一般論文

機能安全規格に基づいた車載充電器のソフトウェアアーキテクチャ設計事例

三田事業所　技術第２部　技術第１課
西川　綾、前田　頼正

1. まえがき

近年、自動車業界においては高機能化の要求が求められており、マイコンに搭載するソフトウェアは年々大規模化している。それに伴い複雑になったソフトウェアの品質が製品の安全性に大きく影響を及ぼすようになっている。このような背景から自動車業界における電気/電子（E/E）システムの安全性に関する国際規格として機能安全規格ISO26262が発行された。

当所でソフトウェア開発を行っている車載充電器（OBC: On Board Charger）は、電気自動車（EV）やプラグインハイブリットカー（PHEV）の駆動用高圧バッテリを充電する装置である。EV/PHEVの充電時に充電機能が故障した場合、発煙・発火などを伴う危険事象が発生する可能性がある。そのため、OBCにISO26262への対応要求があり、ISO26262に対応したソフトウェア開発を行っている。

ISO26262では、製品に要求される安全レベルに応じて、プロダクトが備えるべき機能や、プロセスにおいて実施すべき内容が定義されており、これらに基づいて開発を実施する。特徴的な要求事項としてソフトウェアアーキテクチャ設計でのソフトウェア安全分析及びソフトウェア従属故障分析がある。本稿では、ソフトウェアアーキテクチャ設計とソフトウェア安全分析の事例を紹介する。

2. ISO26262概要

機能安全とは「E/Eシステムの機能不全のふるまいにより引き起こされるハザード（危険）が原因となる不合理なリスクの不在」と定義されている。これはE/Eシステムに故障が発生しても、安全機構（SM:Safety Machanism）を設けることでハザードを許容可能なレベルまで低減することを指す。

図1に従ってISO26262の開発フローを説明する。最上位の要求として安全目標（SG:Safety Goal）と、その安全性要求レベル（ASIL：Automotive Safety Integrity Level）が定義される。システムレベルにおける開発では、最初に機能安全コンセプトを検討し、機能安全要求（FSR: Functional Safety Requirement）を定義する。

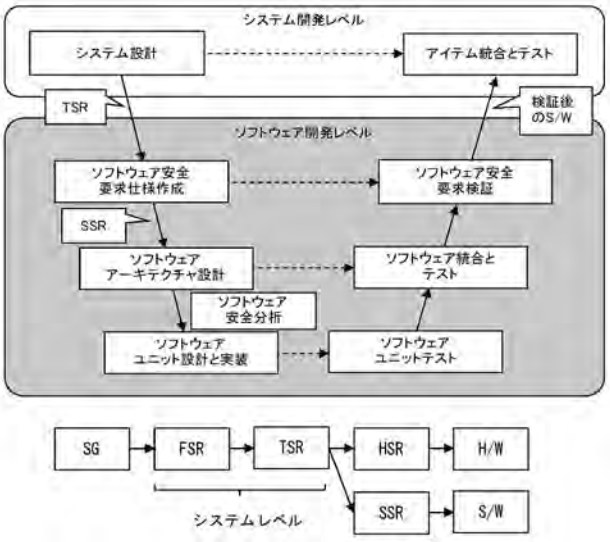

図1. 開発フロー

次にFSRの実現方法を検討し、技術安全要求（TSR:Technical Safety Requirement）として仕様化する。

ソフトウェアレベルにおける開発では、ソフトウェアに割り付けられたTSRを詳細化し、ソフトウェア安全要求（SSR:Software Safety Requirement）を定義する。以降は通常の開発と同様に、ソフトウェアアーキテクチャ設計、ソフトウェアユニット設計と実装を行う流れとなる。SGからSSRまでの仕様例を表1に示す。

表1. 仕様例

項目	仕様例
SG	充放電制御異常に伴う電池の発煙・発火を起こさない。
FSR	OBCはメインリレーカット要求をCAN通信で受信する。
TSR1	CAN信号から通信データに変換を行う。
TSR2	通信データからメインリレーカット要求を受信する。
SSR1	CAN通信機能を用いて受信した通信データから メインリレーカット要求を取得する。
SSR2	メインリレーカット要求を受信した場合、 メインリレー駆動用ポートをOFFする。

