(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号

特許第3832517号

(P3832517)

(45)発行日　平成18年10月11日(2006.10.11)　(24)登録日　平成18年7月28日(2006.7.28)

(51)Int.Cl.		F I		
B25B 13/00	(2006.01)	B25B	13/00	Z
G06F 9/46	(2006.01)	G06F	9/46	340B
B25J 9/16	(2006.01)	B25J	9/16	
G05B 19/4155	(2006.01)	G05B	19/4155	N

請求項の数 11　(全 20 頁)

(21)出願番号　特願平8-195352
(22)出願日　平成8年7月5日(1996.7.5)
(65)公開番号　特開平10-15836
(43)公開日　平成10年1月20日(1998.1.20)
審査請求日　平成14年11月7日(2002.11.7)

(73)特許権者　000002369
セイコーエプソン株式会社
東京都新宿区西新宿２丁目４番１号
(74)代理人　100090387
弁理士　布施　行夫
(74)代理人　100090398
弁理士　大渕　美千栄
(72)発明者　横島　典夫
千葉県習志野市屋敷４丁目３番１号　セイコー精機株式会社内
(72)発明者　塩田　寿美
千葉県習志野市屋敷４丁目３番１号　セイコー精機株式会社内

最終頁に続く

(54)【発明の名称】ロボット用コントローラ及びその制御方法

(57)【特許請求の範囲】

【請求項１】

プリエンプティブなマルチタスク機能を有する汎用オペレーティングシステムを用いたロボット用コントローラであって、

前記マルチタスク機能を構成する各々のタスクを切替るタスク切替手段に対し、前記各々のタスクの切替を指示することにより、前記各々のタスク処理がリアルタイムに実行されるよう制御するリアルタイム制御手段を有し、

前記リアルタイム制御手段は、前記汎用オペレーティングシステムが提供可能な間隔より短い間隔で定期的にイベントを検出し、該イベントに対応する処理を実行する前記各々のタスクへの切替を、前記タスク切替手段に指示するイベント駆動処理を行うイベント駆動手段とを有することを特徴とするロボット用コントローラ。

【請求項２】

請求項１において、

前記リアルタイム制御手段は、外部タイマを用いて前記汎用オペレーティングシステムが提供可能な間隔より短い間隔毎に割り込み信号を発生させる外部割り込み発生手段をさらに有し、

前記イベント駆動手段は、前記外部割り込み発生手段が発生した割り込み信号に同期してイベント駆動処理を行うことを特徴とするロボット用コントローラ。

【請求項３】

請求項１または２のいずれかにおいて、

前記リアルタイム制御手段は、前記イベントの発生に対応する処理を行うプログラムからの要求によって、前記イベントの発生を待つ旨の登録を行うイベント登録手段をさらに有し、

前記イベント駆動手段は、前記イベント登録手段によって発生を待つ旨登録されたイベントの発生を検出したときは、前記イベント登録手段に対して該イベントの発生を待つ旨の登録を要求したプログラムが実行されるタスクへの切替を前記タスク切替手段に指示することを特徴とするロボット用コントローラ。

【請求項４】

請求項１から３のいずれかにおいて、

前記リアルタイム制御手段は、ロボットの有するハードウエアリソースの変化と、マニピュレータ動作又は周辺装置を制御するプログラムで同期、通信を行うための出力データ群との少なくとも一方をイベントとして検出することを特徴とするロボット用コントローラ。

【請求項５】

請求項４において、

前記リアルタイム制御手段は、複数のタスクから参照更新可能な共有メモリ領域に、前記ハードウエアリソースの変化と、マニピュレータ動作又は周辺装置を制御するプログラムで同期、通信を行うための出力データ群との少なくとも一方を管理するためのイベントリソース状態を記憶するイベントリソース状態記憶手段と、

前記マニピュレータ動作と前記周辺装置を制御するプログラムの同期、通信を行うための出力データ群との少なくとも一方に基づき、前記イベントリソース状態記憶手段に記憶されたイベントリソース状態を更新するイベントリソース状態更新手段とを有し、

前記イベント駆動手段は、前記イベントリソース状態更新手段が更新したイベントリソース状態に基づき、ハードウエアリソースであるＩ／Ｏポート等の値やプログラムで同期、通信を行うための入出力データ等を更新するハードウエアリソース更新手段とを含むことを特徴とするロボット用コントローラ。

