
ATK2-SC3-TP の インテグレーションガイドライン

Ver.1.0.0

2016/04/26

Copyright (C) 2015-2016 by Center for Embedded Computing Systems

Graduate School of Information Science, Nagoya Univ., JAPAN

Copyright (C) 2015-2016 by FUJI SOFT INCORPORATED, JAPAN

Copyright (C) 2015-2016 by NEC Communication Systems, Ltd., JAPAN

Copyright (C) 2015-2016 by Panasonic Advanced Technology Development Co., Ltd., JAPAN

Copyright (C) 2015-2016 by Sunny Giken Inc., JAPAN

Copyright (C) 2015-2016 by TOSHIBA CORPORATION, JAPAN

Copyright (C) 2015-2016 by Witz Corporation

Copyright (C) 2015-2016 by AISIN COMCRUISE Co., Ltd., JAPAN

Copyright (C) 2015-2016 by eSOL Co.,Ltd., JAPAN

Copyright (C) 2015-2016 by SCSK Corporation, JAPAN

Copyright (C) 2015-2016 by SUZUKI MOTOR CORPORATION

上記著作権者は、以下の (1)~(3)の条件を満たす場合に限り、本ドキュメント (本ドキュメントを改変したものを含む。以下同じ) を使用・複製・改変・再配布 (以下、利用と呼ぶ) することを無償で許諾する。

- (1) 本ドキュメントを利用する場合には、上記の著作権表示、この利用条件および下記の無保証規定が、そのままの形でドキュメント中に含まれていること。
- (2) 本ドキュメントを改変する場合には、ドキュメントを改変した旨の記述を、改変後のドキュメント中に含めること。ただし、改変後のドキュメントが、TOPPERS プロジェクト指定の開発成果物である場合には、この限りではない。
- (3) 本ドキュメントの利用により直接的または間接的に生じるいかなる損害からも、上記著作権者および TOPPERS プロジェクトを免責すること。また、本ドキュメントのユーザまたはエンドユーザからのいかなる理由に基づく請求からも、上記著作権者および TOPPERS プロジェクトを免責すること。

本ドキュメントは、AUTOSAR (AUTomotive Open System ARchitecture) 仕様に基づいている。上記の許諾は、AUTOSAR の知的財産権を許諾するものではない。AUTOSAR は、AUTOSAR 仕様に基づいたソフトウェアを商用目的で利用する者に対して、AUTOSAR パートナーになることを求めている。

本ドキュメントは、無保証で提供されているものである。上記著作権者およびTOPPERS プロジェクトは、本ドキュメントに関して、特定の使用目的に対する適合性も含めて、いかなる保証も行わない。また、本ドキュメントの利用により直接的または間接的に生じたいかなる損害に関しても、その責任を負わない。

<目次>

1. 概要.....	1
1.1 本文書の目的.....	1
1.2 参考文献.....	1
1.3 環境.....	1
2. 対象アプリケーション.....	3
2.1 概要.....	3
2.2 モデルカーの詳細.....	4
2.2.1 操作系.....	4
2.2.2 制御系.....	6
2.2.3 ボディ系.....	7
3. インテグレーション.....	9
3.1 1ECU 版.....	9
3.1.1 アーキテクチャ.....	9
3.1.2 設計.....	10
3.1.3 TP 機能追加のためのコンフィギュレーション修正方法.....	12
3.2 2ECU 版.....	20
3.2.1 アーキテクチャ.....	20
3.2.2 設計.....	21
3.2.3 TP 機能追加のためのコンフィギュレーション修正方法.....	25
4. 変更履歴.....	37
図 2-1 全体の構成図.....	4
図 2-2 操作系アーキテクチャ.....	5
図 2-3 制御系アーキテクチャ.....	6
図 2-4 ボディ系アーキテクチャ.....	7
図 3-1 1ECU 版アーキテクチャ.....	10
図 3-2 信頼 OSAP と非信頼 OSAP.....	10

図 3-3 Task と Runnable, C2ISR と Process の関係	11
図 3-4 タイムウィンドウの割当	12
図 3-5 Task の設定	13
図 3-6 BuzzerControl 動作フロー	14
図 3-7 ALARM と Task の関係図	15
図 3-8 2ECU 版アーキテクチャ	21
図 3-9 信頼 OSAP と非信頼 OSAP	22
図 3-10 Task と Runnable, C2ISR と Process の関係	23
図 3-11 タイムウィンドウの割当	24
図 3-12 Ecuc_B のタイムウィンドウ割当て	25
図 3-13 Task	26
図 3-14 Task と ALARM の関係	28
表 1-1 参考文書	1
表 1-2 ハードウェア環境	1
表 1-3 構成要素	2
表 1-4 ソフトウェアバージョン	2
表 2-1 操作系のシステム概要	5
表 2-2 制御系のシステム概要	7
表 2-3 ボディ系のシステム概要	8
表 3-1 現在のモデルカー環境	9

1. 概要

1.1 本文書の目的

本文書は、ATK2-SC3-TP を利用し、ECU インテグレーションの際に各 ECU で時間パーティショニング機能を実現するための方法をまとめたガイドラインである。

本文書では、ATK2-SC1 で動作しているモデルカーを題材にして、ATK2-SC3-TP をインテグレーションする際の考え方を記載する。また、モデルカーのコンフィギュレーションファイルを例に挙げ、ATK2-SC3-TP を利用したインテグレーションに必要な設定値を記載する。

1.2 参考文書

本文書から参照している文書、または本文書を理解するために必要な文書を表 1-1 に示す。内容は本文書に包含されていない。

表 1-1 参考文書

文書名	バージョン
ATK2 外部仕様書	Ver.1.2.1
次世代車載システム向け RTE 外部仕様書	Ver.1.3.0
次世代車載システム向け COM 外部仕様書	Ver.1.1.0
1.開発対象アプリケーション	なし
2.開発環境	なし
システム設計書	なし
モデルカー構築マニュアル	なし

1.3 環境

モデルカーのマイコン情報を表 1-2 に示す。

表 1-2 マイコン情報

マイコン	開発環境
RH850/F1L	CCRH

モデルカーの構成要素を表 1-3 に示す。

表 1-3 構成要素

名前	説明
ESC(モータ)	モデルカーの動力.
サーボ	制御信号によってモータの回転軸の角度や速度をコントロールすることが可能.
ECU × 3 個	モデルカーを制御するためのアプリケーションが動作する基盤.
SBDBT5V	5V 入出力対応マイコン基盤. USB の Bluetooth アダプタを使用すると通信モジュールとして使用可能.
Bluetooth USB アダプタ	SBDBT5V に接続することで PS3 コントローラからのデータを受信することが可能.
プレイステーション 3(以下 PS3)コントローラ	モデルカーを操作することが可能.
車載カバー	車体を覆うカバー. ライトが接続されている.
タミヤ製 TLU-01	LED ライトユニット. 接続された LED ライトの点灯を制御する.
タミヤ製 ランプ用 LED	ヘッドランプ用 LED(白色)
	フォグランプ用 LED(白色または黄色)
	ウインカー用 LED(オレンジ色)
	バックランプ用 LED(白色)
	ブレーキランプ用 LED(赤色)
ブザー	起動音, バック音用ブザー
バッテリー	バッテリーにより給電可能.
バッテリー充電器	バッテリーを充電するため.

実際にインテグレーションをおこなったソフトウェアのバージョンを表 1-4 に示す.

表 1-4 ソフトウェアバージョン

ソフトウェア	バージョン	ダウンロードサイト
モデルカー環境	Rev.134	http://dev.toppers.jp/trac_user/contrib/browser/rc_autos_ar_rh850/
atk2-sc3-tp	Ver.1.4.0	https://www.toppers.jp/atk2-e-download.html
a-rtegen	Ver.1.2.2	https://www.toppers.jp/a-rtegen-download.html
a-comstack	Ver.1.2.1	https://www.toppers.jp/a-comstack-download.html
a-can	Ver.1.2.1	https://www.toppers.jp/a-can-download.html

2. 対象アプリケーション

2.1 概要

本節は、「1.開発対象アプリケーション.pdf」から抽出して記載する。

本文書で対象にしているモデルカーは、車両に搭載されている ECU を使用し、PS3 コントローラからの指示を制御することで操作を実現しているものである。

モデルカーの機能は以下 3 種類に分類される。

- ・ 操作系
 - ・ PS3 コントローラからの指示を解析して制御系に送る
 - ・ PS3 コントローラからは SBDBT と Bluetooth により接続され、SBDBT は PS3 コントローラからの指示を RCB3 と呼ばれる形式に変換し、UART を介して操作系を実現する ECU に送る
- ・ 制御系
 - ・ 操作系からの指示により、ボディ制御(各ランプ/ブザー)や車両制御(操舵角/車速)を決定
 - ・ ボディ制御の情報はボディ系に送る
 - ・ 決定した車両制御の情報を元にサーボと ESC(モータ)の制御を行う
 - ・ 制御系を実行するマイコンボードは、モデルカーとサーボと ESC(モータ)に接続されて速度・操舵角を制御可能
- ・ ボディ系
 - ・ 制御系からの指示によりボディの各ランプ/ブザーの ON/OFF を制御する

