Vol.2013-SLDM-160 No.8
Vol.2013-EMB-28 No.8
2013/3/13

車載制御システム向けパーティショニング機構

土本 幸司[1,a)]　川島 裕崇[2,b)]　本田 晋也[2]　高田 広章[1]

概要：昨今の自動車の高機能化に伴い，車載ソフトウェアが大規模・複雑になってきている．その結果，ソフトウェア開発のコストが増大し，大規模なソフトウェアを含む車載システムの安全性を確保することが困難になってきている．こうした背景を受け，高い安全レベルを要求される車載ソフトウェアをできる限り少ないコストで開発するためのパーティショニング機構（DefensiveZone）の開発を行なった．DefensiveZoneでは，小規模なハードウェアをマイコンに付加することによって，あるソフトウェアの不具合が他のソフトウェアへ波及しないように保護を行なう．性能評価においては，DefensiveZone は AUTOSAR OS のメモリ保護機能に比べて半分程度の実行オーバヘッドでソフトウェアの保護を行なえることを確認した．

キーワード：パーティショニング，保護機能，AUTOSAR OS

A Partitioning Mechanism for Automotive Control Systems

Koji TSUCHIMOTO[1,a)]　Hirotaka KAWASHIMA[2,b)]　Shinya HONDA[2]　Hiroaki TAKADA[1]

Abstract: In recent years, automotive software has become more large and more complex with the development of high performance automobiles. As a result, to ensure the safety of the automotive systems including massive software is getting difficult. And the cost of software development has been increasing. From this background, we have developed DefensiveZone which is a partitioning mechanism for development of automotive software which is required high safety level with more lower cost. DefensiveZone provides the protection facilities by appending the small hardware to microcomputer to prevent that some software bugs influence behavior of other software. In performance evaluation, we confirmed that the execution time in DefensiveZone to realize partitioning for software protection is about half compared to AUTOSAR OS with memory protection.

Keywords: Partitioning, Protection Facilities, AUTOSAR OS

1. はじめに

近年，自動車の高機能化が進んでいる．パワーステアリングや ABS（Antilock Brake System）は今では標準装備されるようになり，インテリジェント AFS（Adaptive Front-lighting System）や ACC（Adaptive Cruise Control）などの機能を搭載した自動車が次々に誕生している．また，従来まではメカニカルに行なわれていたアクセルやステアリングなどの制御もバイ・ワイヤ技術によって電子制御化されつつあり，車載ソフトウェアが大規模化・複雑化してきている．このことに伴い，自動車に配置される ECU（Electronic Control Unit）の数も増えてきている．ECU では，エンジンの制御からカーエアコンの室温制御まで，その自動車に関わるあらゆる電気電子機器の制御が行なわれている．しかし，限られた車内スペースに配置することができる ECU の数には限りがある．このため今日では，それまで複数の ECU で行なわれていた処理をひとつの ECU に集約する，ECU 統合が行なわれるようになってきた．ECU 統合によって，車載ソフトウェアは更に大

[1] 名古屋大学大学院情報科学研究科
Furo-cho, Chikusa-ku, Nagoya City, Aichi 464–8601, Japan
[2] 名古屋大学大学院情報科学研究科附属組込みシステム研究センター
Furo-cho, Chikusa-ku, Nagoya City, Aichi 464–8601, Japan
[a)] tsuchimoto@ertl.jp
[b)] hkawashi@ertl.jp

規模かつ複雑になり，ソフトウェアから完全にバグを取り除き，決められた期間で車載ソフトウェアを検証することが困難になりつつある．

統合された ECU では，ひとつの ECU 上で様々な情報が処理されるため，あるソフトウェアで生じたバグが安全性に関わる他のソフトウェアに波及するとシステムの誤動作や制御不能などの不測の事態に陥りかねない．このように今日の自動車業界では，少ない検証コストで自動車の安全性を担保する仕組みが求められている．

