

基于 Jenkins 的汽车电子软件持续集成技术研究

高德志 陈树星 王春锦 王恩东 王宗兴

(一汽解放汽车有限公司商用车开发院, 长春 130011)

【欢迎引用】高德志, 陈树星, 王春锦, 等. 基于 Jenkins 的汽车电子软件持续集成技术研究[J]. 汽车文摘, 2025(2): 13-17.

【Cite this paper】GAO D Z, CHEN S X, WANG C J, et al. Research on Continuous Integration Technology of Automotive Electronic Software Based on Jenkins[J]. Automotive Digest (Chinese), 2025(2): 13-17.

【摘要】针对汽车电子软件开发工作中集成与测试工作繁杂重复、人为操作效率低的问题,设计了一种基于 Jenkins 的汽车发动机 ECU 软件持续集成系统。该方法在车载嵌入式软件开发技术中融合了 Jenkins 持续集成技术,将 SVN 软件版本管理、HighTec 集成编译、目标控制器远程刷写、软件测试及报告通知环节整合成一个自动化流程,搭建了一个基于 Jenkins 技术的汽车电子软件持续集成系统。该系统可大幅缩短软件迭代升级的开发周期,避免人为集成操作失误。

关键词: Jenkins; 持续集成; 汽车电子软件开发; SVN 系统; 邮件通知

中图分类号: U462.1 文献标志码: A DOI: 10.19822/j.cnki.1671-6329.20240150

Research on Continuous Integration Technology of Automotive Electronic Software Based on Jenkins

Gao Dezhi, Chen Shuxing, Wang Chunjin, Wang Endong, Wang Zongxing

(Commercial Vehicle Development Institute, FAW Jiefang Truck Co., Ltd., Changchun 130011)

【Abstract】Aiming at the problems of complicated and repetitive integration and testing which are inefficient in automotive electronics software development, a continuous integration system for automotive engine ECU software based on Jenkins is designed. This method combines Jenkins continuous integration technology and automotive embedded software development technology and connects development process nodes into an automated process which is called Jenkins-based electronics software continuous integration technology. The development process nodes include SVN software version management, HighTec integrated compilation, remote flashing of target ECU, software integration test and mail notification. This Jenkins system can substantially shorten the development cycle of software iterative upgrades and avoid human integration operation errors.

Key words: Jenkins, Continuous integration, Automotive electronic software development, SVN system, Mail notification

0 引言

随着“软件定义汽车”趋势在行业内受到广泛认可,汽车电子软件开发在整车研发流程中的重要性日益凸显。汽车电子软件开发是指在车规级芯片电控单元(Electronic Control Unit, ECU)上进行的软件程序开发,该过程包括需求分析、架构设计、代码开发、集成测试以及释放验证。实际开发项目中产品软件不断迭代升级,工程师频繁集成最新的软件提交物,这要求工程师在专注于需求分析与模块代码开发的同

时,还需投入大量精力处理软件配置管理、代码集成编译、测试及通知等环节,造成研发资源浪费。为解决汽车电子软件开发这一痛点,亟需一种自动化集成方法,提高汽车电子软件迭代升级的效率。

自动化持续集成是一种可以提高软件开发效率和质量的重要实践方法。通过构建自动化的开发集成流程,开发人员可以更快地完成软件交付,并降低错误率。当前在消费电子及 IT 领域,自动化持续集成方案已广泛使用。相比传统软件行业领域,持续集成技术在汽车电子软件领域尚待推广应用。本文针对

汽车电子软件自动化持续集成技术,以目前主流的PowerPC多核架构芯片嵌入式软件研发平台工具链为例,搭建一种基于Jenkins技术的软件持续集成系统。该系统利用功能丰富、开源免费的Jenkins软件,将汽车电子软件开发工作中的软件版本管理、集成编译、目标控制器刷写、软件集成测试及报告通知等标准流程化的操作进行持续自动化执行。

1 基于Jenkins的持续集成系统概述

持续集成系统一般由持续集成引擎、软件配置管理系统、集成编译调试系统、测试系统以及监控通知系统组成,如图1所示。持续集成系统旨在满足敏捷软件开发^[1]的需求。相比于人工集成,持续集成系统分2步实现:(1)各子系统实现自动化运行。(2)持续集成引擎驱动各子系统实现整个持续集成系统运转。