モデルカーに搭載されている 3 個の ECU に操作系・制御系・ボディ系を配置した時の全体の構成図を図 2-1 に示す。ECU 間通信は CAN を使用する。

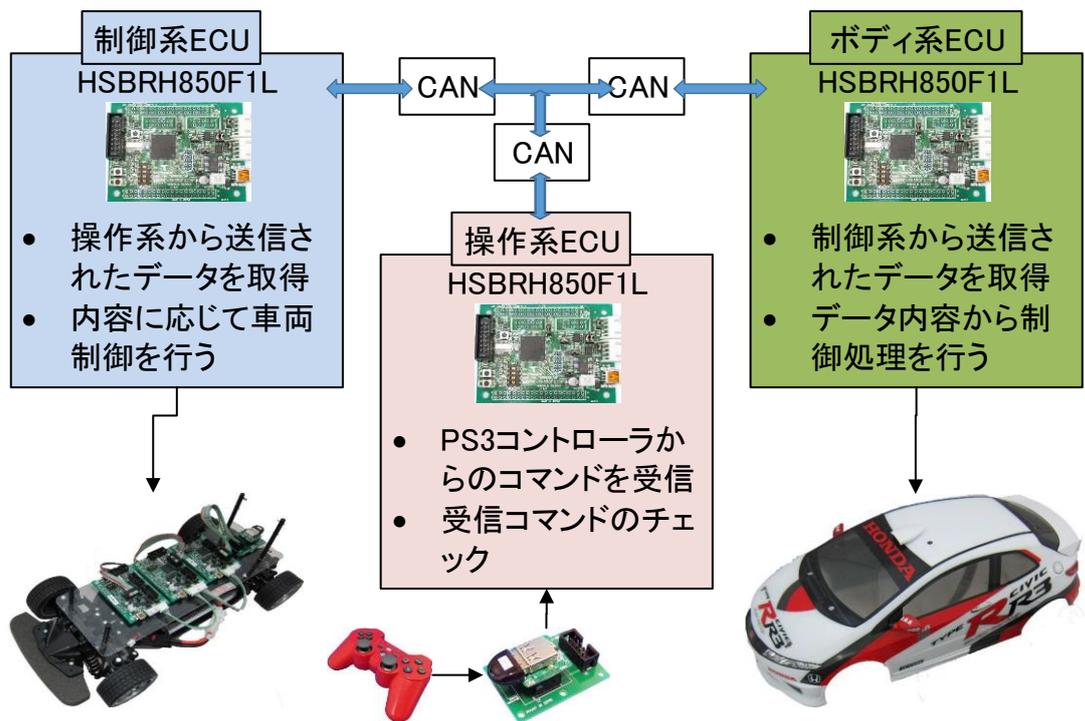


図 2-1 全体の構成図

2.2 モデルカーの詳細

モデルカーを3個のECUで実現した場合の構成を説明する。

SW-Cの種類として、アプリケーション(APP)・複合デバイスドライバー(CDD)が存在する。

2.2.1 操作系

2.2.1.1 アーキテクチャ

操作系のアーキテクチャを図 2-2 に示す。

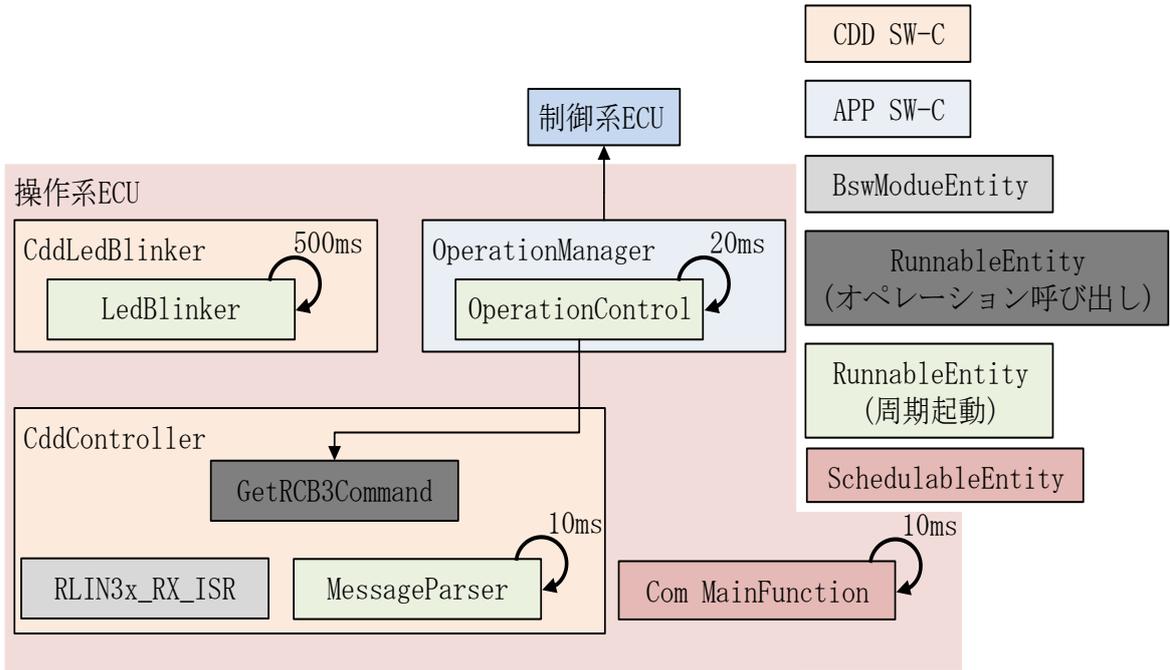


図 2-2 操作系アーキテクチャ

2.2.1.2 システム構成

操作系を構成するシステムの概要を表 2-1 に示す。

表 2-1 操作系のシステム概要

システム名	概要
CddController	UART 経由で RCB3 コントローラから受信したデータ(指示内容)を処理する

	RCB3 コントローラからの指示内容を取得/保持し, 他の SW-C へ提供する
OperationManager	CddController から, RCB3 コントローラからの受信データ(指示内容)を取得する
	RCB3 コントローラからの受信データがある場合, DriveManager(制御系)へ内容を送信する
CddLedBlinker	ECU 動作確認のために, LED を点滅させる
ComMainFunction	COM の MainFunction が動作する SchedulableEntity

2.2.2 制御系

2.2.2.1 アーキテクチャ

制御系のアーキテクチャを図 2-3 に示す.

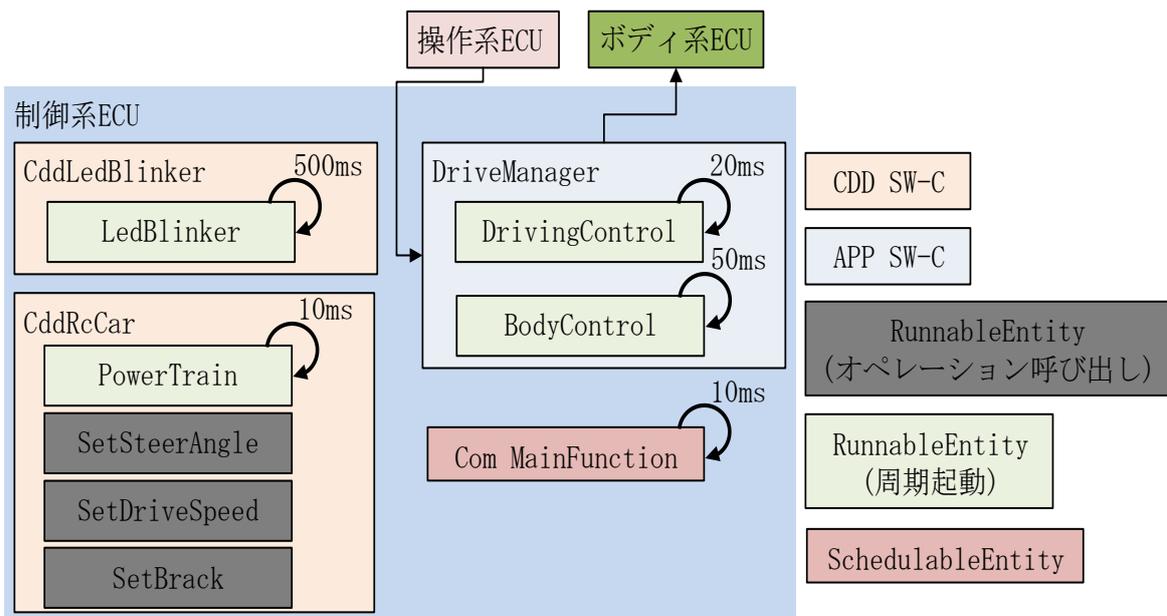


図 2-3 制御系アーキテクチャ

2.2.2.2 システム構成

制御系を構成するシステムの概要を表 2-2 に示す.

表 2-2 制御系のシステム概要

システム名	概要
DriveManager	OperationManager から指示内容を受信する
	受信した指示内容に応じて, CddRcCar へ操舵の制御要求を出す
	受信した指示内容に応じて, CddBodyControl(ボディ系)へ LED, ブザー等の制御要求を送信する
CddRcCar	DriveManager からの制御要求に応じて, モデルカーのサーボやモータを制御する
CddLedBlinker	ECU 動作確認のために, LED を点滅させる
ComMainFunction	COM の MainFunction が動作する SchedulableEntity

2.2.3 ボディ系

2.2.3.1 アーキテクチャ

ボディ系のアーキテクチャを図 2-4 に示す.

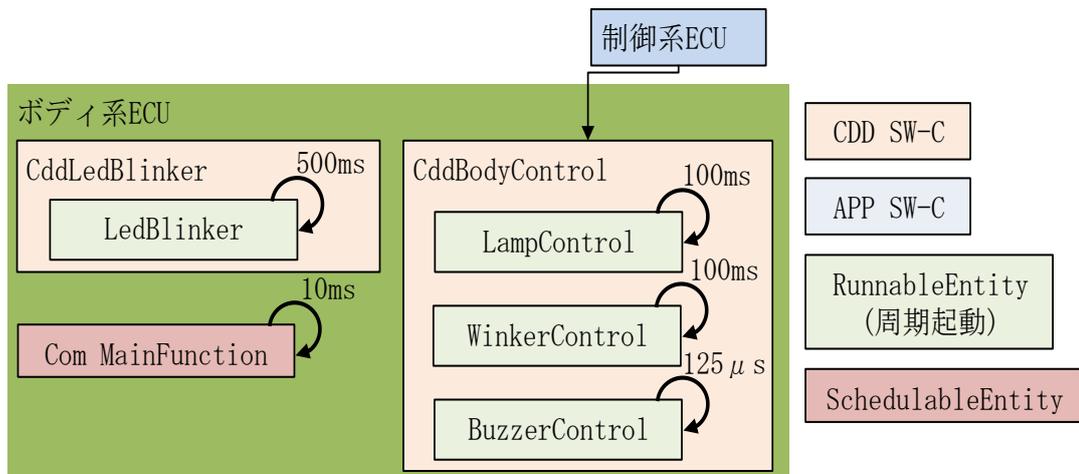


図 2-4 ボディ系アーキテクチャ

2.2.3.2 システム構成

ボディ系を構成するシステムの概要を表 2-3 に示す.

