

(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
2022年6月30日(30.06.2022)



(10) 国際公開番号

WO 2022/137665 A1

(51) 国際特許分類:
B60R 16/02 (2006.01) G06F 11/20 (2006.01)
G06F 11/16 (2006.01)

(21) 国際出願番号: PCT/JP2021/033207

(22) 国際出願日: 2021年9月9日(09.09.2021)

(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 2020-214322 2020年12月23日(23.12.2020) JP

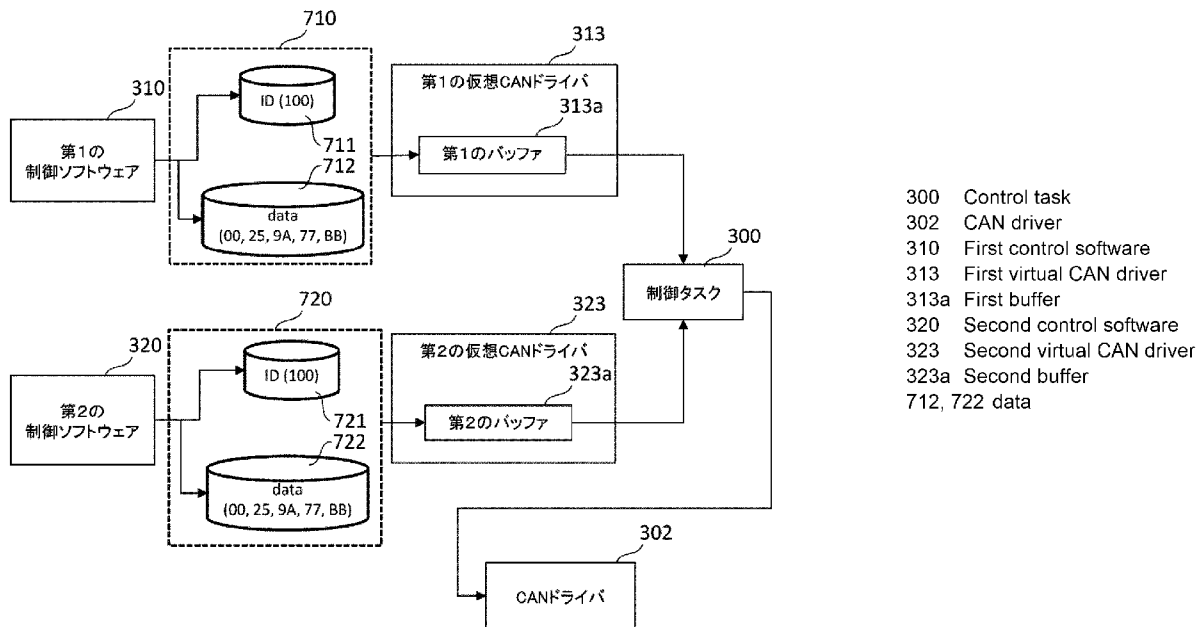
(71) 出願人: 日立 A s t e m o 株式会社(HITACHI
ASTEMO, LTD.) [JP/JP]; 〒3128503 茨城県ひ
たちなか市高場 2 5 2 0 番地 Ibaraki (JP).

(72) 発明者: 溝口 将史 (MIZOGUCHI Masashi);
〒1008280 東京都千代田区丸の内一丁目 6
番 6 号 株式会社日立製作所内 Tokyo (JP).
蛭名 朋仁 (EBINA Tomohito); 〒3128503 茨城
県ひたちなか市高場 2 5 2 0 番地 日立 A
s t e m o 株式会社内 Ibaraki (JP). 石郷岡
祐 (ISHIGOOKA Tasuku); 〒1008280 東京都千
代田区丸の内一丁目 6 番 6 号 株式会社日
立製作所内 Tokyo (JP). 芹沢 一 (SERIZAWA
Kazuyoshi); 〒3128503 茨城県ひたちなか市高
場 2 5 2 0 番地 日立 A s t e m o 株式会
社内 Ibaraki (JP). 長田 健一 (OSADA Kenichi);
〒3128503 茨城県ひたちなか市高場 2 5 2 0 番
地 日立 A s t e m o 株式会社内 Ibaraki (JP).

(54) Title: ELECTRONIC CONTROL DEVICE

(54) 発明の名称: 電子制御装置

図 7



(57) Abstract: The present invention addresses the problem of providing a technology that enables second control software to continue processing by using a peripheral circuit even in situations where an abnormality related to first control software has occurred. This electronic control device comprises: a memory (3) that stores the first control software and the second control software; a CPU (1) that executes the first control software and the second control software; and a peripheral circuit (200) that is used by the first control software and the second control software. The memory is also equipped with

(74) 代理人: 特許業務法人平木国際特許事務所
(HIRAKI & ASSOCIATES); 〒1056232 東京都
港区愛宕二丁目5-1 愛宕グリーンヒルズ
MOR I タワー32階 Tokyo (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, IT, JO, 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, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, WS, 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, 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))

a first buffer (313a) and a second buffer (323a). The CPU suspends at least one of processing for storing first peripheral transmission information in the first buffer and processing for transmitting the first peripheral transmission information to peripherals in a specific situation in which an abnormality related to the first control software has occurred.

(57) 要約: 本発明は、第1の制御ソフトウェアに関する異常が生じた状況においても、第2の制御ソフトウェアがペリフェラル回路を使用して処理を継続することが可能な技術を提供することを課題とする。電子制御装置は、第1の制御ソフトウェア及び第2の制御ソフトウェアを格納するメモリ(3)と、第1の制御ソフトウェア及び第2の制御ソフトウェアを実行するCPU(1)と、第1の制御ソフトウェア及び第2の制御ソフトウェアによって使用されるペリフェラル回路(200)と、を備える。メモリは、第1のバッファ(313a)と、第2のバッファ(323a)とを更に備える。CPUは、第1の制御ソフトウェアに関する異常が生じた特定状況において、第1のペリフェラル送信情報の第1のバッファへの格納処理及び第1のペリフェラル送信情報のペリフェラルへ送信処理の少なくとも一方を中断する。

明 細 書

発明の名称：電子制御装置

技術分野

[0001] 本開示は、電子制御装置に関する。

背景技術

[0002] マイクロコンピュータ（以下、「マイコン」と称呼する）は、CPU（Central Processing Unit）、メモリ、及び、ペリフェラル等を備える。ペリフェラルは、マイコンに様々な機能を提供するために1つ以上のペリフェラル回路を含む。特許文献1は、ペリフェラルが異常状態になった場合にペリフェラルを再起動させる装置を開示している。

先行技術文献

特許文献

[0003] 特許文献1：特開2011-138401号公報

発明の概要

発明が解決しようとする課題

[0004] 従来より、複数の制御ソフトウェアが搭載されたマイコンが知られている。サイズ及びコスト等の観点から、マイコン上に搭載されるペリフェラル回路の数に制限がある。従って、複数の制御ソフトウェアが1つのペリフェラル回路を共用する場合がある。

[0005] 上記の構成において、複数の制御ソフトウェアのうち的一方（以下、「第1の制御ソフトウェア」と称呼する）に関する異常が生じたと仮定する。このような状況においても、複数の制御ソフトウェアのうち他方（以下、「第2の制御ソフトウェア」と称呼する）が、ペリフェラル回路を使用して処理を継続することが求められる。特許文献1に記載された装置は、ペリフェラル回路を再起動する。このような構成の場合、第2の制御ソフトウェアは、ペリフェラル回路の再起動が完了するまでペリフェラル回路を使用することができない。従って、第2の制御ソフトウェアが処理を継続できないとい

う課題がある。

- [0006] そこで、本開示は、第1の制御ソフトウェアに関する異常が生じた状況においても、第2の制御ソフトウェアがペリフェラル回路を使用して処理を継続することが可能な技術を提供する。

課題を解決するための手段

- [0007] 一以上の実施例において、以下の電子制御装置が提供される。当該電子制御装置は、第1の制御ソフトウェア及び第2の制御ソフトウェアを格納するメモリと、前記第1の制御ソフトウェア及び前記第2の制御ソフトウェアを実行するCPUと、前記第1の制御ソフトウェア及び前記第2の制御ソフトウェアによって使用されるペリフェラル回路とを備える。前記メモリは、第1のバッファと、第2のバッファとを更に備える。前記CPUは、前記第1の制御ソフトウェアから前記ペリフェラル回路への送信情報である第1のペリフェラル送信情報を前記第1のバッファに格納する第1の格納処理と、前記第2の制御ソフトウェアから前記ペリフェラル回路への送信情報である第2のペリフェラル送信情報を前記第2のバッファに格納する第2の格納処理と、前記第1のバッファから前記ペリフェラル回路へ前記第1のペリフェラル送信情報を出力する第1の出力処理と、前記第2のバッファから前記ペリフェラル回路へ前記第2のペリフェラル送信情報を出力する第2の出力処理とを実行するように構成される。前記CPUは、前記第1の制御ソフトウェアに関する異常が生じた特定状況において、前記第1の格納処理及び前記第1の出力処理の少なくとも一方を中断し、前記第2の格納処理及び前記第2の出力処理を継続するように構成される。

発明の効果

- [0008] 上記の構成によれば、電子制御装置は、第1の制御ソフトウェアに関する異常が生じた特定状況においても、第2の制御ソフトウェアに関連する処理（第2の格納処理及び第2の出力処理）を継続することができる。本発明に関連する更なる特徴は、本明細書の記述、添付図面から明らかになるものである。また、上記した以外の、課題、構成及び効果は、以下の実施形態の説

明により明らかにされる。

図面の簡単な説明

[0009] [図1]電子制御装置のハードウェア構成を示した図である。

[図2]メモリに格納されているソフトウェアの一例を示した図である。

[図3]第1のMPUにおける第1のアクセス可能領域に関する設定情報の一例である。

[図4]第2のMPUにおける第2のアクセス可能領域に関する設定情報の一例である。

[図5]電子制御装置と外部機器との接続関係を示した図である。

[図6]CANドライバの動作及びCANコントローラの動作を説明するための図である。

[図7]複数のソフトウェア間の入力／出力の関係の一例を示した図である。

[図8]第1の制御ソフトウェアが第1のペリフェラル送信情報を送信する場合の処理の流れを説明するための図である。

[図9]第1のバッファのデータ構造の一例を示した図である。

[図10]第2のバッファのデータ構造の一例を示した図である。

[図11]制御タスクの内部に保持される状態遷移図である。

[図12]複数のソフトウェア間の入力／出力の関係の別の例を示した図である。

[図13]メモリに格納されているソフトウェアの別の例を示した図である。

[図14]診断情報テーブルの一例である。

[図15]初期化ポリシーテーブルの一例である。

[図16]初期化範囲テーブルの一例である。

[図17]第1のダミーバッファのデータ構造の一例を示した図である。

発明を実施するための形態

[0010] 以下、添付図面を参照して複数の実施例について説明する。添付図面は、具体的な実施例を示しているが、本発明の技術的範囲を限定的に解釈するために用いられるものではない。

[0011] 以下の複数の実施例は、それぞれ、複数の制御ソフトウェアが搭載された電子制御装置に関する。説明を簡単にするために、2つの制御ソフトウェア（第1の制御ソフトウェア及び第2の制御ソフトウェア）が搭載された電子制御装置の実施例について説明される。しかし、電子制御装置の構成は、これに限定されず、電子制御装置は、他の制御ソフトウェア（例えば、第3の制御ソフトウェア及び第4の制御ソフトウェア）を更に搭載していてもよい。

[0012] <実施例1>

（全体構成）

図1は、電子制御装置のハードウェア構成を示した図である。電子制御装置1000は、マイコン900と、電源回路901とを備える。マイコン900は、CPU1と、ペリフェラル2と、メモリ3と、バス4とを備える。CPU1、ペリフェラル2及びメモリ3は、バス4を介して互いに接続されている。従って、CPU1は、バス4を介してペリフェラル2及びメモリ3にアクセス可能である。

[0013] メモリ3は、複数のソフトウェアを格納している。CPU1は、メモリ3に格納されたソフトウェアを実行する。

[0014] マイコン900は、更に、自己リセット信号出力端子911と、外部リセット端子912とを備える。自己リセット信号出力端子911及び外部リセット端子912は、それぞれ、電源回路901に接続されている。マイコン900は、自己リセット信号出力端子911を介して電源回路901にリセット要求信号を出力する。電源回路901は、リセット要求信号に応じて、外部リセット端子912に外部リセット信号を入力する。マイコン900が外部リセット信号を受け取ると、マイコン900が再起動される。

[0015] （CPU）

CPU1は、複数のコアを備える。具体的には、CPU1は、メインCPUコア100と、第1のCPUコア101と、第2のCPUコア102とを備える。

- [0016] 第1のCPUコア101は、第1のMPU (Memory Protection Unit) 101aを含む。第1のMPU 101aは、第1のCPUコア101がアクセス可能なメモリ3の領域（以下、「第1のアクセス可能領域」と称する）に関する設定情報を含む。第1のMPU 101aは、第1のCPUコア101が第1のアクセス可能領域にアクセスすることを許可する。第1のMPU 101aは、第1のCPUコア101が第1のアクセス可能領域以外の領域にアクセスすることを禁止する。
- [0017] 第2のCPUコア102は、第2のMPU 102aを含む。第2のMPU 102aは、第2のCPUコア102がアクセス可能なメモリ3の領域（以下、「第2のアクセス可能領域」と称する）に関する設定情報を含む。第2のMPU 102aは、第2のCPUコア102が第2のアクセス可能領域にアクセスすることを許可する。第2のMPU 102aは、第2のCPUコア102が第2のアクセス可能領域以外の領域にアクセスすることを禁止する。
- [0018] メインCPUコア100は、メインOS (Operating System) 301（図2を参照）に関する処理を実行する。第1のCPUコア101は、第1の制御ソフトウェア310（図2を参照）に関する処理を実行する。第2のCPUコア102は、第2の制御ソフトウェア320（図2を参照）に関する処理を実行する。
- [0019] CPU1は上記の構成に限定されない。CPU1は4つ以上のCPUコアを備えてもよい。例えば、CPU1は、複数の第1のCPUコア101を備えてもよい。この場合、複数の第1のCPUコア101のそれぞれが、第1のMPU 101aを含む。例えば、CPU1は、複数の第2のCPUコア102を備えてもよい。この場合、複数の第2のCPUコア102のそれぞれが、第2のMPU 102aを含む。
- [0020] (ペリフェラル)
- ペリフェラル2は、複数のペリフェラル回路を備える。ペリフェラル回路とは、マイコン900に搭載された構成要素のうち、CPU1及びメモリ3

以外の回路を指す。ペリフェラル回路の例として、例えば、CAN (Controller Area Network) コントローラ、アナログ・デジタル・コンバータ (A/Dコンバータ)、ダイレクト・メモリ・アクセスコントローラ、及び、GPIO (General Purpose Input/Output) 等が挙げられる。

[0021] 本例において、ペリフェラル2は、共用ペリフェラル回路200と、第1のペリフェラル回路201と、第2のペリフェラル回路202とを備える。共用ペリフェラル回路200は、CANコントローラ200aを含む。第1のペリフェラル回路201は、アナログ用の第1のGPIOピン201aを含む。第2のペリフェラル回路202は、デジタル用の第2のGPIOピン202aを含む。

