

(19) 日本国特許庁(JP)

(12) 公開特許公報(A)

(11) 特許出願公開番号

特開2017-128308

(P2017-128308A)

(43) 公開日 平成29年7月27日(2017.7.27)

(51) Int.Cl.		F I		テーマコード (参考)
B60R 16/02 (2006.01)		B60R	16/02	660H
G06F 9/48 (2006.01)		B60R	16/02	660B
		G06F	9/46	457

審査請求 未請求 請求項の数 6 O L (全 22 頁)

(21) 出願番号 特願2016-10889 (P2016-10889)
 (22) 出願日 平成28年1月22日 (2016.1.22)

(71) 出願人 000004260
 株式会社デンソー
 愛知県刈谷市昭和町1丁目1番地
 (74) 代理人 100106149
 弁理士 矢作 和行
 (74) 代理人 100121991
 弁理士 野々部 泰平
 (74) 代理人 100145595
 弁理士 久保 貴則
 (72) 発明者 武田 敏彦
 愛知県刈谷市昭和町1丁目1番地 株式会社デンソー内

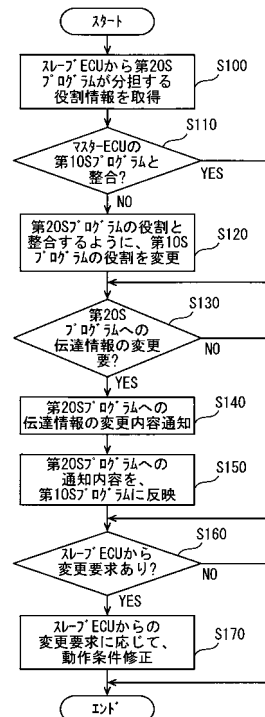
(54) 【発明の名称】 車両用制御システム

(57) 【要約】 (修正有)

【課題】 第1の電子制御装置及び第2の電子制御装置にそれぞれ設けられたOSプログラムにより、第1の電子制御装置と第2の電子制御装置との連携処理を効率的にサポートすること。

【解決手段】 マスターECUが保有する第1OSプログラムは、スレーブECUが保有する第2OSプログラムが分担する役割についての情報を取得S100し、その第2OSプログラムが分担する役割を示す情報に基づき、当該第2OSプログラムと協調して動作できるように、第1OSプログラムの分担する役割を決定S120する。従って、例えば、スレーブECUの能力などに起因して第2OSプログラムの役割分担が変動しても、その役割分担の変動に応じて、第1OSプログラムが、第2OSプログラムと協調して動作できるように、自動的に、分担する役割を変化させることができる。

【選択図】 図4



【特許請求の範囲】

【請求項 1】

車両に搭載された車載機器（31～33）を制御するために、第1の電子制御装置（40）と第2の電子制御装置（60、70、80）とが連携した処理を行う車両用制御システムであって、

前記第1の電子制御装置と前記第2の電子制御装置とは、それぞれ、連携処理を行うための連携処理用アプリケーションプログラム（41a、61a、71a、81a）を有するとともに、それら連携処理用アプリケーションプログラムを協調して実行させるためのベースソフトウェアとしての役割を果たす、所定の役割分担の下に協調して動作する第1OSプログラム（41b）と第2OSプログラム（61b、71b、81b）とをそれぞれ有しており、

10

前記第2の電子制御装置が保有する第2OSプログラムは、分担する役割が異なる複数の中から選定されたものであり、

前記第1の電子制御装置が保有する第1OSプログラムは、分担する役割を変更可能に構成され、それにより、分担する役割が異なる複数の中のいずれの第2OSプログラムとも協調した動作を行うことが可能であって、

前記第1の電子制御装置は、

前記第2の電子制御装置から、前記第2OSプログラムが分担する役割についての情報を取得する取得部（S100）と、

前記取得部により取得した、前記第2OSプログラムが分担する役割を示す情報に基づき、当該第2OSプログラムと協調して動作することができるように、第1OSプログラムの分担する役割を決定する決定部（S120）と、を備える車両用制御システム。

20

【請求項 2】

オペレーティングシステムの役割として複数の項目があり、前記第1OSプログラムと前記第2OSプログラムとは、前記複数の項目について、それぞれ役割分担が定められており、

前記第1の電子制御装置は、

前記第2の電子制御装置に対して、特定の項目に関して伝達する情報の変更を通知する変更通知部（S140）と、

前記変更通知部の変更指示に対応するように、前記特定の項目に関して前記第1OSプログラムが伝達する情報を変更する変更部（S150）と、を有する請求項1に記載の車両用制御システム。

30

【請求項 3】

オペレーティングシステムの役割として複数の項目があり、前記第1OSプログラムと前記第2OSプログラムとは、前記複数の項目について、それぞれ役割分担が定められており、

前記第2の電子制御装置は、

前記第1の電子制御装置に対して、特定の項目の動作条件の変更要求を行う変更指示部（S240）を有し、

前記第1の電子制御装置は、前記変更指示部からの変更指示に応じて、前記特定の項目の前記第1OSプログラムの動作条件を変更する動作条件変更部（S170）を有する請求項1に記載の車両用制御システム。

40

【請求項 4】

前記車両用制御システムは、複数の車載機器を制御するための複数の機器制御部（24～26）と、複数の前記機器制御部の制御を統括するドメイン制御部（21～23）とに階層化され、

前記第1の電子制御装置は、前記ドメイン制御部が実装されたものであり、

前記第2の電子制御装置は、前記機器制御部が実装されたものである請求項1乃至3のいずれかに記載の車両用制御システム。

【請求項 5】

50

前記車両用制御システムは、複数の車載機器を制御するための複数の機器制御部（24～26）と、複数の前記機器制御部の制御を統括するドメイン制御部（21～23）とに階層化され、

前記ドメイン制御部は、少なくとも2つの電子制御装置により構成され、

前記第1の電子制御装置及び前記第2の電子制御装置は、前記ドメイン制御部を構成する少なくとも2つの電子制御装置である請求項1乃至3のいずれかに記載の車両用制御システム。

【請求項6】

前記車両用制御システムは、複数の車載機器を制御するための複数の機器制御部（24～26）と、複数の前記機器制御部の制御を統括するドメイン制御部（21～23）とに階層化され、

さらに、前記ドメイン制御部の上位に位置付けられ、当該ドメイン制御部に対して制御指示を行う上位制御部（13）を備え、

前記第1の電子制御装置は、前記上位制御部が実装されたものであり、

前記第2の電子制御装置は、前記ドメイン制御部が実装されたものである請求項1乃至3のいずれかに記載の車両用制御システム。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、車両に搭載された車載機器を制御するために、第1の電子制御装置と第2の電子制御装置とが連携した処理を行う車両用制御システムに関する。

【背景技術】

【0002】

例えば、特許文献1には、階層的に配置された複数の制御ユニットから構成される車載電子機器制御システムについて記載されている。この車載電子機器制御システムは、例えば、リヤデフォグガーとデフロスターとの動作制御に適用される。車載電子機器制御システムは、リヤデフォグガーの基本制御系の制御ユニットECU₁₁と、その上位に位置付けられる自動化拡張機能系の制御ユニットECU₂₁とを有する。また、車載電子機器制御システムは、デフロスターの基本制御系の制御ユニットECU₁₂と、その上位に位置する自動化拡張機能系の制御ユニットECU₂₂とを有する。さらに、車載電子機器制御システムは、リヤデフォグガーの自動化拡張機能系の制御ユニットECU₂₁及びデフロスターの自動化拡張機能系の制御ユニットECU₂₂の上位に位置付けられる、連携制御ユニットECU₃₁を有している。

【0003】

このような構成において、リヤデフスイッチがオン操作されると、基本制御系の制御ユニットECU₁₁がリヤデフォグガーに通電制御する。このとき、自動化拡張機能系の制御ユニットECU₂₁は、リヤデフォグガーの動作継続時間をタイマー計測し、タイムアップしたときに、リヤデフォグガーをオートオフする自動制御を行う。

【0004】

また、フロントデフスイッチがオン操作されると、基本制御系の制御ユニットECU₂₁が、エアコンの吹出しモードとして、デフロスターを選択する。そして、自動化拡張機能系の制御ユニットECU₂₂は、外気温センサが検出する外気温と、ユーザによるフロントデフスイッチの操作履歴を記録し、窓曇り条件を累積学習する。自動化拡張機能系の制御ユニットECU₂₂は、次回、検出された外気温が窓曇り条件を充足した場合、フロントデフスイッチが操作されずとも、デフロスターをオートオンする自動制御を行う。

【0005】

さらに、連携制御ユニットECU₃₁は、リヤデフォグガーとデフロスターとの動作履歴を記録し、連携動作学習を行う。例えば、デフロスターの動作時に、リヤデフォグガーが一定頻度以上で同時に動作する学習結果となっている場合、デフロスターが動作開始したときに、連携制御ユニットECU₃₁は、リヤデフォグガーの自動化拡張機能系の制御

10

20

30

40

50

ユニット ECU₂₁ に対し、リヤデフォグーをオートオンする指令を出力する。

【先行技術文献】

【特許文献】

【0006】

【特許文献1】特開2009-56817号公報

【発明の概要】

【発明が解決しようとする課題】

【0007】

上述した特許文献1に記載された車載電子機器制御システムのように、制御システムが複数の電子制御装置（制御ユニット）により構成され、各電子制御装置によって連携した制御が行われることがある。このような連携した制御を行うために、例えば、予め電子制御装置間でやり取りされる信号や処理手順などを厳密に定め、それに応じた連携処理用アプリケーションプログラムを作成したとする。この場合、制御システムが、いわゆる専用品となってしまう、例えば車両のグレードや車種に応じた低機能から高機能までのバリエーションに対応したり、電子制御装置を構成するハードウェアの変更に対応したりすることが困難になってしまう。

