

(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
2012 年 9 月 20 日(20.09.2012)



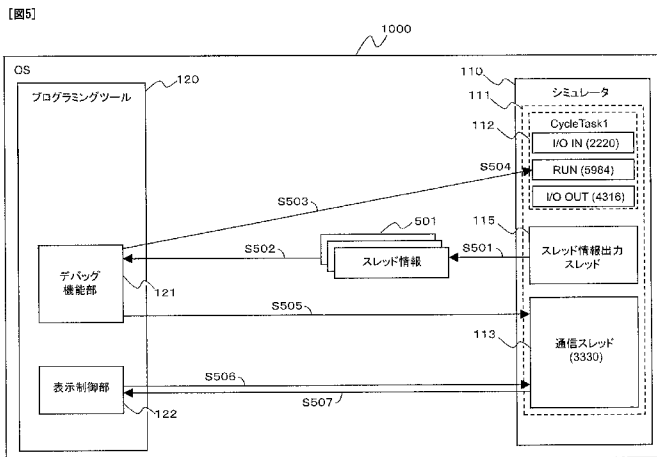
(10) 国際公開番号
WO 2012/124162 A1

- (51) 国際特許分類:
G06F 11/28 (2006.01)
- (21) 国際出願番号: PCT/JP2011/057021
- (22) 国際出願日: 2011 年 3 月 23 日(23.03.2011)
- (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 2011-056383 2011 年 3 月 15 日(15.03.2011) JP
- (71) 出願人 (米国を除く全ての指定国について): オムロン株式会社 (OMRON CORPORATION) [JP/JP]; 〒6008530 京都府京都市下京区塩小路通堀川東入南不動堂町 8 〇 1 番地 Kyoto (JP).
- (72) 発明者; および
- (75) 発明者/出願人 (米国についてのみ): 柴田 義也 (SHIBATA, Yoshiya). 小川 喜明 (OGAWA, Yoshiaki).
- (74) 代理人: 増井 義久, 外 (MASUI, Yoshihisa et al.); 〒5300041 大阪府大阪市北区天神橋 2 丁目北 2 番 6 号 大和南森町ビル 特許業務法人原謙三国際特許事務所 Osaka (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, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PE, PG, PH, PL, PT, RO, RS, RU, 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, SD, SL, SZ, TZ, UG, ZM, ZW), ユーラシア (AM, AZ, BY, KG, KZ, MD, 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

[続葉有]

(54) Title: DEBUGGING DEVICE

(54) 発明の名称: デバッグ装置



110 Simulator
113 Communication thread
115 Thread information output thread
120 Programming tool

121 Debugging function unit
122 Display control unit
501 Thread information

(57) Abstract: This debugging device is implemented by executing a simulation program and a debugging program on a computer OS. The simulation program is for emulating a controller function, and which operates on the OS as one process involving task threads and communication threads. The debugging program is for letting the computer function as an activation means for giving a control program to be debugged to the simulation program and for activating the simulation program, a halting means for halting the simulation program process at a breakpoint of the control program, and a resumption means for resuming the communication thread operation after the halting of the process by the halting means. Thus, a technique for easily implementing a debugger for a controller control program can be provided.

(57) 要約: 本発明のデバッグ装置は、シミュレーションプログラムとデバッグプログラムをコンピュータのOS上で実行させることで実現され、シミュレーションプログラムは、コントローラの機能をエミュレートするものであって、タスクスレッドと通信スレッドを含む1つのプロセスとして、上記

OS上で動作するものであり、デバッグプログラムは、コンピュータを、デバッグ対象とする制御プログラムをシミュレーションプログラムに与えて、シミュレーションプログラムを動作させる動作手段、制御プログラムのブレークポイントでシミュレーションプログラムのプロセスを一時停止する停止手段、および、停止手段によるプロセスの一時停止後に、通信スレッドの動作を再開する再開手段、として機能させるものである。これにより、コントローラの制御プログラムのデバッグを簡単に実現するための技術を提供することができる。

WO 2012/124162 A1

WO 2012/124162 A1



(BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG). 添付公開書類:

— 国際調査報告 (条約第 21 条(3))

明 細 書

発明の名称：デバッグ装置

技術分野

[0001] 本発明は、デバッグ装置に関する。

背景技術

[0002] FA (Factory Automation) においては、工場内に設置される生産設備のデータ収集及び制御を行う各種のスレーブ装置と、複数のスレーブ装置を集中管理するマスター装置（プログラマブルコントローラ（PLC））と、を通信バスを介して接続したフィールドネットワークにより生産設備の制御が行われる。マスター装置は、制御プログラムを実行することで、例えば、接続された複数のスレーブ装置を制御したり、各スレーブ装置から情報を取得したりする。また、マスター装置は、外部装置と通信を行う機能を有する。

[0003] 図7は、フィールドネットワークの構成例を示す図である。図7において、フィールドネットワーク100は、マスター装置200と複数のスレーブ装置300とがケーブル400や装置に備わるI/Oユニット500を介して直接的又は間接的に接続されることにより形成される。スレーブ装置300には、電源ユニット、モータユニット、カウンタユニット、画像ユニット、通信ユニット、I/Oユニット等がある。通信バスのトポロジーは、フィールドネットワークの規格により、ライン、デイジーチェーン、ツリー、スターなど種々のトポロジーをとり得る。

[0004] また、マスター装置200には、ユーザがマスター装置200の動作設定、フィールドネットワーク100の動作状態の表示、ネットワークシステムの設計（例えば、マスター装置とスレーブ装置の接続構成の決定や、マスター装置で実行する制御プログラムの作成）などを行うための管理装置600が接続されることもある。管理装置600は設定ツールがインストールされたパーソナルコンピュータなどにより構成される。

[0005] マスター装置で実行する制御プログラムの一般的な作成手順は以下のとおりである。

(1) まず、管理装置 600 等を実装されている開発環境（プログラミングツール）を用いて、マスター装置で実行させる制御プログラムのプログラムコードを作成する。プログラムは、例えば、ラダー言語、SFC 言語、ストラクチャードテキスト等で記述される。最近では、複数の言語が混在していることも多い。

(2) 作成した制御プログラムをコンパイル・ビルドすることにより、実行ファイルを作成する。

(3) 作成した実行ファイルを管理装置 600 等を実装されているシミュレータソフトを用いて実行（シミュレーション）し、制御プログラムに不具合が無いこと（即ち、マスター装置が意図した動作を行うこと）を確認する。

(4) 不具合が発見されたら（2）に戻りプログラムを修正し、（3）のシミュレーションを実行する。

(5) 完成した実行ファイルをマスター装置のメモリにインストールする。

[0006] 従来、（2）～（4）の作業は、デバッグ機能をもつ専用の PLC シミュレータソフトを用いて、制御プログラムをステップ実行したり、ブレークポイントで一時停止して変数の状態を確認したりするのが一般的であった。

[0007] しかしながら、このようなデバッグ機能をもつシミュレータソフトは、PLC の新製品が出たり、PLC の仕様（ファームウェア）が変更されるたびに、新たに作成しなおさなければならないため、シミュレータソフト自体の開発工数や開発コストが問題となっていた。尚、デバッグに関する先行技術としては、例えば、特許文献 1～5 に開示されたものが知られている。