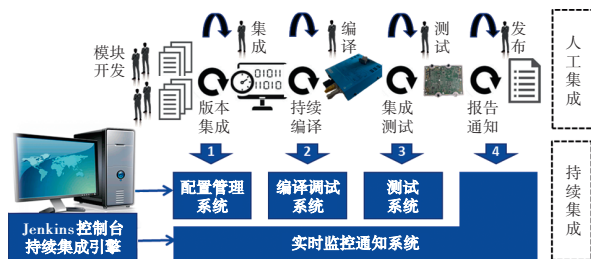


图1 人工集成与持续集成

1.1 持续集成引擎

持续集成引擎通过构建标准流程化任务实现监控自动化任务执行。Jenkins是一个基于Java语言开发的可扩展的持续集成引擎,是由Hudson^[2]技术发展而来的开源免费的持续集成基础工具。Jenkins具有便捷的拓展接口,可以满足不同系统方案需求,如任务构建、测试通知以及软件部署。Jenkins的开放生态为用户提供了丰富的插件资源,主要包括版本控制、构建工具、代码质量度量、构建通知、外部系统集成以及用户界面定制化,这些插件开源免费且具有较高时效性。

1.2 软件配置管理

配置管理是指在软件产品开发过程中,对整个生命周期实施的一系列控制和管理措施,主要包括配置计划管理、版本库管理以及变更控制管理。当前主流的配置管理系统有Meran公司的PVCS(Project Version Control System)、Rational公司的ClearCase以及Collabnet提供的开源工具SVN(SubVersion)。

SVN采用服务器/客户机(Client/Server)模式运行。Server端通过中央资料档案库(Repository)管理

数据的变更时间及版本信息。软件开发项目中的各种版本的程序源码均存储在服务器上,程序开发人员作为客户端可以从服务器获取最新版本数据进行开发工作,也可以随时提交更新后的任务包。

配置管理与持续集成引擎的结合是持续集成系统的关键,设计目标是当开发人员上传提交物时,SVN可以检测到Repository中的内容变更并自动触发Jenkins相关脚本任务执行。

1.3 集成编译调试

编译器是一种可以将高级软件语言的程序生成目标机器可执行程序的软件工具。HighTec编译器适用于PowerPC多核架构芯片,可以提供面向嵌入式系统软件工程的集成开发环境,支持软件编辑、编译链接及调试功能。HighTec编译器内核支持脚本语言开发,通过驱动HighTec工程的自动化脚本程序,实现持续集成系统的自动编译功能。

调试器是一种用于控制软件运行、查看软件运行状态及信息、调整软件执行流程的软件开发辅助工具。嵌入式系统调试器兼有对目标板下载刷写功能。iSYSTEM调试仿真器可适用于PowerPC多核架构芯片调试。winIDEA集成调试环境可与iSYSTEM硬件进行交互操作。winIDEA调试环境提供了丰富的接口功能,通过调用iSYSTEM.Connect API接口,可实现外部应用程序对iSYSTEM进行远程控制,进而实现对目标硬件的刷写和调试操作,为持续集成系统配置远程刷写功能配置提供了技术方案。

1.4 测试驱动及监控通知

软件开发过程中需执行测试用例以验证软件功能,并反馈验证结果通知。软件迭代持续升级阶段需大量测试以保证未修改的软件功能正确运行。软件回归测试通过对原有功能测试确保新修改软件中的原有功能保持正常运行,测试用例一般在已有的测试用例库中选取代码静态测试是迭代升级阶段保证软件质量的重要手段。Ploypspace是一种利用语义分析技术进行软件代码静态测试的工具,在MATLAB环境中,通过加载目标代码,其可便捷地进行软件编程规范检查与程序静态结构分析。

软件测试的自动化技术随着汽车电子软件测试技术的发展,工具接口越来越丰富化。持续集成系统在测试任务中,可直接驱动接口执行自动化测试。在回归测试阶段,持续集成系统通过驱动接口调用已有的自动化测试用例库执行。Jenkins持续集成引