表 2-3 ボディ系のシステム概要

システム名	概要
CddBodyControl	DriveManager(制御系)からの制御要求に応じて、モデルカーの LED やブザー等を制御する
CddLedBlinker	ECU 動作確認のために、LED を点滅させる
ComMainFunction	COM の MainFunction が動作する SchedulableEntity

- CddBodyControl
 - DriveManager(制御系)からの制御要求に応じて、モデルカーの LED やブザー等を制御する
- CddLedBlinker
 - ECU 動作確認のために、LED を点滅させる
- SchedulableEntity(ComMainFunction)
 - COM の MainFunction を動作させる

3. インテグレーション

本章は、ATK2-SC3-TP をインテグレーションする上で必要な設計を行い、設計に基づいて yaml 形式で設定を行う方法を記載する。

モデルカーには 3 個の ECU が備わっており、操作系・制御系・ボディ系をそれぞれの ECU で動作することが可能である。また、操作系・制御系・ボディ系を 1 個、2 個の ECU で動作を分けることも可能である。現在のモデルカー環境には表 3-1 の組み合わせが存在する。ECU(A)、ECU(B)、ECU(C) は別々の ECU とする。

表 3-1 現在のモデルカー環境

ECU 数	ECU(A)	ECU(B)	ECU(C)
1 個	操作系・制御系・ボディ系		
2 個	制御系・操作系	ボディ系	
	操作系	制御系・ボディ系	
3 個	操作系	制御系	ボディ系

本章では上記表の赤字部分を対象に TP 機能を追加する。

1ECU 版では多くの SW-C を 1 個の ECU に統合しており、TP 機能のアプリケーションのパーティション機能を実現できているか確認することを目的としている。

2ECU 版では 1ECU 版には無い ECU 間通信を行うための COM を使用している場合においても、TP 機能を実装できるか確認することを目的としている。

3.1 1ECU 版

3.1.1 アーキテクチャ

1ECU 版アーキテクチャを図 3-1 に示す。1ECU 版では 1 個の ECU にすべての SW-C を統合する。

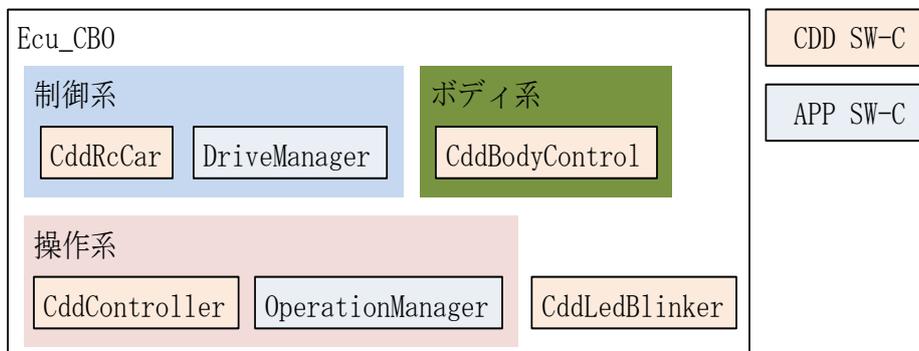


図 3-1 1ECU 版アーキテクチャ

3.1.2 設計

3.1.2.1 信頼 OSAP と非信頼 OSAP

ATK2-SC1 には OSAP の概念がないため、ATK2-SC3-TP のインテグレーションにおいて信頼 OSAP と非信頼 OSAP の設定ポリシーを考える必要がある。

アプリケーション(APP)・複合デバイスドライバ(CDD)には信頼な SW-C と非信頼な SW-C が存在する。ここで、複合デバイスドライバは直接ハードウェアを操作するため本文書では信頼な SW-C として扱う。それに対して、アプリケーションは非信頼な SW-C として扱う。非信頼な SW-C に対しては、その内部で起こった故障が他 SW-C へ伝播しないように、パーティショニングをする必要がある。

上記に従って信頼 OSAP と非信頼 OSAP に分けた場合を図 3-2 に示す。

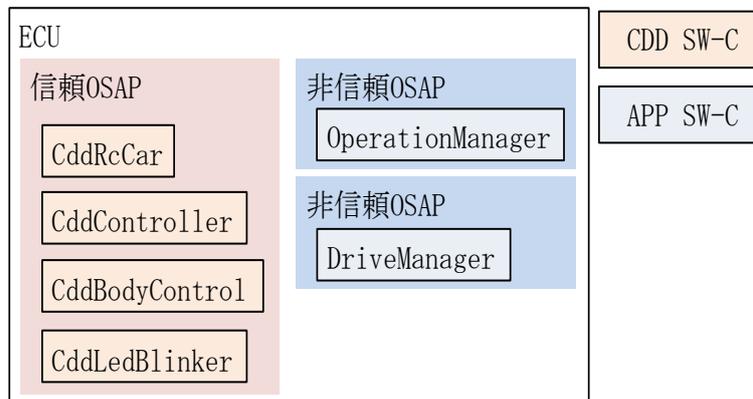


図 3-2 信頼 OSAP と非信頼 OSAP

3.1.2.2 C2ISR にすべきランナブル

ボディ系 CDD SW-C である CddBodyControl には以下の周期起動ランナブルが存在する。

- ・ LampControl : 各ライト制御(100ms 周期起動)
- ・ WinkerControl : 左右のウィンカー制御(100ms 周期起動)
- ・ Buzzercontrol : ブザー制御(125 μ s 周期起動)

ATK2-SC1 において Buzzercontrol ランナブルは 125 μ s 周期で起動するタスクで動作を行っていたが、ATK2-SC3-TP ではシステムサイクルとタイムウィンドウ切り替えオーバーヘッドにより処理が間に合わない可能性がある。また、本来 μ s 単位で起動しなければならない処理については C2ISR で実行すべきである。よって、Buzzercontrol 処理は C2ISR で動作するようする。

3.1.2.3 システムサイクルとタイムウィンドウ

3.1.2.2 において Buzzercontrol 処理は C2ISR で動作することにしたため、 $125\mu s$ で起動していた Task は不要となった。Task と Runnable, C2ISR と Process の関係を図 3-3 に示す。

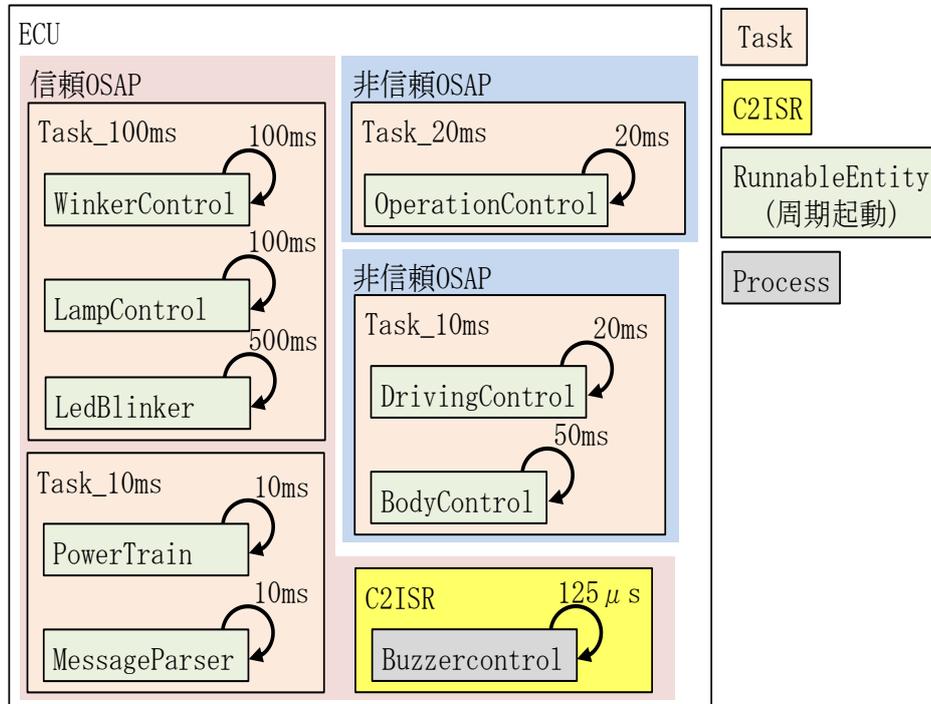


図 3-3 Task と Runnable, C2ISR と Process の関係

本来はランナブルの実行時間を測り、静的解析結果を元にシステムサイクルやタイムウィンドウの個数や長さを決めるべきだが、本文書は、TP 機能が問題なく動作することを目的としているため、システムサイクルとタイムウィンドウの長さは最適化された値ではない。

- ・ システムサイクル

短い起動周期タスクに比べ、システムサイクルを長く設定してしまうとタイムウィンドウの数が増え、データ量が増加してしまう。また、割り込みによる遅延をアイドルウィンドウで吸収しきれずシステムサイクル違反が起きず可能性がある。

システムサイクルはその ECU 内の最小起動 Task を設定することで、タイムウィンドウの数を減らすことが可能となる。ただし、実行時間が足りない場合はシステムサイクルを長く設定する。

1ECU 版での ECU 内の最小起動周期 Task は 10ms であるため、システムサイクル 10ms で設定する。

- ・ タイムウィンドウ

タイムウィンドウは OSAP 毎に設定する必要がある。1ECU 版には非信頼 OSAP が 2 つ、信頼 OSAP が 1 つ、合計 3 つの OSAP が設定されている。それぞれの OSAP にタイムウィンドウを以下のように

に設定する。

- Twd1：非信頼 OSAP(10msTask)が動作するタイムウィンドウ(3ms)
- Twd2：非信頼 OSAP(20msTask)が動作するタイムウィンドウ(2ms)
- Twd3：信頼 OSAP が動作するタイムウィンドウ(4ms)
- Idol：割込みなどの遅延吸収用タイムウィンドウ(1ms)

システムサイクルとタイムウィンドウの設定を図 3-4 に示す。

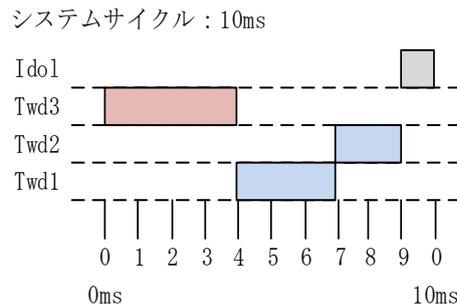


図 3-4 タイムウィンドウの割当

3.1.3 TP 機能追加のためのコンフィギュレーション修正方法

本節では、設計を元に yaml 形式である設定ファイルを修正する方法を記載する。

3.1.3.1 タスク

ATK2-SC1 において以下の Task が設定されている。

- ECU_CBO_TASK_100ms : WinkerControl, LampControl, LedBlinker 動作用 Task
- ECU_CBO_TASK_10ms : PowerTrain, DrivingControl, BodyControl, OperationControl, MessageParser 動作用 Task
- ECU_CBO_TASK_125us : BuzzerControl 動作用 Task
- ECU_INIT_TASK : OS 起動時に初期化する際、一度だけ呼ばれるタスク