先行技術文献

特許文献

[0008] 特許文献1：日本国公開特許公報「特開平 11-15697 号公報（1999

年 1 月 2 2 日公開)」

特許文献2：日本国公開特許公報「特開 2 0 0 1－1 7 5 4 9 9 号公報（2 0 0 1 年 6 月 2 9 日公開）」

特許文献3：日本国公開特許公報「特開 2 0 0 2－4 1 3 2 6 号公報（2 0 0 2 年 2 月 8 日公開）」

特許文献4：日本国公開特許公報「特開 2 0 0 7－7 9 9 6 9 号公報（2 0 0 7 年 3 月 2 9 日公開）」

特許文献5：日本国公開特許公報「特開 2 0 0 8－1 7 1 2 3 6 号公報（2 0 0 8 年 7 月 2 4 日公開）」

発明の概要

発明が解決しようとする課題

- [0009] 本発明は、コントローラの制御プログラムのデバッグを簡単に実現するための技術を提供することを目的とする。

課題を解決するための手段

- [0010] 上記目的を達成するために、本発明は以下の構成を採用する。

- [0011] 本発明のデバッグ装置は、

コントローラ用の制御プログラムをデバッグするためのデバッグ装置であって、

シミュレーションプログラムとデバッグプログラムをコンピュータのOS（Operating System）上で実行させることで実現され、

前記シミュレーションプログラムは、

前記コントローラの機能をエミュレートするものであって、

前記コントローラが制御プログラムを実行する機能をエミュレートするタスクスレッドと、前記コントローラが外部装置との通信を行う機能をエミュレートする通信スレッドとを含む 1 つのプロセスとして、前記 OS 上で動作するものであり、

前記デバッグプログラムは、前記コンピュータを、

デバッグ対象とする制御プログラムを前記シミュレーションプログラム

に与えて、前記シミュレーションプログラムを動作させる動作手段、

前記OSが提供するデバッグ用の命令を使って、前記制御プログラムのブレークポイントで前記シミュレーションプログラムのプロセスを一時停止する停止手段、および、

前記停止手段による前記プロセスの一時停止後に、前記プロセスにおける前記通信スレッドの動作を再開する再開手段、として機能させるものである、

ことを特徴とする。

[0012] 本発明のデバッグ装置では、シミュレーションプログラムはコントローラ（例えばPLC）の機能をエミュレートするだけの非常にシンプルなものであり、デバッグ機能は有していない。本発明のデバッグ装置では、デバッグ機能はデバッグプログラムが有しており、デバッグプログラムがOSのサービス（デバッグ用の命令）を使ってシミュレーションプログラムのプロセスを一時停止するという方法により、デバッグを実現する。このようなプログラム構成を採用することにより、PLCの新製品が出たり、PLCの仕様が変更されたりしても、シミュレーションプログラムの部分を入れ替えるだけで済み、デバッグプログラムの部分は流用可能であるため、デバッグ装置の開発工数や開発工数を従来に比べて低減することが期待できる。

[0013] ただし、上記のようなプログラム構成においては、デバッグプログラムがシミュレーションプログラムのプロセスを単純に一時停止すると、タスクスレッドにより実行されている制御プログラムの動作が停止するだけでなく、通信スレッドのように外部装置との通信を行う機能も停止してしまう。これでは、シミュレーションプログラムによりエミュレートされているPLC（以下、「仮想PLC」と称する。）の状態や内部変数などを外部から確認することができず、制御プログラムのデバッグを行うことができない。そこで、本発明では、シミュレーションプログラムのプロセスを一時停止した後に、通信スレッドの動作を強制的に再開させるという構成上の工夫を採用している。これにより、デバッグプログラムなどの外部のプロセスが、通信スレ

ッドを通じて仮想ＰＬＣの状態や内部変数などの情報を取得することが可能となるため、制御プログラムのデバッグ作業を遂行することができる。

[0014] ところで、ＰＬＣ自体の基本機能（制御プログラムを実行する機能、外部装置と通信する機能など）も、ＰＬＣのマイクロプロセッサ上で動作するプログラムで実現されているのが一般的である（以下、ＰＬＣ自体の基本機能を実現するプログラムを、スレーブ装置等の制御のためにＰＬＣに実行させる制御プログラムと区別するため、「ファームウェア」と称する。）。したがって、本発明のシミュレーションプログラムのようにＰＬＣの機能をエミュレートするだけのシンプルな機能のものは、ＰＬＣのファームウェアのソースコードをそのまま（若しくは若干の修正を加えて）、デバッグ装置のコンピュータのＯＳ用にリコンパイルするだけで、作成することも可能である。この場合は、シミュレーションプログラムの部分の開発は実質的に必要なくなるため、従来に比べて開発工数や開発コストを格段に低減することができる。また、ＰＬＣのファームウェアのソースコードを利用することにより、仮想ＰＬＣにおいてＰＬＣ（実機）の機能を忠実に再現することができるという利点もある。なお、本発明は、制御プログラムのデバッグを行うためのものであるため、少なくともタスクスレッドに相当する部分のファームウェアのソースコードを利用するとよい。もちろん、タスクスレッド以外の部分（通信スレッドなど）のソースコードを利用することも好ましい。

[0015] また、本発明は、上記手段の少なくとも一部を有するデバッグ装置として捉えてもよいし、上記処理の少なくとも一部を含むデバッグ方法、または、かかる方法を実現するためのプログラムやそのプログラムを記録した記録媒体として捉えることもできる。なお、上記手段及び処理の各々は可能な限り互いに組み合わせて本発明を構成することができる。

発明の効果

[0016] 本発明によれば、コントローラの制御プログラムのデバッグを簡単に実現することができる。

図面の簡単な説明

[0017] [図1] 図 1 は、本実施形態に係るデバッグ装置の機能構成の一例を示すブロック図である。

[図2] 図 2 は、表示制御部から通信スレッドに出力する通信コマンド、及び、通信スレッドからの応答の一例を示す図である。

[図3] 図 3 は、本実施形態に係るデバッグ装置のシミュレーション実行時の処理の流れの一例を示す図である。

[図4] 図 4 は、図 1 の構成における課題を示す図であり、プロセス一時停止時の処理の流れの一例を示す図である。

[図5] 図 5 は、本実施形態に係るデバッグ装置の処理の流れの一例を示す図である。

[図6] 図 6 は、スレッド情報の一例を示す図である。

[図7] 図 7 は、フィールドネットワークの構成例を示す図である。

発明を実施するための形態

[0018] (システム構成)

図 1 は、本実施形態に係るデバッグ装置の機能構成を示すブロック図である。このデバッグ装置 1000 は、コントローラの制御プログラムをデバッグするための装置である。

本実施形態では、コントローラがプログラマブルコントローラ（P L C）130 である場合を例に説明する。P L C 130 は、図 7 に例示したような産業用のネットワークシステムにおけるマスター装置（図 7 では符号 200）として利用されるものである。産業用ネットワークには様々な規格が存在し、また P L C 製品についても様々な仕様のものが存在するが、本実施形態のデバッグ装置 1000 はそのいずれにも適用可能である。

[0019] デバッグ装置 1000 は、C P U、主記憶装置（メモリなど）、補助記憶装置（ハードディスクなど）、各種入出力インターフェース、入力装置（キーボード、マウスなど）、表示装置等を備える汎用のコンピュータにより構成可能である。例えば、マイクロソフト社の Windows（登録商標）シリーズ、アップル社の MacOS シリーズなどの汎用の O S（オペレーションシステム）で

動作するコンピュータを好ましく利用できる。デバッグ装置１０００の機能を実現するシミュレーションプログラム及びデバッグプログラムは補助記憶装置に格納されており、ＣＰＵがこれらのプログラムを主記憶装置にロードし、ＯＳ上で実行することにより、後述するデバッグ装置１０００の各種の機能が実現される。