車載ソフトウェアに対する既存の保護手法には AUTOSAR OS の保護機能があるが，ソフトウェアの検証コストや実行オーバヘッドが増加するなどの問題点がある．こうした背景を受け，我々はハードウェアを工夫することにより自動車の安全性を担保するパーティショニング機構 DefensiveZone を開発し，その評価を行なった．DefensiveZone では，保護機能を実現するハードウェアを付加することによって車載制御システムの安全性が保証されるべき部分の保護を行なう．

本稿は次のように構成される．1 章はじめに．2 章ではソフトウェア保護の必要性と AUTOSAR OS の保護機能について説明する．3 章では提案する保護手法 DefensiveZone について説明し，4 章で DefensiveZone の要件と仕様・保護の実現方法について述べる．5 章では DefensiveZone の評価について記す．最後の 6 章でまとめを述べて本稿を結ぶ．

2. 準備

2.1 パーティショニング

パーティショニングとは，不具合の伝播からソフトウェアを保護するための概念であり機能安全のひとつである．具体的には，自動車の電気・電子システムに関する機能安全国際規格である ISO26262[1] で「ある設計を実現するための，機能もしくはエレメントの分離」と定められている．また，ISO26262 では，異なる安全レベルが求められるソフトウェアが同一の CPU 上で動作する場合，高い安全レベルが求められるソフトウェアを低いものから保護する必要があると規定されている（図 1）．

2.2 保護機能

本稿で述べる保護機能とは，ソフトウェア（処理単位）の何らかの性質を保護するための機能であり，具体的には時間保護とメモリ保護を指す．図 1 に示すように，パーティショニングなどによって保護機能が提供されないとソフトウェアの暴走や誤動作によって別のソフトウェアの動作に支障を来たす恐れがある．

2.2.1 時間保護

時間保護とは，処理単位（タスクや割込みハンドラ）の実行タイミングやプロセッサ実行時間を保証することである．あるアプリケーション A が別のアプリケーション B の動作を妨げることなく，スケジューリングなどによって与えられたアプリケーション B の利用できる時間に影響を与えないとき，“アプリケーション B は時間的に保護される” もしくは “アプリケーション B に時間保護が提供される” と表現する．

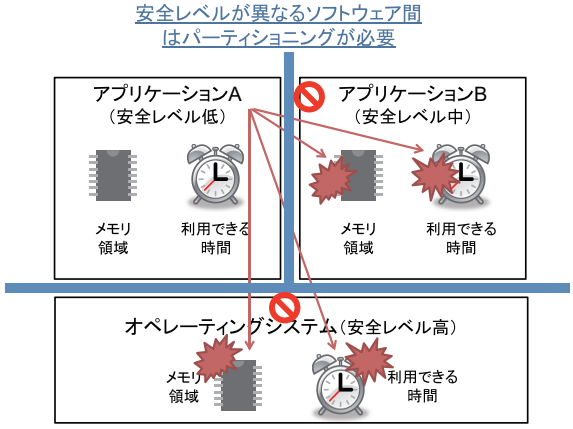

図 1 保護機能の必要性

2.2.2 メモリ保護

メモリ保護とは，処理単位が使用するメモリ領域が不正にアクセスされないことを保証することである．あるアプリケーション A が別のアプリケーション B のメモリ領域に不正にアクセスすることなく，アプリケーション B が使用するメモリ領域に影響を与えないとき，“アプリケーション B のメモリ領域は保護される” もしくは “アプリケーション B にメモリ保護が提供される” と表現する．

2.3 既存のソフトウェア保護手法

ソフトウェアに対する既存の保護手法として，AUTOSAR OS の保護機能 [2] がある．AUTOSAR OS は車載システム向けリアルタイム OS 仕様であり，自動車メーカなどで組織されるコンソーシアムによって標準化が進められている．AUTOSAR OS には，提供される機能の違いから 4 つのスケーラビリティクラス（SC: Scalability Class）が存在し，SC1 が保護機能をもっていない基本クラス，SC3 がメモリ保護機能を有するクラスとなっている．

SC3 では OS アプリケーション（以降，OSAP と記す）というタスクなどの集合の単位でメモリ保護が行なわれる．OSAP には信頼 OSAP と非信頼 OSAP があり，非信頼 OSAP から信頼 OSAP へのアクセスを制限することで信頼 OSAP に属する処理単位の保護が行なわれる．

