LSI IP デザイン・アワード応募書類表紙（企業）

タイトル： **オープンソース FlexRay 通信：TimeTriggered OS（TT-OS）と FlexRay 通信ミドルウェア**

技術分野： 車両用次世代通信，タイムトリガ型リアルタイム OS

応募者： 服部博行，大西秀一，森川聡久，片岡歩，中村和彦，中村俊夫，高田広章

所属機関： 株式会社ヴィッツ,株式会社サニー技研,名古屋大学情報科学研究科

1．研究・開発の目的・狙い

近年の自動車制御は、多くの ECU（Electronic Control Unit ）を用いて、相互に通信等の技術を利用して情報を交換しながらより細密な制御を実現しています。一方で、ECU 間の情報交換を担う通信技術は多様化し、自動車メーカが簡単に接続できず、接続検証に多大な工数を費やしています。また、次世代の車両制御では、現世代の通信プロトコルを用いた通信では、データ転送量が少なく対応できないとの試算がされています。弊社は、この事実を解決するために、次世代通信として最も有力とされている FlexRay 通信用のミドルウェアを開発し、オープンソースとして公開することにより、データ転送量や接続性を高め技術的な課題を解決するとともに、自動車制御業界へ貢献することを目的としています。

2．研究・開発の概要

1) **利用分野：**統合自動車制御用電装部品
2) **特徴：**次世代通信の実現とタイムトリガ型リアルタイム OS の実現
3) **種類：**ソフト VC
4) **規模：**タイムトリガ型リアルタイム OS　7,655 byte（ROM コードサイズ）
   FlexRay 通信ミドルウェア　　25,217 byte（ROM コードサイズ）
   ※ ルネサステクノロジ製 M32C および同社製コンパイラを使用。
5) **性能：**タイムトリガ型リアルタイム OS の RAM メモリ必要量 116 byte , ROM メモリ必要量 854 byte
   FlexRay 通信ミドルウェアの RAM メモリ必要量 1,198 byte ROM メモリ必要量 4,006 byte
   ※送信メッセージ中のアプリデータは全て状態データを使用。
   メッセージ数：8、1 メッセージ中のアプリデータ数：8

3．訴求点および効果

欧州を中心として仕様策定されている FlexRay 通信ミドルウェアを世界に先駆けて開発し、オープンソースソフトウェアとして公開します。また、FlexRay 通信ミドルウェアを稼動させるタイムトリガ型リアルタイム OS を仕様策定から実施しました。この OS は、欧州で策定された仕様と比較し、タイムトリガ型制御とイベントトリガ型制御を両立する性能を保持し､他に類を見ない新しい仕様の OS であり、この OS も FlexRay 通信ミドルウェア同様、オープンソースとして公開します。

これらの開発成果は、単なる研究やオープンソースに留めるのではなく、実製品への適用を視野に入れています。そのための活動として、欧州で開催された FlexRay Product Days に出展し、本ソフトウェア部品の有用性をアピールし、普及促進にも努めています。

アピール指標：新規性、実用性、品質

オープンソース FlexRay 通信：TimeTriggered OS と FlexRay 通信ミドルウェア
(第 8 回 IP アワード応募書類 特徴の説明書)

服部博行† 大西秀一† 森川聡久† 片岡歩 中村和彦‡ 中村俊夫‡ 高田広章◆

†株式会社ヴィッツ ‡株式会社サニー技研 ◆名古屋大学情報科学研究科

E-mail: †{hat,ohnishi,morikawa,kata}@witz-inc.co.jp, ‡{nakamura.kazuhiko,nakamura.toshio}@sunnygiken.co.jp ◆hiro@ertl.jp

1. FlexRay 通信とシステム LSI

FlexRay 通信は FlexRay コンソーシアム[1]により仕様策定された, 先進自動車アプリケーション向けの通信規格である. 欧州の自動車メーカや自動車部品メーカを中心に自動車の電子アーキテクチャのオープンな標準を作成するために設立された AUTOSAR[2]においても, FlexRay 通信を用いた標準化が検討されており, 次世代の自動車通信分野において欧州ではデファクトスタンダードになりつつある. さらに日本国内においても, 日本の自動車メーカが中心となり, 自動車制御システム向けのネットワーク技術, ミドルウェア, ソフトウェア基盤等を協調して開発するために設立された JasPar[3]で製品への適応方法等について検討を開始し始めている. このように, FlexRay 通信は, 次世代の自動車制御装置において必要不可欠な通信であり, その通信ミドルウェアは自動車開発に不可欠な IP (ソフトウェア設計資産) の一つであるといえる.