[0020] （プログラム構成）

図１において、シミュレータ１１０は、シミュレーションプログラムにより実現される機能を示し、プログラミングツール１２０は、デバッグプログラムにより実現される機能を示す。なお、図１には、比較のために実機のＰＬＣ１３０も示している。実機のＰＬＣ１３０においても、その基本機能は、ＰＬＣ用のリアルタイムＯＳ上で動作するプログラムで実現されている。図１の例では、制御プログラムを実行して各種スレーブ装置を制御する機能をタスクスレッド１３２が担い、制御プログラムの内部変数、エラーコード、ＰＬＣ１３０やスレーブ装置のステータスなどの各種情報を外部装置との間で送受する機能を通信スレッド１３３が担っている。

[0021] シミュレータ１１０は、ＰＬＣ１３０の機能をエミュレートする仮想ＰＬＣである。シミュレータ１１０のプログラムは、１つのプロセス１１１としてＯＳ上で実行され、タスクスレッド１１２、通信スレッド１１３など複数のスレッドを含むマルチスレッドプログラムである。タスクスレッド１１２は、ＰＬＣ１３０が制御プログラムを実行する機能をエミュレートするものである。即ち、シミュレータ１１０の制御プログラムの実行結果と、ＰＬＣ１３０の制御プログラムの実行結果は基本的に一致する。通信スレッド１１３は、ＰＬＣ１３０が外部装置との通信を行う機能をエミュレートするものである。

[0022] 本実施形態では、シミュレータ１１０のプログラム（タスクスレッド１１２の部分と通信スレッド１１３の部分）は、実機のＰＬＣ１３０用に開発されたファームウェアのソースコードをそのまま（若しくは若干の修正を加えたもの）を利用して作成されている。通常、デバッグ装置１０００のＯＳと

、PLC 130のOSとは異なるため、シミュレータ 110の実行ファイルは、ファームウェアのソースコードをデバッグ装置 1000のOS用にリコンパイル・ビルドして作成されるが、もし両者のOSが互換性を有しているような場合は、PLC 130のファームウェア（実行ファイル）をそのままシミュレータ 110として利用することも可能である。

[0023] プログラミングツール 120は、デバッグ機能部 121と表示制御部 122を有する。デバッグ機能部 121は、上記OSが提供するデバッグ用のAPI (Application Program Interface) を使って、デバッグ機能を果たす。例えば、上記OSが提供するデバッグ用の命令を使って、PLC 130の制御プログラムのブレークポイントでシミュレーションプログラムのプロセスを一時停止する（停止手段）。

[0024] 表示制御部 122は、通信スレッド 113（及び、PLC 130の通信スレッド 133）との通信結果を不図示の表示装置に表示する。表示制御部 122から通信スレッドに出力する通信コマンド、及び、通信スレッドからの応答の一例を図2に示す。例えば、実行中の制御プログラム内の変数のリストを取得したい場合には、表示制御部 122は、「List Var」というコマンドを出力する。それにより、通信スレッドから「Var 1～Var 3」といった変数のリストが返される。変数は、例えば、上記実行結果や該実行結果を算出する際に使用される変数などである。また、実行中の制御プログラム内の変数の現在値を取得したい場合には、表示制御部 122は「Get Var」というコマンドを出力する。それにより、通信スレッドから変数の現在値が返される（図2の例では、変数 Var 1を指定して「Get Var 1」が出力されたため、変数 Var 1の現在値「230」が返される）。

[0025] （シミュレーション実行時の処理の流れ）

図3は、本実施形態に係るデバッグ装置のシミュレーション実行時の処理の流れの一例を示す図である。

[0026] まず、ユーザは、プログラミングツール 120のエディタ 123を用いて、制御プログラムのソースファイル 301（ソースプログラム）を作成する

(ステップS 3 0 1)。

- [0027] そして、プログラミングツール1 2 0のコンパイラ1 2 4が、ユーザからの指示に応じて、ソースファイル3 0 1をコンパイルする(ステップS 3 0 2)。それにより、制御プログラムの実行ファイル3 0 2とデバッグ情報3 0 3とが作成される。デバッグ情報3 0 3は、ソースファイル3 0 1内の位置と、タスクスレッド1 1 2が制御プログラムを実行する際に実行ファイル3 0 2が展開されるメモリ上のアドレスとを対応付ける情報である。
- [0028] 次に、プログラミングツール1 2 0のタスク設定部1 2 5が、ユーザからの指示に応じて、タスク設定(ステップS 3 0 2で作成された複数の実行ファイル3 0 2の実行手順の設定)を行い、設定結果としてタスク設定情報3 0 4を生成する(ステップS 3 0 3)。
- [0029] そして、プログラミングツール1 2 0が、ユーザからの指示に応じてシミュレータ1 1 0を起動する。
- [0030] 次に、ステップS 3 0 5～S 3 1 1において、プログラミングツール1 2 0が、デバッグ対象とする制御プログラムをシミュレータ1 1 0に与え、シミュレータを動作させる(動作手段)。
- [0031] 具体的には、シミュレータ1 1 0のスケジューラ1 1 4が、ステップS 3 0 3で生成されたタスク設定情報3 0 4を読み込み、タスクスレッド1 1 2を生成する(ステップS 3 0 5)。また、スケジューラ1 1 4が、ステップS 3 0 2で作成された複数の実行ファイル3 0 2を読み込む(ステップS 3 0 6)。
- [0032] 次に、デバッグ機能部1 2 1が、プロセス1 1 1にアタッチする(ステップS 3 0 7)。これにより、デバッグ可能となる。
- [0033] そして、プログラミングツール1 2 0が、シミュレータ1 1 0に接続する(ステップS 3 0 8)。ここで、“接続する”とは、例えば、プログラミングツール1 2 0とシミュレータ1 1 0とを互いに通信可能な状態にすることを意味する。
- [0034] 次に、プログラミングツール1 2 0が、通信スレッド1 1 3にシミュレー

ションの実行コマンドを発行する（ステップS 3 0 9）。

[0035] 通信スレッド 1 1 3 は、実行コマンドが発行されると、スケジューラ 1 1 4 に対してシミュレーションの実行開始を指示する（ステップS 3 1 0）。

[0036] スケジューラ 1 1 4 は、シミュレーションの実行開始が指示されると、タスクスレッド 1 1 2 に実行を指示する（ステップS 3 1 1）。それにより、タスクスレッド 1 1 2 が動作し、シミュレーションが開始される。

[0037] そして、表示制御部 1 2 2 は、通信スレッド 1 1 3 に通信コマンドを発行し（ステップS 3 1 2）、通信スレッド 1 1 3 がそれに応答することにより（ステップS 3 1 3）、所望の情報が表示装置で表示される。

[0038] 以上が、本実施形態に係るデバッグ装置のシミュレーション実行時の処理の流れである。

[0039] （図 1 の構成における課題）

上述したように、図 1 に示すようなデバッグ装置を用いることにより、P L C の制御プログラムのデバッグを簡単に実現することが可能となる。しかしながら、本実施形態に係るデバッグ装置では、シミュレータ 1 1 0 は 1 つのプロセス 1 1 1 として動作しており、デバッグ機能部 1 2 1 は O S が提供するデバッグ用の A P I を用いてデバッグ機能を果たしている。そのため、ブレークポイントで、タスクスレッド 1 1 2 のみではなく、プロセス 1 1 1 全体が一時停止してしまい、通信スレッド 1 1 3 との通信が途絶してしまう。これでは、プログラミングツール 1 2 0 から、シミュレータ 1 1 0 による仮想 P L C の内部変数やステータスなどを確認することができず、制御プログラムが正常に動作しているか否かを確認することができない。即ち、デバッグ装置がデバッグとして機能しなくなってしまう。

