

(12) 特許協力条約に基づいて公開された国際出願

(19) 世界知的所有権機関
国際事務局

(43) 国際公開日
2017年11月30日(30.11.2017)



(10) 国際公開番号

WO 2017/203776 A1

(51) 国際特許分類:
H02P 29/00 (2016.01) G06F 9/48 (2006.01)

(21) 国際出願番号: PCT/JP2017/007625

(22) 国際出願日: 2017年2月28日(28.02.2017)

(25) 国際出願の言語: 日本語

(26) 国際公開の言語: 日本語

(30) 優先権データ:
特願 2016-104066 2016年5月25日(25.05.2016) JP

(71) 出願人: 日本電産株式会社 (NIDEC CORPORATION) [JP/JP]; 〒6018205 京都府京都市南区久世殿城町 3 3 8 番地 Kyoto (JP).

(72) 発明者: 張 綱(ZHANG, Gang); 〒6018205 京都府京都市南区久世殿城町 3 3 8 番地 日本電産株式会社内 Kyoto (JP).

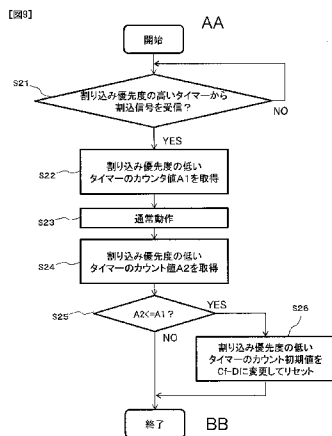
(74) 代理人: 北村 秀明 (KITAMURA, Hideaki); 〒6018205 京都府京都市南区久世殿城町 3 3 8 番地 Kyoto (JP).

(81) 指定国(表示のない限り、全ての種類の国内保護が可能): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) 指定国(表示のない限り、全ての種類の広域保護が可能): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), ユーラシア (AM, AZ, BY, KG, KZ, RU, TJ, TM), ヨーロッパ (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS,

(54) Title: MOTOR DRIVE SYSTEM, MOTOR CONTROL SYSTEM, AND SELF-PROPELLED ROBOT

(54) 発明の名称: モータ駆動システム、モータ制御システムおよび自走ロボット



S21 Has interrupt signal been received from high-interrupt-priority timer?
S22 Acquire counter value A1 of low-interrupt-priority timer
S23 Normal operation
S24 Acquire counter value A2 of low-interrupt-priority timer
S25 A2 < A1?
S26 Reset low-interrupt-priority timer with count start value thereof changed to Cf - D
AA Start
BB End

(57) Abstract: [Problem] To periodically enable motor control software and an operating system to be respectively executed on the same processor. [Solution] A processor of the motor drive system comprises: first and second timers which output first and second interrupt signals, respectively, when respective threshold values are reached; and a processor core. The first timer counts at a speed equal to or at an integer multiple of that of the second timer. The processor core executes motor control software in response to the first interrupt signal, and an operating system task in response to the second interrupt signal. On the occurrence of the second interrupt signal while the motor control software is being executed, the processor core sets a count start value of the second timer to a value at which the counting speed is increased by count value D with respect to a second threshold value as a reference, or to a value at which the second threshold value is counted longer by count value D. The count value D is greater than or equal to a count value which is equivalent to a difference between the occurrence time of the second interrupt signal and the processing completion time of the motor control software.

SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM,
GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

添付公開書類：

一 国際調査報告（条約第21条(3)）

(57) 要約：【課題】 モータ制御のソフトウェアおよびOSの各々を、同一のプロセッサ上で周期的に実行させる。【解決手段】 モータ駆動システムのプロセッサは、閾値に到達すると第1および第2割込信号をそれぞれ出力する第1および第2タイマーと、プロセッサコアとを備えている。第1タイマーは第2タイマーと等倍または整数倍の速くカウントする。プロセッサコアは、第1割込信号に応答してモータ制御ソフトウェアを実行し、第2割込信号に応答してOSのタスクを実行する。プロセッサコアは、モータ制御ソフトウェアの実行中に第2割込信号が発生すると、第2タイマーのカウント開始値を、第2閾値を基準としてカウント値Dだけ早くカウントされる値、または、第2閾値をカウント値Dだけ長くカウントされる値に設定する。カウント値Dは、第2割込信号の発生時刻とモータ制御ソフトウェアの処理完了時刻との差分に相当するカウント値以上である。

明 細 書

発明の名称：

モータ駆動システム、モータ制御システムおよび自走ロボット

技術分野

[0001] 本開示は、1つの信号処理回路（プロセッサ）が、複数のタスクを開始するタイミングを調整する技術に関する。

背景技術

[0002] 近年のモータ制御器には、モータの回転を制御する機能だけでなく、他の多くの機能が搭載されるようになりつつある。たとえば、モータ制御器をインターネットに接続させ、または他の機器と相互に通信させるための通信機能である。そのため、ユーザまたはアプリケーションソフトウェアが、モータ制御器の各機能を容易に利用できるよう、オペレーティングシステム（OS）が導入されるようになってきた。

[0003] モータ制御器を有するモータ駆動システムに導入されるOSとして、リアルタイムOSが知られている。リアルタイムOSは、プロセッサの時間（CPU時間）を管理し、ソフトウェア処理の指定された時間に正確に呼び出すリアルタイム性を持つ。リアルタイムOSは、システムタイマーからの割り込み信号に基づいて動作する。リアルタイムOSは、割り込み信号が発生する度に時間の経過を検知し、実行すべき処理をタスクとして、CPU時間、メモリ空間等の資源を割り当てる。

[0004] システムタイマーに割り込みを発生させる時間間隔を短くすると、OSはより細かく時間を管理することができる。ただしそうすると、単位時間におけるOSの呼び出し回数が増える。一般的には、呼び出しの度に、スタックへのデータの退避、スタックからのデータの復旧等を行わなければならないため、OSの導入によるオーバーヘッドが大きくなる。結果的に、OSによって占有されるCPU時間が増える。システムタイマー割り込みの時間間隔を無限に小さくできるわけではなく、現在では、1ミリ秒以上に設定するの

が現実的である。

[0005] 一方、制御ソフトウェアもまた、制御対象を制御するために一定時間ごとに繰り返し実行される。制御ソフトウェアを繰り返し実行する時間間隔は、制御周期と呼ばれている。制御周期は制御対象の性質によって異なり、制御対象の時定数によって決定されることが多い。たとえば、モータの動作を制御する制御ソフトウェアは、1ミリ秒よりもはるかに短い周期で実行される必要がある。

[0006] たとえば特開平10-161890号公報は、プログラムに一定の実行時間を割り当てるスケジューラを開示する。所与の周期で複数のプログラムを実行するにあたり、各プログラムの実行時間の合計値が当該周期を超える場合、つまり周期制約を満たせない場合がある。すると、スケジューラは、あるプログラムの処理を分割して、分割された処理が複数の周期にわたって実行されるよう割り当てる。それにより、当該周期で複数のプログラムの実行を実現している。

先行技術文献

特許文献

[0007] 特許文献1：日本国公開公報平10-161890号公報

発明の概要

発明が解決しようとする課題

[0008] 相対的に短い制御周期の制御ソフトウェアと、相対的に長い時間間隔でシステムタイマー割り込みを発生させてタスクを実行するOSとを同じプロセッサ上で動作させることが求められている。各々に設定された周期で制御ソフトウェアおよびOSを確実に実行することが必要とされる。

[0009] 本開示は、モータ制御のソフトウェアおよびOSの各々を、同一のプロセッサ上で周期的に実行する技術を提供する。

課題を解決するための手段

[0010] 本開示の例示的なモータ駆動システムによれば、モータ制御ソフトウェア

の実行中に、オペレーティングシステムを動作させるための第2割込信号が発生したときは、第2タイマーのカウント開始値を、第2閾値を基準としてカウント値Dだけ早くカウントされる値に設定し、または、第2閾値をカウント値Dだけ長くカウントされる値に設定する。当該カウント値Dは、第2割込信号が発生した時刻 T_a とモータ制御ソフトウェアの処理が完了した時刻 T_b との差分に相当するカウント値以上である。第1タイマーおよび第2タイマーは、クロック信号に基づいて、各カウント値を同じ周期で変化させ、または、一方のカウント値を他方のカウント値のN倍（N：2以上の整数）の周期で変化させる。これにより、各割り込みが干渉しないよう調整され、タスクを開始するタイミングを調整することができる。よって、モータ制御ソフトウェアの実行タイミングとオペレーティングシステムのタスクの実行タイミングとの重複を避けることができる。

[0011] 本開示の例示的なモータ制御システムは、上述のモータ駆動システムと、前記モータ駆動システムのモータの駆動を制御する制御信号を出力するシステムコントローラとを備えている。

[0012] 本開示の例示的な自走ロボットは、上述のモータ制御システムと、複数の車輪であって、前記モータによって駆動される車輪を含む複数の車輪と、物を載置する搬送テーブルとを備えている。

発明の効果

[0013] 本発明の例示的な実施形態によれば、モータ制御ソフトウェアおよびオペレーティングシステムのタスクの実行タイミングを調整するため、それぞれの実行タイミングの重複を避けることができる。

図面の簡単な説明

[0014] [図1]図1は、フィードバック型モータ制御モデルに含まれる、三種類の制御に関するループ構造を示す図である。

[図2]図2は、モータ制御のソフトウェアの処理ループの関係を示す図である。

[図3]図3は、モータ駆動システム1の構成を示す図である。

[図4]図4は、モータ制御ソフトウェア20の実行タイミングとリアルタイムOS30のタスク実行タイミングとが干渉している様子を示す図である。

[図5]図5は、本開示によるモータ駆動システム1のハードウェア構成の例を示す図である。

[図6]図6は、プロセッサ10の内部構造を示す図である。

[図7]図7は、システムタイマー122の内部構成を示す図である。

[図8]図8は、プロセッサ10の初期動作の手順を示すフローチャートである。

[図9]図9は、プロセッサ10のタスク起動タイミング調整処理の手順を示すフローチャートである。

[図10]図10は、主として、図9の処理を実行したときのシステムタイマー122のカウント値の変化を示す図である。

[図11]図11は、変形例によるプロセッサ10のタスク起動タイミング調整処理の手順を示すフローチャートである。

[図12]図12は、主として、図9の処理を実行したときのシステムタイマー122のカウント値の変化を示す図である。

[図13]図13は、自走ロボット200の構成を示す図である。

[図14]図14は、自走ロボット200のハードウェア構成を示す図である。

発明を実施するための形態

[0015] まず、図1～図3を参照しながら、モータ制御器の信号処理回路（プロセッサ）によって実行されるモータ制御のソフトウェアおよびオペレーティングシステムを説明し、続いて図4を参照しながら本願発明者が見出した課題を説明する。その後、図5以降の図面を参照しながら、本開示に係るモータ制御システムを説明する。以下の説明では、「オペレーティングシステム」を「OS」と略記する。

[0016] 図1は、フィードバック型モータ制御モデルに含まれる、三種類の制御に関するループ構造を示す。本モデルに従って設計されたモータ制御のソフトウェアは、主として通常電流制御処理a、速度制御処理bおよび位置制御処

理cをプロセッサに実行させる。これら三つの制御処理はそれぞれ異なる制御周期で実行させることが可能である。なお、各処理a～cの具体的な内容の説明は省略する。

[0017] 電流制御処理aおよび速度制御処理bの各制御周期は、モータのインダクタンスおよび回転慣性モーメントによって決定される。電流制御処理aおよび速度制御処理bは、それぞれ1ミリ秒未満、たとえば50ナノ秒の制御周期で実行することが求められ得る。

[0018] 1ミリ秒以上間隔で時間を管理するOSは、当然に、1ミリ秒未満の制御周期を必要とするモータ制御のソフトウェアを管理することはできない。そこで、たとえばプロセッサのペリフェラルであるハードウェアタイマーを用いて1ミリ秒未満の時間間隔で割込信号を発生させ、割込信号に基づいてプロセッサ上でモータ制御ソフトウェアを実行させることが考えられる。

[0019] 図2は、モータ制御のソフトウェアの処理ループの関係を示す。ハードウェアタイマーが50ナノ秒の時間間隔で割込信号を発生させると、割込信号の受信に応答して、モータ制御のソフトウェアはプロセッサに都度電流制御処理aを実行させる。プロセッサは、割込信号が10回発生する毎に速度制御を実行し、1000回発生する毎に位置制御を実行する。これにより、電流制御処理aは50ナノ秒毎に実行され、速度制御処理bは500ナノ秒毎に実行される。電流制御処理aおよび速度制御処理bは1ミリ秒未満の時間間隔で実行される。

[0020] 一方、同一プロセッサ上で実行されるリアルタイムOSに関しては、異なるタイマー、たとえばシステムタイマーを利用して、1ミリ秒ごとに割込信号を発生させればよい。これにより、1ミリ秒以上の制御周期を必要とする種々のソフトウェアをリアルタイムOS管理下のタスクとして実行することができる。

