符合功能安全 ASIL D 的电机控制器软件架构设计研究

王江涛[1],龚道清[2],卢 苗[2],任广辉[3],王 云[2]
[1.华东师范大学国家可信嵌入式软件工程技术研究中心,上海 200062;2.广东省大湾区集成电路与系统应用研究院,广东 广州 510535;3.中科意创(广州)科技有限公司,广东 广州 510535]

摘 要:本文根据 ISO26262 功能安全要求,基于 ST 的 SPC 系列多核微控制器,结合 AUTOSAR 方法论,设计符合功能安全 ASILD 等级要求的主驱电机控制器软件架构。在软件架构层级上对初始架构进行分析的基础上实施软件架构级别的安全分析,包括软件的 FMEA 分析以及 STPA 分析,得到相应安全机制,并进行了软件的实验验证。经过软件安全分析论证以及实验验证,该电机控制器软件架构符合功能安全 ASIL D 要求。

关键词:功能安全;ASIL D;电机控制器;软件架构;多核微控制器
中图分类号:TM301.2 **文献标识码:**A **文章编号:**1004-7344(2022)35-0139-06

0 引言

随着汽车电子系统的复杂度不断提高，系统失效导致的安全风险也随之提高。为了降低这些安全风险，相应电子电气系统符合功能安全要求逐渐成为趋势，国际标准化组织也在 2018 年发布了第二版 ISO26262 道路车辆功能安全标准[1]。主驱电机作为电动汽车的动力心脏,其安全性是行业一直以来关注的焦点。为了支持高阶自动驾驶，对于主驱电机控制器的功能安全要求也将更高，其功能安全的实现也成为电机控制器产品化的研究重点。

文献[2]针对纯电动汽车的电机控制器,进行了功能安全概念及系统开发的论述,并提出了 EGAS 三层架构的应用方法。文献[3]基于相电压和电流进行扭矩估算,论述了根据估算扭矩进行扭矩监控的方法。文献[4]分析了基于功能安全要求的电机控制器硬件设计方案,并指出了自由转动作为安全状态的适用场景。文献[5]从理论角度分析了主动短路作为安全关断方法的优缺点。文献[6]针对功率级的 HiL 开发了电机控制器系统级功能安全测试的测试用例以及测试场景，并指出了该方法对于符合功能安全验证要求的有效性。

本文依据 ISO26262 功能安全要求，基于 ST 的 SPC 系列多核微控制器,结合 AUTOSAR 方法论,设计了符合功能安全 ASIL D 等级要求的主驱电机控制器软件架构;并介绍了软件安全分析方法的实例。基于不同层级的实验平台,验证了基于该架构开发的软件对于实现 ASIL D 等级功能安全电机控制器产品的有效性。

1 电机控制器功能安全开发

1.1 安全目标与安全完整性等级

根据 ISO26262 要求,确定系统的安全目标与汽车安全完整性等级(ASIL),需要通过危害分析和分析评估(HARA)得出。HARA 是基于潜在伤害的严重度(S)、场景暴露度(E)和危害事件的可控度(C)三个参数分析确定系统的安全目标及其安全等级。针对主驱电机控制器系统的功能,进行危害分析和分析评估,通过分析系统的潜在风险,确定其安全完整性等级,进而归纳总结出安全目标及相关功能安全要求。考虑到篇幅限制,本文列举了扭矩控制功能的部分 HARA 结果作为参考,如表 1 所示。

表 1 电机控制器的危害和风险分析

故障类型	驾驶场景	风险类型	E	S	C	ASIL	安全目标
输出转矩大于请求转矩	行人区域低速通过	与行人发生碰撞	E4	S3	C3	D	输出转矩不应当大于请求扭矩
未请求转矩时输出转矩	城市或乡村道路	与前车追尾	E4	S3	C2	C	在未请求转矩输出时不输出转矩
请求转矩与输出转矩方向不一致	高速湿滑道路	车辆侧滑	E3	S3	C3	C	输出转矩方向应当与请求转矩一致

1.2 基于多核处理器的电机控制器系统安全架构

基于上文危害分析和风险评估分析确定出来的安全目标与安全完整性等级,即可执行 ASIL D 要求的电机控制器功能安全产品开发。安全产品需要基于高置信度的安全架构进行开发，通常会采用行业广泛认可的 EGAS 三层架构进行设计。在使用 EGAS 架构时,为

