

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第6821497号
(P6821497)

(45) 発行日 令和3年1月27日(2021.1.27)

(24) 登録日 令和3年1月8日(2021.1.8)

(51) Int.Cl. F I
G05B 19/05 (2006.01) G05B 19/05 A

請求項の数 8 (全 17 頁)

<p>(21) 出願番号 特願2017-88962 (P2017-88962) (22) 出願日 平成29年4月27日 (2017. 4. 27) (65) 公開番号 特開2018-185772 (P2018-185772A) (43) 公開日 平成30年11月22日 (2018. 11. 22) 審査請求日 令和1年12月27日 (2019. 12. 27)</p>	<p>(73) 特許権者 502129933 株式会社日立産機システム 東京都千代田区神田練堀町3番地 (74) 代理人 110002365 特許業務法人サンネクスト国際特許事務所 (72) 発明者 天野 隆 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 (72) 発明者 今井 光洋 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 審査官 山村 秀政</p>
---	--

最終頁に続く

(54) 【発明の名称】 産業用コントローラのプログラム開発システム及び産業用コントローラのプログラム開発システムにおけるデータ共有方法

(57) 【特許請求の範囲】

【請求項1】

制御対象機器のための制御プログラムと所定の処理を実行する情報プログラムとが共有変数を用いて共有記憶領域でデータを共有する産業用コントローラと、前記情報プログラムを開発可能な情報プログラム開発コンピュータと、前記制御プログラムを開発可能な制御プログラム開発コンピュータとを有する、産業用コントローラのプログラム開発システムであって、

前記制御プログラム開発コンピュータは、

前記制御プログラムに含まれる前記共有変数の変数情報を抽出し、前記共有記憶領域に関するマップファイルと、前記共有記憶領域へのアクセスの際に使用されるヘッダファイルとを作成するファイル作成部を備え、

前記情報プログラム開発コンピュータは、

前記ヘッダファイルを用いて前記情報プログラムを生成するプログラム作成部を備え、

前記産業用コントローラが実行する前記情報プログラムは、

前記マップファイルを用いて前記共有記憶領域にアクセスする

ことを特徴とするプログラム開発システム。

【請求項2】

前記プログラム作成部は、

前記共有記憶領域にアクセスするための共有アクセス部を作成して前記情報プログラムを生成し、

前記情報プログラムは、
前記共有アクセス部を呼び出して前記共有記憶領域へのアクセスを行う
ことを特徴とする請求項 1 に記載のプログラム開発システム。

【請求項 3】

前記ヘッダファイルは、
前記情報プログラムが前記共有記憶領域から読み込んだデータが格納される読み込み用
変数宣言と、
前記情報プログラムが前記共有記憶領域に書き込むデータが格納される書き込み用変数
宣言と、

前記情報プログラムが前記共有変数に対してアクセスするためのアクセス関数と、
を含むことを特徴とする請求項 1 に記載のプログラム開発システム。

10

【請求項 4】

前記マップファイルは、
少なくとも 1 つの前記共有記憶領域を互いに識別するための共有記憶領域名と、
少なくとも 1 つの前記共有記憶領域のサイズと、
少なくとも 1 つの前記共有記憶領域の先頭アドレスと、前記共有アクセス部が読み込む
ためのデータが格納されている領域の先頭アドレスとの差である読み込みオフセットと、
少なくとも 1 つの前記共有記憶領域の先頭アドレスと、前記共有アクセス部がデータを
書き込むための領域の先頭アドレスとの差である書き込みオフセットと、

前記共有変数のうち読み込み変数に関する読み込み変数情報と、

20

前記共有変数のうち書き込み変数に関する書き込み変数情報と、

を含むことを特徴とする請求項 2 に記載のプログラム開発システム。

【請求項 5】

前記制御プログラム開発コンピュータは、

前記共有変数に関する設定を受付可能な画面を表示することを特徴とする請求項 1 に記
載のプログラム開発システム。

【請求項 6】

前記情報プログラムと前記制御プログラムとを隔離するコンテナを備えることを特徴と
する請求項 1 に記載のプログラム開発システム。

【請求項 7】

制御対象機器のための制御プログラムと所定の処理を実行する情報プログラムとが共有
変数を用いて共有記憶領域でデータを共有する産業用コントローラと、前記情報プログラ
ムを開発可能な情報プログラム開発コンピュータと、前記制御プログラムを開発可能な制
御プログラム開発コンピュータとを有する、産業用コントローラのプログラム開発システ
ムにおけるデータ共有方法であって、

30

前記制御プログラム開発コンピュータが、前記制御プログラムに含まれる前記共有変数
の変数情報を抽出し、前記共有記憶領域に関するマップファイルと、前記共有記憶領域へ
のアクセスの際に使用されるヘッダファイルとを作成するファイル作成ステップと、

前記情報プログラム開発コンピュータが前記ヘッダファイルを用いて前記情報プログラ
ムを生成するプログラム作成ステップと、

40

前記産業用コントローラが実行する前記情報プログラムが前記マップファイルを用いて
前記共有記憶領域にアクセスするデータアクセスステップと、

を有することを特徴とするデータ共有方法。

【請求項 8】

前記プログラム作成ステップでは、

前記情報プログラム開発コンピュータが、前記共有記憶領域にアクセスするための共有
アクセス部を作成して前記情報プログラムを生成し、

前記産業用コントローラが実行する前記情報プログラムは、

前記共有アクセス部を呼び出して前記共有記憶領域へのアクセスを行う

ことを特徴とする請求項 7 に記載のデータ共有方法。

50

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、いわゆるプログラマブルロジックコントローラを組み込んだ機器としての産業用コントローラに関し、特に、共有記憶領域においてデータを共有する技術に関する産業用コントローラにおけるデータ共有方法に適用して好適なものである。

【背景技術】

【0002】

一般的に、プログラマブルロジックコントローラにおいて高レベル言語プログラム（以下「情報プログラム」という）を実行しようとした場合、情報プログラムは、モータなどのデバイスのアドレスが不明であるため、このようなデバイスを認識することができない。そのため、従来は、手動でデバイスのアドレスを割り当てることにより、プログラマブルロジックコントローラのデバイスと周辺機器との間において通信を可能としている（特許文献1参照）。

10

【先行技術文献】

【特許文献】

【0003】

【特許文献1】特許5079166号公報

【発明の概要】

【発明が解決しようとする課題】

20

【0004】

しかしながら、デバイスの数が膨大な場合、手動によるそのような割り当て作業自体がそもそも困難となる。

【0005】

本発明は以上の点を考慮してなされたもので、手動による割り当て作業を必要とせず開発効率を向上することができる産業用コントローラ及び産業用コントローラにおけるデータ共有方法を提案しようとするものである。