SC3 の保護機能によってソフトウェアの保護を実現しようとするとき，次のような問題点がある．

- MPU などのハードウェアが必須
- 非信頼 OSAP のデバイスドライバの品質確保
- OS の実行オーバヘッドが増加する

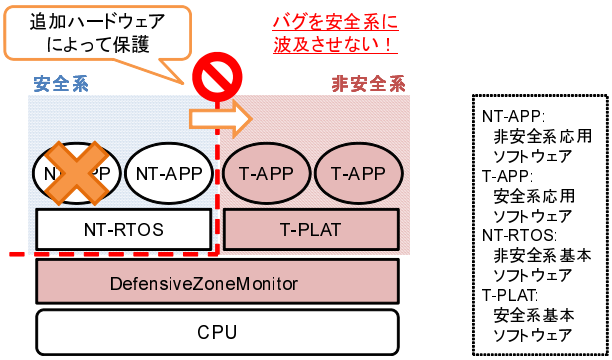

図 2 DefensiveZone の概念

SC3 のメモリ保護機能は MPU（Memory Protection Unit）などの保護機構を備えているマイコン上で実現されるため，使用できるマイコンが一部の高機能なものに限られる．また SC3 では，非信頼 OSAP のデバイスドライバは OS と同じ権限レベルで動作するため，アプリケーションからの切離しや高い品質を確保することが求められる．加えて SC3 では，OS がアクセス権のチェックや OSAP の切替え，MPU の設定などを行なうため，保護機能なし（SC1）の場合と比べ実行オーバヘッドが増加する．

3. 提案手法: DefensiveZone

3.1 概要

DefensiveZone は今回開発した車載制御システム向けのパーティショニング機構である．これは MPU などをもたない安価な既存マイコンに保護機能を実現する小規模のハードウェアを付加することによって，車載制御システムの安全性を担保する仕組みになっている．図 2 に示すように，DefensiveZone では元々ひとつだったシステムを，安全性が保証されるべきパーティション（以降，安全系と記す）と安全性が損なわれても構わないパーティション（以降，非安全系と記す）に分け，ひとつの CPU 上で仮想的に 2 つのシステムを実行している．

追加ハードウェアは主に安全系の保護を行ない，DefensiveZoneMonitor というソフトウェアが安全系/非安全系の切替えを行なっている．追加ハードウェアによってシステムの状態（どちらの系が実行中か）やアクセス権をチェックすることで，非安全系のソフトウェアで不具合が発生しても安全系のソフトウェアに影響しない仕組みになっている．DefensiveZoneMonitor や付加されるハードウェアは小規模であり，使用するマイコンを変更する必要がないため，OS の規模や検証コストが抑えられると期待できる．DefensiveZone は，安全系に対して時間保護機能とメモリ保護機能を提供する．

3.2 用語定義

DefensiveZone を説明する上で使用する用語について以下に定める．

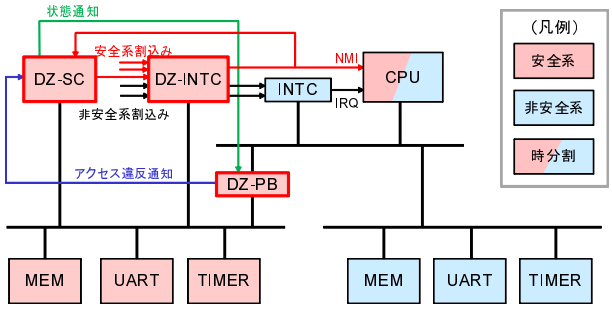

図 3 DefensiveZone のハードウェア構成

システムは図 3 および図 4 のように構成されるものとする．また，システムにおける，安全状態とはシステムの信頼性が保証されている状態とし，非安全状態とはシステムの信頼性が保証されていない状態とする．

安全系ソフトウェアは車載制御システムの安全性に関わるソフトウェアとし，非安全系ソフトウェアは安全系ソフトウェア以外のソフトウェアとする．

安全系メモリは安全系ソフトウェアのコード（以降，バイナリと記す）が配置されているメモリないしメモリ領域とし，非安全系メモリは非安全系バイナリが配置されているメモリないしメモリ領域とする．