10

【0008】

そこで、第1の電子制御装置と第2の電子制御装置とが連携した処理を行う場合に、連携処理用アプリケーションの他に、それら連携処理用アプリケーションプログラムを協調して実行させるためのベーシックソフトウェアとしての役割を果たすOSプログラムを、第1及び第2の電子制御装置にそれぞれ設けることが考えられる。

20

【0009】

例えば、OSプログラムが、各電子制御装置間の通信データの検証、データの変換、同期メカニズム、リソースの配分等をサポートすることにより、第1及び第2の電子制御装置における連携処理用アプリケーションプログラムは、それぞれの制御機能に特化して作成することができ、制御機能の追加、変更などにも容易に対応可能となる。また、各OSプログラムが、各電子制御装置のハードウェアを抽象化することができるので、この点においても、連携処理用アプリケーションの汎用性を高めることができる。

【0010】

ただし、第1及び第2の電子制御装置による連携処理をサポートするOSプログラムを、第1及び第2の電子制御装置にそれぞれ設ける場合、それらのOSプログラムは、定められた役割分担の下に協調して動作する必要がある。しかしながら、その役割分担が常に一定であるとする、例えば、用いられる電子制御装置の能力の相違などに起因して、必ずしもそれぞれの電子制御装置の能力に見合った役割とはならず、連携処理のサポートを効率的に行いえない虞がある。

30

【0011】

本発明は、上述した点に鑑みてなされたものであり、第1の電子制御装置及び第2の電子制御装置にそれぞれ設けられたOSプログラムにより、第1の電子制御装置と第2の電子制御装置との連携処理を効率的にサポートすることが可能な車両用制御システムを提供することを目的とする。

40

【課題を解決するための手段】

【0012】

上記目的を達成するために、本発明による車両用制御システムは、

車両に搭載された車載機器（31～33）を制御するために、第1の電子制御装置（40）と第2の電子制御装置（60、70、80）とが連携した処理を行うものであって、

第1の電子制御装置と第2の電子制御装置とは、それぞれ、連携処理を行うための連携処理用アプリケーションプログラム（41a、61a、71a、81a）を有するとともに、それら連携処理用アプリケーションプログラムを協調して実行させるためのベーシックソフトウェアとしての役割を果たす、所定の役割分担の下に協調して動作する第1OSプログラム（41b）と第2OSプログラム（61b、71b、81b）とをそれぞれ有

50

しており、

第2の電子制御装置が保有する第2OSプログラムは、分担する役割が異なる複数の中から選定されたものであり、

第1の電子制御装置が保有する第1OSプログラムは、分担する役割を変更可能に構成され、それにより、分担する役割が異なる複数の中のいずれの第2OSプログラムとも協調した動作を行うことが可能であって、

第1の電子制御装置は、

第2の電子制御装置から、第2OSプログラムが分担する役割についての情報を取得する取得部(S100)と、

取得部により取得した、第2OSプログラムが分担する役割を示す情報に基づき、当該第2OSプログラムと協調して動作することができるように、第1OSプログラムの分担する役割を決定する決定部(S120)と、を備える。

10

【0013】

このように、本発明による車両用制御システムによれば、第1の電子制御装置が保有する第1OSプログラムは、第2OSプログラムが分担する役割についての情報を取得し、その第2OSプログラムが分担する役割を示す情報に基づき、当該第2OSプログラムと協調して動作することができるように、第1OSプログラムの分担する役割を決定する。従って、例えば、第2の電子制御装置の能力に起因して第2OSプログラムの役割分担が変動しても、その役割分担の変動に応じて、第1OSプログラムが、第2OSプログラムと協調して動作することができるように、自動的に、分担する役割を変化させることができる。その結果、第1の電子制御装置及び第2の電子制御装置に設けられたOSプログラムにより、第1の電子制御装置と第2の電子制御装置との連携処理を効率的にサポートすることが可能になる。

20

【0014】

上記括弧内の参照番号は、本発明の理解を容易にすべく、後述する実施形態における具体的な構成との対応関係の一例を示すものにすぎず、なんら本発明の範囲を制限することを意図したものではない。

【0015】

また、上述した特徴以外の、特許請求の範囲の各請求項に記載した技術的特徴に関しては、後述する実施形態の説明及び添付図面から明らかになる。

30

【図面の簡単な説明】

【0016】

【図1】実施形態に係る車両用制御システムの構成を示す構成図である。

【図2】運動ドメイン制御部のPTCが、電子制御装置にPTCアプリケーションとして実装され、その電子制御装置のアプリケーションソフトの1つとしてOSプログラムを設けた例を示す図である。

【図3】OSプログラムが提供するサービスの一覧の例を示す図である。

【図4】マスターECUにおける第1OSプログラムの具体的な処理内容を示すフローチャートである。

【図5】スレーブECUにおける第2OSプログラムの具体的な処理内容を示すフローチャートである。

40

【図6】マスターECUの第1OSプログラムの役割を軽くし、スレーブECUにおける第2OSプログラムの役割を重くした例を示す図である。

【図7】マスターECUの第1OSプログラムの役割を重くし、スレーブECUにおける第2OSプログラムの役割を軽くした例を示す図である。

【図8】マスターECUの第1OSプログラムが、第2OSプログラムへの伝達情報の変更が必要と判定するいくつかの例を説明するための説明図である。

【図9】マスターECUの第1OSプログラムが、スレーブECUの第2OSプログラムから動作条件の変更要求を受けるいくつかの例を説明するための説明図である。

【発明を実施するための形態】

50

【 0 0 1 7 】

以下、本発明の実施形態に係る車両用制御システムについて、図面を参照しつつ説明する。本実施形態に係る車両用制御システム 1 は、例えば、図 1 に示すように、走行駆動源として、エンジン 3 1 と電動モータ（モータジェネレータ）3 3 とを有するハイブリッド車両に適用され、このハイブリッド車両に搭載された各種の車載機器 3 1 ~ 3 5 を制御するために用いられる。しかしながら、本実施形態に係る車両用制御システム 1 は、ハイブリッド車両における車載機器 3 1 ~ 3 5 を制御に適用されることに限られる訳ではなく、エンジンのみを有する通常の車両や、電動モータのみを有する電動車両の車載機器の制御に適用されても良い。

【 0 0 1 8 】

図 1 は、上述したハイブリッド車両における複数の車載機器 3 1 ~ 3 5 を制御するために、車両用制御システム 1 が有する各種機能の一例を機能ブロック図として表したものである。ただし、図 1 には、車両用制御システム 1 が有する機能の全てが示されている訳ではない。これは、説明の便宜のため、図 1 には、本実施形態に係る車両用制御システム 1 の構成の一例しか示していないためである。

【 0 0 1 9 】

図 1 において、車両用制御システム 1 は、車載機器としてのエンジン 3 1 , トランスミッション (T M) 3 2 、モータジェネレータ (M G) 3 3 、ブレーキ装置 3 4 、及びステアリング装置 3 5 などを制御するための機能を有する。しかしながら、上述したように、車両用制御システム 1 は、さらに、例えば、サスペンション、高圧バッテリー、エアコン装置などのその他の車載機器を制御するための機能を有していても良い。

【 0 0 2 0 】

図 1 に示すように、車両用制御システム 1 は、各種の車載機器 3 1 ~ 3 5 を制御するための機能が予め複数の論理ブロック（機能ブロック）1 2 ~ 1 6、2 2 ~ 2 6 に分けられ、それら複数の論理ブロック 1 2 ~ 1 6、2 2 ~ 2 6 間の連結関係を規定することによって構成されている。すなわち、車両用制御システム 1 における各種の車載機器 3 1 ~ 3 5 を制御するための論理構造が、論理ブロック 1 2 ~ 1 6、2 2 ~ 2 6 と、それら論理ブロック 1 2 ~ 1 6、2 2 ~ 2 6 間の連結関係によって規定されている。そして、車両用制御システム 1 は、複数の論理ブロック 1 2 ~ 1 6、2 2 ~ 2 6 が、規定された連結関係に従って連携して動作することにより、各種の車載機器 3 1 ~ 3 5 を制御する。

【 0 0 2 1 】

なお、図 1 には示していないが、各論理ブロック 1 2 ~ 1 6、2 2 ~ 2 6 は、少なくとも 1 つ、通常は多数の制御ブロックを有している。各論理ブロック 1 2 ~ 1 6、2 2 ~ 2 6 は、それら多数の制御ブロックにおける演算処理を適宜組み合わせることにより、それぞれの機能（役割）を発揮する。