【課題を解決するための手段】

【0006】

かかる課題を解決するため、本発明においては、制御対象機器のための制御プログラムと、所定の処理を実行する情報プログラムとが共有変数を用いて共有記憶領域でデータを共有する産業用コントローラであって、前記制御プログラムに含まれる前記共有変数の変数情報を抽出し、前記共有記憶領域に関するマップファイルと、前記共有記憶領域へのアクセスの際に使用されるヘッダファイルとを作成するファイル作成部と、前記ヘッダファイルを用いて前記情報プログラムを生成するプログラム作成部と、を備え、前記情報プログラムは、前記マップファイルを用いて前記共有記憶領域にアクセスすることを特徴とする。

30

【0007】

また、本発明においては、制御対象機器のための制御プログラムと、所定の処理を実行する情報プログラムとが共有変数を用いて共有記憶領域でデータを共有する産業用コントローラにおけるデータ共有方法であって、前記産業用コントローラが、前記制御プログラムに含まれる前記共有変数の変数情報を抽出し、前記共有記憶領域に関するマップファイルと、前記共有記憶領域へのアクセスの際に使用されるヘッダファイルとを作成するファイル作成ステップと、前記産業用コントローラが前記ヘッダファイルを用いて前記情報プログラムを生成するプログラム作成ステップと、前記情報プログラムが前記マップファイルを用いて前記共有記憶領域にアクセスするデータアクセスステップと、を有することを特徴とする。

40

【発明の効果】

【0008】

本発明によれば、手動による割り当て作業を必要とせず開発効率を向上することができ

50

る。

【図面の簡単な説明】

【0009】

【図1】第1の実施の形態による産業用コントローラ及びプログラム開発環境のハードウェア構成の一例を示す図である。

【図2】第1の実施の形態による産業用コントローラ及びプログラム開発環境のソフトウェア構成の一例を示す図である。

【図3】第1の実施の形態による共有変数の設定画面の一例を示す図である。

【図4】第1の実施の形態によるヘッダファイルの一例を示す図である。

【図5】第1の実施の形態によるマップファイルの一例を示す図である。

10

【図6】第1の実施の形態によるコンテナの一例を示すイメージ図である。

【図7】第1の実施の形態による制御プログラム関連の処理の一例を示すフローチャートである。

【図8】第1の実施の形態による情報プログラム関連の処理の一例を示すフローチャートである。

【図9】第2の実施の形態による制御プログラム関連の処理の一例を示すフローチャートである。

【発明を実施するための形態】

【0010】

以下、図面について、本発明の一実施の形態について詳述する。

20

【0011】

(1) 第1の実施の形態

(1-1) ハードウェア構成

図1は、第1の実施の形態による産業用コントローラ（以下「コントローラ」とも呼ぶ）103及びプログラム開発環境のハードウェア構成の一例を示す。制御プログラム開発PC101、情報プログラム開発PC102、及び産業用コントローラ（以下単に「コントローラ」という）103は、基本的にほぼ同様のハードウェア構成を有し、後述するように目的に応じたソフトウェアが動作している。

【0012】

ネットワーク105は、制御プログラム開発PC101、情報プログラム開発PC102及びコントローラ103を接続する。ネットワーク106は、コントローラ103及び制御対象機器104を接続する。

30

【0013】

制御プログラム開発PC101は、制御プログラムを開発するためのコンピュータである。制御プログラム開発PC101は、EPROM107、CPU108、メインメモリ109、周辺制御装置110、ネットワークI/F111、不揮発性記憶装置112及びユーザI/F装置113を備えている。

【0014】

周辺制御装置110は、バス114を介して各種装置を相互に接続する。ネットワークI/F111は、ネットワーク105に接続するためのインターフェースである。不揮発性記憶装置112は、各種プログラムやファイルを格納する装置である。ユーザI/F装置113は、ユーザが必要なデータの入力等を行う装置である。

40

【0015】

情報プログラム開発PC102は、情報プログラムを開発するためのコンピュータである。情報プログラム開発PC102は、EPROM115、CPU116、メインメモリ117、周辺制御装置118、ネットワークI/F119、不揮発性記憶装置120及びユーザI/F装置121を備えている。

【0016】

周辺制御装置118は、バス132を介して各種装置を相互に接続する。ネットワークI/F119は、ネットワーク105に接続するためのインターフェースである。不揮発

50

性記憶装置 120 は、各種プログラムやファイルを格納する装置である。ユーザ I/F 装置 121 は、ユーザが必要なデータの入力等を行う装置である。

【0017】

コントローラ 103 は、EPROM 122、CPU 123、メインメモリ 124、周辺制御装置 126、ネットワーク I/F 125、ネットワーク I/F 127、不揮発性記憶装置 128 及びユーザ I/F 装置 129 を備えている。

【0018】

周辺制御装置 126 は、バス 133 を介して各種装置を相互に接続する。ネットワーク I/F 125、127 は、各々対応するネットワーク 105、106 に接続するためのインターフェースである。不揮発性記憶装置 128 は、各種プログラムやファイルを格納する装置である。ユーザ I/F 装置 129 は、ユーザが必要なデータの入力等を行う装置である。

【0019】

コントローラ 103 では、EPROM 122 又は不揮発性記憶装置 128 に格納されているオペレーティングシステム (OS)、各種プログラム及び各種ファイルをメインメモリ 124 に展開することにより、CPU 123 を用いて各種の演算等を実行する。

【0020】

制御対象機器 104 は、センサ 130 及びアクチュエータ 131 を含んでいる。センサ 130 は、例えば、手動スイッチ、リレー、光電スイッチ、近接スイッチ、リミットスイッチ、重量計、温度センサ、圧力センサ、振動センサ、音波センサ、流量計、流速計、ガスセンサ、速度センサ、回転計、電流計、電圧計、電力計、水質センサ、色判別センサ、またはカメラのようなデバイスである。

【0021】

一方、アクチュエータ 131 は、例えば、モータ、発電機、ヒーター、制御弁、シリンダ、ソレノイド、ランプ、表示器、ブザー、スピーカー、ポンプ、圧縮機、空調機、冷凍機、またはコンベアのようなデバイスである。

【0022】

(1-2) ソフトウェア構成

図 2 は、第 1 の実施の形態によるコントローラ 103 及びプログラム開発環境のソフトウェア構成の一例を示す。なお、図示の例では、図 1 における制御対象機器 104 が省略されている。

【0023】

本実施の形態では、制御プログラム 207 を実行するタスク処理の最後に共有アクセス部 208 が共有記憶領域 224 にアクセスする場合を例示している。なお、制御プログラムにグローバル変数が含まれる場合は、最高優先度のタスクがこのグローバル変数を共有記憶領域 224 に書き込む。

