

(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2010-149537
(P2010-149537A)

(43) 公開日 平成22年7月8日(2010.7.8)

(51) Int.Cl.
B6OR 16/02 (2006.01)

F I
B6OR 16/02 660B

テーマコード (参考)

審査請求 未請求 請求項の数 8 O L (全 21 頁)

(21) 出願番号 特願2008-326720 (P2008-326720)
(22) 出願日 平成20年12月23日 (2008.12.23)

(71) 出願人 395011665
株式会社オートネットワーク技術研究所
三重県四日市市西末広町1番14号
(71) 出願人 000183406
住友電装株式会社
三重県四日市市西末広町1番14号
(71) 出願人 000002130
住友電気工業株式会社
大阪府大阪市中央区北浜四丁目5番33号
(74) 代理人 100078868
弁理士 河野 登夫
(72) 発明者 木下 有里
三重県四日市市西末広町1番14号 株式会社オートネットワーク技術研究所内

最終頁に続く

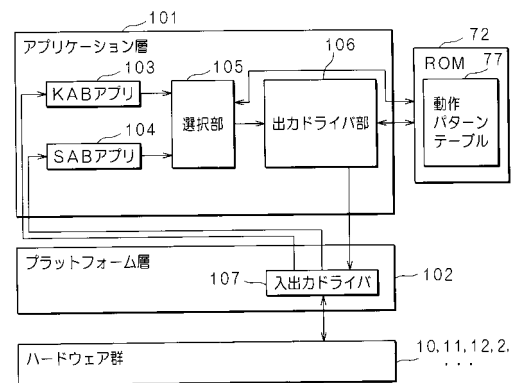
(54) 【発明の名称】 制御装置、制御方法及びコンピュータプログラム

(57) 【要約】

【課題】組み込み系のコンピュータプログラムにオブジェクト指向型の概念を適用し、メモリを節約して装置の簡素化を実現させ、開発効率を向上させることができる制御装置、制御方法及びコンピュータプログラムを提供する。

【解決手段】マイコンはアプリケーション層101における1つの選択部105の機能により、複数のアプリ(103, 104) 夫々からの動作要求を受け付け、動作要求に対応する制御対象の負荷と、該負荷に行なわせる動作パターンを選択し、選択した負荷及び動作パターンを示す情報を出力ドライバ部106の機能へ与える。マイコンは、1つの出力ドライバ部106の機能により与えられた情報が示す負荷へ、動作パターンに対応する制御信号を生成し、プラットフォーム層102における物理的な信号の入出力を実現する入出力ドライバ107から信号を出力させる。

【選択図】 図3



【特許請求の範囲】**【請求項 1】**

マイクロコンピュータを備え、該マイクロコンピュータが各機能に対応する複数のアプリケーションプログラムに基づく処理を行なうことにより複数の負荷の動作を制御する制御装置において、

前記複数のアプリケーションプログラムに基づく処理からの動作要求を受け付け、該動作要求及び要求元のアプリケーションプログラムに対応する機能に応じて制御対象の負荷、及び該負荷に行なわせる動作を選択する選択手段、及び、

該選択手段が選択した負荷、及び動作を示す情報が与えられた場合に、前記動作を実現させる制御信号を生成して前記負荷へ出力する出力手段

10

を備えることを特徴とする制御装置。

【請求項 2】

連続的オン、オフ、断続的オン、待機時間、又は断続的オンの場合の回数若しくはオン時間及びオフ時間を含む動作特定要素の内のいずれか一又は複数の組み合わせからなり、各々識別情報が付与された動作パターンのテーブルを記憶する手段を備え、

前記選択手段は、前記テーブルから前記動作要求に対応する動作パターンを抽出し、抽出した動作パターンに付与されている識別情報を前記出力手段に与え、

前記出力手段は、前記識別情報により動作を特定するようにしてあること

を特徴とする請求項 1 に記載の制御装置。

20

【請求項 3】

前記テーブルは制御対象の負荷毎に記憶されており、

前記選択手段は、前記動作要求に応じて選択した負荷と、該負荷のテーブルから抽出した動作パターンに付与されている識別情報とを前記出力手段に与えるようにしてあることを特徴とする請求項 2 に記載の制御装置。

【請求項 4】

前記テーブルは、前記動作特定要素毎に記憶されており、

前記選択手段は、前記動作要求に応じて選択した負荷に行なわせる動作を、前記動作特定要素毎のテーブルから夫々動作パターンを読み出して組み合わせるようにしてあり、

選択した負荷を示す情報と、組み合わせた動作パターン夫々に付与されている識別情報の組み合わせとを前記出力手段に与えるようにしてあること

30

を特徴とする請求項 2 に記載の制御装置。

【請求項 5】

前記選択手段は、前記動作要求に対応する動作パターンが前記テーブルに存在しない場合、前記動作パターンを前記テーブルに追加する手段を備えること

を特徴とする請求項 2 に記載の制御装置。

【請求項 6】

前記複数のアプリケーションプログラムには優先度が付与されており、

前記選択手段は、複数のアプリケーションプログラムに基づく処理から夫々複数の動作要求を受け付けた場合、前記複数のアプリケーションプログラムの内の優先度が高いアプリケーションプログラムからの動作要求に応じた負荷及び動作を選択するようにしてあること

40

を特徴とする請求項 1 乃至 5 に記載の制御装置。

【請求項 7】

マイクロコンピュータを用い、該マイクロコンピュータに各機能に対応する複数のアプリケーションプログラムに基づく処理を行なわせることにより複数の負荷の動作を制御する方法において、

前記マイクロコンピュータが前記複数のアプリケーションプログラムと異なるコンピュータプログラムに基づき、

前記アプリケーションプログラムに基づく処理からの動作要求を受け付け、

受け付けた動作要求及び要求元のアプリケーションプログラムに対応する機能に応じて

50

制御対象の負荷、及び該負荷に行なわせる動作を選択し、

選択した負荷及び動作を示す情報に基づき、前記動作を実現させる制御信号を生成して前記負荷へ出力する

ことを特徴とする制御方法。

【請求項 8】

制御対象である複数の負荷が接続されているコンピュータに、アプリケーションプログラムに基づく処理を行なわせて前記複数の負荷の動作を制御させるコンピュータプログラムにおいて、

コンピュータに、

前記アプリケーションプログラムに基づく処理からの動作要求及び要求元のアプリケーションプログラムに対応する機能に応じて制御対象の負荷、及び該負荷に行なわせる動作を選択する選択手段、及び、

該選択手段が選択した負荷、及び動作を示す情報が与えられた場合に、前記動作を実現させる制御信号を生成して前記負荷へ出力する出力手段

として機能させることを特徴とするコンピュータプログラム。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、コンピュータプログラムに基づき各種負荷の動作制御を行なう制御装置に関する。特に、コンピュータプログラムの構成を、メモリを節約させることを可能とする構成として装置の簡素化を実現させ、且つ各負荷用の多様な機能の増減に柔軟に対応させることができ、開発効率を向上させることができる制御装置、制御方法及びコンピュータプログラムに関する。

【背景技術】

【0002】

マイクロコンピュータ（以下、マイコンという）を利用し、マイコン内蔵ROMに書き込まれたコンピュータプログラムに基づいて制御を行なう制御装置の場合、コンピュータプログラムは所謂組み込み系の言語で記述される。組み込み系の言語で記述するコンピュータプログラムは、装置の簡素化のためにメモリなどリソースを節約し、必要最低限の処理を制御対象毎に且つ機能毎に実装する構成が基本的である。

【0003】

図11は、従来のマイコンにおけるコンピュータプログラムの構成の概略を示す概略図である。図11に示すコンピュータプログラムの構成は特に、制御対象（負荷A及びB）の動作を制御する制御装置にてマイコンにより実行されるコンピュータプログラムの例を示している。図11は、コンピュータプログラムにより実現される機能をブロックにて示している。図11の例のコンピュータプログラムによって負荷Aと負荷Bとを制御する機能が実現される場合、図11に示すように、当該コンピュータプログラムは例えば入力信号に対する判定処理、入力信号に基づく負荷に対する動作条件の判定処理、各負荷への信号出力の調停、動作毎の負荷への信号出力などの機能が、アプリケーションプログラムにより実現したい機能（ア）及び機能（イ）夫々に対応して、更に負荷A及びB毎に実現される。

【0004】

コンピュータプログラムの実装上も、夫々の機能に応じたプログラムコード（実行ファイル）が作成され、いずれのプログラムコードもメモリ上に予め記憶されており、CPUにより読み出される。したがって、図11に示したブロック夫々に応じてメモリが占有される。このとき、負荷に対する調停処理の内容は負荷A及びBによらず同じであって、且つ負荷A及び負荷Bにおける調停処理は同期的に行なう必要がない。それにも拘わらず、調停処理を実現する機能は負荷A及びBとで夫々存在する。また、マイコンでの処理では機能（ア）のアプリケーションにおける処理と、機能（イ）のアプリケーションにおける処理とは同期的に動作する必要がないにも拘わらず、同一の動作を実行する負荷Aの調停