Buzzercontrol 処理は C2ISR で動作することにしたため、125 μ s で起動していた ECU_CBO_TASK_125us は削除する。

また、DrivingControl と BodyControl, OperationControl はそれぞれ別の非信頼 OSAP で動作するため、信頼 OSAP に所属する ECU_CBO_TASK_10ms から設定を外す。

DrivingControl と BodyControl が動作する ECU_APP_TASK_10ms, OperationControl が動作する ECU_APP_TASK_20ms を設定する。

ATK2-SC3-TP において以下の Task を設定する。(※赤文字が修正・追加した箇所)

- ECU_APP_TASK_10ms : DrivingControl, BodyControl 動作用 Task
- ECU_APP_TASK_20ms ; OperationControl 動作用 Task
- ~~ECU_CBO_TASK_125us~~ : ~~BuzzerControl~~ 動作用 Task
- ECU_CBO_TASK_100ms ; WinkerControl, LampControl, LedBlinker 動作用 Task
- ECU_CBO_TASK_10ms ; PowerTrain, MessageParser 動作用 Task
- ECU_INIT_TASK : OS 起動時に初期化する際、一度だけ呼ばれるタスク

ATK2-SC1 から AK2-SC3-TP に修正した際の Task を図 3-5 に示す。

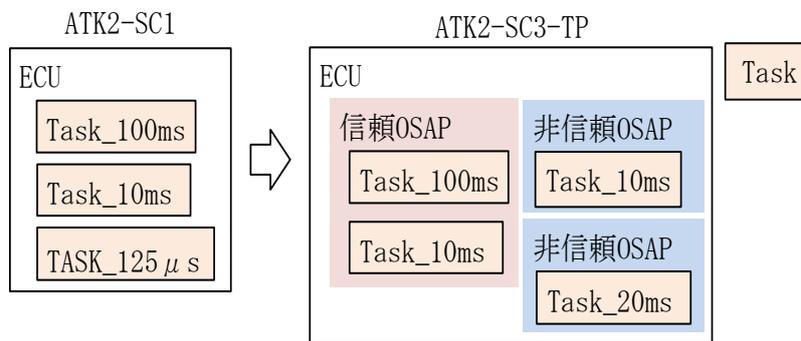


図 3-5 Task の設定

3.1.3.2 C2ISR

Buzzercontrol 処理は C2ISR で動作することにしたため、新たな C2ISR を設定する。ATK2-SC3-TP において以下の C2ISR を設定する。(※赤文字が修正・追加した箇所)

- **C2ISR_FOR_BUZZER** : BuzzerControl 動作用 C2ISR

C2ISR を周期起動するため、ターゲット依存部に target_c2isr_for_buzzer.h を作成し、C2ISR_FOR_BUZZER 周期起動関数を準備する。

モデルカー起動時に C2ISR_FOR_BUZZER 周期起動関数を呼ぶため、Os_Hook.c に target_c2isr_for_buzzer.h のインクルード宣言を追加し、StartupHook において C2ISR_FOR_BUZZER 周期起動関数を呼び出すように修正する。

C2ISR による BuzzerControl が動作するフローを図 3-6 に示す。

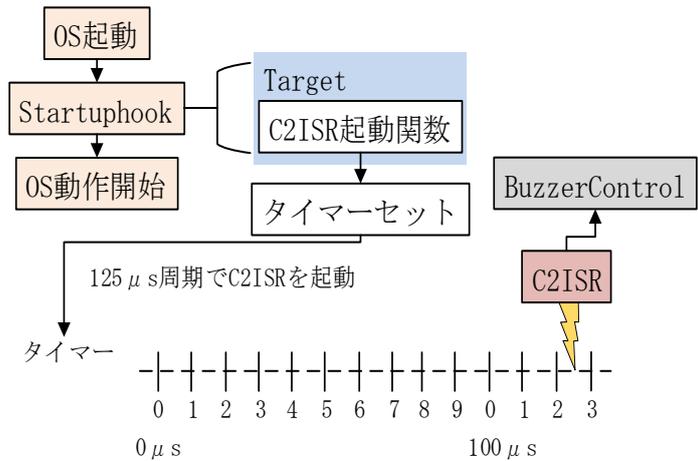


図 3-6 BuzzerControl 動作フロー

3.1.3.3 アラーム

ATK2-SC1 において以下のアラームが設定されている。

- ・ ECU_CBO_ALARM_100ms : 100ms 周期で Task を起こすための ALARM
- ・ ECU_CBO_ALARM_10ms : 10ms 周期で Task を起こすための ALARM
- ・ ECU_CBO_ALARM_125us : 125 μs 周期で Task を起こすための ALARM

Buzzercontrol 処理は C2ISR で動作することにしたため、125 μs で起動していた ECU_CBO_TASK_125us を起動するための ECU_CBO_ALARM_125us は削除する。

10ms と 20ms で動作する Task を新たに設定したため、その Task を起動周期する ALARM を追加する。

ATK2-SC3-TP において以下のアラームを設定する。(※赤文字が修正・追加した箇所)

- ・ ECU_CBO_ALARM_100ms : 100ms 周期で Task を起こすための ALARM
- ・ ECU_CBO_ALARM_10ms : 10ms 周期で Task を起こすための ALARM
- ・ ~~ECU_CBO_ALARM_125us : 125 μs 周期で Task を起こすための ALARM~~
- ・ ECU_APP_ALARM_10ms : 10ms 周期で Task を起こすための ALARM
- ・ ECU_APP_ALARM_20ms : 20ms 周期で Task を起こすための ALARM

ATK2-SC1 から ATK2-SC3-TP に修正した際の ALARM と Task の関係を図 3-7 に示す。

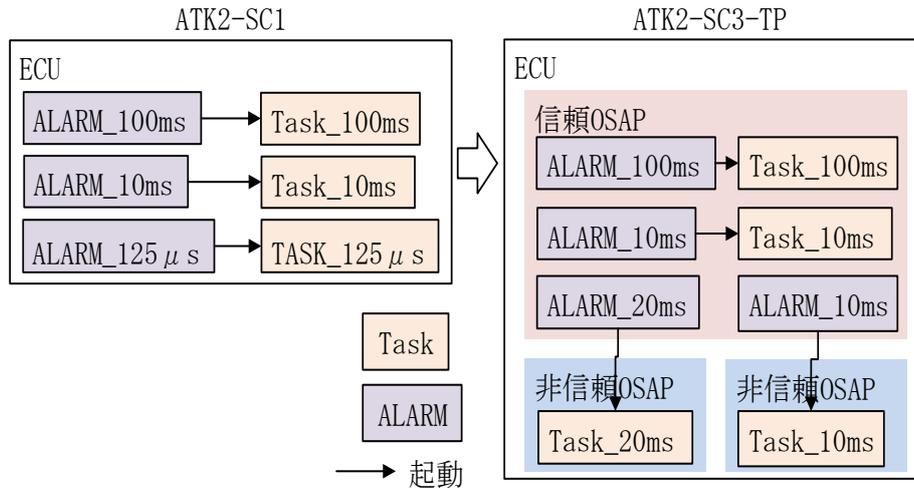


図 3-7 ALARM と Task の関係図

3.1.3.4 SW-C の実体

Rte の設定である RteSwComponentInstance ではランナブルごとの Event に Task をマッピングする必要がある。

ATK2-SC1 において、以下のように設定されている。

- SWCI_CddRcCar
 - IB_CR_TimingEvent_PowerTrain_10ms
 - RteEventRef : PowerTrain_timingEvent
 - RteMappedToTaskRef : ECU_CBO_TASK_10ms
 - RteUsedOsAlarmRef : ECU_CBO_ALARM_10ms
- SWCI_DriveManager
 - IB_DM_TimingEvent_DrivingControl_20ms
 - RteEventRef : DrivingControl_timingEvent
 - RteMappedToTaskRef : ECU_CBO_TASK_10ms
 - RteUsedOsAlarmRef : ECU_CBO_ALARM_10ms
 - IB_DM_TimingEvent_BodyControl_20ms
 - RteEventRef : BodyControl_timingEvent
 - RteMappedToTaskRef : ECU_CBO_TASK_10ms
 - RteUsedOsAlarmRef : ECU_CBO_ALARM_10ms
- SWCI_CddBodyControl
 - IB_CBC_TimingEvent_WinkerControl_100ms

- RteEventRef : WinkerControl_timingEvent
- RteMappedToTaskRef : ECU_CBO_TASK_100ms
- RteUsedOsAlarmRef : ECU_CBO_ALARM_100ms
- IB_CBC_TimingEvent_BuzzerControl_125us
 - RteEventRef : BuzzerControl_timingEvent
 - RteMappedToTaskRef : ECU_CBO_TASK_125us
 - RteUsedOsAlarmRef : ECU_CBO_ALARM_125us
- IB_CBC_TimingEvent_LampControl_100ms
 - RteEventRef : LampControl_timingEvent
 - RteMappedToTaskRef : ECU_CBO_TASK_100ms
 - RteUsedOsAlarmRef : ECU_CBO_ALARM_100ms
- SWCI_OperationManager
 - IB_CM_TimingEvent_OperationControl_20ms
 - RteEventRef : OperationControl_timingEvent
 - RteMappedToTaskRef : ECU_CBO_TASK_20ms
 - RteUsedOsAlarmRef : ECU_CBO_ALARM_20ms
- SWCI_OperationManager
 - IB_CC_TimingEvent_MessageParser_10ms
 - RteEventRef : MessageParser_timingEvent
 - RteMappedToTaskRef : ECU_CBO_TASK_10ms
 - RteUsedOsAlarmRef : ECU_CBO_ALARM_10ms
- SWCI_CddLedBlinker
 - IB_CBL_TimingEvent_LedBlinker_500ms
 - RteEventRef : LedBlinker_timingEvent
 - RteMappedToTaskRef : ECU_CBO_TASK_100ms
 - RteUsedOsAlarmRef : ECU_CBO_ALARM_100ms