[0022] 共用ペリフェラル回路200は、第1の制御ソフトウェア310 (即ち、第1のCPUコア101) 及び第2の制御ソフトウェア320 (即ち、第2のCPUコア102) の両方によって使用されるペリフェラル回路である。なお、共用ペリフェラル回路200は、メインCPUコア100からもアクセスされる。

[0023] 第1のペリフェラル回路201は、第1の制御ソフトウェア310 (即ち、第1のCPUコア101) のみによって使用されるペリフェラル回路である。第2のペリフェラル回路202は、第2の制御ソフトウェア320 (即ち、第2のCPUコア102) のみによって使用されるペリフェラル回路である。

[0024] (ソフトウェア)

図2は、メモリ3に格納されているソフトウェアを示した図である。メモリ3は、複数のソフトウェアを格納している。メモリ3は、制御タスク300と、メインOS301と、CANDライバ302とを格納している。メインCPUコア100は、これらのソフトウェア300~302を実行する。

[0025] メインCPUコア100によって実行される各ソフトウェアについて説明する。以降の説明において、ソフトウェア300~302が主語として記載される。しかし、ソフトウェア300~302を実行する主体はメインCP

Ｕコア１００であるので、主語がメインＣＰＵコア１００に置き換えられてもよい。

[0026] メインＯＳ３０１は、ＣＡＮドライバ３０２を初期化する。ＣＡＮドライバ３０２は、共用ペリフェラル回路２００に対応するソフトウェアであり、ＣＡＮコントローラ２００ａに動作命令を出力するソフトウェアである。

[0027] 更に、メインＯＳ３０１は、第１のサブＯＳ３１１及び第２のサブＯＳ３２１を起動する。即ち、メインＯＳ３０１は、第１のサブＯＳ３１１の起動命令を第１のＣＰＵコア１０１に送信し、第２のサブＯＳ３２１の起動命令を第２のＣＰＵコア１０２に送信する。

[0028] 更に、メインＯＳ３０１は、１ｍｓの周期で制御タスク３００を起動する。本例では、制御タスク３００の実行周期が１ｍｓであるが、これに限定されない。制御タスク３００の詳細な処理内容は後述する。

[0029] メモリ３は、更に、第１の制御ソフトウェア３１０と、第１のサブＯＳ３１１と、第１のＧＰＩＯピンドライバ３１２と、第１の仮想ＣＡＮドライバ３１３とを格納している。第１のＣＰＵコア１０１は、これらのソフトウェア３１０～３１３を実行する。

[0030] 第１のＣＰＵコア１０１によって実行される各ソフトウェアについて説明する。以降の説明において、ソフトウェア３１０～３１３が主語として記載される。しかし、ソフトウェア３１０～３１３を実行する主体は第１のＣＰＵコア１０１であるので、主語が第１のＣＰＵコア１０１に置き換えられてもよい。

[0031] 第１のサブＯＳ３１１は、メインＯＳ３０１からの起動命令に従い、第１のＧＰＩＯピンドライバ３１２を初期化する。第１のＧＰＩＯピンドライバ３１２は、第１のＧＰＩＯピン２０１ａに動作命令を出力するソフトウェアである。

[0032] 第１のサブＯＳ３１１は、メインＯＳ３０１からの起動命令に従い、第１の制御ソフトウェア３１０を起動する。第１の制御ソフトウェア３１０は、１つ以上のタスクを含む。第１の制御ソフトウェア３１０のタスクは、第１

のサブOS 311によって所定のタイミングで実行される。本例において、第1の制御ソフトウェア310は、所定のタイミングで第1のペリフェラル送信情報を送信する。第1のペリフェラル送信情報は、CANコントローラ200aへ向けて送信される情報である。

[0033] 更に、第1のサブOS 311は、メインOS 301からの起動命令に従い、第1の仮想CANDライバ313を初期化する。第1の仮想CANDライバ313は、第1の制御ソフトウェア310とCANDライバ302との間に介在するソフトウェアであって、第1のペリフェラル送信情報を処理するソフトウェアである。具体的には、第1の仮想CANDライバ313は、第1のバッファ313aを含む。第1の仮想CANDライバ313は、第1のペリフェラル送信情報を第1のバッファ313aに格納する。

[0034] メモリ3は、更に、第2の制御ソフトウェア320と、第2のサブOS 321と、第2のGPIOピンドライバ322と、第2の仮想CANDライバ323とを格納している。第2のCPUコア102は、これらのソフトウェア320～323を実行する。

[0035] 第2のCPUコア102によって実行されるソフトウェアについて説明する。以降の説明において、ソフトウェア320～323が主語として記載される。しかし、ソフトウェア320～323を実行する主体は第2のCPUコア102であるので、主語が第2のCPUコア102に置き換えられてもよい。

[0036] 第2のサブOS 321は、メインOS 301からの起動命令に従い、第2のGPIOピンドライバ322を初期化する。第2のGPIOピンドライバ322は、第2のGPIOピン202aに動作命令を出力するソフトウェアである。

[0037] 第2のサブOS 321は、メインOS 301からの起動命令に従い、第2の制御ソフトウェア320を起動する。第2の制御ソフトウェア320は、1つ以上のタスクを含む。第2の制御ソフトウェア320のタスクは、第2のサブOS 321によって所定のタイミングで実行される。本例において、

第2の制御ソフトウェア320は、所定のタイミングで第2のペリフェラル送信情報を送信する。第2のペリフェラル送信情報は、CANコントローラ200aへ向けて送信される情報である。

[0038] 更に、第2のサブOS321は、メインOS301からの起動命令に従い、第2の仮想CANDライバ323を初期化する。第2の仮想CANDライバ323は、第2の制御ソフトウェア320とCANDライバ302との間に介在するソフトウェアであって、第2のペリフェラル送信情報を処理するソフトウェアである。具体的には、第2の仮想CANDライバ323は、第2のバッファ323aを含む。第2の仮想CANDライバ323は、第2のペリフェラル送信情報を第2のバッファ323aに格納する。

[0039] (第1のMPUにおける設定情報)

図3は、第1のMPU101aにおける第1のアクセス可能領域に関する設定情報を示した図である。第1のCPUコア101は、ソフトウェア310～313の格納領域に対して読み取り権及び書き込み権を有する。第1のCPUコア101は、ソフトウェア310～313の格納領域以外の領域に対して読み取り権及び書き込み権を有していない。これにより、第1の制御ソフトウェア310（即ち、第1のCPUコア101）が、メインOS301及び第2の制御ソフトウェア320に影響を与えることを防止できる。

[0040] (第2のMPUにおける設定情報)

図4は、第2のMPU102aにおける第2のアクセス可能領域に関する設定情報を示した図である。第2のCPUコア102は、ソフトウェア320～323の格納領域に対して読み取り権及び書き込み権を有する。第2のCPUコア102は、ソフトウェア320～323の格納領域以外の領域に対して読み取り権及び書き込み権を有していない。これにより、第2の制御ソフトウェア320（即ち、第2のCPUコア102）が、メインOS301及び第1の制御ソフトウェア310に影響を与えることを防止できる。

[0041] なお、メインCPUコア100は、メモリ3の全ての領域にアクセスすることができる。

[0042] (外部の制御装置との接続構成)

図5は、電子制御装置1000と外部機器との接続関係を示した図である。以下では、第1の制御ソフトウェア310及び第2の制御ソフトウェア320のそれぞれの処理内容について説明する。

[0043] ・第1の制御ソフトウェア

複数のセンサ511及び複数のアクチュエータ512は、電子制御装置1000の外部に配置されている。複数のセンサ511及び複数のアクチュエータ512は、それぞれ、第1のGPIOピン201aに接続されている。

[0044] 第1の制御ソフトウェア310は、第1のGPIOピン201aにアクセスして、複数のアクチュエータ512のそれぞれに対してアナログ信号を入力する。更に、第1の制御ソフトウェア310は、第1のGPIOピン201aを介して、複数のセンサ511のそれぞれからアナログ信号を受け取る。第1の制御ソフトウェア310は、これらのアナログ信号に基づいて、第1のペリフェラル送信情報を送信する。

[0045] 本例において、第1のペリフェラル送信情報は、第1の制御ソフトウェア310から他の電子制御装置540のそれぞれに対して送信されるCANメッセージである。なお、第1のペリフェラル送信情報は、後述するように、第1のバッファ313aを介して、CANコントローラ200aに出力される。CANコントローラ200aは、第1のペリフェラル送信情報を受け取る。CANコントローラ200aは、第1のペリフェラル送信情報を物理的信号に変換し、当該物理的信号をCANバス530を介して他の電子制御装置540に対して送信する。

[0046] ・第2の制御ソフトウェア

複数のセンサ521及び複数のアクチュエータ522は、電子制御装置1000の外部に配置されている。複数のセンサ521及び複数のアクチュエータ522は、それぞれ、第2のGPIOピン202aに接続されている。

[0047] 第2の制御ソフトウェア320は、第2のGPIOピン202aにアクセスして、複数のアクチュエータ522のそれぞれに対してデジタル信号を入

力する。更に、第2の制御ソフトウェア320は、第2のGPIOピン202aを介して、複数のセンサ521のそれぞれからデジタル信号を受け取る。第2の制御ソフトウェア320は、これらのデジタル信号に基づいて、第2のペリフェラル送信情報を送信する。

[0048] 本例において、第2のペリフェラル送信情報は、第2の制御ソフトウェア320から他の電子制御装置540のそれぞれに対して送信されるCANメッセージである。なお、第2のペリフェラル送信情報は、後述するように、第2のバッファ323aを介して、CANコントローラ200aに出力される。CANコントローラ200aは、第2のペリフェラル送信情報を受け取る。CANコントローラ200aは、第2のペリフェラル送信情報を物理的信号に変換し、当該物理的信号をCANバス530を介して他の電子制御装置540に対して送信する。

[0049] (CANDライバ及びCANコントローラの動作)

図6は、CANDライバ302の動作及びCANコントローラ200aの動作の概要を示した図である。以下では、第1の制御ソフトウェア310が第1のペリフェラル送信情報を送信した場合について説明する。CANDライバ302は、CANコントローラ200aの内部レジスタ200bの所定の場所に第1のペリフェラル送信情報を格納する。CANコントローラ200aは、第1のペリフェラル送信情報を所定のCANプロトコルに従いCANフレーム600に格納する。そして、CANコントローラ200aは、CANフレーム600を物理的信号に変換してCANバス530に送信する。

[0050] なお、第2の制御ソフトウェア320が第2のペリフェラル送信情報を送信した場合についても、CANDライバ302及びCANコントローラ200aは、上記と同じように動作する。

[0051] (ソフトウェアの動作)

図7は、複数のソフトウェア間の入力／出力の関係を示した図である。本例によれば、CANDライバ302の前段に、2つの仮想的なCANDライバ(即ち、第1の仮想CANDライバ313及び第2の仮想CANDライバ

323)が存在する。第1の制御ソフトウェア310及び第2の制御ソフトウェア320が、それぞれ、第1の仮想CANドライバ313及び第2の仮想CANドライバ323を使用する。制御タスク300は、第1の仮想CANドライバ313及び第2の仮想CANドライバ323に交互にアクセスして、第1のペリフェラル送信情報710及び第2のペリフェラル送信情報720を交互にCANドライバ302に出力する。これにより、第1の制御ソフトウェア310の処理と第2の制御ソフトウェア320の処理とを分離しながら、1つのCANドライバ302を使用することができる。

[0052] 具体的には、第1の制御ソフトウェア310は、第1のペリフェラル送信情報710を第1の仮想CANドライバ313に送信する。第1の仮想CANドライバ313は、第1のペリフェラル送信情報710を第1のバッファ313aに格納する。

[0053] 第2の制御ソフトウェア320は、第2のペリフェラル送信情報720を第2の仮想CANドライバ323に送信する。第2の仮想CANドライバ323は、第2のペリフェラル送信情報720を第2のバッファ323aに格納する。

[0054] 制御タスク300は、第1のバッファ313aから第1のペリフェラル送信情報710を取り出し、第1のペリフェラル送信情報710をCANドライバ302に出力する。

[0055] 制御タスク300は、第2のバッファ323aから第2のペリフェラル送信情報720を取り出し、第2のペリフェラル送信情報720をCANドライバ302に出力する。

[0056] 以下、図7に示した各種ソフトウェアの具体的な処理内容について説明する。

[0057] ・第1の制御ソフトウェア

図8は、第1の制御ソフトウェア310が第1のペリフェラル送信情報710を送信する場合の処理の流れを説明する図である。第1の制御ソフトウェア310は、第1のペリフェラル送信情報710として、CANメッセー

ジID及びデータを指定する。第1の制御ソフトウェア310は、CAN_send_virtio1(ID, data)をコールする。CAN_send_virtio1(ID, data)は、第1のペリフェラル送信情報710を第1の仮想CANドライバ313へ送信するコマンドである。第1のペリフェラル送信情報710は、CANメッセージID711及びデータ712を含む。

[0058] ・第1の仮想CANドライバ

第1の制御ソフトウェア310がCAN_send_virtio1(ID, data)をコールすると、第1の仮想CANドライバ313は、第1のペリフェラル送信情報710を受け取る。第1の仮想CANドライバ313は、第1のペリフェラル送信情報710を第1のバッファ313aに格納する。

[0059] ・第2の制御ソフトウェア

第2の制御ソフトウェア320が第2のペリフェラル送信情報720を送信する場合についても同様である。第2の制御ソフトウェア320は、第2のペリフェラル送信情報720として、CANメッセージID及びデータを指定する。第2の制御ソフトウェア320は、CAN_send_virtio2(ID, data)をコールする。CAN_send_virtio2(ID, data)は、第2のペリフェラル送信情報720を第2の仮想CANドライバ323へ送信するコマンドである。図7に示すように、第2のペリフェラル送信情報720は、CANメッセージID721及びデータ722を含む。

[0060] ・第2の仮想CANドライバ

第2の制御ソフトウェア320がCAN_send_virtio2(ID, data)をコールすると、第2の仮想CANドライバ323は、第2のペリフェラル送信情報720を受け取る。第2の仮想CANドライバ323は、第2のバッファ323aに第2のペリフェラル送信情報720を格納する。

[0061] ・第1のバッファ

図9は、第1のバッファ313aのデータ構造を示した図である。第1のバッファ313aは、First-In-First-Outキューの実装の1つであるリングバッファ構造を有する。第1のバッファ313aの各要素は、CANメッセ

ージIDとデータの組からなる構造体である。リングバッファの実体は、構造体配列buffer_virtio1[12]である。なお、ここでは説明の都合上、配列の要素数を12としているが、これに限定されない。

[0062] 第1の制御ソフトウェア310がCAN_send_virtio1(ID, data)をコールすると、第1の仮想CANドライバ313は、バッファ終端位置を表すインデックスbackwardを以下のように演算する ($\text{backward} \leftarrow (\text{backward} + 1) \% 12$)。ここで、「%」は、(backward + 1)を12で割ったときの余りを求める演算子である。なお、インデックスbackwardが配列終端 (backward = 11) である場合、第1の仮想CANドライバ313は、backwardに「0」を代入する。第1の仮想CANドライバ313は、第1のペリフェラル送信情報710をbuffer_virtio1[backward]に格納する。このように、第1の仮想CANドライバ313は、第1のペリフェラル送信情報710を第1のバッファ313aの最後尾に格納する。以降において、このような第1のペリフェラル送信情報710の格納処理は、「第1の格納処理」と称呼される。