FlexRay 通信を効率よく利用するためには, 従来の制御で広く利用されていたイベント駆動型からタイムトリガ型の制御に変更する必要がある. そのため, 利用するオペレーティングシステムはタイムトリガ制御に対応する必要がある.

欧州の自動車メーカが中心となり自動車制御用基盤ソフトウェアの標準仕様を策定する OSEK/VDX[4]では, タイムトリガに対応したオペレーティングシステム仕様[1]を策定し, 欧州を中心とした多くのオペレーティングシステムベンダがこの仕様に準拠したオペレーティングシステムを開発しているものの, 自動車制御装置への適応には問題点も含まれている.

2. FlexRay 通信対応の概要

FlexRay 通信への対応には, OSEK/VDX OS 仕様[2]に準拠した TOPPERS/OSEK カーネル[5]上で稼動する TTM (Time Triggered Module), FlexRay デバイスを制御する FlexRay デバイスドライバ, FlexRay 通信管理を行う FlexRay-NM(Network Management)およびタイムトリガ処理に対応しアプリケーションとのインターフェースを担う TT-COM (Time Triggered Communication) などのソフトウェア部品を利用し実現する.

ターゲット環境として, 現時点では, ルネサステクノロジ社の M32C プロセッサおよび同社製の FlexRay デバイスをサポートしているが, 他に NEC 製, フリースケール製などの各種プロセッサへのポーティングが進行中である.

FlexRay 通信対応の目的は, 次世代車両通信として利用が見込める通信であるにも関わらず, その詳細仕様などの決定が遅く, どのような想定で利用するべきか不明な点が多いため, 現在までに決定している仕様を元に, 車両への適応検証などの研究用途として評価することである. ただし, 単なる研究用途だけでなく, 現世代通信の置き換えとしての利用を模索するために, 現状の自動車メーカが利用している通信システムへの要求/機能を調査し, さらに, 自動車メーカが FlexRay 通信に期待する要求事項などを盛り込み, 製品適応に対応できる機能, 性能, 品質を実現している. また, FlexRay コンソーシアムに参加している半導体メーカから FlexRay 通信の特徴や FlexRay デバイス情報の提供を受け, 現在のデバイス機能を“素直”に利用して実現している. これは, 今後 FlexRay コンソーシアムで仕様検討が進み, 仕様変更が行われた場合でも柔軟に対応できるように留意したためである.

これらのソフトウェア部品は, TOPPERS プロジェクト[6]から 2006 年 1 月中旬より TOPPERS プロジェクト会員向けに早期リリースし, 最終的にオープンソースソフトウェアとして公開する予定である. 研究用途や組込み利用を想定して, これらのソフトウェア部品は, フリーソフトウェアとしても特に柔軟な TOPPERS ライセンス[7]による配布とし, 製品への利用, 販売等に制限を与えない.

3. FlexRay 通信対応の機能および特徴

本 FlexRay 通信対応ソフトウェアは大きく 3 種類に大別することが出来る. 一つは TOPPERS/OSEK カーネルに TTM と呼ぶタイムトリガ処理を実現するミドルウェアを組み合わせた TT-OS (Time Triggered Operating System), 二つ目は FlexRay デバイスドライバ, FlexRay-NM, TT-COM から構成される FlexRay 通信ミドルウェア, 三つ目は, これらのミドルウェアおよびアプリケーションをオペレーティングシステムと接続するコンフィグレーションツールである SG (System Generator) である. このうち, 三つ目の SG は静的なコンフィグレーションを行うものであり, ターゲット上では稼動しない開発ツールに分類されるため, 本説明から除外する.

3.1 TT-OS の機能および特徴

FlexRay 通信を実現する上で重要となるタイムトリガ処理は, OSEK 仕様 OS にタイムトリガモジュールを追加して実現し, 本構成では TT-OS と称する. すなわち, TT-OS = OSEK OS + TTM とあらわすことができる.

TT-OS は, OSEK/VDX が, タイムトリガ通信を実現するために仕様策定した OSEK/VDX Time-Triggered Operating System (以下 OSEKtime)[1] を参考にしているが, 最終的な仕様は大きく異なっている. すなわち, OSEKtime はタイムトリガ処理を実現することを目的とし, 非タイムトリガ処理はタイムトリガ処理を

---

[1] 欧州の自動車メーカ, 電装部品メーカが中心となり次世代車両通信仕様を策定する団体 http://www.flexray.com

[2] Automotive Open System Architecture http://www.autosar.org/

[3] Japan Automotive Software Platform Architecture http://www.jaspar.jp

[4] Open System and the corresponding interfaces for automotive electronics / Vehicle Distributed executive http://www.osek-vdx.org/