ランナブル Buzzercontrol は削除したため、Buzzercontrol 用のマッピング設定を削除する。

DrivingControl と BodyControl は別 OSAP の新たに設定した Task で動作するため RteEvent を修正する必要がある。また、使用するアラームも Ref する必要があるため修正が必要である。

ATK2-SC3-TP において、以下のように設定を行う。(※赤文字が修正・追加した箇所)

- SWCI_CddRcCar
 - IB_CR_TimingEvent_PowerTrain_10ms
 - RteEventRef : PowerTrain_timingEvent
 - RteMappedToTaskRef : ECU_CBO_TASK_10ms
 - RteUsedOsAlarmRef : ECU_CBO_ALARM_10ms

- SWCI_DriveManager
 - IB_DM_TimingEvent_DrivingControl_20ms
 - RteEventRef : DrivingControl_timingEvent
 - RteMappedToTaskRef : ECU_APP_TASK_10ms
 - RteUsedOsAlarmRef : ECU_APP_ALARM_10ms
 - IB_DM_TimingEvent_BodyControl_20ms
 - RteEventRef : BodyControl_timingEvent
 - RteMappedToTaskRef : ECU_APP_TASK_10ms
 - RteUsedOsAlarmRef : ECU_APP_ALARM_10ms
- SWCI_CddBodyControl
 - IB_CBC_TimingEvent_WinkerControl_100ms
 - RteEventRef : WinkerControl_timingEvent
 - RteMappedToTaskRef : ECU_CBO_TASK_100ms
 - RteUsedOsAlarmRef : ECU_CBO_ALARM_100ms
 - ~~• IB_CBC_TimingEvent_BuzzerControl_125us
 - ~~• RteEventRef : BuzzerControl_timingEvent~~
 - ~~• RteMappedToTaskRef : ECU_CBO_TASK_125us~~
 - ~~• RteUsedOsAlarmRef : ECU_CBO_ALARM_125us~~~~
 - IB_CBC_TimingEvent_LampControl_100ms
 - RteEventRef : LampControl_timingEvent
 - RteMappedToTaskRef : ECU_CBO_TASK_100ms
 - RteUsedOsAlarmRef : ECU_CBO_ALARM_100ms
- SWCI_OperationManager
 - IB_CM_TimingEvent_OperationControl_20ms
 - RteEventRef : OperationControl_timingEvent
 - RteMappedToTaskRef : ECU_APP_TASK_20ms
 - RteUsedOsAlarmRef : ECU_APP_ALARM_20ms
- SWCI_OperationManager
 - IB_CC_TimingEvent_MessageParser_10ms
 - RteEventRef : MessageParser_timingEvent
 - RteMappedToTaskRef : ECU_CBO_TASK_10ms
 - RteUsedOsAlarmRef : ECU_CBO_ALARM_10ms
- SWCI_CddLedBlinker
 - IB_CBL_TimingEvent_LedBlinker_500ms
 - RteEventRef : LedBlinker_timingEvent
 - RteMappedToTaskRef : ECU_CBO_TASK_100ms
 - RteUsedOsAlarmRef : ECU_CBO_ALARM_100ms

3.1.3.5 RteOsInteraction

RteOsInteraction では、Rte と OS 間の連携の設定を行う項目である。この項目では期待する起動オフセット、起動周期、ランナブルの起動のために使用する OS アラームを設定する。

ATK2-SC1 において、以下の設定がされている。

- RteUsedOsActivation1
 - RteExpectedTickDuration : 0.01
 - RteActivationOsAlarmRef : ECU_CBO_ALARM_10ms
- RteUsedOsActivation2
 - RteExpectedTickDuration : 0.000125
 - RteActivationOsAlarmRef : ECU_CBO_ALARM_125us
- RteUsedOsActivation3
 - RteExpectedTickDuration : 0.1
 - RteActivationOsAlarmRef : ECU_CBO_ALARM_100ms

RteUsedOsActivation2 は起動周期 125 μ s の Alarm の設定のため、削除する。

新たに設定した ECU_APP_ALARM_10ms と ECU_APP_ALARM_20ms の設定を追加する。

ATK2-SC3-TP において、以下の設定を行う。

- RteUsedOsActivation1
 - RteExpectedTickDuration : 0.01
 - RteActivationOsAlarmRef : ECU_CBO_ALARM_10ms
- RteUsedOsActivation2
 - RteExpectedTickDuration : 0.1
 - RteActivationOsAlarmRef : ECU_CBO_ALARM_100ms
- RteUsedOsActivation3
 - RteExpectedTickDuration : 0.01
 - RteActivationOsAlarmRef : ECU_APP_ALARM_10ms
- RteUsedOsActivation4
 - RteExpectedTickDuration : 0.02
 - RteActivationOsAlarmRef : ECU_APP_ALARM_20ms

3.1.3.6 パーティション

EcucPartitionCollection では、OSAP によるパーティション構成の設定を行う。設定項目にはパーティション内で動作する SW-C を設定する必要がある。

ATK2-SC1 において、OSAP の概念がないためパーティションの設定はない。

ATK2-SC3-TP において以下の SW-C の設定を行う。

- EcucPartition_NT_OM
 - OperationManager
- EcucPartition_NT_DMBC
 - DriveManager
- EcucPartition_T
 - CddController
 - CddRcCar
 - CddBodyControl
 - CddLedBlinker

3.1.3.7 OSAP

ATK2-SC1 には OSAP の概念がないため設定がない。

OSAP の設定には、信頼 or 非信頼か、Alarm、Task、Isr、Partition、TimeWindow、Partition の Ref を行う必要がある。

ATK2-SC3-TP において以下の設定を行う。

- ECU_CBO_TOSAP
 - OsTrusted : 信頼 OSAP
 - OsAppAlarmRef
 - ECU_CBO_ALARM_100ms
 - ECU_CBO_ALARM_10ms
 - ECU_APP_ALARM_10ms
 - ECU_APP_ALARM_20ms
 - OsAppTaskRef
 - ECU_CBO_TASK_100ms
 - ECU_CBO_TASK_10ms
 - ECU_INIT_TASK
 - OsAppIsrRef
 - RLIN3x_RX_ISR
 - C2ISR_FOR_BUZZER
 - OsAppEcucPartitionRef : EcucPartition_T
 - OsSystemCycleTimeWindowRef : TwdCBO
- ECU_CBO_NTOSAP_OM
 - OsTrusted : 非信頼 OSAP
- OsAppTaskRef : ECU_APP_TASK_20ms

- OsAppEcucPartitionRef : EcucPartition_NT_OM
- OsSystemCycleTimeWindowRef : TwdAppOM
- ECU_CBO_NTOSAP_DMBC
 - OsTrusted : 非信頼 OSAP
 - OsAppTaskRef : ECU_APP_TASK_10ms
 - OsAppEcucPartitionRef : EcucPartition_NT_DMBC
 - OsSystemCycleTimeWindowRef : TwdAppDMBC

3.1.3.8 システムサイクルとタイムウィンドウ

ATK2-SC1 において、システムサイクルの概念がないため設定はない。

システムサイクルの設定はその ECU 内の最小起動周期 Task に設定するほうがよいため、最小起動周期である 10ms に設定する。

タイムウィンドウの設定にはそのタイムウィンドウの長さや開始時間を設定する必要がある。また、必ずアイドル時間を設ける必要があるため、最後のタイムウィンドウの終了時間がシステムサイクル時間より短くする必要がある。

ATK2-SC3-TP において、以下のように設定する。

- OsSystemCycle
 - システムサイクル : 0.01
- TwdCBO
 - TW の長さ : 0.004
 - TW の開始時間 : 0.0
- TwdAppDMBC
 - TW の長さ : 0.003
 - TW の開始時間 : 0.004
- TwdAppOM
 - TW の長さ : 0.002
 - TW の開始時間 : 0.007

3.2 2ECU 版

3.2.1 アーキテクチャ

2ECU 版アーキテクチャを図 3-8 に示す。2ECU 版では 2 個の ECU に SW-C を分けてコンフィギュレーションを行う。

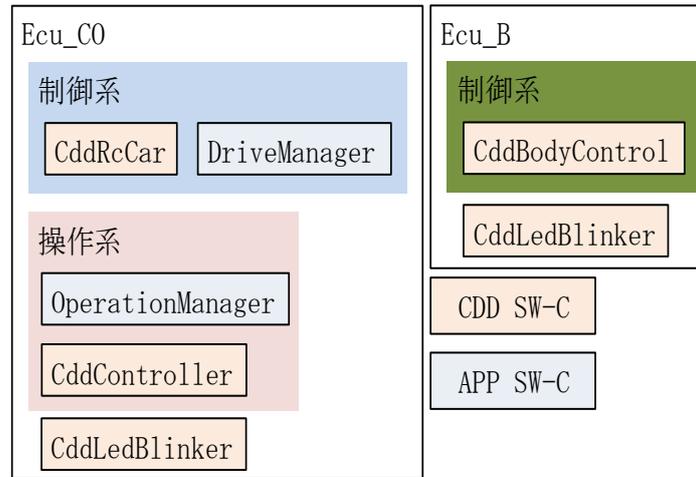


図 3-8 2ECU 版アーキテクチャ

3.2.2 設計

3.2.2.1 信頼 OSAP と非信頼 OSAP

1ECU 版と同じ考え方で APP SW-C は非信頼 OSAP・CDD SW-C は信頼 OSAP とする。
上記に従って信頼 OSAP と非信頼 OSAP に分けた場合を図 3-9 に示す。

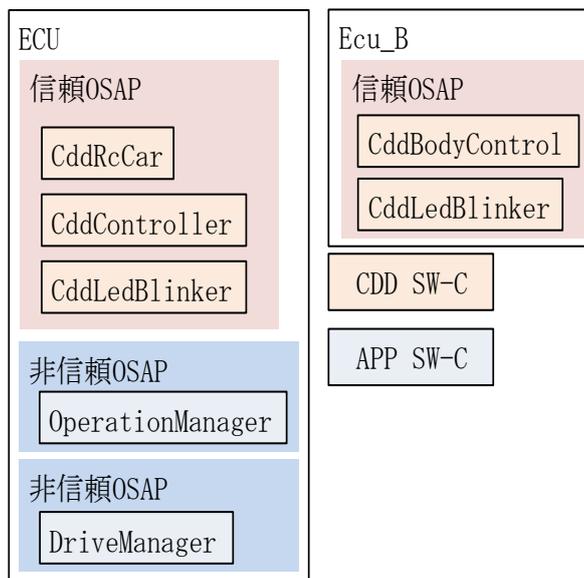


図 3-9 信頼 OSAP と非信頼 OSAP

3.2.2.2 C2ISR にすべきランナブル

1ECU 版と同様に Buzzercontrol 処理は C2ISR で動作するように修正する。

3.2.2.3 システムサイクルとタイムウィンドウ

1ECU 版と同様に $125\mu\text{s}$ で起動していた Task は不要となった。1ECU 版と違い COM 通信するための SheduleEntity が ECU 毎に設定されている。Task と Runnable, C2ISR と Process の関係を図 3-10 に示す。