了降低系统成本与开发难度，可对 ASIL 等级进行恰当的分解，将其分解为基本功能与安全功能两部分。对基本功能的开发执行 QM 等级标准要求，而对安全功能的开发执行 ASIL 等级标准要求。对于本文的主驱电机控制器而言，将系统安全等级要求分解为 ASIL D=QM(D)+ASIL D(D)。

EGAS 三层架构通过对电机控制器的功能进行分层，从而实现电机控制器的安全实时监控。为了实现上述分层，通常有两种解决方式，一种是通过多核处理器实现，另一种是通过双芯片微处理器实现[7-8]。

双芯片微处理器的系统安全架构如图 1 所示。双芯片微处理器的系统安全架构通过功能芯片实现扭矩控制功能，即执行没有安全完整性等级要求的 Level1 功能，单独的安全芯片实现扭矩监控的安全功能，即执行 ASIL D 的 Level2 和 Level3 功能。该架构基于单独的硬件芯片，充分保证了硬件资源与软件代码的完全独立，可以简单可靠的实现 ASIL D 等级要求的系统，但产品硬件成本偏高且硬件结构稍显复杂，为了降低产品的物料成本，本文设计了通过多核处理器实现的系统安全架构。

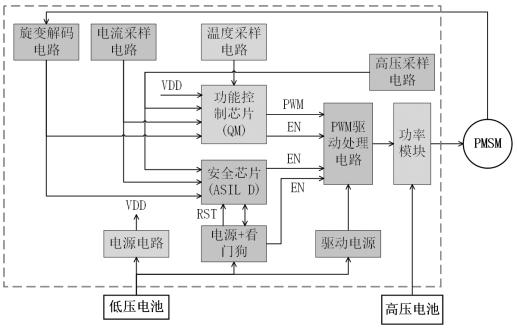


图 1 双芯片微处理器的安全架构

本文基于多核处理器的系统安全架构设计如图 2 所示。基于选用的 ST SPC 系列单片机提供的多核处理器，该架构在硬件层实现了对基本控制功能与安全监控功能的分离，需要在软件层面实现满足 ASIL D 的 EGAS 架构。该架构中 CPU1 为非锁步核，在该非安全核中执行扭矩控制相关的程序（Level1），而 CPU2 和 CPU0 是锁步核，在这两个安全核中执行安全功能，即在 CPU2 执行扭矩监控相关的程序(Level2)，同时与外部的看门狗实现对单片机本身及其中运行程序流的监控(Level3)。CPU2 通过实时的对扭矩进行计算与监控，当监测到扭矩异常影响安全时，通过 CPU2 发出的指令信号输出到 PWM 驱动处理电路进行 PWM 波封锁实现安全关断，从而激活电机控制器系统安全状态；当看门狗监测到程序流或芯片异常后，在直接输出使能信号对 PWM 驱动处理电路进行封锁的同时激活电机控制器系统安全状态。

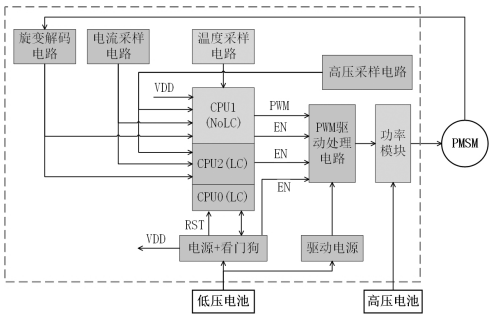


图 2 多核处理器的系统安全架构

2 符合功能安全 ASIL D 的软件架构设计

2.1 基于 AUTOSAR 的软件架构设计

根据 ISO26262 对软件开发的要求以及 1.2 节的分析，为了通过软件实现 EGAS 架构，软件架构的设计至关重要，需要将软件元素之间进行解耦，而 AUTOSAR 架构则很好的满足了此要求。图 3 是基于 AUTOSAR 方法论针对本文电机控制器设计的软件静态架构。