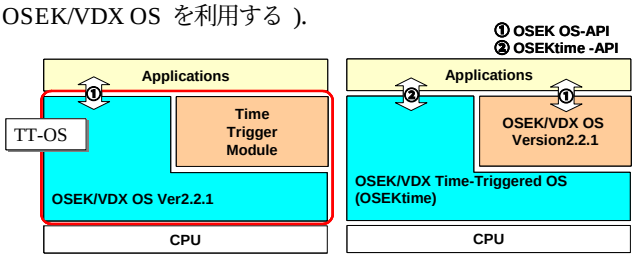
[5] 応募代表者らが 2004 年に開発した OSEK OS 仕様に準拠したリアルタイムカーネル. 現在オープンソースソフトウェアとして一般公開中

[6] http://www.toppers.jp/

[7] http://www.toppers.jp/license.html

実行していない空き時間を利用してサービスを提供する．一方，TT-OS の仕様は，タイムトリガ機能を利用した，FlexRay 通信機能を実現することを目的としているものの，FlexRay 通信と非タイムトリガで実行するべき制御処理を両立することに注力している．この仕様の違いは，自動車メーカが FlexRay 通信の利用を想定している製品を複数選定し，製品適応時に必要となる機能および性能を明確にした結果である．そのため，TT-OS の機能および特徴は，OSEKtime との比較を行うことにより明確となる．

TT-OS のタイムトリガ機能は，TTM を OSEK/VDX OS 上で稼動させることにより実現している．一方 OSEK/VDX が提唱する方法は，タイムトリガ処理はタイムトリガ機能を提供する OSEKtime で実現し，非タイムトリガ処理は，OSEKtime のアイドル処理時にサービスを提供する (通常，アイドル処理に OSEK/VDX OS を利用する).



TT-OS構成図　　参考：OSEKtime構成図

TT-OS を利用した場合のソフトウェア構成では，OS は OSEK/VDX OS のみで実現できるため，以下の優位点がある

- **小メモリリソース**

  TT-OS は OSEK/VDX OS に TTM を組み合わせて実現するため，タイムトリガ機能を実現するにあたり，TTM モジュールの追加のみで可能となる．

[単位:Byte

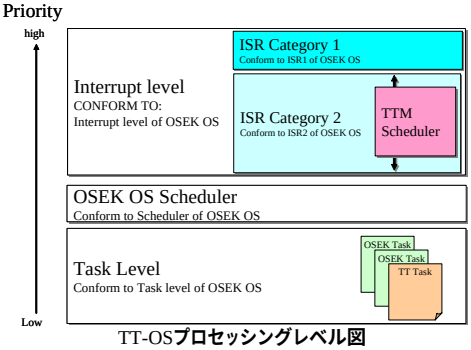
	ROM		RAM
	CODE	DATA	DATA
TOPPERS/OSEKカーネル	5,851	64	103
TTM	1,804	790	13

※TT-OSはTOPPERS/OSEK + TTM で算出可能
※計測はルネサス製コンパイラを利用し，対象CPUはM32C

- **既存アプリケーションの移植性**

  FlexRay 通信を利用する場合，今まではタイムトリガを提供するのは OSEKtime のみであり，これらの通信を利用する場合は，例え OSEK/VDX OS 上で稼動しているアプリケーションであっても OSEKtime への移植，もしくは，OSEKtime のアイドル時間に動作する制限のある環境に適応させる必要があった．本 TT-OS はイベントドリブン処理とタイムトリガ処理が共に OSEK/VDX OS に準拠した OS 上で稼動するため，サービスルーチンの変更や時間的制約を受けることが無く，従来同様の優先度ベースによるタスク設計をすることが可能となるため，新たに移植もしくは適応作業を必要としない．
  すなわち，OS が 1 種類のみの実装で実現しているため，アプリケーション設計者は OS を使い分ける必要がなく，従来同様の設計ポリシーが継承でき，新たに OS を習得する必要がない．

これらの優位点に加え，OS 構成上の特徴としてプロセッシングレベルが他のタイムトリガを提供する OS と比べて特徴がある．すなわち，TT-OS のプロセッシングレベルは，イベントドリブン OS である OSEK/VDX OS のプロセッシングレベルを踏襲し，タイムトリガ処理要求部 (以下，TTM スケジューラ) を OSEK/VDX OS の割込みレベルで実現している．この割込みレベルで実行する TTM スケジューラは，イベントドリブン処理に影響を極力与えないよう割込みレベルも任意設定とし，TTM スケジューラ内ではタイムトリガ処理本体を実行しない (OSEKtime は割込み処理内でタイムトリガ処理本体を実行する) ．すなわち，TTM スケジューラは OSEK/VDX OS にタイムトリガ処理に必要な処理要求を発行するのみである．これにより，アプリケーション設計者は，タイムトリガタスクおよび非タイムトリガタスクの動作順序が優先度ベースとなり，重要な処理は優先度を高くすることによりシステムとして動作を保障することができる．また，優先順位を設計者自身が決定できるため，低優先度タスクの遅延時間予測も容易になる．