安全系割込みは安全系ソフトウェアが使用する割込みでシステムの安全性に関わる割込みとし，非安全系割込みは非安全系ソフトウェアが使用する安全系割込み以外の割込みとする．

安全系デバイスは安全系ソフトウェアが使用する周辺デバイスとし，非安全系デバイスは安全系デバイスに含まれない周辺デバイスとする．

3.3 ハードウェア構成

DefensiveZone のハードウェア構成を図 3 に示す．DefensiveZone は一般的なマイコンの構成に，保護機能を実現するハードウェアを追加することで安全系の保護を行なう．DefensiveZone では，タイマーなどの基本ハードウェアはそれぞれの系に対して専用に割付けられている．以下に追加されたハードウェアについて説明する．

- DZ-SC（ステートコントローラ）
  現在のシステムの状態（安全状態もしくは非安全状態）を管理するレジスタ．安全系割込みが発生するとハードウェア的に安全状態がセットされ，安全系から非安全系に処理が切替わるときにソフトウェア的に非安全状態がセットされる．
- DZ-INTC（安全系割込みコントローラ）
  すべての割込みの入口で安全系割込みと非安全系割込みを受け取る．常に安全系割込みを優先して受付け，安全系割込みは NMI として CPU に通知される．非安全系割込みは後段に接続される INTC（非安全系割込みコントローラ）にデリゲートされる．

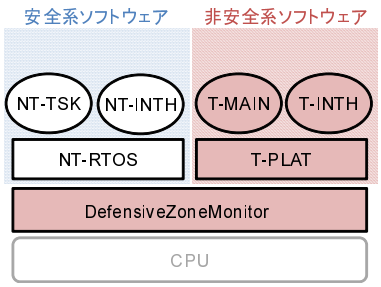

図 4　DefensiveZone のソフトウェア構成

- DZ-PB（プロテクションブリッジ）
  非安全系デバイスと安全系デバイスを隔てるスイッチ．DZ-SC が非安全状態を示しているとき，CPU 側から安全系デバイス側へのアクセスを物理的に禁止する．

3.4　ソフトウェア構成

DefensiveZone のソフトウェア構成を図 4 に示す．

- 安全系ソフトウェア
  小規模で信頼性が担保されているソフトウェア．メインループ処理（T-MAIN）と割込みハンドラ（T-INTH）から構成される．
- 非安全系ソフトウェア
  規模が大きく信頼性の比較的低いソフトウェア．AUTOSAR OS などの保護機能なしの RTOS（NT-RTOS）が動作することを想定している．
- DefensiveZoneMonitor
  安全系/非安全系の切替えを行なうソフトウェア．バイナリは安全系ソフトウェアと共有し，その信頼性は担保されている．

安全系バイナリと非安全系バイナリはそれぞれ分かれており，別のメモリないしメモリ領域に配置される．

3.5　AUTOSAR OS/SC3 との比較

DefensiveZone と AUTOSAR OS/SC3 の相違点を以下にまとめる．SC3 に比べて DefensiveZone は，

- 小規模なハードウェアを追加することで保護を行なう
- ソフトウェアの検証コストを抑えられる
- 保護が限定されている

次に各点について説明する．DefensiveZone は保護機能を実現するハードウェアを付加することでソフトウェアの保護を行なうので，これまでのマイコンを変更することなく保護機能を付与できる．一方，SC3 は MPU や MMU などの機構をもったマイコン上でしか保護機能を提供できない．

SC3 では OS がメモリ保護のための処理を行なうのでソフトウェアが複雑になる．その一方，DefensiveZone は安全系ソフトウェアが十分小さい規模で設計されているため，ソフトウェアの検証コストを少なく抑えることができる．