10

20

30

40

50

部の処理、負荷 B の調停部の処理を実現するプログラムが夫々に含まれて重複している。また、負荷 A 及び B への動作毎の信号出力部もほぼ処理の内容は同一で出力される信号の波形が異なる程度であるはずにも拘わらず、機能 (ア) 及び (イ) のアプリケーションに夫々含まれており、同時に使用される可能性がないにもかかわらず重複して夫々プログラムとして存在する。

【 0 0 0 5 】

このような構成では、同時に使用される可能性がないプログラムコードが多数重複して存在し、メモリを占有する量が多くなる。

【 0 0 0 6 】

また車両用の制御装置の分野では、近年では車両の機能の多様化、複雑化に伴って、従来のように実現される機能と、機能に対応する制御対象とが 1 対 1 ではない。例えば、従来では、運転者が意図的に方向指示器ランプ (ウィンカー) を点滅させるなど、進行方向を指示する方向指示機能と、制御対象となる方向指示器ランプとがほぼ 1 対 1 で対応する構成であった。しかしながら、近年では、方向指示器ランプは、方向指示機能を実現させるための制御対象であると共に、車外にいるユーザへの情報通知機能を実現するための制御対象である場合がある。この場合、同一の方向指示器ランプの点滅制御であっても方向指示機能に基づく点滅制御と、情報通知機能に基づく点滅制御との夫々について 1 対 1 の対応で独立にプログラムコードが作成されて記憶される。図 1 1 の例では、負荷 A の動作 a 及び負荷 B の動作 b は、機能 (ア) のアプリケーションのみならず、機能 (イ) のアプリケーションにも対応している。

10

20

【 0 0 0 7 】

このように、複数の機能に 1 の制御対象が対応し、機能毎且つ制御対象毎の制御処理単位にプログラムコードがメモリを占有するのでメモリの使用が非効率的である。

【 0 0 0 8 】

マイコンを用いて負荷の動作を制御する制御装置では、モータなどの制御対象のように回転数等の情報が必要な負荷以外の制御対象に対しては大抵、オン、オフの切り換えによって制御する。制御信号を生成する処理などは結果的に同じ内容の処理を行ない、単に制御対象が異なるのみであっても、図 1 1 の説明図に示したように、負荷夫々に対して信号出力機能を実現するプログラムコードがメモリに記憶される。更に、制御対象が同一であっても、各機能を実現するアプリケーションプログラムが複数実行可能である場合、各アプリケーションプログラムに応じて信号出力機能を実現するプログラムコードがメモリに記憶される。

30

【 0 0 0 9 】

しかもこの場合、各負荷を同時的に制御するのではなく、シーケンシャルに各負荷へ制御信号を出力して各負荷で制御信号に応じた動作が行なわれ、各プログラムコードに占有されるメモリが同時的にアクセスされることもないにも拘わらず、マイコンに実行させるための各動作に対応するプログラムコードがいずれも、メモリを予め占有するので、メモリの使用効率が非効率的である。

【 0 0 1 0 】

これに対し、制御装置に求められる機能の多様化、複雑化が進むにつれ、オブジェクト指向型の言語によって記述されるプログラムをマイコンが読み出して実行する制御装置も提案されている (特許文献 1 など参照)。このように、処理が実行される都度にメモリが確保され、処理が終了した場合にはメモリが開放されて他に使用できるなど、メモリを効率的に利用することができる。しかしながらこの場合、プログラム実行中にメモリを動的に確保するなどの処理をマイコンに実現させる基本機能 (ライブラリ) が必要である。このようにオブジェクト指向に基づく設計では、基本機能を実現するためのライブラリプログラムコードがメモリを使用する上、動的なメモリ確保、動的なライブラリのリンクなどの処理が想定され、オブジェクト指向型のコンピュータプログラムの構成により各種機能が実現できるのはメモリの記憶容量に余裕があるからである。

40

【特許文献 1】特開 2 0 0 8 - 7 7 2 2 0 号公報

50

【発明の開示】

【発明が解決しようとする課題】

【0011】

しかしながら、負荷制御を行なうマイコンを用いた制御装置では、メモリの容量に制限があるからこそ組み込み系の言語で記述されるコンピュータプログラムを利用するのであって、メモリの記憶容量に余裕を持たせてもよい特定の制御装置以外では、オブジェクト指向型のソフトウェア構成は適していない。一方で、従来のコンピュータプログラムの構成のように、共有できる処理であっても制御対象の負荷毎、及び機能毎にプログラムコードがメモリを占有する構成では、非効率的である。共通する同様な処理は共有化して再利用が可能な構成とし、共有化された処理はそれに対応するプログラムコードのみでメモリを占有することが望ましい。

10

【0012】

本発明は斯かる事情に鑑みてなされたものであり、組み込み系のコンピュータプログラムにオブジェクト指向型の概念を適用し、メモリを節約して装置の簡素化を実現させ、開発効率を向上させることができる制御装置、制御方法及びコンピュータプログラムを提供することを目的とする。

【課題を解決するための手段】

【0013】

第1発明に係る制御装置は、マイクロコンピュータを備え、該マイクロコンピュータが各機能に対応する複数のアプリケーションプログラムに基づく処理を行なうことにより複数の負荷の動作を制御する制御装置において、前記複数のアプリケーションプログラムに基づく処理からの動作要求を受け付け、該動作要求及び要求元のアプリケーションプログラムに対応する機能に応じて制御対象の負荷、及び該負荷に行なわせる動作を選択する選択手段、及び、該選択手段が選択した負荷、及び動作を示す情報が与えられた場合に、前記動作を実現させる制御信号を生成して前記負荷へ出力する出力手段を備えることを特徴とする。

20

【0014】

第2発明に係る制御装置は、連続的オン、オフ、断続的オン、待機時間、又は断続的オンの場合の回数若しくはオン時間及びオフ時間を含む動作特定要素の内のいずれか一又は複数の組み合わせからなり、各々識別情報が付与された動作パターンのテーブルを記憶する手段を備え、前記選択手段は、前記テーブルから前記動作要求に対応する動作パターンを抽出し、抽出した動作パターンに付与されている識別情報を前記出力手段に与え、前記出力手段は、前記識別情報により動作を特定するようにしてあることを特徴とする。

30

【0015】

第3発明に係る制御装置は、前記テーブルは制御対象の負荷毎に記憶されており、前記選択手段は、前記動作要求に応じて選択した負荷と、該負荷のテーブルから抽出した動作パターンに付与されている識別情報とを前記出力手段に与えるようにしてあることを特徴とする。

【0016】

第4発明に係る制御装置は、前記テーブルは、前記動作特定要素毎に記憶されており、前記選択手段は、前記動作要求に応じて選択した負荷に行なわせる動作を、前記動作特定要素毎のテーブルから夫々動作パターンを読み出して組み合わせるようにしてあり、選択した負荷を示す情報と、組み合わせた動作パターン夫々に付与されている識別情報の組み合わせとを前記出力手段に与えるようにしてあることを特徴とする。

40

【0017】

第5発明に係る制御装置は、前記選択手段は、前記動作要求に対応する動作パターンが前記テーブルに存在しない場合、前記動作パターンを前記テーブルに追加する手段を備えることを特徴とする。

【0018】

第6発明に係る制御装置は、前記複数のアプリケーションプログラムには優先度が付与

50

されており、前記選択手段は、複数のアプリケーションプログラムに基づく処理から夫々複数の動作要求を受け付けた場合、前記複数のアプリケーションプログラムの内の優先度が高いアプリケーションプログラムからの動作要求に応じた負荷及び動作を選択するようにしてあることを特徴とする。

【0019】

第7発明に係る制御方法は、マイクロコンピュータを用い、該マイクロコンピュータに各機能に対応する複数のアプリケーションプログラムに基づく処理を行なわせることにより複数の負荷の動作を制御する方法において、前記マイクロコンピュータが前記複数のアプリケーションプログラムと異なるコンピュータプログラムに基づき、前記アプリケーションプログラムに基づく処理からの動作要求を受け付け、受け付けた動作要求及び要求元のアプリケーションプログラムに対応する機能に応じて制御対象の負荷、及び該負荷に行なわせる動作を選択し、選択した負荷及び動作を示す情報に基づき、前記動作を実現させる制御信号を生成して前記負荷へ出力することを特徴とする。

10

【0020】

第8発明に係るコンピュータプログラムは、制御対象である複数の負荷が接続されているコンピュータに、アプリケーションプログラムに基づく処理を行なわせて前記複数の負荷の動作を制御させるコンピュータプログラムにおいて、コンピュータに、前記アプリケーションプログラムに基づく処理からの動作要求及び要求元のアプリケーションプログラムに対応する機能に応じて制御対象の負荷、及び該負荷に行なわせる動作を選択する選択手段、及び、該選択手段が選択した負荷、及び動作を示す情報が与えられた場合に、前記動作を実現させる制御信号を生成して前記負荷へ出力する出力手段として機能させることを特徴とする。