ソフトウェアの機能不全によりSSRの要求を満たせなくなると、ハザードに繋がる可能性がある。予めソフトウェア安全分析を行うことで危険事象を抽出し、対策を施すことが要求される。ソフトウェア安全分析によりSGの侵害に繋がる危険事象が見つかった場合は、対策として追加でSMを盛り込み、改めて分析を行う必要がある。

3. ソフトウェアアーキテクチャ設計

今回開発したソフトウェアにおける設計事例を説明する。開発の目的は、充電機能に関連する開発済みのソフトウェアにSMを追加し、ISO26262に準拠させることである。SGとFSRは車両メーカで検討され、TSRへの展開は三菱電機(株)の担当である。本稿では、これらの説明は割愛し、当社が担当したSSRに対するアーキテクチャ設計とソフトウェア安全分析について説明する。

設計方針として、安全が要求される機能と要求されない機能で、関数の集合であるコンポーネント(SWC)を分け、既存機能に対して極力変更を行わないこととした。

最初にSSRのインプットであるTSRを、ASILによって分類した。ASILは一般的な品質管理で許容されるレベル(QM)と、安全機構が必要になるレベル(QM以外)に分類される。QMの場合はISO26262を適用した開発が不要なため、既存のコンポーネントに割り当てた。QM以外の場合は、ISO26262を適用した開発が必要なため、新規に作成するコンポーネントに割り当てた。

ソフトウェアにQMとQM以外のコンポーネントが混在する場合、ISO26262では、両者の間に独立性の確保が要求される。このような構成を実現するため、独立性に関する要求を追加して、TSRからSSRを定義した。

OBCは、ISO26262に対応するために、CPUコアを二つ搭載したマイコンを採用している。既存機能はメインコアであるCPUに配置されているため、新規に作成する安全関連機能はサブコアであるPCU(Peripheral Control Unit)に配置することで、パーティショニングを実施した。以上を踏まえて定義したSSRを表2に、各コアへの配置結果を図2に示す。

表2. SSR例

ID	名称	内容
SSR1	メインリレーカット要求取得	CAN通信機能を用いて受信したCAN通信データからメインリレーカット要求を取得する。
SSR2	CAN通信診断	受信した通信データのCRC(Cyclic Redundancy Check)判定を行いCRCが不一致の場合は、CAN通信データを更新しない。
SSR3	ポート遮断判定	メインリレーカット要求を受信した場合、メインリレー駆動用ポートをOFFする。
SSR4	ポート制御	OFF指示を受けた場合にポートからLo出力を行う。
SSR5	独立性要求	SSR1の信憑性診断にあたるSSR2は独立性を確保する。

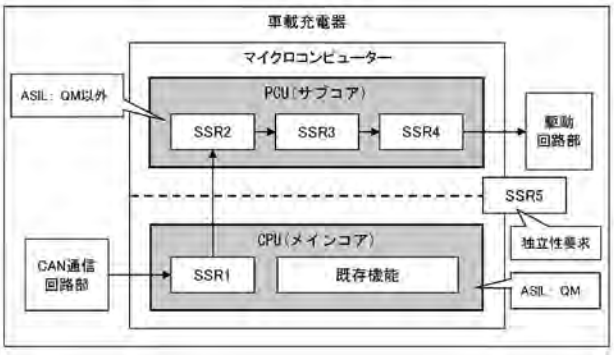

図2. SSR配置例

CPUに配置されている既存機能は、安全に関連するコンポーネントと、そうでないコンポーネントが混在していた。各コンポーネントの機能を分析し、ASILがQMとなるSSRのみを割り当てた。これによって、CPUに配置する既存機能のコンポーネントについては、独立性の確保が不要となるアーキテクチャ設計とした。各コアに対するコンポーネントの配置結果を図3に示す。