[0063] ・第2のバッファ

図10は、第2のバッファ323aのデータ構造を示した図である。第2のバッファ323aは、First-In-First-Outキューの実装の1つであるリングバッファ構造を有する。第2のバッファ323aの各要素はCANメッセージIDとデータの組からなる構造体である。リングバッファの実体は、構造体配列buffer_virtio2[12]である。なお、ここでは説明の都合上、配列の要素数を12としているが、これに限定されない。

[0064] 第2の制御ソフトウェア320がCAN_send_virtio2(ID, data)をコールすると、第2の仮想CANドライバ323は、バッファ終端位置を表すインデックスbackwardを以下のように演算する ($\text{backward} \leftarrow (\text{backward} + 1) \% 12$)。なお、インデックスbackwardが配列終端 (backward = 11) である場合、第2の仮想CANドライバ323は、backwardに「0」を代入する。第2の仮想CANドライバ323は、第2のペリフェラル送信情報720をbuffer_virtio2[backward]に格納する。このように、第2の仮想CANドライバ323

は、第2のペリフェラル送信情報720を第2のバッファ323aの最後尾に格納する。以降において、このような第2のペリフェラル送信情報720の格納処理は、「第2の格納処理」と称呼される。

[0065] ・制御タスク

制御タスク300は、第1のバッファ313aから第1のペリフェラル送信情報710を取り出す。図9に示すように、制御タスク300は、バッファ先頭位置を示すインデックスforwardに含まれたデータbuffer_virtio1[forward]を、第1のペリフェラル送信情報710として取り出す。そして、制御タスク300は、インデックスforwardを以下のように演算する ($\text{forward} \leftarrow (\text{forward} + 1) \% 12$)。なお、インデックスforwardが配列終端 ($\text{forward} = 11$) である場合、制御タスク300は、インデックスforwardに「0」を代入する。制御タスク300は、第1のペリフェラル送信情報710をCANドライバ302に出力する。以降において、このような第1のペリフェラル送信情報710の出力処理は、「第1の出力処理」と称呼される。

[0066] 制御タスク300は、第2のバッファ323aから第2のペリフェラル送信情報720を取り出す。図10に示すように、制御タスク300は、バッファ先頭位置を示すインデックスforwardに含まれたデータbuffer_virtio2[forward]を、第2のペリフェラル送信情報720として取り出す。そして、制御タスク300は、インデックスforwardを以下のように演算する ($\text{forward} \leftarrow (\text{forward} + 1) \% 12$)。なお、インデックスforwardが配列終端 ($\text{forward} = 11$) である場合、制御タスク300は、インデックスforwardに「0」を代入する。制御タスク300は、第2のペリフェラル送信情報720をCANドライバ302に出力する。以降において、このような第2のペリフェラル送信情報720の出力処理は、「第2の出力処理」と称呼される。

[0067] 図11は、制御タスク300の内部に保持される状態遷移図である。制御タスク300は、上述のように、メインOS301によって1msの周期で起動されるタスクである。制御タスク300は、第1の出力処理及び第2の出力処理を交互に実行する。

- [0068] 例えば、いま、制御タスク300が第1状態St1であると仮定する。この場合、制御タスク300は、第1の出力処理を実行する。
- [0069] 制御タスク300は、第1の出力処理を実行した後に、第2のバッファ323aが空であるか否かを判定する。第2のバッファ323aが空でない場合は、制御タスク300は、第1状態St1から第2状態St2へと遷移する。一方、第2のバッファ323aが空であり且つ第1のバッファ313aが空でない場合（即ち、第2のバッファ323aのみが空である場合）、制御タスク300は第1状態St1を維持する。この場合、制御タスク300は、第1の出力処理を繰り返し実行する。
- [0070] なお、制御タスク300が第1の出力処理を実行した後に、第1のバッファ313a及び第2のバッファ323aの両方が空である場合、制御タスク300は、第1状態St1から第3状態St3へと遷移する。この場合、制御タスク300は、何も動作をしない。その後、第1のバッファ313aが空でない場合（即ち、第1のペリフェラル送信情報710が第1のバッファ313aに格納された場合）、制御タスク300は、第3状態St3から第1状態St1へと遷移する。或いは、第2のバッファ323aが空でない場合（即ち、第2のペリフェラル送信情報720が第2のバッファ323aに格納された場合）、制御タスク300は、第3状態St3から第2状態St2へと遷移する。
- [0071] 制御タスク300が、第1状態St1から第2状態St2に遷移したと仮定する。この場合、制御タスク300は、第2の出力処理を実行する。
- [0072] 制御タスク300は、第2の出力処理を実行した後に、第1のバッファ313aが空であるか否かを判定する。第1のバッファ313aが空でない場合は、制御タスク300は、第2状態St2から第1状態St1へと遷移する。一方、第1のバッファ313aが空であり且つ第2のバッファ323aが空でない場合（即ち、第1のバッファ313aのみが空である場合）、制御タスク300は第2状態St2を維持する。この場合、制御タスク300は、第2の出力処理を繰り返し実行する。

[0073] なお、制御タスク300が第2の出力処理を実行した後に、第1のバッファ313a及び第2のバッファ323aの両方が空である場合、制御タスク300は、第1状態St1から第3状態St3へと遷移する。この場合、制御タスク300は、何も動作をしない。その後、第1のバッファ313aが空でない場合（即ち、第1のペリフェラル送信情報710が第1のバッファ313aに格納された場合）、制御タスク300は、第3状態St3から第1状態St1へと遷移する。或いは、第2のバッファ323aが空でない場合（即ち、第2のペリフェラル送信情報720が第2のバッファ323aに格納された場合）、制御タスク300は、第3状態St3から第2状態St2へと遷移する。

[0074] （本実施例の特徴）

電子制御装置1000において、第1の制御ソフトウェア310に関する異常が生じ得る。以下、このような状況は、「特定状況」と称呼される。特定状況は、例えば、第1の制御ソフトウェア310に異常が生じた状況、並びに、第1の制御ソフトウェア310に関するソフトウェア及びハードウェアに異常が生じた状況を含む。

[0075] 本例において、特定状況は、以下の（1）～（3）のうちの少なくとも1つが生じた状況を意味する。

（1）第1の制御ソフトウェア310が通常の処理以外の処理（所謂、例外処理）を実行する。ここで、通常の処理は、第1のペリフェラル送信情報710の送信処理である。例外処理は、例えば、第1のアクセス可能領域以外へのアクセス（領域外アクセス）、及び、デッドラインミスに関する処理等を含む。

（2）第1の制御ソフトウェア310に関連するソフトウェア（例えば、第1のGPIOピンドライバ312）に異常が生じた。

（3）第1の制御ソフトウェア310に関連するハードウェアに異常が生じた。このようなハードウェアとして、（a）CPU1及びメモリ3、（b）第1のGPIOピン201aに接続されたセンサ511及びアクチュエー

タ522、並びに、(c)第1のペリフェラル送信情報710の送信先である共用ペリフェラル回路200等が挙げられる。

[0076] 第1の制御ソフトウェア310は、特定状況において、第1の中断指令を第1の仮想CANドライバ313に送信する。第1の中断指令は、第1の格納処理を中断する指令、及び、第1のバッファ313aを初期化する指令を含む。第1の仮想CANドライバ313は、第1の中断指令に応じて、第1の格納処理を中断する（禁止する）。更に、第1の仮想CANドライバ313は、第1の中断指令に応じて、第1のバッファ313aを初期化する（即ち、第1のバッファ313aを空にする）。第1の仮想CANドライバ313は、バッファ終端位置を示すインデックスbackwardにバッファ先頭位置を示すインデックスforwardを代入することにより、第1のバッファ313aを初期化することができる。

[0077] このように、第1の仮想CANドライバ313は、第1の中断指令に応じて、第1の格納処理を中断し、第1のバッファ313aを初期化する。第1のバッファ313aが空になるので、制御タスク300は、第2状態St2を維持する（図11を参照）。他の言い方をすれば、制御タスク300は、第1の出力処理を中断する。一方、制御タスク300は、第2の格納処理及び第2の出力処理を継続する。

[0078] 本実施例によれば、CANドライバ302の前段に、2つの仮想的なCANドライバ313及び323が存在する。第1の制御ソフトウェア310及び第2の制御ソフトウェア320が、それぞれ、第1の仮想CANドライバ313及び第2の仮想CANドライバ323を使用する。制御タスク300は、第1の仮想CANドライバ313及び第2の仮想CANドライバ323に交互にアクセスして、第1のペリフェラル送信情報710及び第2のペリフェラル送信情報720を交互にCANドライバ302に出力する。このように、第1の制御ソフトウェア310の処理と第2の制御ソフトウェア320の処理とが分離される。従って、第1の制御ソフトウェア310に関する異常が生じた特定状況においても、第2の制御ソフトウェア320が、第2

の格納処理及び第2の出力処理を継続できる。即ち、第2の制御ソフトウェア320が、第2のペリフェラル送信情報720のCANコントローラ200aへの送信を継続することができる。

[0079] 更に、特定状況において、不適切な第1のペリフェラル送信情報710がCANドライバ302に出力される可能性がある。第1の仮想CANドライバ313は、第1のバッファ313aを初期化するので、上記のような問題が生じるのを防ぐことができる。

[0080] 本例においては、第1のCPUコア101が第1の制御ソフトウェア310に関する処理を実行し、第2のCPUコア102が第2の制御ソフトウェア320に関する処理を実行する。従って、2つのCPUコア101及び102が互いに独立して処理を継続できる。一方のCPUコアによって実行される処理が、他方のCPUコアによって実行される処理に影響を与えることを防ぐことができる。

[0081] なお、第2の制御ソフトウェア320に関する異常が生じた場合でも同様の処理が実行される。第2の制御ソフトウェア320は、第2の中断指令を第2の仮想CANドライバ323に送信する。第2の中断指令は、第2の格納処理を中断する指令、及び、第2のバッファ323aを初期化する指令を含む。第2の仮想CANドライバ323は、第2の中断指令に応じて、第2の格納処理を中断する（禁止する）。更に、第2の仮想CANドライバ323は、第2のバッファ323aを初期化する。第2の仮想CANドライバ323は、バッファ終端位置を示すインデックスbackwardにバッファ先頭位置を示すインデックスforwardを代入することにより、第2のバッファ323aを初期化することができる。

[0082] <実施例2>

実施例2における電子制御装置1000の構成について説明する。以下では、実施例1との違いを中心に説明する。図12は、本例における複数のソフトウェア間の入力／出力の関係を示した図である。制御タスク300は、第1の格納処理、第2の格納処理、第1の出力処理及び第2の出力処理の全

てを実行するように構成される。

[0083] 制御タスク３００は、第１の制御ソフトウェア３１０から第１のペリフェラル送信情報７１０を受け取り、第１の格納処理を実行する。更に、制御タスク３００は、第２の制御ソフトウェア３２０から第２のペリフェラル送信情報７２０を受け取り、第２の格納処理を実行する。

[0084] 制御タスク３００は、図１１の状態遷移図に従って、第１の出力処理及び第２の出力処理を実行する。

[0085] 第１の制御ソフトウェア３１０は、特定状況において、第３の中断指令を制御タスク３００に送信する。第３の中断指令は、第１の格納処理を中断する指令のみを含む。制御タスク３００は、第３の中断指令に応じて、第１の格納処理を中断する。制御タスク３００は、第１のバッファ３１３ａが空になるまで第１の出力処理を継続する。第１のバッファ３１３ａが空になると、制御タスク３００は、第１の出力処理を中断する。一方、制御タスク３００は、第２の格納処理及び第２の出力処理を継続する。

[0086] 上記の構成によれば、特定状況においても、第２の制御ソフトウェア３２０が、第２の格納処理及び第２の出力処理を継続できる。

[0087] 別の例において、第３の中断指令は、第１の出力処理を中断する指令のみを含む。制御タスク３００は、第３の中断指令に応じて、第１の出力処理を中断する。制御タスク３００は、第１の格納処理のみを継続してもよい。制御タスク３００は、第１のバッファ３１３ａの配列の要素の全てが埋まるまで、第１の格納処理を継続してもよい。第１のバッファ３１３ａの配列の要素の全てが埋まると、制御タスク３００は、第１の格納処理を中断する。一方、制御タスク３００は、第２の格納処理及び第２の出力処理を継続する。

[0088] 更に別の例において、制御タスク３００は、特定状況において第１の格納処理及び第１の出力処理の両方を中断してもよい。従って、第１のバッファ３１３ａ内の第１のペリフェラル送信情報７１０が保持（ホールド）される。

[0089] <実施例３>

実施例3における電子制御装置1000の構成について説明する。以下では、実施例1との違いを中心に説明する。図13に示すように、本例において、メモリ3は、初期化タスク304と、診断タスク305とを更に格納している。メインCPUコア100は、これらのソフトウェア304及び305を更に実行する。実施例1と同様の構成については、同一の符号を付してその説明を省略する。

[0090] メインOS301は、CANドライバ302を初期化する。メインOS301は、1msの周期で制御タスク300及び初期化タスク304を起動する。本例では、制御タスク300及び初期化タスク304の実行周期が1msであるが、これに限定されない。

[0091] 更に、メインOS301は、10msの周期で診断タスク305を起動する。本例では、診断タスク305の実行周期が10msであるが、これに限定されない。

[0092] (診断情報テーブル)

図14は、診断タスク305が生成する診断情報テーブル1400の例である。診断情報テーブル1400は、構成項目として、診断対象1401と、状態1402とを含む。診断対象1401は、診断の対象となるハードウェア又はソフトウェアである。状態1402は、診断対象1401が正常な状態又は異常な状態であるかを表す情報である。

[0093] 診断タスク305は、10msの周期で、診断情報テーブル1400の診断対象1401のそれぞれが正常に動作しているかを監視する。それに応じて、診断タスク305は、診断情報テーブル1400の状態1402を更新する。

[0094] 例えば、診断タスク305は、電子制御装置1000のハードウェアが正常に動作しているかを確認する。ここでのハードウェアは、メインCPUコア100、第1のCPUコア101、第2のCPUコア102、CANコントローラ200a、第1のGPIOピン201a及び第2のGPIOピン202aを含む。