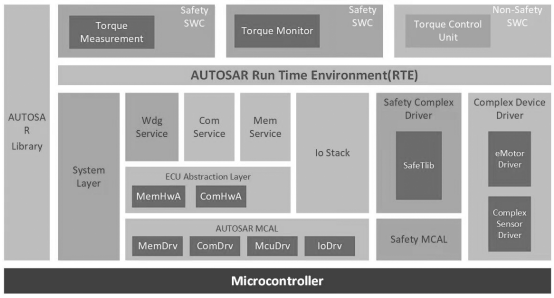


图 3 电机控制器软件静态架构

图中标深色的为带有安全等级的软件模块，底层主要是系统服务模块，看门狗服务模块，安全 MCAL 模块以及安全相关的复杂驱动模块，应用层带有安全等级的模块是扭矩计算与扭矩监控模块；其余部分包含电机控制应用层均为 QM 软件模块。其中安全相关的复杂驱动模块主要包含 SafeTKit 等模块。

2.2 软件架构设计中的安全机制

软件架构设计除开指导软件开发，也需要设计一些安全机制保证满足功能安全要求。而基于多核处理器开发功能安全产品，安全软件与功能软件是通过同一个单片机实现，因此保证安全软件的免于干扰将是实现软件功能安全的关键一环。根据 ISO26262 要求，如果系统设计者不能证明所开发安全软件可以免于非安全软件的干扰，则需要通过技术手段实现免于干扰。为了实现免于干扰，需要从内存、时间和执行以及信息交互等方面着手。

内存的免于干扰，主要需要实现未经授权访问安全相关软件组件的内存区域的防护，是由支持内存保护机制的操作系统来保证，这里的操作系统需要满足

可扩展性等级 3 或 4。在操作系统满足了相应要求的情况下，硬件需要具备内存保护单元（MPU：Memory Protection Unit）以支持实现内存保护机制。

在操作系统中，用于实现内存保护部分的软件被称为安全上下文（Safe Context），控制了软件组件在任务切换和中断的期间的隔离，可以防止一个软件组件在未经许可的情况下，写入另一个软件组件的内存。安全上下文被开发为 ASIL D 等级，因此被授权在运行期间，重新配置受 MPU 保护的各种任务和中断的内存范围。这保证了在保存和恢复上下文数据（包括 MPU 配置）方面与 ASIL D 的一致性。

通过对操作系统对象（如任务和中断）进行逻辑分组，并将其对应到 MPU 配置中。并借助于 AUTOSAR 中定义的操作系统应用程序（OS-Application）的概念，则可用于配置操作系统。基于该配置，根据 QM 过程开发的基础软件模块被合并到一个单独的操作系统应用程序中并与 ASIL 等级的软件隔离，因此避免了 ASIL 软件的内存被 QM 等级的软件进行错误的改写。

基于软件分区，我们将应用层的电机控制功能模块以及扭矩监控功能模块进行分区保护，以及底层安全相关模块和非安全相关模块进行分区保护，从而实现内存上面的免于干扰。

时间和执行的保护需要基于操作系统的时间保护实现，信息交互的保护则需要基于 AUTOSAR 的端到端通信库实现。限于篇幅，这里仅对内存的免于干扰的原理做详细的阐述。

除了上述的安全机制，为了实现控制器的安全运行环境，需要设计 EGAS 架构中的 Level3 来保证，本文设计的 Level3 软件功能架构如图 4 所示。其中的 SafeTKit 软件模块，基于单片机内部集成的硬件安全机制，通过检测这些硬件安全机制是否能够如预期正常工作，可以帮助实现控制器的安全运行环境。

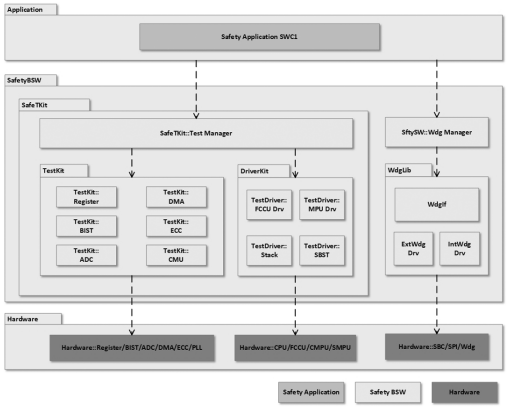


图 4 Level3 软件功能架构

SafeTKit 是基于 SPC 系列单片机开发的功能安全测试库，满足 ASIL D 的功能安全等级要求。SafeTKit 提供两种单片机的测试功能，一种是针对潜伏故障的检测功能，这种检测功能可在 Earlypre-Run，Pre-Run，Post-Run 等阶段运行一次即可；另一种是对单点故障的检测功能，可在 Run 阶段周期性运行。SafeTKit 软件包括如下两个部分。