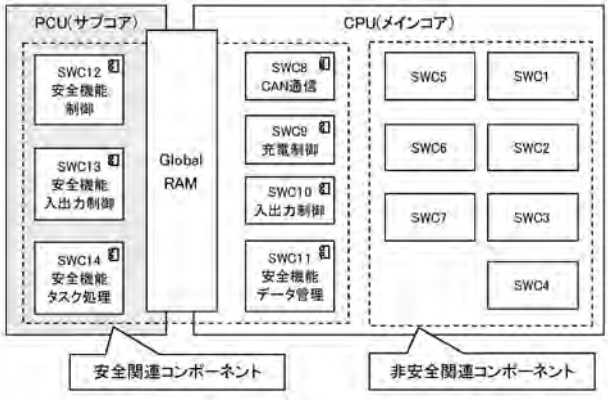

図3. 各コアのコンポーネント配置

PCUとCPUに配置したSSR間の独立性を確保するため、以下に示すアーキテクチャとした。

- CPUとPCU間のデータ授受はGlobal RAMを介してのみ行う。
- CPUからPCUに伝えるデータと、PCUからCPUに伝えるデータを異なる領域に配置する。
- 各領域に対して、各コアはデータの書き込み又は読み込みの一方向のアクセスのみ行う。
- マイコンのメモリプロテクションユニットを使用して、各コアからの許可されていない領域に対する書き込みを防止し、メモリアクセスの独立性を担保する。

Global RAMを介したコア間のデータフローを含む、コンポーネント図を図4に示す。

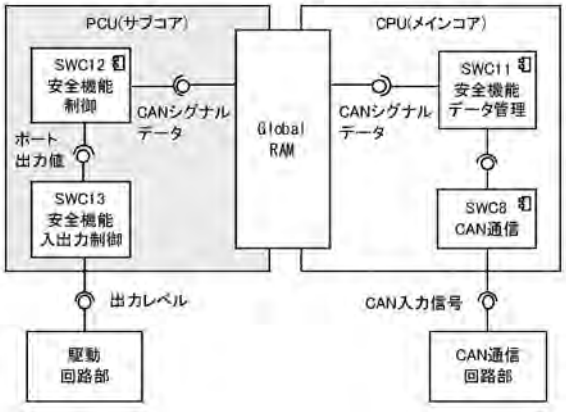

図4. コア間のデータ授受

4. ソフトウェア安全分析

4.1 ソフトウェア安全分析とは

ソフトウェアの機能不全により、SSRの要求を満たせない場合は、システムがハザードに繋がる可能性がある。事前に各コンポーネントについて、SGの侵害に繋がる危険事象を特定し、ソフトウェアアーキテクチャ設計の妥当性やSMに漏れがないことを検証する。

ソフトウェア安全分析の手順を図5に示す。ソフトウェア安全分析及びソフトウェア従属故障分析を行った結果、対策が必要となった場合は新たなSMを追加し、ソフトウェアアーキテクチャを見直す。追加の対策が不要になるまで分析を繰り返し実施する。

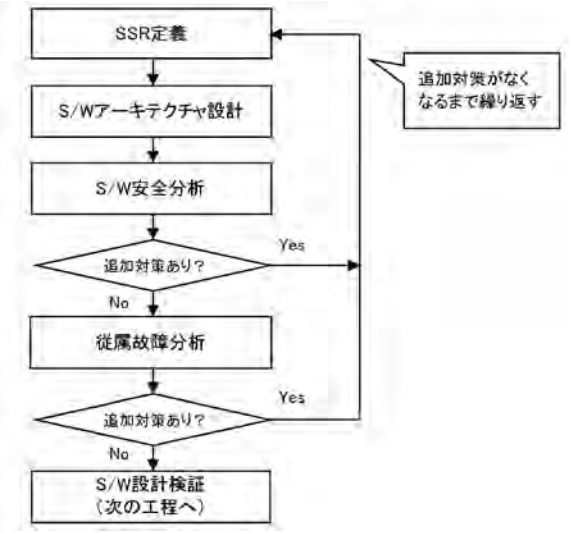

図5. ソフトウェア安全分析手順

4.2 ソフトウェア安全分析事例

ISO26262では、安全分析手法として定性的FMEA、定性的FTA、HAZOP、定性的ETAが示されている。システムやハードウェアは定量的分析手法が適用可能であるが、ソフトウェアは定量的な分析が難しいため、定性的分析手法を用いる。今回の開発では、HAZOPのガイドワードを用いて、ソフトウェアFMEAを実施した。