【 0 0 2 2 】

例えば、論理ブロックとしてのエンジン制御部 2 4 は、エンジン 3 1 の運転状態を検出するべく、各種のセンサからのセンサ信号を入力して、論理ブロック内で取り扱うことができる信号に変換する制御ブロックを有する。また、センサ信号から把握されるエンジン 3 1 の運転状態から現状の発生トルクを算出する制御ブロックを有する。さらに、上位の論理ブロック（パワートレインコーディネータ (P T C) 2 2) から指示された指令トルクと現状の発生トルクとに差異がある場合に、その差異をなくすための目標とするエンジン運転状態を算出する制御ブロックを有する。また、目標エンジン運転状態を達成するためのスロットルバルブ開度、燃料噴射量と燃料噴射時期、及び点火時期を算出する制御ブロックを有する。ただし、これらは単なる例示であって、エンジン制御部 2 4 は、その機能を発揮するために必要な、その他の演算処理を行う制御ブロックを有する場合もあり得る。また、例示された制御ブロックを含め、エンジン制御部 2 4 内の制御ブロックは、適宜、統合されたり、逆に、細分化されたりすることが可能なものである。

【 0 0 2 3 】

車両用制御システム 1 は、実際には、各論理ブロック 1 2 ~ 1 6、2 2 ~ 2 6 を、プロ

10

20

30

40

50

グラムやデータベースなどの制御アプリケーションとして、複数の電子制御装置に振り分けて実装することにより具現化される。この場合、複数の電子制御装置は、論理ブロック 12 ~ 16、22 ~ 26 の連結関係を維持できるように、個別の通信線を介して接続されたり、各電子制御装置が共通のネットワークに接続され、連結関係に従う所望の電子制御装置同士が通信可能に構成されたりする。なお、必ずしも各論理ブロック 12 ~ 16、22 ~ 26 をそれぞれ別個の電子制御装置に実装する必要はなく、幾つかの論理ブロックを共通の電子制御装置に実装しても良い。

【0024】

本実施形態に係る車両用制御システム 1 は、複数の車載機器 31 ~ 35 の機能に応じて予め複数のドメインに区分けされている。別の表現をすれば、各論理ブロック 12 ~ 16、22 ~ 26 が果たすべき機能のまとまりに応じて、予め複数のドメイン 10、20 に区分けされている。

10

【0025】

複数の車載機器 31 ~ 35 は、いずれかのドメイン 10、20 に割り振られる。そして、割り振られたドメイン 10、20 に属するドメイン制御部 11、21 が、各車載機器 31 ~ 35 の制御目標値を算出して、車載機器 31 ~ 35 の制御を司る車載機器制御部 15、16、24 ~ 26 へ出力したりする。なお、図 1 に示すように、各ドメイン制御部 11、21 は、少なくとも 1 つの論理ブロックから構成される。

【0026】

具体的には、図 1 に示す例では、制御システム 1 は、運動ドメイン 10 とエネルギードメイン 20 とに区分けされている。そして、運動ドメイン 10 には、前後方向及び横方向における車両の挙動を制御する機能を担う運動ドメイン制御部 11 が設けられている。また、エネルギードメイン 20 には、車両を加速させたり、減速させたり、あるいは速度を一定に保つように、車両の動力を制御する機能を担うエネルギードメイン制御部 21 が設けられている。

20

【0027】

また、本実施形態に係る車両用制御システム 1 では、図 1 に示すように、各ドメイン制御部 11、21 の下に、対応するドメイン制御部 11、21 からの指令（制御目標値）に従い、各車載機器 31 ~ 35 の動作状態を制御する車載機器制御部 15、16、24 ~ 26 が設けられている。これら車載機器制御部 15、16、24 ~ 26 は、各車載機器 31 ~ 35 の動作状態をドメイン制御部 11、21 からの制御目標値に近づけるための制御信号を生成し、各車載機器 31 ~ 35 に出力する。

30

【0028】

具体的には、図 1 に示す例では、運動ドメイン制御部 11 の下に、ブレーキ装置 34 を制御するブレーキ制御部 15 及びステアリング装置 35 を制御するステアリング制御部 16 が設けられている。また、エネルギードメイン制御部 21 の下に、エンジン 31 を制御するエンジン制御部 24、トランスミッション 32 を制御するトランスミッション（TM）制御部 25、及びモータジェネレータ 33 を制御する MG 制御部 26 が設けられている。

【0029】

なお、モータジェネレータ 33 は、車両の減速時などに回生エネルギーを発生する。MG 制御部 26 の上位の論理ブロックであるモータジェネレータコディネータ（MGC）23 は、その回生エネルギーの生成も管理する。このエネルギーは、インバータによって DC 変換され、図示しない高圧バッテリーに蓄電される。

40

【0030】

ここで、図 1 に論理ブロック 12 ~ 16、22 ~ 26 として例示した、車両用制御システム 1 が有する各種の機能について詳しく説明する。

【0031】

車両用制御システム 1 には、各論理ブロック 12 ~ 16、22 ~ 26 が与えられた機能を発揮するために必要な各種の情報が入力される。例えば、図示しない各種のセンサによ

50

って、ハイブリッド車両の運転のため、運転者によって操作される各種の操作部（アクセルペダル、ブレーキペダル、シフトレバー、ステアリングホイールなど）の操作が検出され、その操作検出信号が車両用制御システム 1 に入力される。また、車両の走行状態（例えば、速度、加速度、ヨーレートなど）や、各種の車載機器 3 1 ~ 3 5 の動作状態（例えば、エンジン温度、エンジン回転数、トランスミッション変速比、インバータ温度、モータ回転数、ブレーキ油圧、操舵角など）を検出するセンサからの動作検出信号も、車両用制御システム 1 に入力される。

【 0 0 3 2 】

上述した各種の信号は、車両用制御システム 1 の各ドメイン制御部 1 1、2 1 や車載機器制御部 1 5、1 6、2 4 ~ 2 6 に与えられる。

10

【 0 0 3 3 】

例えば、運動ドメイン制御部 1 1 には、運転者による各種の操作部の操作を示す操作検出信号、及び車両の走行状態を検出するセンサからの動作検出信号が入力される。そして、運動ドメイン制御部 1 1 は、原則として、車両が運転者による操作部の操作に応じた挙動を示すように、ブレーキ装置 3 4 及びステアリング装置 3 5 の制御目標値を算出する。具体的には、車両挙動制御部 1 2 が、車両の挙動を安定させつつ、運転者の操作に対応するように車両の挙動を制御すべく、前後挙動制御部 1 3 に対して前後方向の目標加速度（目標減速度）を与えるとともに、左右挙動制御部 1 4 に対して左右方向の目標加速度を与える。前後挙動制御部 1 3 は、与えられた前後方向の目標加速度（目標減速度）を実現すべく、エネルギードメイン制御部 2 1 のパワートレインコーディネータ（PTC）2 2 に対して目標駆動トルク（加速トルク又は制動トルク）を出力するとともに、ブレーキ制御部 1 5 に対して、ブレーキ装置 3 4 の制御目標値である目標制動トルクを出力する。また、左右挙動制御部 1 4 は、与えられた左右方向の目標加速度を実現すべく、ステアリング制御部 1 6 に対して、ステアリング装置 3 5 の制御目標値である目標アシストトルクを出力する。

20

【 0 0 3 4 】

なお、例えば、運動ドメイン制御部 1 1 に、車両の走行車線を区画する白線の認識情報や、先行車両や障害物の情報など、車両の外部環境に関する情報を与えるようにしてもよい。これにより、運動ドメイン制御部 1 1 において、例えば、白線によって区画される走行車線を逸脱しないように、ステアリング装置 3 5 のアシスト力を調整する（レーンキープアシスト）ように制御目標値を算出することが可能となる。また、運動ドメイン制御部 1 1 にて、例えば、先行車両や障害物との衝突を避けるように、ブレーキ装置 3 4 やステアリング装置 3 5 の制御目標値を算出することが可能となる。

30

【 0 0 3 5 】

また、エネルギードメイン制御部 2 1 には、例えば、図示しない高圧バッテリーの電圧や電流を検出するセンサ信号や、車両の走行状態を示すセンサ信号などが入力される。エネルギードメイン制御部 2 1 の MGC 2 3 は、それらのセンサ信号に基づいて、高圧バッテリーの蓄電量を算出する。さらに、MGC 2 3 は、主として、高圧バッテリーの蓄電量に基づいて、モータジェネレータ 3 3 が発生可能な最大 MG トルクを算出して、PTC 2 2 に与える。PTC 2 2 は、運動ドメイン制御部 1 1 から与えられた目標駆動トルク（加速トルク）を最も効率良く実現するために、モータジェネレータ 3 3 が発生可能な最大 MG トルクや、センサ信号に基づく車両の走行状態を考慮しつつ、エンジン 3 1 が発生すべき目標エンジントルク、トランスミッション 3 2 が実現すべき目標変速比、及びモータジェネレータ 3 3 が発生すべき目標 MG トルクを算出する。算出された目標エンジントルク、目標変速比、及び目標 MG トルクは、それぞれ、制御目標値として、エンジン制御部 2 4、TM 制御部 2 5、及び MGC 2 3 に与えられる。また、TM 制御部 2 5 に対しては、クラッチの動作に関する制御目標値（クラッチの接続開始時期や、クラッチの接続完了までの時間など）も与えられても良い。

40

【 0 0 3 6 】

さらに、エネルギードメイン制御部 2 1 の MGC 2 3 は、車両の減速時等において、主

50

として、高圧バッテリーの蓄電量に基づいてモータジェネレータ33が発生可能な回生電力量を算出する。この回生電力量に対応する回生制動トルクに関する情報は、MGC23から、PTC22を介して運動ドメイン制御部11の前後挙動制御部13に与えられる。