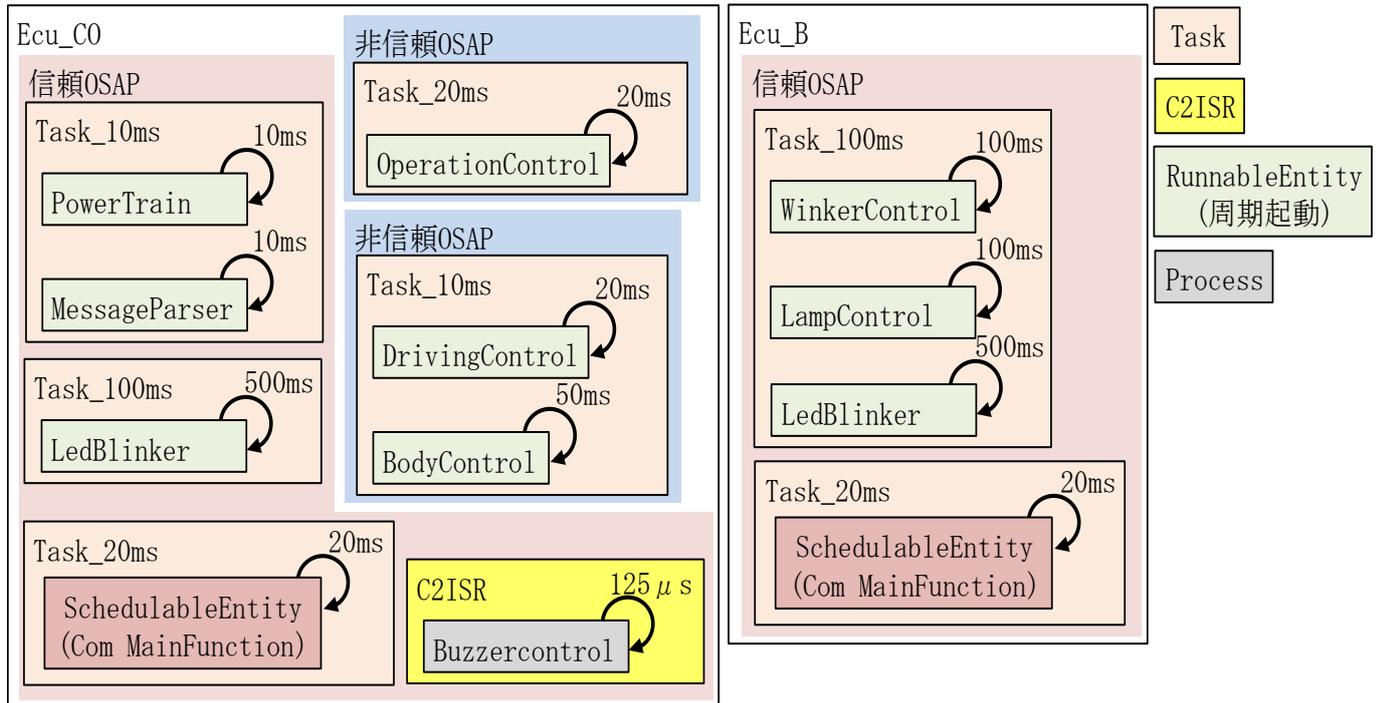


図 3-10 Task と Runnable, C2ISR と Process の関係

1ECU 版と同様に TP 機能が問題なく動作しているか確認することを目的としているため、システムサイクルとタイムウィンドウは最適化された値ではない。

・ Ecuc_CO

・ システムサイクル

OSAP に複数のタイムウィンドウを設定しても動作することを確認するため、20ms にシステムサイクルを設定する。こうすることで、10ms で起動しなければならない OSAP に 2 個のタイムウィンドウを割り当てる必要がある。

・ タイムウィンドウ

タイムウィンドウは OSAP 毎に設定する必要がある。Ecuc_CO には非信頼 OSAP が 2 つ, 信頼 OSAP が 1 つ, 合計 3 つの OSAP が設定されている。OSAP に複数のタイムウィンドウを設定しても動作に問題がないか確認するため、それぞれの OSAP にタイムウィンドウを以下のように設定する。

- ・ Twd1 : 非信頼 OSAP(10msTask)が動作するタイムウィンドウ(2.5ms)
- ・ Twd2 : 非信頼 OSAP(10msTask)が動作するタイムウィンドウ(2.5ms)
- ・ Twd3 : 非信頼 OSAP(20msTask)が動作するタイムウィンドウ(2.5ms)
- ・ Twd4 : 信頼 OSAP が動作するタイムウィンドウ(5ms)

- Twd5 : 信頼 OSAP が動作するタイムウィンドウ(5ms)
- Idol : 割込みなどの遅延吸収用タイムウィンドウ(2ms)

Ecuc_CO でのシステムサイクルとタイムウィンドウの設定を図 3-11 に示す。

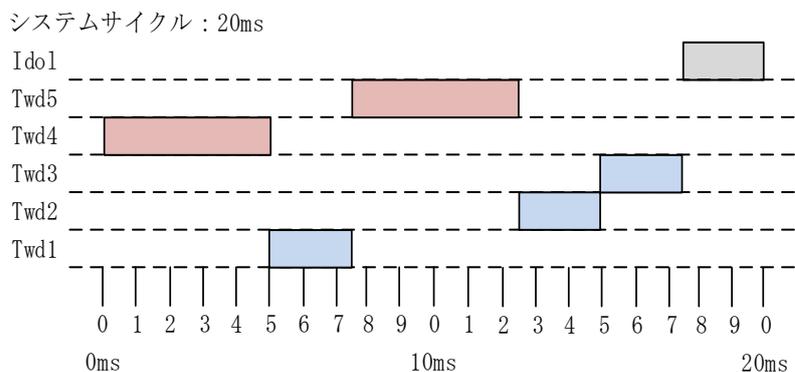


図 3-11 タイムウィンドウの割当

- Ecuc_B
 - システムサイクル

Ecuc_B での ECU 内の最小起動周期 Task は 20ms であるため, システムサイクル 20ms で設定する。

- タイムウィンドウ

タイムウィンドウは OSAP 毎に設定する必要がある. Ecuc_B には信頼 OSAP が 1 つ設定されている. それぞれの OSAP にタイムウィンドウを以下のように設定する.

- Twd1 : 信頼 OSAP が動作するタイムウィンドウ(15ms)
- Idol : 割込みなどの遅延吸収用タイムウィンドウ(5ms)

Ecuc_B でのシステムサイクルとタイムウィンドウの設定を図 3-12 に示す。

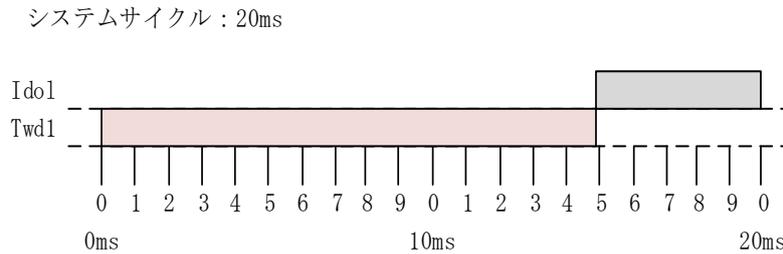


図 3-12 Ecuc_B のタイムウィンドウ割当て

3.2.3 TP 機能追加のためのコンフィギュレーション修正方法

本節では、設計を元に yaml 形式である設定ファイルを修正する方法を記載する。

3.2.3.1 タスク

ATK2-SC1 において以下の Task が設定されている。

- Ecuc_B
 - ECU_B_TASK_100ms : WinkerControl, LampControl, LedBlinker 動作用 Task
 - ECU_B_TASK_125us : BuzzerControl 動作用 Task
 - ECU_INIT_TASK : OS 起動時に初期化する際、一度だけ呼ばれる Task
 - ECU_B_SCHM_TASK : COM の MainFunction 動作用 Task
- Ecuc_CO
 - ECU_CO_TASK_10ms : PowerTrain, DrivingControl, BodyControl, OperationControl, MessageParser 動作用 Task
 - ECU_CO_TASK_100ms : LedBlinker 動作用 Task
 - ECU_INIT_TASK : OS 起動時に初期化する際、一度だけ呼ばれる Task
 - ECU_CO_SCHM_TASK : COM の MainFunction 動作用 Task

1ECU 版と同様に Buzzercontrol 処理は C2ISR で動作することにしたため、Ecuc_B に設定されている 125 μ s で起動していた ECU_B_TASK_125us は削除する。

また、DrivingControl と BodyControl, OperationControl はそれぞれ別の非信頼 OSAP で動作するため、信頼 OSAP に所属する ECU_CBO_TASK_10ms から設定を外す。

DrivingControl と BodyControl が動作する ECU_CO_TASK_10ms, OperationControl が動作する ECU_CO_TASK_20ms を設定する。

ATK2-SC3-TP において以下の Task を設定する。(※赤文字が修正・追加した箇所)

- Ecuc_B

- ECU_B_TASK_100ms : WinkerControl, LampControl, LedBlinker 動作用 Task
- ECU_INIT_TASK : OS 起動時に初期化する際、一度だけ呼ばれる Task
- ECU_B_SCHM_TASK : COM の MainFunction 動作用 Task
- Ecuc_CO
 - ECU_CO_TASK_10ms : DrivingControl, BodyControl 動作用 Task
 - ECU_CO_TASK_20ms : OperationControl 動作用 Task
 - ECU_CO_TASK_100ms : LedBlinker 動作用 Task
 - ECU_INIT_TASK : OS 起動時に初期化する際、一度だけ呼ばれる Task
 - ECU_CO_SCHM_TASK : COM の MainFunction 動作用 Task

