectively improve the quality of software, reduce software defects, reduce the risk of software, discover the potential problems of software, and enhance the standardization, stability, and reliability of code. According to the characteristics of embedded software, combined with many years of testing practice and referring to relevant standards, a series of testing procedures are given, including testing tools, testing content, testing process standardization, and other requirements. The test environment of the embedded system includes both the object and host. However, during the unit test phase, embedded software can be performed in the host environment, unless it is performed in the target environment.

**Keywords:** target machine environment; Embedded software; unit testing

### 前言:

随着信息技术的迅速发展, 计算机技术在现代社会中的作用日益突出, 在不同的行业中, 不同的应用环境下, 出现了很多不同的应用程序。嵌入式软件是整个嵌入式系统的重要组成部分, 由于其硬件、软件紧密结

合、实时、稳定等特点, 在航天、智能汽车、智能医疗等领域得到了广泛的应用。在实际工程中, 嵌入式软件测试包含四个级别: 单元测试、部件测试、配置项测试、系统测试, 本文就嵌入式软件的单元测试展开讨论。

## 一、单元测试基本理论

软件单元是软件的基础，软件单元测试是将软件划分为独立的单元进行独立的测试，并按照软件的具体设计要求进行测试。单元测试的目标是检验各单元的功能、性能、接口和其他的设计需求，在软件详细的设计规范中都有明确的规定。在单元试验中，通常采取静态和动态两种方法。静态测试不需要执行代码，它可以用手工或者工具来查找包含在代码，文件中的错误。动态试验是基于真实代码来检验试验单位是否正确。静态测试主要有文件检验、静态分析、程式码校验。

## 二、嵌入式软件现状分析

在进行嵌入式系统的开发过程中，通常会按照技术规格的要求，在软件单元的测试阶段，通过大量的实例来覆盖和分析这些实例，使软件的测试实例覆盖到整个代码结构。单元测试是在主机环境中进行的，由于没有特殊的硬件资源，可以实现多个人的并行工作，这样就能快速地在项目的早期解决问题，加快开发过程。但是，由于CPU的差异，主机和靶机使用不同的指令，所以在数据的字长、内存字节的排列方式、大小段、浮点数的表达方式上都有很大的差异。在两种环境中产生可执行程序时，所使用的编译器、连接器也会有很大的差别。

## 三、目标机单元测试环境开发

当对象计算机进行单元测试时，编写并产生单元测试脚本，并将测试代码插入。由于对操作系统的文件系统的依赖，主要体现在测试库的功能上，所以，测试工具需要用其他方式来代替有关的文件系统。比如QASystem的Cantata单元测试软件，该软件可以用于靶机，可以通过对目标机器编译程序进行配置，从而实现对测试库的功能的重构，以及对文件系统进行自定义的替换。本论文中使用了串行通讯传输功能，取代了文件传输<sup>[1]</sup>。

在实际应用过程中，还必须针对硬件环境的特性，对BSP进行相应的开发。为实现整个工程的单元测试脚本的自动执行，还必须提供一个自动装载和运行控制的方式，以便可以按次序进行测试脚本。

其中，Cantata工具软件的主要任务是插入被测的代码，从而在测试过程中容易地获取代码的覆盖范围；对试验的结果进行图表的分析，方便分析；根据测试库源代码，提供部署工具，以产生对象测试功能。C语言标准的编译环境；提供了编译的功能，可以把测试的代码编译成OJM；提供连接功能，将已编译的待测及所测程式码obj档、启动代码、目标板支持套件、C语言标准库、

目标机测试库等等，组成了一个可执行的程序。提供重新定位的功能，可以将以上所描述的可执行程序按照目标面板的地址空间指派，转换为一个可以被加载到一个目标环境中的镜像。为目标环境加载可执行程序，并对其进行操作；完成对试验的总结、分析等。在此系统中，下位机在上电后，软件会自动开始，并等待下载。在软件下载完毕后，无需人为干预，软件下载后自动完成。软件按要求装入计算机所需的软件，并与下位机握手，并完成软件的装入<sup>[2]</sup>。结果捕获程序以命令行方式启动，并支持文件名称和路径，一旦接收到搜索结果就会自动退出。首先，基于编译环境和CPU类型，对该工具进行了测试。针对目标机的实际状况，产生起始码及靶板支援套件。利用代码插件来完成测试代码，并利用编译连接工具完成了机器测试库、单元测试脚本、测试后测试代码、启动代码、模板支持软件等的编译链接。利用装入工具装入可执行程式，利用结果捕捉程式取得测试结果，利用结果分析工具进行测试，显示试验结果<sup>[3]</sup>。

## 四、单元测试技术

在通过审核的过程中，测试员根据文件的完整性、一致性和正确性，检查测试文档的内容。文档的完整性是对文档的完整性进行检验，它包含了与系统文件相关的信息；文件集中在同一术语，名词，数据描述和文件内部和不同文件之间是否一致；文件的正确性会检验文件的内容是否正确，是否包含了错误，是否存在歧义。静态分析通常依靠软件检测，而静态分析能够检测出软件代码和对其结构的静态分析。在代码规则中，通过逐行的方式对所测试的软件进行检测，判断它与具体的程式码要求相符。静态结构分析包括：控制流分析、数据流分析、接口分析和表达式分析，在此基础上，接口分析可以检测功能的调用，表达式分析可以发现在代码中使用括号、下标越界等问题。代码检查的焦点是程序变量的初始化、接口、操作符、例外处理、程式码注释、程式的规范，程式的执行。一些嵌入式软件的代码达到数万行，仅靠手工的方式进行，这对测试者来说是一个很大的挑战，因此通常会把代码审核放在静态分析的后面，这样就能把大部分的代码规则都找出来，这样就能让测试人员把注意力放在仪器不能探测的地方<sup>[4]</sup>。