【0024】

(1-2-1) 制御プログラム開発 PC

まず、制御プログラム開発 PC 101 について説明する。ソースファイル 201 は、制御プログラム 207 のソースコードが記述されたファイルである。エディタ 202 は、ソースファイル 201 を作成するためのプログラムである。コンパイラ 203 は、ソースファイル 201 から中間言語ファイルを作成する。リンカ 204 は、中間言語ファイルとライブラリ 205 を結合して実行ファイルを生成する。ライブラリ 205 は、プログラムの関数を再利用するために予め準備されている中間言語ファイルである。

【0025】

IDE 制御部 206 は、エディタ 202、コンパイラ 203 及びリンカ 204 等を制御するプログラムである。なお、エディタ 202、コンパイラ 203、リンカ 204、ライブラリ 205 及び IDE 制御部 206 をまとめて、統合開発環境プログラム (IDE: Integrated Development Environment) と呼ぶ。

【0026】

10

20

30

40

50

さらに制御プログラム開発PC101は、共有アクセス部208、ヘッダファイル209、マップファイル210、共有領域アクセス部211及び連携ファイル生成部212を備えている。共有領域アクセス部211は、制御プログラム207が共有記憶領域224へアクセスするための共有アクセス部208のソースファイルである。連携ファイル生成部212は、ヘッダファイル209及びマップファイル210を生成する。

【0027】

(1-2-2) 情報プログラム開発コンピュータ

次に、情報プログラム開発PC102について説明する。ソースファイル213は、情報プログラム219のソースコードが記述されたファイルである。エディタ214は、ソースファイル213を作成するためのプログラムである。コンパイラ215は、ソースファイル213から中間言語ファイルを作成する。リンカ216は、中間言語ファイルとライブラリ217を結合して実行ファイルを生成する。ライブラリ217は、プログラムの関数を再利用するために予め準備されている中間言語ファイルである。

10

【0028】

なお、上述したコントローラ103においては、複雑な算術演算、SCADA、MES、クラウドシステムとの情報の送受信のようなデータ処理が必要な場合がある。このようなデータ処理は、前述の制御装置特有の言語でプログラミングすることが困難な場合があり、C言語やJava(登録商標)言語などの情報通信分野で使われるプログラミング言語を用いて記述する場合がある。このようなデータ処理を実行するプログラムを「情報プログラム」と呼ぶ。この情報プログラムは、例えば情報プログラム219に相当する。

20

【0029】

IDE制御部218は、エディタ214、コンパイラ215及びリンカ216等を制御するプログラムである。なお、エディタ214、コンパイラ215、リンカ216、ライブラリ217及びIDE制御部218をまとめて、統合開発環境プログラムと呼ぶ。

【0030】

さらに情報プログラム開発PC102は、共有アクセス部220、ヘッダファイル209及び連携ファイルアクセス部222を備えている。共有アクセス部220は、情報プログラム219が共有記憶領域224へアクセスするためのプログラムである。連携ファイルアクセス部222は、ヘッダファイル209をコントローラ103から取得したり、情報プログラム219と共有アクセス部220をコントローラへ送信したりする。

30

【0031】

(1-2-3) コントローラ

一方、コントローラ103は、シーケンス制御装置、モーション制御装置またはプログラマブル・ロジック・コントローラ(PLC)とも呼ばれる。コントローラ103は、ラダー・ロジック(LD言語)、シーケンシャル・ファンクション・チャート(FC言語)、ファンクション・ブロック(FBD言語)、ストラクチャード・テキスト(ST言語)、またはインストラクション・リスト(LD言語)のような制御装置特有のプログラミング言語で制御内容が記述されたプログラムを実行する。

【0032】

このような制御内容が実行可能なプログラムを「制御プログラム」と呼ぶ。制御プログラム207においては一連の処理をタスクと呼び、タスクごとに繰り返し処理する周期を設定できる。制御プログラム207において使用する変数には、入力専用変数及び出力専用変数が存在する。コントローラ103は、このような制御プログラム207のみならず、情報プログラム219も実行する。

40

【0033】

共有アクセス部208は、制御プログラム207が共有記憶領域224へアクセスするためのプログラムである。ライブラリ223は、制御プログラム207や共有アクセス部208が実行するために必要な共有関数の処理を実行する。ライブラリ226は、情報プログラム219が共有記憶領域224にアクセスするために、マップファイル210を参照してアクセス関数の処理を実行する。構成アクセス部229は、連携ファイルアクセス

50

部 2 2 2 へヘッダファイル 2 0 9 を送信したり、連携ファイルアクセス部 2 2 2 から情報プログラム 2 1 9 と共有アクセス部 2 2 0 を受信したりする。タスクスケジューラ 2 2 8 は、タスクを実行する。タスクは、設定されている周期でそのタスクに登録されているプログラムを実行する。タスクに制御プログラム 2 0 7 と共有アクセス部 2 0 8 が登録されている場合、タスクは、実行周期に合わせて制御プログラム 2 0 7 と共有アクセス部 2 0 8 を実行する。共有記憶領域 2 2 4 は、制御プログラム 2 0 7 と情報プログラム 2 1 9 が共有する変数のデータを保存する。共有アクセス部 2 2 0 は、ライブラリ 2 2 6 を経由して共有記憶領域 2 2 4 にアクセスし、例えば、情報プログラム 2 1 9 の読み込み用構造体 4 0 1 にデータを読み込んだり、書き込み用構造体 4 0 2 のデータを共有記憶領域 2 2 4 の連続したアドレスの領域に書き込んだりする。

10

【 0 0 3 4 】

ヘッダファイル 2 0 9 は、連携ファイル生成部 2 1 2 が生成する情報プログラム 2 1 9 を作成する際に用いられるヘッダファイルである。マップファイル 2 1 0 は、連携ファイル生成部 2 1 2 が生成する情報プログラム 2 1 9 が共有記憶領域 2 2 4 へアクセスするために使用するファイルである。

【 0 0 3 5 】

コンテナ処理部 2 2 7 は、制御プログラム 2 0 7 と情報プログラム 2 1 9 とが互いに影響を及ぼさないようにするためのプログラムである。

【 0 0 3 6 】**(1 - 3) 共有変数の設定画面**

図 3 は、第 1 の実施の形態による共有変数の設定画面としての共有変数設定部 3 0 1 の一例を示す。なお、この共有変数設定部 3 0 1 は、制御プログラム開発 P C 1 0 1 によって表示される、設定受け付け画面である。

20

【 0 0 3 7 】

共有変数設定部 3 0 1 は、情報プログラム 2 1 9 と共有する変数を設定するための G U I (Graphic Use Interface) で、連携ファイル生成部 2 1 2 が生成する。共有チェック欄 3 0 2 は、チェックした場合には、変数名 3 0 3 に示す変数について情報プログラム 2 1 9 と変数を共有し、チェックしない場合にはそれについて情報プログラム 2 1 9 と変数共有しないことを表す。変数名 3 0 3 は、情報プログラムと共有する変数名である。型 3 0 4 は、変数の型である。周期 3 0 5 は、共有記憶領域 2 2 4 において共有される変数データを保存する周期を設定するための欄である。