[0021] 図3は、モータ駆動システム1の構成を示す。

[0022] モータ駆動システム1は、プロセッサ10と、モータMとを有している。モータMの回転は、プロセッサ10がそれぞれ実行するモータ制御ソフトウ

エア20およびリアルタイムOS30のタスクによって制御される。

[0023] 図3では、理解の便宜のため、ハードウェアとしてのプロセッサ10およびモータMと、ソフトウェア（コンピュータプログラム）であるモータ制御ソフトウェア20およびリアルタイムOS30とを併記している。モータ制御ソフトウェア20およびリアルタイムOS30は、いずれもコンピュータプログラムに過ぎず、実際の処理の主体となるのはプロセッサ10である。しかしながら実際には、各コンピュータプログラムとプロセッサ10とが協働して見かけ上1つの具体的な装置として動作し、それにより各プログラムによって提供される機能が実現されている。そこで本明細書では、コンピュータプログラムが動作の主体であるかのように記載することがある。たとえば、モータ制御ソフトウェア20がモータを制御する、と記載したり、リアルタイムOS30が種々の命令をプロセッサ10に実行させる、などと記載することがある。ここでいうコンピュータプログラムは、モータ制御ソフトウェア20およびリアルタイムOS30に限られず、各ソフトウェアによって管理され実行されるプログラム（処理）である「タスク」も含まれる。

[0024] プロセッサ10は、複数のタイマーを有している。図3には、2つのタイマー、すなわちシステムタイマー122およびハードウェアタイマー123が示されている。

[0025] システムタイマー122およびハードウェアタイマー123は独立して動作する。いずれのタイマーもタイマーカウンタを有しており、クロック信号を受け取るとタイマーカウンタをカウントアップまたはカウントダウンする。タイマーカウンタ値が予め定められた値になると、オーバーフローまたはアンダーフローが発生したとして、タイマーは割込信号を出力する。

[0026] 本開示では、システムタイマー122は1ミリ秒ごとに割込信号を出力し、ハードウェアタイマー123は50マイクロ秒ごとに割込信号を出力するとして説明する。

[0027] モータ制御ソフトウェア20は、ハードウェアタイマー123からの割込信号によって実行される。図1を参照しながら説明したように、モータ制御

ソフトウェア 20 は、電流制御処理 a、速度制御処理 b および位置制御処理 c を含む。本開示では、少なくとも電流制御処理 a は 50 マイクロ秒ごとに実行される。

[0028] リアルタイム OS 30 は、システムタイマー 122 からの割込信号によって実行される。リアルタイム OS 30 では、スケジューラ 32 と、2つの待ち行列 34 a および 34 b とが管理されている。

[0029] スケジューラ 32 は、リアルタイム OS 30 上で動作する処理（タスク）を、実行可能処理の待ち行列 12 a、ブロック処理の待ち行列 12 b、またはプロセッサ上に割り当てる。また、スケジューラ 32 は、システムタイマーから割込信号を受け取り、その都度、管理下のどのタスクをプロセッサ 10 に実行させるか判断する。その判断の結果、スケジューラ 32 は、たとえば実行可能な最高優先度を持つタスクをプロセッサ 10 に実行させる。また、スケジューラ 32 は、プロセッサ 10 からのイベント待ち処理を中断する信号に応答して、所定のタスクをプロセッサ 10 に実行させる。

[0030] 本開示では、リアルタイム OS 30 のスケジューラ 32 によって、1 ミリ秒ごとに何らかのタスクが実行されることになる。

[0031] 次に、図 4 を参照しながら、本願発明者が見出した、モータ駆動システム 1 の動作上の課題を説明する。なお、以下の説明では、システムタイマー 122 およびハードウェアタイマー 123 は、各タイマーカウンタがオーバーフローしたときに割込信号を発生させるとする。ただし、これまでは、システムタイマー 122 は 1 ミリ秒ごとに、ハードウェアタイマー 123 は 50 マイクロ秒ごとにオーバーフローが発生する例を挙げていた。つまり、ハードウェアタイマー 123 はシステムタイマー 122 より 20 倍速くオーバーフローが発生する例を挙げていた。しかしながら、図面上煩雑になるため、以下では、ハードウェアタイマー 123 はシステムタイマー 122 より 2 倍速くオーバーフローが発生するとして説明する。

[0032] 図 4 は、モータ制御ソフトウェア 20 の実行タイミングとリアルタイム OS 30 のタスク実行タイミングとが干渉している様子を示す。図 4 (a) お

よび（b）はモータ制御ソフトウェア20の動作に関するタイミングを示し、図4（c）および（d）は、リアルタイムOS30の動作に関するタイミングを示す。

[0033] システムタイマー122（図4（d））およびハードウェアタイマー123（図4（b））の各々は、異なる周期で割り込み信号を出力するため、いずれ、両者の割込信号が同じタイミングもしくは近いタイミングで発生することになる。「近い」とは、先に発生した割り込み信号に基づいて実行されたタスクが終了する前に、後の割り込み信号が発生することを言う。図4は、両者の割込信号が近いタイミングで発生したときの状況を示している。

[0034] モータ制御ソフトウェア20について見ると、図4（b）に示すハードウェアタイマー123のカウント値がオーバーフローした時刻T0で、ハードウェアタイマー123は割り込み信号を出力する。その結果、プロセッサ10は、時刻T0からモータ制御ソフトウェア20の処理を実行する。処理は時刻T2まで継続する。時刻T0からT2までの時間帯では、プロセッサ10はモータ制御ソフトウェア20の処理に占有される。

[0035] 一方、リアルタイムOS30について見ると、図4（d）に示すシステムタイマー122のカウント値がオーバーフローした時刻T1で、システムタイマー122は割り込み信号を出力する。しかしながら、図示されるように、時刻T1は、時刻T0からT2までの時間帯に含まれている。プロセッサ10は、リアルタイムOS30のタスクを実行することができない。

[0036] そのため、リアルタイムOS30のタスクの実行は時刻T2まで保留され、時刻T2から開始される。本来、時刻T1から実行されるべきタスクが、時刻T2から実行されるため、 $(T2 - T1)$ の時間E1だけタスクの実行開始が遅延する。この遅延時間E1を、「プリエンプションによる開始遅延」と言う。図4（c）には、時刻T3から時刻T4までの期間もプリエンプションによる遅延時間E2も示されている。

[0037] 図4（d）に示すように、システムタイマー122によるオーバーフロー割り込みは周期Taで発生している。本来、制御周期Taで実行されるべき

リアルタイムOS 30のタスクは、プリエンプションによる開始遅延が発生すると、制御周期T_bで実行されることになり、想定された周期によるタスクの実行が阻害される。

[0038] 一般的には、制御周期が相対的に短い割り込み信号には高い優先度を与え、制御周期が相対的に長い割り込み信号には低い優先度を与える、という規則を採用する。同じ規則をモータ制御ソフトウェアとリアルタイムシステムが共存するシステムに適用する場合、リアルタイムシステムを駆動するタイマーの割り込みに低い優先度を与え、モータ制御ソフトウェアを駆動するタイマーの割り込みに高い優先度を与えることになる。そのため、図4では、モータ制御ソフトウェア20が先に実行されている場合には、システムタイマー122からの割り込み信号が発生しても、優先度が低いため、リアルタイムOS 30のタスクにプリエンプションによる開始遅延が発生していた。

[0039] ただし、上述の規則を適用しない場合には、リアルタイムOS 30のタスクが先に実行されていることにより、モータ制御ソフトウェア20にプリエンプションによる開始遅延が発生する状況も想定され得る。高速に回転するモータMを正確に制御するために、モータ制御ソフトウェア20の制御周期は相対的に短く設定されている。したがって、プリエンプションによる開始遅延が発生すると、モータ駆動システム1の動作に影響を与え得る。影響の程度は、リアルタイムOS 30に発生したプリエンプションによる開始遅延の影響よりも大きいと考えられる。

[0040] 本願発明者は、モータ制御ソフトウェア20およびリアルタイムOS 30の各々が、予め設定された制御周期で実行されるようにするために、本開示にかかる技術をなすに至った。以下、図5から13を参照しながら本開示の構成および動作を説明する。

[0041] まず、図5～7を参照しながら、本開示の処理を実現するためのモータ駆動システム1の具体的な構成を説明する。その後図8～13を参照しながら、プロセッサ10の具体的な処理を説明する。

[0042] 図5は、本開示によるモータ駆動システム1のハードウェア構成の例を示

す図である。

- [0043] モータ駆動システム 1 は、モータ M と、互いにバス 8 によって接続された、ROM (リードオンリーメモリ) 2、RAM (ランダムアクセスメモリ) 4、I/F (入出力インタフェース) 6 およびプロセッサ 10 を有している。図示されていない他の回路またはデバイス (AD 変換器など) が付加的にバス 11 に接続されていても良い。
- [0044] モータ M は、便宜上、モータ本体だけではなく、モータ本体を駆動するモータ駆動回路も含む。モータ M は、PWM 信号を受け取り、当該 PWM 信号に基づいて回転する。
- [0045] ROM 2 は、リアルタイム OS 30 のデータを格納する不揮発性メモリである。ROM 2 は、さらに、モータ制御ソフトウェア 20 を格納していてもよい。ただしモータ制御ソフトウェア 20 は、後述するプロセッサ 10 のプログラム用フラッシュメモリに格納されていてもよい。RAM 4 は、ワークメモリとして使用される揮発性メモリである。
- [0046] I/F 6 は、データ入出力端子であり、モータ駆動システム 1 が外部装置とデータを授受する際に利用される。I/F 6 は種々の接続端子であり得る。I/F 6 は、リアルタイム OS 30 のタスクによって生成されたデータを送信するための通信端子であってもよい。または、ROM 2 が書換可能であるときにおいて、I/F 6 は、新たなバージョンのリアルタイム OS 30 を受け取るデータ入力端子であってもよい。または I/F 6 は、無線による通信を行う無線通信回路であってもよい。
- [0047] プロセッサ 10 は、RAM 4 に格納されたリアルタイム OS 30 を実行し、またはプロセッサ 10 内部の SRAM に格納されたモータ制御ソフトウェア 20 を実行する。
- [0048] モータ駆動システム 1 に電源が投入されると、プロセッサ 10 は ROM 2 からリアルタイム OS 30 のデータを読み出し、RAM 4 に展開する。これにより、プロセッサ 10 はリアルタイム OS 30 を実行することができる。また、プロセッサ 10 は、ROM 2 からモータ制御ソフトウェア 20 を読み

出し、プロセッサ 10 内部の S R A M に展開する。これにより、プロセッサ 10 はモータ制御ソフトウェア 20 を実行することができる。

[0049] プロセッサ 10 が、後述する本開示の処理を実行することにより、モータ制御ソフトウェア 20 およびリアルタイム O S 30 のタスクをそれぞれ周期的に実行する。プロセッサ 10 は、モータ制御ソフトウェア 20 を実行することによって生成した PWM 信号をモータ M に送り、モータ M の回転を制御する。また、プロセッサ 10 は、リアルタイム O S 30 を動作させることによって生成したデータを、I / F 6 を介して外部に送信する。

[0050] モータ駆動システム 1 は、たとえば 32 ビットの汎用的なマイクロコントローラによって実現され得る。そのようなマイクロコントローラは、1 個または複数の集積回路チップから構成され得る。

[0051] 次に、プロセッサ 10 の内部構造を説明する。

[0052] 図 6 は、プロセッサ 10 の内部構造を示す。

[0053] プロセッサ 10 は、内部データバス 119 で互いに接続された、プロセッサコア 120 と、分周器 121 と、システムタイマー 122 と、ハードウェアタイマー 123 と、S R A M 124 と、プログラム用フラッシュメモリ 125 と、E E P R O M 126 と、定周期／可変周期タイマー群 127 と、周辺回路群 128 と、レジスタ 129 とを有する。

[0054] プロセッサコア 120 は、内部にさらに種々の構成を有する演算回路である。プロセッサコア 120 の内部には、たとえば、プログラムカウンタ、命令レジスタ、命令デコーダ、スタックポインタ、汎用レジスタ、タイマー割り込み制御回路を有している。これらはいずれも一般的であり、詳細な説明は省略する。

[0055] 分周器 121 は、プロセッサ 10 の外部に設けられた、水晶またはシリコンを用いる発振子 118 から外部クロック信号を受け取り、外部クロック信号を分周して所定の周波数のクロック信号を出力する。

[0056] システムタイマー 122 は、分周器 121 から受け取ったクロック信号に基づいて、プロセッサ 10 を動作させるための所定周波数の内部クロック信

号を生成する。

[0057] またシステムタイマー 1 2 2 およびハードウェアタイマー 1 2 3 は、それぞれ、分周器 1 2 1 から受け取ったクロック信号を利用してタイマーカウンタ（後述）をカウントアップし、カウンタ値が予め定められた値になると割込信号をプロセッサコア 1 2 0 に出力する。なお、ハードウェアタイマーの数は任意である。複数個設けられていてもよい。また、タイマーカウンタは、カウントダウンを行ってもよい。本開示ではカウントアップの例を挙げて説明する。

[0058] なお、分周器 1 2 1 から、どの周波数のクロック信号をシステムタイマー 1 2 2 およびハードウェアタイマー 1 2 3 に入力するかは、プロセッサコア 1 2 0 が選択することができる。

[0059] S R A M 1 2 4 は、プロセッサコア 1 2 0 が動作する際に利用されるワークメモリであり、書き換え頻度が高いデータを格納する。

[0060] プログラム用フラッシュメモリ 1 2 5 は、プログラムおよび書き換え頻度が低いデータを格納するメモリである。

[0061] E E P R O M 1 2 6 は、電源が遮断されても記録内容が消えない不揮発メモリの一種であり、電氣的に内容を書き換えることが可能なメモリである。以下で説明する、予め設定された各種の値は、E E P R O M 1 2 6 に格納されている。