TT-OSプロセッシングレベル図

また，TT-OS は，OSEKtime では対応が難しいとされるタイムトリガ処理と非タイムトリガ処理の両立が可能である．すなわち，OSEKtime 環境下では，OSEK/VDX OS の非タイムトリガ処理はタイムトリガ処理の要求がないアイドル時間でのみサービスが提供される．そのため，これら非タイムトリガ処理は，たとえ割り込み処理要求であったとしても，タイムトリガで実現される FlexRay 通信により阻害されることとなる．この問題点を，エンジン制御を例にして説明する．通常エンジン制御はクランク軸の回転に同期して処理を実行する必要があり，タイミングを逃すと失火や排気部位での燃焼などとなり，安全性や環境上問題となる．通常これらのイベント処理は OSEK/VDX OS で実現する．このエンジン制御で用いる情報交換に OSEKtime 環境での FlexRay 通信を利用した場合，FlexRay 通信は OSEKtime でサービスされる．すなわち，OSEK OS 上で稼動するエンジン制御は，OSEKtime 上で稼動する通信制御のアイドル時間に実行されることとなり，正しくエンジンを制御することは難しい．一方，TT-OS はシステム全体で処理の優先順位を設計者が指定することが可能となり，必要な処理はタイムトリガ処理より優先的に実行することができ，システム全体で安全に制御することができる．この特徴が TT-OS の最大の特徴であるといえる．

TT-OS を構成する TTM 機能は，OSEKtime 仕様および OSEK/VDX OS 仕様を参考にし，タスク起動要求の他に，コールバック関数機能，デッドラインモニタリング機能を実現した．

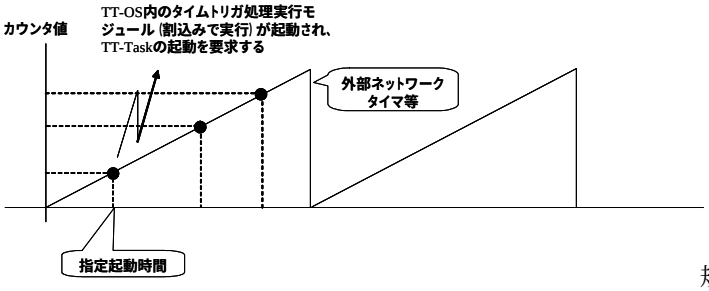
コールバック関数機能は，TTM スケジューラ内からコールバック関数の呼び出しを実行することができる．この機能は OSEKtime には見られないが，OSEK/VDX OS のサービスルーチン内からコールバック関数を呼び出す機能があるため，TTM の機能に加えた．これはタイムトリガタスクが起動要求されて起動されるまでの遅延が許容されない場合などに有効であるが，TTM スケジューラは割込み処理であるため，割込み処理内からのアプリケーション関数を呼び出すこととなり，短い処理に限定して利用しなければならない．

タイムトリガ機能の特徴として, ある種のサービスを外部時間に同期して実行する必要があり, タイムトリガ処理部のタスク起動処理により, 所望とするタスクを起動することができることは前述で説明した. しかし, 所望とするタイミングでのタスク起動が実現できたとしても, 何らかの影響により指定時間までに処理が完了していない, もしくは情報が作成されていない場合にシステムとして致命的な問題が発生しないよう, デッドラインモニタリング機能を実現した. TTM スケジューラで静的に決められた情報により指定タスクが完了しているかをモニタし, 未完了の場合は, 設計者が意図したミス処理を実行する.

タイムトリガ機能は, 外部時間に同期して処理をする事であり, 本 TTM では, 処理するべき時刻を外部機器から提供される時刻をそのまま利用した. その結果, TTM 内部で時刻補正処理を省くことを可能し, 同時に, 通信同期がずれた場合の同期方法も, 補正計算をすることなく同期させる手法も考案した.