30

【 0 0 3 8 】

キャンセルボタン 3 0 6 は、押下すると、共有変数設定部 3 0 1 での作業を破棄し終了する一方、OK ボタン 3 0 7 は、押下すると、設定された情報を基にヘッダファイル 2 0 9 及びマップファイル 2 1 0 を生成する。アクセス権限 3 0 8 は、情報プログラムに付与するアクセス権限を表している。情報プログラムは、このアクセス権限 3 0 8 が「Read」であれば、変数名 3 0 3 に対応する変数のデータを読み込むことができる一方、「Write」であればそれを書き込むこともできる。

【 0 0 3 9 】**(1 - 4) ヘッダファイル**

図 4 は、情報プログラム 2 1 9 を作成するためのヘッダファイル 2 0 9 の一例を示す。読み込み用構造体 4 0 1 は、情報プログラム 2 1 9 が共有記憶領域 2 2 4 の読み込みオフセット値 (読み込み領域) から読み込んだデータが格納される構造体である。書き込み用構造体 4 0 2 は、情報プログラム 2 1 9 が共有記憶領域 2 2 4 の書き込みオフセット値 (書き込み領域) に書き込むデータが格納される構造体である。ヘッダファイルには、構造体に限らずプログラム言語で使用可能な変数の宣言を記載してもよい。共有すべき変数がない場合は、対象となるアクセス用変数の宣言は記載されない。

40

【 0 0 4 0 】

全体 read アクセス関数 4 0 3 は、情報プログラム 2 1 9 がすべての読み込み用変数のデータを読み込むための関数である。関数の呼び出し時に共有記憶領域 2 2 4 の読み込み領

50

域からprDataにデータを読み込む。全体writeアクセス関数404は、情報プログラム219がすべての書き込み用変数のデータを書き込むための関数である。関数の呼び出し時にpwDataから共有記憶領域224の書き込み領域にデータを書き込む。全体writeのreadアクセス関数405は、情報プログラム219がすべての書き込み用変数のデータを読み込むための関数である。readアクセス関数405は、関数の呼び出し時に共有記憶領域224の書き込み領域からpwDataにデータを読み込む。個別readアクセス関数406は、情報プログラム219が個々の読み込み用変数のデータを読み込むための関数である。個別writeアクセス関数407は、情報プログラム219が個々の書き込み用変数のデータを書き込み領域に書き込むための関数である。個別writeのreadアクセス関数408は、情報プログラム219が個々の書き込み用変数のデータを書き込み領域から読み込むための関数である。

10

【0041】

(1-5) マップファイル

図5は、マップファイル210をCSV形式で生成した一例である。マップファイル210は、共有アクセス部220が共有記憶領域224にアクセスする際に使用される。

【0042】

共有記憶領域名501は、共有記憶領域224を識別するための名称である。例えばOSがLinux(登録商標)であれば共有記憶領域としてファイルを作成するため、そのファイルの保存位置を表す絶対パスを共有記憶領域名501とする。

【0043】

サイズ502は、共有記憶領域224で保存しているデータのサイズである。読み込みオフセット503は、共有記憶領域224の先頭アドレスと、共有アクセス部220が読み込むためのデータが格納されている領域の先頭アドレスとの差を表す。

20

【0044】

書き込みオフセット504は、共有記憶領域224の先頭アドレスと、共有アクセス部220がデータを書き込むための領域の先頭アドレスとの差を表す。型505は、共有記憶領域224に書き込む制御プログラム207で使用しているデータの型である。

【0045】

変数名506は、共有記憶領域224に書き込むデータが格納されている、制御プログラム207で使用している変数の名前である。変数オフセット507は、共有アクセス部220が読み込むためのデータが格納されている領域の先頭アドレスと、対象の変数のデータが格納されている領域の先頭アドレスとの差を表す。型508は、共有記憶領域224から読み込む制御プログラム207で使用しているデータの型である。

30

【0046】

変数名509は、共有記憶領域224から読み込んだデータを格納する制御プログラム207で使用している変数の名称である。変数オフセット510は、共有アクセス部220が読み込むためのデータが格納されている領域の先頭アドレスと、対象の変数のデータが格納されている領域の先頭アドレスとの差を表す。型505から変数オフセット510の情報は、個別の変数のみにアクセスする場合や、必要に応じて構造体のアライメントの調整に使用する。例えば、複数の情報プログラム219がありプログラミング言語が異なる場合、構造体のメンバが保存される領域の構造がマップファイル210と異なる場合がある。その場合、対象言語用に作成されるライブラリ226は、対象言語の構造体のアライメントに合わせてオフセット値を算出し、共有記憶領域224にアクセスする。型のサイズは、ライブラリ226で型名とサイズの変換テーブルを保持していてもよいし、マップファイル210に記載しておいてもよい。

40

【0047】

図6は、制御プログラム207用のコンテナ及び情報プログラム219用のコンテナのイメージ図である。図示の例では、図2の各構成と同一の符号が付してある構成は、ほぼ同様の機能を有しているため、以下では異なる点を中心として説明する。

【0048】

50

コンテナ処理部 227 は、制御プログラム 207 と情報プログラム 219 とが互いに影響を及ぼさないようにするために、制御プログラム 207 及び情報プログラム 219 のためのコンテナを作成する。

【0049】

具体的には、コンテナ処理部 227 は、制御プログラム 207 と情報プログラム 219 とがそれぞれ専用で使用するプロセス、ファイルシステム及びネットワークの名前空間と、CPU 123 とメインメモリ 124 とのリソースを作成する。

【0050】

本実施の形態では、制御プログラム 207 用のコンテナを「制御コンテナ」と呼ぶ一方、情報プログラム 219 用のコンテナを「情報コンテナ」と呼ぶことにする。コンテナの具体的な作成方法については後述する。

10

【0051】

図 7 は、制御プログラム 207 に関連する処理の一例を示す。まず、ステップ S600 では、コンテナ処理部 227 が、図 6 に示すように、制御プログラム 207 用の制御コンテナ及び情報プログラム 219 用の情報コンテナを作成する。

【0052】

図 6 コンテナ処理部 227 は、共有記憶領域 224 が作成されるディレクトリを制御コンテナ及び情報コンテナのファイルシステムにマウント（図示の点線に相当）して共有ディレクトリを作成し、制御プログラム 207 と情報プログラム 219 からアクセスできるようにする。また、コンテナ処理部 227 は、ヘッダファイル 209 とマップファイル 210 を保存するための共有ディレクトリと、情報プログラム 219 及び共有アクセス部 220 を保存するための共有ディレクトリも作成する。その際、コンテナ処理部 227 は、後者の共有ディレクトリについては、情報コンテナのファイルシステムにのみマウントして共有する。制御プログラム 207 は制御コンテナ内で実行され、情報プログラム 219 は情報コンテナ内で実行される。