一方，DefensiveZone では非安全系の保護は行なわれない，安全系はひとつしかもつことができないなどのデメリットがある．これは実アプリケーションにおける保護を考えたとき，SC3 の保護機能が過剰であることが少なくないことから，シンプルに保護機能を実現する機構として DefensiveZone が開発されたためである．保護の仕組みがシンプルであるため，ソフトウェアの保護に関わる実行オーバヘッドを小さく抑えることができる．

4.　DefensiveZone の要件と保護の実現方法

本章では，DefensiveZone の保護機能に関する要件定義とそれに対する要求仕様について述べ，時間保護とメモリ保護の実現方法について説明する．

4.1　要件定義

DefensiveZone が満たすべき性質を以下にまとめる．

要件 1　安全系メモリへは非安全系ソフトウェアからアクセスできないこと

要件 2　安全系ソフトウェアの動作は非安全系ソフトウェアによって妨げられないこと

要件 3　安全系デバイスへは非安全系ソフトウェアからアクセスできないこと

要件 4　安全系ソフトウェアで使用するレジスタは非安全系ソフトウェアによって変更されないこと

4.2　要求仕様

前節の要件を満たすための DefensiveZone に対する要求仕様を以下に定める．

4.2.1　システム一般

システムはひとつの CPU（割込みコントローラを含む）とひとつ以上のメモリならびに周辺デバイスから構成され，同一の CPU 上で安全系ソフトウェアと非安全系ソフトウェアが時分割で実行されること．また，システムは安全状態と非安全状態のいずれかの状態をとり，安全系ソフトウェア実行中は安全状態，非安全系ソフトウェア実行中は非安全状態であること．

4.2.2　メモリ

システムが非安全状態にあるとき，CPU から安全系メモリへのアクセスはできないこと．また，システムの状態のいかんに関わらず，非安全系デバイスから安全系メモリへのアクセスはできないこと．

4.2.3　割込み

非安全系割込みはシステムが非安全状態にあるときに限り受付けられ，その挙動は非安全系の RTOS の仕様に従うこと．また，安全系割込みはシステムの状態のいかんに関わらず受付けられ，安全系割込みハンドラが実行されること．なお，安全系割込みの多重割込みは認めない．

システムが非安全状態にあるとき，実行中処理単位のい

かんに関わらず安全系割込みは即座に受付けられること．また，システムが安全状態にあるとき，メインループ処理実行中であれば安全系割込みは即座に受付けられること．

4.2.4 デバイス

システムが非安全状態であるとき，CPU から安全系デバイスへのアクセスはできないこと．また，システムの状態のいかんに関わらず，非安全系デバイスから安全系デバイスへのアクセスはできないこと．

4.2.5 レジスタ

非安全系ソフトウェアの実行によって，安全系ソフトウェアで使用するレジスタセットが変更されないこと．

4.3 時間保護の実現

DefensiveZone は，安全系ソフトウェアに対して時間保護機能を提供する．これは安全系ソフトウェアの動作は非安全系ソフトウェアの実行によって妨げられることはないことを意味している．DefensiveZone では，安全系ソフトウェア実行中は，すべての非安全系割込みを禁止することで安全系を優先して実行させる．また，安全系ソフトウェアは要求に応じて即座に実行される．これは安全系割込みコントローラ DZ-INTC が常に安全系割込みを優先して受付けることによって実現している．

このため，非安全系の処理単位の実行中に安全系割込みを受けた場合は，非安全系の処理を中断して，安全系割込みハンドラに処理が移行する．

4.4 メモリ保護の実現

DefensiveZone は，安全系ソフトウェアに対してメモリ保護機能を提供する．すなわち，非安全系ソフトウェアから安全系ソフトウェアに対して不正にアクセスできないような仕組みになっている．これはプロテクションブリッジ DZ-PB によって，非安全状態の CPU や非安全系デバイスから安全系メモリや安全系デバイスへのアクセスを物理的に禁止することによって実現している．

非安全状態の CPU から安全系メモリへアクセスがあった場合，アクセス違反が DZ-SC に通知され，DZ-SC から DZ-INTC へアクセス違反の安全系割込みが入りアクセス違反時の処理が行なわれる．