【0037】

前後挙動制御部13は、車両挙動制御部12から目標減速度が与えられた場合、その目標減速度を実現するための目標制動トルクを算出する。そして、モータジェネレータ33が回生制動トルクを発生可能である場合には、極力、その回生制動トルクを活用するように、ブレーキ装置34による目標制動トルクと、回生ブレーキによる目標回生制動トルクとを定める。この目標回生制動トルクは、制御目標値として、PTC22を介して、エネルギードメイン制御部21のMGC23に与えられる。

10

【0038】

ブレーキ制御部15は、前後挙動制御部13から与えられたブレーキ装置34の制御目標値である目標制動トルクに従い、ブレーキ装置34を制御する。より具体的には、ブレーキ制御部15は、ブレーキ装置34が目標制動トルクを発生するようにブレーキフルード圧を制御するための制御信号を出力する。また、ステアリング制御部16も、ステアリング装置35が左右挙動制御部14から与えられた目標アシストトルクを発生するように制御信号を出力して、ステアリング装置35におけるアシストトルクを制御する。

【0039】

エンジン制御部24は、PTC22から与えられた目標エンジントルクを実現するための制御信号をエンジン31に出力する。より詳細には、エンジン制御部24は、エンジン31の運転状態を検出する各種のセンサ（回転数、温度、空気流量等）からのセンサ信号を入力する。そして、センサ信号から把握されるエンジンの運転状態から現状の発生トルクを算出する。エンジン制御部24は、現状の発生トルクを目標エンジントルクに近づけるためのエンジン運転状態を算出し、その算出したエンジン運転状態を達成するための燃料噴射量と燃料噴射時期、及び点火時期を求め、これらに応じた噴射制御信号及び点火制御信号をエンジン31に出力する。

20

【0040】

TM制御部25も、PTC22から目標変速比が与えられ、この与えられた目標変速比を実現するための制御信号をトランスミッション32に出力する。また、TM制御部25は、トランスミッション32において変速比を変更する場合、クラッチの動作に関する制御目標値に従って、クラッチの動作を制御するための制御信号も出力する。

30

【0041】

MG制御部26は、MGC23から目標MGトルクが与えられた場合には、その目標MGトルクを発生させるように、モータジェネレータ33のインバータへ制御信号を出力する。一方、MG制御部26は、MGC23から目標回生制動トルクが与えられた場合には、その目標回生制動トルクに相当する制動力をモータジェネレータ33が車軸に対して付与するように、モータジェネレータ33のインバータへ制御信号を出力する。

【0042】

上述したように、車両用制御システム1を構成する各論理ブロック12～16、22～26は、複数の論理ブロック12～16、22～26により連携した制御処理を行うことで、各車載機器31～35を制御する。例えば、PTC22は、運動ドメイン10における前後挙動制御部13と協調したドメイン間連携制御、及び、MGC23、エンジン制御部24、並びにTM制御部25と協調したドメイン内連携制御を実行する。

40

【0043】

ここで、上述した車両用制御システム1において、各論理ブロック12～16、22～26が実装される電子制御装置間でやり取りされる信号や処理手順などを厳密に定め、それに応じた連携処理用アプリケーションプログラムを作成することも可能である。しかしながら、この場合、車両用制御システム1が、いわゆる専用品となってしまう、例えば車両のグレードや車種に応じた低機能から高機能までのパリエーションに対応したり、電子制御装置を構成するハードウェアの変更に対応したりすることが困難になってしまう。

50

【 0 0 4 4 】

そこで、各論理ブロック 1 2 ~ 1 6、2 2 ~ 2 6 が実装される複数の電子制御装置が連携した処理を行う場合に、連携処理用アプリケーションの他に、それら連携処理用アプリケーションプログラムを協調して実行させるためのベシックソフトウェアとしての役割を果たす OS プログラムを、各電子制御装置に設けることが考えられる。

【 0 0 4 5 】

例えば、OS プログラムが、各電子制御装置間における通信データの検証、データの交換、同期メカニズム、リソースの配分等をサポートすることにより、各電子制御装置における連携処理用アプリケーションプログラムは、それぞれの制御機能に特化して作成することができ、制御機能の追加、変更などにも容易に対応可能となる。また、各 OS プログラムが、各電子制御装置のハードウェアを抽象化することができるので、この点においても、連携処理用アプリケーションの汎用性を高めることができるようになる。

10

【 0 0 4 6 】

ただし、各電子制御装置による連携処理をサポートする OS プログラムを、各電子制御装置にそれぞれ設ける場合、それらの OS プログラムは、定められた役割分担の下に協調して動作する必要がある。しかしながら、その役割分担が常に一定であるとすると、例えば、用いられる電子制御装置の能力の相違に起因して、必ずしもそれぞれの電子制御装置の能力に見合った役割とはならず、連携処理のサポートを効率的に行いえない虞がある。

【 0 0 4 7 】

そこで、本実施形態に係る車両用制御システム 1 では、各電子制御装置による連携処理をサポートする OS プログラムの役割を変更可能に構成した。具体的には、連携した処理を行う少なくとも 2 つの電子制御装置において、一方をマスター ECU、他方をスレーブ ECU とした場合に、まず、マスター ECU が保有する第 1 OS プログラムは、スレーブ ECU が保有する第 2 OS プログラムが分担する役割についての情報を取得する。そして、第 1 OS プログラムは、第 2 OS プログラムが分担する役割を示す情報に基づき、当該第 2 OS プログラムと協調して動作することができるように、第 1 OS プログラムの分担する役割を決定する。従って、例えば、スレーブ ECU の能力に起因して第 2 OS プログラムの役割分担が変動しても、その役割分担の変動に応じて、第 1 OS プログラムが、第 2 OS プログラムと協調して動作することができるように、自動的に、分担する役割を変化させることができる。その結果、マスター ECU 及びスレーブ ECU に設けられた各 OS プログラムにより、マスター ECU とスレーブ ECU との連携処理を効率的にサポートすることが可能になる。

20

30

【 0 0 4 8 】

まず、図 2 を参照して、各電子制御装置に OS プログラムを設ける場合の構成の一例を説明するとともに、OS プログラムが連携処理をサポートするために実施する処理内容などの具体例を説明する。

【 0 0 4 9 】

図 2 は、エネルギードメイン制御部 2 1 の PTC 2 2 が、電子制御装置 4 0 に PTC アプリケーション 4 1 a として実装され、その電子制御装置 4 0 にアプリケーションの 1 つとして OS プログラム 4 1 b を設けた場合の構成を示している。なお、図 2 には、PTC 2 2 と直接的に通信を行って連携処理を実行する論理ブロック（前後挙動制御部 1 3、MGC 2 3、エンジン制御部 2 4、TM 制御部 2 5）として機能するアプリケーション 5 1 a、6 1 a、7 1 a、8 1 a をそれぞれ実装した電子制御装置 5 0、6 0、7 0、8 0 の構成も示している。これら電子制御装置 5 0、6 0、7 0、8 0 においても、アプリケーションの 1 つとして、それぞれ OS プログラム 5 1 b、6 1 b、7 1 b、8 1 b が設けられている。

40

【 0 0 5 0 】

なお、PTC アプリケーション 4 1 a、前後挙動制御アプリケーション 5 1 a、MGC アプリケーション 6 1 b、TM 制御アプリケーション 7 1 a、EMS 制御アプリケーション 8 1 a が、それぞれ連携処理用アプリケーションに相当する。

50

【0051】

また、図2には、各電子制御装置40、50、60、70、80が、それぞれ1つのOSプログラム41b、51b、61b、71b、81bを有するように示されているが、各電子制御装置40、50、60、70、80は、通信相手毎に、異なるOSプログラムを有する。通信相手が異なることにより、連携処理をサポートするために提供すべきサービス内容も異なるためである。例えば、PTCアプリケーション41aが実装された電子制御装置40は、上位のドメイン制御部である前後挙動制御部13としての電子制御装置50との連携処理をサポートするためのOSプログラム、同じエネルギードメイン制御部21に属するMGC23としての電子制御装置60との連携処理をサポートするためのOSプログラム、下位の機器制御部であるTM制御部25としての電子制御装置70との連携処理をサポートするためのOSプログラム、及びエンジン制御部24としての電子制御装置80との連携処理をサポートするためのOSプログラムをそれぞれ有している。

10

【0052】

ただし、PTC22としての電子制御装置40は、機器制御部に分類されるTM制御部25、エンジン制御部24がそれぞれ実装された電子制御装置70及び電子制御装置80との連携処理をサポートするためのOSプログラムを共通化しても良い。この場合、電子制御装置70及び電子制御装置80のOSプログラムの役割や動作条件が相違していてもそれらの相違を包含するデータやサービスを提供するように、電子制御装置40のOSプログラムを設定することで、OSプログラムを共通化することができる。

20

【0053】

そして、原則として、上位の電子制御装置がマスターECUとなり、下位の電子制御装置がスレーブECUとなる。なお、同位の電子制御装置の場合、いずれか一方がマスターECUとなり、他方がスレーブECUとなる。

【0054】

電子制御装置40～80は、それぞれ、メモリを有しており、図2に示す構造のソフトウェアを、それぞれのメモリに記憶している。各電子制御装置40～80は、図2に示すように、いわゆるAUTOSAR(AUTomotive Open System ARchitecture)に準拠するソフトウェア構造を有している。そこで、PTCアプリケーション41aを実装した電子制御装置40を代表例として、各電子制御装置40～80のソフトウェア構造について説明する。