20

【0053】

ステップ S601 では、ユーザが、制御プログラム開発 PC 101 を用いて、既述の IDE を使用してソースコードをまとめて管理するためのプロジェクトを作成し、制御プログラム 207 を作成する。ステップ S602 では、ユーザが、共有変数設定部 301 を起動するために、既述の IDE の該当するメニューを選択する。選択されると、連携ファイル生成部 212 がソースファイル 201 から変数を抽出する。

30

【0054】

ステップ S603 では、連携ファイル生成部 212 が、共有変数設定部 301（図 3 参照）を生成して表示する。ユーザは、情報プログラム 219 と共有する変数の共有チェック欄 302 にチェックし、アクセス権限 308 及び周期 305 を設定する。設定が完了した後、ユーザが OK ボタン 307 を押下すると、連携ファイル生成部 212 は、設定内容をメモリに保存する。

【0055】

ステップ S604 では、連携ファイル生成部 212 が、設定された値をメモリから取得する。

40

【0056】

ステップ S605 では、連携ファイル生成部 212 がヘッダファイル 209 及びマップファイル 210 を生成する。連携ファイル生成部 212 は、アクセス権限 308 が「Read」に設定された変数を読み込み用構造体 401 としてヘッダファイル 209 に定義する。次に連携ファイル生成部 212 は、アクセス権限 308 が「Write」に設定された変数を書き込み用構造体 402 としてヘッダファイル 209 に定義する。そして、連携ファイル生成部 212 は、全体 read アクセス関数 403、全体 write アクセス関数 404、全体 write の read アクセス関数 405、個別 read アクセス関数 406、個別 write アクセス関数 407 および個別 write の read アクセス関数 408 をヘッダファイル 209 に定義する。

【0057】

50

次に連携ファイル生成部 212 は、情報プログラム 219 がヘッダファイル 209 に定義された構造体変数の構造で共有記憶領域 224 にアクセスできるようにマップファイル 210 を生成する。具体的には、連携ファイル生成部 212 は、共有記憶領域名 501、共有記憶領域名で生成するファイルのサイズ 502、読み込みオフセット 503、及び書き込みオフセット 504 をマップファイル 210 に出力する。連携ファイル生成部 212 は、共有記憶領域 224 の先頭アドレスから順に「Read」に設定された変数を保存し、連続して「Write」に設定された変数を保存できるように共有記憶領域 224 にアクセスするためのマップファイル 210 を生成する。連携ファイル生成部 212 は、ヘッダファイル 209 で定義した構造体のサイズからサイズ 502、読み込みオフセット 503、書き込みオフセット 504 を算出する。次に連携ファイル生成部 212 は、読み込み用構造体 401 定義されている変数の型 505、変数名 506、変数オフセット 507 をマップファイル 210 に出力する。次に連携ファイル生成部 212 は、書き込み用構造体 402 定義されている変数の型 508、変数名 509、変数オフセット 510 をマップファイル 210 に出力する。連携ファイル生成部 212 は、変数オフセット (507 と 510) を変数の定義順序、変数のサイズ、および情報プログラム 219 の構造体のアライメントを考慮して算出して出力する。

10

【0058】

ステップ S606 では、連携ファイル生成部 212 は、共有記憶領域 224 の名前を決定して共有記憶領域 224 を作成し、この共有記憶領域 224 で共有変数のデータを保存する領域にアクセスできるソースコード (共有領域アクセス部 211) を生成する。具体的には、連携ファイル生成部 212 は、ヘッダファイル 209 に定義する構造体の変数の順番、マップファイル 210 に出力したオフセットを基にして、共有変数設定部 301 において、「Read」に設定された読み込み用変数のデータを共有記憶領域 224 に書き込むソースコードと、「Write」に設定された書き込み用変数のデータを共有記憶領域 224 から制御プログラム 207 の変数に読み込むソースコードを生成する。

20

【0059】

連携ファイル生成部 212 は、制御プログラム 207 をタスクに登録し、タスクの実行周期として周期 305 を設定する。タスクの実行周期は、制御プログラム作成者等により手動で設定されてもよい。連携ファイル生成部 212 は、制御プログラム 207 の次に実行されるように共有記憶領域 224 にアクセスする共有アクセス部 208 を同じタスクに登録する。

30

【0060】

ステップ S607 では、ユーザが既述の IDE で実行メニューを選択すると、IDE 制御部 206 はソースコードをビルドし、制御プログラム 207 及び共有アクセス部 208 を生成する。

【0061】

ステップ S608 では、制御プログラム開発 PC 101 において IDE 制御部 206 が、制御プログラム 207 及び共有アクセス部 208 を既述の制御コンテナ内の共有ディレクトリ以外の場所へ送信し、ヘッダファイル 209 及びマップファイル 210 を制御コンテナ内の共有ディレクトリへ送信する。

40

【0062】

ステップ S609 では、IDE 制御部 206 が、コントローラ 103 のタスクスケジューラ 228 を実行する。ステップ S610 では、タスクスケジューラ 228 がタスクを実行し、タスクは設定された実行周期で制御プログラム 207 を実行する。制御プログラム 207 は、制御処理を実行する。一度起動したタスクは、周期的に登録されたプログラムを実行するため再度タスクスケジューラからは実行されない。

【0063】

ステップ S611 では、タスクが制御プログラム 207 の実行後に共有アクセス部 208 を実行する。ステップ S612 では、共有アクセス部 220 が、ライブラリ 223 の処理も活用しつつ共有記憶領域名 501 とサイズ 502 を指定して共有記憶領域 224 をオ

50

オープン（アクセス可能な状態）し、マップファイル 210 に出力した情報を基に Read 権限の変数のデータを書き込む一方、Write 権限の変数のデータを読み込む。次にステップ S610 が実行される。

【0064】

図 8 は、情報プログラム 219 の関連処理の一例を示す。ステップ S701 では、ユーザが既述の IDE を使用してソースコードをまとめて管理するためのプロジェクトを作成するウィザードを実行する。

【0065】

ステップ S702 では、ウィザードが実行されると、IDE 制御部 218 は、連携ファイルアクセス部 222 を実行してコントローラ 103 の構成アクセス部 229 と通信し、ヘッダファイル 209 を取得する。

10

【0066】

ステップ S703 では、連携ファイルアクセス部 222 が、情報プログラム 219 のソースコードとなるファイルのヘッダファイル宣言部の位置にヘッダファイル 209 の宣言を入力してプロジェクトの作成が完了する。