HAZOP(Hazard and Operability Study)は、分析対象に対し、予め決められたガイドワードを用いて、ハザード及び後工程への影響や検出可能性について分析する手法である。HAZOPのガイドワード例を表3に示す。

表3. HAZOPガイドワード

ガイドワード	意味
No, Not, None	データがない。データを取得できない。
More	データが意図した値より大きくなる。
Less	データが意図した値より小さくなる。
Late	処理が意図した時間より遅く終わる。
Early	処理が意図した時間より早く終わる。
Before	処理の順番が早い。○○する前に△△してしまう。
After	処理の順番が遅い。○○した後に△△してしまう。
Reverse	データが意図せず逆を意味する。符号が反転する。
Part Of	データが複数のうち一部しかない。 複数データでない場合は該当無しとする。 ただし通信データの一部が欠ける場合は該当とする。
As Well As	正常な結果に加え、意図しない結果を指す。 複数データでない場合は該当無しとする。 ただし可変長なデータは該当とする。
Other Than	No, Not, None～As Well Asの観点以外で、 考え得る故障を指す。

HAZOPのガイドワードを用いたソフトウェアFMEAの事例を表4に示す。

表4. ソフトウェアFMEA事例

コンポーネントID	入出力情報	入力出力	HAZOP	故障モード	Safety Goal Violation				新たなSSR	対策
					SG侵害の影響シナリオ	盛り込み済みの対策仕様	対策仕様該当SSR	SG侵害有無		
SWC-12	CANシグナルデータ-メインリレーカット要求1、2	入力	No,Not,None	CPUからシグナルを取得できない。	①CPUからCANシグナルデータを取得できず、SMとして動作出来ない。 ②CPUからメインリレーカット要求1または2のみ取得できない。	①S/Wの対策なし ②取得できたメインリレーカット要求からの遮断要求で遮断処理する。	①- ②(SSR2)	①有 ②無	-	システムとして、上位ECUから充電停止する。
			More	PCUが3hより大きい値を取得する。	CPUから送された受信値より大きい値で動作する。3hより大きければ遮断処理できない。	正しく取得できたCANシグナルデータからの遮断要求で遮断処理する。	(SSR2)	無	-	-
			Less	PCUが1hより小さい値を取得する。	CPUから送された受信値より小さい値で動作する。1hより小さければ遮断処理できない。	正しく取得できたCANシグナルデータからの遮断要求で遮断処理する。	(SSR2)	無	-	-
			Late	PCUのGlobalRAM参照タイミングが遅い。	シグナル取得から遮断処理までを500ms以上かかると要求を満たせない。	S/Wの対策なし	-	有	-	システムとして、上位ECUから充電停止する。
			Early	PCUのGlobalRAM参照タイミングが早い。	遮断処理が早く実施される。 (安全側)	-	-	無	-	-
			Before	データを取得するも更新できず、前回値を使用する。	前回値を取得し続けると正しく遮断処理できない。	S/Wの対策なし	-	有	-	システムとして、上位ECUから充電停止する。
SWC-12	ポート出力値	出力	No,Not,None	信号を出力できない。	出力値を初期値から変更できない。 (安全側)	-	-	無	-	-
			More	有効範囲より大きい値で出力する。	①1hより大きければDC充電コンタクタP側OFFとして処理する。 ②1hの場合は遮断処理できない。	①SWC-13で遮断側に判定する。 (安全側) ②S/Wの対策なし	①(SSR4) ②-	①無 ②有	-	システムとして、上位ECUから充電停止する。
			Less	有効範囲より小さい値で出力する。	0hより小さい値を出力できない。 (安全側)	-	-	無	-	-
			Late	信号の出力が遅れ、CAN信号取得から遮断処理まで設計処理時間より10ms以上遅れて処理する。	シグナル取得から遮断処理までを500ms以上かかると要求を満たせない。	S/Wの対策なし	-	有	-	システムとして、上位ECUから充電停止する。
			Early	信号の出力が早まり、CAN信号取得から遮断処理まで400ms以下で処理する。	遮断処理が早く実施される。 (安全側)	-	-	無	-	-