[0040] 以下、上記課題について、図 4 を用いて説明する。図 4 は、図 1 の構成における課題を示す図であり、プロセス 1 1 1 の一時停止時の処理の流れの一例を示す図である。

[0041] まず、ユーザが、エディタ 1 2 3 を用いて、ブレークポイントを設定する（ステップS 4 0 1）。それにより、ソースファイル上のブレークポイント

の位置情報 401 が生成される。

[0042] 次に、デバッグ機能部 121 が、ステップ S401 で生成された位置情報 401 とデバッグ情報 303 を用いて、ブレークポイントに対応するメモリ上のアドレス 402 を求める（ステップ S402）。

[0043] そして、デバッグ機能部 121 は、ステップ S402 で求めたアドレスに、OS が提供するデバッグ用のブレーク命令（停止命令）を書き込む（ステップ S403）。それにより、ユーザが設定したブレークポイントでプロセス 111 全体が一時停止する（ステップ S404）。

[0044] その結果、通信スレッド 113 は一時停止するため、ステップ S405 で表示制御部 122 が通信スレッドに通信コマンドを発行したとしても、表示制御部 122 は、通信スレッド 113 から該通信コマンドに対する応答を得ることができなくなってしまう。

[0045] なお、このような課題は、シミュレータ 110 の機能をシンプルなものにし、かつ、OS のサービスであるデバッグ用 API を利用してプロセスの一時停止を行う構成を採用したために、生じるものである。言い換えると、ステップ実行などのデバッグ機能を備える高機能なシミュレータを用いれば、発生しない課題である。しかしながら、このような高機能なシミュレータを作成するには前述のように多大な開発工数と開発コストが必要となるため好ましくない。そこで本実施形態では、別の工夫を加えることにより、上記課題を解決することとした。

[0046] （上記課題を解決するための構成）

上記課題を解決するために、本実施形態では、デバッグ機能部 121 によるプロセス 111 の一時停止後に、通信スレッド 113 の動作を自動で再開する処理を行う（再開手段）。なお、本実施形態では、当該処理はデバッグ機能部 121 により行われるものとするが、デバッグ機能部 121 以外の機能により行われてもよい。なお、再開するスレッドは通信スレッド 113 のみに限らない。タスクスレッド 112 以外の複数のスレッドの動作が自動で再開されてもよい。また、デバッグ装置をデバッガとして機能させるために

必要なスレッド以外のスレッドが自動で再開されてもよい。

[0047] 以下、上記処理を含めた本実施形態に係るデバッグ装置 1000 の処理の流れについて図 5 を用いて説明する。図 5 は、本実施形態に係るデバッグ装置 1000 の処理の流れの一例を示す図である。

[0048] まず、シミュレータ 110 のスレッド情報出力スレッド 115 が、スレッド情報 501 を出力する（ステップ S501）。そして、デバッグ機能部 121 が、出力されたスレッド情報 501 を取得する（ステップ S502）。

[0049] スレッド情報 501 は、例えば、一時停止するスレッドを示す情報である。具体的には、各スレッドには識別子が設定されており、スレッド情報 501 として、図 6 の（A）に示すような、一時停止するスレッドのリストが用いられる。図 6 の（A）のスレッド情報は、「CycleTask1__RUN」、「CycleTask1__IO__IN」、「CycleTask1__IO__OUT」、「CycleTask2__RUN」、「CycleTask2__IO__IN」、「CycleTask2__IO__OUT」、「TimeGenerator」の 6 つのタスクスレッドが一時停止するスレッドであることを示す。図中、「＝」の後に付加されている数値は、各スレッドの識別子である。スレッド情報 501 は、ブレークポイントがヒットする前（例えば、シミュレーションの実行開始直後や、プログラミングツール 120 とシミュレータ 110 の接続（ステップ S308）直後）に出力される。

[0050] 次に、デバッグ機能部 121 が、OS が提供するデバッグ用の API を用いて、タスクスレッド 112 にブレークポイントを設定する（ステップ S503）。即ち、図 4 のステップ S403 と同様の処理が行われる。それにより、ユーザが設定したブレークポイントでプロセス 111 全体が一時停止する（ステップ S504）。

[0051] プロセス 111 全体が一時停止すると、デバッグ機能部 121 は、ステップ S502 で取得したスレッド情報に基づいて、動作の再開が必要なスレッドを解釈し、シミュレータ 110 に対して当該スレッドの動作の再開を指示する（ステップ S505）。本実施形態では、図 6 の（A）のリストに含ま

れないスレッドの動作の再開が指示される。それにより、通信スレッド 1 1 3 の動作が再開される。なお、デバッグ機能部 1 2 1 は、この時点でスレッド情報 5 0 1 を取得してもよい。

[0052] その結果、表示制御部 1 2 2 は、ステップ S 5 0 6 で通信スレッド 1 1 3 に通信コマンドを発行した場合に、ステップ S 5 0 7 で通信スレッド 1 1 3 から該通信コマンドに対するして応答を得ることができ、所望の情報を表示装置に表示することができる。即ち、デバッグ装置をデバッガとして機能させることができる。

[0053] 以上述べたように本実施形態のデバッグ装置 1 0 0 0 では、シミュレータ 1 1 0 は P L C 1 3 0 の機能をエミュレートするだけの非常にシンプルなものであり、デバッグ機能は有していない。そして、プログラミングツール 1 2 0 のデバッグ機能部 1 2 1 が O S のサービス（デバッグ用の命令）を使ってシミュレータ 1 1 0 のプロセス 1 1 1 を一時停止するという方法により、デバッグを実現する。このようなプログラム構成を採用することにより、P L C の新製品が出たり、P L C の仕様が変更されたりしても、シミュレータ 1 1 0 のプログラムの部分を入れ替えるだけで済み、プログラミングツール 1 2 0 の部分は流用可能であるため、デバッグ装置全体の開発工数や開発工数を従来に比べて低減することが期待できる。

[0054] また本実施形態では、P L C 1 3 0 のファームウェアのソースコードを利用して、デバッグ装置 1 0 0 0 のシミュレータ 1 1 0 を作成している。これにより、シミュレータ 1 1 0 のプログラムの部分の開発は実質的に必要なくなるため、従来に比べて開発工数や開発コストを格段に低減することができる。また、P L C 1 3 0 のファームウェアのソースコードを利用することにより、シミュレータ 1 1 0 の仮想 P L C において実機の P L C 1 3 0 の機能を忠実に再現することができるという利点もある。

[0055] なお、本実施形態では、スレッド情報に基づいて動作を再開するスレッドが決定される構成としたが、スレッド情報は用いなくてもよい。スレッドの動作の再開を指示するデバッグ機能部 1 2 1 等に動作を再開するスレッドを

予め把握させておいてもよい。

- [0056] なお、本実施形態では、スレッド情報が一時停止するスレッドを示す情報である場合について説明したが、スレッド情報はこれに限らない。例えば、スレッド情報は動作を再開するスレッドを示す情報であってもよい。具体的には、スレッド情報501として、図6の（B）に示すような、動作を再開するスレッドのリストを用いてもよい。図6の（B）のスレッド情報は、通信スレッドである「Connection」と「Communication」が再開するスレッドであることを示す。スレッド情報が動作を再開するスレッドのリストである場合、ステップS505では、リストに含まれるスレッドの動作の再開を指示すればよい。