（1）单片机测试功能库，主要包含 FCCU 的测试、寄存器的测试、DMA 的测试、逻辑/内存的自建测试、内存的 ECC 功能测试、时钟监控功能测试以及 ADC 功能测试等。

（2）单片机安全相关功能驱动库，主要包括 FCCU 驱动、MPU 驱动、堆栈保护监控以及基于软件的自检驱动（SBST）。

SafeTKit 的自检测试分为以个阶段进行运行：Early pre-run 阶段，FCCU 对应的故障触发未配置完成，它不会标志出任何由硬件安全机制检测到的错误，因此不能进行单片机潜伏故障的故障注入测试，此时可以测试 FCCU 错误信号。Pre-run 阶段会测试单片机的各个部件的安全机制以及 FCCU 的对于测试的响应，其中包括操作系统启动前完成的测试以及在激活操作系统后立即进行的测试。Run 阶段会周期性执行单点故障测试，如基于软件的自检测试，该功能需要基于用户的软件以及 SafeTKit 当中的驱动共同完成；除此之外，基于 SafeTKit 提供的驱动，单片机的硬件安全机制将会进行故障监测，如果检测到错误，SafeTKit 会触发 FCCU 警报。

使用 AUTOSAR 架构来实现电机控制器软件，可以把 SafeTKit 作为 AUTOSAR 的复杂设备驱动，如图 3 所示。在 AUTOSAR 架构中，可以在 EcuM 中触发 SafeTKit 的初始化以及执行 Pre-run 测试。Pre-run 测试可以被拆分为几个部分，并分别被触发，例如分别在操作系统启动前后触发测试。当然，SafeTKit 也可以集成到非 AUTOSTR 环境中。图 5 是在 AUTOSAR 架构中 FCCU 子模块 Pre-run 测试的软件动态架构。

3 软件设计的安全分析

由于软件架构当中具有多个执行安全功能的模块，为了在早期发现软件当中可能存在的安全缺陷，软件架构设计需要执行安全分析。同时，根据 ISO26262 的要求，需要通过软件安全分析对软件架构进行迭代设计。

3.1 软件架构 FMEA 分析

FMEA 分析是一种基于归纳的安全分析方法。FMEA 分析的目的是系统地评估故障在软件组件输出端的传播，检查软件系统是否在的适当位置定义了适当的安全机制，并定义相应的相关的发现和防范措施以降低风险。表 2 是对扭矩控制控制模块中的输出信号处理模块的软件 FMEA 分析。

表 2 电机控制器的软件 FMEA 表格

组件名称	失效模式		失效模式影响	系统层级的失效影响	严重度	预防措施	发生度	发现措施	检测度	AP
输出信号计算(扭矩控制模块)	错误的执行时间(过早)		力矩控制不正确	扭矩偏差超出允许范围	10	扭矩监控	6	软件检查	3	H
	错误的执行时间(过晚)		力矩控制不正确	扭矩偏差超出允许范围	10	扭矩监控	6	软件检查	3	H
	请求执行失败		力矩控制不正确	扭矩偏差超出允许范围	10	扭矩监控	6	软件检查	3	H
	错误的执行		力矩控制不正确	扭矩偏差超出允许范围	10	扭矩监控	6	软件检查	3	H
	旋转变压器转速	过小	力矩控制不正确	扭矩偏差超出允许范围	10	扭矩监控	6	软件检查	3	H
		过大	力矩控制不正确	扭矩偏差超出允许范围	10	扭矩监控	6	软件检查	3	H
	存储数据请求	真	力矩控制不正确	扭矩偏差超出允许范围	10	扭矩监控	6	软件检查	3	H
		假	力矩控制不正确	扭矩偏差超出允许范围	10	扭矩监控	6	软件检查	3	H

