



「AUTOSAR-Wiki」サービスの 開始について



2014年11月19日

NPO法人 TOPPERSプロジェクト

Email: autosar@toppers.jp

背景～AUTOSAR利用時の課題～

AUTOSARが仕様を公開しているが...

- ▶ 曖昧な仕様、未定義な仕様が多数
 - ▶ 原文だけでは、理解できないことが多々存在する
 - ▶ 他社の解釈や意見を聞きたい
 - ▶ 1社で結論を出すリスクが大きい
- ▶ 各ベンダはそれぞれ独自に改変/拡張している
- ▶ TOPPERSでも独自に改変/拡張している

AUTOSARという1つの仕様に対して、少なくとも国内企業間で同じ解釈・考え方を持つべきではないか？

- ▶ でなければ、共通プラットフォーム仕様にならない
- ▶ AUTOSARが掲げる
「Cooperate on standards, compete on implementation」
を少なくとも国内では実現する
- ▶ そのために情報交換、ディスカッションする場が必要
 - ▶ 手段はWebが現実的
 - ▶ **AUTOSARに関するWikipediaのようなサイトが理想的**

「AUTOSAR-Wiki」の概要

- ▶ 前述のAUTOSARが抱える問題点を解決するため、AUTOSAR仕様の情報交換を行う場を提供する
- ▶ AUTOSARを構成するRTE、BSWモジュールの仕様における、API仕様、コンフィギュレーション仕様に関する情報を中心に取り扱う
- ▶ tracを活用し、Wikiによる仕様解説、チケットによる質問、意見交換を行う
 - ▶ TOPPERSプロジェクト会員でない方は、閲覧のみ可能とする
 - ▶ TOPPERSプロジェクトが適任と認めた場合、TOPPERS非会員であってもアカウントを付与することがある
- ▶ 本サイトに掲載された情報は、オープンな扱いとし、任意の場所から誰でも閲覧可能とする

http://dev.toppers.jp/trac_user/ap/

「AUTOSAR-Wiki」運用イメージ

TOPPERS会員企業



- Wiki更新
- チケット(質問)発行

TOPPERS 非会員企業



閲覧のみ可能

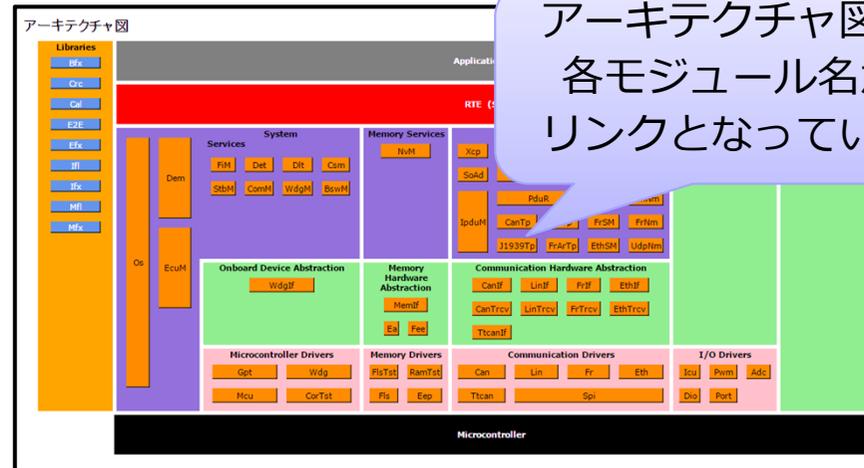


TOPPERS/Webサーバ



AUTOSARのパラメータ名
等でヒットする

Webサイトイメージ



モジュール毎にAPI、コンフィギュレーション
パラメータに関する情報を掲載

モジュール一覧 (80modules)

General

Module	ID [Dec]	ID [Hex]
Standard Types(Std)	197	0xC5

(Imodule)

Communication Stack

Module	ID [Dec]	ID [Hex]
CAN Driver(Can)	80	0x50
CAN Interface(CanI)	60	0x3C
CAN Network Management(CanNm)	31	0x1F
CAN State Manager(CanSM)	140	0x8C
CAN Transceiver Driver(CanTrcv)	70	0x46
CAN Transport Layer(CanTp)	35	0x23
COM(Com)	50	0x32
COM Manager(ComM)	12	0x0C
Communication Stack Types(ComType)	196	0xC4
Ethernet Driver(Eth)	88	0x58
Ethernet Interface(EthI)	65	0x41
Ethernet State Manager(EthSM)	143	0x8F
Ethernet Transceiver Driver(EthTrcv)	73	0x49
FlexRay AUTOSAR Transport Layer(FrArTp)	38	0x26
FlexRay Driver(Fr)	81	0x51
FlexRay ISO Transport Layer(FrTp)	36	0x24
FlexRay Interface(FrI)	61	0x3D
FlexRay Network Management(FrNm)	32	0x20
FlexRay State Manager(FrSM)	142	0x8E
FlexRay Transceiver Driver(FrTrcv)	71	0x47
IPDU Multiplexer(IpduM)	52	0x34
LIN Driver(Lin)	82	0x52
LIN Interface(LinI)	62	0x3E
LIN Network Management(LinNm)	63	0x3F
LIN State Manager(LinSM)	141	0x8D
LIN Transceiver Driver(LinTrcv)	64	0x40