20

【0021】

第1発明、第7発明及び第8発明では、マイコンはアプリケーションプログラムを実行するのみならず、アプリケーションプログラムに基づく処理によって発せられる動作要求を受け付け、受け付けた動作要求及び要求元のアプリケーションプログラムに対応する機能に基づいて対象となる負荷及び具体的な動作内容を選択する手段、及び、選択された負荷及び動作内容に基づいて、負荷に選択された動作を実行せしめる制御信号を生成して負荷へ出力する手段としても夫々機能する。当該機能は、アプリケーションプログラムと同様の、ドライバへの指示を具体的に実行する特定のアプリケーションとして実現される。

30

【0022】

このとき選択手段としての機能では、動作要求元のアプリケーションを特定するが、出力手段としての機能では、動作要求元のアプリケーションを特定することなしに、選択された負荷へ選択された動作内容に応じた制御信号を生成して出力する。出力手段の機能は、いずれの負荷に対しても共通して利用することが可能である。

【0023】

第2発明では、動作の内容を具体的に特定する動作パターンは、負荷毎にテーブル化されて記憶され、選択手段としての機能では、動作パターンテーブルからいずれかの動作パターンを選択して出力手段へ与えればよい。出力手段では、識別情報にて示される動作パターンを動作パターンテーブルから特定して当該動作パターンに対応させた制御信号を生成すればよく、選択手段にて動作を示す情報として具体的な動作特定要素を夫々与え、出力手段にて動作特定要素を夫々解釈する構成とするよりも、処理が単純化されて装置を簡素化させることができる。また、更に各機能で実現される具体的な動作を変更する場合は動作パターンテーブルの内容を更新すればよい点で、柔軟性に優れる。

40

【0024】

第3発明では、負荷の動作の内容を具体的に特定する動作パターンは、負荷毎にテーブル化されて記憶されている。これにより、負荷毎に特定の動作を行なうべく設計されている場合、動作パターンとして負荷毎に最小限の動作パターンを記憶しておけばよく、動作パターンテーブルが占めるメモリの容量は少なく済む。

【0025】

50

第4発明では、負荷の動作の内容を具体的に特定する動作パターンは、負荷の違いによらずに連続的オン、オフ（停止）、断続的オン、待機時間、又は断続的オンの場合の回数、若しくはオン時間とオフ時間にて記述可能である。連続的オン、オフ、断続的オンなどの基本的な動作内容、待機時間、回数、又はオン時間若しくはオフ時間夫々について、パラメータをテーブル化し、夫々のテーブルから動作内容と、待機時間又はオン時間及びオフ時間とを組み合わせることにより、動作パターンテーブルとして記憶しておくべき内容は少なくてもよく、テーブルが占有するメモリを節約することができる。出力手段としての機能は1つに共有化して再利用できる一方で、自由な組み合わせで多様な動作を実現させることができる。

【0026】

第5発明では、第2発明において、選択手段がアプリケーションプログラムから受け付けた動作要求に対応する動作パターンを動作パターンテーブルから抽出できない場合、新たに動作パターンが動作パターンテーブルに追加される。これにより、アプリケーションプログラムにおける仕様変更に対応的に動作パターンに対応させ、多様な動作を実現させることができる。

【0027】

第6発明では、動作要求が複数のアプリケーションプログラムから受け付けられた場合、選択手段にて各アプリケーションプログラムに対応する機能の優先度に基づき、いずれの動作要求を優先的に受け付けるかが判断されるから、出力手段としての機能では、動作要求元のアプリケーションプログラムを認識することなしに、選択手段としての機能にて優先された動作のみ実行させるべく制御信号を生成して出力すればよい。出力手段としての機能は1つに共有化して再利用できる一方で、アプリケーションプログラムに対応する機能の内の優先度が高い機能を優先的に実現させるべく適宜選択して制御信号を生成して出力させることができる。

【発明の効果】

【0028】

本発明による場合、組み込み系のコンピュータプログラムに基づき制御を行なう制御装置でも、各機能を実現させる動作の内容に対応する制御信号を生成して負荷へ出力する処理を各機能にとって共有化し、1つの出力手段の機能を各種の負荷の制御に利用する。これにより、制御信号を生成して出力する機能を実現させるプログラムコードが負荷毎に且つ機能毎に記憶されて実行される構成よりも、共通して利用できるプログラムコードを1つメモリに記憶して実行可能に構成しておくことにより各機能を実現可能であるからメモリの占有量は少なく済む。メモリを節約できるので装置を簡素化させることができる。更に、実現したい機能毎に、実際の負荷制御の出力手段を夫々実装せずとも、共有化された出力手段の機能を再利用すればよいので、開発時に負荷への出力を意識せず実現させたい機能に対応する動作内容を特定できる構成とする条件を守ればアプリケーションプログラムを開発のみで済み、開発効率を向上させることができる点、優れた効果を奏する。

【発明を実施するための最良の形態】

【0029】

以下、本発明をその実施の形態を示す図面に基づいて具体的に説明する。なお、以下に示す実施の形態では、車両に搭載されているヘッドライト、ドアなどのボディ系の各種制御対象を制御する車載ボディECUに本発明の制御装置を適用した車載制御システムを例に挙げて説明する。

【0030】

（実施の形態1）

図1は、実施の形態1における車載制御システムの構成を示す構成図である。車載制御システムは、車両の前部左右に搭載されるヘッドライト10, 10、方向指示器ランプ11, 11及びスモールランプ12, 12と、同様に車両の前部に搭載されるホーン2と、車室内に搭載されるブザー3と、車両のドア部に設置され、ドアの開閉を検知するドア開閉センサ4, 4及びドアロックを検知するドアロックセンサ5, 5と、車室内のキーを探

10

20

30

40

50

知するキーセンサ 6 と、夫々を制御するボディ ECU 7 とを含む。

【 0 0 3 1 】

ヘッドライト 1 0 , 1 0、方向指示器ランプ 1 1 , 1 1 及びスモールランプ 1 2 , 1 2 は夫々ボディ ECU 7 に接続されており、ボディ ECU 7 により夫々の点灯、消灯が制御される。

【 0 0 3 2 】

ホーン 2 及びブザー 3 はボディ ECU 7 に接続されており、ボディ ECU 7 により吹鳴が制御される。

【 0 0 3 3 】

ドア開閉センサ 4 , 4 及びドアロックセンサ 5 , 5 はボディ ECU 7 に接続されており、夫々ドア開閉状態を知らしめる信号又はドアロックの開閉状態を知らしめる信号をボディ ECU 7 へ出力する。

10

【 0 0 3 4 】

キーセンサ 6 はボディ ECU 7 に接続されており、キーセンサ 6 にて検知したキーの有無を示す信号を ECU 7 へ出力する。

【 0 0 3 5 】

ボディ ECU 7 は、ドア開閉センサ 4 , 4、ドアロックセンサ 5 , 5 又はキーセンサ 6 から出力される信号に基づき、例えばエンジンが停止しており、車室内にキーが存在しない状態で外部からドアロックがされたことをドアロックセンサ 5 , 5、キーセンサ 6 にて検知した場合、方向指示器ランプ 1 1 , 1 1、又はスモールランプ 1 2 , 1 2 に複数回点滅させるべく制御信号を出力するなどしてキーレスアンサーバック機能を実現する。また、ボディ ECU 7 は、ドア開閉センサ 4 , 4、ドアロックセンサ 5 , 5 又はキーセンサ 6 から出力される信号に基づき、ドアロックがされている状態で正規なキー以外でドアロックが解除されようとしていることをドアロックセンサ 5 , 5 で検知した場合、ホーン 2 及びブザー 3 を吹鳴させ、ヘッドライト 1 0 などを点灯させるべく制御信号を出力するなどしてセキュリティアンサーバック機能を実現する。

20

【 0 0 3 6 】

なお、実施の形態 1 では、キーレスアンサーバック機能、セキュリティアンサーバック機能に関する各構成部の処理のみ記載するが、ボディ ECU 7 が他の機能を実現するために他の制御対象となる負荷（アクチュエータ）を制御することが可能な構成でもよいことは勿論である。

30

【 0 0 3 7 】

このような各種機能をボディ ECU 7 にて実現するためのボディ ECU 7 内部の構成及び処理について説明する。ボディ ECU 7 は後述するように、マイコンを備えて、マイコン内臓の ROM（Read Only Memory）に記憶されているコンピュータプログラムに基づく処理を実行することにより各種機能を実現する。