【請求項６】

請求項３又は請求項３に従属する請求項４、５のいずれかにおいて、

イベントの発生を待つ旨の登録の要求を行ったプログラムを、スワップアウトされないように定期的にダミイ起動することをタスク切替手段に指示する手段を含むことを特徴とするロボット用コントローラ。

【請求項７】

プリエンプティブなマルチタスク機能を有する汎用オペレーティングシステムを用いたロボット用コントローラの制御方法であって、

外部タイマを用いて、前記汎用オペレーティングシステムが提供可能な間隔より短い間隔で定期的に割り込み信号を発生させる外部割り込み発生ステップと、

前記外部割り込み発生ステップにおいて発生された割り込み信号に同期して、イベントを検出し、該イベントに対応する処理が実行されるタスクへ、タスクを切替えることを前記汎用オペレーティングシステムに指示するイベント駆動処理を行う、イベント駆動ステップを有することを特徴とする制御方法。

【請求項８】

請求項７において、

前記イベントの発生に対応する処理を行うプログラムからの要求によって、前記イベントの発生を待つ旨の登録を行うイベント登録ステップをさらに有し、

前記イベント駆動ステップは、前記イベント登録ステップにおいて発生を待つ旨登録されたイベントの発生を検出したときは、前記イベント登録ステップに対して該イベントの発生を待つ旨の登録を要求したプログラムが実行されるタスクへの切替を前記汎用オペレーティングシステムに指示することを特徴とする制御方法。

【請求項９】

請求項７または８のいずれかにおいて、

ロボットの有するハードウエアリソースの変化と、マニピュレータ動作又は周辺装置を制御するプログラムで同期、通信を行うための出力データ群との少なくとも一方をイベントとして検出することを特徴とする制御方法。

【請求項１０】

請求項９において、

複数のタスクから参照更新可能な共有メモリ領域に設定されたイベントリソーステーブルに記憶された、前記ハードウエアリソースの変化と、マニピュレータ動作又は周辺装置を制御するプログラムで同期、通信を行うための出力データ群との少なくとも一方を管理するためのイベントリソース状態を、

前記マニピュレータ動作と前記周辺装置を制御するプログラムの同期、通信を行うための出力データ群との少なくとも一方に基づき更新するイベントリソース状態更新ステップをさらに有し、

前記イベント駆動ステップは、

前記イベントリソース状態更新ステップにおいて更新されたイベントリソース情報に基づき、ハードウエアリソースであるＩ／Ｏポート等の値やプログラムで同期、通信を行うための入出力データ等を更新するハードウエアリソース更新ステップとを含むことを特徴とする制御方法。

【請求項１１】

請求項８又は請求項８に従属する請求項９、１０のいずれかにおいて、

イベントの発生を待つ旨の登録の要求を行ったプログラムを、スワップアウトされないように定期的にダミイ起動することを前記汎用オペレーティングシステムに指示するステップを含むことを特徴とする制御方法。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】

本発明はロボット用コントローラ及びその制御方法に関し、特にイベントの発生に対応する処理をリアルタイムで行うロボット用コントローラ及びパソコン用汎用オペレーティングシステムを用いた制御方法に関する。

【０００２】

【背景技術】

従来のロボット用コントローラは、例えばスカラー型マニピュレータや直角座標用マニピュレータのように特定のマニピュレータの動作制御を行う場合が多く、そのような場合、通常独自開発のロボット制御専用オペレーティングシステム（以下ＯＳという）を使用していた。

【０００３】

しかし、最近では、マニピュレータの動作制御を行うだけでなく、周辺装置の制御或いは外部装置との通信機能等のシステム全体の制御を行うロボット用コントローラの需要が高くなっている。この様な場合、ロボット用コントローラのＯＳとして、マルチタスク機能を有するＯＳを使用する必要があり、ほとんどの場合は独自開発の専用ＯＳ又は汎用的なリアルタイムＯＳ（ｐＳＯＳ、ＶｘＷｏｒｋｓ、ＶＲＴＸ、ＯＳ－９等）が使用されている。

【０００４】

つまり、マニピュレータや周辺装置等の制御にはリアルタイム性が要求されることから、ロボット用コントローラのＯＳは処理のリアルタイム性を確保できることが重要な要件となってくる。

【０００５】

しかし、このようなリアルタイム性を確保できる専用ＯＳの開発には、多大な工数を要する。また、汎用的なリアルタイムＯＳを使用すると、リアルタイム性の確保は容易であるが、高価であるため、コスト面で問題が生じる。