【0067】

ステップ S704 では、ユーザが、情報プログラム 219 と、共有記憶領域 224 へアクセスするための共有アクセス部 220 のソースコードとを作成する。ユーザは、ヘッダファイル 209 に定義された関数を基にして、共有アクセス部 220 のソースコードを作成する。

20

【0068】

ステップ S705 では、ユーザが既述の IDE で実行メニューを選択すると、IDE 制御部 218 はソースコードをビルドし、情報プログラム 219 及び共有アクセス部 220 を生成する。共有アクセス部 220 は、情報プログラム 219 に含まれていてもよい。

【0069】

ステップ S706 では、連携ファイルアクセス部 222 が、情報プログラム 219 及び共有アクセス部 220 を構成アクセス部 229 に送信する。コンテナ処理部 227 は、構成アクセス部 229 が情報プログラム 219 及び共有アクセス部 220 を受信したときに、情報コンテナを作成してもよい。コンテナ処理部 227 は、情報プログラム 219 が終了したときに情報コンテナを削除してもよい。構成アクセス部 229 は、情報プログラム 219 及び共有アクセス部 220 を情報コンテナ内からアクセスできる共有ディレクトリに保存する。本実施の形態では、構成アクセス部 229 を介して情報コンテナ内に情報プログラム 219 と共有アクセス部 220 を保存したが、連携ファイルアクセス部 222 が情報コンテナ内の FTP サーバ等と通信して情報コンテナ内に情報プログラム 219 と共有アクセス部 220 を保存してもよい。ステップ S707 では、IDE 制御部 218 が、例えば SSH 接続で情報コンテナ内にログインして情報プログラム 219 を実行する。

30

【0070】

ステップ S708 では、情報プログラム 219 がデータ処理を実行する。ステップ S709 では、情報プログラム 219 が共有アクセス部 220 を実行する。ステップ S710 では、共有アクセス部 220 がアクセス関数を実行するために、ライブラリ 226 を呼び出す。ライブラリ 226 は、共有ディレクトリからマップファイル 210 を取得する。ライブラリ 226 は、あらかじめマップファイル 210 を取得するためのパス名を保持している。

40

【0071】

ステップ S711 では、共有アクセス部 220 が、ライブラリ 226 経由で共有記憶領域 224 へのアクセスを実行する。

【0072】

(1-6) アクセス方法 (1-6-1) 全体 read アクセス :

ライブラリ 226 は、関数の引数から読み込み用構造体 401 の先頭アドレスを取得する。ライブラリ 226 は、書き込みオフセット 504 と読み込みオフセット 503 の差か

50

ら読み込むデータサイズを算出する。ライブラリ 2 2 6 は、共有記憶領域名 5 0 1 を指定して共有記憶領域 2 2 4 をオープンし、読み込みオフセット 5 0 3 の位置から算出したサイズ分のデータを読み込み、取得した読み込み用構造体 4 0 1 の先頭アドレスから順に読み込んだデータを書き込む。

【 0 0 7 3 】

(1 - 6 - 2) 全体writeアクセス :

ライブラリ 2 2 6 は、関数の引数から書き込み用構造体 4 0 2 の先頭アドレスを取得する。ライブラリ 2 2 6 は、サイズ 5 0 2 と書き込みオフセット 5 0 4 の差から書き込むデータサイズを算出する。ライブラリ 2 2 6 は、共有記憶領域名 5 0 1 を指定して共有記憶領域 2 2 4 をオープンし、書き込み用構造体 4 0 2 の先頭アドレスの位置から算出したサイズ分のデータを読み込み、共有記憶領域 2 2 4 の書き込みオフセット 5 0 3 の位置から読み込んだデータを順に書き込む。

10

【 0 0 7 4 】

(1 - 6 - 3) 全体writeのreadアクセス :

ライブラリ 2 2 6 は、関数の引数から書き込み用構造体 4 0 2 の先頭アドレスを取得する。ライブラリ 2 2 6 は、サイズ 5 0 2 と書き込みオフセット 5 0 4 の差から読み込むデータサイズを算出する。ライブラリ 2 2 6 は、共有記憶領域名 5 0 1 を指定して共有記憶領域 2 2 4 をオープンし、書き込みオフセット 5 0 3 の位置から算出したサイズ分のデータを読み込み、取得した書き込み用構造体 4 0 2 の先頭アドレスから順に読み込んだデータを書き込む。

20

【 0 0 7 5 】

(1 - 6 - 4) 個別readアクセス :

ライブラリ 2 2 6 は、関数の引数から読み込み用構造体 4 0 1 の先頭アドレスと個別変数の先頭アドレスを取得する。ライブラリ 2 2 6 は、取得したアドレスの差を計算して個別変数のオフセット値を算出する。ライブラリ 2 2 6 は、マップファイル 2 1 0 を参照し、一致する変数オフセット 5 0 7 の変数を特定する。ライブラリは、特定した変数の型 5 0 5 からサイズを特定する。ライブラリ 2 2 6 は、共有記憶領域名 5 0 1 を指定して共有記憶領域 2 2 4 をオープンし、読み込みオフセット 5 0 3 の位置から算出したオフセット分だけ離れた位置から特定したサイズ分のデータを読み込み、取得した個別変数の先頭アドレスから順に、読み込んだデータを書き込む。

30

【 0 0 7 6 】

(1 - 6 - 5) 個別writeアクセス :

ライブラリ 2 2 6 は、関数の引数から書き込み用構造体 4 0 2 の先頭アドレスと個別変数の先頭アドレスを取得する。ライブラリ 2 2 6 は、取得したアドレスの差と書き込みオフセット 5 0 4 を足して個別変数のオフセット値を算出する。ライブラリ 2 2 6 は、マップファイル 2 1 0 を参照し、算出したオフセット値と一致する変数オフセット 5 1 0 の変数を特定する。ライブラリ 2 2 6 は、特定した変数の型 5 0 8 からサイズを特定する。ライブラリ 2 2 6 は、共有記憶領域名 5 0 1 を指定して共有記憶領域 2 2 4 をオープンし、個別変数の先頭アドレスから特定したサイズ分のデータを読み込み、共有記憶領域 2 2 4 の先頭アドレスから算出したオフセット分だけ離れた位置から、個別変数の先頭アドレスから読み込んだデータを書き込む。

40

【 0 0 7 7 】

(1 - 6 - 6) 個別writeのreadアクセス :

ライブラリ 2 2 6 は、関数の引数から書き込み用構造体 4 0 2 の先頭アドレスと個別変数の先頭アドレスを取得する。ライブラリ 2 2 6 は、取得したアドレスの差に書き込みオフセット 5 0 4 を足して個別変数のオフセット値を算出する。ライブラリ 2 2 6 は、マップファイル 2 1 0 を参照し、変数オフセット 5 1 0 の差が一致する変数を特定する。ライブラリ 2 2 6 は、特定した変数の型 5 0 8 からサイズを特定する。ライブラリ 2 2 6 は、共有記憶領域名 5 0 1 を指定して共有記憶領域 2 2 4 をオープンし、算出したオフセット分だけ離れた位置から特定したサイズ分のデータを読み込み、取得した個別変数の先頭ア