- [0095] 更に、診断タスク305は、第1のアクセス可能領域（第1のCPUコア101がアクセスできるメモリ3の領域）及び第2のアクセス可能領域（第2のCPUコア102がアクセスできるメモリ3の領域）にメモリエラーが発生していないかを確認する。メモリエラーは、パリティチェック及びCRC（Cyclic Redundancy Check）等の方法で検出されてもよいし、これ以外の方法で検出されてもよい。
- [0096] 診断タスク305は、第1の制御ソフトウェア310が正常に動作しているかを確認する。診断タスク305は、第1の制御ソフトウェア310が例外処理を実行している場合、第1の制御ソフトウェア310に異常が生じていると判定してもよい。更に、診断タスク305は、第2の制御ソフトウェア320が正常に動作しているかを確認する。
- [0097] 診断タスク305は、診断対象1401が正常に動作していると判定した場合、対応する状態1402を「正常」に設定する。診断タスク305は、診断対象1401に異常が生じていると判定した場合、対応する状態1402を「異常」に設定する。これにより、初期化タスク304は、診断情報テーブル1400を参照することにより、診断対象1401のそれぞれが正常に動作しているか否かを確認することができる。
- [0098] なお、診断情報テーブル1400は、上記の例に限定されない。診断タスク305は、第1のサブOS311及び第2のサブOS321がそれぞれ正常に動作しているかを確認してもよい。
- [0099] 図15は、初期化タスク304が保持する初期化ポリシーテーブル1500の例である。初期化ポリシーテーブル1500は、構成項目として、異常が生じている対象1501と、初期化ポリシー1502とを含む。初期化ポリシー1502は、どの初期化処理を実行するかを表す。本例において、初期化処理は、全体初期化処理、第1の初期化処理、及び、第2の初期化処理を含む。
- [0100] 初期化タスク304は、診断情報テーブル1400を参照することにより、診断対象1401に異常が生じているかを判定する。診断対象1401に

異常が生じている場合、初期化タスク304は、初期化ポリシーテーブル1500を参照することにより、その診断対象（即ち、異常が生じている対象1501）に対応する初期化ポリシー1502を取得する。初期化タスク304は、当該取得した初期化ポリシー1502に従って、初期化処理を実行する。

[0101] メインCPUコア100に異常が生じている場合、初期化タスク304は、全体初期化処理を実行する。更に、CANコントローラ200aに異常が生じている場合、初期化タスク304は、全体初期化処理を実行する。

[0102] 一方、「第1のCPUコア101に関連するハードウェア及びソフトウェア1510」に異常が生じている場合、初期化タスク304は、第1の初期化処理を実行する。

[0103] 更に、「第2のCPUコア102に関連するハードウェア及びソフトウェア1520」に異常が生じている場合、初期化タスク304は、第2の初期化処理を実行する。

[0104] 図16は、初期化タスク304が保持する初期化範囲テーブル1600の例である。初期化範囲テーブル1600は、各初期化処理が、どのハードウェア及びソフトウェアを初期化するかを表す。

[0105] 全体初期化処理は、全てのハードウェア及びソフトウェアを初期化する処理である。初期化タスク304は、以下のように全体初期化処理を実行する。初期化タスク304は、自己リセット信号出力端子911を介して電源回路901にリセット要求信号を出力する。電源回路901は、リセット要求信号に応じて、外部リセット端子912に外部リセット信号を入力する。マイコン900が外部リセット信号を受け取ると、マイコン900が再起動される。このように、マイコン900上の全てのハードウェア及びソフトウェアが初期化される。

[0106] 第1の初期化処理は、第1のCPUコア101に関連するハードウェア及びソフトウェアを初期化する処理である。初期化タスク304は、以下のように第1の初期化処理を実行する。初期化タスク304は、第1のCPUコ

ア１０１を再起動する。更に、初期化タスク３０４は、第１のサブＯＳ３１１を再起動する。その後、初期化タスク３０４は、第１のサブＯＳ３１１に所定の起動命令を送信する。第１のサブＯＳ３１１は、初期化タスク３０４からの起動命令に従い、第１のＧＰＩＯピンドライバ３１２を初期化する。更に、第１のサブＯＳ３１１は、第１の仮想ＣＡＮドライバ３１３を初期化する。これにより、第１のバッファ３１３ａが空になる。更に、第１のサブＯＳ３１１は、第１の制御ソフトウェア３１０を再起動する。更に、第１のサブＯＳ３１１は、第１のアクセス可能領域を初期化する。

[0107] 第２の初期化処理は、第２のＣＰＵコア１０２に関連するハードウェア及びソフトウェアを初期化する処理である。初期化タスク３０４は、以下のように第２の初期化処理を実行する。初期化タスク３０４は、第２のＣＰＵコア１０２を再起動する。更に、初期化タスク３０４は、第２のサブＯＳ３２１を再起動する。その後、初期化タスク３０４は、第２のサブＯＳ３２１に所定の起動命令を送信する。第２のサブＯＳ３２１は、初期化タスク３０４からの起動命令に従い、第２のＧＰＩＯピンドライバ３２２を初期化する。更に、第２のサブＯＳ３２１は、第２の仮想ＣＡＮドライバ３２３を初期化する。これにより、第２のバッファ３２３ａが空になる。更に、第２のサブＯＳ３２１は、第２の制御ソフトウェア３２０を再起動する。更に、第２のサブＯＳ３２１は、第２のアクセス可能領域を初期化する。

[0108] 以上のように、初期化タスク３０４は、異常が生じたハードウェア及びソフトウェアに応じて、初期化するハードウェア及びソフトウェアを変更する。例えば、第１のＣＰＵコア１０１に関連するハードウェア及びソフトウェア１５１０（図１５を参照）に異常が生じている場合、初期化タスク３０４は、第１の初期化処理を実行する。初期化タスク３０４は、第１のＣＰＵコア１０１に関連するハードウェア及びソフトウェア１５１０のみを初期化することができる。第１の初期化処理が実行されている間、第１の格納処理及び第１の出力処理は中断される。一方、第１の初期化処理が実行されている間、第２の仮想ＣＡＮドライバ３２３は第２の格納処理を継続し、制御タス

ク 3 0 0 は第 2 の出力処理を継続する。従って、第 2 の制御ソフトウェア 3 2 0 が、第 2 のペリフェラル送信情報 7 2 0 の C A N コントローラ 2 0 0 a への送信を継続することができる。

[0109] 第 2 の C P U コア 1 0 2 に関連するハードウェア及びソフトウェア 1 5 2 0 (図 1 5 を参照) に異常が生じている場合、初期化タスク 3 0 4 は、第 2 の初期化処理を実行する。初期化タスク 3 0 4 は、第 2 の C P U コア 1 0 2 に関連するハードウェア及びソフトウェア 1 5 2 0 のみを初期化することができる。第 2 の初期化処理が実行されている間、第 2 の格納処理及び第 2 の出力処理は中断される。一方、第 2 の初期化処理が実行されている間、第 1 の仮想 C A N ドライバ 3 1 3 は第 1 の格納処理を継続し、制御タスク 3 0 0 は第 1 の出力処理を継続する。従って、第 1 の制御ソフトウェア 3 1 0 が、第 1 のペリフェラル送信情報 7 1 0 の C A N コントローラ 2 0 0 a への送信を継続することができる。

[0110] <実施例 4>

実施例 4 における電子制御装置 1 0 0 0 の構成について説明する。以下では、実施例 3 との違いを中心に説明する。メモリ 3 は、第 1 の初期化処理フラグを管理する領域と、第 2 の初期化処理フラグを管理する領域とを備える。

[0111] 第 1 の初期化処理フラグは、第 1 のアクセス可能領域において管理される。第 1 の初期化処理フラグは、True 又は False のいずれかの値を取る。第 1 の初期化処理フラグは、第 1 の初期化処理が実行されている期間において、True の値である。第 1 の初期化処理フラグは、第 1 の初期化処理が実行されていない期間において、False の値である。

[0112] 第 1 の初期化処理フラグが False の場合、第 1 の仮想 C A N ドライバ 3 1 3 は第 1 の格納処理を実行し、制御タスク 3 0 0 は第 1 の出力処理を実行する。初期化タスク 3 0 4 は、第 1 の初期化処理を開始する際に、第 1 の初期化処理フラグを True に設定する。従って、第 1 の仮想 C A N ドライバ 3 1 3 は第 1 の格納処理を中断し、制御タスク 3 0 0 は第 1 の出力処理を中断する。

初期化タスク 304 は、第 1 の初期化処理が完了した際に、第 1 の初期化処理フラグを False に設定する。従って、第 1 の仮想 C A N ドライバ 313 は第 1 の格納処理を再開し、制御タスク 300 は第 1 の出力処理を再開する。上記の構成によれば、第 1 の初期化処理の完了に応じて、第 1 の格納処理及び第 1 の出力処理を再開させることができる。

[0113] 第 2 の初期化処理フラグは、第 2 のアクセス可能領域において管理される。第 2 の初期化処理フラグは、True 又は False のいずれかの値を取る。第 2 の初期化処理フラグは、第 2 の初期化処理が実行されている期間において、True の値である。第 2 の初期化処理フラグは、第 2 の初期化処理が実行されていない期間において、False の値である。

[0114] 第 2 の初期化処理フラグが False の場合、第 2 の仮想 C A N ドライバ 323 は第 2 の格納処理を実行し、制御タスク 300 は第 2 の出力処理を実行する。初期化タスク 304 は、第 2 の初期化処理を開始する際に、第 2 の初期化処理フラグを True に設定する。従って、第 2 の仮想 C A N ドライバ 323 は第 2 の格納処理を中断し、制御タスク 300 は第 2 の出力処理を中断する。初期化タスク 304 は、第 2 の初期化処理が完了した際に、第 2 の初期化処理フラグを False に設定する。従って、第 2 の仮想 C A N ドライバ 323 は第 2 の格納処理を再開し、制御タスク 300 は第 2 の出力処理を再開する。上記の構成によれば、第 2 の初期化処理の完了に応じて、第 2 の格納処理及び第 2 の出力処理を再開させることができる。

[0115] 本例は、実施例 1 に適用されてもよい。例えば、第 1 の仮想 C A N ドライバ 313 は、第 1 のバッファ 313 a を初期化する際に、第 1 の初期化処理フラグを True に設定する。第 1 の仮想 C A N ドライバ 313 は第 1 の格納処理を中断し、制御タスク 300 は第 1 の出力処理を中断する。第 1 の仮想 C A N ドライバ 313 は、第 1 のバッファ 313 a の初期化が完了した際に、第 1 の初期化処理フラグを False に設定する。従って、第 1 の仮想 C A N ドライバ 313 は第 1 の格納処理を再開し、制御タスク 300 は第 1 の出力処理を再開する。

[0116] <実施例 5>

実施例 5 における電子制御装置 1000 の構成について説明する。以下では、実施例 1 との違いを中心に説明する。本例において、メモリ 3 は、第 1 のダミーバッファを更に含む。図 17 は、第 1 のダミーバッファ 313b のデータ構造を示した図である。第 1 のダミーバッファ 313b は、First-In-First-Out キューの実装の 1 つであるリングバッファ構造を有する。第 1 のダミーバッファ 313b の各要素は、所定時間 t_m だけ待機する処理（所謂、wait 処理）である。リングバッファの実体は、構造体配列 dummy_buffer_virtio1[12] である。なお、ここでは説明の都合上、配列の要素数を 12 としているが、これに限定されない。

[0117] 上記の時間 t_m は、第 1 の制御ソフトウェア 310 が CAN_send_virtio1(ID, data) をコールした時点から、CAN ドライバ 302 が第 1 のペリフェラル送信情報 710 を CAN コントローラ 200a の内部レジスタ 200b に格納する時点までの処理時間の平均値である。他の言い方をすれば、時間 t_m は、第 1 のペリフェラル送信情報 710 が第 1 の制御ソフトウェア 310 から CAN コントローラ 200a へ到着するまでの時間の平均値である。上記の処理時間の平均値が 50ms である場合、wait 処理における時間 t_m は、50ms に設定される。

[0118] 第 1 の制御ソフトウェア 310 は、上述の特定状況において、第 1 の中断指令を第 1 の仮想 CAN ドライバ 313 に送信する。第 1 の仮想 CAN ドライバ 313 は、第 1 の中断指令に応じて、第 1 の格納処理を中断する。第 1 の仮想 CAN ドライバ 313 は、第 1 のバッファ 313a を初期化する。

[0119] ここで、第 1 の仮想 CAN ドライバ 313 は、第 1 のバッファ 313a を初期化する前に、第 1 のバッファ 313a に格納されている第 1 のペリフェラル送信情報 710 の数を取得する。即ち、第 1 の仮想 CAN ドライバ 313 は、第 1 のバッファ 313a の中で、第 1 のペリフェラル送信情報 710 が格納されている配列の要素数を取得する。例えば、12 個の要素の中で、7 個の要素に第 1 のペリフェラル送信情報 710 が格納されていると仮定す

る。この場合、第1の仮想CANドライバ313は、同じ数のwait処理を第1のダミーバッファ313bに格納する。

[0120] 第1の仮想CANドライバ313は、7個のwait処理を第1のダミーバッファ313bに格納する。図17に示すように、第1の仮想CANドライバ313は、dummy_buffer_virtio1[0]~dummy_buffer_virtio1[6]のそれぞれに、wait処理を格納する。更に、第1の仮想CANドライバ313は、バッファ先端のインデックスforwardに「0」を代入し、バッファ終端のインデックスbackwardに「6」を代入する。

[0121] 制御タスク300は、特定状況において、第1のバッファ313aの代わりに第1のダミーバッファ313bからデータを取り出す。即ち、制御タスク300は、第1のダミーバッファ313bのバッファ先頭位置を示すインデックスforwardに含まれたデータ（即ち、wait処理）を取り出して、wait処理を実行する。そして、制御タスク300は、インデックスforwardを以下のように演算する（ $forward \leftarrow (forward + 1) \% 12$ ）。なお、インデックスforwardが配列終端（ $forward = 11$ ）である場合、制御タスク300は、インデックスforwardに「0」を代入する。次に、制御タスク300は、第2の出力処理を実行する。

[0122] このように、制御タスク300は、第1のダミーバッファ313b及び第2のバッファ323aからデータを交互に取り出す。即ち、制御タスク300は、wait処理（50msの待機処理）と第2の出力処理とを交互に実行する。

[0123] なお、第1の仮想CANドライバ313は、特定状況において、所定の第1の周期でwait処理を第1のダミーバッファ313bに格納してもよい。これにより、第1の仮想CANドライバ313は、第1の制御ソフトウェア310がCAN_send_virtio1(ID, data)をコールするタイミングを模擬する。第1の仮想CANドライバ313がwait処理を格納する第1の周期は、第1の制御ソフトウェア310がCAN_send_virtio1(ID, data)をコールする時間間隔の平均値である。なお、第1の周期は、一定の周期に限定されず、上記時

間隔の平均値をパラメータとして用いてポアソン確率過程に従って演算された値であってもよい。

[0124] 通常、制御タスク300は第1の出力処理及び第2の出力処理を交互に実行する。従って、制御タスク300は、実質的に一定の周期で第2の出力処理を実行する。これに対して、実施例1においては、制御タスク300が第1の出力処理を実行することなく第2の出力処理を連続して実行する（図11を参照）ので、第2の出力処理が実行される周期が極端に短くなる。本例によれば、制御タスク300はwait処理と第2の出力処理を交互に実行するので、第2の出力処理が実行される周期が一定の周期に保たれる。

[0125] <実施例6>

実施例6における電子制御装置1000の構成について説明する。本例において、第1の制御ソフトウェア310は、ハイブリッド電気自動車（Hybrid Electric Vehicle、以下、「HEV」と称呼する）におけるエンジンを制御するソフトウェアである。例えば、第1の制御ソフトウェア310は、エンジンでトルクを発生させるためのソフトウェアである。第1の制御ソフトウェア310は、エンジンで発生させるべきトルク目標値に関連する情報（吸気バルブの開口量、燃料噴射量、及び、点火のタイミング等）を第1のペリフェラル送信情報710として送信する。