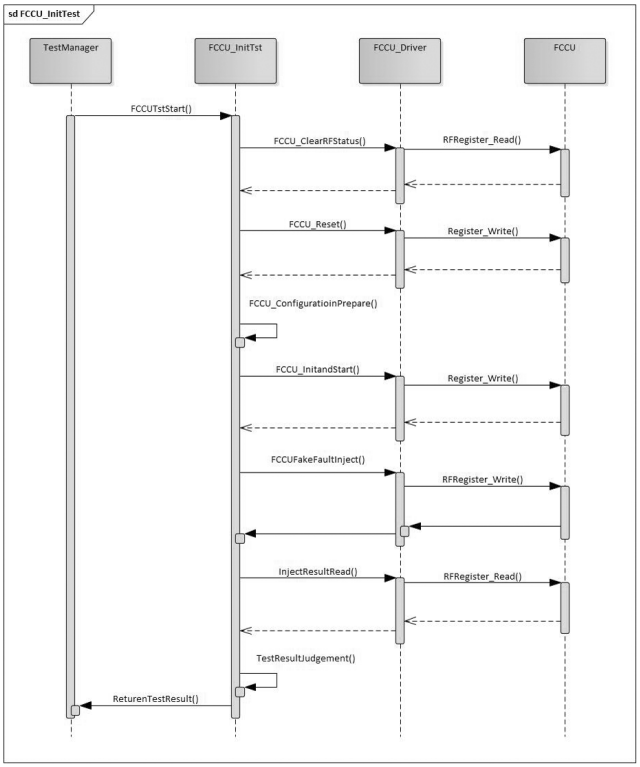


图 5 FCCU 模块 Pre-run 测试软件动态架构

3.2 基于 STPA 的软件安全分析

系统理论过程分析（Systems-Theoretic Processes Analysis, STPA）是基于系统工程原理和控制理论进行危害识别和安全约束设计的分析方法[9]。传统的危害分析方法 FTA 和 FMEA 均是基于可靠性理论去分析一个或多个部件失效所造成的危害。与这些传统方法不同的是，STPA 除了可以分析由部件设计缺陷或者部件之间不安全的交互所造成的事故，还可以用于消除或减轻系统早期设计阶段的潜在危害。STPA 作为一种迭代的方法，使开发与设计同时进行，在设计形成早期就进行分析，并且给出相关的决策对设计进行改进，使系统开发经济有效[10]。STPA 应用于软件的分析，将有助于发现软件当中的动态部分的潜在失效。

通过应用 STPA 对扭矩控制软件系统的分析，得到以下结论。

(1)影响电机控制功能安全的事故类型主要有两种：①当电机控制系统处于激活状态时，发生非预期输出扭矩；②当没有扭矩请求时，电机控制扭矩输出被意外激活。

(2)系统级危害主要有两类：①车辆发生非预期加速；②车辆发生非预期减速。

(3)如图 6 所示，通过电机控制软件系统安全分析控制流程架构，可分析电机控制各个子系统之间的交互安全性。

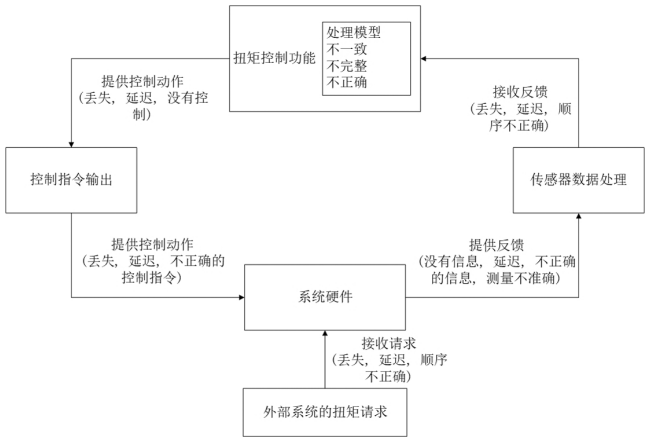


图 6 电机控制软件系统控制流程架构

(4)子系统之间非安全控制动作可分为四类，以输入信号计算为例，第一类是没有提供数据，如没有电流信息和旋变信息；第二类是提供错误数据，如错误的旋变信息；第三类是提供数据的延时较大，如电流信息非实时；第四类是提供的信息不准确，如旋变信息不准确。

(5)安全需求：安全需求分析对应于 ISO26262 中功能安全要求分析过程。使用 STPA 时，每个非安全控制动作将转换成在子系统或部件上的安全需求。如(4)中所述第一类非安全动作转化成安全需求就输入信号处理模块需要处理完整的电流与旋转变压器信息。

(6)分析安全影响因素，解决措施是在软件控制系统当中增加对于输入信号的完整性诊断，以保证在电机控制系统失效的情况下，控制系统可以发现这些错误的输入，进而采取相应动作使系统进入安全状态。