ATK2-SC1 から AK2-SC3-TP に修正した際の Task を図 3-13 に示す。

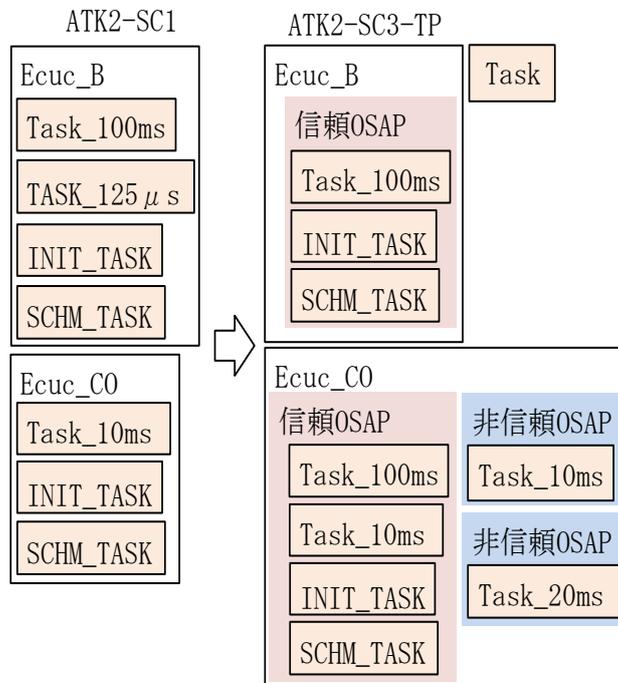


図 3-13 Task

3.2.3.2 C2ISR

3.1.3.2 と同様の修正を行う。

Ecuc_B にブザー処理用 C2ISR の設定を追加する。(※赤文字が修正・追加した箇所)

- **C2ISR_FOR_BUZZER** : BuzzerControl 動作用 C2ISR

3.2.3.3 アラーム

ATK2-SC1 において以下のアラームが設定されている。

- Ecuc_B
 - ECU_B_ALARM_100ms : 100ms 周期で Task を起こすための ALARM
 - ECU_B_ALARM_125us : 125 μ s 周期で Task を起こすための ALARM
 - ECU_B_ALARM_20ms_SCHM : 20ms 周期で Task を起こすための ALARM
- Ecuc_CO
 - ECU_CO_ALARM_100ms : 100ms 周期で Task を起こすための ALARM
 - ECU_CO_ALARM_10ms : 10ms 周期で Task を起こすための ALARM
 - ECU_CO_ALARM_20ms_SCHM : 20ms 周期で Task を起こすための ALARM

1ECU 版と同様に Buzzercontrol 処理は C2ISR で動作することにしたため、Ecuc_B に設定されている 125 μ s で起動していた ECU_B_TASK_125us を起動するための ECU_B_ALARM_125us は削除する。

Ecuc_CO に 10ms と 20ms で動作する Task を新たに設定したため、その Task を起動周期する ALARM を追加する。

ATK2-SC3-TP において以下のアラームを設定する。(※赤文字が修正・追加した箇所)

- Ecuc_B
 - ECU_B_ALARM_100ms : 100ms 周期で Task を起こすための ALARM
 - ECU_B_ALARM_20ms_SCHM : 20ms 周期で Task を起こすための ALARM
- Ecuc_CO
 - ECU_CO_ALARM_10ms : 20ms 周期で Task を起こすための ALARM
 - ECU_CDD_ALARM_10ms : 10ms 周期で Task を起こすための ALARM
 - ECU_CO_ALARM_20ms : 20ms 周期で Task を起こすための ALARM
 - ECU_CO_ALARM_100ms : 100ms 周期で Task を起こすための ALARM
 - ECU_CO_ALARM_20ms_SCHM : 20ms 周期で Task を起こすための ALARM

ATK2-SC1 から ATK2-SC3-TP に修正した際の ALARM と Task の関係を図 3-14 に示す。

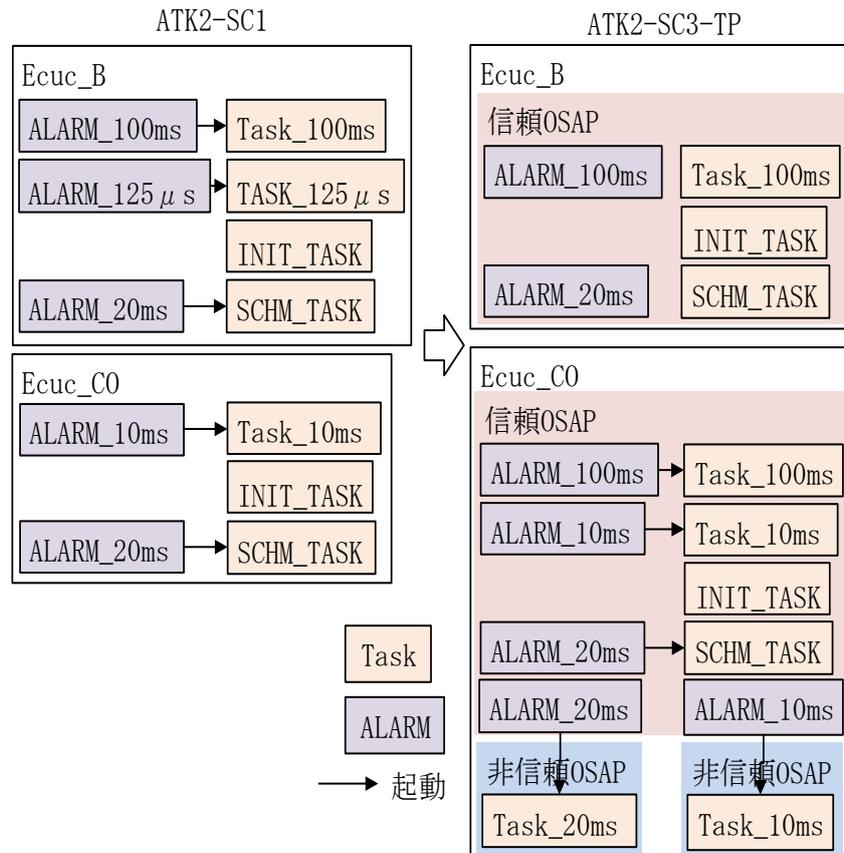


図 3-14 Task と ALARM の関係

3.2.3.4 SW-C の実体

RTE の設定である RteSwComponentInstance ではランナブルごとの Event に Task のマッピングと使用する ALARM を設定する必要がある。

ATK2-SC1 において、以下のように設定されている。

- Ecuc_B
 - SWCI_CddBodyControl
 - IB_CBC_TimingEvent_WinkerControl_100ms
 - RteEventRef : WinkerControl_timingEvent
 - RteMappedToTaskRef : ECU_B_TASK_100ms
 - RteUsedOsAlarmRef : ECU_B_ALARM_100ms
 - IB_CBC_TimingEvent_BuzzerControl_125us
 - RteEventRef : BuzzerControl_timingEvent

- RteMappedToTaskRef : ECU_B_TASK_125us
 - RteUsedOsAlarmRef : ECU_B_ALARM_125us
 - IB_CBC_TimingEvent_LampControl_100ms
 - RteEventRef : LampControl_timingEvent
 - RteMappedToTaskRef : ECU_B_TASK_100ms
 - RteUsedOsAlarmRef : ECU_B_ALARM_100ms
 - SWCI_CddLedBlinker
 - IB_CBL_TimingEvent_LedBlinker_500ms
 - RteEventRef : LedBlinker_timingEvent
 - RteMappedToTaskRef : ECU_B_TASK_100ms
 - RteUsedOsAlarmRef : ECU_B_ALARM_100ms
- Ecuc_CO
 - SWCI_OperationManager
 - IB_CM_TimingEvent_OperationControl_20ms
 - RteEventRef : OperationControl_timingEvent
 - RteMappedToTaskRef : ECU_CO_TASK_10ms
 - RteUsedOsAlarmRef : ECU_CO_ALARM_10ms
 - SWCI_CddController
 - IB_CC_TimingEvent_MessageParser_10ms
 - RteEventRef : MessageParser_timingEvent
 - RteMappedToTaskRef : ECU_CO_TASK_10ms
 - RteUsedOsAlarmRef : ECU_CO_ALARM_10ms
 - SWCI_CddRcCar
 - IB_CR_TimingEvent_PowerTrain_10ms
 - RteEventRef : PowerTrain_timingEvent
 - RteMappedToTaskRef : ECU_CO_TASK_10ms
 - RteUsedOsAlarmRef : ECU_CO_ALARM_10ms
 - SWCI_DriveManager
 - IB_DM_TimingEvent_DrivingControl_20ms
 - RteEventRef : DrivingControl_timingEvent
 - RteMappedToTaskRef : ECU_CO_TASK_10ms
 - RteUsedOsAlarmRef : ECU_CO_ALARM_10ms
 - IB_DM_TimingEvent_BodyControl_10ms
 - RteEventRef : BodyControl_timingEvent
 - RteMappedToTaskRef : ECU_CO_TASK_10ms
 - RteUsedOsAlarmRef : ECU_CO_ALARM_10ms
 - SWCI_CddLedBlinker

- IB_CBL_TimingEvent_LedBlinker_500ms
 - RteEventRef : LedBlinker_timingEvent
 - RteMappedToTaskRef : ECU_CO_TASK_100ms
 - RteUsedOsAlarmRef : ECU_CO_ALARM_100ms

1ECU 版と同様にランナブル Buzzercontrol は削除したため、Ecuc_B の Buzzercontrol 用のマッピング設定を削除する。

Ecuc_CO の DrivingControl と BodyControl は別 OSAP の新たに設定した Task で動作するため RteEvent を修正する必要がある。また、使用するアラームも修正が必要である。

ATK2-SC3-TP において、以下のように設定を行う。(※赤文字が修正した箇所)