【 0 0 3 8 】

図 2 は、実施の形態 1 における車載制御システムに含まれるボディ ECU 7 の内部構成を示すブロック図である。ボディ ECU 7 は、CPU 7 1、ROM 7 2、RAM（Random Access Memory）7 3 及び I/O 7 4 を含むマイコン 7 0 を備えて構成される。他に、ドア開閉センサ 4 , 4、ドアロックセンサ 5 , 5、キーセンサ 6 などのセンサと車載ネットワークを介して接続される場合は通信手段としてネットワークコントローラ、トランシーバをマイコン 7 0 内又は外に備える構成としてもよい。

40

【 0 0 3 9 】

CPU 7 1 は、ROM 7 2 に記憶されているコンピュータプログラムを RAM 7 3 に読み出して実行することにより、上述のようなキーレスアンサーバック機能又はセキュリティアンサーバック機能を実現する。

【 0 0 4 0 】

ROM 7 2 には、フラッシュメモリを利用する。ROM 7 2 には、CPU 7 1 が読み出して実行するコンピュータプログラムとして、アプリケーションプログラム 7 5 , 7 5 ,

50

...、アプリケーションドライバプログラム76、及びドライバプログラム78が記憶されている。またROM72には、CPU71が各制御対象の動作内容を特定するために参照する動作パターンテーブル77が記憶されている。なお、ROM72は基本的に、ボディECU7の稼働中は読み出し専用ではあるが、CPU71に実行させたいコンピュータプログラムの更新、又は動作パターンテーブル77の更新が必要な場合に書換が可能である。

【0041】

RAM73には、DRAM (Dynamic RAM)、SRAM (Static RAM)等を利用する。RAM73にはCPU71が読み出すコンピュータプログラムがロードされる他、処理によって発生する各種情報が一時的に記憶される。

10

【0042】

I/O74は、マイコン70と外部とのインタフェースである。マイコン71はI/O74により、ヘッドライト10、10、方向指示器ランプ11、11又はスモールランプ12、12、ホーン2若しくはブザー3へ制御信号を出力する。

【0043】

次に、マイコン70のCPU71がROM72に記憶されているコンピュータプログラムを読み出して実現するマイコン70内の詳細な機能について説明する。図3は、実施の形態1におけるボディECU7のマイコン70のCPU71にて実現される機能を概念的に示す機能ブロック図である。

【0044】

CPU71は、ROM72に記憶されているアプリケーションプログラム75、75、アプリケーションドライバプログラム76、及びドライバプログラム78を読み出して実行することにより、アプリケーション層101、プラットフォーム層102の階層に分けられたソフトウェア構造にて動作する。

20

【0045】

CPU71がアプリケーションプログラム75、75、...の内、キーレスアンサーバック用のアプリケーションプログラム75に基づき、キーレスアンサーバック機能を実現するアプリケーション、即ちKAB (Keyless Answer Back) アプリ103として機能する。同様にCPU71は、セキュリティアンサーバック用のアプリケーションプログラム75に基づき、SAB (Security Answer Back) アプリ104として機能する。KABアプリ103及びSABアプリ104としての機能は構造的に、アプリケーション層101に位置する。

30

【0046】

CPU71はアプリケーションドライバプログラム76に基づき、選択部105及び出力ドライバ部106としての機能を実現する。選択部105及び出力ドライバ部106としての機能は、ハードウェア群(制御対象群、即ちヘッドライト10、方向指示器ランプ11、スモールランプ12、ホーン2、ブザー3等)を直接的に制御すべく物理層の信号を扱う所謂ドライバの機能ではなく、KABアプリ103又はSABアプリ104などの各種アプリケーションの処理の内の信号生成処理などの共通部分を1つにしたモジュールであり、アプリケーション層101に位置する。

40

【0047】

そしてCPU71はドライバプログラム78に基づき、入出力ドライバ107として機能する。入出力ドライバ107としての機能は所謂ドライバの機能を実現し、ハードウェア群を直接的に制御すべくI/O74へ実際に物理層の制御信号の入出力を行なう。入出力ドライバ107としての機能は構造的に、プラットフォーム層102に位置する。

【0048】

CPU71の選択部105としての機能は、CPU71がKABアプリ103又はSABアプリ104としての機能により制御対象となるハードウェア、例えばヘッドライト10へ点灯要求などの動作要求を発した場合に、動作要求を受け付け、制御対象と、制御対象に行なわせる具体的な動作とを特定し、特定した制御対象及び動作の内容を示す情報を

50

出力ドライバ部 106 へ与える。

【0049】

CPU71の出力ドライバ部106としての機能は、選択部105から与えられる制御対象、及び制御対象に行なわせる具体的な動作の情報に基づき、動作を実現させるための制御信号を生成して、制御対象のハードウェアへ制御信号を出力させるべくプラットフォーム層102の入出力ドライバ107へ指示する。

【0050】

このように、KABアプリ103及びSABアプリ104など各アプリケーション毎、制御対象毎に、制御信号を生成して入出力ドライバ107へ夫々出力する構成(図11参照)とするよりも、本実施の形態1のボディECU7では、各アプリケーションからの動作要求を選択部105にて一括して受け付け、制御信号の生成を行なう出力ドライバ部106の機能を共有化する構成とする。KABアプリ103及びSABアプリ104などのいずれのアプリケーションでも、制御がオン、オフのスイッチングにて実現する場合、動作要求に応じて制御信号を生成する処理は、動作の具体的内容が異なるのみで共通した処理であるから共有化が可能である。また、ボディECU7では、各制御対象のハードウェアの制御を同時に行なうことはなく、各アプリケーションの機能も順に実現されるから制御信号の生成処理が同時に行なえる構成とする必要はないので共有化が可能である。当該処理を1つにして各機能から再利用可能に共有化した構成とすることにより、ROM72及びRAM73におけるメモリ占有量を削減できる。

【0051】

具体的には、CPU71にて各機能が実現される場合、図3の各ブロックにて示す各機能に対応する実行プログラムファイル(プログラムコード)が必要となる。制御対象の決定及び制御信号の生成処理の機能を1つの選択部105及び出力ドライバ部106に共有化して、各種機能から利用される構成とすることにより、複数のアプリケーションプログラム75, 75, ... 夫々に対応させたアプリケーションドライバプログラムを複数実行する構成よりもROM72及びRAM73における記憶容量が少なくて済む。KABアプリ103及びSABアプリ104夫々のために出力ドライバ部106に対応するプログラムコードを用意する必要がないからである。また、実現したい各機能について実際に制御信号を生成する処理を実装せずとも、選択部105及び出力ドライバ部106の機能を再利用すればよいので、機能の開発は動作要求を発するアプリケーションプログラム75, 75, ... の開発のみで済み、開発効率を向上させることができる。

【0052】

そして実施の形態1においてCPU71は選択部105の機能にて動作の内容を特定するに際し、ROM72に予め記憶させてある動作パターンテーブル77を参照して特定する。動作パターンテーブル77には、実施の形態1では制御対象となるハードウェア毎に想定される複数の動作パターンが記憶されており、選択部105は動作パターンテーブル77から動作パターンを特定する識別情報を出力ドライバ部106へ与え、出力ドライバ部106では与えられた識別情報から特定される動作パターンに基づき当該動作パターンに対応する制御信号が入出力ドライバ107から出力されるように入出力ドライバ107へ指示する。

【0053】

図4は、実施の形態1におけるボディECU7のROM72に記憶されている動作パターンテーブル77の内容例を示す説明図である。実施の形態1において動作パターンテーブル77は、制御対象毎に動作パターンが記憶されているが、図4に示す内容例では、制御対象としてスモールランプ12, 12の動作パターンの例を挙げている。

【0054】

動作パターンは、動作をオンとするまでの待機時間、連続的にオンとするか、断続的にオンとするか、又は1回のみオンとするかなどの動作方式、動作方式が断続的オンである場合のオンとする回数、断続的動作である場合のオン時間とオフ時間とを含む動作特定要素の組み合わせにて特定される。図4の内容例では、スモールランプ12, 12の動作パ

10

20

30

40

50

ターンは、オン即ち点灯させるまでの待機時間、連続的 point 灯か又は断続的 point 灯即ち点滅かの動作方式、動作方式が点滅の場合の点滅の回数、動作方式が点滅の場合の点灯時間と消灯時間とを含む。このような動作特定要素により、スモールランプ 1 2 , 1 2 に行なわせる動作を特定する。

【 0 0 5 5 】

そして、動作パターンテーブル 7 7 では、各動作パターンに識別情報として番号が付与されている。図 4 の内容例において、「 1 」の動作パターンは、スモールランプ 1 2 , 1 2 を、待機することなく「 1 0 0 0 m s (ミリ秒) 」の間、点灯させることで「 1 回」「点滅」させるパターンであり、「 2 」の動作パターンは、スモールランプ 1 2 , 1 2 を、待機することなく「 7 5 0 m s 」点灯させ、「 7 5 0 m s 」消灯させることを繰り返して「 2 回」「点滅」させるパターンである。「 3 」の動作パターンは、「 1 0 m s 」待機した後、オフが指示されるまで連続的に「点灯」させるパターンである。