5. DefensiveZone の評価

DefensiveZone の有用性を確認するために，既存手法である AUTOSAR OS/SC3 を用いた場合と DefensiveZone を用いた場合とで次の項目に関して比較評価を行なった．I. コードサイズ．II. ハードウェア規模．III. 実行性能．評価には，RTOS として TOPPERES/ATK2[3] を用いた．TOPPERS/ATK2 は NCES（名古屋大学大学院情報科学研究科附属組込みシステム研究センター）で開発が進められている，AUTOSAR OS 仕様に準拠した次世代車載システム向け RTOS である．ATK2 には保護機能なしのクラス SC1 とメモリ保護機能を有するクラス SC3 がある．

表 1 コードサイズの比較

機能別オブジェクト	コードサイズ [$byte$]		
	SC1	DZ	SC3
AUTOSAR OS の基本機能	28800	28800	35104
安全系ソフトウェア		1536	
DefensiveZoneMonitor		796	
メモリ保護関連			2208
OSAP 関連			6048
合計	28800	31132	43360
対 SC1 増分比		+8.10%	+50.56%

5.1 コードサイズ

ATK2/SC1（保護機能なし）を基準としたとき，DefensiveZone ならびに ATK2/SC3 ではどの程度コードサイズが増加するか評価した．評価結果を表 1 に示す．

表 1 より，ATK2/SC3 ではコード量がおよそ 50%増加するのに対して，DefensiveZone ではコード量の増加が 10%以下に抑えられていることが分かる．SC3 では，比較的コードサイズが大きいメモリ保護関連や OSAP 関連のコードが新たに追加されていることに加え，AUTOSAR OS の基本機能のコードが SC1 と比べて増加している．一方，DefensiveZone では，安全系ソフトウェアのコードと DefensiveZoneMonitor のコードが増えているだけである．これらのコードは 2332[byte]（ソフトウェア全体の 7.49%）と小さく，DefensiveZone では少ないコード量増加でソフトウェアの保護を実現できる結果となった．

5.2 ハードウェア規模

DefensiveZone では，保護機能を実現するために次のハードウェアを付加することはすでに述べた．

- DZ-SC（ステートコントローラ）
- DZ-INTC（安全系割込みコントローラ）
- DZ-PB（プロテクションブリッジ）

これらのハードウェアの追加によってハードウェア規模がどの程度増加するか評価を行なった．本来ならばプロセッサの合成面積と比較したいところだが，今回ターゲットとした CPU がハード IP（Intellectual Property）である Altera 社のエンベデッドプロセッサ NiosII であったためハードウェア面積を比較することが困難だった．そのため今回は，OpenCores ベースの CAN（Controller Area Network）コントローラ（受信メッセージボックス: 32 個，送信メッセージボックス: 16 個）を 90nm プロセスで論理合成したときのハードウェア面積との比較を行なった．評価結果を表 2 に示す．

表 2 より DefensiveZone の追加ハードウェアの面積は CAN コントローラの 15%程度と十分小さく，小規模なハー

表 2　ハードウェア規模の評価結果

合成面積 [um^2]			
CAN コントローラ		DefensiveZone 関連ハードウェア	
OpenCores ベース 受信 MsgBox: 32 送信 MsgBox: 16	308315.45	DZ-SC	1702.61
		DZ-INTC	40750.52
		DZ-PB	2915.54
		合計	45368.67
対 CAN コントローラ比		14.71%	