[0062] 定周期／可変周期タイマー群 1 2 7 は、PWM信号を生成するための複数のタイマーが組み合わされた回路である。定周期タイマーと可変周期タイマーとを組み合わせることにより、パルス幅変調された信号を生成することができる。

[0063] 周辺回路群 1 2 8 は、上述した回路以外の回路の集合である。周辺回路群 1 2 8 は、たとえば、高精度アナログ回路（A D コンバータ、D A コンバータ）、モータ制御回路、シリアル信号とパラレル信号とを相互に変換する U A R T (Universal Asynchronous Receiver Transmitter)、通信インタフェース装置を含む。

[0064] レジスタ 129 は、システムタイマー 122 およびハードウェアタイマー 123 の各々の優先度を指定するデータを保持する。本開示では、ハードウェアタイマー 123 の優先度の方がシステムタイマー 122 の優先度よりも高く設定される。たとえば小さい値ほど優先度が高いという規則を採用した場合には、ハードウェアタイマー 123 の優先度の方が、システムタイマー 122 の優先度よりも小さい値が設定される。優先度が相対的に高いタイマーからの割込信号は、優先度が相対的に低いタイマーからの割込信号よりも優先して処理される。たとえば、プロセッサコア 120 は、リアルタイム OS 30 のタスクの実行中であっても、ハードウェアタイマー 123 から割込信号を受けたときは、モータ制御ソフトウェア 20 を優先して実行する。本開示において、上述の優先度を「割り込み優先度」と呼ぶことがある。

[0065] 本開示においては、ハードウェアタイマー 123 のオーバーフロー割り込みに応答してプロセッサコア 120 がモータ制御ソフトウェア 20 を実行している時間帯に、システムタイマー 122 のオーバーフロー割り込みが発生したとしても、プロセッサコア 120 はシステムタイマー 122 の割り込みの処理を行わない。したがって、リアルタイム OS 30 の実行は保留される。

[0066] 次に、システムタイマー 122 の構成を説明する。

[0067] 図 7 は、システムタイマー 122 の内部構成を示す。

[0068] システムタイマー 122 は、タイマーカウンタ 130 および自動再ロードレジスタ (ARR : Auto Reload Register) 132 を有している。ARR 132 は、上限カウント値 C_f を示すデータ 136 を保持している。タイマーカウンタ 130 は、分周器 121 から供給されるクロック信号に基づいてカウンタ値を、カウント初期値 C₀ からカウントアップする。そしてタイマーカウンタ 130 は、カウンタ値が ARR 132 に保持された上限カウント値 C_f に到達するとカウントを停止し、プロセッサコア 120 に、タイマーカウンタ 130 にオーバーフローが発生したことを通知するオーバーフロー割込信号を出力する。その後、タイマーカウンタ 130 は、カウンタ値をカウ

ント初期値C 0 にリセットする。

[0069] 本開示では、カウント初期値C 0 から上限カウント値C f に到達するまでの時間は、1 ミリ秒である。ただし後述の処理により、その値を変更して、オーバーフローが発生するまでの時間を一時的に変更することにより、その後、オーバーフローが発生するタイミングをずらすことができる。

[0070] 発振子1 1 8 および分周器1 2 1 は十分な精度のクロック信号を出力する。当該クロック信号を受けてカウントアップするシステムタイマー1 2 2 およびハードウェアタイマー1 2 3 もまた、十分な精度でカウントアップし、周期P ごとにオーバーフローを発生させ得る。

[0071] タイマーカウンタ1 3 0 は、通常は、カウント初期値C 0 から上限カウント値C f までカウントする。しかしながら、プロセッサコア1 2 0 からカウント設定値C s を指定する信号を受け取ると、当該カウント設定値C s からカウントを開始する。カウント設定値C s を適宜設定することにより、タイマーカウンタ1 3 0 がオーバーフローするまでの時間を調整することができる。

[0072] 本実施形態では、タイマーカウンタ1 3 0 は指定されたカウント設定値C s からカウントアップし、上限カウント値C f に到達すると、カウント値をカウント初期値C 0 にリセットする。したがって、タイマーカウンタ1 3 0 は、カウント設定値C s が指定された時だけ、カウントを開始する値を変更する。

[0073] なお、タイマーカウンタ1 3 0 がオーバーフローするまでの時間を調整する他の方法として、カウント初期値C 0 を変更するのではなく、上限カウント値C f を変更する方法が考えられる。上限カウント値C f をより小さく設定すれば、タイマーカウンタ1 3 0 がオーバーフローするまでの時間は短くなり、上限カウント値C f をより大きく設定すれば、タイマーカウンタ1 3 0 がオーバーフローするまでの時間は長くなる。上限カウント値C f を変更した後は、変更後の値が引き続き採用されてもよいし、タイマーカウンタ1 3 0 が一旦リセットされた後は所定値に戻してもよい。

[0074] ハードウェアタイマー１２３（図６）の構成もまた、図７に示すシステムタイマー１２２の構成と同じであり得る。ただしハードウェアタイマー１２３においては、カウント初期値Ｃ０から上限カウント値Ｃｆに到達するまでの時間は、５００マイクロ秒である。システムタイマー１２２よりも２倍速い理由は、ハードウェアタイマー１２３のタイマーカウンタ１３０は、システムタイマー１２２のタイマーカウンタ１３０よりも、１クロック信号あたり２倍の大きさにカウント値を変化させるからである。この結果、同じクロック信号を用い、同じ上限カウント値が設定されていたとしても、ハードウェアタイマー１２３はシステムタイマー１２２よりも常に２倍速くオーバーフロー割込信号を発生させることができる。

[0075] 上述の説明から理解されるように、プロセッサコア１２０は、システムタイマー１２２およびハードウェアタイマー１２３の各々について、タイマーカウンタ１３０を読み出すことも可能であるし、タイマーカウンタ１３０のカウント初期値Ｃ０および／または上限カウント値Ｃｆを書き換えることが可能である。

[0076] 次に、図８～図１１を参照しながら、本開示によるプロセッサ１０の動作を説明する。

[0077] 図８は、プロセッサ１０の初期動作の手順を示すフローチャートである。

[0078] ステップＳ１１において、プロセッサ１０は、システムタイマーおよびハードウェアタイマーに同一のカウントアップ周期を設定する。具体的には、プロセッサ１０のプロセッサコア１２０は、システムタイマー１２２およびハードウェアタイマー１２３に、分周器１２１から同じ周波数のクロック信号を入力する。これにより、システムタイマー１２２およびハードウェアタイマー１２３は、等しい周期でカウントアップを行う。

[0079] ステップＳ１２において、プロセッサコア１２０は、通常動作を行う。通常動作とは、プロセッサコア１２０が、システムタイマー１２２からオーバーフロー割込信号およびハードウェアタイマー１２３からのオーバーフロー割込信号に応答して、モータ制御ソフトウェア２０およびリアルタイムＯＳ

30を切り換えて実行する動作である。

[0080] 図8の処理により、モータ駆動システム1の動作が開始される。

[0081] 次に、図9および図10を参照する。

[0082] 図9は、プロセッサ10のタスク起動タイミング調整処理の手順を示すフローチャートである。図9の処理は、図8の開始後から、通常処理と平行して実行される。また図10は、主として、図9の処理を実行したときのシステムタイマー122のカウンタ値の変化を示す。図10(a)および(b)はハードウェアタイマー123に基づくモータ制御ソフトウェア20の動作に関するタイミングを示す。図10(c)および(d)は、システムタイマー122に基づくリアルタイムOS30の動作に関するタイミングを示す。また図10(e)は図10(d)の部分拡大図である。

[0083] 図9のステップS21において、プロセッサコア120は、割り込み優先度の高いタイマーから割込信号を受信するまで、通常処理を実行する。ここでいう「割り込み優先度の高いタイマー」とは、ハードウェアタイマー123である。なお、割り込み優先度はレジスタ129(図6)に予め設定されている。プロセッサコア120は、ハードウェアタイマー123から割込信号を受け取ると、続くステップS22を実行する。図10(b)に示されるように、時刻T0において、ハードウェアタイマー123から割込信号が出力されている。

[0084] ステップS22において、プロセッサコア120は、割り込み優先度の低いタイマーのカウンタ値A1を取得する。ここでいう「割り込み優先度の低いタイマー」とはシステムタイマー122である。プロセッサコア120は、取得したカウンタ値A1をSRAM124に格納する。図10(d)に示されるように、時刻T0において、システムタイマー122のカウンタ値A1が取得されている。

[0085] ステップS23において、プロセッサコア120は、通常動作を行う。すなわちプロセッサコア120は、ハードウェアタイマー123から割込信号を受け取っているため(ステップS21)、モータ制御ソフトウェア20を

実行する（図10（a））。モータ制御ソフトウェア20の実行が完了した後、プロセッサコア120は次のステップS24を実行する。

[0086] ステップS24において、プロセッサコア120は、割り込み優先度の低いタイマーのカウント値A2を取得する。プロセッサコア120は、取得したカウンタ値A2をSRAM124に格納する。図10（d）に示されるように、時刻T10において、システムタイマー122のカウント値A2が取得されている。

[0087] ステップS25において、プロセッサコア120は、取得した各カウンタ値A1およびA2が、 $A2 \leq A1$ であるか否かを判定する。 $A2 \leq A1$ である場合には処理はステップS26に進み、 $A2 > A1$ である場合には処理はステップS26に進む。ステップS25の処理の意味は、モータ制御ソフトウェア20の実行中に、割り込み優先度の低いシステムタイマー122がオーバーフローしたか否かを判定することにある。システムタイマー122はカウントアップしているため、後に取得した値A2が、先に取得した値A1以下の場合には、システムタイマー122がオーバーフローして一旦リセットされたことを意味する。ステップS26の判定処理を実行することにより、プロセッサコア120は、システムタイマー122およびハードウェアタイマー123の割込信号が同じタイミングもしくは近いタイミングで発生したことを知ることができる。

[0088] なお、図10（d）は、 $A2 \leq A1$ が成り立つ場合の例である。時刻T0と時刻10の間で、システムタイマー122はオーバーフロー割り込みを出力していることが理解される。

[0089] ステップS26において、プロセッサコア120は、割り込み優先度の低いシステムタイマー122のカウント初期値Csを、カウント上限値CfからDだけ前の値、つまり、カウント上限値Cfから見てDだけ早くカウントされる値（ $Cf - D$ ）に変更してシステムタイマー122をリセットする。これにより、システムタイマー122はカウント初期値 $Cf - D$ からカウントを開始する。

[0090] ここで、カウント値Dは、優先度の高いモータ制御ソフトウェア20の処理が終了した後にすぐに優先度の低い割り込みが発生するように設定する値である。より具体的には、図10(d)に示すように、システムタイマー122のオーバーフローが発生し割り込み信号が発生する時刻T11と、モータ制御ソフトウェア20の処理が完了した時刻T10との差に相当するカウント値以上の値である。カウント値Dを(T11-T10)に設定すると、優先度の高いモータ制御ソフトウェア20の処理が終了した後の次のカウントで優先度の低い割り込みが発生する。カウント値Dを(T11-T10)より長く設定して若干の余裕を持たせてもよい。たとえば、モータ制御ソフトウェア20の最悪実行時間に相当するカウント値をカウント値Dとして設定することができる。図10(e)は、(T11-T10)より長い時間に相当するカウント値Dを設定したときの例である。システムタイマー122は、T10よりも前の時刻Txにリセットされ、カウント値Cs(=Cf-D)からカウントが開始されている。

[0091] 上述の図9のステップS25の説明は、カウントアップするシステムタイマー122に適用される処理である。システムタイマー122がカウントダウンを行う場合には、先に取得されたカウント値A1と、後に取得されたカウント値A2との関係は、 $A1 \leq A2$ に読み替えればよい。 $A1 \leq A2$ が成り立つとき、プロセッサコア120は、モータ制御ソフトウェア20の実行中にシステムタイマー122からのアンダーフロー割込信号が発生したことを検出する。

[0092] 次に、図11および図12を参照しながら、変形例を説明する。図9および図10の例では、システムタイマー122に設けられたタイマーカウンタ130のカウント初期値を変更した。本変形例では、タイマーカウンタ130の上限カウント値を変更する。

[0093] 図11は、変形例によるプロセッサ10のタスク起動タイミング調整処理の手順を示すフローチャートである。また図12は、主として、図9の処理を実行したときのシステムタイマー122のカウント値の変化を示す。