4 软件的实验验证

4.1 软件单元的验证

为了验证 SafeTKit 软件单元的功能及其有效性，搭建了如图 7 所示的软件单元验证实验平台。由于软件单元较多，本文仅列举指令集测试这一个例子。

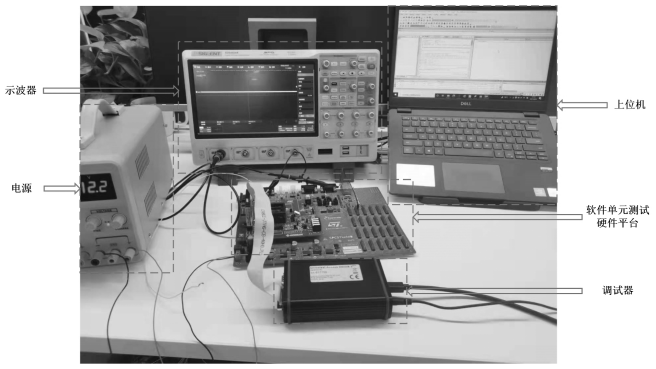


图 7 软件单元验证实验平台

指令集测试通过调用 SafeTKit 当中的 SBST Driver 模块，再配合上用户应用代码，通过逐组测试可在 CPU 上运行的指令集中的指令代码，进而检查 CPU 是否存在影响安全的故障。具体的测试流程如图 8 所示。

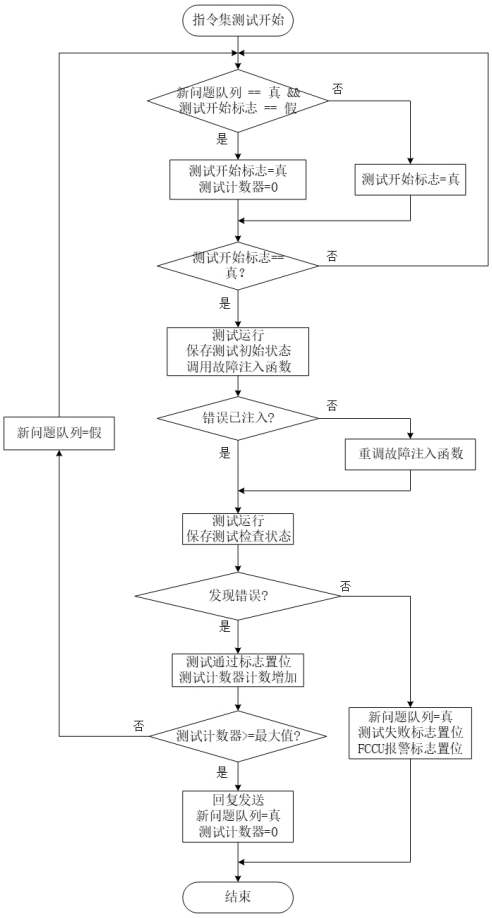
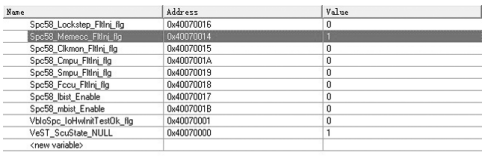


图 8 指令集测试流程

指令集测试以外部看门狗的问题作为触发信号，并将测试结果回复给看门狗。当进行指令集测试时，首先判断是否为新的问题的同时，检查测试开始标志是否置位，若为新的问题且当前测试开始标志未置位，则将指令测试计数器赋值为 0，然后开始运行指令集，某一段指令运行完并保存运行结果后调用故障注入功能函数，当故障注入之后，再次运行该段指令并判断对于当前运行的指令是否检测到故障，若检测到故障，则测试通过，同时指令测试计数器递增；递增之后检查是否大于最大值，如果大于最大值，意味着该组指令测试结束，在结果发送之后置位新问题标志并将计数清零。如果不大于最大值，则接着进行未测试指令的检查。一旦在故障注入后检测不到故障，则认为 CPU 发生异常，则指令测试错误标志置位，同时单片机硬件故障处理单元发送报警信号。