- **トリガ時刻取得**

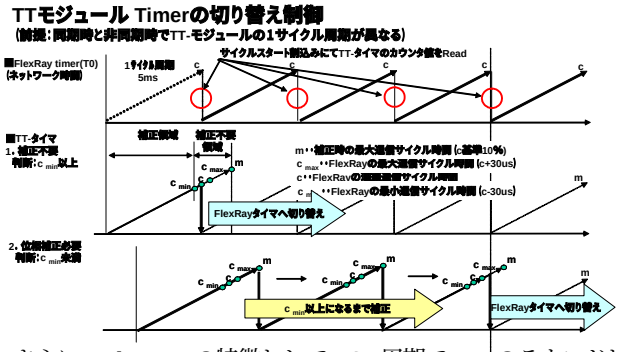
  TTM が利用する基本となる時間は, 外部からの割込みにより取得する. 本 TT-OS は, FlexRay 通信機能を実現するために開発したため, 基本時間を得る割込みは FlexRay 通信用デバイスのタイマを利用する. 多くのタイムトリガ機能を提供する OS は, 外部時間 (FlexRay の場合は, 外部ネットワーク時間をさし, グローバル時間を呼ぶ場合が多い) と内部時間 (プロセッサ内部の時間をさし, ローカル時間と呼ぶ場合が多い) を別々に管理し, 外部時間と内部時間の差を周期的に計算して, 周期開始時のオフセット処理やクロックの伸張などで外部時間と内部時間の補正を行う. このような場合, 周期的に処理負荷の多いオフセット計算や比例計算をしなくてはならない. そのため, 周期タイミングでの処理負荷や周期内での起動誤差 (周期のはじめと終わりなど) が大きくなる傾向が考えられる. これらの問題点を解決するために, 本 TTM では, FlexRay デバイスに指定サイクルの指定時間に割込みを発生させる方式にした. この方式を利用することにより, タイムトリガタスクの起動トリガとなる割込みは正確にプロセッサに伝達され, かつ, 周期毎の補正処理が不要となる. 反面, FlexRay デバイスの機能を利用することにより, FlexRay デバイス仕様と密接になる問題が発生する. この問題点への対応として, 国内半導体メーカの FlexRay デバイスを調査した結果, すべての半導体メーカが同一サプライヤからデバイスの供給を受けており, 国内での利用は問題にならないことと, 仮に海外半導体への対応でデバイス機能の利用ができない場合であっても, FlexRay デバイスドライバで対応することにより, TTM への影響を最小限に抑えている.



- **時間およびサイクルの同期方法**

  FlexRay 通信が正常に通信を行っている場合は, 前述で説明したとおり, 所望とするタイミングを外部時間タイマから取得するため, 正確な時間を取得できる. しかし, 何らかの問題 (スタートアップ時や障害等) が発生すると, 通信エラー状態となり, 外部時間が得られなくなる. この様な場合, 例え通信ができなくても, プロセッサ内部で処理をする必要がないとは判断できない. この様な場合を想定し, TTM 内部に内部時間作成用のタイマを常に稼動させておき, 障害時の対応を行う. 具体的には, プロセッサ内部のインテリジェントタイマ (タイマ値 0 からカウントアップし, 指定タイマ値になると, タイマ満了割込みを発生し, タイマ値を初期化して, 再スタートとなるタイマ) を用いて, 内部カウンタを動作させる. この時, タイマ満了値を外部時間周期より長く設定 (デフォルトでは外部周期の 110%) し, 正常時は FlexRay デバイスからのサイクル開始割込みにより, タイマ値を初期化する. この初期化が内部時間の補正となり, サイクル毎の複雑な演算を必要としていない. また, 仮に同期ずれなどの通信障害が発生した場合は, FlexRay デバイスから通知されるエラー割込みを捕らえる, もしくは FexRay デバイスから発行されるはずのサイクル開始割込みが通達される前 (もしくは通達なし) にインテリジェントタイマの満了割込みを確認した場合は, 即座に利用するタイマをプロセッサ内部タイマに切り替える. このように通信エラー状態であっても, 同期通信時ほど正確なタイミングでは無いが, 処理の継続を可能とする (通信が行われないため, 正確なタイミングは不要である).

  時間同期は, FlexRay デバイスが他のノードと同期が完了すると, 正常時同様にサイクルの開始割込みが発生する. この割込みと内部時間を比較し, 補正不要と判断した時点で内部タイマを初期化して時間の同期が完了する. ここで重要となる点は, 外部時間周期と内部時間周期が同じである場合, 両者の周期開始タイミングは縮まらない. この解決方法として, 前述した内部周期時間を外部周期より長めに設定し, 補正不要領域に入らない部位での外部周期開始は無処理とし, 外部周期と内部周期の時間差を利用して, 外部時間と内部時間を同期させる.



  さらに, FlexRay の特徴として, 64 周期で 1 つのラウンドとして扱うため, 周期の開始が内部と外部で一致したとしても実行しているサイクルが一致していないと問題がある. このサイクルの同期も前述した周期時間差を利用し, 徐々に時間およびサイクルを合わせる. この方式を提供することにより, サイクル抜けやタイムトリガ処理の抜けを防止することができる.