モジュール一覧の
各モジュール名も
リンクとなっている

OS

概要

- AUTOSARプラットフォームにおけるリアルタイムOS
 - 車に不可欠なソフトウェア、動かし難いモノを持つ
 - MCAL層が難いため、マイコンに依存した実装が伴う
- OSEK/VDX仕様との上位互換とされ、OSEK仕様書に記載されている内容については、AUTOSAR仕様書には記載されていない
- OSEK仕様書もAUTOSAR同様、WinCCで公開されている
- OSEK仕様とAUTOSAR仕様で若干の矛盾等も存在する
- コンフィギュレーション方法は、AUTOSAR標準のXML方式に統合される
- OSの機能セット別に、4つのスケールクラスが規定されている
 - スケールクラス1(SCL1)
 - 基本セット
 - スケールクラス2(SC2)
 - 基本セット + タイミング保護
 - スケールクラス3(SC3)
 - 基本セット + メモリ保護
 - スケールクラス4(SC4)
 - 基本セット + タイミング保護 + メモリ保護
 - すべてのSCをマルチコアに拡張可能である
- 用語一覧
 - C1ISR: カガコリISR
 - C2ISR: カガコリ2ISR
 - OSAPI: OSアプリケーション
- API一覧
 - ～は OSEKのため非規定

OSEK仕様API	ID [Dec]	ID [Hex]	AUTOSAR仕様API	ID [Dec]	ID [Hex]
ActivateTask	GetApplicationID	0	0x00

OSの例

API仕様(OSEK)

ActivateTask

StatusType ActivateTask(TaskType TaskID)

- TaskIDで指定したタスクを起動する
- 多重に起動要求を行った場合、要求した順序で起動される
 - 拡張タスクが、多重起動要求できないのは、待ち状態に入ってしまうと次の起動要求を要求した順序で処理できないからと考えられる
 - TOPPERS/ATK2では、必要性が低いことから、要求した順序を保持しない実装としている
 - タスクの実行順序の優先順位は、本来タスク優先度で指定するものであり、同一優先度タスクが実行される順序が処理に依存するべきではない、という考え方である
- 到着時間監視において、ActivateTaskを発行した側と、発行された側のどちらのタイミング保護違反となるかは規定されていない
 - マルチコアでは、どちらのコアでプロテクションフックを起動するか等の問題がある

TerminateTask

StatusType TerminateTask(void)

- 呼び出したタスクを終了する
- 多重起動要求されているタスクが、TerminateTaskを実行した場合、再度実行可能状態となるが、TerminateTaskを呼ばずに終了(returnを呼ぶ等)した場合に、再度実行可能状態となるかは規定されていない
 - TOPPERS/ATK2では、以下のような実装としている
 - タスクが不正終了(returnを呼ぶ)した場合:多重起動は有効
 - タスクが強制終了(保護違反発生した場合:多重起動は有効
 - 所属するOSAPが強制終了した場合:多重起動はリセット

ChainTask

StatusType ChainTask(TaskType TaskID)

- 呼び出したタスクの終了と、TaskIDで指定したタスクの起動を行う(TerminateTask + ActivateTask)

Schedule

StatusType Schedule(void)

- 呼び出した時点での、すべての実行可能状態のタスクの優先度に従って、再スケジューリングを行う
 - つまり、呼び出し元タスクが、ノンプリエンティブタスクでなければ、呼び出す意味がない

GetTaskID

StatusType GetTaskID(TaskRefType TaskID)

- 実行状態のタスクIDを取得する
 - 複数タスクから呼び出される共用関数や、C2ISR、各フックルーチンから使用される
- 本APIは、エラーフックでのデバッグにおける使用が考えられるが、割り込み禁止時のAPI呼び出し違反[OS093]により起動したエラーフックでは、割り込み禁止状態のため、GetTaskIDが使用できないという不整合がある
 - TOPPERS/ATK2では、実害がないため、割り込み禁止状態であっても、GetTaskIDを呼び出せるように実装している

GetTaskState

StatusType GetTaskState(TaskType TaskID, TaskStateRefType State)

- TaskIDで指定したタスクの状態を取得する
- 本APIは、エラーフックでのデバッグにおける使用が考えられるが、割り込み禁止時のAPI呼び出し違反[OS093]により起動したエラーフックでは、割り込み禁止状態のため、GetTaskStateが使用できないという不整合がある
 - TOPPERS/ATK2では、実害がないため、割り込み禁止状態であっても、GetTaskStateを呼び出せるように実装している

AUTOSAR仕様の不備および、
ATK2での対応方法を説明