30

【0055】

電子制御装置40では、図2に示すように、ハードウェアであるマイクロコントローラ47に対して、ソフトウェアが搭載され、そのソフトウェアは、大きくは、基本ソフトウェア43～46、ランタイム環境(RTE)42、及び、アプリケーション層に含まれるアプリケーションソフトウェアに分けられる。その中で、PTCアプリケーション41aは、アプリケーション層に実装され、このPTCアプリケーション41aが実行されることにより、電子制御装置40は、PTC22としての機能を発揮する。PTCアプリケーション41aは、実際には、PTC22としての各種の機能ごとに、複数のサブアプリケーションに分割されている。また、OSプログラム41bも、一つのアプリケーションソフトウェアとして、アプリケーション層に実装されている。

40

【0056】

基本ソフトウェア43～46は、マイクロコントローラ抽象化層43、ECU抽象化層44、サービス層45に階層化され、階層が高くなるほど、抽象化度合いが強くなり、各種のハードウェアから独立するようになっていく。また、基本ソフトウェア43～46は、複合ドライバ46を含んでいる。

【0057】

マイクロコントローラ抽象化層43は、基本ソフトウェア43～46の最下層に位置し、マイクロコントローラドライバ、メモリドライバ、通信ドライバ、I/Oドライバなどを有する。このマイクロコントローラ抽象化層43は、マイクロコントローラ47のハードウェア構成に依存する部分である。しかし、このマイクロコントローラ抽象化層43に

50

よって、マイクロコントローラ 47 とその周辺機器とが抽象化されるため、これより上位の階層は、マイクロコントローラ 47 及び周辺機器から独立することになる。なお、マイクロコントローラ抽象化層 43 は、通信ドライバとして、複数の種類の通信ドライバ（例えば、CAN（登録商標）ドライバ、LINドライバ、FlexRay（登録商標）ドライバなど）を有している。そして、電子制御装置 40 は、例えば、通信相手の電子制御装置がドメイン内か、ドメイン外かなどにより、異なる通信ドライバ、すなわち異なる通信ネットワークを使用して通信を行う。

【0058】

ECU抽象化層 44 は、マイクロコントローラ抽象化層 43 の上位に位置している。この ECU 抽象化層 44 は、電子制御装置 40 の基本コンポーネントを抽象化することにより、上位のソフトウェア層を、電子制御装置 40 のハードウェアから独立させるものである。この ECU 抽象化層 44 は、搭載機器抽象化、メモリハードウェア抽象化、通信ハードウェア抽象化、I/Oハードウェア抽象化などを含む。

10

【0059】

サービス層 45 は、その一部が、ECU抽象化層 44 の上位に位置し、アプリケーションのための基本的なサービスを提供するためのものである。具体的には、サービス層 45 は、OS、車両ネットワークの通信と管理、メモリサービス、診断サービス、ECU状態管理などのサービスを提供する。このサービス層 45 は、大部分がハードウェアから独立している。

【0060】

複合ドライバ 46 は、複雑なセンサやアクチュエータを操作するための特殊な機能やタイミング要求を満たすような、他のレイヤにはない複雑な機能が必要な場合に使用されるものである。この複合ドライバ 47 は、例えば、燃料を噴射するインジェクタを駆動制御する場合などに使用される。

20

【0061】

そして、RTE 42 は、上述した各層からなる基本ソフトウェア 43 ~ 46 の上位に位置し、アプリケーション層に含まれる各種のアプリケーション 41a、41b が、電子制御装置 40 に依存しないようにするためのものである。そのため、RTE 42 は、アプリケーション 41a、41b 間の通信や、アプリケーション 41a、41b と基本ソフトウェア 43 ~ 46 との通信を提供する。

30

【0062】

このように、AUTOSAR に準拠したソフトウェア構造を採用することで、アプリケーション 41a、41b は、マイクロコントローラ 47 や電子制御装置 40 のハードウェアに依存せずに済むので、アプリケーション 41a、41b の再利用性を高めることができる。

【0063】

次に、OSプログラム 41b について、詳しく説明する。図 2 に示すように、OSプログラム 41b は、アプリケーションソフトの 1 つとして具現化される。この OSプログラム 41b は、PTCアプリケーション 41b の各サブアプリケーションが、他の電子制御装置 50 ~ 80 と連携した処理を行う際に、その連携処理をサポートするための各種のサービスを提供する。なお、図 2 に示す例では、現状の AUTOSAR に準拠した基本ソフトウェアを前提とし、ECU間連携を実現する OSプログラムは、アプリケーションソフトとして、RTE を介し基本ソフトウェアと結合されている。しかし、OSプログラムは、アプリケーションソフトとして具現化する以外に、基本ソフトウェア内に組み入れることも可能である。

40

【0064】

図 3 に、OSプログラム 41b が提供するサービスの一例を示す。なお、図 3 においては、OSプログラム 41b により提供される各種のサービスを、DVFB（Domain Virtual Function Bus）、ドメインシステム OS（Domain System OS）、ドメインシステム抽象化（Domain System Abstraction）の 3 つの階層に分類して示している。OSプロ

50

グラム 4 1 b は、図 3 に示すサービスの一部を提供するものであっても良いし、逆に、図 3 に示すサービスは一例であって、図 3 に示すサービス以外のサービスをも含むものであっても良い。

【 0 0 6 5 】

D V F B に分類されるドメイン外内抽象化 (Domain External-Internal Abstraction) サービスは、例えば、マスター E C U からスレーブ E C U に、連携処理を行うためのデータ (制御目標データやセンサ検出データなど) が送信される場合に、マスター E C U とドメイン E C U において実行される O S プログラムの各サービスの処理順を設定したり、そのサービスが実行される時間に関する整合を図ったりするものである。また、D V F B に分類される意味解釈ゲートウェイ (Semantic Gateway) サービスは、ドメイン外内にてデータが異なる場合に、対応するデータに翻訳するものである。

10

【 0 0 6 6 】

ドメインシステム O S に分類されるリソース調停 (Resource Arbitration) サービスは、例えば、マスター E C U とスレーブ E C U とが同じ電子制御装置に実装され、メモリやコアなどのリソースを共用する場合に、そのリソースの配分や調整を行うものである。システムモード (System Mode) サービスは、マスター E C U における制御モードと、スレーブ E C U における制御モードとが整合した制御モードとなるように、モード管理を行うものである。イベント管理 (Event Manager) サービスは、例えばマスター E C U とスレーブ E C U とで同期した処理を行わせるトリガとなるイベントの生成と配信を行うものである。スケジューラ (Scheduler) サービスは、マスター E C U での連携処理のスケジュールに基づいて、スレーブ E C U での連携処理のスケジュールを定めるものである。同期メカニズム (Synchronized Mechanism) サービスは、イベントもしくは時間での制御の同期を図るためのものである。クロック (Clock) サービスは、マスター E C U とスレーブ E C U とで共通の時刻を生成するとともに、必要に応じて、データにタイムスタンプを行うものである。セキュリティ (Security) サービスは、例えば I D データに基づいて正規の通信相手を認証したり、パリティチェックなどにより通信データの正常性を確認したりするためのものである。診断 (Diagnostic) サービスは、例えば、マスター E C U 及びスレーブ E C U の異常診断を行うとともに、必要に応じてフェールセーフ処理を実行するものである。安全機能メカニズム (Safety function Mechanism) サービスは、異常が重大である場合に、マスター E C U とスレーブ E C U との連携機能を停止するものである。

20

30

【 0 0 6 7 】

ドメインシステム抽象化に分類される機能分散抽象化 (Function Distribution Abstraction) サービスは、上述した各種のサービスを実現する上で、マスター側の第 1 O S プログラムと、スレーブ側の第 2 O S プログラムとの役割分担の管理を行うものである。つまり、本実施形態では、マスター側の第 1 O S プログラムとスレーブ側の第 2 O S プログラムとが協調して動作することにより、上述した各種のサービスを実現する。その際、本実施形態では、マスター側の第 1 O S プログラムとスレーブ側の第 2 O S プログラムとの役割を変更可能に構成している。そのため、マスター側の第 1 O S プログラムとスレーブ側の第 2 O S プログラムとが、それぞれ、どのような役割を担っているかを管理することが必要となる。支配抽象化 (Ownership Abstraction) サービスは、マスター側の第 1 O S プログラムとスレーブ側の第 2 O S プログラムとによる、制御対象機器やセンサ機器などに対する制御責任の分担の管理を行うものである。なお、制御責任とは、制御対象機器に対して制御信号を出力したり、センサ機器からの検出信号の受信処理を行ったり、制御対象機器やセンサ機器が正常に動作しているか監視したりする役割を担っていることをいう。

40

【 0 0 6 8 】

次に、図 4 及び図 5 のフローチャートを参照しつつ、マスター側の第 1 O S プログラムとスレーブ側の第 2 O S プログラムとが協調して動作可能とするための、設定方法について説明する。なお、第 1 O S プログラム及び第 2 O S プログラムの設定は、車両の製造ラインにおいて、車両用制御システムが構築された段階で実施される。また、機能向上のた

50

めに、マスターECUやスレーブECUのアプリケーションソフトウェアが更新された場合にも実施される。

【0069】

図4のフローチャートは、マスター側の第1OSプログラムによって実行される処理を示し、図5のフローチャートは、スレーブ側の第2OSプログラムによって実行される処理を示している。