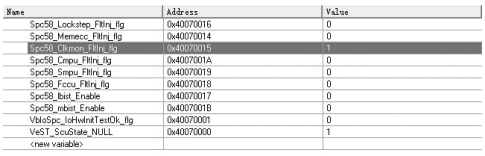
通过修改用户软件，将上述步骤中的错误检测函数修改成屏蔽错误检测功能。因此当指令集测试的错误注入后，无法检测到应该检测到的 CPU 错误，指令集测试功能直接触发了 FCCU 的故障标志，FCCU 的故障输出引脚由 PWM 波被拉高，从而表征了单片机的 CPU 故障，而电机控制器系统也将根据该信号的变化进行更进一步的安全响应，从而避免了 CPU 的单点故障导致的系统安全风险。

类似于指令集测试，通过对内存、时钟以及 FCCU 等单片机硬件模块进行操作，注入故障（图 9~图 11），得到单片机对于故障的响应测试结果如图 12 所示，表明相应的软件单元已满足功能要求，可以有效的完成功能安全自测试。



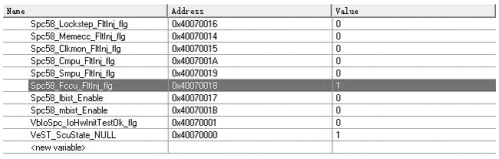
Name	Address	Value
Spc58_Lockstep_FltInj_flg	0x40070016	0
Spc58_Memecc_FltInj_flg	0x40070014	1
Spc58_Clkmon_FltInj_flg	0x40070015	0
Spc58_Cmpu_FltInj_flg	0x4007001A	0
Spc58_Smpu_FltInj_flg	0x40070019	0
Spc58_Fccu_FltInj_flg	0x40070018	0
Spc58_Ibist_Enable	0x40070017	0
Spc58_mbist_Enable	0x4007001B	0
VbloSpc_IoHwInitTestOk_flg	0x40070001	0
VeST_ScuState_NULL	0x40070000	1
<new variable>		

图 9 注入内存故障



Name	Address	Value
Spc58_Lockstep_FltInj_flg	0x40070016	0
Spc58_Memecc_FltInj_flg	0x40070014	0
Spc58_Clkmon_FltInj_flg	0x40070015	1
Spc58_Cmpu_FltInj_flg	0x4007001A	0
Spc58_Smpu_FltInj_flg	0x40070019	0
Spc58_Fccu_FltInj_flg	0x40070018	0
Spc58_Ibist_Enable	0x40070017	0
Spc58_mbist_Enable	0x4007001B	0
VbloSpc_IoHwInitTestOk_flg	0x40070001	0
VeST_ScuState_NULL	0x40070000	1
<new variable>		

图 10 注入时钟故障



Name	Address	Value
Spc58_Lockstep_FltInj_flg	0x40070016	0
Spc58_Memecc_FltInj_flg	0x40070014	0
Spc58_Clkmon_FltInj_flg	0x40070015	0
Spc58_Cmpu_FltInj_flg	0x4007001A	0
Spc58_Smpu_FltInj_flg	0x40070019	0
Spc58_Fccu_FltInj_flg	0x40070018	1
Spc58_Ibist_Enable	0x40070017	0
Spc58_mbist_Enable	0x4007001B	0
VbloSpc_IoHwInitTestOk_flg	0x40070001	0
VeST_ScuState_NULL	0x40070000	1
<new variable>		

图 11 注入 FCCU 故障

4.2 软件集成验证

软件集成测试不仅是保证系统完整性的前提，同时也可以对软件架构及软件设计的符合性进行验证。为了验证扭矩监控功能，本文在多个工况下进行了故

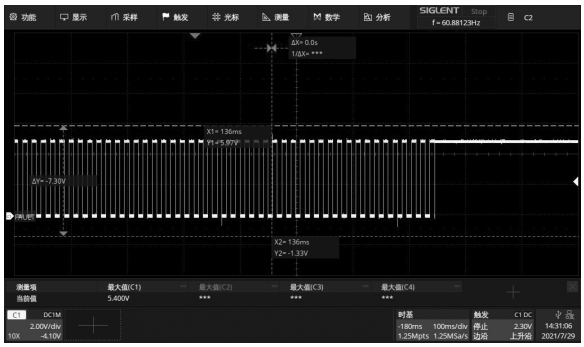


图 12 软件单元故障注入测试结果

障注入测试。该测试通过在软件中制造扭矩控制错误，从而触发扭矩监控功能并使系统进入主动短路的安全状态。基于图 13 的软件集成验证实验平台，500rpm/50Nm 工况的实验结果列举如下。

