

FORD LDM Localization - 功能 #3386

Task # 3240 (进行中): FORD文档输出

功能 # 3349 (进行中): SWQA文档

功能 # 3356 (已解决): TDR_RQT_003701_705379 Stack Overflow/Underflow

TDR_RQT_003701_705379 001堆栈溢出/下溢

2025-03-17 11:19 - 力常 张

状态:	进行中	开始日期:	2025-03-17
优先级:	普通	计划完成日期:	
指派给:	云浩 汪	% 完成:	50%
类别:		预期时间:	0.00 小时
目标版本:	FORD_TDR_Documents	耗时:	0.00 小时
描述 参考软件DR-003701-707983 Stack Overflow/Underflow 需求描述 若堆栈发生溢出，软件无法从此状态恢复，必须重新初始化。 · 软件需在堆栈溢出/下溢发生后的25毫秒内检测到该状态。 · 当堆栈溢出/下溢时，模块必须完成以下操作： 更新参数标识符PID D701的字节#1（若适用） 强制执行复位（Force a Reset） 技术规范 在并发系统（concurrent systems）中，堆栈溢出（stack overflow）是最常见的问题之一。其成因之一是程序员未为任务分配足够内存；为避免此问题，开发者倾向于为堆栈分配过量内存，导致程序可能永远用不到这些资源。 理想情况下，每个任务都保留了足够的资源，这样就不会浪费内存，并且每个任务都有可用的内存（即使在最糟糕的情况下）。实现完美的平衡可能非常困难（分配尽可能多的堆栈，但尽可能灵活）。 即便采用最优策略和/或方法，堆栈尺寸计算在多数情况下仅为预估。因此，若分配资源无法满足任务需求，堆栈溢出风险始终存在。 例如，若程序员在堆栈已满时尝试执行"压栈（push）"操作，将发生堆栈溢出（stack overflow）；反之，若系统在堆栈为空时执行"弹栈（pop）"操作，则触发堆栈下溢（stack underflow）。 堆栈溢出/下溢的负面后果包括： . 内存损坏（Memory corruption） . 任意代码执行（Arbitrary code execution） . 系统崩溃（System crash） . 未定义行为（Undefined behavior） 故障诊断难点： 确定堆栈溢出/下溢为软件问题的根本原因极具挑战性。 故障日志记录要求： 对于支持软件健康监测与报告（Software Health Monitoring & Reporting）功能的模块，其最低要求已通过PARSED文档发布。请参阅该文档以获取潜在附加要求（参见链接）。 本设计规则支持RQT-003701-705379 “堆栈溢出/下溢”，适用于所有带软件的电子模块，是符合RQT的方法。 设计规则关闭机制：这些设计规则的结束是软件技术设计评审（TDR）的完成，并通过“SWQA通用TDR检查表”问题解决： 阐述堆栈溢出/下溢（stack overflow/underflow）的预防与检测技术 IRH08009 A) 如何检测堆栈溢出/下溢？ B) 堆栈溢出/下溢是否会导致复位（reset）？ C) 堆栈溢出/下溢是否被记录（如DTC或供应商内部故障日志）？ D) 堆栈余量（stack margin）是多少？如何检测余量？ E) 如何确定所需的堆栈资源？			

历史记录

- #1 - 2025-03-17 11:20 - 力常 张
- 状态从 新建 变更为 已解决
- #2 - 2025-03-24 14:55 - 稚媛 黄
- 目标版本从 H005_SW0007169.A003.2 变更为 FORD_TDR_Documents
- #3 - 2025-03-24 15:29 - 稚媛 黄
- 主题从 DR-003701-707983 堆栈溢出/下溢 变更为 TDR_RQT_003701_705379 001堆栈溢出/下溢

-描述 已更新。

#4 - 2025-03-25 13:19 - 涛陆

-描述 已更新。

-状态从 已解决 变更为 进行中

-指派给从 力常张 变更为 云浩汪

-%完成从 100 变更为 50

在集成测试时，测试堆栈溢出情况