50

ドレスから順に、読み込んだデータを書き込む。

【0078】

次にステップS708を実行する。共有アクセス部208とライブラリ226が共有記憶領域224に書き込む処理は、書き込み処理が中断されず連続した処理となるアトミック処理とする。

【0079】

(1-7)本実施の形態の作用効果等

以上のような構成によれば、制御プログラム207に含まれる共有変数の変数情報が抽出され、共有記憶領域224に関するマップファイル210と、共有記憶領域224へのアクセスの際に使用されるヘッダファイル209とが作成される。また、ヘッダファイル209を用いて情報プログラム219が生成される。この情報プログラム219は、マップファイル210を用いて共有記憶領域224にアクセスする。このようにすると、制御プログラム207と情報プログラム219との間で同一の共有変数名が自動的に生成されることになり、管理者の手動による、制御プログラム209の制御対象機器であるデバイスのアドレスの割り当て作業を必要としなくなるため、開発効率を向上することができる。

10

【0080】

(2)第2の実施の形態

第2の実施の形態では、第1の実施の形態と同様の構成及び動作については説明を省略し、異なる点を中心として説明する。第2の実施の形態では、第1の実施の形態とは異なり、制御プログラム207のタスクとは別のタスクとして共有アクセス部208が実行され、共有記憶領域224にアクセスする。

20

【0081】

図9は、制御プログラム関連の処理の一例を示す。ステップS801では、連携ファイル生成部212が、共有記憶領域224の名前を決定し、共有記憶領域224を作成し、共有変数のデータを共有記憶領域224にアクセスできるソースコード(共有領域アクセス部211)を生成する。連携ファイル生成部212は、制御プログラム207のタスクとは別の共有アクセスタスクを作成し、実行する周期として共有変数設定部301から取得した周期を設定する。連携ファイル生成部212は、制御プログラム207を実行するタスクの優先度よりも共有アクセスタスクの実行優先度を低く設定する。

30

【0082】

ステップS802では、タスクスケジューラ228が、タスクを実行し、タスクは設定された実行周期で登録されている制御プログラム207を実行する。制御プログラム207は、制御処理を実行する。次にステップS803が実行される。このタスクは、設定された実行周期で制御プログラム207を繰り返し実行する。ステップS803では、タスクスケジューラ228が共有アクセスタスクを実行し、共有アクセスタスクが登録されている共有アクセス部208を実行する。次にステップS612を実行する。この共有アクセスタスクは、設定された実行周期でステップ803とステップ612を繰り返し実行する。

【0083】

以上のような構成によれば、制御プログラム207のタスクとは独立した共有アクセス部208が、制御プログラム207より実行優先度が低い状態で共有記憶領域224にアクセスするため、制御プログラム207の影響を受けにくくすることができる。

40

【0084】

(3)その他の実施形態

上記実施形態は、本発明を説明するための例示であり、本発明をこれらの実施形態にのみ限定する趣旨ではない。本発明は、その趣旨を逸脱しない限り、様々な形態で実施することができる。例えば、上記実施形態では、各種プログラムの処理をシーケンシャルに説明したが、特にこれにこだわるものではない。従って、処理結果に矛盾が生じない限り、処理の順序を入れ替え又は並行動作するように構成しても良い。

50

【産業上の利用可能性】

【0085】

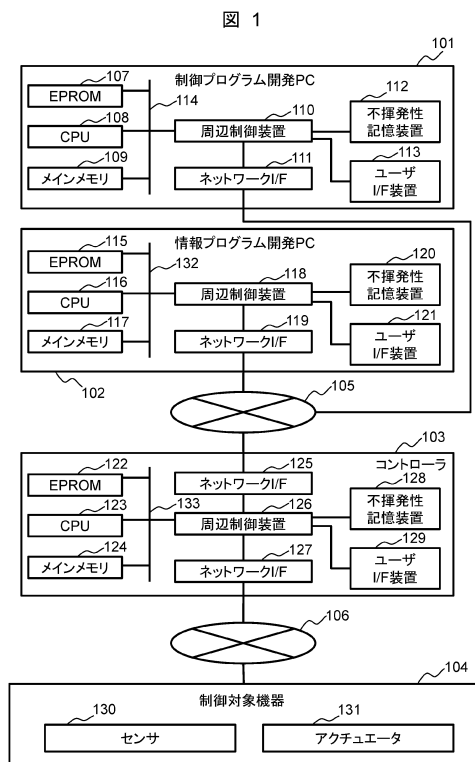
本発明は、いわゆるプログラマブルロジックコントローラを組み込んだ機器としての産業用コントローラ及び産業用コントローラにおけるデータ共有方法に広く適用することができる。

【符号の説明】

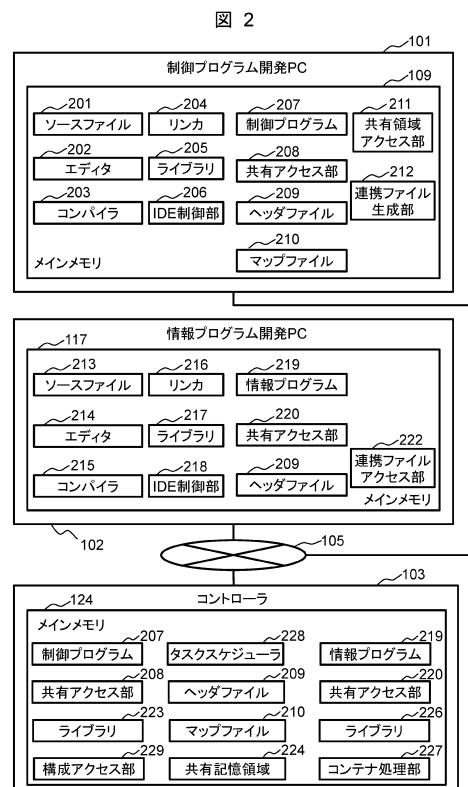
【0086】

101.....制御プログラム開発PC、102.....情報プログラム開発PC、103.....コントローラ、104.....制御対象機器、130.....センサ、131.....アクチュエータ。