- [0094] 図11の処理のうち、ステップS21からS25までは図9と同じであるため説明を省略する。
- [0095] ステップS31において、プロセッサコア120は、割り込み優先度の低いシステムタイマー122の上限カウント値Cfを、これまでの値からカウントDだけ遅くカウントされる値に変更する。変更前の上限カウント値CfをCfaとおくと、変更後の上限カウント値Cfは、 $Cfa + D$ と表現される。これにより、システムタイマー122はこれまでの上限カウント値よりもカウントDだけ遅い時間にオーバーフローする。つまり、システムタイマー122からのオーバーフロー割込信号の発生を、カウント値Dに相当する時間だけ遅らせることができる。
- [0096] 図12(d)には、カウント上限値Cfが($Cfa + D$)に設定された結果、モータ制御ソフトウェア20の処理が完了した時刻以後の時刻T12においてシステムタイマー122がオーバーフロー割込信号を発生させていることが示されている。
- [0097] ステップS32において、プロセッサコア120は、システムタイマー122のカウント値リセット後に、システムタイマー122の上限カウント値Cf($=Cfa + D$)を、再びDだけ小さい値に設定する。つまり、ステップS31において変更される前の上限カウント値Cfaに戻す。これにより、上限カウント値変更後のシステムタイマー122は、変更前のシステムタイマー122と比較すると、カウント値Dに相当する時間だけ遅くカウントを開始し、オーバーフローを発生させることになる。よって、ハードウェアタイマー123のオーバーフロー割込信号に応答して実行されるモータ制御ソフトウェア20の処理後に、システムタイマー122がオーバーフローを発生し、リアルタイムOS30のタスクを実行させることができる。
- [0098] なお、上述の説明では、ハードウェアタイマー123はシステムタイマー122よりも2倍または20倍速くオーバーフロー割込信号を発生させる例を挙げた。しかしながらこれらの値は一例に過ぎない。ハードウェアタイマー123はシステムタイマー122よりもN倍(N:正の整数)高い周波数

でカウント値を変化させればよい。そのために、プロセッサコア 120 は、分周器 121 からハードウェアタイマー 123 に入力されるクロック信号の周波数を、システムタイマー 122 に入力されるクロック信号よりも N 倍高周波にすればよい。なお、N は 1 であってもよい。このときは、ハードウェアタイマー 123 の仕様が優先されるため、ハードウェアタイマー 123 も 1 ミリ秒未満でオーバーフローを発生させることになる。その場合には、プロセッサコア 120 は、システムタイマー 122 からの割込信号が N 回発生する度に 1 回、割込信号に応答してリアルタイム OS 30 のタスクを実行すればよい。

[0099] あるいは、システムタイマー 122 のカウント値の変化量を 1 としたとき、ハードウェアタイマー 123 はカウント値の変化量を N（N：正の整数）に設定して、カウント値を変化させればよい。

[0100] また、上述の説明では、モータ制御ソフトウェア 20 およびリアルタイム OS 30 の 2 つのソフトウェアを動作させる例を挙げた。しかしながら、リアルタイム OS 30 の制御周期よりも速い他のソフトウェアをさらに同時に動作させることも可能である。その場合には、複数のハードウェアタイマー 123 が用いられる。プロセッサコア 120 は、全てのソフトウェアの実行タイミング終了後にリアルタイム OS 30 のタスクが実行されるよう、システムタイマー 122 のカウント開始値または上限カウント値を調整する。たとえばプロセッサコア 120 は、上述の処理により、各ハードウェアタイマー 123 のオーバーフロータイミングおよびソフトウェアの実行終了時刻に基づいて、システムタイマー 122 のカウント開始値または上限カウント値をそれぞれ決定し、重複が生じないカウント開始値または上限カウント値を一つ決定すればよい。

[0101] 具体例として、50 マイクロ秒の実行周期を持つモータ制御ソフトウェアと、100 マイクロ秒周期の実行周期を持つレゾルバーリーダーソフトウェアと、1 ミリ秒または 1 ミリ秒より遅い周期で動作する複数のタスクを管理するリアルタイム OS 30 とを、プリアンクションによる開始遅延を生じさせ

ることなくそれぞれ動作させることができる。

[0102] (応用例) 以下、上述のモータ駆動システムを搭載した応用例を説明する。

[0103] 図13は、自走ロボット200の構成を示す。自走ロボット200は、搬送テーブル201と、システムコントローラ202と、モータ駆動システム203Aおよび203Bと、2つの駆動輪204Aおよび204Bとを有する。

[0104] 搬送テーブル201は運搬物を載置するテーブルである。

[0105] システムコントローラ202は、上位装置250からの有線または無線による指示に基づいて、自走ロボット200の進行方向および走行速度を制御する。具体的には、システムコントローラ202は、2つの駆動輪である左輪204Aおよび右輪204Bの回転速度および回転方向を決定し、モータ駆動システム203Aおよび203Bへ制御信号を送信する。

[0106] モータ駆動システム203Aおよび203Bは、システムコントローラ202からの制御信号に基づいて、それぞれ、駆動輪204Aおよび204Bの回転速度および回転方向を独立して制御する。

[0107] 図14は、自走ロボット200のハードウェア構造を示す。参考のため、自走ロボット200を構成しない上位装置150を破線で示す。

[0108] システムコントローラ202は、プロセッサ10Cを有している。プロセッサ10Cは、図6で示すハードウェア構造を有している。そのため、プロセッサ10Cの詳細な説明は省略する。

[0109] モータ駆動システム203Aは、プロセッサ10Aと、モータM-Aとを有する。モータ駆動システム203Bは、プロセッサ10Bと、モータM-Bとを有する。モータ駆動システム203Aおよび203Bは、それぞれ、図5に示すモータ駆動システム1に相当する。システムコントローラ202は、モータの駆動を制御する種々の制御信号、たとえば、モータの回転を開始させる制御信号、モータの回転速度を指示する制御信号、モータの回転を停止させる制御信号を、モータ駆動システム203Aおよび203Bの各々

に出力する。モータ駆動システム203Aおよび203Bは、システムコントローラ202からの制御信号に基づいて、それぞれ上述の処理を行って、モータ制御ソフトウェア20およびリアルタイムOS30を実行する。さらに、システムコントローラ202のプロセッサ10Cもまた、モータ制御ソフトウェアおよびリアルタイムOSを実行し、上述の同期処理を行ってもよいし、モータ制御ソフトウェアに代えて、リアルタイムOSよりも速い制御周期の他のソフトウェアを実行してもよい。

[0110] なお、プロセッサ10Aおよび10Bとプロセッサ112Bとは、必ずしも同じメーカーの同じ型番の製品である必要はない。モータM-AおよびM-Bについても同様である。自走ロボット200として必要とされる性能を有していればよい。モータ駆動システム203Aおよび203Bは、それぞれインテリジェントモータと呼ばれることもある。

[0111] なお、システムコントローラ202、モータ制御システム203Aおよび203Bをまとめて、モータ制御システムと呼ぶことがある。

[0112] 上述の例は、2つの後輪が駆動輪であるとして説明したが、これは一例である。たとえば前輪および後輪を別個のモータで駆動する二輪車、複数の車輪を別個のモータで駆動する三輪車、四輪車などにも、本開示は適用可能である。さらに、本開示は、システムコントローラが1つのモータを有するモータ駆動システムの動作を制御するモータ制御システムにも適用可能である。

[0113] 本開示によれば、異なる周期を持つ互いに無関係な複数の制御プログラムを同一プロセッサ上で並列的に実行させる場合であっても、互いの実行が干渉せず、プリエンプションによる開始遅延を生じない。従来はセマフォなどによってプログラム間の同期を行っていた。しかしながら本開示の方法によれば、セマフォを使うことなく、オーバーヘッドなしに実行順序を保証することができる。

[0114] 本開示における例示的なモータ駆動システムにおいて、プロセッサコアは、カウント値Dとして、モータ制御ソフトウェアの最悪実行時間に相当するカ

ウント値を設定する。

[0115] 第1タイマーは、カウント値が第1閾値に到達した後は予め設定された第1初期値からカウント値を変化させる。カウント値が第2閾値に到達した後は、第2タイマーは、予め設定されたカウント開始値と異なる第2初期値からカウント値を変化させる。

[0116] 第2タイマーが、クロック信号に基づいてカウント値を増加させるときにおいて、モータ制御ソフトウェアの実行終了時の第2タイマーのカウント値A2が、モータ制御ソフトウェアの実行開始時の第2タイマーのカウント値A1以下であれば、プロセッサコアは、モータ制御ソフトウェアの実行中に第2割込信号が発生したことを検出する。

[0117] 第2タイマーが、クロック信号に基づいてカウント値を減少させるときにおいて、モータ制御ソフトウェアの実行終了時の第2タイマーのカウント値A2が、モータ制御ソフトウェアの実行開始時の第2タイマーのカウント値A1以上であれば、プロセッサコアは、モータ制御ソフトウェアの実行中に第2割込信号が発生したことを検出する。

[0118] プロセッサコアは、オペレーティングシステムのタスクの実行中であっても、第2割込信号が発生したときは、モータ制御ソフトウェアを優先して実行する。

[0119] 第1タイマーが、第2タイマーと同じ周波数でカウント値を変化させるときにおいて、第1タイマーのカウント値の変化量は、第2タイマーのカウント値の変化量よりもN倍大きい。

[0120] 第1タイマーが、第2タイマーと同じ周波数でカウント値を変化させるときにおいて、プロセッサコアは、第2割込信号がN回発生する度に1回、第2割込信号に応答してオペレーティングシステムのタスクを実行する。

[0121] 第1タイマーは、第2タイマーが受け取るクロック信号よりもN倍高い周波数のクロック信号を受け取る。

産業上の利用可能性

[0122] 本開示の例示的なモータ駆動システムは、相対的に短い制御周期の制御ソ

フトウェアと、相対的に長い時間間隔でシステムタイマー割り込みを発生させてタスクを実行するOSとを同じプロセッサ上で動作させる際に有用である。

符号の説明

[0123] 1…モータ駆動システム 10…プロセッサ 120…プロセッサコア
 121…分周器 122…システムタイマー 123…ハードウェアタイマ
ー 124…SRAM 125…プログラム用フラッシュメモリ 126…
EEPROM 127…定周期／可変周期タイマー群 128…周辺回路群
 129…レジスタ130…タイマーカウンタ 132…自動再ロードレジ
スタ Cs…カウント設定値 Cf…上限カウント値

請求の範囲

[請求項1]

モータと、
クロック信号を出力する発振回路と、
前記クロック信号に基づいて動作して、前記モータを制御するプロセッサと
を有するモータ駆動システムであって、
前記プロセッサは、
設定された第1 カウント開始値からカウント値を変化させ、前記
カウント値が第1 閾値に到達した時刻に第1 割込信号を出力する第1
タイマーと、
設定された第2 カウント開始値からカウント値を変化させ、前記
カウント値が第2 閾値に到達した時刻に第2 割込信号を出力する第2
タイマーと、
前記第1 割込信号および前記第2 割込信号に応答して、前記モータを制御するためのモータ制御ソフトウェアおよびオペレーティングシステムを切り換えて実行するプロセッサコアと
を備えており、
前記クロック信号に基づいて、前記第1 タイマーは、前記第2 タイマーと同じ周波数またはN倍（N：2以上の整数）の周波数でカウント値を変化させ、
前記プロセッサコアは、
前記第1 割込信号に応答して前記モータ制御ソフトウェアを実行し、前記第2 割込信号に応答して前記オペレーティングシステムのタスクを実行し、
前記モータ制御ソフトウェアの実行中に前記第2 割込信号が発生したときは、前記第2 タイマーの前記カウント開始値を、前記第2 閾値を基準としてカウント値Dだけ早くカウントされる値に設定し、または、前記第2 閾値をカウント値Dだけ長くカウントされる値に設定

し、

前記カウント値Dは、前記第2割込信号が発生した時刻T_aと前記モータ制御ソフトウェアの処理が完了した時刻T_bとの差分に相当するカウント値以上である、モータ駆動システム。

[請求項2] 前記プロセッサコアは、前記カウント値Dとして、前記モータ制御ソフトウェアの最悪実行時間に相当するカウント値を設定する、請求項1に記載のモータ駆動システム。

[請求項3] 前記第1タイマーは、前記カウント値が前記第1閾値に到達した後は予め設定された第1初期値から前記カウント値を変化させ、

前記第2タイマーは、前記カウント値が前記第2閾値に到達した後は、予め設定された、前記カウント開始値と異なる第2初期値から前記カウント値を変化させる、請求項1または2に記載のモータ駆動システム。

[請求項4] 前記第2タイマーが、前記クロック信号に基づいて前記カウント値を増加させるときにおいて、

前記モータ制御ソフトウェアの実行終了時の前記第2タイマーのカウント値A₂が、前記モータ制御ソフトウェアの実行開始時の前記第2タイマーのカウント値A₁以下であれば、前記プロセッサコアは、前記モータ制御ソフトウェアの実行中に前記第2割込信号が発生したことを検出する、請求項1から3のいずれかに記載のモータ駆動システム。

[請求項5] 前記第2タイマーが、前記クロック信号に基づいて前記カウント値を減少させるときにおいて、

前記モータ制御ソフトウェアの実行終了時の前記第2タイマーのカウント値A₂が、前記モータ制御ソフトウェアの実行開始時の前記第2タイマーのカウント値A₁以上であれば、前記プロセッサコアは、前記モータ制御ソフトウェアの実行中に前記第2割込信号が発生したことを検出する、請求項1から3のいずれかに記載のモータ駆動シス

テム。

[請求項6] 前記プロセッサコアは、前記オペレーティングシステムのタスクの実行中であっても、前記第2割込信号が発生したときは、前記モータ制御ソフトウェアを優先して実行する、請求項1から5のいずれかに記載のモータ駆動システム。

[請求項7] 前記第1タイマーが、前記第2タイマーと同じ周波数でカウント値を変化させるときにおいて、前記第1タイマーのカウント値の変化量は、前記第2タイマーのカウント値の変化量よりもN倍大きい、請求項1から6のいずれかに記載のモータ駆動システム。