このような特徴を持ちタイムトリガ処理を提供する TTM は新規に追加したサービスルーチンは驚くほど少ない(OS としてのサービスは OSEK/VDX OS の API で提供). 以下に TT-OS (TTM) にて新たに提供したサービスルーチンを列記する.

機能	API名称	備考
TTMの動作開始指令	ttmInitTTM	
エラーフック	ttmErrorHook	エラー発生時に呼び出されるフックルーチン
スタートアップフック	ttmStartupHook	TTM動作開始時に呼び出されるフックルーチン
ネットワーク同期ずれフック	ttmNetworkNoSyncHook	ネットワーク同期から外れた場合に呼び出されるフックルーチ
ネットワーク同期フック	ttmNetworkSyncHook	ネットワーク同期時に呼び出されるフックルーチン
サイクルスタート通知	ttmIntNetworkCycleStart	
ネットワーク割込み通知	ttmIntNetworkTimer	指定サイクル、指定クロック時に呼び出される
ネットワーク同期ずれ通知	ttmIntNetworkCCF	ネットワーク同期ずれ検出時に呼び出される
ネットワーク状態取得	ttmGetSyncStatus	
サイクルカウント，クロック取得	ttmGetNetworkTime	
TTM実行中サイクルカウント取得	ttmGetActionCycleCount	
利用中のタイマ情報取得	ttmGetTime	

以上の説明から明確なように，TT-OS は軽量コンパクトにタイムトリガを実現し，従来の OSEK/VDX OS を最大限利用したリアルタイム OS であると言える．また，タイムトリガ処理とイベント処理のどちらも対応できるため，OSEKtime と比較して応用範囲が広く，利用しやすく，移植性が高い特徴があり，ソフトウェアの部品化を支援する基盤ソフトウェア部品である．

3.2 FlexRay 通信ミドルウェアの機能および特徴

FlexRay 通信を実現するためには，物理的なデバイス提供が必要であるばかりではなく，そのデバイスをソフトウェアで制御するデバイスドライバが必要となる．また，アプリケーションの負荷を軽減するために，FlexRay 通信を利用するアプリケーションが共通に必要となるデータ処理部位をミドルウェア機能に用意するべきである．さらに，FlexRay 通信は LIN 通信等と異なり，各ノードが FlexRay バスに接続されているノードを意識する必要があり，バス上ノードの接続状況や Wake Up/Sleep を管理する機能もミドルウェアに用意するべきである．FlexRay 通信ミドルウェアはこれらの機能をすべて満たした FlexRay 通信専用のミドルウェアであり，各機能を順に，FlexRay デバイスドライバ，TT-COM，FlexRay-NM として開発した．

- **FlexRay デバイスドライバ**
  FlexRay デバイスドライバは FlexRay デバイスを制御するために用意されたサービスルーチン群であり，デバイスに依存する部位である．（現在はルネサス製 FlexRay[8]デバイスを対象とし，他のデバイスに対応するためには，このデバイスドライバ部位を移植することとなる．）通常のデバイスドライバの場合，アプリケーションからはデバイスの利用開始時に公開されている初期化ルーチンをコールするのみであり，FlexRay 通信ミドルウェアのデバイスドライバも同様の思想となる．しかし，FlexRay 通信はネットワーク時間に同期して処理を行う必要があるため，他のデバイスドライバと異なり，ネットワーク時間との同期に必要な時間通知機能，サイクル開始割込み，同期ずれ通知割込みなどを外部公開し，同期処理を可能とするサービスを提供している．3.1 章で説明した TT-OS は，このデバイスドライバ提供機能を利用して同期処理を実現しているため，TT-OS には必須となるサービスであり，TT-OS と密接に結合している部位である．そのため，他のデバイスに移植する場合は，これらのサービスを TT-OS に提供する必要がある．
- **TT-COM**
  TT-COM は，3.1 章で説明した TT-OS に対応した Communication ミドルウェアであり，OSEK/VDX が規定した OSEK/VDX Communication Version 3.0.3 [3]（OSEK COM），OSEK/VDX Fault-Tolerant Communication Version 1.0 [4] (OSEK FT-COM）を参考に TT-OS 仕様を考慮して仕様策定した通信用データ処理ミドルウェアである．主な機能とし，フィルタ制御，Que 制御，メッセージの Pack/UnPack 制御，エンディアン変換，通信途絶チェックなどを持ち，アプリケーションから直接指令を受け，また通信状態を返却する I/F 部を受け持つ部位である．
  この I/F 部には複数のサービス関数を用意しており，アプリケーションからは以下の関数で送信・受信制御を行うことができる．