擎可以提供邮件通知功能,也可以通过扩展插件接口集成定制的通知系统,将测试报告发送至相关工程师,工程师根据结果决定进行发布或继续迭代优化。

2 基于 Jenkins 的发动机 ECU 软件持续集成系统设计

Jenkins 持续集成系统对基于苏州国芯 CCFC3007 PT 芯片的发动机控制器开发平台进行设计,整合开发流程中的 SVN 工具进行软件版本同步,整合 HighTec 编译工具及 iSYSTEM 调试工具实现编译下载,整合 Polyspace、CANoe 工具以及 Mail 系统实现软件测试及

报告通知,如图 2 所示。

2.1 持续集成系统搭建

选取工作站电脑搭建 Jenkins 软件持续集成系统服务器,其环境参数如表 1 所示。Jenkins 是基于 Java 开发的一种持续集成工具,Windows 操作系统中仍需安装 Java 环境。Jenkins 安装部署的默认端口号 8080 为常用端口号,为避免与系统中其他服务端口冲突,系统选用 8081 端口进行安装,指令如下:

CMD 指令:java -jar jenkins.war --httpPort=8081

Jenkins 持续集成系统服务器可以通过任意客户端电脑 IP 访问服务器端,进行工程任务配置及任务执行状态的查看。

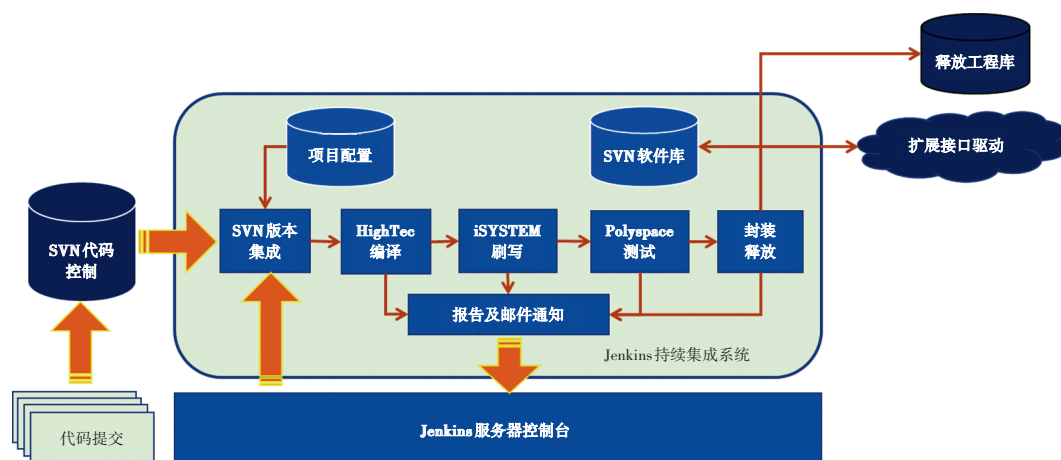


图 2 Jenkins 持续集成系统

表 1 Jenkins 服务器环境关键参数

配置项	参数
CPU	Intel Xeon Gold 5118 @2.30GHz
内存	64G
操作系统	WIN7 企业版
Java 环境	1.8.0_73
Jenkins 版本	2.176

2.1.1 SVN 版本管理

待集成软件通过 SVN 客户端上传至 SVN 服务器,SVN 服务器的版本管理系统将提交软件的版本管理信息、变更原因载入 SVN 数据库。Jenkins 持续集成系统通过开源插件 Subversion 获取 SVN 数据库访问权限,将待集成的软件远程下载、版本同步至持续集成系统服务器。Jenkins 持续集成系统通过配置定时任务和 HOOK 机制触发持续集成服务器与 SVN 服务器代码同步。HOOK 机制^[3]在 SVN 服务器端负责驱动 post-commit 脚本,该脚本负责触发 Jenkins 启动任务,脚本关键内容如下:

HOOK 脚本指令:

H:\curl\bin\curl.exe

http://admin:GaoDZ@10.60.175.14:8081/job/CI_Jenkins_SVN_HighTec_winIDEA/build?token=Jenkins