10

【 0 0 5 6 】

選択部 1 0 5 としての機能により CPU 7 1 は、例えば K A B アプリ 1 0 3 の機能から「ロックしました」の通知に対応するスモールランプ 1 2 , 1 2 の 1 回点滅要求を受け付けた場合、スモールランプ 1 2 , 1 2 の動作パターンとして図 4 の内容例において動作パターン「 1 」を選択する。そして選択部 1 0 5 としての機能により CPU 7 1 は、スモールランプ 1 2 , 1 2 を示す情報、例えば制御対象の番号と、選択した動作パターンの番号(「 1 」)とを出力ドライバ部 1 0 6 へ与える。このとき選択部 1 0 5 としての機能では、「ロックしました」の通知要求のみ受け付け、「ロックしました」の通知要求に対応する動作として「 1 」の動作パターンを特定して出力する構成としてもよい。

20

【 0 0 5 7 】

そしてこの場合、出力ドライバ部 1 0 6 としての機能により CPU 7 1 は、選択部 1 0 5 から与えられたスモールランプ 1 2 , 1 2 を示す情報と、動作パターンの識別情報、この場合「 1 」とから動作パターンテーブル 7 7 を参照し、スモールランプ 1 2 , 1 2 を待機することなく「 1 0 0 0 m s (ミリ秒) 」の間のみ点灯させて「 1 回」「点滅」させるパターンであることを特定し、当該パターンに対応する制御信号が入出力ドライバ 1 0 7 から出力されるように入出力ドライバ 1 0 7 へ指示する。

【 0 0 5 8 】

このように、選択部 1 0 5 及び出力ドライバ部 1 0 6 としての機能により CPU 7 1 は、動作パターンテーブル 7 7 を参照して動作の内容を特定することにより、処理が単純化されて装置を簡素化させることができる。また具体的な動作を変更する場合は動作パターンテーブル 7 7 の内容を更新すればよく、多様な動作を柔軟に実現させることができる。

30

【 0 0 5 9 】

次に、出力ドライバ部 1 0 6 にて実現される CPU 7 1 の処理の詳細を説明する。図 5 は、CPU 7 1 が出力ドライバ部 1 0 6 として機能する際の処理の具体例を概念的に示す説明図である。図 5 には、CPU 7 1 が出力ドライバ部 1 0 6 として機能している際に R A M 7 3 を占有する各種情報をブロックにて示している。

【 0 0 6 0 】

図 5 中の上段には、ボディ E C U 7 の制御対象であるヘッドライト 1 0 , 1 0 、方向指示器ランプ 1 1 , 1 1 、スモールランプ 1 2 , 1 2 、ホーン 2 、ブザー 3 夫々について、動作状態、動作回数、各動作の経過時間が制御対象毎の固有パラメータが記憶されている。なお、同一の制御対象であっても、該制御対象に動作を要求するアプリケーションプログラム毎に動作状態を記憶してもよい。動作状態の内容は、停止中、オン(点灯)、オフ(消灯)などのいずれかであり、回数とは断続的動作の場合の回数、例えば現在の点滅回数を示す。なお、斜線にて示す動作状態、動作回数及び経過時間の内容は随時書き換えられる。

40

【 0 0 6 1 】

図 5 中の下段には、動作パターンテーブル 7 7 が示されている。動作パターンテーブル 7 7 は R O M 7 2 に記憶されている。図 5 に示す R A M 7 3 上の情報は、CPU 7 1 によ

50

って一時的に抽出された動作パターンテーブル 77 の一部である。

【 0 0 6 2 】

図 5 中の中段には、出力ドライバ部 106 に対応するプログラムコードがロードされていることが示されている。プログラムコードにおいて選択部 105 から与えられた制御対象を示す情報と動作パターンを特定する識別情報とにより、制御信号の出力先を特定し、当該出力先の固有パラメータの動作状態及び動作回数と、特定された動作パターンの動作特定要素を比較して、動作状態に応じた信号を生成し、動作状態及び動作回数を書き換えて処理を終了し、次の他の制御対象に対する動作要求に対応する処理を実行する。

【 0 0 6 3 】

このとき、複数の制御対象毎、又は複数のアプリケーションプログラム 75, 75 毎に、複数の出力ドライバ部 106 のプログラムコードが展開されて夫々対応付けられる構成ではなく、1つのプログラムコードで複数の制御対象への制御信号生成処理が実現していることに注目すべきである。

【 0 0 6 4 】

図 6 は、オブジェクト指向型における各制御対象毎の制御処理用のプログラムの実現例を比較例として示す説明図である。図 6 中の左側に、出力ドライバ部 106 の処理に対応する共通の制御信号生成処理を持つ制御クラスの定義を示している。制御クラスは、動作状態、動作回数及び経過時間を固有パラメータとして有し、制御信号生成処理では、これらの固有パラメータを内部で可変情報として処理を実行する。当該制御クラスを、制御対象毎に実体（インスタンス）化することにより、右側に示すように各制御対象について処理を実際実現するプログラムコードが生成される。図 6 の例では、ヘッドライト制御、スモールランプ制御、及びホーン制御について実体化されている。この場合、制御信号生成処理の内容はいずれも共通するものの、いずれも実体化されるから各インスタンスについてメモリが確保され、プログラムコード部分は図 5 に示した場合よりも約 3 倍のメモリを占有する。

【 0 0 6 5 】

このように、1つのプログラムコードで各制御対象毎の固有パラメータを動作の進行具合に応じて書き換え、各アプリケーションからの動作要求毎に順に実行することにより、オブジェクト指向型の概念において1つの共通する制御処理を持つクラスを各制御対象毎に実体化して、クラスが持つ共通処理を各制御対象の固有パラメータに対して実行する処理を、メモリ容量に制限がある組み込み系のコンピュータプログラムでも実現することができる。

【 0 0 6 6 】

図 7 は、CPU 71 が出力ドライバ部 106 として機能する際の処理手順の具体例を示すフローチャートである。CPU 71 は、動作要求を受け付けた選択部 105 から制御対象を示す情報と、制御対象に行なわせるべき動作の動作パターンの情報とを与えられた場合、図 5 に示したようにメモリに展開された1つのプログラムコードに基づき以下の処理を実行する。なお、CPU 71 は、以下に示すフローチャートの処理を終了した後、所定時間待った後に再度処理を開始して繰り返し、適宜その時点での動作状態及び経過時間等に基づいて処理を進める。なお、以下に示す処理手順では、制御対象としてスモールランプ 12, 12 を例とする。

【 0 0 6 7 】

CPU 71 は、与えられた制御対象の情報、ここではスモールランプ 12, 12 の情報に基づき、スモールランプ 12, 12 の固有パラメータが記載されているメモリを参照し（ステップ S1）、固有パラメータの内の動作状態を読み出す（ステップ S2）。

【 0 0 6 8 】

CPU 71 は、ステップ S2 にて読み出した動作状態が、停止中、オン（点灯）状態、オフ（消灯）状態のいずれであるかを判別し（ステップ S3）、動作状態が停止中状態である場合は処理をステップ S4 へ、動作状態がオン状態である場合は処理をステップ S7 へ、動作状態がオフ状態である場合は処理をステップ S13 へ進める。

10

20

30

40

50

【 0 0 6 9 】

ステップ S 3 にて、CPU 7 1 は、スモールランプ 1 2 , 1 2 への現在の動作要求に対応する動作状態の内容が停止中状態であると判断された場合 (S 3 : 停止中)、連続的オン又は断続的オンが要求されているか否かを判断する (ステップ S 4)。CPU 7 1 は、連続的オン即ちここではスモールランプ 1 2 , 1 2 の点灯、又は、断続的オン、即ちここではスモールランプ 1 2 , 1 2 の点滅が要求されていると判断した場合 (S 4 : Y E S)、制御対象の動作状態をオン状態、即ちスモールランプ 1 2 , 1 2 の動作状態をオン (点灯) 状態とし (ステップ S 5)、実際にスモールランプ 1 2 , 1 2 へオン信号を出力すべく入出力ドライバ 1 0 7 へ指示し (ステップ S 6)、処理を終了する。

【 0 0 7 0 】

なお、ステップ S 4 にて CPU 7 1 は、連続的オン及び断続的オンのいずれの動作も要求されていないと判断した場合 (S 4 : N O)、停止中状態を維持するため処理をそのまま終了する。

【 0 0 7 1 】

ステップ S 3 にて、CPU 7 1 は、スモールランプ 1 2 , 1 2 への現在の動作状態の内容がオン (点灯) であると判断した場合 (S 3 : オン)、動作要求に対応する制御対象の固有パラメータの内の、現在のオンとなつてからの経過時間、即ちここではスモールランプ 1 2 , 1 2 の点滅又は点灯要求に対応するオンとなつてからの経過時間を参照し、経過時間が、与えられている動作パターンの動作特定要素の内のオン時間を経過しているか否かを判断する (ステップ S 7)。