在动态测试中，常用的测试技术包括白盒测试和黑盒测试，而白盒检测技术是基于软件的内在逻辑构造，将被测试的物体视为一个开放的箱子。在完成测试后，判断用例和代码的逻辑覆盖，常用的逻辑覆盖包括语句覆盖、判断覆盖、条件组合覆盖、MC/DC（修改条件判定覆盖）。在实际应用中，常用的是语句覆盖、判断覆

盖、MC/DC覆盖。白盒测验对应的是黑箱测验，包含：等价类划分，边值分析，错误推断，因果关系图等。有的被测单元可以设计出无限多的试验案例，但是在实际测量时，必须保证测试用例的数目是固定的，不然就不能进行测试。等价类划分法把测试输入分成几个类别，在每个类别中选取“代表”，将输入的数据从无穷大变为有穷值。在过去的工程实践中，由于数据的限制，数据处理程序常常会出现错误。在确定数据区间的试验输入时，采用边界值分析方法，应采用最大、最小、稍大、稍小的试验案例，充分地检验了程序的边界数据处理能力。

### 五、嵌入式软件单元测试经验

在软件工程的生命周期模式中，每项工作都是按照下列次序进行的：需求分析，软件设计，代码，测试，验收，交货和维护。若程序码完成后再进行测试，则会导致程序码的执行延迟，进而影响整个工程的进度。在具体的项目中，首先要做的是软件的设计，要看一遍软件的设计手册，了解它的功能和性能。然后再进行相关的测试。通过对软件代码的编写，可以实现对系统的测试，从而大大缩短了开发过程。在对文档进行审核的过程中，一般的情况是，文档的编写与系统的需求是一致的，但是在内容的完整性和准确性方面存在着很大的问题，而最主要的问题在于文档的一致性。当文件编制者在更改文件中的某个地方或者术语的时候，他们会忘记对其他文档和其他文档的修改。比如：在嵌入式软件系统中，软件和硬件之间的交互比较多，经常将数据从硬件中读出，然后将计算的结果输出给外界，文件中包含了大量的数据定义和数据取值范围，这就造成了数据的定义和数值范围的改变。许多开发者对文件的使用并不重视，将其作为衡量软件质量的唯一依据。然而，文档是代码的基石，高质量的文档是高质量软件开发的先决条件<sup>[5]</sup>。

文档中的微小错误会造成软件的致命缺陷，在软件开发的进程中，文件是必不可少的，它应该和代码一样重要。在进行静态分析之前，为了防止因规则冲突而导致的误报、漏报，必须利用测试工具对其进行编程。通过对某些在动态测试中不易发现的代码错误进行检测，

从而提高了测试的整体工作效率。在软件开发过程中，主要的模块、关键的逻辑和复杂的算法，包括终端设计和响应、多任务调度、数据过滤、人机交互、故障查询等。在评审阶段，程式设计师将程式码逻辑解释，参加者提出问题，并进行讨论，以找出程式中的错误。在嵌入式软件单元测试中，通常都要对被测单元进行常规和非常规的测试，以检测被测单元能否正确地处理异常输入。在大部分时间里，当所有的测试用例都通过白盒测试方法来完成时，程序达到了一定的逻辑覆盖率，被测单元的动态测试就会结束。但在某些特定的场合下，当被测单元进行了逻辑覆盖测试之后，其相应的功能并没有得到足够的测试，那么测试过程还没有结束，然后，测试人员应采用“黑箱”的方式来设计尚未完成的功能，直至所有的功能都被测试。

### 六、结论

由于单元测试环境的不同，导致目标机器与宿主之间的单元测试结果不一致，从而影响到宿主所进行的单元测试的可靠性，尤其是CPU字长度和指令集合的不同。通过开发合适的目标机器测试环境，可以有效地将宿主的单元测试脚本引入到目标机器中，避免了环境的不同对测试结果的影响。在高安全性的关键软件中，要在目标机上进行单元试验，以提高其可靠性。

### 参考文献：

- [1]张猛, 毛亮. 航天嵌入式软件的单元测试方法探讨[J]. 航天器工程, 2020, 15(2): 32-35.
- [2]尤中桐, 徐智. 嵌入式软件动态单元测试方法研究[J]. 电子制作, 2020(14): 17-18, 5.
- [3]高智杰, 雷红瑛, 史国华. 基于覆盖率的嵌入式软件单元测试研究[C]. // 全国第七届嵌入式系统与单片机学术交流会论文集. 2021: 153-155.
- [4]王磊, 康智, 刘金龙, 等. 基于VzWorks的嵌入式软件单元动态测试研究[C]. // 2020年全国信息、电子与控制技术学术会议(IECT'2020)论文集. 2020: 232-236.
- [5]王晶, 孙越强, 陶鹏. 空间环境监测软件单元测试方法探讨[C]. // 中国空间科学学会空间探测专业委员会第十七次学术会议论文集(上册). 北京: 中国空间科学学会, 2022: 246-250.