[0126] 第2の制御ソフトウェア320は、HEVにおける複数の動力源（エンジン及びインバータ・モータ）を制御するソフトウェアである。第2の制御ソフトウェア320はエンジン及びインバータ・モータのそれぞれで発生させるべきトルク目標値に関する情報を第2のペリフェラル送信情報720として送信する。

[0127] HEVは、エンジンを発電用の動力源として使用し、且つ、モータを走行用の動力源として使用する。本例によれば、エンジン制御用のソフトウェア及びHEV制御用のソフトウェアが1つのマイコン900に搭載される。従って、低コスト化を実現することができる。

[0128] <実施例7>

実施例 7 における電子制御装置 1000 の構成について説明する。本例において、第 1 の制御ソフトウェア 310 は、機能安全規格 ISO26262 による QM(Quality Management) に従ったソフトウェアである。第 2 の制御ソフトウェア 320 は、ASIL (Automotive Safety Integrity Level) -C に従ったソフトウェアである。

[0129] 従って、第 2 の制御ソフトウェア 320 が要求される安全基準は、第 1 の制御ソフトウェア 310 が要求される安全基準よりも高い。第 1 の制御ソフトウェア 310 に関する異常が生じた特定状況においても、より安全基準が高い第 2 の制御ソフトウェア 320 が、第 2 のペリフェラル送信情報 720 の CAN コントローラ 200a への送信を継続することができる。第 1 の制御ソフトウェア 310 に関する異常が第 2 の制御ソフトウェア 320 に影響を与えないので、電子制御装置 1000 は、FFI (Freedom From Interference) の機能を実現することができる。

[0130] <実施例 8>

実施例 8 における電子制御装置 1000 の構成について説明する。特定状況において、第 2 の制御ソフトウェア 320 は、通常の処理（第 2 のペリフェラル送信情報 720 の送信処理）に加えて、第 1 の制御ソフトウェア 310 の処理の一部を実行するように構成されてもよい。上記の構成によれば、特定状況が生じた場合においても、電子制御装置 1000 は、第 1 の制御ソフトウェア 310 及び第 2 の制御ソフトウェア 320 の両方の処理を維持できる。

[0131] 例えば、第 1 の制御ソフトウェア 310 は、上述のように、エンジンを制御するソフトウェアである。第 2 の制御ソフトウェア 320 は、上述のように、HEV における複数の動力源（エンジン及びインバータ・モータ）を制御するソフトウェアである。この場合、第 2 の制御ソフトウェア 320 は、特定状況において、第 1 の制御ソフトウェア 310 の処理の一部（エンジンで発生させるべきトルク目標値に関連する情報の出力）を更に実行してもよい。

[0132] <実施例 9>

実施例 9 における電子制御装置 1000 の構成について説明する。特定状況において、第 2 の制御ソフトウェア 320 が自身の機能を低下させてもよい。例えば、第 2 の制御ソフトウェア 320 は、第 2 のペリフェラル送信情報 720 のデータの内容を変更したり、又は、通常の処理（第 2 のペリフェラル送信情報 720 の送信処理）を実行する速度を変更してもよい。上記の構成によれば、第 2 の制御ソフトウェア 320 が処理を継続できる可能性を高めることができる。

[0133] 例えば、第 1 の制御ソフトウェア 310 は、エンジンを制御するソフトウェアであり、第 2 の制御ソフトウェア 320 は、HEV における複数の動力源（エンジン及びインバータ・モータ）を制御するソフトウェアである。この構成において、第 2 の制御ソフトウェア 320 は、特定状況において、インバータ・モータで発生されるトルクの目標値を比較的小さい値に変更して、最低限のトルクで車両を走行させてもよい（即ち、フェールセーフ走行を実行してもよい）。この構成によれば、異常が発生した車両を安全な位置まで移動させることができる。

[0134] 以上、本発明の実施形態について詳述したが、本発明は、前記の実施形態に限定されるものではなく、特許請求の範囲に記載された本発明の精神を逸脱しない範囲で、種々の設計変更を行うことができるものである。例えば、前記した実施の形態は本発明を分かりやすく説明するために詳細に説明したものであり、必ずしも説明した全ての構成を備えるものに限定されるものではない。また、ある実施形態の構成の一部を他の実施形態の構成に置き換えることが可能であり、また、ある実施形態の構成に他の実施形態の構成を加えることも可能である。さらに、各実施形態の構成の一部について、他の構成の追加・削除・置換をすることが可能である。

符号の説明

[0135] 1000 電子制御装置

1 CPU

- 2 ペリフェラル
- 3 メモリ
- 4 バス
- 3 0 0 制御タスク
- 3 0 1 メインOS
- 3 0 2 CANドライバ
- 3 1 0 第1の制御ソフトウェア
- 3 1 1 第1のサブOS
- 3 1 2 第1のGPIOピンドライバ
- 3 1 3 第1の仮想CANドライバ
- 3 1 3 a 第1のバッファ
- 3 2 0 第2の制御ソフトウェア
- 3 2 1 第2のサブOS
- 3 2 2 第2のGPIOピンドライバ
- 3 2 3 第2の仮想CANドライバ
- 3 2 3 a 第2のバッファ

請求の範囲

- [請求項1] 第1の制御ソフトウェア及び第2の制御ソフトウェアを格納するメモリと、
- 前記第1の制御ソフトウェア及び前記第2の制御ソフトウェアを実行するCPUと、
- 前記第1の制御ソフトウェア及び前記第2の制御ソフトウェアによって使用されるペリフェラル回路と
- を備え、
- 前記メモリは、第1のバッファと、第2のバッファとを更に備え、
- 前記CPUは、
- 前記第1の制御ソフトウェアから前記ペリフェラル回路への送信情報である第1のペリフェラル送信情報を前記第1のバッファに格納する第1の格納処理と、
- 前記第2の制御ソフトウェアから前記ペリフェラル回路への送信情報である第2のペリフェラル送信情報を前記第2のバッファに格納する第2の格納処理と、
- 前記第1のバッファから前記ペリフェラル回路へ前記第1のペリフェラル送信情報を出力する第1の出力処理と、
- 前記第2のバッファから前記ペリフェラル回路へ前記第2のペリフェラル送信情報を出力する第2の出力処理と
- を実行するように構成され、
- 前記CPUは、前記第1の制御ソフトウェアに関する異常が生じた特定状況において、前記第1の格納処理及び前記第1の出力処理の少なくとも一方を中断し、
- 前記第2の格納処理及び前記第2の出力処理を継続するように構成された、
- 車両制御装置。
- [請求項2] 請求項1に記載の車両制御装置において、

前記CPUは、前記特定状況において、前記第1のバッファを初期化する第1の初期化処理を実行するように構成された、

車両制御装置。

[請求項3]

請求項2に記載の車両制御装置において、

前記CPUは、前記第1の初期化処理において、前記第1の制御ソフトウェアに関連するソフトウェア及びハードウェアを更に初期化するように構成された、

車両制御装置。

[請求項4]

請求項2に記載の車両制御装置において、

前記CPUは、前記第1の初期化処理が完了した後に、前記第1の格納処理及び前記第1の出力処理を再開するように構成された、

車両制御装置。

[請求項5]

請求項1に記載の車両制御装置において、

前記メモリは、第1のダミーバッファを更に備え、

前記第1のダミーバッファは、所定時間だけ待機する待機処理を含み、

前記CPUは、前記特定状況において、前記待機処理と前記第2の出力処理とを交互に実行するように構成された、

車両制御装置。

[請求項6]

請求項5に記載の車両制御装置において、

前記所定時間は、前記第1のペリフェラル送信情報が前記第1の制御ソフトウェアから前記ペリフェラル回路へ到着するまでの時間の平均値である、

車両制御装置。

[請求項7]

請求項1に記載の車両制御装置において、

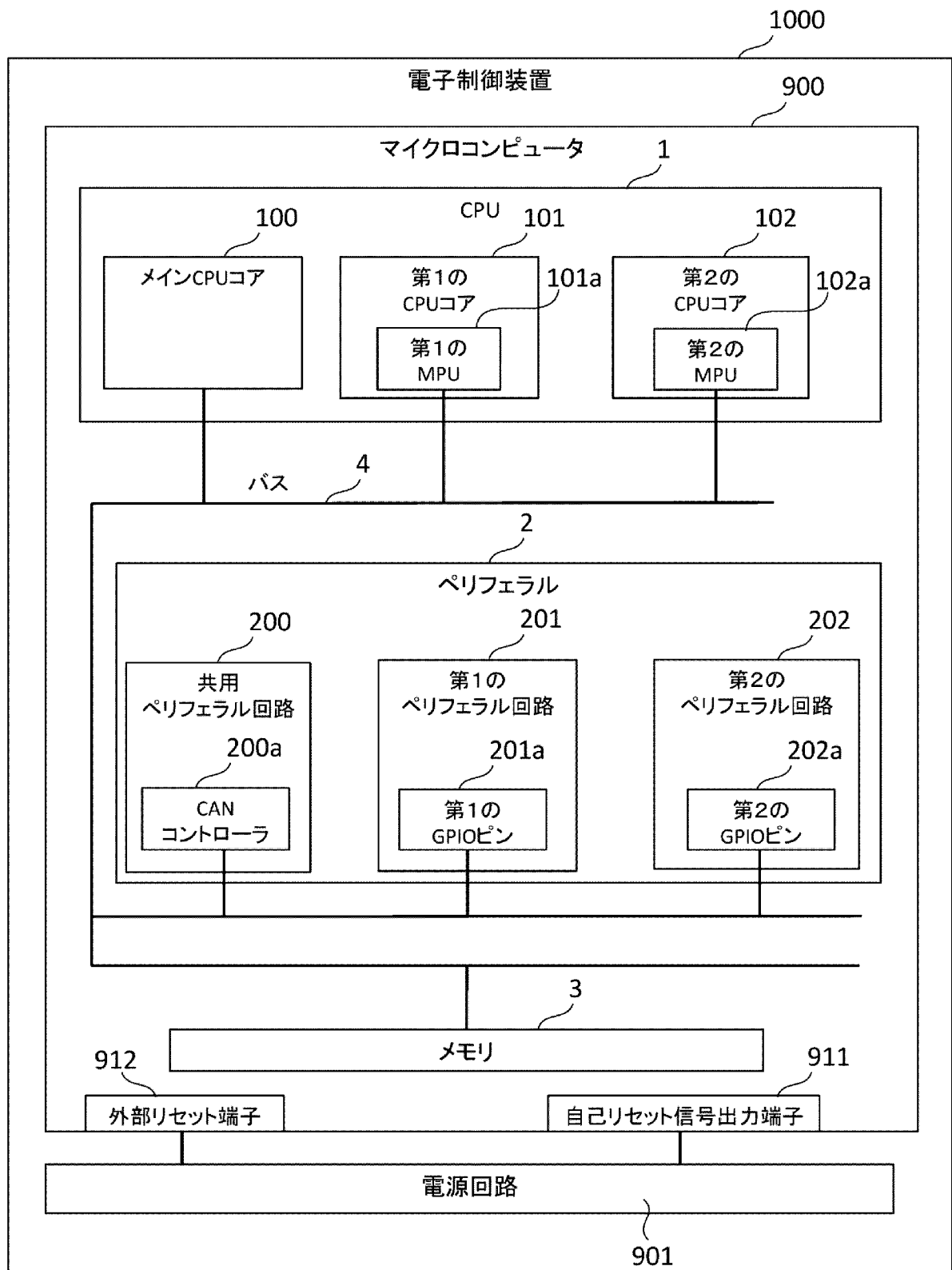
前記第2の制御ソフトウェアが要求される安全基準は、前記第1の制御ソフトウェアが要求される安全基準よりも高い、

車両制御装置。

- [請求項8] 請求項 1 に記載の車両制御装置において、
 前記第 2 の制御ソフトウェアは、前記特定状況において、前記第 1
 の制御ソフトウェアの処理の一部を実行するように構成された、
 車両制御装置。
- [請求項9] 請求項 1 に記載の車両制御装置において、
 前記第 1 の制御ソフトウェアは、ハイブリッド電気自動車における
 エンジンを制御するソフトウェアであり、
 前記第 2 の制御ソフトウェアは、前記ハイブリッド電気自動車にお
 ける前記エンジン及びモータを制御するソフトウェアである、
 車両制御装置。
- [請求項10] 請求項 1 に記載の車両制御装置において、
 前記特定状況は、
 前記第 1 の制御ソフトウェアが通常の処理以外の例外処理を実行す
 る状況、
 前記第 1 の制御ソフトウェアに関連するソフトウェアに異常が生じ
 た状況、及び、
 前記第 1 の制御ソフトウェアに関連するハードウェアに異常が生じ
 た状況、
 の少なくとも 1 つを含む、
 車両制御装置。
- [請求項11] 請求項 1 に記載の車両制御装置において、
 前記 CPU は、第 1 の CPU コア及び第 2 の CPU コアを少なくと
 も含み、
 前記第 1 の CPU コアは、前記第 1 の制御ソフトウェアを実行する
 ように構成され、
 前記第 2 の CPU コアは、前記第 2 の制御ソフトウェアを実行する
 ように構成された、 車両制御装置。

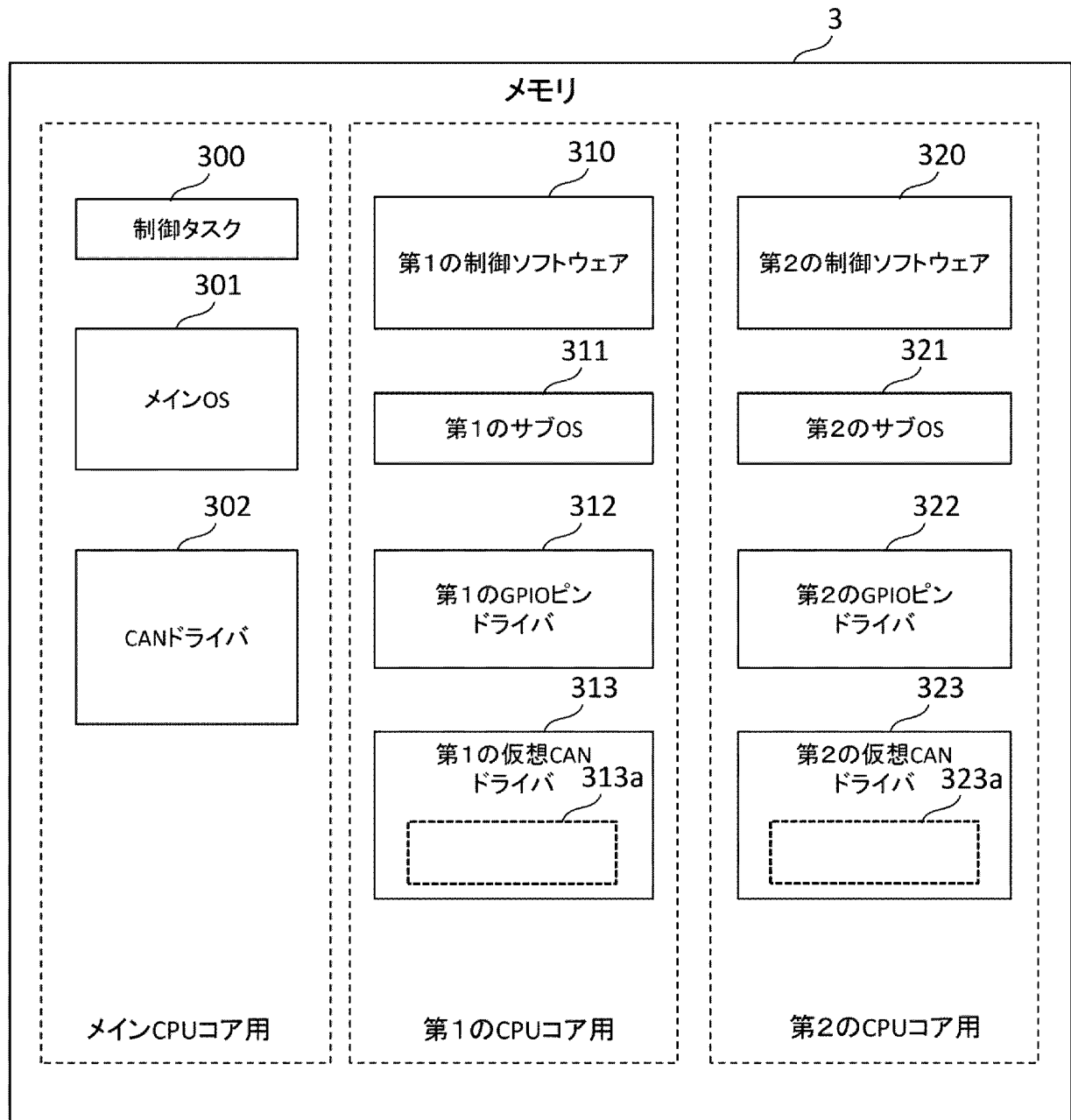
[図1]