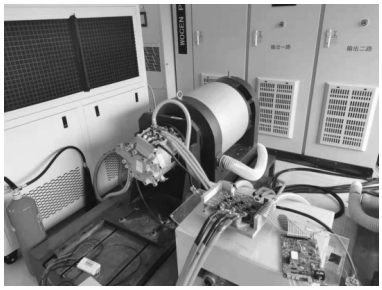


图 13 软件集成验证实验平台

图 14 和图 15 是 500rpm/50Nm 工况下进入主动短路时的三相电流波形以及 dq 轴电流波形。

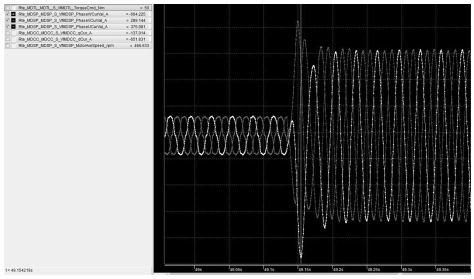


图 14 主动短路时的三相电流波形

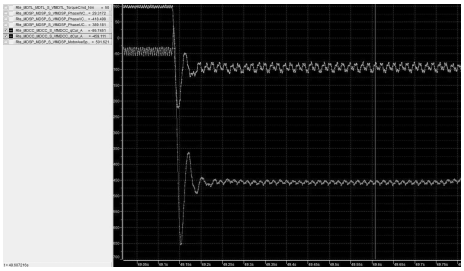


图 15 主动短路时的 dq 轴电流波形

5 结语

本文概述了电机控制器的安全目标以及安全完整性等级的确定，依据此进行了系统架构的概要设计。并根据 ISO26262 功能安全要求，基于 ST 的 SPC 系列多核微控制器，结合 AUTOSAR 方法论，设计了符合功能安全 ASIL D 等级要求的主驱电机控制器软件架构。在软件架构设计中，阐述了软件架构设计中的重要安全机制，包括软件分区和 SafeTKit 功能。在对安全机制进行分析的基础上，阐述了在软件架构级别进行安全分析的实施过程，使用的软件安全分析方法包括 FMEA 分析以及 STPA 分析，与传统的软件安全分析方法不同，本文进行了软件的 STPA 分析，覆盖了软件当中的动态交互部分。最后进行软件的实验验证，软件的实验验证包含指令集测试模块的软件单元的测试和扭矩监控功能的软件集成测试。经过软件安全分析论证以及实验验证，基于该电机控制器软件架构设计的软件可实现相应功能，同时符合功能安全 ASIL D 要求；进一步地，将该软件应用于相应的电机控制器产品也可使整个产品符合功能安全要求。

参考文献

[1] International organization for standardization. Road vehicles - Functional safety: ISO 26262: 2018 [S]. Vernier: International organization for standardization, 2018.

[2] SABBELLA R R, ARUNACHALAM M. Functional safety development of motor control unit for electric vehicles[C]//IEEE. 2019 IEEE Transportation electrification conference. Bengaluru: 2019 IEEE Transportation electrification conference, 2019: 270-276.

[3] PENG Z Y, DU C H, ZHOU A J, et al. Motor torque estimation and security control for electric vehicles (EV) based on parameters feature extraction[J].Research Square, 2021.

[4] BATCHU R. Functional safety in inverter hardware[J]. SAE Technical paper, 2016.

[5] 吴志红，陆科，朱元.车用电机控制器功能安全及主动短路分析[J].同济大学学报，2018(9)：1298-1305.

[6] WANG B, MA K, HUANG X, et al.System -level functional safety testing of MCU based on power - HIL[J]. DEStech transactions on engineering and technology research, 2019(4): 203-210.

[7] 伍理勋，陈建明，陈磊，等.电动汽车电机驱动控制器功能安全架构研究[J].控制与信息技术，2018(3)：1-5，16.

[8] 黄金柱. 异构双处理器系统功能安全设计方法研究 [D]. 武汉：华中科技大学，2012.

[9] LEVESON N G. Engineering a safer world: systems thinking applied to safety[M]. Cambridge: The MIT Press, 2018.

[10] 徐燕钟，德明，尹帅.基于系统理论过程分析的软件安全性分析[J].计算机应用，2013，33(增刊 2)：238-240.

作者简介：王江涛(1971—)，男，汉族，黑龙江哈尔滨人，硕士研究生，高级工程师，主要从事嵌入式系统设计工作。