SVN 触发持续集成系统构建任务后,Jenkins 控制台将提交 SVN 时的备注作为任务执行的触发描述,提交代码时须附清晰合理的变更描述,如图 3 所示。

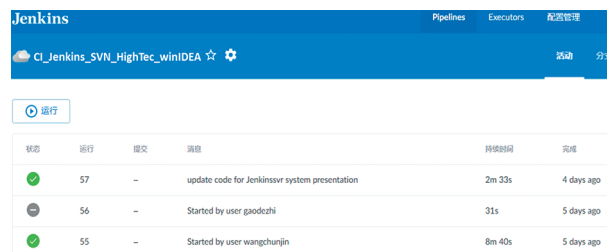


图 3 持续集成任务项目控制台

2.1.2 HighTec 自动编译

PowerPC 多核架构 CCFC3007PT 芯片软件开发项目选用 HighTec 作为编译器工具。HighTec 编译器通过集成开发环境进行嵌入式软件工程环境配置、代码编辑以及软件编译,编译生成的二进制可执行文件用

于CCFC3007PT芯片刷写。持续集成系统HighTec编译器使用命令行方式通过make指令执行编译操作,关键指令如下:

make指令: -clean 清除已有输出文件,重新构建。

持续集成系统自动编译系统搭建步骤:(1)持续集成服务器对SVN同步的软件工程进行HighTec编译器配置,完成编译环境搭建。(2)利用make指令通过命令行脚本的方式实现脚本化运行HighTec编译。(3) Jenkins系统任务中添加make指令脚本完成编译任务设置,如图4所示。

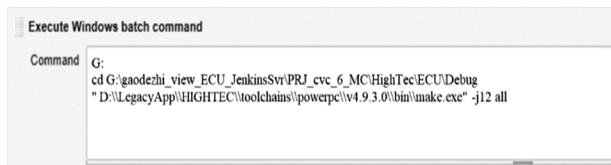


图4 Jenkins配置HighTec编译任务

2.1.3 IC5000自动刷写

持续集成系统自动编译成功后,调用Jenkins集成安装路径下IC5000调试器的集成调试环境winIDEA以加载工作空间的配置信息,并自动下载代码至目标板。

持续集成系统自动刷写系统搭建步骤:

(1)持续集成服务系统中对编译成功的软件搭建winIDEA环境。目标调试器设置为IP地址远程访问,刷写端口类型设置为DAP2,将刷写文件列表及路径与HighTec的输出文件关联,winIDEA工程配置信息以“.xjrf”类型的文件形式保存。

(2)利用iSYSTEM提供的Python软件开发工具包中的isystem.connect类函数,加载winIDEA工程配置,用以下Python脚本执行连接、下载、运行操作。

Python指令:

```

# 连接winIDEA加载项目配置
cmgr = ic.ConnectionMgr()
cmgr.connectMRU(r'H:\bat\ECU.xjrf')
# 执行下载、运行
dbg = ic.CDebugFacade(cmgr)
dbg.download()
dbg.run()
    
```

(3)与自动编译系统类似,Jenkins持续集成系统任务中驱动Python脚本完成刷写任务设置。

2.1.4 测试驱动

持续集成系统完成编译、下载和运行后,测试环节通过Jenkins的任务构建命令驱动测试程序自动运行^[4],因此持续集成系统测试环节的先决条件是具备可

运行的测试工具软件。本系统以Polyspace静态代码测试为例,持续集成系统自动检测系统搭建步骤如下:

(1)持续集成服务器为目标软件工程建立Code Prover与Bug Finder检测的Polyspace工程。

(2)编写命令行自动化脚本调用Polyspace工程运行Code Prover与Bug Finder检测,设置报告文件名称、报告模板、文档格式及输出路径,关键指令如下:

CMD指令:

```

"..\\polyspace-code-prover-nodesktop.exe"
-langC-options-file    "..\\options_command.txt"
-report-template      "..\\Developer.rpt"
-report-output-format  pdf
-report-output-name   CP_Result
-results-dir          "..\\ACP_Result"
    
```

(3) Jenkins持续集成系统任务中驱动Polyspace测试运行脚本完成静态测试任务设置。

除静态测试外,持续集成系统在目标板上触发软件运行后,可远程驱动功能测试系统运行,如硬件在环(Hardware In Loop, HIL)测试系统^[5]。针对基于CANoe^[6]、CANdiva等工具自主开发的CAPL(Communication Access Programming Language)是一种专门用于控制器局域网(Controller Area Network, CAN)通信和诊断的脚本语言,由Vector公司开发测试用例库,Jenkins通过调用canoe32.exe脚本执行测试用例文件^[7],实现定制功能测试系统运行。