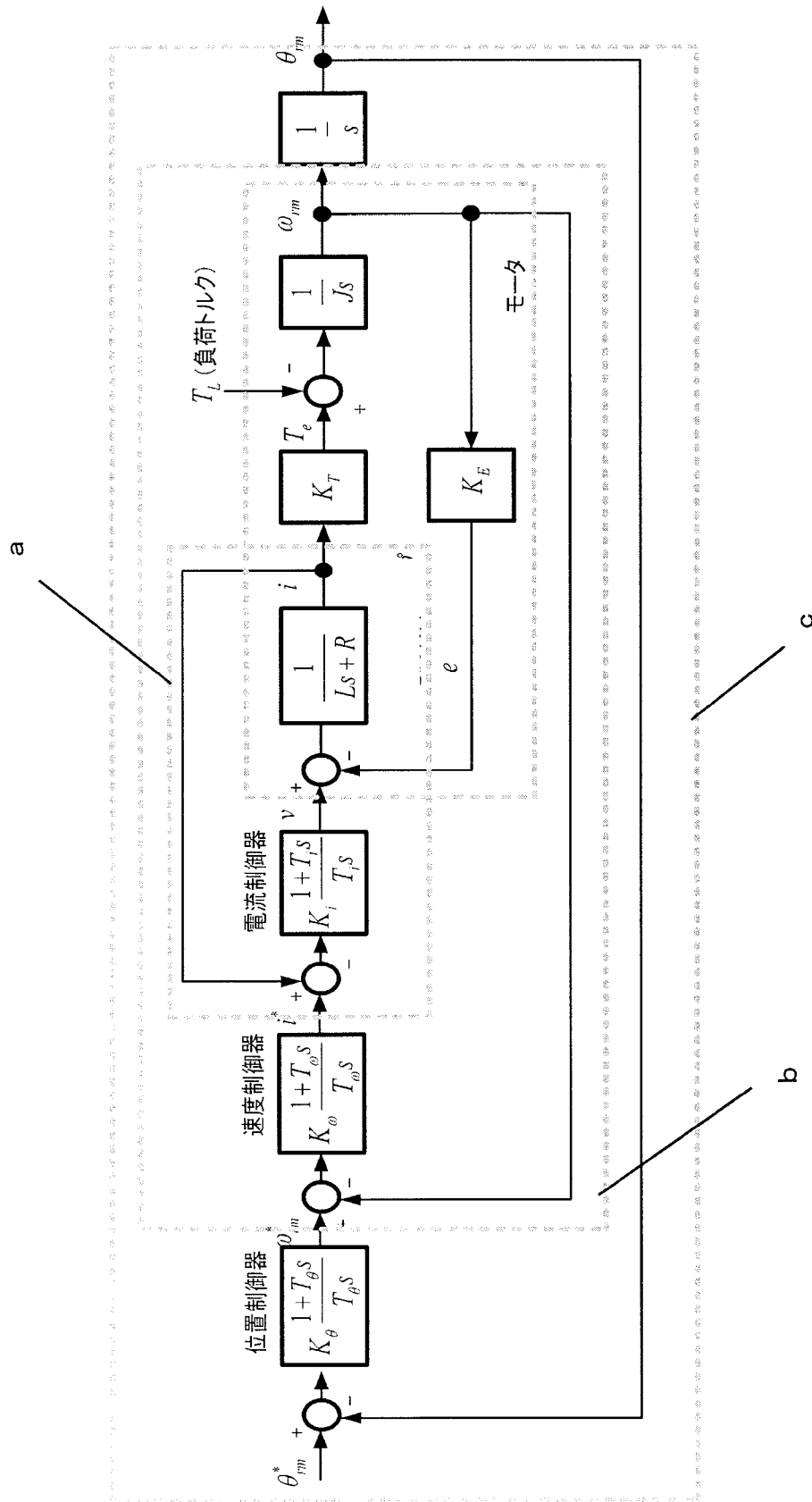
[請求項8] 前記第1タイマーが、前記第2タイマーと同じ周波数でカウント値を変化させるときにおいて、前記プロセッサコアは、前記第2割込信号がN回発生する度に1回、第2割込信号に応答して前記オペレーティングシステムのタスクを実行する、請求項1から6のいずれかに記載のモータ駆動システム。

[請求項9] 前記第1タイマーは、前記第2タイマーが受け取るクロック信号よりもN倍高い周波数のクロック信号を受け取る、請求項1から6のいずれかに記載のモータ駆動システム。

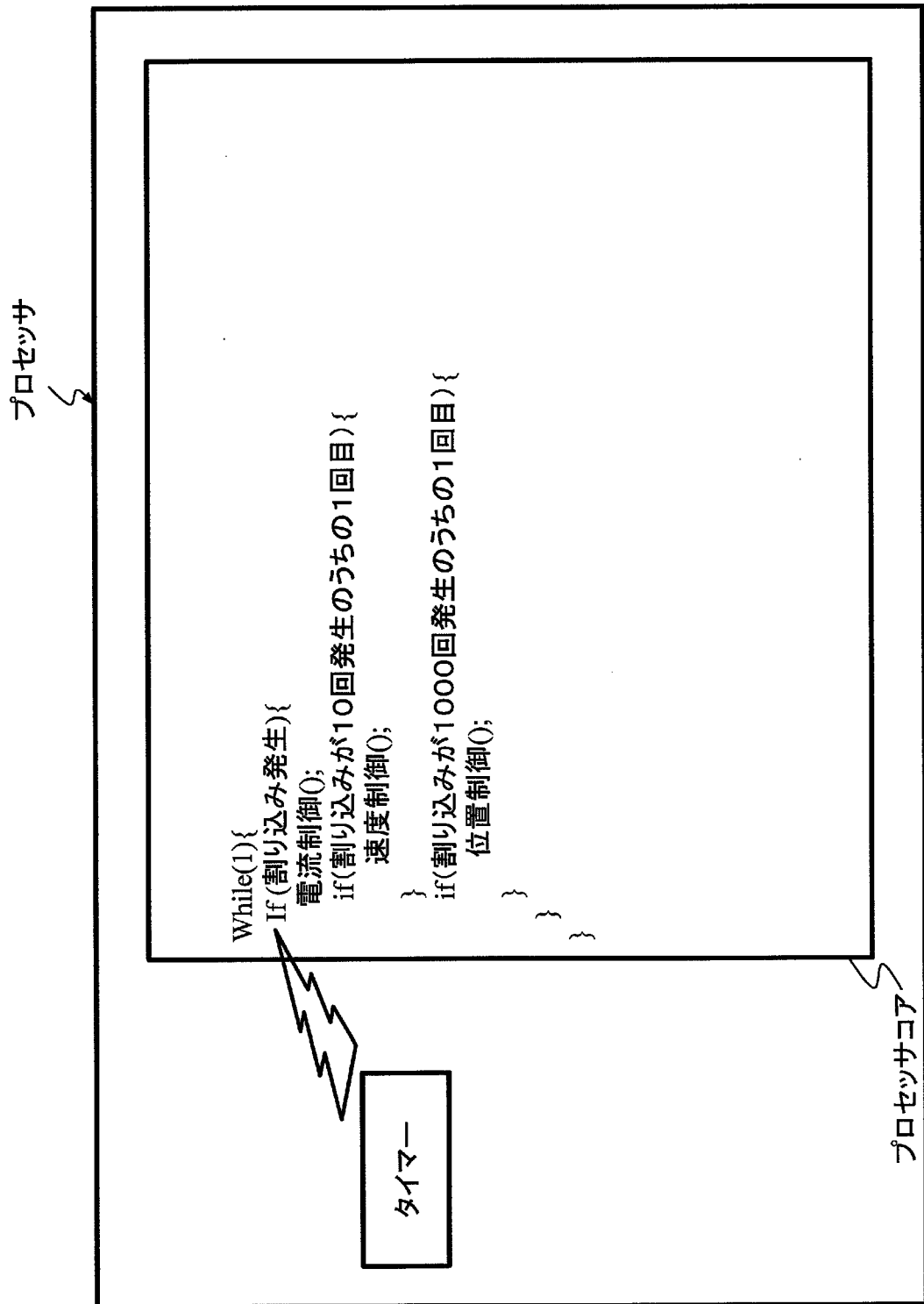
[請求項10] 請求項1から9のいずれかに記載のモータ駆動システムと、前記モータ駆動システムのモータの駆動を制御する制御信号を出力するシステムコントローラとを備えたモータ制御システム。

[請求項11] 請求項10に記載のモータ制御システムと、
複数の車輪であって、前記モータによって駆動される車輪を含む複数の車輪と、
物を載置する搬送テーブルとを備えた自走ロボット。

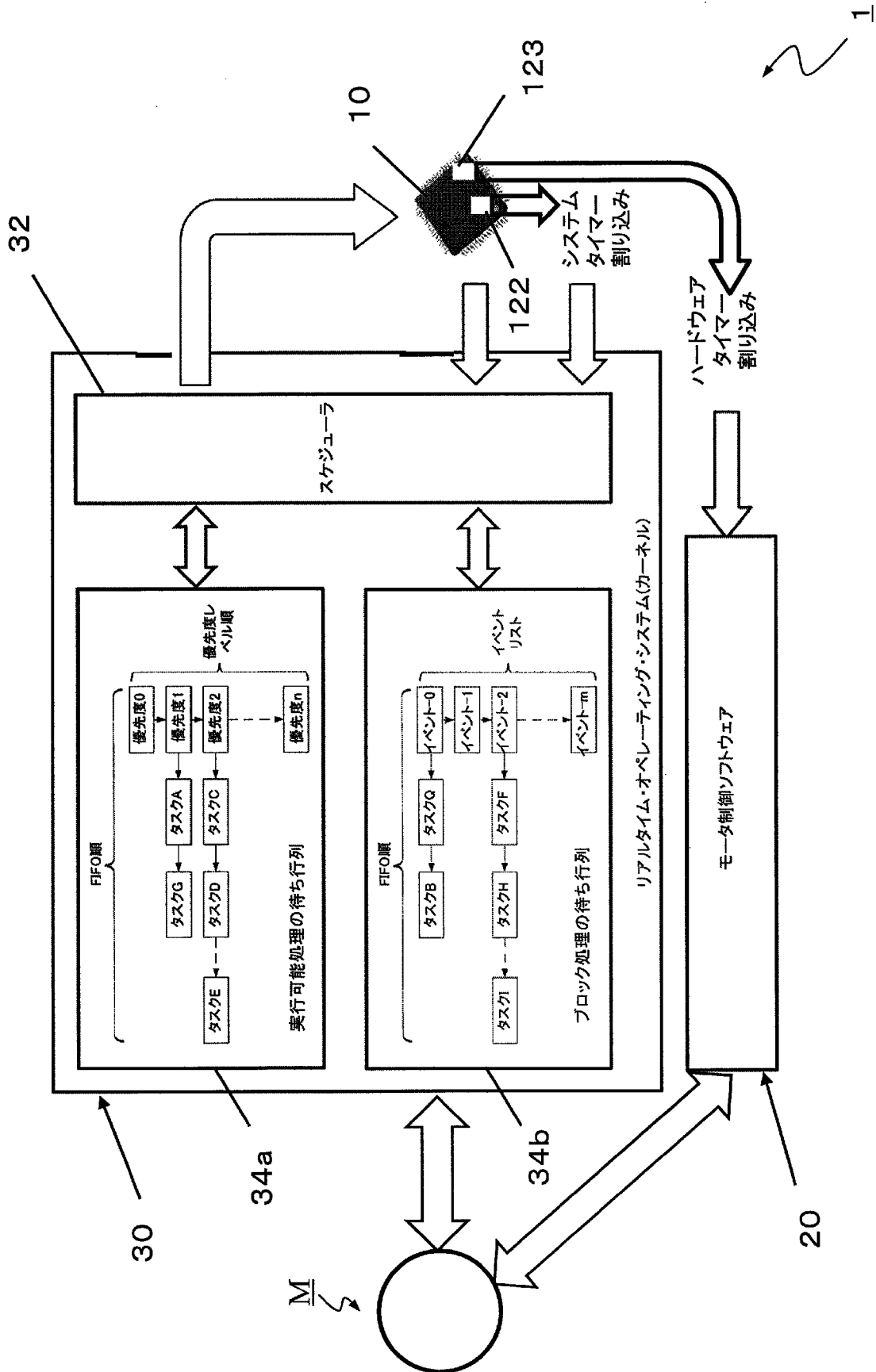
[図1]