【0070】

まず、図4のフローチャートのステップS100では、スレーブECUから、スレーブECUが保有する第2OSプログラムが分担する役割情報を取得する。この役割情報として、マスターECUは、上述した各種サービスに関して、スレーブECUの第2OSプログラムが担う役割を示す情報をそれぞれ取得しても良いし、第2OSプログラムに、予め、各種サービスに関する第2OSプログラムの役割の組み合わせを一義的に示す役割パターン情報を保有させておき、その役割パターン情報を取得しても良い。

10

【0071】

このように、スレーブECUが保有する第2OSプログラムは、上述した各種サービスに関して、第2OSプログラムが分担する役割が異なる複数のパターンの中から選定されたものである。このため、車両用制御システムの構成や、スレーブECUの能力などに応じて、第2OSプログラムの役割を適切に設定することが可能になる。

【0072】

続くステップS110では、取得した第2OSプログラムの各種サービスに関する役割が、マスターECUにおける第1OSプログラムの現状の役割と整合しているか否かを判定する。すなわち、マスターECUの第1OSプログラムは、平均的な第2OSプログラムの役割に対応するように、予め設定されている。ステップS110では、このような第1OSプログラムの基本設定で、既に第2OSプログラムの役割に整合しているか否かを判定する。このようにすれば、続くステップS120での第1OSプログラムの変更量を抑えることが可能になる。ただし、第1OSプログラムを事前に設定せずに、取得した第2OSプログラムに対応するように、すべてのサービス項目について、設定するようにしても良い。その場合、ステップS110の処理は省略可能である。

20

【0073】

ステップS120では、各種サービスに関する第2OSプログラムの役割と整合するように、第1OSプログラムの役割を変更する。この際、第2OSプログラムの役割と整合していないサービス項目に関してのみ、第1OSプログラムの役割を変更すれば良いので、第1OSプログラムの変更処理を簡便に行うことができる。

30

【0074】

ここで、第1OSプログラムと第2OSプログラムとの役割が、各サービス項目に関して、どのように変化するかに関して、典型例を2例ほど示す。

【0075】

図6は、マスターECUの第1OSプログラムの役割を軽くし、スレーブECUにおける第2OSプログラムの役割を重くした例を示している。

【0076】

この場合、ドメイン外内抽象化サービスについて、マスターECUの第1OSプログラムは、スレーブECUの第2OSプログラムによる各サービスの処理順処理時間の設定には関与せず、スレーブECUの第2OSプログラムが、当該第2OSプログラムによるサービスの処理順や処理時間の設定を行う。また、意味解釈ゲートウェイサービスに関しては、マスターECUから送信されるデータの意味翻訳を、第2OSプログラムにおいて実行する。

40

【0077】

リソース調停サービスに関しては、マスターECUにおける連携処理を優先して実行するために、第1OSプログラムが、その連携処理のためのリソースを確保する処理を行うが、残りリソースの利用調整は、第2OSプログラムにおいて行う。システムモードサー

50

ビスに関しては、第1OSプログラムが連携処理のモードを生成し、第2OSプログラムは、その連携処理モードを受信し、対応するローカルモードへの変換を行い、スレーブECUの制御モードを、変換したローカルモードに設定する。

【0078】

イベント管理サービスに関しては、第1OSプログラムは、連携処理に用いる連携イベントを生成するとともに、スレーブECUに配信する。そして、第2OSプログラムは、配信された連携イベントを、スレーブECUにおいて取り扱うことが可能なローカルイベントに変換する。スケジューラサービスに関しては、第1OSプログラムが連携処理のスケジュールを生成し、第2OSプログラムが、その連携処理スケジュールをスレーブECUにおけるローカルスケジュールに変換する。同期メカニズムサービスに関しては、第1OSプログラムが、同期手段として、所定のイベントによる同期指示を選択した場合、その所定のイベントによる同期を実現するための実現機能を第2OSプログラムが選択する。例えば、第2OSプログラムは、イベント管理サービスがローカルイベントを生成している場合には、そのローカルイベントを同期手段として選択する。

10

【0079】

クロックサービスに関しては、第1OSプログラムが連携時刻を生成し、第2OSプログラムが、その連携時刻をスレーブECUにおけるローカル時刻に変換する。セキュリティサービスに関しては、第1OSプログラムが、マスターECUの通信相手の認証と、受信データの正常性の確認を行い、第2OSプログラムが、スレーブECUの通信相手の認証と、受信データの正常性の確認を行う。診断サービスに関しては、第1OSプログラムが、マスターECUにおける連携処理の異常検知と、異常時のフェールセーフ処理を行い、第2OSプログラムが、スレーブECUにおける連携処理の異常検知と、異常時のフェールセーフ処理を行う。安全機能メカニズムサービスに関しては、第1OSプログラムが、マスターECUにおける連携機能の停止を司り、第2OSプログラムが、スレーブECUにおけるローカル機能の停止を司る。

20

【0080】

機能分散抽象化サービスに関しては、マスターECUの第1OSプログラムは、マスター-スレーブ間の機能分担の管理に関与せず、スレーブECUの第2OSプログラムが、その機能分担の管理を行う。同様に、支配抽象化サービスについても、マスターECUの第1OSプログラムは、責任分担管理に関与せず、スレーブECUの第2OSプログラムが、その責任分担の管理を行う。

30

【0081】

一方、図7は、マスターECUの第1OSプログラムの役割を重くし、スレーブECUにおける第2OSプログラムの役割を軽くした例を示している。

【0082】

この場合、ドメイン外内抽象化サービスについて、マスターECUの第1OSプログラムは、スレーブECUの第2OSプログラムによる各サービスの処理順、処理時間の設定を行い、その設定内容を示すデータを送信する。スレーブECUの第2OSプログラムは、設定内容を示すデータを受信し、その受信されたデータに従って、各種のサービスを実行する。また、意味解釈ゲートウェイサービスに関しては、マスターECUの第1OSプログラムが、データの意味翻訳を行うとともに、翻訳されたデータをスレーブECUに送信する。スレーブECUの第2OSプログラムは、翻訳されたデータを受信し、処理に使用する。

40

【0083】

リソース調停サービスに関しては、第1OSプログラムが、マスターECUによる連携処理のためのリソースを確保する処理、さらに、残りリソースの配分処理も行う。第2OSプログラムは、配分されたリソースを使用するように、リソース管理を行う。システムモードサービスに関しては、第1OSプログラムが連携処理のモードを生成し、さらに、その連携イベントを対応するローカルモードに変換して、スレーブECUに送信する。第2OSプログラムは、変換されたローカルモードを受信し、スレーブECUの制御モード

50

を、受信したローカルモードに設定する。

【0084】

イベント管理サービスに関しては、第1OSプログラムは、連携処理に用いる連携イベントを生成するとともに、その連携イベントを、スレーブECUにおいて取り扱うことが可能なローカルイベントに変換する。第2OSプログラムは、ローカルイベントを受信し、そのローカルイベントに基づく処理が行われるようにする。スケジューラサービスに関しては、第1OSプログラムが連携処理のスケジュールを生成し、さらに、その連携処理スケジュールをスレーブECUにおけるローカルスケジュールに変換して、送信する。第2OSプログラムは、ローカルスケジュールを受信し、そのローカルスケジュールに基づくスケジュール管理を行う。同期メカニズムサービスに関しては、第1OSプログラムが

10

【0085】

クロックサービスに関しては、第1OSプログラムが連携時刻を生成し、さらに、その連携時刻をスレーブECUにおけるローカル時刻に変換して送信する。第2OSプログラムは、ローカル時刻を受信し、処理に使用する。セキュリティサービスに関しては、第1OSプログラムが、マスターECU及びスレーブECUについて一括して管理する。例えば、第1OSプログラムが、マスターECU及びスレーブECUの通信相手の認証を行ったり、マスター-スレーブ間で送受信されるデータの正常性の確認を行ったりする。この場合、第2OSプログラムは、セキュリティサービスは扱わない。診断サービスに関しては、第1OSプログラムが、マスターECUにおける連携処理の異常検知と、異常時のフェールセーフ処理を行うとともに、異常検知結果をスレーブECUに通知する。第2OSプログラムは、その異常検知通知を受信する。安全機能メカニズムサービスに関しては、第1OSプログラムが、マスターECUにおける連携機能の停止を司るとともに、必要時にローカル機能の停止を指示する。第2OSプログラムは、第1OSプログラムからのローカル機能の停止指示に従い、スレーブECUにおけるローカル機能を停止させる。

20

【0086】

機能分散抽象化サービスに関しては、マスターECUの第1OSプログラムが、マスター-スレーブ間の機能分担の管理を行い、スレーブECUの第2OSプログラムは、その機能分担の管理に関与しない。同様に、支配抽象化サービスについても、マスターECUの第1OSプログラムが、責任分担管理を行い、スレーブECUの第2OSプログラムは、その責任分担の管理に関与しない。

30

【0087】

以上、第1OSプログラムと第2OSプログラムとの役割分担例を説明したが、この役割分担は、上述した2例以外に、マスターECUでの連携処理の処理応答と、スレーブECUでのローカル処理での処理応答との違い、マスターECU及びスレーブECUの能力など種々の要因によって、各サービス毎に役割分担が変化しえるため、様々なバリエーションが考えられる。

【0088】

再び、図4のフローチャートに戻って、説明を続ける。ステップS120までの処理により、第1OSプログラムと第2OSプログラムとが協調して連携処理をサポートできるように、第1OSプログラムが設定される。しかし、場合によっては、マスター側からスレーブ側に伝達される情報の種類を変更したり、マスター側における動作条件を修正したりといった微調整を行うことで、第1OSプログラムと第2OSプログラムとが協調した処理をより円滑に行いうる場合がある。