図 1



[図2]

図 2



[図3]

図 3

メモリ領域	読み取り権	書き込み権
メインOSの格納領域		
第1のサブOSの格納領域	可	可
第2のサブOSの格納領域		
第1の制御ソフトウェアの格納領域	可	可
第2の制御ソフトウェアの格納領域		
CANドライバの格納領域		
第1のGPIOピンドライバの格納領域	可	可
第1の仮想CANドライバの格納領域	可	可
第2のGPIOピンドライバの格納領域		
第2の仮想CANドライバの格納領域		
制御タスクの格納領域		

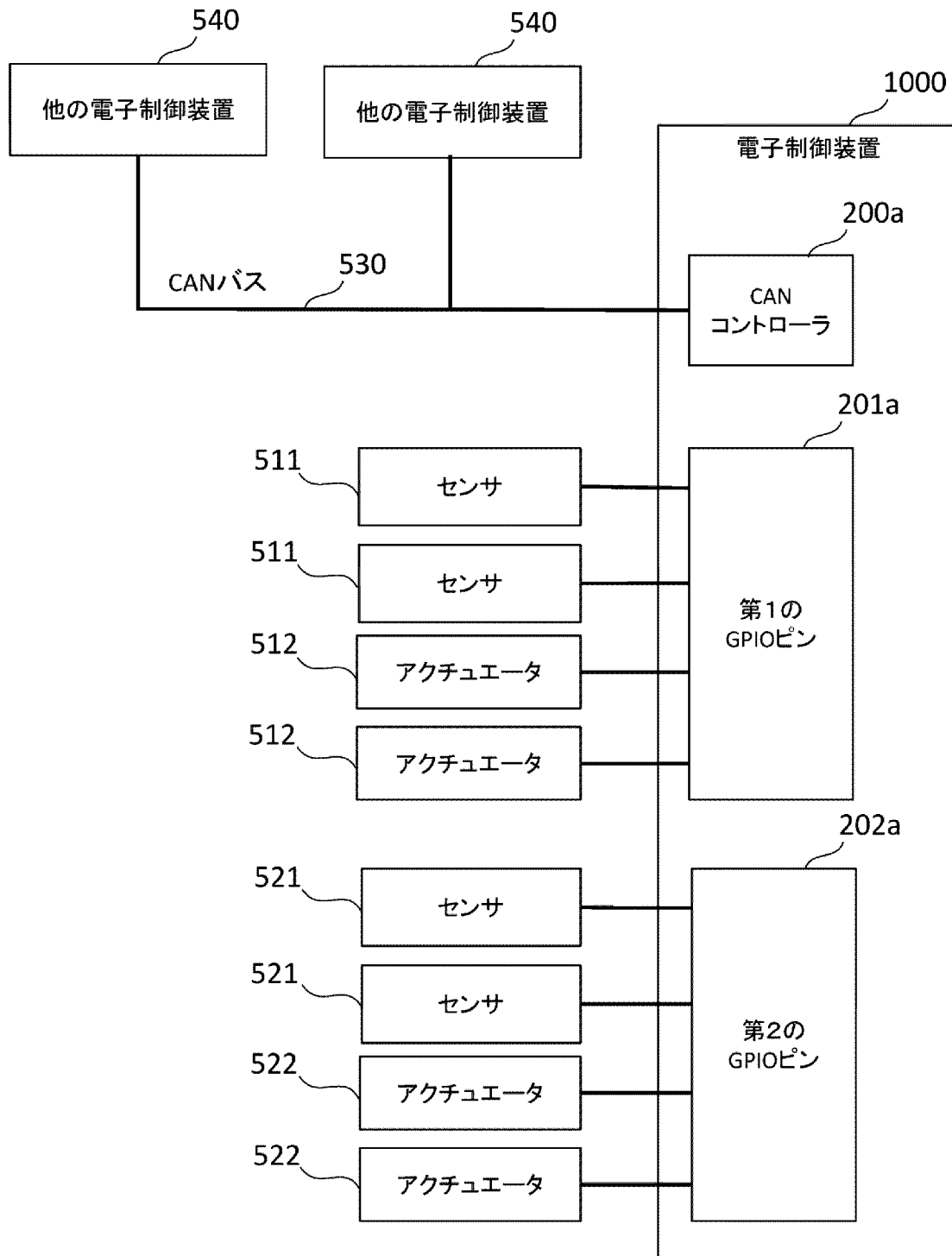
[図4]

図 4

メモリ領域	読み取り権	書き込み権
メインOSの格納領域		
第1のサブOSの格納領域		
第2のサブOSの格納領域	可	可
第1の制御ソフトウェアの格納領域		
第2の制御ソフトウェアの格納領域	可	可
CANドライバの格納領域		
第1のGPIOピンドライバの格納領域		
第1の仮想CANドライバの格納領域		
第2のGPIOピンドライバの格納領域	可	可
第2の仮想CANドライバの格納領域	可	可
制御タスクの格納領域		

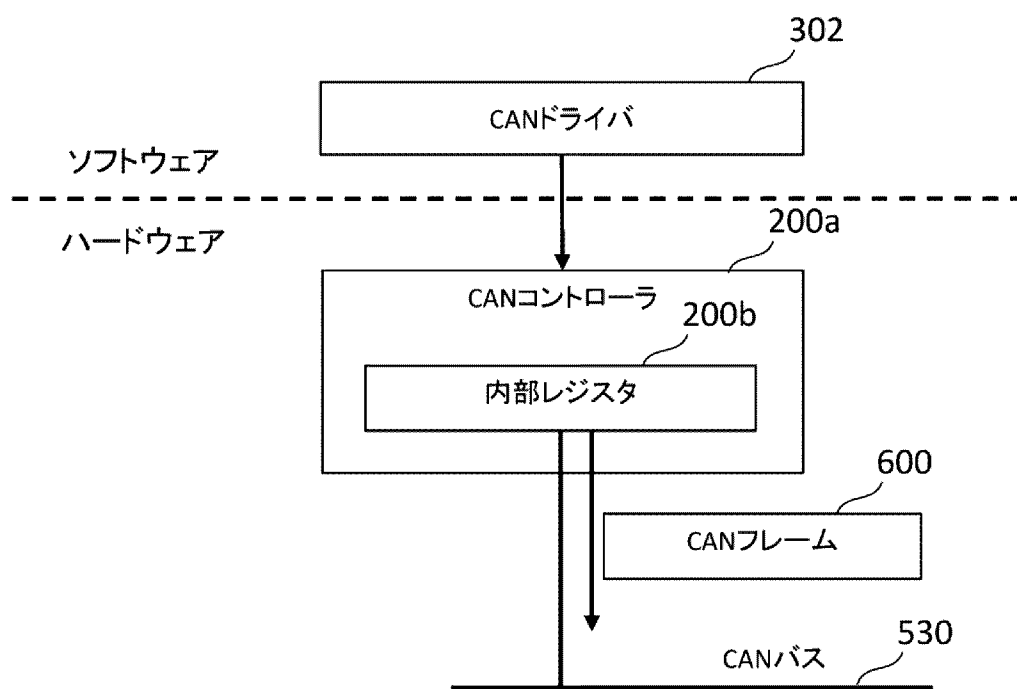
[図5]

図 5



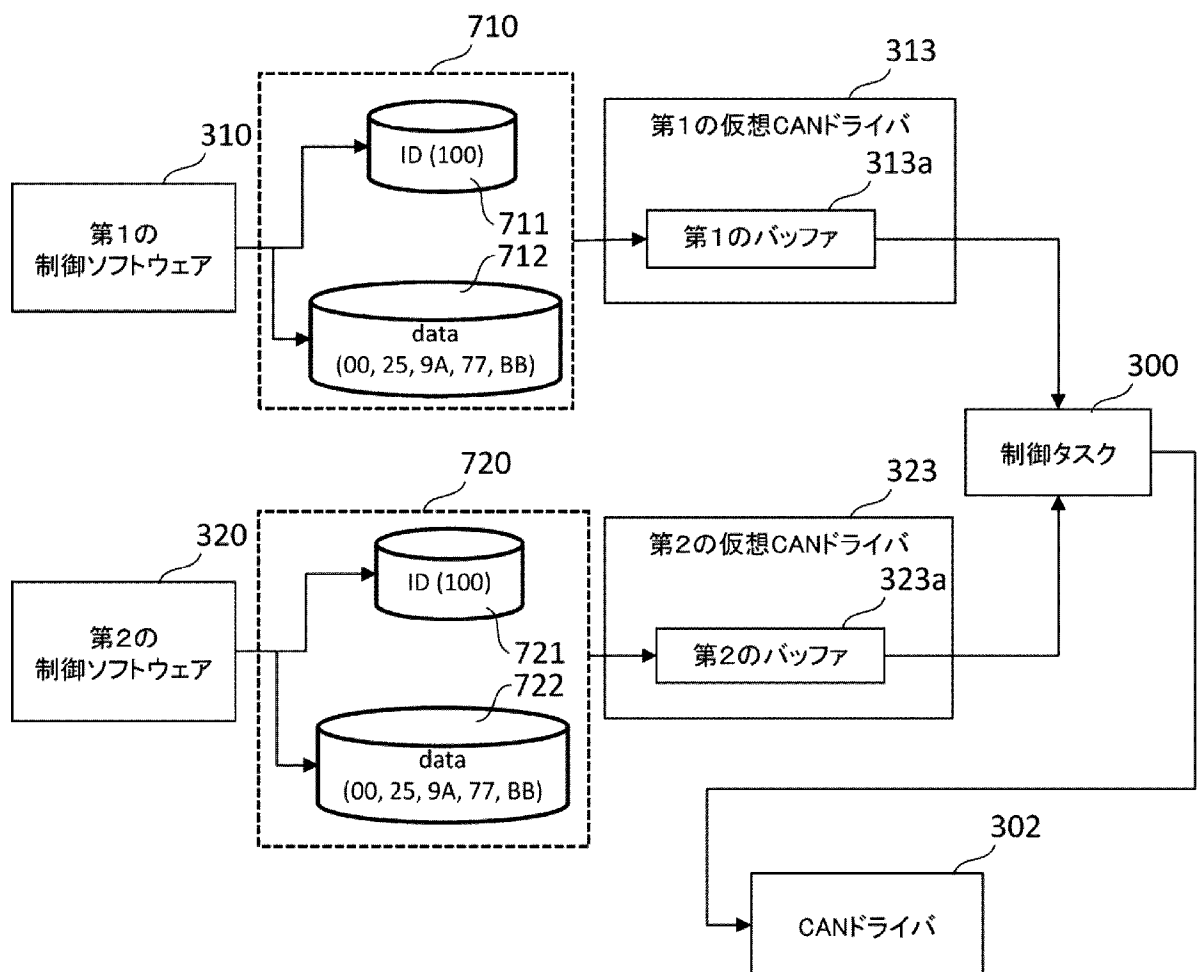
[図6]

図 6



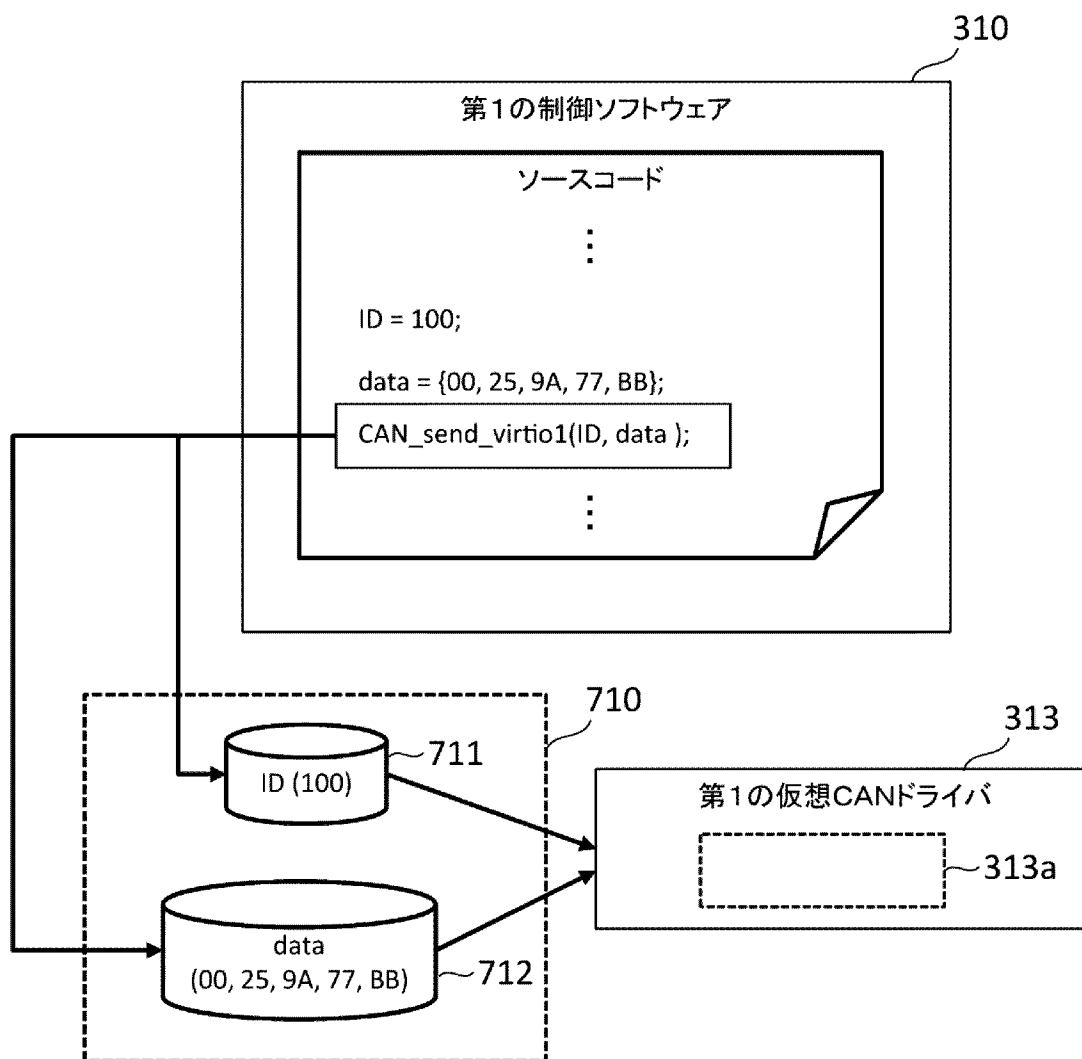
[図7]