機能	API名称	備考
メッセージ送信	ttSendMessage	
メッセージ受信	ttReceiveMessage	
フレーム送信実行処理	ttSendFrame	MessageSend処理（送信コピー処理）
フレーム受信実行処理	ttReceiveFrame	MessageReceive処理（受信コピー処理）
TT-COMイニシャル関数	ttInitCOM	
TT-COMバッファ初期化関数	ttInitBufferCOM	
TT-COMステータス情報	ttReadFlag	フラグの読み出し
TT-COM周期処理関数	ttTickCOM	途絶判定処理 送信キューに溜まったアプリケーションデータの送信処理
FIFO受信フレーム取得	ttReadFifo	

フィルタ制御は，アプリケーションエンジニアが規定できる要素を残しながら，一般的な処理をあらかじめミドルウェアに含めている機能であり，データがアクティブ（非ゼロ）やデータの変化が発生した場合（前回データと今回データが変化した場合）にデータを送信するなどの条件を指定できる．どちらの場合も，アプリケーションのデータ作成のタイミングで送信データを TT-COM に通達するのみで，指定されたフィルタ条件に一致した場合に送信したり，受信したりすることができる機能を提供する制御である．Que 制御は，短期間に発生する状態変化をすべて送信および受信するために，状態変化時にデータをバッファリングし，送受信する制御である．バッファ構成が Que 構造であるため，Que 制御と呼ぶ．
FlexRay 通信の通信単位は，スロットと呼ばれる単位で通信を行い，そのスロットには複数の情報を保持することができる．多くのスロットは，複数のアプリケーション部位からデータが作られ，それがひとつのスロットにまとめられてデータ送受信が行われる．OS などを用いて，それぞれの部位が非同期で動作している場合，スロット内のまとめられる各データがどのタイミングで集まるか定まらない場合が多く，また，それらのデータをまとめる処理をアプリケーションの処理にするのは移植上問題がある．そのため，これらのメッセージデータをまとめる機能をメッセージ Pack 制御，受信データからメッセージデータを分離する処理をメッセージ UnPack 制御と呼び，FlexRay 通信ミドルウェアに含めている．また，FlexRay 通信バスに接続されている各ノードのエンディアンは統一されているわけではなく，各ノードはデータ送信時に FlexRay 通信で用いるエンディアンに変換し，受信時にはプロセッサエンディアンに変換する必要がある．これらの処理も通信ミドルウェアに含めている．通信途絶制御は，ネットワーク管理（FlexRay-NM）と協調し，接続ノードもしくは自ノードが通信途絶状況に陥っているかどうかを判断し，アプリケーションに通知する機能を提供している．

---

[8] ルネサスは，欧州の BOSCH からライセンスを供与されて FlexRay デバイスを作成しているため，正式には BOSCH 仕様のデバイスと言える．尚，日本の半導体メーカの多くは BOSCH 仕様を利用している．

上記の機能セットは，OSEK COM 仕様[3]においても同等の機能セットを規定しているが，本 TT-COM は，TT-OS より提供されるタイムトリガ機能に同期して送受信が実行できるような機構およびタイミング通知機構を兼ね備えており，FlexRay 通信などの外部ネットワーク時間に同期して処理を必要とする通信システムに利用できるミドルウェアである．

一方，OSEK/VDX は，タイムトリガ通信用の通信ミドルウェアとして，OSEK FT-COM 仕様[4]を規定し，障害対策を考慮した通信ミドルウェアを規定している．本 TT-COM はこの OSEK FT-COM 仕様は参考にしているものの，FT-COM の最大の特徴である Fault Tolerant 部位を取り除いている．FlexRay 通信の場合，FlexRay デバイスで Fault Tolerant を考慮しており，二重化などに対応している．そのため，FlexRay 通信用のソフトウェアミドルウェアに必要な障害対策が何であるべきかが正しく分析できていないため，現 TT-COM はソフトウェアで障害対策をするのではなく，FlexRay デバイスをそのまま利用し，このミドルウェアを利用し，FlexRay 通信を実現した後に，障害対策の必要性などを検討する．

- **FlexRay-NM (Network Management)**

  FlexRay 通信には FlexRay バスに最低 3 ノード (FlexRay コンソーシアムでは最低 4 ノードを推奨)がネットワーク時間を作成するために必要であり，通常，FlexRay バスには複数のノードが接続されている．これらのバスに接続されているノードが，何らかの障害状況に陥った場合や省電力対応などの理由でノードが休止できる場合などのネットワーク状況を管理する必要がある．これらの管理機能は，各ノードに管理機構が必要となるものの，処理内容は共通で良いため，ソフトウェアによるミドルウェアで提供すればよい．本 FlexRay 通信ミドルウェアにおいては，FlexRay-NM として提供している．