40

【0089】

そのため、図4のフローチャートのステップS130では、第2OSプログラムへの伝達情報の変更が必要であるか否かを判定する。そして、伝達情報の変更が必要と判定した場合には、ステップS140において、第2OSプログラムへの伝達情報の変更内容を通知する。さらに、ステップS150において、ステップS140での通知内容に応じた伝

50

達情報の変更を、第1OSプログラムに反映させる。

【0090】

マスターECUの第1OSプログラムが、第2OSプログラムへの伝達情報の変更が必要と判定するいくつかの例を、図8を参照して説明する。

【0091】

図8に示すように、伝達情報の変更が必要となる可能性がある対象サービスは、システムモードサービス、イベント管理サービス、スケジューラサービス、安全機能メカニズムサービスなど、いくつか考えられる。

【0092】

例えば、システムモードサービスの場合、スレーブECUにおけるアプリケーションソフトのバージョンアップなどによりスレーブECUが変更された場合、マスターECUの連携処理モードに紐づくローカルモードが変化することが予測される。この場合には、マスターECUの第1OSプログラムからローカルモードそのものの伝達に代えて、対応するモードの目的（起動、通常制御、停止など）を伝達するように変更する。これにより、スレーブECUの第2OSプログラムは、連携制御モードに対応する適切なモードを選択することが可能になる。

10

【0093】

また、イベント管理サービスの場合も、システムモードサービスの場合と同様に、スレーブECUが変更された場合、マスターECUの連携イベントに対応するローカルイベントが変化することが予測される。従って、この場合も、マスターECUの第1OSプログラムからローカルイベントそのものの伝達に代えて、対応するイベントの目的（割込、定期、異常検知など）を伝達するように変更する。これにより、スレーブECUの第2OSプログラムは、連携イベントに対応する適切なローカルイベントを生成することが可能になる。

20

【0094】

さらに、スケジューラサービスの場合も、システムモードサービスの場合と同様に、スレーブECUが変更された場合、マスターECUの連携スケジュールに対応するローカルスケジュールが変化することが予測される。従って、この場合も、マスターECUの第1OSプログラムからローカルスケジュールそのものの伝達に代えて、対応するスケジュールの目的（XXの後処理など）を伝達するように変更する。これにより、スレーブECUの第2OSプログラムは、連携スケジュールに対応する適切なローカルスケジュールを生成することが可能になる。

30

【0095】

また、安全機能メカニズムサービスの場合、マスター側がスレーブ側の停止指示を行う場合であって、スレーブ側が、マスター側とは独立した機能も備えている場合には、マスター側から停止指示を与えてしまうと、スレーブ側の状況を無視することになる可能性がある。そのため、マスターECUの第1OSプログラムは、スレーブの停止指示を目的で伝達する（制御終了、異常時停止など）。これにより、スレーブECUの第2OSプログラムは、自身の処理状況を踏まえ、その停止目的に応じた対処を取ることが可能になる。

40

【0096】

図4のフローチャートのステップS160では、スレーブECUの第2OSプログラムからの変更要求を受けたか否かを判定する。変更要求を受けたと判定した場合には、ステップS170に進み、スレーブECUの第2OSプログラムからの変更要求に応じて、第1OSプログラムの動作条件の修正を行う。

【0097】

マスターECUの第1OSプログラムが、スレーブECUの第2OSプログラムから動作条件の変更要求を受けるいくつかの例を、図9を参照して説明する。

【0098】

図9に示すように、動作条件の変更が必要となる可能性がある対象サービスは、ドメイン外内抽象化サービス、リソース調停サービス、同期メカニズムサービス、機能分散抽象

50

化サービス、支配抽象化サービス、安全機能メカニズムサービスなど、いくつか考えられる。

【0099】

例えば、マスターECUに対して複数のスレーブECUがあり、それら複数のスレーブECUによる制御の応答時間に差があり、スレーブ側の個別の調整では、複数のスレーブECUによる制御の応答時間の差に対応することができない場合など、スレーブ側の処理時間の整合が困難になることがありえる。このような場合、ドメイン内外抽象化サービスに関して、スレーブ側からマスター側に、データ通知タイミングの変更を要求する。例えば、マスターECUが、PTC22が実装された電子制御装置40であり、スレーブECUが、MGC23が実装された電子制御装置60及びエンジン制御部24が実装された電子制御装置80である場合を想定する。この場合、MGC23が実装された電子制御装置60のOSプログラム61b(第2OSプログラム)、もしくはエンジン制御部24が実装された電子制御装置80のOSプログラム81b(第2OSプログラム)は、PTC22が実装された電子制御装置40のOSプログラム41b(第1OSプログラム)に対して、エンジン制御部24が実装された電子制御装置80に対するよりも十分に遅くMGC23が実装された電子制御装置60に制御目標値(目標モータトルク)を通知するように要求する。これにより、エンジン31とモータジェネレータ33との応答時間の差を吸収することができ、エンジン31とモータジェネレータ33とにより、目標通りの駆動トルクを発生させることが可能となる。

10

【0100】

また、スレーブECUが制御責任を負う制御対象機器やセンサ機器がある場合、スレーブECUが扱うデータの中で、それら制御対象機器やセンサ機器の異常状態の監視のためのデータと、その他のデータとでは安全要求が異なる場合がある。そのような場合には、リソース調停サービスに関して、スレーブ側からマスター側に、例えば共用メモリにおいて、スレーブECUで使用するデータの安全別にメモリ領域を確保するように要求する。これにより、安全要求に見合ったデータの取り扱いを行うことが容易になる。

20

【0101】

また、同期メカニズムサービスに関して、スレーブ側で、同期を実現するための実現機能を選択することができない場合に、同期指示を出す条件をスレーブ側から指定しても良い。

30

【0102】

さらに、複数のスレーブがある場合に、マスター-スレーブ間の機能分担管理、責任分担管理を行うのに都合が良いのはマスター側であるため、そのような状況では、機能分散抽象化サービス及び支配抽象化サービスに関して、スレーブ側からマスター側に、機能分担管理、責任分担管理を行うよう要求しても良い。

【0103】

また、複数のスレーブがある場合、各スレーブでどのような手順で停止処理を行うのが適切であるか不明である状況では、安全機能サービスに関して、スレーブ側からマスター側に、各スレーブでのローカルの機能停止処理手順やその条件などを通知するよう要求しても良い。

40

【0104】

図5のフローチャートは、上述したように、スレーブ側の第2OSプログラムによって実行される処理を示しており、ステップS200では、マスターECUに対して、スレーブECUが保有する第2OSプログラムが分担する役割情報を提供する。

【0105】

続くステップS210では、マスターECUから、伝達情報の変更要求があったか否かを判定する。変更要求ありと判定した場合には、ステップS220に進み、マスターECUからの変更要求に応じて、第2OSプログラムが取得する情報を変更する。さらに、必要に応じて、変更後の情報の処理手順を定めておく。

【0106】

50

そして、ステップS230では、マスターECUの第1OSプログラムの動作条件の変更が必要であるか否かを判定する。動作条件の変更が必要であると判定した場合には、ステップS240において、マスターECUに対して、第1OSプログラムの動作条件の変更依頼を行う。そして、ステップS250において、依頼した第1OSプログラムの動作条件の変更内容を、第2OSプログラムに反映させるよう、第2プログラムを修正する。

【0107】

以上、本発明の好ましい実施形態について説明したが、本発明は、上述した実施形態になんら制限されることなく、本発明の主旨を逸脱しない範囲において、種々変形して実施することが可能である。

【0108】

例えば、上述した実施形態では、マスターECUに対して複数のスレーブECUがあり、スレーブ側の個別の調整では、スレーブ側の処理時間の整合が困難な場合、ドメイン内外抽象化サービスに関して、スレーブ側からマスター側に、データ通知タイミングの変更を要求した。しかしながら、スレーブECU同士が通信可能に構成されている場合には、スレーブECUの第2OSプログラム同士の通信によって、スレーブ側の処理時間の整合を図っても良い。例えば、一方のスレーブECUの第2OSプログラムによる処理手順や処理時間の情報に基づいて、他方のスレーブECUの第2OSプログラムにおける、処理手順や処理時間を相互に整合するように調整する。このように、スレーブECU同士が協調すれば、互いの処理の時間整合等も容易に図ることが可能になる。

【符号の説明】

【0109】

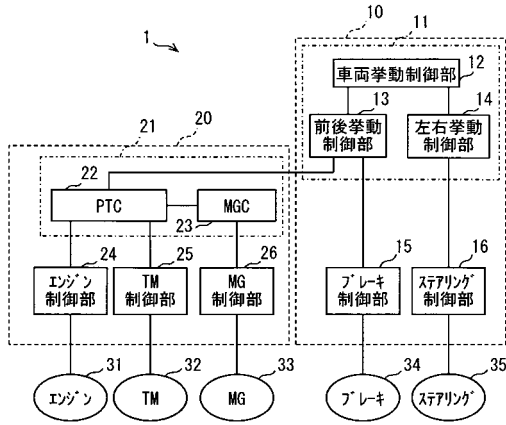
1 制御システム、11 運動ドメイン制御部、12 車両挙動制御部、13 前後挙動制御部、14 左右挙動制御部、15 ブレーキ制御部、16 ステアリング制御部、21 エネルギードメイン制御部、22 パワートレインコーディネータ、23 モータジェネレータコーディネータ、24 エンジン制御部、25 TM制御部、26 MG制御部、31 エンジン、32 トランスミッション、33 モータジェネレータ、34 ブレーキ装置、35 ステアリング装置、40 電子制御装置、41a PTCアプリケーション、41b OSプログラム、42 RTE、43 マイクロコントローラ抽象化層、44 ECU抽象化層、45 サービス層、46 複合ドライバ、47 マイクロコントローラ