符号の説明

- [0057] 1 設計支援システム
- 2 ディスプレイ
- 3 CPU
- 4 入力装置
- 31 記憶手段
- 32 取得手段
- 33 入力手段
- 34 生成手段
- 35 画像出力手段
- 100 フィールドネットワーク
- 110 シミュレータ
- 111 プロセス
- 112 タスクスレッド
- 113 通信スレッド
- 114 スケジューラ
- 115 スレッド情報出力スレッド
- 120 プログラミングツール

- 1 2 1 デバッグ機能部
- 1 2 2 表示制御部
- 1 2 3 エディタ
- 1 2 4 コンパイラ
- 1 2 5 タスク設定部
- 1 3 0 マスター装置
- 1 3 3 通信スレッド
- 2 0 0 マスター装置
- 3 0 0 スレーブ装置
- 3 0 1 ソースファイル
- 3 0 2 実行ファイル
- 3 0 3 デバッグ情報
- 3 0 4 タスク設定情報
- 4 0 0 ケーブル
- 4 0 1 位置情報
- 4 0 2 アドレス
- 5 0 0 ユニット
- 5 0 1 スレッド情報
- 6 0 0 管理装置
- 1 0 0 0 デバッグ装置

請求の範囲

- [請求項1] コントローラ用の制御プログラムをデバッグするためのデバッグ装置であって、
- シミュレーションプログラムとデバッグプログラムをコンピュータのOS上で実行させることで実現され、
- 前記シミュレーションプログラムは、
- 前記コントローラの機能をエミュレートするものであって、
- 前記コントローラが制御プログラムを実行する機能をエミュレートするタスクスレッドと、前記コントローラが外部装置との通信を行う機能をエミュレートする通信スレッドとを含む1つのプロセスとして、前記OS上で動作するものであり、
- 前記デバッグプログラムは、前記コンピュータを、
- デバッグ対象とする制御プログラムを前記シミュレーションプログラムに与えて、前記シミュレーションプログラムを動作させる動作手段、
- 前記OSが提供するデバッグ用の命令を使って、前記制御プログラムのブレークポイントで前記シミュレーションプログラムのプロセスを一時停止する停止手段、および、
- 前記停止手段による前記プロセスの一時停止後に、前記プロセスにおける前記通信スレッドの動作を再開する再開手段、として機能させるものである、
- ことを特徴とするデバッグ装置。
- [請求項2] 前記シミュレーションプログラムの少なくともタスクスレッドの部分は、前記コントローラのファームウェアのソースコードを利用して作成されたものであることを特徴とする請求項1に記載のデバッグ装置。
- [請求項3] コントローラ用の制御プログラムをデバッグするためのプログラムであって、

コンピュータのOS上で実行されるシミュレーションプログラムとデバッグプログラムを含み、

前記シミュレーションプログラムは、

前記コントローラの機能をエミュレートするものであって、

前記コントローラが制御プログラムを実行する機能をエミュレートするタスクスレッドと、前記コントローラが外部装置との通信を行う機能をエミュレートする通信スレッドとを含む1つのプロセスとして、前記OS上で動作するものであり、

前記デバッグプログラムは、前記コンピュータに、

デバッグ対象とする制御プログラムを前記シミュレーションプログラムに与えて、前記シミュレーションプログラムを動作させる動作ステップと、

前記OSが提供するデバッグ用の命令を使って、前記制御プログラムのブレークポイントで前記シミュレーションプログラムのプロセスを一時停止する停止ステップと、

前記停止ステップによる前記プロセスの一時停止後に、前記プロセスにおける前記通信スレッドの動作を再開する再開ステップと、を実行させる

ことを特徴とするプログラム。

[請求項4] 請求項3に記載のプログラムを記録していることを特徴とするコンピュータ読取可能な記録媒体。

[請求項5] コントローラ用の制御プログラムをデバッグするためのデバッグ方法であって、

コンピュータが、シミュレーションプログラムとデバッグプログラムを前記コンピュータのOS上で実行することで実現され、

前記シミュレーションプログラムは、

前記コントローラの機能をエミュレートするものであって、

前記コントローラが制御プログラムを実行する機能をエミュレー

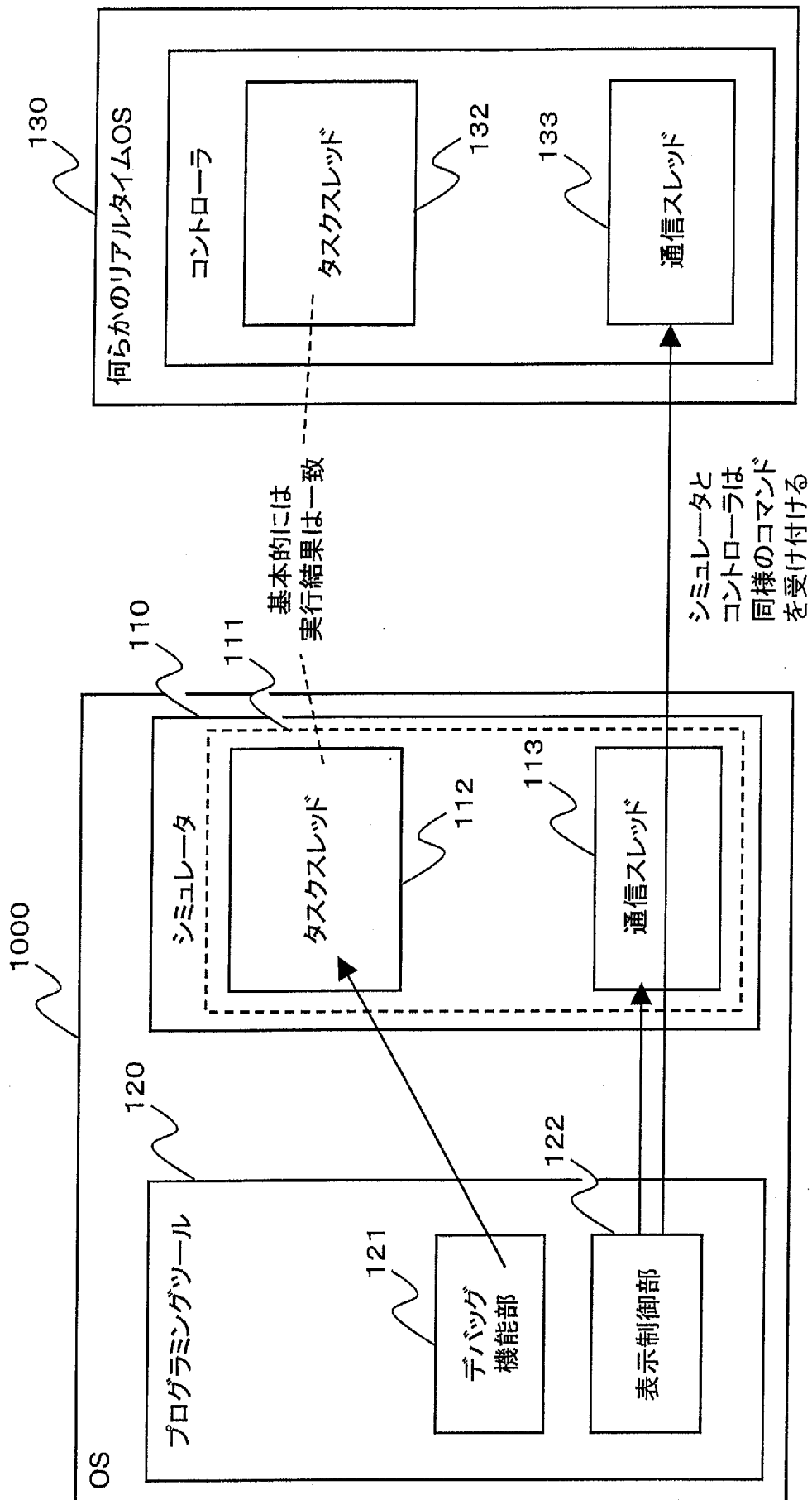
トするタスクスレッドと、前記コントローラが外部装置との通信を行う機能をエミュレートする通信スレッドとを含む1つのプロセスとして、前記OS上で動作するものであり、