  FlexRay-NM も TT-COM 同様に，FlexRay デバイス機能を最大限活用し，FlexRay デバイスが提供するネットワーク管理処理を有効に利用している．具体的には，FlexRay 通信メッセージフレーム内にネットワーク管理用ビットを規定することができ，その管理ビットに各ノードのアライブ状況 (正常起動状況) とスリープ許可状況 (休止可能状況) を割り当てて，ノード管理を行っている．これらの管理ビット操作は，FlexRay デバイスが自動的に処理を行い，ソフトウェアで処理をする必要はない．そのため，FlexRay-NM は，管理ビットを参照し，適切な情報をアプリケーションに通達する機能を提供しているに留めている．

  この方針は，FlexRay 通信のネットワーク管理機能に必要な要求事項が明確でないことと，現在 FlexRay デバイスから提供されるネットワーク管理が改良の余地があると判断しているため，最低限の機能のみ提供することである．

4. 動作環境と開発環境

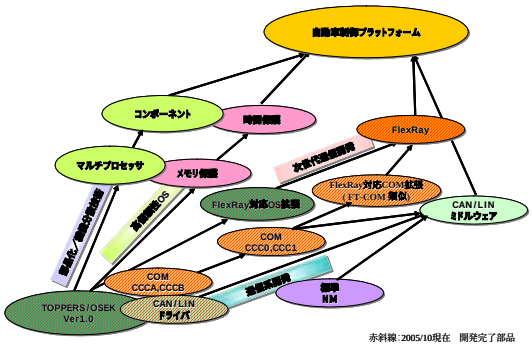
1 章で述べた通り，TT-OS および FlexRay 通信ミドルウェアは，ターゲットプロセッサとして，現時点では，ルネサステクノロジ製 M32C をサポートしている．開発時に用いたターゲットボードは，ルネサステクノロジ製 FlexRay 評価ボード (M3A-0847G21) であり，このボードを用いて動作を確認した．開発環境は，ルネサステクノロジ製コンパイラ (NC308 Version 5.20 Release 02) を用いて開発し，同社製デバッガ(KD3083 Version 3.30 Release 1) ，同社製簡易エミュレータ (M3A-0655 FoUSB) を用いてデバッグを実施した．

5. 開発成果について

応募代表者は，TT-OS および FlexRay 通信ミドルウェアを搭載した「FlexRay Steering by Wire」(以下デモ装置)[9]を実現し，FlexRay 通信技術の検証実験を行い，FlexRay 通信の耐障害性およびイベント処理とタイムトリガ処理の両立が実現できていることを確認した． また，ESEC2005[10]および FlexRay Product Days 2005[11] にデモ装置を出展して本ミドルウェアの有用性を広くアピールし，国内外から問い合わせおよび評価をいただいている．

6. 今後の計画

本 TT-OS および FlexRay 通信ミドルウェアと応募代表者らが中心となり研究開発中の保護機能および現世代通信ミドルウェア (経済産業省の平成 17 年度地域新生コンソーシアム研究開発事業 (中部地区) ：「自動車統合制御用組込み OS の開発」) を結合し次世代の標準的な自動車制御用プラットフォームの叩き台として完成させ，名古屋大学高田･冨山研究室より JasPar に提案する計画である．また，AUTOSAR のメンバ社を通じて AUTOSAR でも検討していただけるように提案したいと考えている．



7. 謝辞

本 TT-OS および FlexRay 通信ミドルウェアを開発するにあたり，トヨタ自動車統合システム開発部の細谷様をはじめ，自動車メーカの立場からご意見をいただきまいた．ここに記して謝意を表します．

文　献

[1] OSEK/VDX，OSEK/VDX Time-Triggered Operating System Version 1.0 ，OSEK VDX，July 24th 2001

[2] OSEK/VDX，OSEK/VDX Operating System Version 2.2.1 ，OSEK VDX，January 16th 2003

[3] OSEK/VDX，OSEK/VDX Communication Version 3.0.3 ，OSEK/VDX，July 20th 2004

[4] OSEK/VDX, OSEK/VDX Fault –Tolerant Communication Version 1.0 ，OSEK/VDX，July 24th 2001

[5] TOPPERS プロジェクトホームページ，http://www.toppers.jp/

[6] 株式会社ヴィッツ，TT-OS Stage-1 外部仕様書 Version 1.0, 2005 年 11 月

[7] 株式会社サニー技研，FlexRay 通信ソフト TT-COM モジュール仕様書 Rev 1.01 ， 2005 年 10 月

9 紹介デモビデオ http://www.witz-inc.co.jp/pi/img/frsbwd.wvx

10 組込みシステム開発技術展 http://www.reedexpo.co.jp/ESEC/

11 http://www.hanser.de/seminare/index.asp?fz_id=2521814468-81&task=010