10

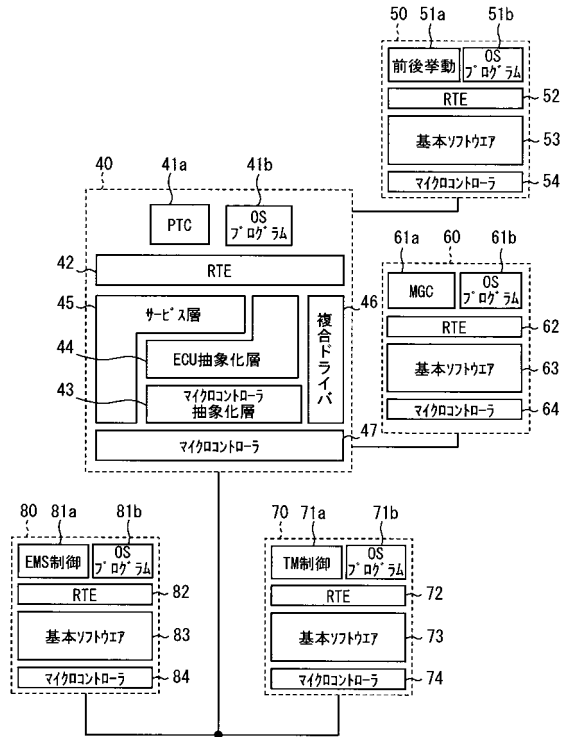
20

30

【図1】



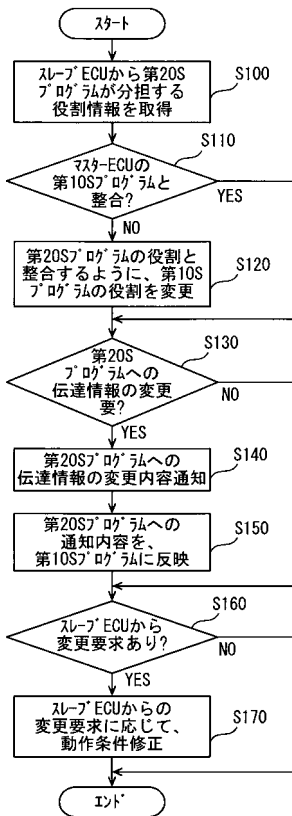
【図2】



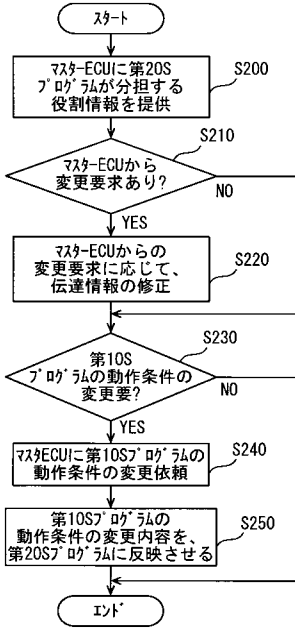
【図3】

階層	サービス	機能
Domain System OS	Domain External-Internal Abstraction	各サービスの処理順設定、時間整合
	Semantic Gateway	ドメイン内外翻訳
	Resource Arbitration	リソースの配分、調停
	System Mode	モード管理
	Event Manager	イベント生成と配信
	Scheduler	複数ECUでの連携スケジュール管理
	Synchronized Mechanism	イベントもしくは時間での制御同期
	Clock	共通の時刻生成とスキャ
	Security	なりすまし防止、データ保証
	Diagnostic	異常診断、フェールーフ
Domain System Abstraction	Safety Function Mechanism	異常時停止処理
	Function Distribution Abstraction	役割分担管理
	Ownership Abstraction	機器の責任分担管理

【図4】



【 図 5 】



【 図 6 】

階層	サビス	マスターECU	スレーブECU
DVFB	Domain External-Internal Abstraction	(スレーブECUでのサビスの処理順、処理時間には関与しない)	サビスの処理順、処理時間設定
	Semantic Gateway	データ送信	データの意味翻訳
	Resource Arbitration	リソース確保	残リソースの利用調整
Domain System OS	System Mode	連携モード生成	スレーブにおけるロカライズモードに変換
	Event Manager	連携イベント生成	ロカライズモードに変換
	Scheduler	連携スケジュール生成	ロカライズモードに変換
	Synchronized Mechanism	イベントによる同期指示	ロカライズモードでの実現機能を選択
	Clock	連携時刻生成	ロカライズモードでの時刻に変換
	Security	連携対象、データに対し実行	通信対象、データに対して実行
	Diagnostic	連携異常検知とフェールオーバー	ロカライズモードでの異常検知とフェールオーバー
	Safety Function Mechanism	連携機能停止	ロカライズモードでの機能停止
	Function Distribution Abstraction	(マスター側で管理しない)	マスター-スレーブ機能分担管理
Domain System Abstraction	Ownership Abstraction	(マスター側で管理しない)	マスター-スレーブ責任分担管理

【 図 7 】

階層	サビス	マスターECU	スレーブECU
DVFB	Domain External-Internal Abstraction	スレーブECUにおけるサビスの処理順、処理時間指示	データ受信、処理
	Semantic Gateway	データ意味翻訳、送信	データ受信、処理
Domain System OS	Resource Arbitration	リソース確保、残リソース配分	配分リソース使用
	System Mode	連携モードをロカライズモードに変換、送信	モード受信、処理
	Event Manager	連携イベントをロカライズモードに変換、送信	ロカライズモード受信、処理
	Scheduler	連携スケジュールをロカライズモードに変換、送信	ロカライズモード受信、処理
	Synchronized Mechanism	ロカライズモードでの同期手段設定、送信	ロカライズモードでの同期手段受信、使用
	Clock	連携時刻をロカライズモードに変換、送信	ロカライズモードでの時刻受信、使用
	Security	マスター、スレーブ一括	(スレーブではセキュリティ行わらず)
	Diagnostic	異常検知とフェールオーバー、スレーブ通知	異常通知受信
Domain System Abstraction	Safety Function Mechanism	連携機能停止、ロカライズモード指示	ロカライズモード停止指示受信、処理
	Function Distribution Abstraction	Sys OS M-S機能分担管理	(スレーブ側で管理しない)
	Ownership Abstraction	Sys OS M-S責任分担管理	(スレーブ側で管理しない)

【 図 8 】

分類	サビス	マスター側	スレーブ側	情報伝達
マスター → スレーブ	System Mode	連携モードをロカライズモードに変換、送信	モード受信、処理	(状況)スレーブの変更で、連携モードと紐づくモード変化。 (マスター → スレーブ) 情報伝達)モードを、目的で伝達
	Event Manager	連携イベントをロカライズモードに変換、送信	ロカライズモード受信、処理	(状況)スレーブの変更で、ロカライズモード変化。 (マスター → スレーブ) 情報伝達)イベントを、目的で伝達
	Scheduler	連携スケジュールをロカライズモードに変換、送信	ロカライズモード受信、処理	(状況)スレーブの変更で、ロカライズモード変化。 (マスター → スレーブ) 情報伝達)スケジュールを、目的で伝達。(e.g. XXの後処理)
	Safety Function Mechanism	連携機能停止、スレーブ停止指示	スレーブ停止指示受信、処理	(状況)停止処理時にスレーブ側の状況を無視する可能性。 (マスター → スレーブ) 情報伝達)停止を、目的で伝達。対応をスレーブ側に一任。

【 図 9 】

分類	サービス	マスタ側	スレーブ側	重要要求
スレーブ ⇒ マスタ	Domain Ex-Internal Abstraction	マスタ送信	マスタ受信、処理順、処理時間設定	(状況) スレーブ側の処理時間の整合困難 (スレーブ ⇒ マスタ-変更要求) マスタ通知がマスタが変更 (e.g. EMS, MGC同時通知で無く、MGCに十分遅く通知)
	Resource Arbitration	リソース確保	残リソースの利用調整	(状況) スレーブの責任分担に応じて安全要求異なる。 (スレーブ ⇒ マスタ-変更要求) スレーブで使用するマスタの安全別リソース領域確保。
	Synchronized Mechanism	イベントによる同期指示	ロ-カでの実現機能を選択	(状況) スレーブ側で対応可能な指示が欲しい。 (スレーブ ⇒ マスタ-変更要求) 同期指示を出す条件をスレーブ側から指定。
	Function Distribution Abstraction	(Master側で管理しない)	マスタ-スレーブ機能分担管理	(状況) スレーブが複数あり、分担管理に都合が良いのはマスタ側。 (スレーブ ⇒ マスタ-変更要求) マスタ側にて、機能分担管理
	Ownership Abstraction	(Master側で管理しない)	マスタ-スレーブ責任分担管理	↑
	Safety Function Mechanism	連携機能停止	ロ-カ機能停止	(状況) 複数のスレーブがある場合、どのような手順でロ-カ処理するのが適切か不明。 (スレーブ ⇒ マスタ-変更要求) ロ-カの処理手順/条件の通知要求。