ソフトウェアFMEAは、以下の(1)～(5)の手順で分析を実施した。

(1) 分析対象の抽出

分析対象となる信号の抽出を行う。本事例では、入出力信号を対象とした。

(2) 故障モードの検討

HAZOPガイドワードに従って、抽出した各信号がどのような故障モードとなるか検討する。

(3) SG侵害の影響シナリオの検討

検討した故障モードから、各SGを侵害する影響シナリオを分析して整理する。

(4) 盛り込み済みの対策仕様の抽出

侵害有となった影響シナリオに対して、既に対策が盛り込まれている場合は、該当するSSRを記入する。

(5) 対策

SG侵害有になった場合は対策を検討する。対策についてはシステムレベルも含めて検討を行い、ソフトウェアレベルで対策する場合は新たなSSRを定義する。

手順(4)はISO26262で要求されていない手順である。当初、ISO26262に従ってソフトウェア安全分析を実施したところ、コンポーネントの機能不全がSGの侵害に結び付く故障モードがあるものの、他のSSRで対策済みであり、ソフトウェアに新たなSMを設ける必要はないとの結果になった。これは、システムレベルの開発において、SG侵害となる故障モードに対して、予めSMを設けているためである。ISO26262では、ソフトウェア安全分析でSMの追加が必要となった場合は、SSRに仕様化することが求められている。今回の事例では、本来導出されるべきSMとそのSSRを明確にするため、手順(4)を追加してソフトウェア安全分析を実施した。

4.3 ソフトウェア従属故障分析

CPUとPCUに配置した安全関連コンポーネント間のパーティショニングが、SSRで定義された独立性要求を満たしているか確認するため、アーキテクチャレベルでソフトウェア従属故障分析を実施した。

従属故障とは一つのコンポーネントの機能不全により、他のコンポーネントも機能不全となるような故障モードを指す。従属故障には、カスケード故障と共通原因故障の二種類の故障モードがある。カスケード故障は一つのコンポーネントの機能不全により、他のコンポーネントが影響を受けて機

能不全となる故障モードを指す。共通原因故障は関連するコンポーネントが共通の原因により、両方とも機能不全となる故障モードを指す。従属故障の例を図6に示す。

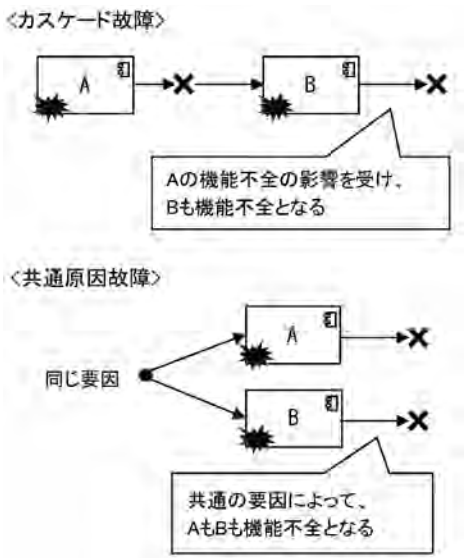

図6. 従属故障例

4.4 ソフトウェア従属故障分析事例

ソフトウェア従属故障分析の事例を表6に示す。ソフトウェア従属故障分析は、ソフトウェアFMEAとは異なり、安全関連コンポーネントが非安全関連コンポーネントの機能不全の影響を受けないことを確認する。このため、全てのコンポーネントに対して分析が必要である。以下の(1)～(3)の手順で分析を実施した。

(1) 従属故障チェック項目の定義

他のコンポーネントとの関連項目を共通リソースとし、想定される従属故障の分類と故障モードを定義する。

表5. ソフトウェア従属故障チェック項目

共通リソース	分類	故障モード
共通ハードウェア	共通原因	メモリ故障 実行時間異常 など
同一タイプ コンポーネント	共通原因	同一アルゴリズム 共通のライブラリ など
共有入力情報	カスケード	関数の引数 共通使用するグローバル変数 など
機能間通信	カスケード	インターフェースの引数 共通使用するグローバル変数へのアクセス データまたはメッセージの破壊/損失/遅延/順序
意図しない影響	カスケード	メモリ異常アクセス メモリバッファオーバーフロー メモリバッファアンダーフロー
システマチックな結合	共通原因	ツールバグ ヒューマンエラー プロセス不備