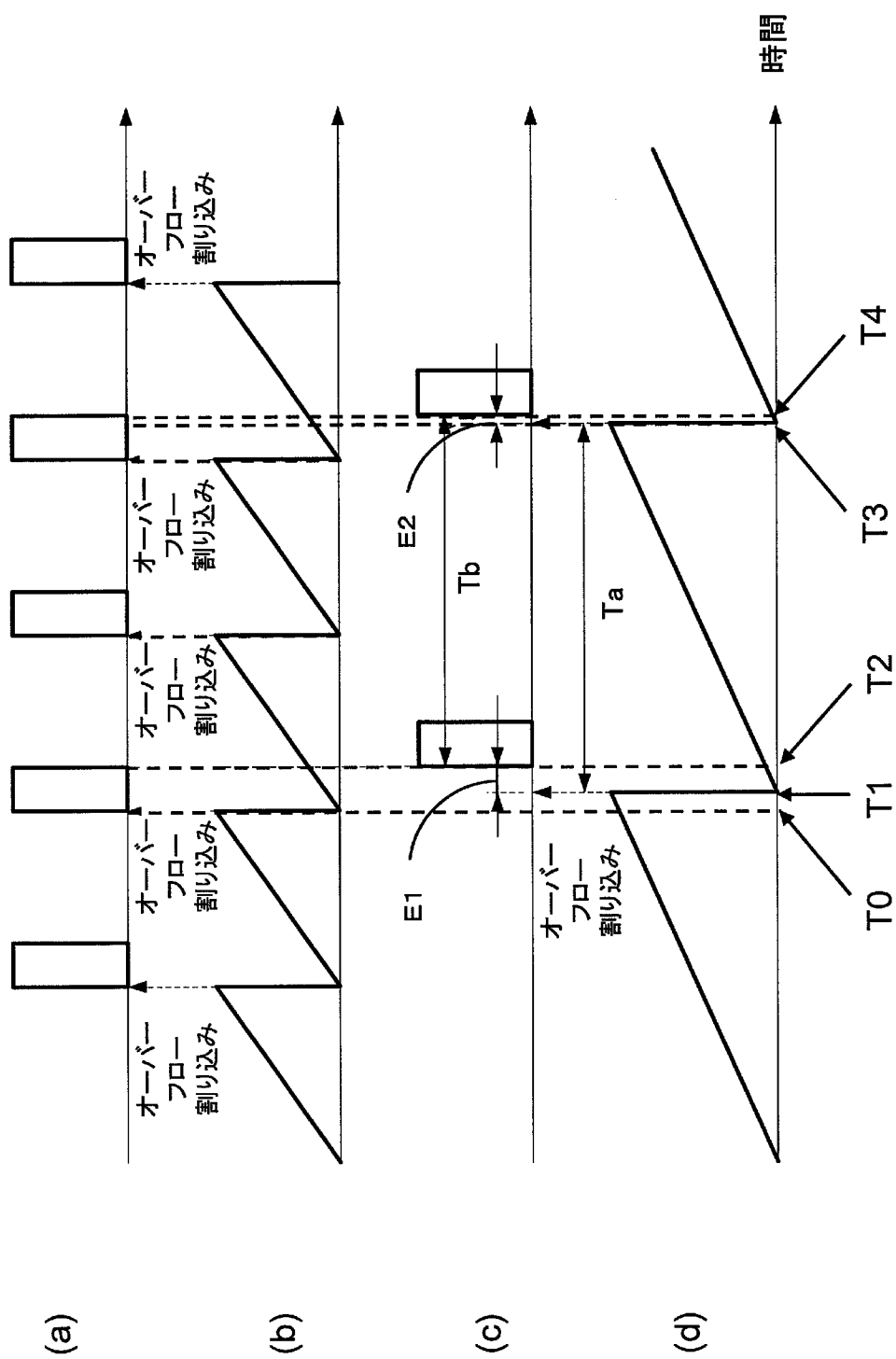
[図2]



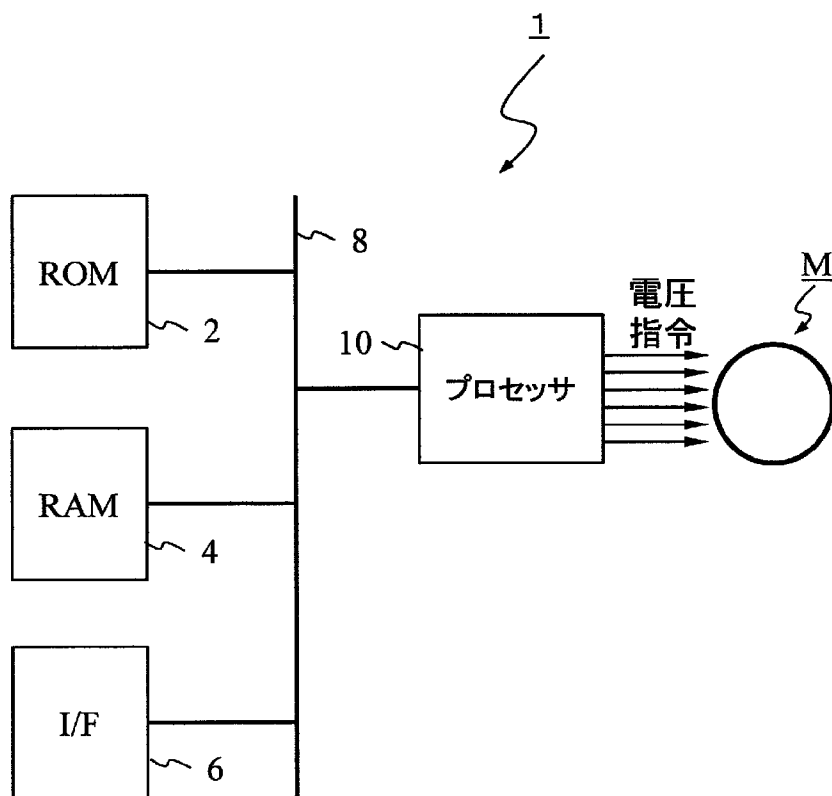
[図3]



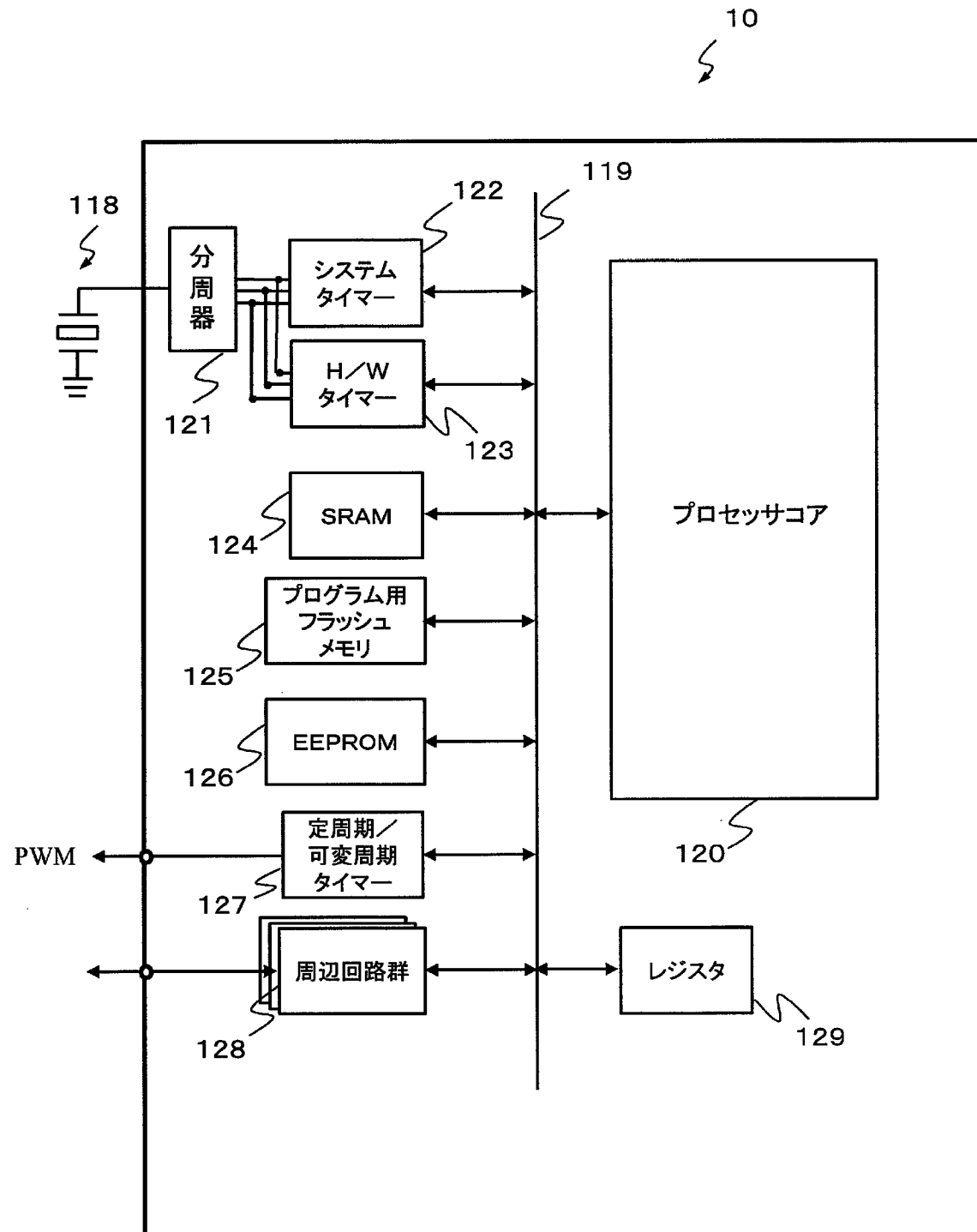
[図4]



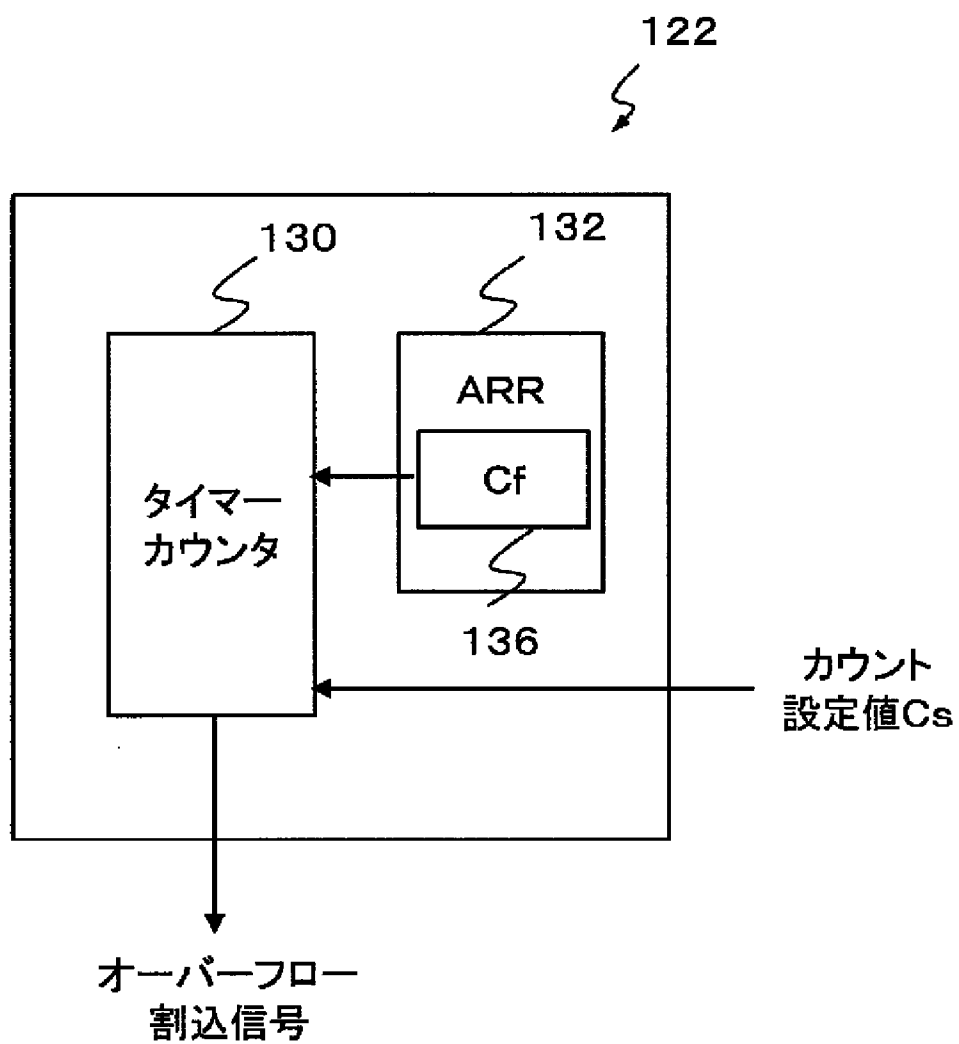
[図5]