図 7



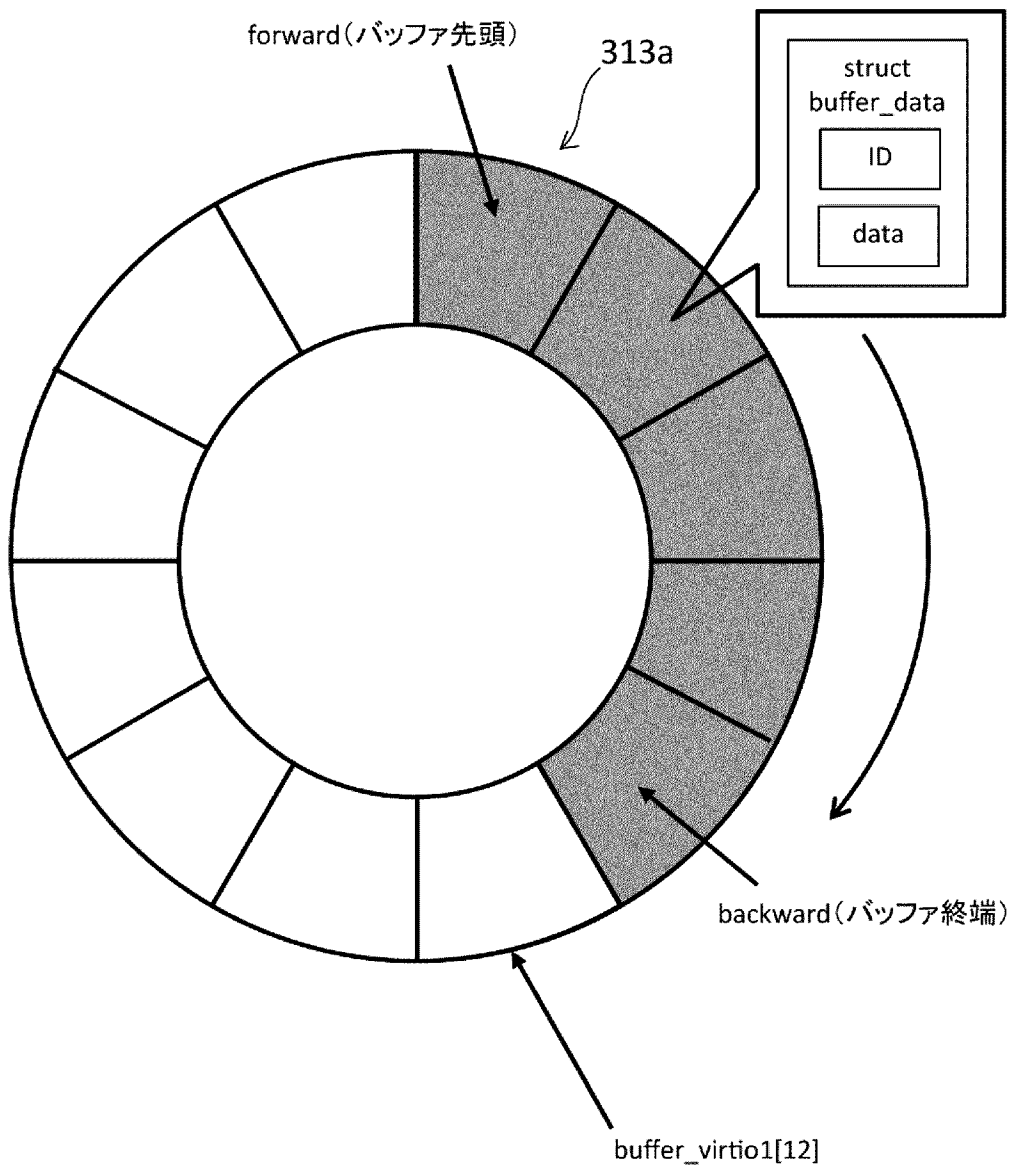
[図8]

図 8



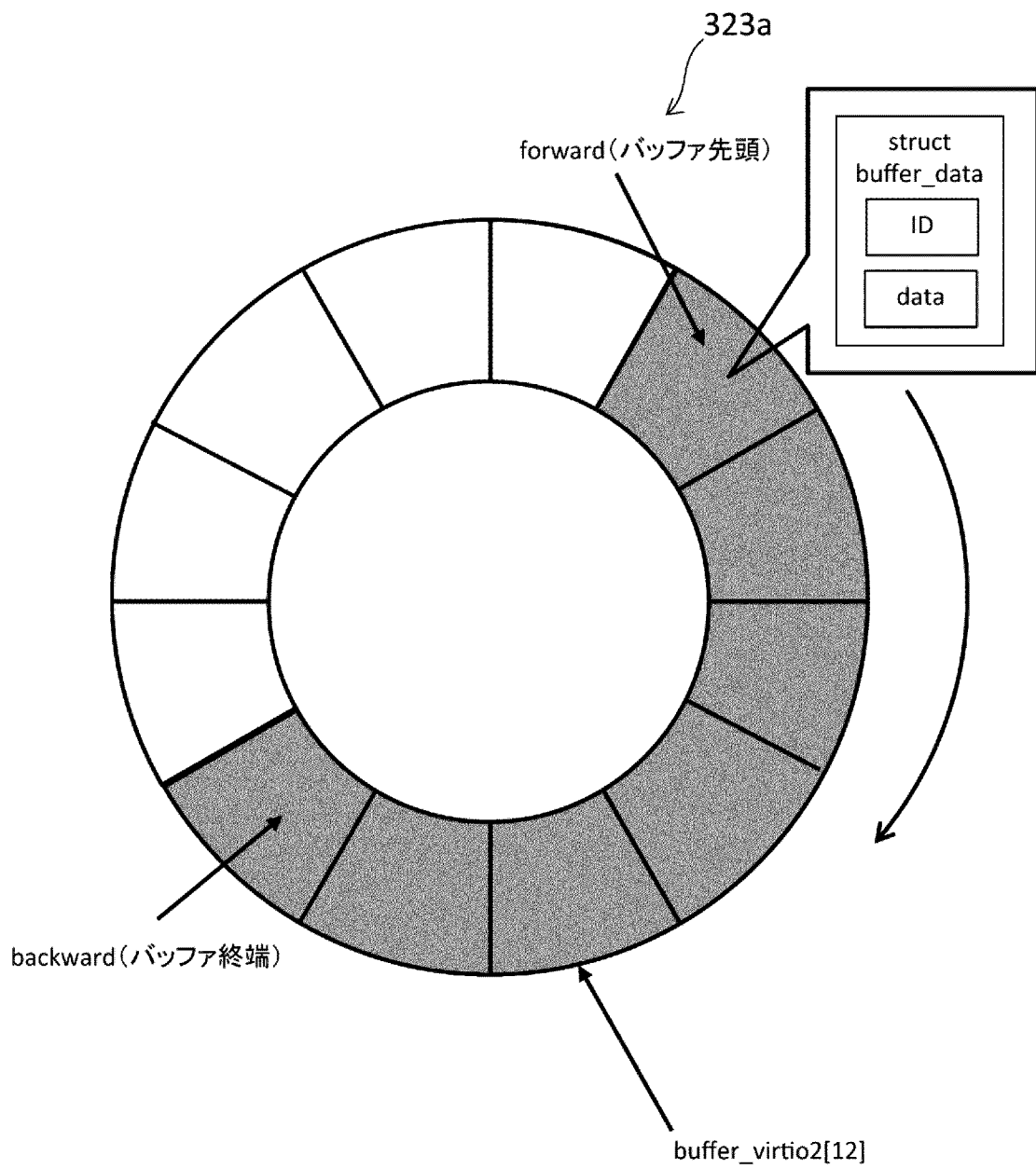
[図9]

図 9



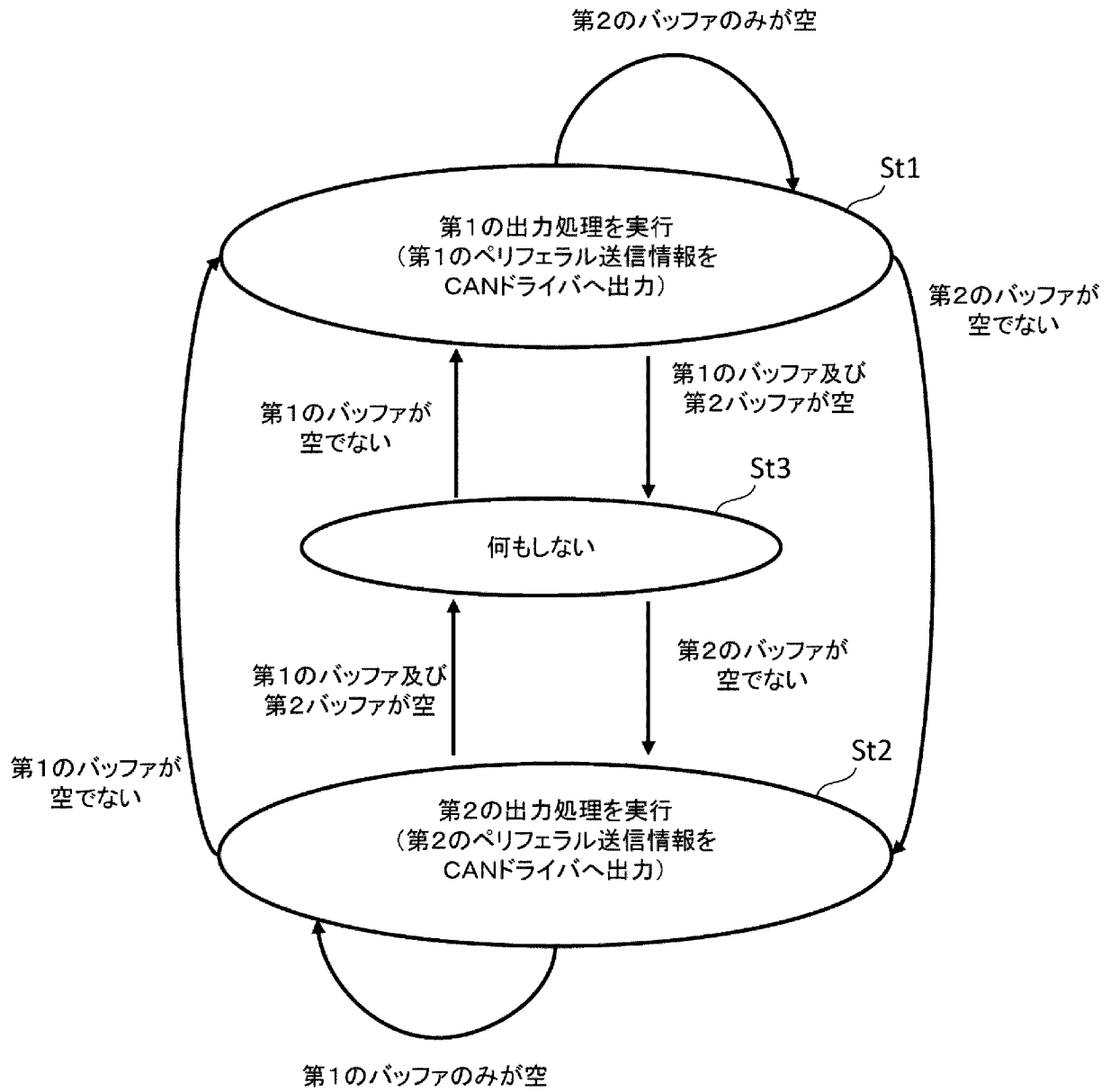
[図10]

図 1 0



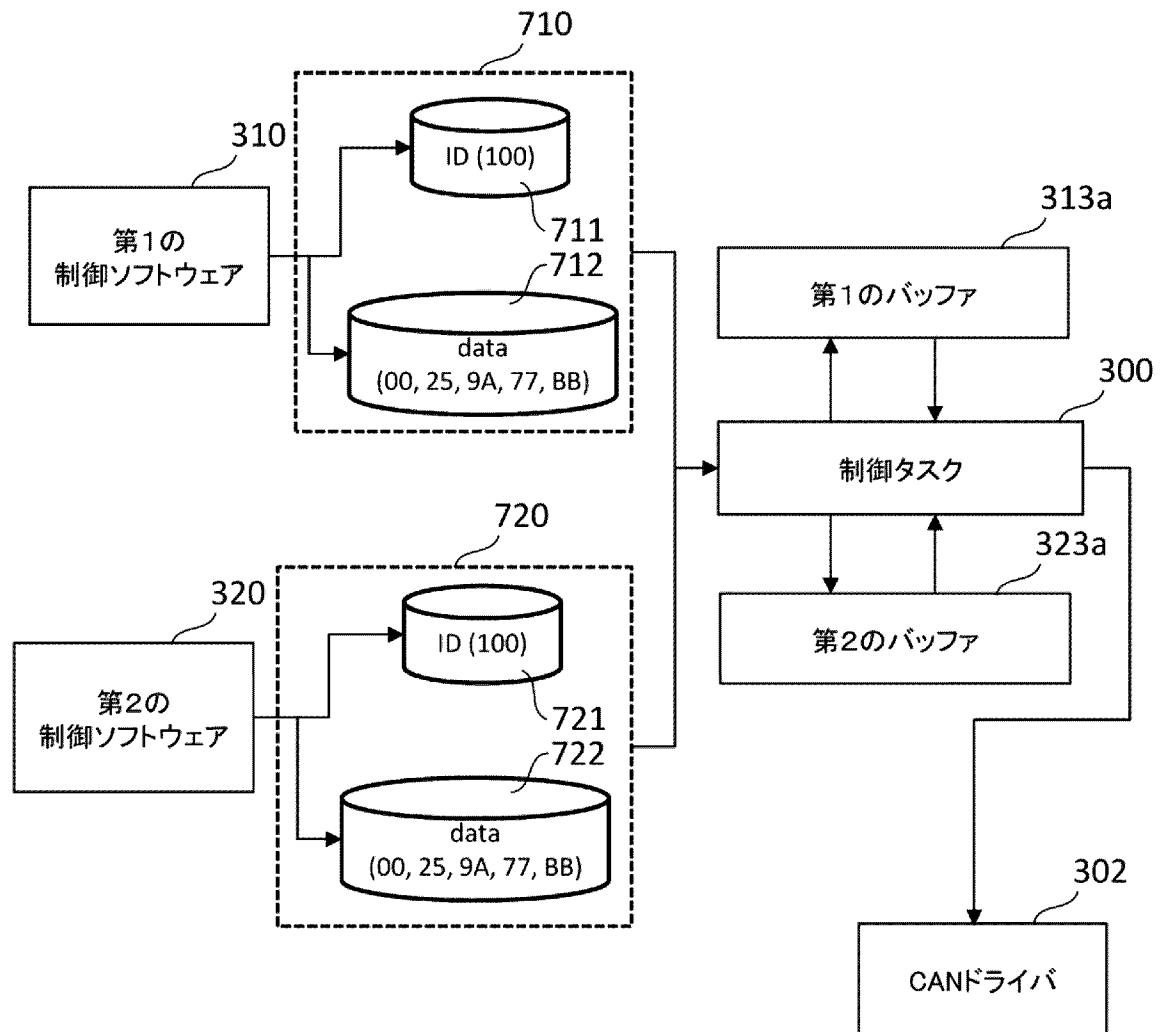
[図11]