【 0 0 7 2 】

CPU 7 1 はステップ S 7 にて、オン時間が経過していると判断した場合 (S 7 : Y E S)、与えられている動作パターンの動作特定要素の動作方式が断続的オンのときは動作回数、即ちここでは点滅回数を加算し (ステップ S 8)、実際にスモールランプ 1 2 , 1 2 へオフ信号を出力すべく入出力ドライバ 1 0 7 へ指示する (ステップ S 9)。

【 0 0 7 3 】

次に CPU 7 1 は、ステップ S 8 で加算した後の動作回数は、即ちここでは点滅回数は、与えられている動作パターンの動作特定要素の内の動作回数を超えているか否かを判断する (ステップ S 1 0)。CPU 7 1 は、ステップ S 1 0 にて動作回数を超えていないと判断した場合 (S 1 0 : N O)、断続的オン、即ち点滅動作は継続するので制御対象の動作状態の内容をオフ状態、即ちここではスモールランプ 1 2 , 1 2 の動作状態を消灯状態へ遷移させ (ステップ S 1 1)、処理を終了する。CPU 7 1 は、ステップ S 1 0 にて動作回数を超えていると判断した場合 (S 1 0 : Y E S)、断続的オン、即ち点滅動作は終了するので動作状態の内容を停止状態へ遷移させ (ステップ S 1 2)、処理を終了する。

【 0 0 7 4 】

CPU 7 1 はステップ S 7 にて、オン時間が経過していないと判断した場合 (S 7 : N O)、オン状態即ち点灯状態を維持するため処理をそのまま終了する。

【 0 0 7 5 】

ステップ S 3 にて、CPU 7 1 は、スモールランプ 1 2 , 1 2 への現在の動作状態の内容がオフ (消灯) であると判断した場合 (S 3 : オフ)、動作要求に対応する制御対象の固有パラメータの内の、現在のオフとなつてからの経過時間、即ちここではスモールランプ 1 2 , 1 2 の点滅又は点灯要求に対応するオフとなつてからの経過時間を参照し、経過時間が、与えられている動作パターンの動作特定要素の内のオフ時間を経過しているか否かを判断する (ステップ S 1 3)。

【 0 0 7 6 】

CPU 7 1 はステップ S 1 3 にて、オフ時間が経過していると判断した場合 (S 1 3 : Y E S)、制御対象の動作状態をオン状態、即ちスモールランプ 1 2 , 1 2 の動作状態をオン (点灯) 状態とし (ステップ S 1 4)、実際にスモールランプ 1 2 , 1 2 へオン信号を出力すべく入出力ドライバ 1 0 7 へ指示し (ステップ S 1 5)、処理を終了する。

【 0 0 7 7 】

10

20

30

40

50

なお、ステップ S 1 3 にて CPU 7 1 は、オフ時間が経過していないと判断した場合（S 1 3 : NO）、オフ状態を維持するため処理をそのまま終了する。

【 0 0 7 8 】

上述のような処理により、各制御対象の固有パラメータの内の動作状態と、動作回数、及び経過時間の内容を変更することにより、各制御対象及び該制御対象夫々に対して動作要求を発するアプリケーションプログラム毎にオブジェクトが実体化されて処理が行なわれているようなオブジェクト指向的な処理が実現できる。なお、車載 ECU などでは、制御対象毎に巡回的に高速に処理を実行すればよい点が利用され、同一のプログラムコードを用いている。更に、上述のように 1 つのアプリケーションドライバプログラム 7 6 のプログラムコードを複数の制御対象及び複数の機能を実現するアプリケーションプログラム 7 5 , 7 5 , ... に基づく制御に対応させて用いることにより、処理の実行中に RAM 7 3 に対しプログラムコードが占有する容量が少なく済み、また、動的に容量が確保されることもなく煩雑なメモリ管理も要しないので、マイコンに基づく ECU における処理に適している。

10

【 0 0 7 9 】

このように ROM 7 2 及び RAM 7 3 の記憶容量を節約できるのでボディ ECU の内部構成を簡素化させることができる。更に、ボディ ECU にて実現したい機能毎に、アプリケーションドライバプログラム 7 6 に対応するプログラムを実装せずとも、共有化されたアプリケーションドライバプログラム 7 6 の機能を共有化するから、開発時は選択部 1 0 5 としての機能に対するインタフェースを考慮すればアプリケーションプログラム 7 5 , 7 5 , ... の開発のみで済み、開発効率を向上させることができる

20

【 0 0 8 0 】

（実施の形態 2）

実施の形態 1 では、図 4 に示したように、動作パターンテーブル 7 7 の具体的内容は、制御対象毎の動作パターンを一覧にした構成とした。これに対し、実施の形態 2 における動作パターンテーブル 7 7 の内容は、動作パターンを特定する動作特定用素毎の内容を特定する形式であり、いずれの制御対象でも単一の動作パターンテーブル 7 7 を利用する構成とする。

【 0 0 8 1 】

実施の形態 2 における車載制御システムの構成は、動作パターンテーブル 7 7 の内容以外は、実施の形態 1 における構成と共通である。したがって、以下の説明では、実施の形態 1 の構成と共通する構成について実施の形態 1 と同一の符号を用い、相違点である動作パターンテーブル 7 7 の具体的内容、及びそれを用いた動作パターンの特定の方法について説明する。

30

【 0 0 8 2 】

図 8 は、実施の形態 2 におけるボディ ECU の ROM 7 2 に記憶されている動作パターンテーブル 7 7 の内容例を示す説明図である。図 8 に示すように、実施の形態 2 における動作パターンテーブル 7 7 は、時間テーブル、動作方式テーブル及び回数テーブルのように動作特定要素毎にテーブル化されており、いずれからも番号で内容を選択して動作特定要素を特定することができる。

40

【 0 0 8 3 】

図 8 に示す例では、時間テーブルには、待機時間又はオン時間、オフ時間の内容を特定する候補として「0ms」、「5ms」などが含まれており、夫々識別情報として番号で選択が可能である。動作方式テーブルには動作方式として連続的オン、断続的オンを含み、番号で選択できる。なお、スモールランプ 1 2 , 1 2 に対して動作方式として連続的オンを選択した場合、具体的な動作は点灯であるし、断続的オンを選択した場合、具体的な動作は点滅である。更に、回数テーブルには回数を特定する候補として「0」、「1」、「2」などが含まれており、これも番号で選択が可能である。

【 0 0 8 4 】

例えば、図 4 の説明図に示した番号「1」の動作パターンは、図 8 に示した動作パター

50

ンテーブル 77 の内容を用いれば、待機時間は「1」、動作方式は「2」、回数は「2」、オン時間及びオフ時間は「5」及び「1」として、(待機時間、動作方式、回数、オン時間、オフ時間) = (1, 2, 2, 5, 1) のようにパラメータとして動作パターンを特定して出力ドライバ部 106 へ与えることができる。

【0085】

このような動作パターンテーブル 77 の内容とすることにより、ROM 72 を更に節約することができ、効果的である。また、実施の形態 1 のような負荷毎に動作パターンを予め記憶しておく場合には、新たなパターンが増えた場合には動作パターンテーブル 77 に追加しなければならないが、実施の形態 2 のような構成とすることにより、各制御対象について選択部 105 の機能により自由な動作パターンを選択することが可能となる。したがって、多様な動作を実現できる点でも効果的である。

10

【0086】

(変形例)

上述のようなボディ ECU 7 において、マイコン 70 の CPU 71 は選択部 105 としての機能により、次のような処理を行なう構成としてもよい。つまり、KAB アプリ 103 若しくは SAB アプリ 104、又は新たなアプリケーションプログラム 75 に基づくアプリにより、新たな動作要求がされた場合、選択部 105 は動作パターンテーブル 77 に対応する動作パターンが無いことを確認して新たに動作パターンを追加する。

【0087】

図 9 は、選択部 105 としての機能により CPU 71 が動作要求に対応する動作パターンを新たに動作パターンテーブル 77 に追加する処理手順の一例を示すフローチャートである。

20

【0088】

CPU 71 は、選択部 105 としての機能により、KAB アプリ 103 又は SAB アプリ 104 を含む各アプリケーションプログラム 75, 75, ... に基づく各アプリからの動作要求を受け付ける (ステップ S21)。CPU 71 は、動作パターンテーブル 77 に、受け付けた動作要求に対応する動作パターンが有るか否かを判断する (ステップ S22)。

【0089】

CPU 71 は、動作要求に対応する動作パターンがあると判断した場合 (S22: YES)、対応する動作パターンを選択して (ステップ S23)、出力ドライバ部 106 の機能へ動作パターンを特定する情報を与え (ステップ S24)、処理を終了する。