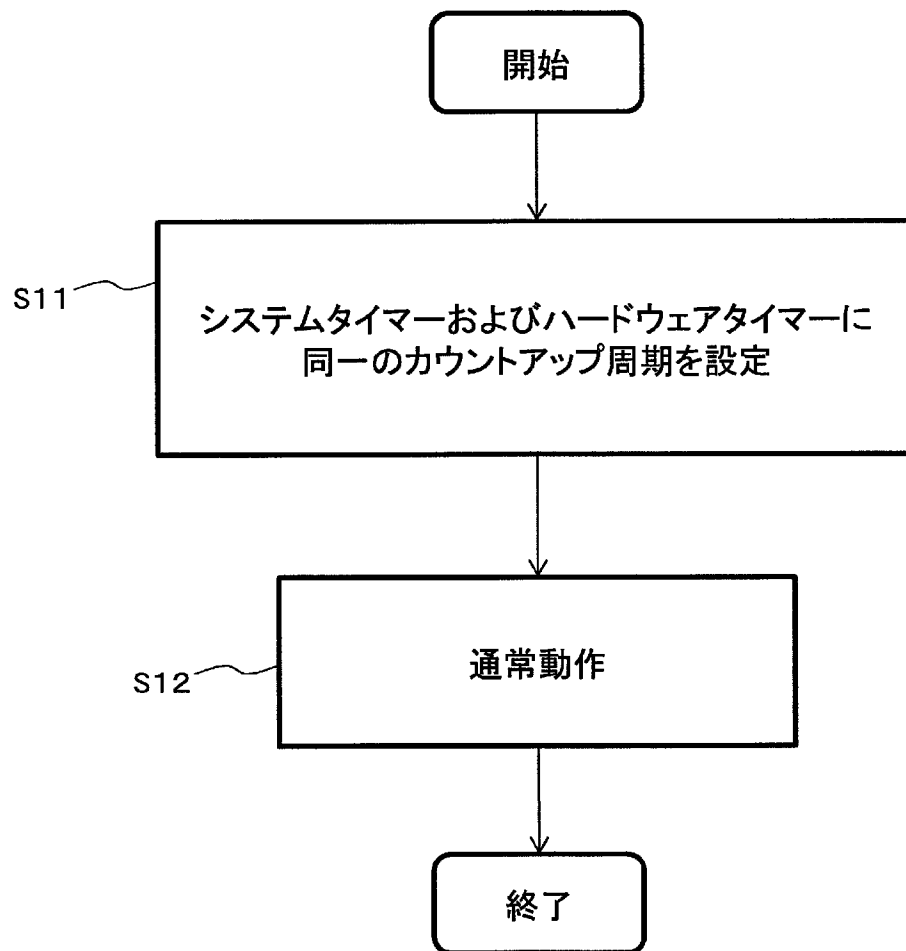
[図6]



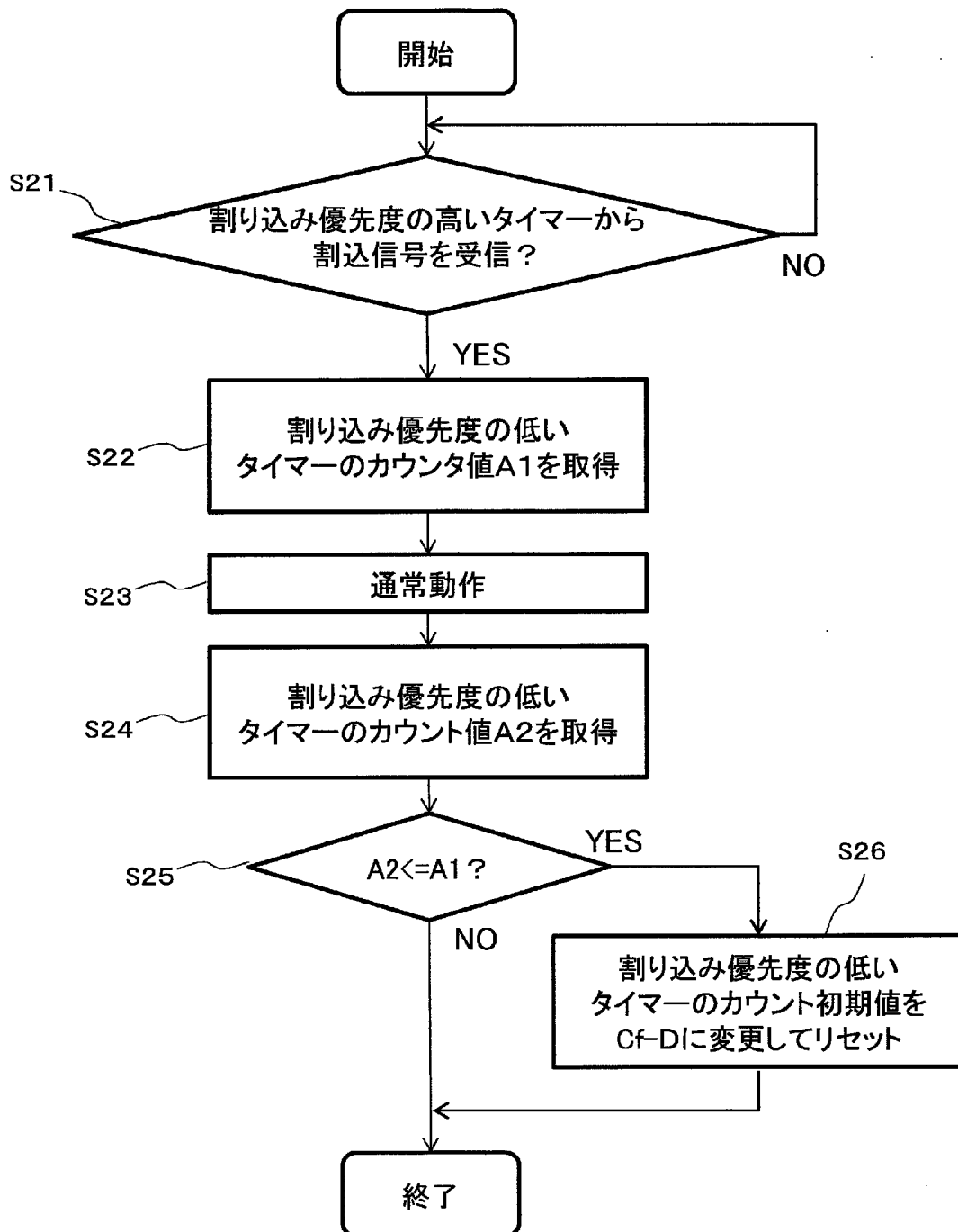
[図7]



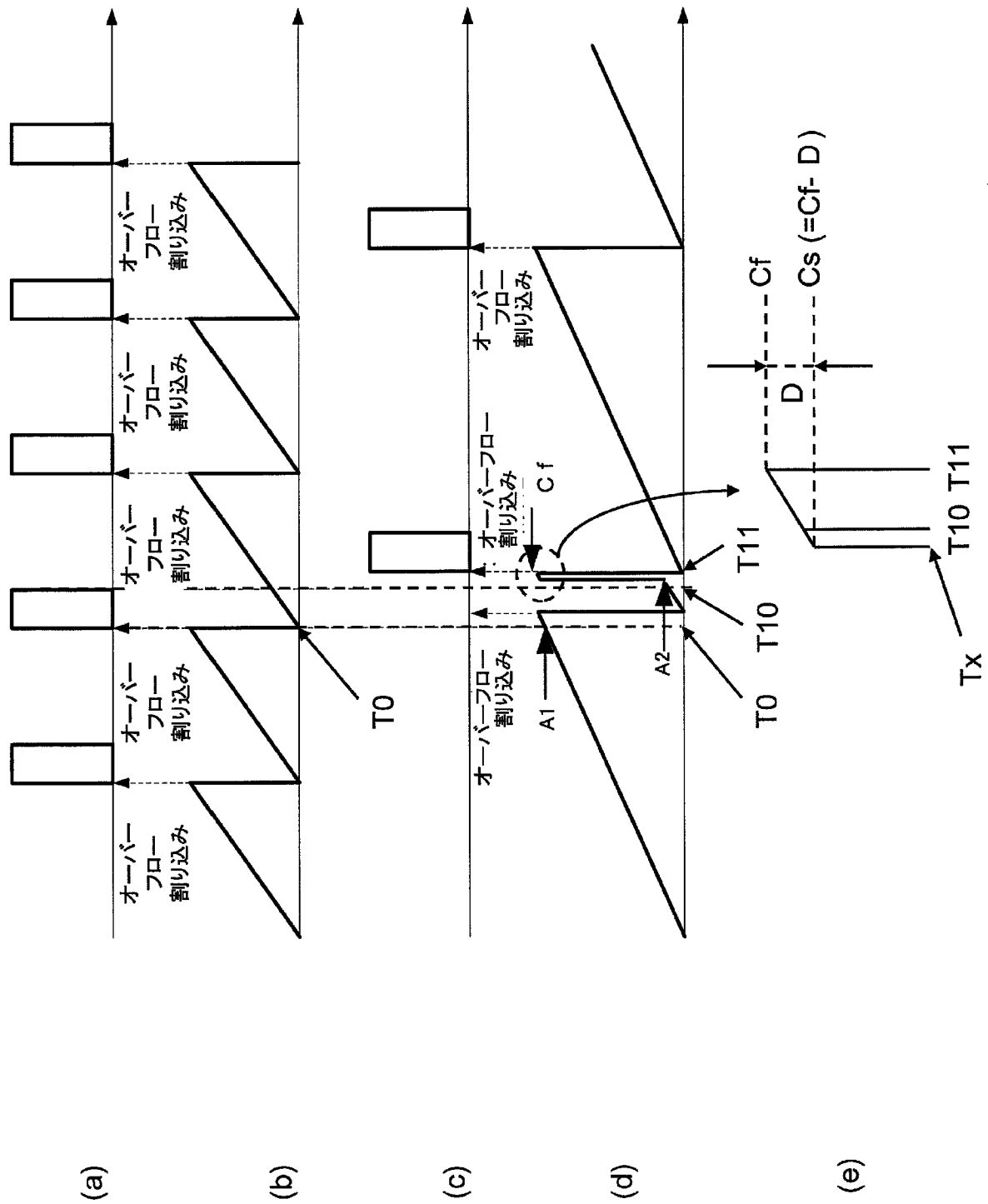
[図8]



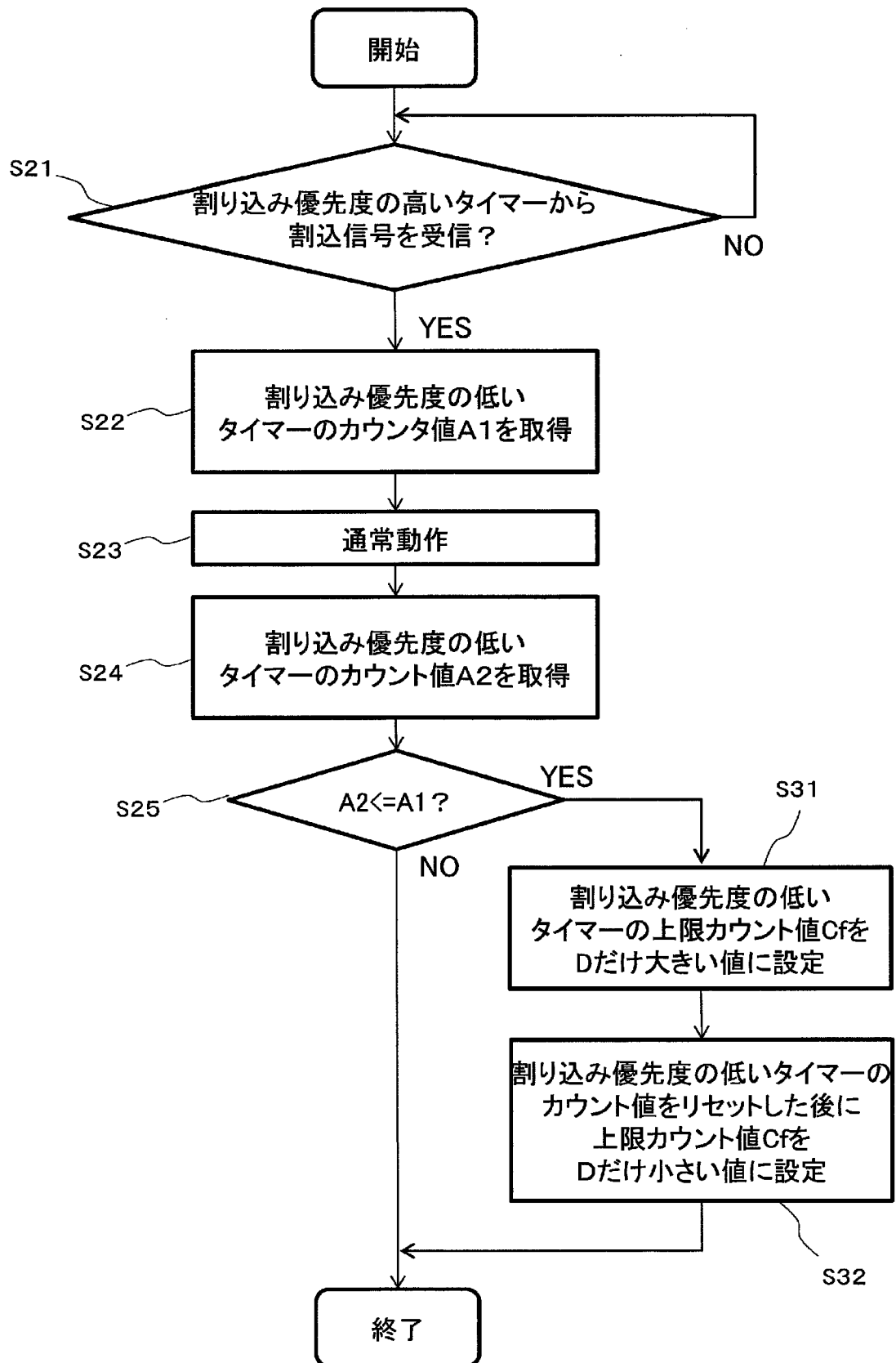
[図9]