前記コンピュータが、前記デバッグプログラムを実行することにより、デバッグ対象とする制御プログラムを前記シミュレーションプログラムに与えて、前記シミュレーションプログラムを動作させる動作ステップと、

前記コンピュータが、前記デバッグプログラムを実行することにより、前記OSが提供するデバッグ用の命令を使って、前記制御プログラムのブレークポイントで前記シミュレーションプログラムのプロセスを一時停止する停止ステップと、

前記コンピュータが、前記デバッグプログラムを実行することにより、前記停止ステップによる前記プロセスの一時停止後に、前記プロセスにおける前記通信スレッドの動作を再開する再開ステップと、を有することを特徴とするデバッグ方法。

[図1]



[図2]

Command : ListVar

Response : VAR://PROGRAM1/Var1

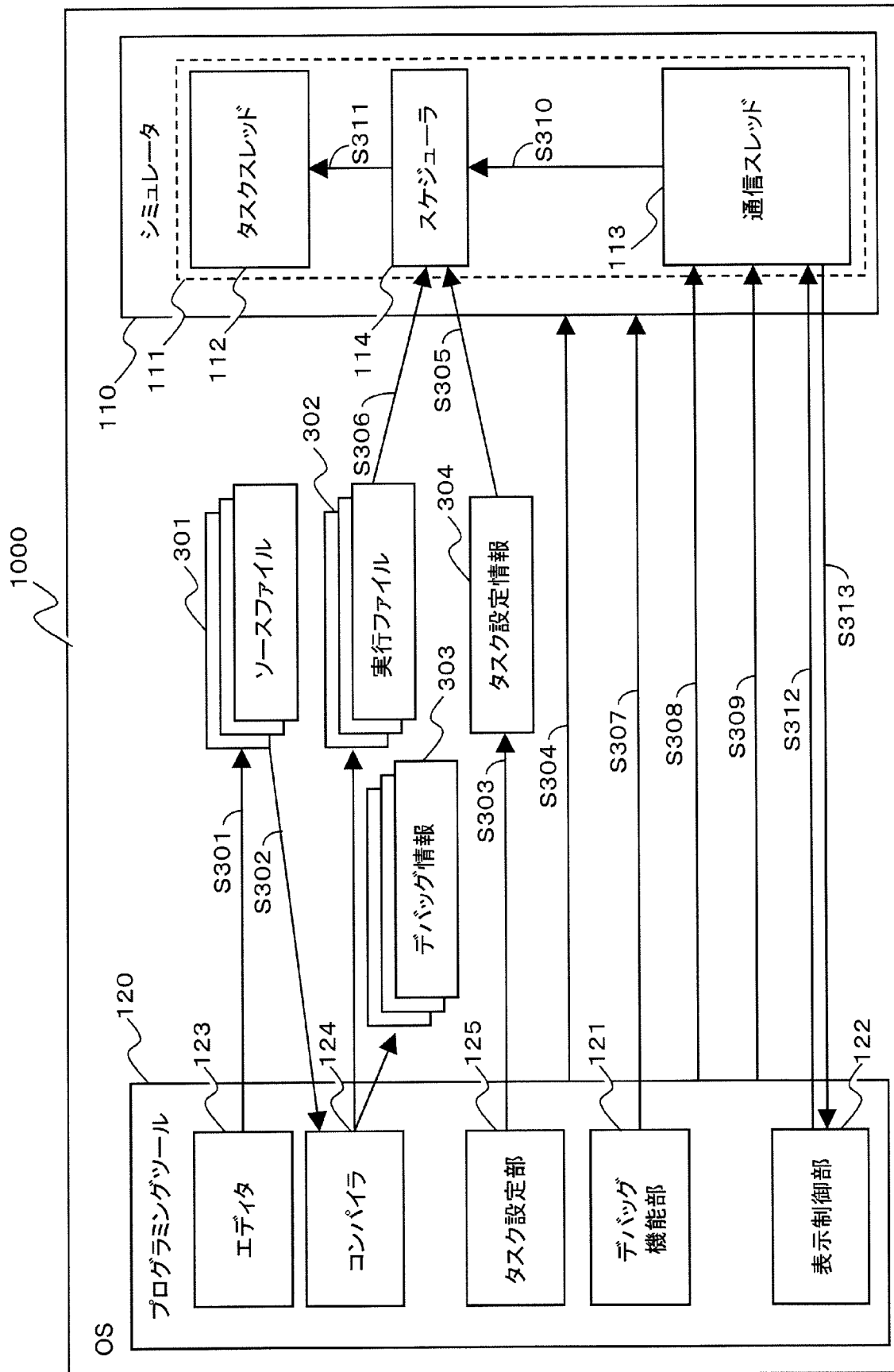
VAR://PROGRAM1/Var2

VAR://PROGRAM1/Var3

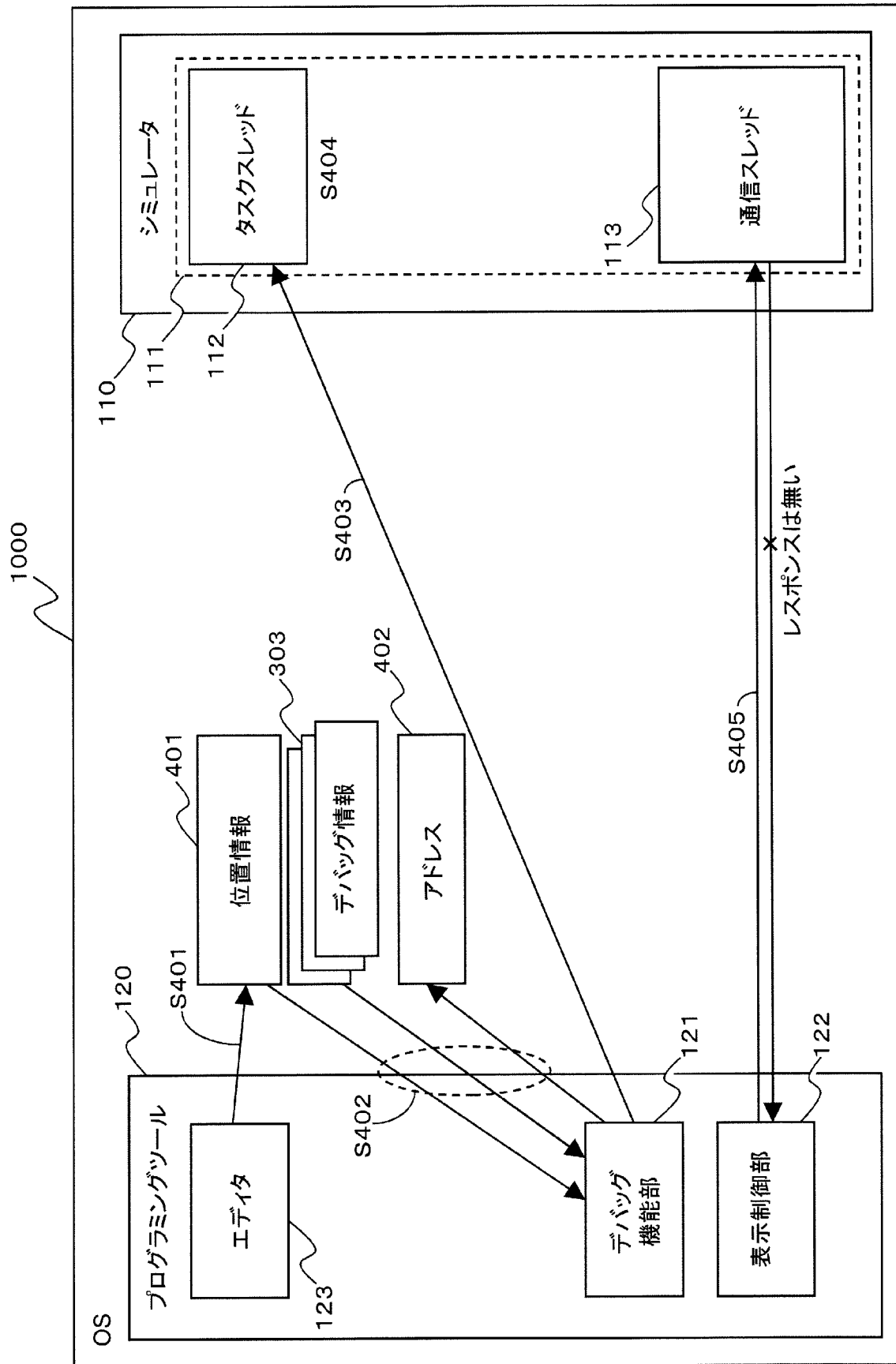
Command : GetVar(VAR://PROGRAM1/Var1)