30

【0090】

CPU 71 は、ステップ S22 にて動作要求に対応する動作パターンが無いと判断した場合 (S22: NO)、受け付けた動作要求に対応する動作パターンを特定する (ステップ S25)。具体的には、例えば動作要求が「2 回点滅要求」などを識別可能な命令「blink (回数パラメータ)」などのコマンドであれば、動作パターンテーブル 77 の内容が制御対象毎であるときには制御対象に対する「2」回「点滅」する動作パターンの有無を判断すればよい。また、動作パターンテーブル 77 の内容が、動作特定要素毎であって、例えば回数テーブルに「2」に対応する候補の有無で判断すればよい。

40

【0091】

CPU 71 は、ステップ S25 で特定した動作パターンを新たに作成して追加し (ステップ S26)、出力ドライバ部 106 の機能へ動作パターンを特定する情報を与え (S24)、処理を終了する。

【0092】

ステップ S26 において、新たに動作パターンを追加する場合、動作パターンテーブル 77 が制御対象毎の動作パターンである場合は特定した動作パターンの動作特定要素を夫々設定して追加し新たな番号を付与する。動作パターンテーブル 77 が動作特定要素毎のテーブルである場合、新たな候補をいずれかの動作特定要素のテーブルに追加する。例えば動作要求が上述のように「2」回「点滅」である場合に、回数テーブルに「2」が存在

50

しなかったときは、ステップ S 2 6 にて回数テーブルに「2」を追加すればよい。

【0093】

このように、動作パターンテーブル 7 7 の内容を更新することが可能とするように選択部 1 0 5 の機能を構成することにより、メモリを節約する構成で且つオブジェクト指向的な処理が可能であると共に、更に多様な動作を実現することができる。

【0094】

更に、ボディ E C U 7 のマイコン 7 0 の C P U 7 1 は、選択部 1 0 5 としての機能により、次のような処理を行なう構成としてもよい。K A B アプリ 1 0 3 若しくは S A B アプリ 1 0 4、又は新たなアプリケーションプログラム 7 5 に基づくアプリにより動作要求が同時的にされた場合、選択部 1 0 5 は優先度が高いアプリからの動作要求を優先して受け付ける。以下に示す例では、セキュリティに関する S A B アプリ 1 0 4 が最も優先度が高いとする。なお、優先度の高低は R O M 7 2 に記憶されており、マイコン 7 0 の C P U 7 1 が参照可能である。

10

【0095】

図 1 0 は、動作要求の要求元のアプリケーションプログラムの優先度に応じて、C P U 7 1 が制御対象及び動作の内容を優先的に選択する処理手順の一例を示すフローチャートである。

【0096】

C P U 7 1 は、選択部 1 0 5 としての機能により、K A B アプリ 1 0 3 又は S A B アプリ 1 0 4 を含む各アプリケーションプログラム 7 5 , 7 5 , ... に基づくアプリからの動作要求を受け付ける (ステップ S 3 1)。

20

【0097】

C P U 7 1 は、他のアプリから他に動作要求を受け付けているか否かを判断する (ステップ S 3 2)。C P U 7 1 は、他に動作要求を受け付けていると判断した場合 (S 3 2 : Y E S)、受け付けている動作要求の要求元の各アプリのうちの最も優先度が高いアプリからの動作要求を優先させる (ステップ S 3 3)。他の動作要求については保持しておく。

【0098】

C P U 7 1 は優先させた動作要求に対応する動作パターンを選択し (ステップ 3 4)、処理を終了する。C P U 7 1 は以降、実施の形態 1 の図 7 のフローチャートに示した処理を実行する。

30

【0099】

C P U 7 1 は、ステップ S 3 2 にて他に動作要求を受け付けていないと判断した場合 (S 3 2 : N O)、そのままステップ S 3 1 で受け付けた動作要求に対応する動作パターンを選択し (S 3 4)、処理を終了し、図 7 のフローチャートに示した処理を実行する。

【0100】

このように、アプリケーション層 1 0 1 の選択部 1 0 5 及び出力ドライバ部 1 0 6 の機能は 1 つに共有化して再利用できる一方で、各アプリケーションプログラム 7 5 , 7 5 , ... が対応する機能の内、優先度が高い機能を優先的に実現させるべく適宜選択することができる。なおこのとき、出力ドライバ部 1 0 6 では、いずれの負荷への信号出力が優先的かを判断する必要がない。優先度の設定は、各動作要求を一括して受け付ける選択部 1 0 5 で変更でき、当該ソフトウェア構成は柔軟性に優れる。

40

【0101】

なお、開示された実施の形態は、全ての点で例示であって制限的なものではないと考えられるべきである。本発明の範囲は上述の説明ではなくて特許請求の範囲によって示され、特許請求の範囲と均等の意味及び範囲内での全ての変更が含まれることが意図される。

【図面の簡単な説明】

【0102】

【図 1】実施の形態 1 における車載制御システムの構成を示す構成図である。

【図 2】実施の形態 1 における車載制御システムに含まれるボディ E C U の内部構成を示

50

すブロック図である。

【図 3】実施の形態 1 におけるボディ ECU のマイコンの CPU にて実現される機能を概念的に示す機能ブロック図である。

【図 4】実施の形態 1 におけるボディ ECU の ROM に記憶されている動作パターンテーブルの内容例を示す説明図である。

【図 5】CPU が出力ドライバ部として機能する際の処理の具体例を概念的に示す説明図である。

【図 6】オブジェクト指向型における各制御対象毎の制御処理用のプログラムの実現例を比較例として示す説明図である。

【図 7】CPU が出力ドライバ部として機能する際の処理手順の具体例を示すフローチャートである。

【図 8】実施の形態 2 におけるボディ ECU の ROM に記憶されている動作パターンテーブルの内容例を示す説明図である。

【図 9】選択部としての機能により CPU が動作要求に対応する動作パターンを新たに動作パターンテーブルに追加する処理手順の一例を示すフローチャートである。

【図 10】動作要求の要求元のアプリケーションプログラムの優先度に応じて、CPU が制御対象及び動作の内容を優先的に選択する処理手順の一例を示すフローチャートである。

【図 11】従来のマイコンにおけるコンピュータプログラムの構成の概略を示す概略図である。

【符号の説明】

【0103】

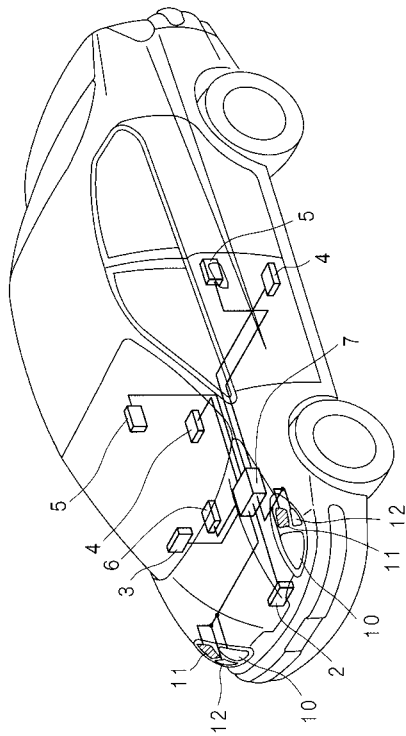
- 10 ヘッドライト（負荷）
- 11 方向指示器ランプ（負荷）
- 12 スモールランプ（負荷）
- 2 ホーン（負荷）
- 3 ブザー（負荷）
- 7 ボディ ECU（制御装置）
- 70 マイコン（マイクロコンピュータ）
- 72 ROM（記憶手段）
- 75 アプリケーションプログラム
- 76 アプリケーションドライバプログラム（コンピュータプログラム）
- 77 動作パターンテーブル
- 105 選択部（選択手段）
- 106 出力ドライバ部（出力手段）