(2) 影響先コンポーネントの抽出

従属故障チェック項目に基づき、共通原因故障の場合は、リソースを共有する他のコンポーネントを抽出する。カス

表6. ソフトウェア従属故障分析事例

コンポーネントID	共通リソース	影響先コンポーネント		対策		判定
		共通原因故障	カスケード故障			
		コンポーネントID	コンポーネントID	要否	内容	
SWC-8	共通ハードウェア	SWC-9,SWC-10 SWC-11		済	CPU内のROM/RAMチェックにより充電しないため問題なし。	OK
	同一タイプコンポーネント	なし		-		OK
	共有入力情報		なし	-		OK
	機能間通信		SWC-7,SWC-9 SWC-11	済	メインリレーカット要求よりSMから遮断処理するため問題なし。	OK
	意図しない影響		SWC-7,SWC-9 SWC-11	済	システムとして、上位ECUから充電停止する。	OK
	システマチックな結合	SWC-2,SWC-7 SWC-9,SWC-11		要	規程に従い開発を行い、不具合がないことを検証する。	NG
SWC-12	共通ハードウェア	SWC-13,SWC-14		済	メインリレーカット要求による従来機能側故障診断で充電停止するため問題なし。	OK
	同一タイプコンポーネント	なし		-		OK
	共有入力情報		なし	-		OK
	機能間通信		SWC-13	済	メインリレーカット要求による従来機能側故障診断で充電停止するため問題なし。	OK
	意図しない影響		SWC-13,SWC-14	済	メインリレーカット要求による従来機能側故障診断で充電停止するため問題なし。	OK
	システマチックな結合	SWC-13,SWC-14		要	規程に従い開発を行い、不具合がないことを検証する。	NG

ケード故障の場合は、コンポーネントの機能不全により影響を受ける他のコンポーネントを抽出する。

(3) 対策内容の検討

抽出されたコンポーネントの従属故障に対して、対策の要否を判断する。既に対策済みの場合はその対策内容を記載する。新たに対策が必要な場合は、対策内容を検討する。

今回はマイコン機能の活用や既存機能で対策可能な場合は問題なしと判断した。対策が必要と判断された故障モードは、共通原因故障の原因になる開発環境の不具合である。これについては、後工程のデザインレビューとテストによる検証を実施し、問題なしと判断した。

5. むすび

本稿では、ISO26262に対応したソフトウェアアーキテクチャ設計と、ソフトウェア安全分析の事例を紹介した。今回の事例では、マイコンの機能を用いてパーティショニングを行うことで、ISO26262の要求事項に対応したソフトウェアを開発するとともに、既存機能に対する変更を最小化することができた。

ISO26262の対応が求められる開発では、ソフトウェア安全分析やソフトウェア従属故障分析の要求事項を理解し、ソフトウェア安全要求の分析段階で、設計戦略を立てることが重要である。これにより、後工程での仕様追加や設計変更を防止し、効率的な開発が可能となる。

近年、車載機器の開発において、ISO26262の対応した製品開発は当たり前のものとなりつつある。今後、より一層ISO26262に対する理解を深めて開発事例を蓄積し、三菱電機(株)とともに、より安全で効率的な車載機器の開発に貢献していく。

最後に本開発に際して、貴重な御意見、御指導を頂いた三菱電機(株)の方々に深く感謝を申し上げる。

参考文献

(1)DNV GL ビジネス・アシュアランス・ジャパン株式会社,ISO26262エンジニアリングコース ソフトウェア編

執筆者紹介

西川 綾 ニシカワ アヤ
2006年入社。主に車載用充電器のソフトウェア開発に従事。現在、三田事業所技術第２部技術第１課。

前田 頼正 マエダ ヨリマサ
2003年入社。主に車載用充電器のソフトウェア開発に従事。現在、三田事業所技術第２部技術第１課。