- Ecuc_B
 - SWCI_CddBodyControl
 - IB_CBC_TimingEvent_WinkerControl_100ms
 - RteEventRef : WinkerControl_timingEvent
 - RteMappedToTaskRef : ECU_B_TASK_100ms
 - RteUsedOsAlarmRef : ECU_B_ALARM_100ms
 - ~~• IB_CBC_TimingEvent_BuzzerControl_125us~~
 - ~~• RteEventRef : BuzzerControl_timingEvent~~
 - ~~• RteMappedToTaskRef : ECU_B_TASK_125us~~
 - ~~• RteUsedOsAlarmRef : ECU_B_ALARM_125us~~
 - IB_CBC_TimingEvent_LampControl_100ms
 - RteEventRef : LampControl_timingEvent
 - RteMappedToTaskRef : ECU_B_TASK_100ms
 - RteUsedOsAlarmRef : ECU_B_ALARM_100ms
 - SWCI_CddLedBlinker
 - IB_CBL_TimingEvent_LedBlinker_500ms
 - RteEventRef : LedBlinker_timingEvent
 - RteMappedToTaskRef : ECU_B_TASK_100ms
 - RteUsedOsAlarmRef : ECU_B_ALARM_100ms
- Ecuc_CO
 - SWCI_OperationManager
 - IB_CM_TimingEvent_OperationControl_20ms
 - RteEventRef : OperationControl_timingEvent
 - RteMappedToTaskRef : **ECU_CO_TASK_20ms**
 - RteUsedOsAlarmRef : **ECU_CO_ALARM_20ms**
 - SWCI_CddController

- IB_CC_TimingEvent_MessageParser_10ms
 - RteEventRef : MessageParser_timingEvent
 - RteMappedToTaskRef : **ECU_CDD_TASK_10ms**
 - RteUsedOsAlarmRef : ECU_CO_ALARM_10ms
- SWCI_CddRcCar
- IB_CR_TimingEvent_PowerTrain_10ms
 - RteEventRef : PowerTrain_timingEvent
 - RteMappedToTaskRef : **ECU_CDD_TASK_10ms**
 - RteUsedOsAlarmRef : ECU_CO_ALARM_10ms
- SWCI_DriveManager
 - IB_DM_TimingEvent_DrivingControl_20ms
 - RteEventRef : DrivingControl_timingEvent
 - RteMappedToTaskRef : ECU_CO_TASK_10ms
 - RteUsedOsAlarmRef : ECU_CO_ALARM_10ms
 - IB_DM_TimingEvent_BodyControl_10ms
 - RteEventRef : BodyControl_timingEvent
 - RteMappedToTaskRef : ECU_CO_TASK_10ms
 - RteUsedOsAlarmRef : ECU_CO_ALARM_10ms
- SWCI_CddLedBlinker
 - IB_CBL_TimingEvent_LedBlinker_500ms
 - RteEventRef : LedBlinker_timingEvent
 - RteMappedToTaskRef : ECU_CO_TASK_100ms
 - RteUsedOsAlarmRef : ECU_CO_ALARM_100ms

3.2.3.5 RteOsInteraction

RteOsInteraction では、RTE と OS 間の連携の設定を行う項目である。この項目では期待する起動オフセット、起動周期、ランナブルの起動のために使用する OS アラームを設定する。

ATK2-SC1 において、以下の設定がされている。

- Ecuc_B
 - RteUsedOsActivation_ECU_B_ALARM_125us
 - RteExpectedTickDuration : 0.000125
 - RteActivationOsAlarmRef : ECU_B_ALARM_125us
 - RteUsedOsActivation_ECU_B_ALARM_20ms_SCHM
 - RteExpectedTickDuration : 0.02
 - RteActivationOsAlarmRef : ECU_B_ALARM_20ms_SCHM
- RteUsedOsActivation_ECU_B_ALARM_100ms

- RteExpectedTickDuration : 0.1
- RteActivationOsAlarmRef : ECU_B_ALARM_100ms
- Ecuc_CO
 - RteUsedOsActivation1
 - RteExpectedTickDuration : 0.01
 - RteActivationOsAlarmRef : ECU_CO_ALARM_10ms
 - RteUsedOsActivation2
 - RteExpectedTickDuration : 0.02
 - RteActivationOsAlarmRef : ECU_CO_ALARM_20ms_SCHM
 - RteUsedOsActivation3
 - RteExpectedTickDuration : 0.1
 - RteActivationOsAlarmRef : ECU_CO_ALARM_100ms

RteUsedOsActivation_ECU_B_ALARM_125us は起動周期 125 μ s の Alarm の設定のため、削除する。

新たに設定した ECU_CO_ALARM_20ms の設定を追加する。

ATK2-SC3-TP において、以下の設定を行う。

- Ecuc_B
 - ~~• RteUsedOsActivation_ECU_B_ALARM_125us~~
 - ~~• RteExpectedTickDuration : 0.000125~~
 - ~~• RteActivationOsAlarmRef : ECU_CBO_ALARM_10ms~~
 - RteUsedOsActivation_ECU_B_ALARM_20ms_SCHM
 - RteExpectedTickDuration : 0.02
 - RteActivationOsAlarmRef : ECU_B_ALARM_20ms_SCHM
 - RteUsedOsActivation_ECU_B_ALARM_100ms
 - RteExpectedTickDuration : 0.1
 - RteActivationOsAlarmRef : ECU_B_ALARM_100ms
- Ecuc_CO
 - RteUsedOsActivation1
 - RteExpectedTickDuration : 0.01
 - RteActivationOsAlarmRef : ECU_CO_ALARM_10ms
 - RteUsedOsActivation2
 - RteExpectedTickDuration : 0.02
 - RteActivationOsAlarmRef : ECU_CO_ALARM_20ms_SCHM
 - RteUsedOsActivation3
 - RteExpectedTickDuration : 0.1

- RteActivationOsAlarmRef : ECU_CO_ALARM_100ms
- **RteUsedOsActivation4**
 - **RteExpectedTickDuration : 0.02**
 - **RteActivationOsAlarmRef : ECU_CO_ALARM_20ms**

3.2.3.6 パーティション

EcucPartitionCollection では、OSAP によるパーティション構成の設定を行う。設定項目にはパーティション内で動作する SW-C を設定する必要がある。

ATK2-SC1 において、OSAP の概念がないためパーティションの設定はない。

ATK2-SC3-TP において以下の SW-C の設定を行う。

- Ecuc_B
 - EcucPartition_T
 - CddBodyControl
 - CddLedBlinker
- Ecuc_CO
 - EcucPartition_NT_OM
 - OperationManager
 - EcucPartition_NT_DM
 - DriveManager
 - EcucPartition_T
 - CddController
 - CddRcCar
 - CddLedBlinker

3.2.3.7 OSAP

ATK2-SC1 には OSAP の概念がないため設定がない。

OSAP の設定には、信頼 or 非信頼か、Alarm, Task, Isr, Partition, TimeWindow, Partition の Ref を行う必要がある。また、1ECU 版とは違い COM による通信をするため、非信頼が通信をするために信頼 OSAP で通信用の信頼関数を設定する必要がある。

ATK2-SC3-TP において以下の設定を行う。

- Ecuc_B
 - ECU_B_TOSAP
 - OsTrusted : 信頼
 - OsAppAlarmRef
 - ECU_B_ALARM_100ms

- ECU_B_ALARM_20ms_SCHM
 - OsAppTaskRef
 - ECU_B_TASK_100ms
 - ECU_INIT_TASK
 - ECU_B_SCHM_TASK
 - OsAppIsrRef
 - C2ISR_FOR_CAN_RX0
 - C2ISR_FOR_BUZZER
 - OsAppEcucPartitionRef
 - EcucPartition_T
 - Rte-CsCallTf-OperationManager
 - OsTrustedFunctionName : TRUSTED_Rte-CsCallTf-OperationManager
 - Rte-CsCallTf-DriveManager
 - OsTrustedFunctionName : TRUSTED_Rte-CsCallTf-DriveManager
 - Rte-ComSendSignalTf-EcucPartition_T
 - OsTrustedFunctionName : TRUSTED_Rte-ComSendSignal-EcucPartition_T
- Ecuc_CO
 - ECU_CO_TOSAP
 - OsTrusted : 信頼 OSAP
 - OsAppAlarmRef
 - ECU_CO_ALARM_10ms
 - ECU_CDD_ALARM_10ms
 - ECU_CO_ALARM_20ms
 - ECU_CO_ALARM_100ms
 - ECU_CO_ALARM_20ms_SCHM
 - OsAppTaskRef
 - ECU_CDD_TASK_10ms
 - ECU_CO_TASK_100ms
 - ECU_INIT_TASK
 - ECU_CO_SCHM_TASK
 - OsAppIsrRef
 - RLIN3x_RX_ISR
 - C2ISR_FOR_CAN_RX0
 - C2ISR_FOR_CAN_TX0
 - OsAppEcucPartitionRef : EcucPartition_T
- ECU_CO_NTOSAP_OM
 - OsTrusted : 非信頼 OSAP

- OsAppTaskRef : ECU_CO_TASK_20ms
- OsAppEcucPartitionRef : EcucPartition_NT_OM
- ECU_CO_NTOSAP_DM
 - OsTrusted : 非信頼 OSAP
 - OsAppTaskRef : ECU_CO_TASK_10ms
 - OsAppEcucPartitionRef : EcucPartition_NT_DM

3.2.3.8 システムサイクルとタイムウィンドウ

ATK2-SC1 において、システムサイクルの概念がないため設定はない。

タイムウィンドウの設定にはそのタイムウィンドウの長さや開始時間を設定する必要がある。また、必ずアイドル時間を設ける必要があるため、最後のタイムウィンドウの終了時間がシステムサイクル時間より短くする必要がある。

ATK2-SC3-TP において、以下のように設定する。

- Ecuc_CO
 - OsSystemCycle
 - システムサイクル : 0.02
 - TwdSchM1
 - TW の長さ : 0.005
 - TW の開始時間 : 0.0
 - TwdAppDM1
 - TW の長さ : 0.0025
 - TW の開始時間 : 0.005
 - TwdSchM2
 - TW の長さ : 0.005
 - TW の開始時間 : 0.0075
 - TwdAppDM2
 - TW の長さ : 0.0025
 - TW の開始時間 : 0.0125
 - TwdAppOM1
 - TW の長さ : 0.0025
 - TW の開始時間 : 0.015
- Ecuc_B
 - OsSystemCycle
 - システムサイクル : 0.02
 - TwdSchM1
 - TW の長さ : 0.015

-
- TW の開始時間 : 0.0

4. 変更履歴

Version	Date	Detail	Editor
1.0.0	2016/04/26	・ 新規作成	NCES
		・	