10

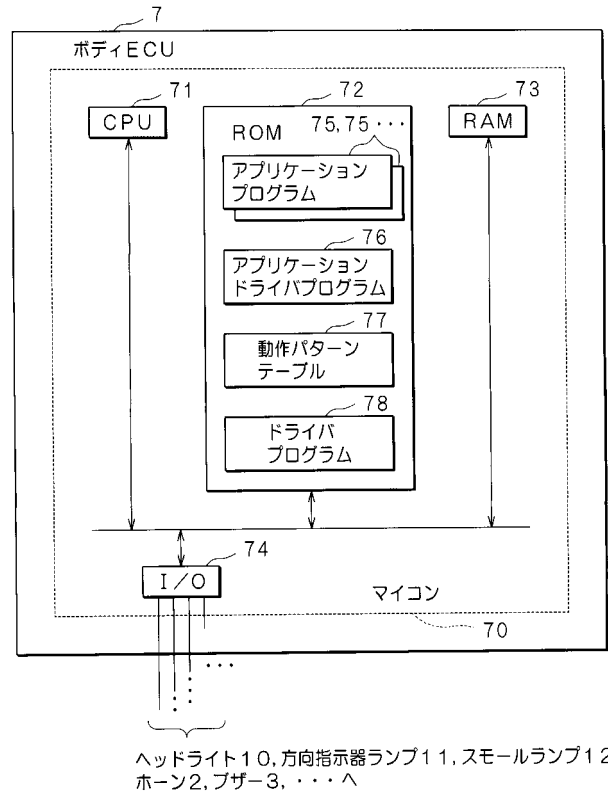
20

30

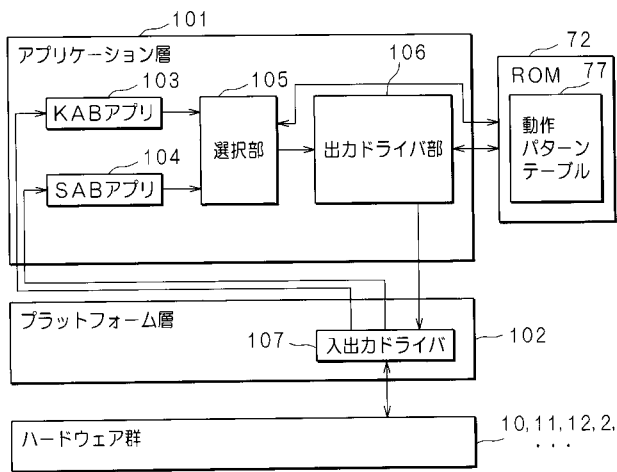
【 図 1 】



【 図 2 】



【 図 3 】

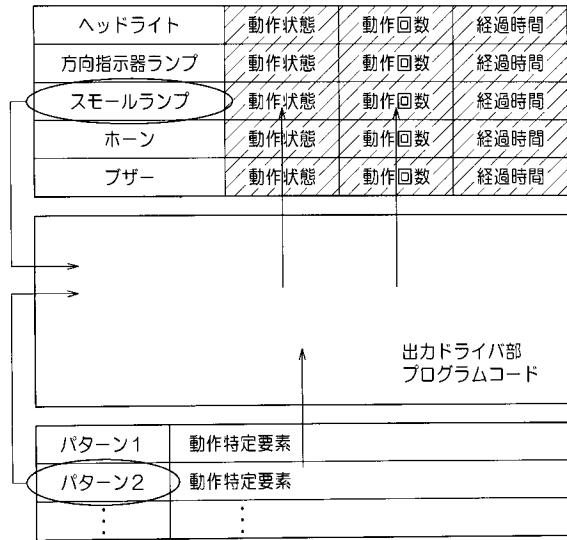


【 図 4 】

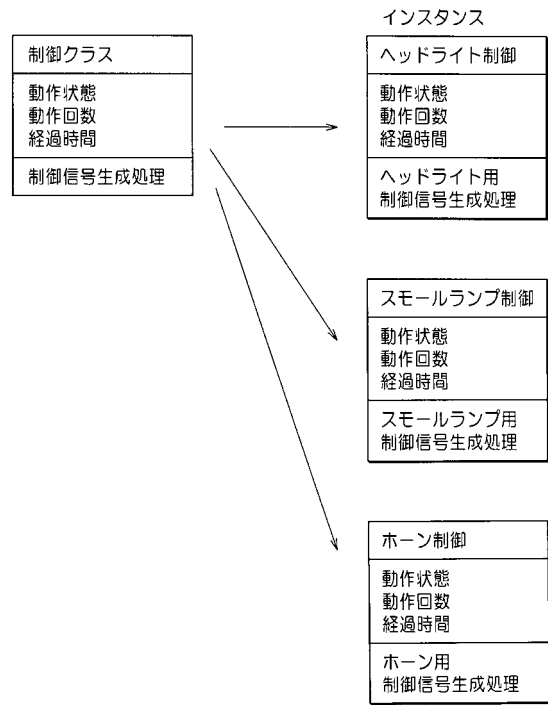
スモールランプ動作パターン

番号	待機時間	動作方式	回数	オン時間/オフ時間
1	0ms	点滅	1回	1000ms/0ms
2	0ms	点滅	2回	750ms/750ms
3	10ms	点灯	0回	0ms/0ms
⋮	⋮	⋮	⋮	⋮

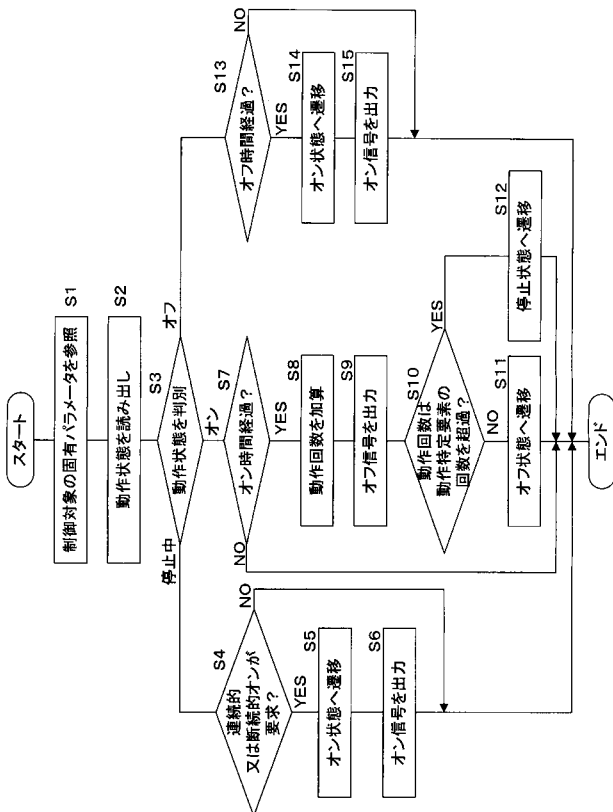
【図5】



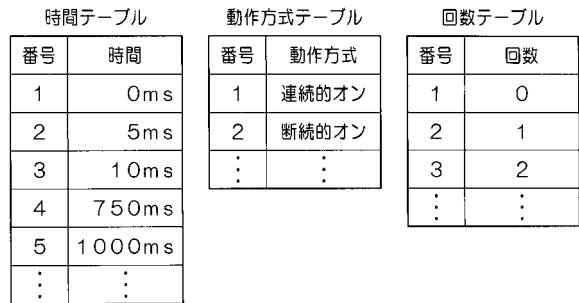
【図6】



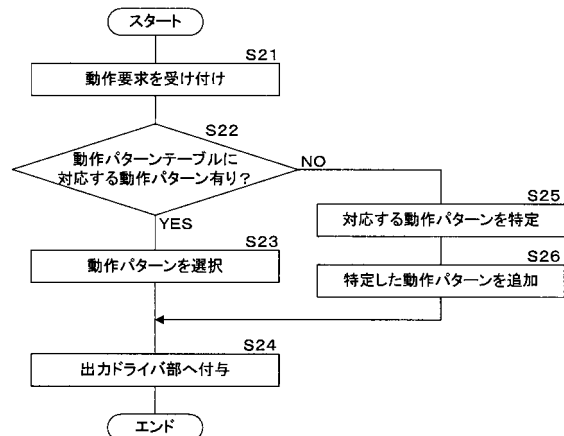
【図7】



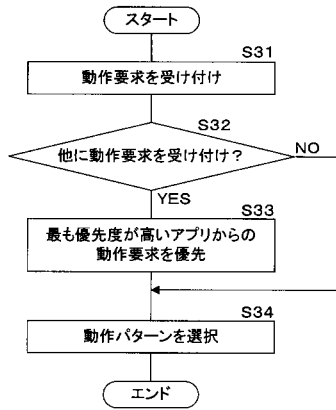
【図8】



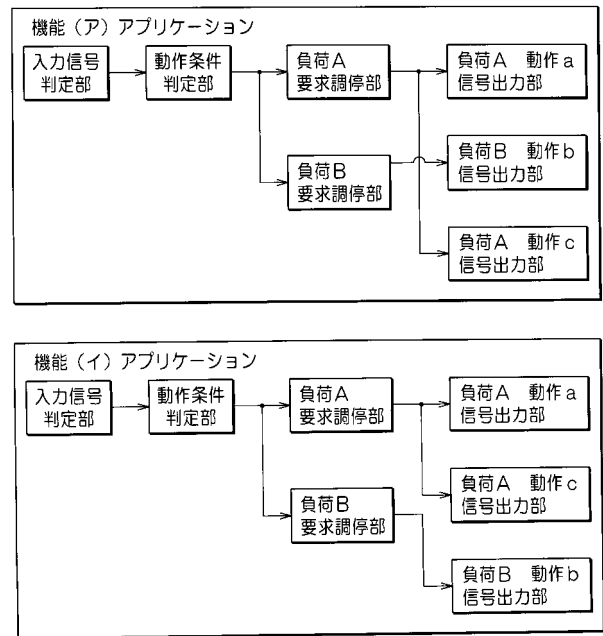
【図9】



【 図 1 0 】



【 図 1 1 】



フロントページの続き

(72)発明者 藤田 和仁

三重県四日市市西末広町1番14号 株式会社オートネットワーク技術研究所内