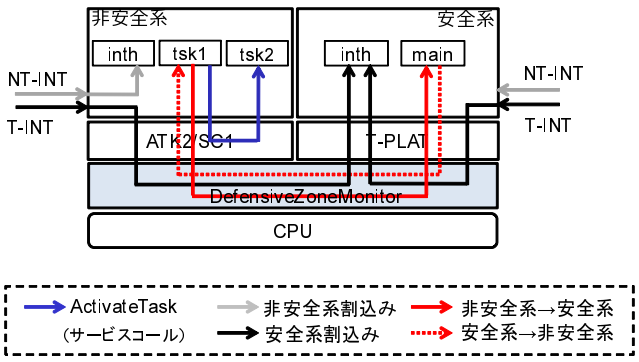

図 5　性能評価の実行パス（DefensiveZone）

ドウェア付加によって保護を実現できることが分かった．

5.3　実行性能

実行性能評価では，次の 3 つの評価項目に対して評価を行なった．①サービスコール実行時間，②割込み応答時間，③系切替え時間．具体的に①では ATK2 のシステムサービスである ActivateTask（タスクの起動）にかかる時間を測定している．これにより OS 実行時間のオーバヘッドを評価している．なお，システムサービスとは RTOS が予め備えもっている，カーネル機能を呼出すための関数である．②では 4 パターンの割込み応答性能（割込み要求が発行されてからハンドラに処理が移行するまでの時間）を評価している．これまで述べてきたように，システムは { 安全系，非安全系 } を実行している状態があり，それぞれ { 安全系，非安全系 } 割込みを受ける可能性がある．このため割込み応答時間としては 4 つの測定パターンが考えられる．③では，2 つのシステム間の処理移行時間を測定している．つまり，安全系→非安全系に処理が移るオーバヘッドとその逆のパターンについて評価を行なっている．①〜③の実行パスは DefensiveZone の場合，図 5 のようになる．

評価結果について表 3 に記す．なお，この性能評価では NiosII/f（動作周波数: 50[MHz]）をターゲット CPU としている．表 3 中の N/A は対応する実現方法が存在しないことを意味している．また，評価項目「系切替え時間」において ATK2/SC3 では 2 つの値が記されているが，これは実現方法の違いによるものである．上段 (1) はシステムサービス ActivateTask によって系を切り替えた場合の値で，下段 (2) は TerminateTask のシステムサービスによって系を切り替えたときの値である．

表 3　性能評価の結果

評価項目		実行時間 [us]	
		DefensiveZone	ATK2/SC3
サービスコール実行時間		13.40	28.92
非安全系割込み応答	安全系実行時	N/A	N/A
	非安全系実行時	7.40	N/A
非安全系割込み応答	安全系実行時	7.70	11.30
	非安全系実行時	7.40	8.82
系切替え時間	非安全系 → 安全系	14.50	26.42(1)
			23.82(2)
	安全系 → 非安全系	8.46	28.12(1)
			24.96(2)

表 3 から分かるように今回評価を行なったすべての項目で，DefensiveZone の実行時間のほうが ATK2/SC3 よりも短く，少ない実行オーバヘッドで保護機能を実現できる結果となった．これはサービスコールを行なうときの手間が ATK2/SC3 では大きいことが大きな要因であると考えられる．SC3 ではサービスコールをソフトウェア割込みで実現しており，割込み後に，OSAP のスタックや CPU モードの切替え，MPU の設定などを行なっている．一方 DefensiveZone では，サービスコールを単なる関数呼出しで実現している．SC3 での OSAP の切替えは基本的にサービスコールによって行なわれるので，系切替え時間においてもサービスコールの実行オーバヘッドが影響してくる結果となった．

6.　むすび

本稿では，ハードウェアを工夫することでソフトウェアを保護する新たな手法 DefensiveZone の提案を行なった．DefensiveZone では，元々ひとつであったシステムを安全系と非安全系にパーティショニングし，付加したハードウェアと DefensiveZoneMonitor によって安全系のプロセッサ実行時間とメモリ領域が保護される．評価の結果，DefensiveZone は AUTOSAR OS/SC3 と比べて次の利点があることを確認できた．コードサイズの増加は微小である，追加ハードウェアは小規模である，実行オーバヘッドを半分程度に抑えられる．このことより，厳しい時間制約とコストパフォーマンスが求められる車載制御システムに DefensiveZone を適用することは有用であると考えられる．

参考文献

[1] ISO : ISO 26262 Road vehicles -Functional safety-, Part1〜9 (2011).
[2] AUTOSAR : Specification of Operating System (online), 入手先 ⟨www.autosar.org/download/R4.0/AUTOSAR_SWS_OS.pdf⟩ (2013.02.08).
[3] TOPPERS プロジェクト : TOPPERS/ATK2(online), 入手先 ⟨www.toppers.jp/atk2.html⟩ (2013.02.08).