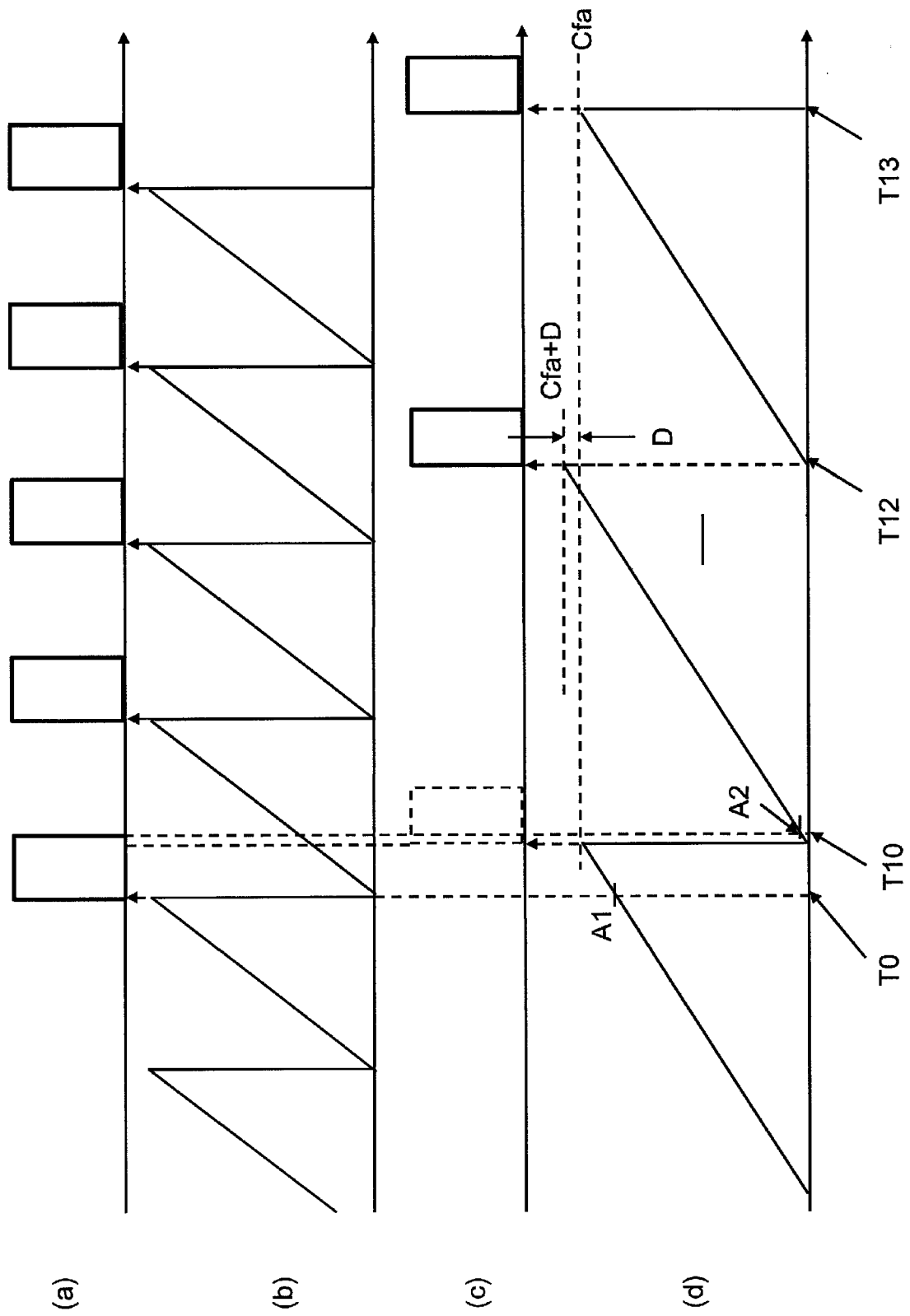
[図10]



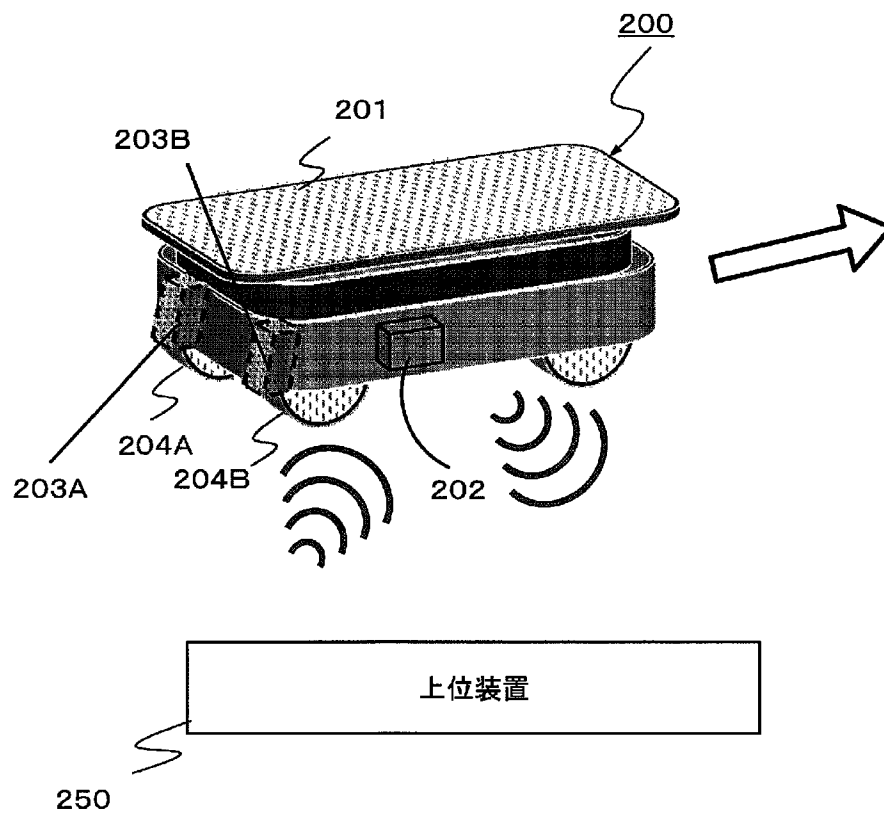
[図11]



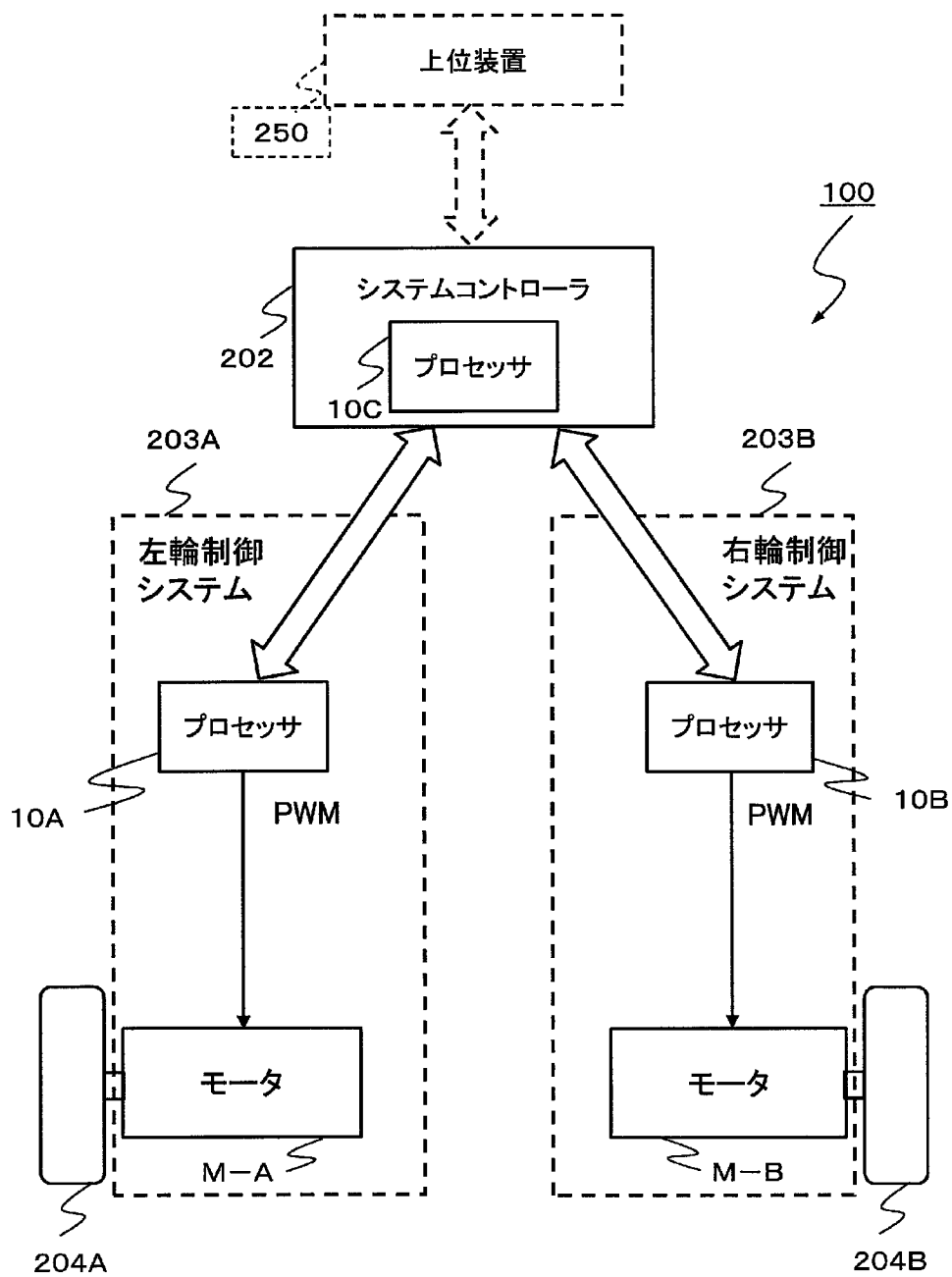
[図12]



[図13]



[図14]



INTERNATIONAL SEARCH REPORT

International application No.

PCT/JP2017/007625

A. CLASSIFICATION OF SUBJECT MATTER

H02P29/00(2016.01)i, G06F9/48(2006.01)i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

H02P29/00, G06F9/48

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Jitsuyo Shinan Koho 1922-1996 Jitsuyo Shinan Toroku Koho 1996-2017

Kokai Jitsuyo Shinan Koho 1971-2017 Toroku Jitsuyo Shinan Koho 1994-2017

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	JP 2002-199766 A (Sony Corp.), 12 July 2002 (12.07.2002), entire text; all drawings (Family: none)	1-11
A	JP 2002-272184 A (Nissan Motor Co., Ltd.), 20 September 2002 (20.09.2002), entire text; all drawings (Family: none)	1-11

☐ Further documents are listed in the continuation of Box C.

☐ See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search

18 May 2017 (18.05.17)

Date of mailing of the international search report

30 May 2017 (30.05.17)

Name and mailing address of the ISA/

Japan Patent Office

3-4-3, Kasumigaseki, Chiyoda-ku,

Tokyo 100-8915, Japan

Authorized officer

Telephone No.

A. 発明の属する分野の分類（国際特許分類（I P C））

Int.Cl. H02P29/00(2016.01)i, G06F9/48(2006.01)i

B. 調査を行った分野

調査を行った最小限資料（国際特許分類（I P C））

Int.Cl. H02P29/00, G06F9/48

最小限資料以外の資料で調査を行った分野に含まれるもの

日本国実用新案公報	1922-1996年
日本国公開実用新案公報	1971-2017年
日本国実用新案登録公報	1996-2017年
日本国登録実用新案公報	1994-2017年

国際調査で利用した電子データベース（データベースの名称、調査に使用した用語）

C. 関連すると認められる文献

引用文献の カテゴリー*	引用文献名 及び一部の箇所が関連するときは、その関連する箇所の表示	関連する 請求項の番号
A	JP 2002-199766 A（ソニー株式会社）2002. 07. 12, 全文、全図（ファミリーなし）	1-11
A	JP 2002-272184 A（日産自動車株式会社）2002. 09. 20, 全文、全図（ファミリーなし）	1-11

C欄の続きにも文献が列挙されている。

パテントファミリーに関する別紙を参照。

* 引用文献のカテゴリー

「A」特に関連のある文献ではなく、一般的技術水準を示すもの
「E」国際出願日前の出願または特許であるが、国際出願日以後に公表されたもの
「L」優先権主張に疑義を提起する文献又は他の文献の発行日若しくは他の特別な理由を確立するために引用する文献（理由を付す）
「O」口頭による開示、使用、展示等に言及する文献
「P」国際出願日前で、かつ優先権の主張の基礎となる出願

の日の後に公表された文献

「T」国際出願日又は優先日後に公表された文献であって出願と矛盾するものではなく、発明の原理又は理論の理解のために引用するもの
「X」特に関連のある文献であって、当該文献のみで発明の新規性又は進歩性がないと考えられるもの
「Y」特に関連のある文献であって、当該文献と他の1以上の文献との、当業者にとって自明である組合せによって進歩性がないと考えられるもの
「&」同一パテントファミリー文献

国際調査を完了した日

18.05.2017

国際調査報告の発送日

30.05.2017

国際調査機関の名称及びあて先

日本国特許庁（I S A/J P）

郵便番号100-8915

東京都千代田区霞が関三丁目4番3号

特許庁審査官（権限のある職員）

尾家 英樹

電話番号 03-3581-1101 内線 3357

3V

9335