Response : 230

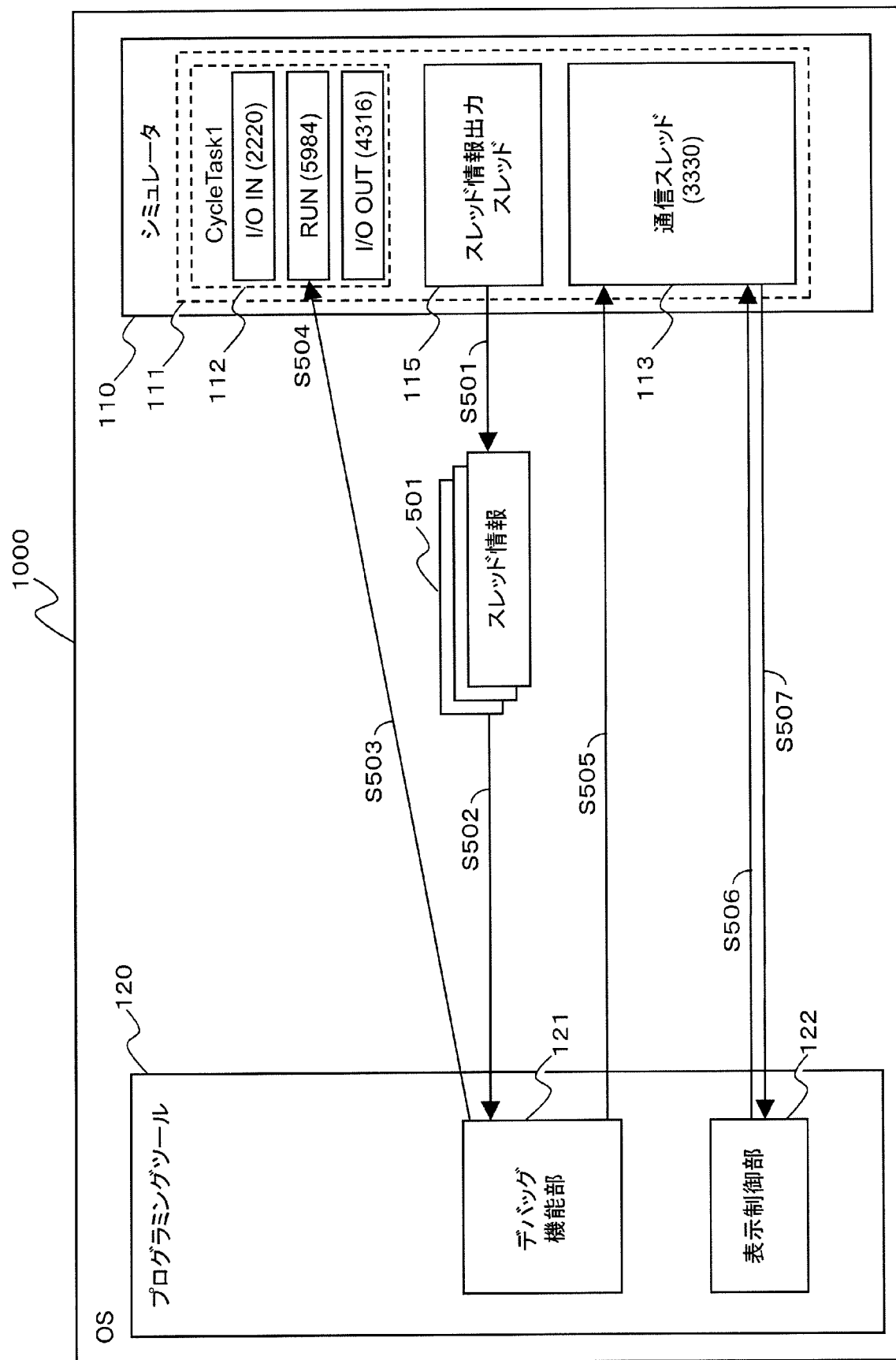
[図3]



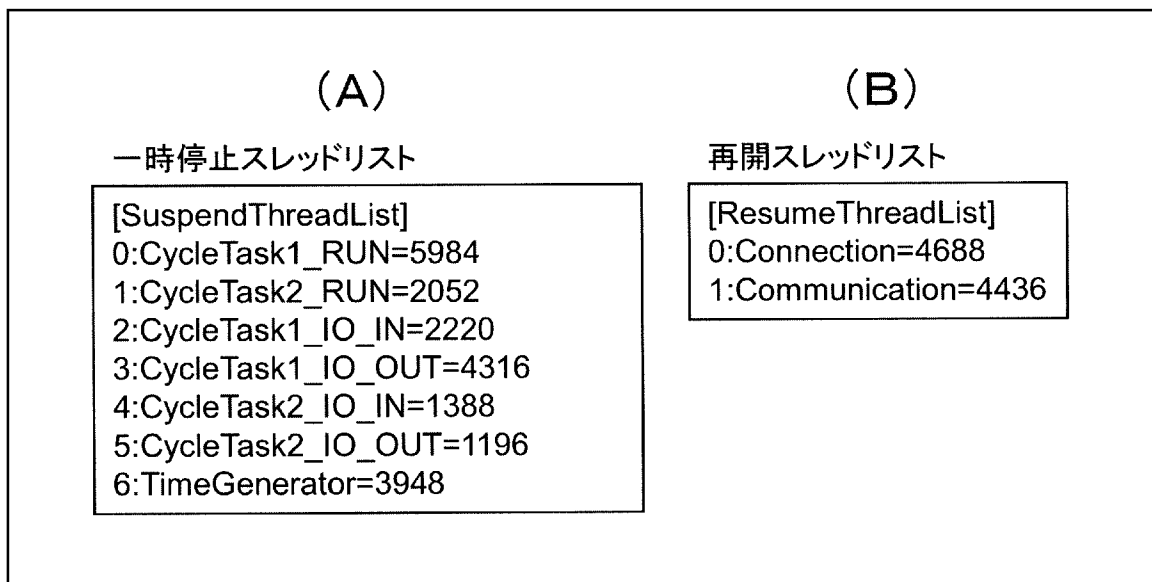
[図4]