【図1】

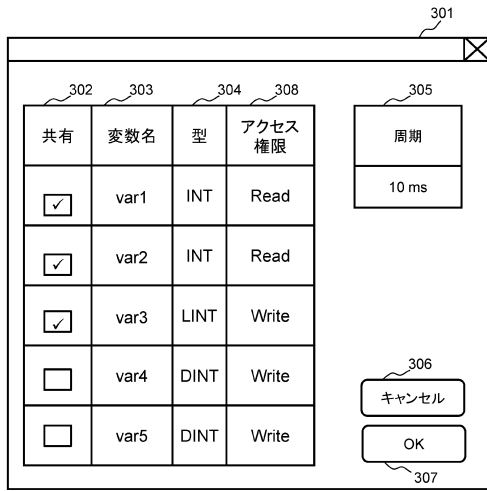


【図2】



【 図 3 】

図 3



【 図 4 】

図 4

```

/* data_sharing.h */
#ifndef __DATA_SHARING_H__
#define __DATA_SHARING_H__

//読み込み用構造体
typedef struct _shared_rdata {
    short var1;
    short var2;
} shared_rdata;

//書き込み用構造体
typedef struct _shared_wdata {
    long long var3;
} shared_wdata;

// 全体readアクセス関数
extern int read_all(shared_rdata* prData);

// 全体writeアクセス関数
extern int write_all(shared_wdata* pwData);

// 全体writeのreadアクセス関数
extern int read_w_all(shared_wdata* pwData);

// 個別readアクセス関数
extern int read(void* prData_var, shared_rdata* prData);

// 個別writeアクセス関数
extern int write(void* pwData_var, shared_wdata* prData);

// 個別writeのreadアクセス関数
extern int read_w(void* pwData_var, shared_wdata* prData);

#endif

```

【 図 5 】

図 5

```

# 共有記憶領域名, サイズ, 読み込みオフセット, 書き込みオフセット
/dev/shm/shmdata, 128, 0, 4

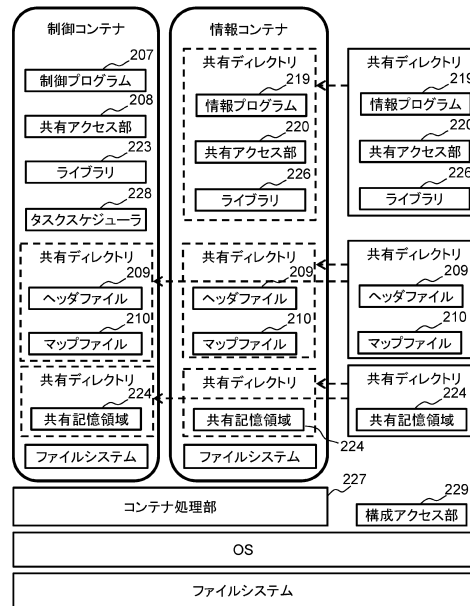
# 読み込み変数
# 505 ~ 506
# 型, 変数名, 変数オフセット
INT, var1, 0
INT, var2, 2

# 書き込み変数
# 508 ~ 509
# 型, 変数名, 変数オフセット
LINT, var3, 4

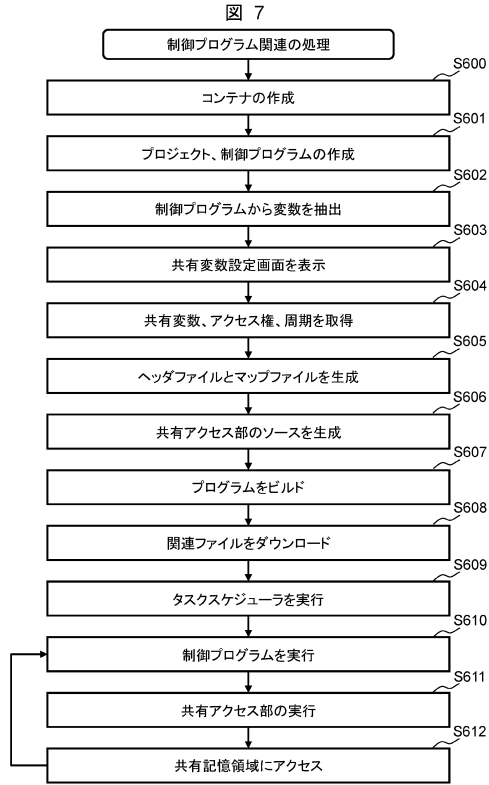
```

【 図 6 】

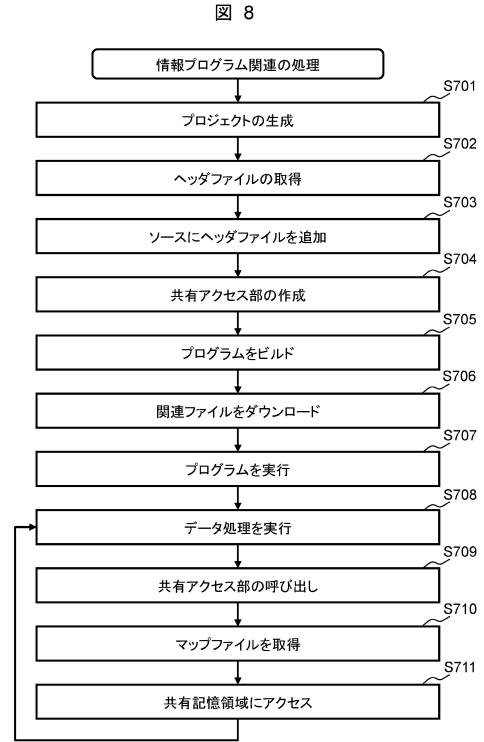
図 6



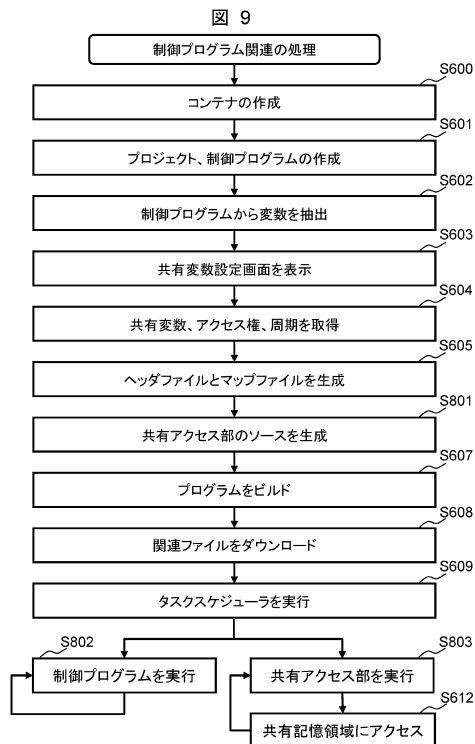
【 図 7 】



【 図 8 】



【 図 9 】



フロントページの続き

(56)参考文献 特開平04 - 003205 (JP, A)
特開2000 - 132208 (JP, A)
特開2015 - 215865 (JP, A)
国際公開第2007 / 110953 (WO, A1)
特許第5079166 (JP, B1)

(58)調査した分野(Int.Cl., DB名)
G05B 19/05