図 1 1



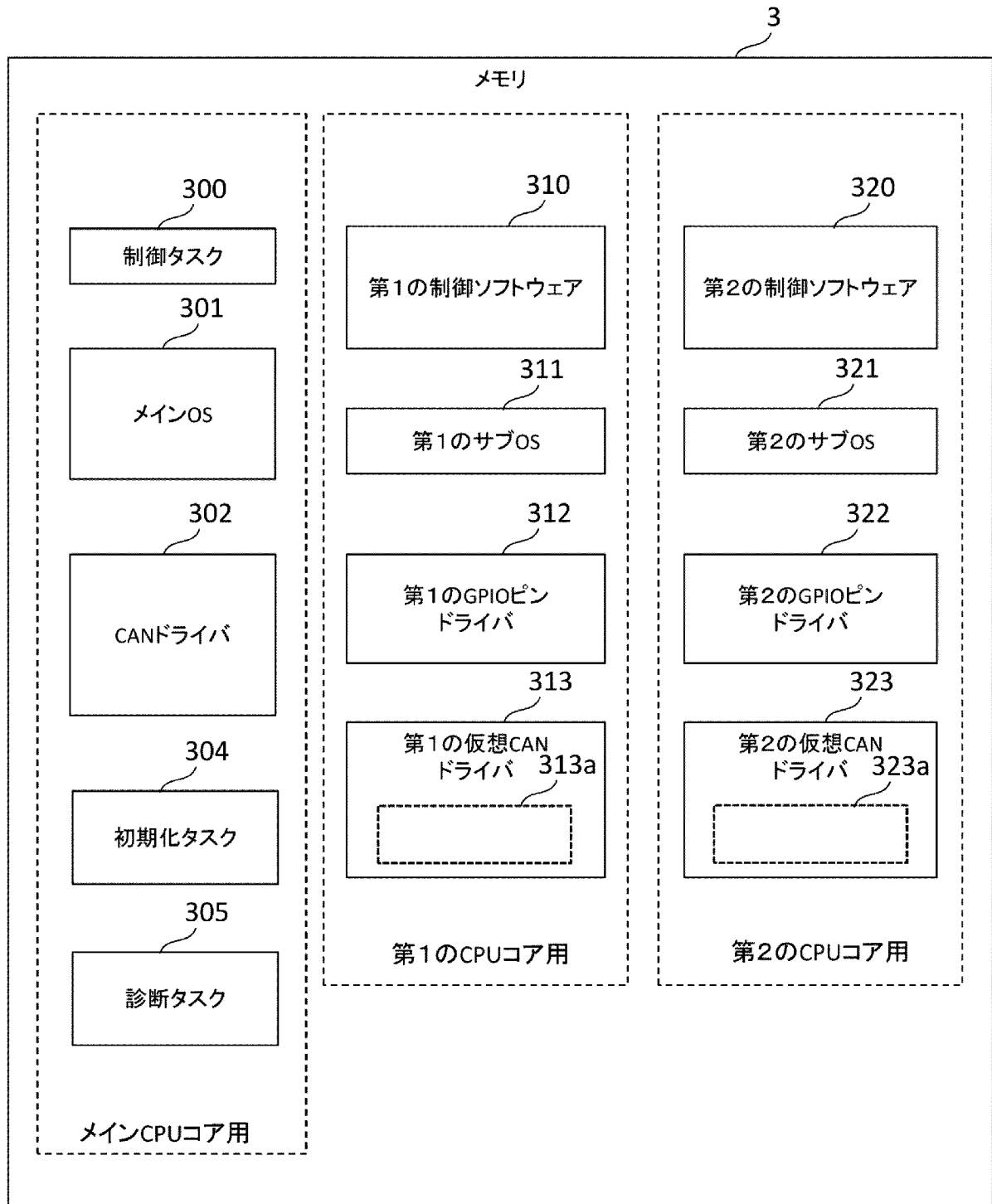
[図12]

図 1 2



[図13]

図 1 3



[図14]

図 1 4



The diagram shows a table labeled 1400. The table has two columns: 1401 (Diagnostic Object) and 1402 (Status). The table lists various components and their status. The status is 'Normal' for most components, except for the first control software, which is 'Abnormal'.

診断対象	状態
メインCPUコア	正常
第1のCPUコア	正常
第2のCPUコア	正常
CANコントローラ	正常
第1のGPIOピン	正常
第2のGPIOピン	正常
第1のアクセス可能領域	正常
第2のアクセス可能領域	正常
第1の制御ソフトウェア	異常
第2の制御ソフトウェア	正常

[図15]

図 1 5

1500	
1501	1502
異常が生じている対象	初期化ポリシー
メインCPUコア	全体初期化处理
CANコントローラ	全体初期化处理
1510 {	第1の初期化处理
	第1の初期化处理
	第1の初期化处理
	第1の初期化处理
	第1の初期化处理
1520 {	第2の初期化处理
	第2の初期化处理
	第2の初期化处理
	第2の初期化处理
	第2の初期化处理

[図16]

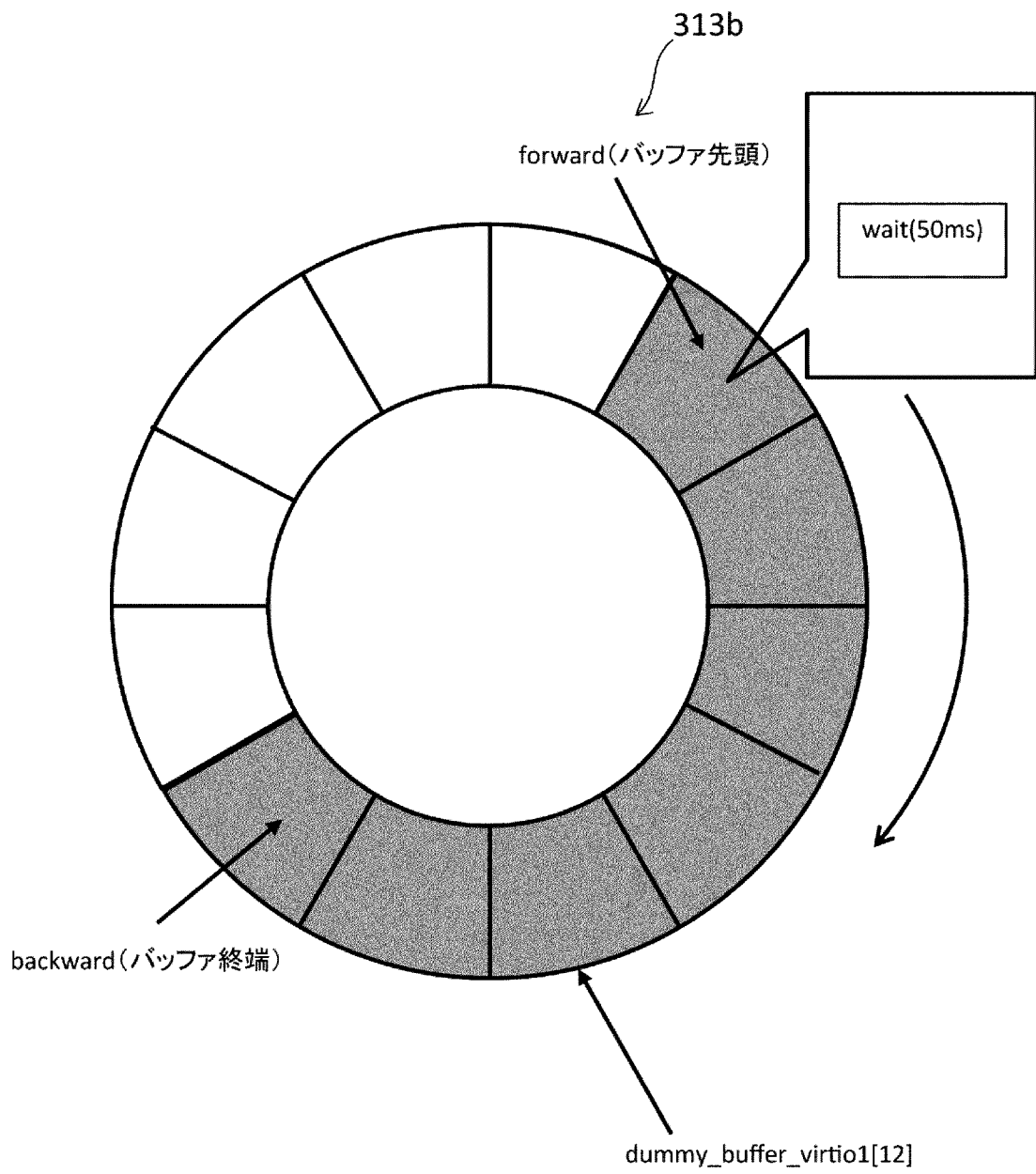
図 1 6

1600
↙

	全体 初期化処理	第1の 初期化処理	第2の 初期化処理
メインCPUコア	○		
第1のCPUコア	○	○	
第2のCPUコア	○		○
CANコントローラ	○		
第1の仮想CANDライバ (第1のバッファ)	○	○	
第2の仮想CANDライバ (第2のバッファ)	○		○
第1のGPIOピンドライバ	○	○	
第2のGPIOピンドライバ	○		○
第1のアクセス可能領域	○	○	
第2のアクセス可能領域	○		○
第1の制御ソフトウェア	○	○	
第2の制御ソフトウェア	○		○

[図17]

図 1 7



INTERNATIONAL SEARCH REPORT

International application No.

PCT/JP2021/033207

A. CLASSIFICATION OF SUBJECT MATTER**B60R 16/02**(2006.01)i; **G06F 11/16**(2006.01)i; **G06F 11/20**(2006.01)i

FI: G06F11/16 616; B60R16/02 660H; G06F11/20 697

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)

B60R16/02; G06F11/16; G06F11/20

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

Published examined utility model applications of Japan 1922-1996

Published unexamined utility model applications of Japan 1971-2021

Registered utility model specifications of Japan 1996-2021

Published registered utility model applications of Japan 1994-2021

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 2012-198806 A (DENSO CORP) 18 October 2012 (2012-10-18) entire text, all drawings	1-11
A	JP 2018-100091 A (NSK LTD) 28 June 2018 (2018-06-28) entire text, all drawings	1-11
A	JP 2006-39768 A (MITSUBISHI ELECTRIC CORP) 09 February 2006 (2006-02-09) entire text, all drawings	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

10 November 2021

Date of mailing of the international search report

30 November 2021

Name and mailing address of the ISA/JP

Japan Patent Office (ISA/JP)
3-4-3 Kasumigaseki, Chiyoda-ku, Tokyo 100-8915
Japan

Authorized officer

Telephone No.

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.

PCT/JP2021/033207

Patent document cited in search report			Publication date (day/month/year)	Patent family member(s)	Publication date (day/month/year)
JP	2012-198806	A	18 October 2012	DE 102012204644 A1 entire text, all drawings	
JP	2018-100091	A	28 June 2018	(Family: none)	
JP	2006-39768	A	09 February 2006	(Family: none)	

A. 発明の属する分野の分類（国際特許分類（IPC）） B60R 16/02(2006.01)i; G06F 11/16(2006.01)i; G06F 11/20(2006.01)i FI: G06F11/16 616; B60R16/02 660H; G06F11/20 697														
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（IPC）） B60R16/02; G06F11/16; G06F11/20														
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 <table border="0"> <tr> <td>日本国実用新案公報</td> <td>1922 - 1996年</td> </tr> <tr> <td>日本国公開実用新案公報</td> <td>1971 - 2021年</td> </tr> <tr> <td>日本国実用新案登録公報</td> <td>1996 - 2021年</td> </tr> <tr> <td>日本国登録実用新案公報</td> <td>1994 - 2021年</td> </tr> </table>			日本国実用新案公報	1922 - 1996年	日本国公開実用新案公報	1971 - 2021年	日本国実用新案登録公報	1996 - 2021年	日本国登録実用新案公報	1994 - 2021年				
日本国実用新案公報	1922 - 1996年													
日本国公開実用新案公報	1971 - 2021年													
日本国実用新案登録公報	1996 - 2021年													
日本国登録実用新案公報	1994 - 2021年													
国際調査で使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 2012-198806 A（株式会社デンソー）18.10.2012（2012 - 10 - 18） 全文，全図	1-11												
A	JP 2018-100091 A（日本精工株式会社）28.06.2018（2018 - 06 - 28） 全文，全図	1-11												
A	JP 2006-39768 A（三菱電機株式会社）09.02.2006（2006 - 02 - 09） 全文，全図	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。														
<table border="0"> <tr> <td>* 引用文献のカテゴリー</td> <td>“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抵触するものではなく、発明の原理又は理論の理解のために引用するもの</td> </tr> <tr> <td>“A” 特に関連のある文献ではなく、一般的な技術水準を示すもの</td> <td>“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</td> </tr> <tr> <td>“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</td> <td>“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</td> </tr> <tr> <td>“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）</td> <td>“&” 同一パテントファミリー文献</td> </tr> <tr> <td>“O” 口頭による開示、使用、展示等に言及する文献</td> <td></td> </tr> <tr> <td>“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</td> <td></td> </tr> </table>			* 引用文献のカテゴリー	“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抵触するものではなく、発明の原理又は理論の理解のために引用するもの	“A” 特に関連のある文献ではなく、一般的な技術水準を示すもの	“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	“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	“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	“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）	“&” 同一パテントファミリー文献	“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	
* 引用文献のカテゴリー	“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抵触するものではなく、発明の原理又は理論の理解のために引用するもの													
“A” 特に関連のある文献ではなく、一般的な技術水準を示すもの	“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													
“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	“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													
“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）	“&” 同一パテントファミリー文献													
“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														
国際調査を完了した日 10.11.2021	国際調査報告の発送日 30.11.2021													
名称及びあて先 日本国特許庁(ISA/JP) 〒100-8915 日本国 東京都千代田区霞が関三丁目4番3号	権限のある職員（特許庁審査官） 井上 宏一 5B 4177 電話番号 03-3581-1101 内線 3544													

国際調査報告
パテントファミリーに関する情報

国際出願番号

PCT/JP2021/033207

引用文献			公表日	パテントファミリー文献	公表日
JP	2012-198806	A	18.10.2012	DE 102012204644 A1 全文, 全図	
JP	2018-100091	A	28.06.2018	(ファミリーなし)	
JP	2006-39768	A	09.02.2006	(ファミリーなし)	