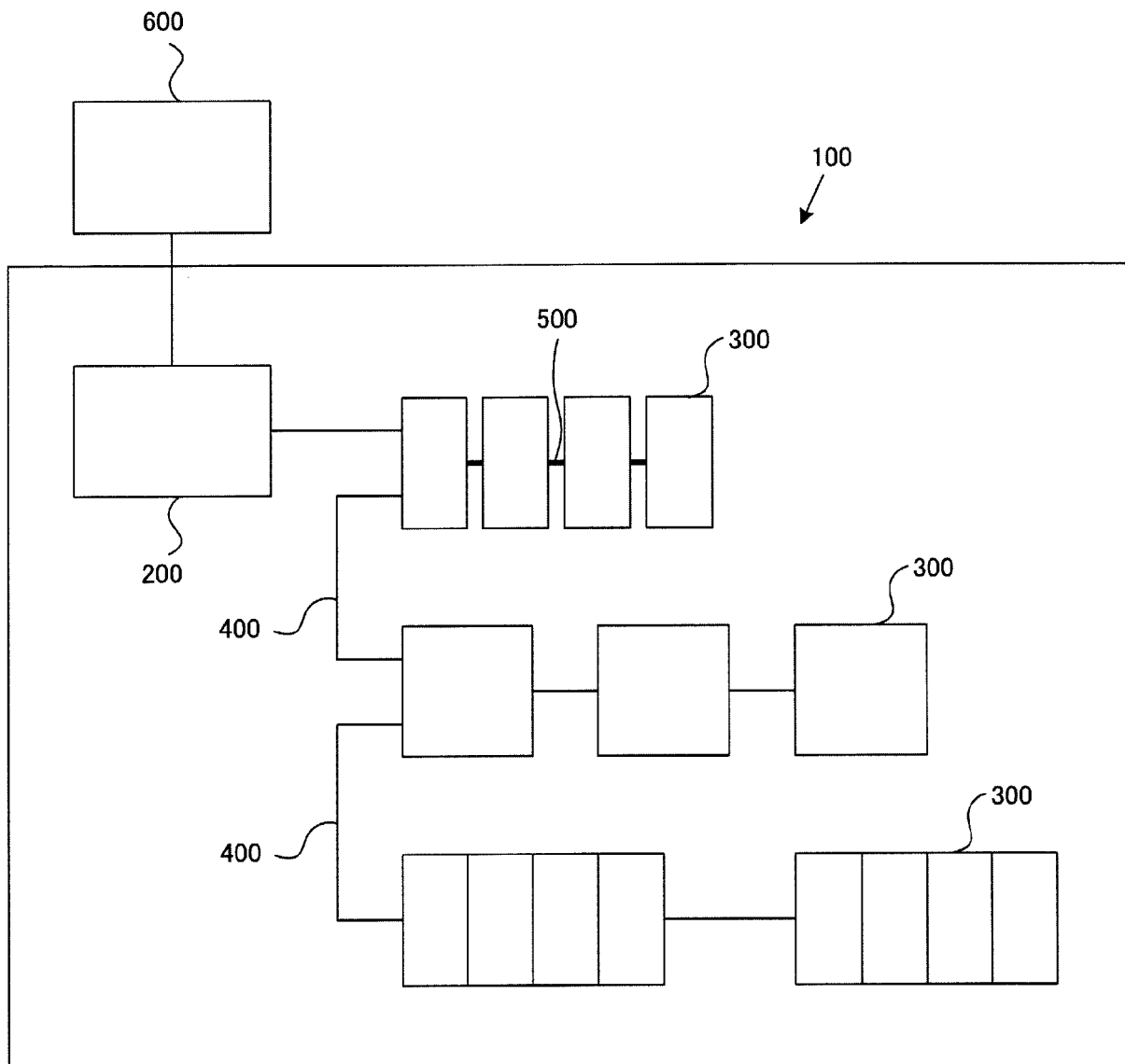
[図5]



[図6]



[図7]



INTERNATIONAL SEARCH REPORT

International application No.

PCT/JP2011/057021

A. CLASSIFICATION OF SUBJECT MATTER

G06F11/28 (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)

G06F11/28, 11/36

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 1996-2011

Kokai Jitsuyo Shinan Koho 1971-2011 Toroku Jitsuyo Shinan Koho 1994-2011

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.
Y	JP 5-274177 A (Yokogawa Electric Corp.), 22 October 1993 (22.10.1993), all pages; all drawings (Family: none)	1-5
Y	JP 2009-193109 A (Advantest Corp.), 27 August 2009 (27.08.2009), claim 2; paragraphs [0010], [0075] to [0081] & US 2009/0204849 A1	1-5
Y	JP 6-161817 A (Yokogawa Electric Corp.), 10 June 1994 (10.06.1994), paragraph [0004] (Family: none)	1-5



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

25 May, 2011 (25.05.11)

Date of mailing of the international search report

07 June, 2011 (07.06.11)

Name and mailing address of the ISA/

Japanese Patent Office

Authorized officer

Facsimile No.

Telephone No.

INTERNATIONAL SEARCH REPORT

International application No.

PCT/JP2011/057021

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	JP 2000-250777 A (Canon Inc.), 14 September 2000 (14.09.2000), paragraph [0019] (Family: none)	2
A	Kazuya KURIYAMA et al., "Simulation Development Environment for Real-Time System", Journal of the Society of Instrument and Control Engineers, 1992.07, vol.31, no.7, pages 811 to 815	1-5

A. 発明の属する分野の分類（国際特許分類（I P C））

Int.Cl. G06F11/28(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. G06F11/28, 11/36

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	1 9 2 2 - 1 9 9 6 年
日本国公開実用新案公報	1 9 7 1 - 2 0 1 1 年
日本国実用新案登録公報	1 9 9 6 - 2 0 1 1 年
日本国登録実用新案公報	1 9 9 4 - 2 0 1 1 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
Y	JP 5-274177 A（横河電機株式会社）1993.10.22, 全頁, 全図 （ファミリーなし）	1 - 5
Y	JP 2009-193109 A（株式会社アドバンテスト）2009.08.27, 【請求項2】, 第【0010】, 【0075】 - 【0081】段落 & US 2009/0204849 A1	1 - 5
Y	JP 6-161817 A（横河電機株式会社）1994.06.10, 第【0004】段落 （ファミリーなし）	1 - 5

☒ 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

2 5 . 0 5 . 2 0 1 1

国際調査報告の発送日

0 7 . 0 6 . 2 0 1 1

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 1 0 0 - 8 9 1 5

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）

5 B

9 3 6 7

多賀 実

電話番号 0 3 - 3 5 8 1 - 1 1 0 1 内線 3 5 4 5

C (続き) . 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
Y	JP 2000-250777 A (キヤノン株式会社) 2000. 09. 14, 第【0019】段落 (ファミリーなし)	2
A	栗山和也 外 1 名, シミュレーションによるリアルタイムシステム 開発環境, 計測と制御, 1992. 07, 第 31 巻, 第 7 号, p. 811-815	1 - 5