2.2 通知与释放

持续集成系统通过Jenkins自有邮件通知系统将模块工程师、集成工程师及下一环节接口工程师作为邮件接收人,邮件正文的内容列举持续集成系统的项目名称、SVN库路径、版本管理信息^[8]、触发集成原因以及Jenkins持续集成系统内部状态信息,如图5所示。邮件通过格式重点突出持续集成任务^[9]的最终结果,通过附件提供编译记录和测试报告持续集成过程细节。



图5 Jenkins持续集成系统项目结果反馈

3 应用效果

Jenkins持续集成系统中,工程师使用SVN客户端提交软件模块,SVN服务器检测提交物检入成功,触发持续集成服务器Jenkins_SVN_HighTec_winIDEA项目,依次执行代码版本同步、HighTec编译、iSYSTEM刷写以及Polyspace静态测试,将每个环节的执行结果进行邮件通知。全部流程在无人值守的情况下Jenkins持续集成系统平均用时12 min,同样的环节人工集成平均用时21 min,由此可得基于Jenkins技术的软件持续集成技术有效缩短汽车电子软件集成时间,同时降低人为操作失误的影响。

表2 集成时间对比

	人工集成	Jenkins持续集成
版本同步	3 min	<1 min
编译	5 min	2 min 25 s
刷写	8 min	7 min 29 s
静态测试*	3 min	2 min
邮件通知	2 min	<1 min
总计	21 min	12 min

注:*静态测试环节使用单个软件模块为例

4 总结与展望

本文提出了一种基于Jenkins的汽车电子软件持续集成技术,以CFCC3007PT芯片方案的控制器开发为平台,通过Jenkins的强大驱动功能,实现了软件版本集成、编译下载、测试以及邮件通知等环节的自动执行^[10]。未来自动化测试技术不断发展,可进一步提高软件开发效率,精简集成测试环节的人力投入。

持续集成技术的广泛应用加速了软件行业的发

展,基于Jenkins的汽车电子软件持续集成技术的应用助推了“软件定义汽车”的行业变革转型。未来汽车电子架构中为满足汽车“电动化、智能化、网联化、共享化”要求,软件功能将以智能应用软件的形式存在。不断升级的车载智能应用软件衍生更多软件集成需求,届时持续不断的交付发布软件、提高软件的时间附加值、抢占软件发布的市场先机将是Jenkins持续集成技术最为重要的作用。

参考文献

- [1] 田莉. 论如何借鉴敏捷软件开发方法来改进软件项目管理[C]// 第三十二届中国(天津)2018'IT、网络、信息技术、电子、仪器仪表创新学术会议论文集, 2018: 330-333.
- [2] 牛璟. 分布式系统中的持续集成系统的研究与实现[D]. 上海: 复旦大学, 2012.
- [3] 姜文, 刘立康. 基于SVN的应用软件持续集成[J]. 计算机测量与控制, 2016, 24(3): 109-113.
- [4] 李英玲, 王青. 持续集成测试用例集优化综述研究[J]. 软件学报, 2018, 29(10): 3021-3050.
- [5] 车晓镭. 汽车动力总成电控单元硬件在环测试系统研究[D]. 长春: 吉林大学, 2011.
- [6] 钱俊磊. 基于CANoe和Jenkins的ECU软件自动化测试系统的设计与实现[J]. 汽车实用技术, 2019(22): 64-67.
- [7] 徐永新, 朱娟, 王裕鹏. 基于ECU_TEST的ECU报文自动测试研究与应用[J]. 汽车电器, 2018(10): 71-74.
- [8] 王静, 夏禹. 一种汽车软件版本管理方法, 系统及介质: 202410109232[P]. 2024-07-23.
- [9] 徐江川. 集群部署方法, 装置, 车辆和存储介质: 202311114096[P]. 2024-07-23.
- [10] 谢众玉, 王功亮, 刘西学, 等. 一种汽车软件持续集成测试方法及系统: 202310822237[P]. 2024-07-23.

(责任编辑 梵玲)