【０００６】

一方、最近では、安価で高機能なパソコン用汎用ＯＳが流通し、一般に使用されている。従って、このようなパソコン用汎用ＯＳをロボット用コントローラに使用することが出来れば、開発工数の削減及びコストダウンが可能となる。さらに、利用者の修得が容易であること、開発環境が充実しているため開発期間をさらに短縮出来ること、市販のハードウエア及びソフトウエアを活用した拡張が可能なこと等のメリットがある。

【０００７】

しかし、安価なパソコン及びパソコン用の汎用ＯＳでは、マルチタスク機能を有していても、タスク切替が低速でかつ、切替間隔の短縮が不可能であり、リアルタイム性の確保が困難であると言う問題がある。

【０００８】

前述したように、マニピュレータや周辺装置等の制御にはリアルタイム性が要求されるため、事象が起こってからその事象の処理をするタスクが起動されるまでの時間が長時間であれば、実用にならないのである。

【０００９】

さらに、前記汎用的なパソコン及びパソコン用の汎用ＯＳを用いた場合、事象が起こってから、システムのタイムスライスによってその事象の処理をするタスクが起動されるまでの時間は、事象が起こるタイミングによりばらつき、これが処理の再現性の品質を低下させるという問題もある。

【００１０】

【発明が解決しようとする課題】

そこで、前記汎用的なパソコン及びパソコン用の汎用ＯＳのように、いわゆるリアルタイムＯＳではないＯＳ上で、高速なタスク切替、起動時間のばらつき等なくすことを実現すれば、前記パソコン用の汎用ＯＳの数々のメリットを生かして、少ない開発工数で高機能かつ利用者の修得が容易で、市販のハードウエア及びソフトウエアを活用した拡張が可能なロボット用コントローラを提供することが出来る。

【００１１】

本発明は、この様な課題に鑑みてなされたものであり、その目的は、汎用的なパソコン及びパソコン用の汎用ＯＳを用いたロボット用コントローラ及びその制御方法を提供することである。

【００１２】

【課題を解決するための手段】

前記目的を達成するため、請求項１記載の発明は、

プリエンプティブなマルチタスク機能を有するタスク切替手段と、

前記タスク切替手段にタスクの切替を指示することにより、イベントの発生に対応する処理がリアルタイムに実行されるよう制御するリアルタイム制御手段を有するロボット駆動用コントローラであって、

前記リアルタイム制御手段は、

リアルタイム処理を実行するのに必要な繰り返し間隔で定期的にイベントを検出し、該イベントに対応する処理が実行されるタスクへの切替を前記タスク切替手段に指示するイベント駆動処理を行うイベント駆動手段と、

を含むことを特徴とする。

【００１３】

ここにおいて、プリエンプティブ方式とは、一般的にＣＰＵの処理を一定時間毎に区切って、ＯＳが優先度に応じてアプリケーション毎に配分することをいう。すなわち、アプリケーションがＣＰＵを明け渡す前にＯＳが時間毎に処理を先取りして別のアプリケーションに切り替えてしまう仕組みをいう。従って、プリエンプティブなマルチタスク機能をそなえていると、複数のアプリケーションを実行している場合、片方のアプリケーションが処理の終了の宣言をするまで、別のアプリケーションが待たなければならないという事態を招かず、ユーザーが実感する待ち時間は大幅に軽減され、見かけ上の効率があがる。

【００１４】

請求項１の様にすると、前記繰り返し間隔毎にイベントを検出し、該イベントに対応した処理が実行されるタスクを駆動することが出来る。従ってイベントの発生にリアルタイムに対応するのに必要な繰り返し間隔毎にイベント駆動処理を行うことにより、イベントに対応する処理がリアルタイムに実行されるよう制御することができる。

【００１５】

また、処理すべきイベントが一元的に管理出来るため、イベントの管理が容易になる。

【００１６】

請求項２記載の発明は、

請求項１において、

前記リアルタイム制御手段は、

外部タイマを用いて前記繰り返し間隔毎に割り込み信号を発生させる外部割り込み発生手段をさらに有し、

前記イベント駆動手段は、

前記外部割り込み発生手段が発生した割り込み信号に同期してイベント駆動処理を行うことを特徴とする。

【００１７】

請求項７記載の発明は、

プリエンプティブなマルチタスク機能を有する汎用オペレーティングシステムを用いたロボット用コントローラにおいて、イベント発生に対応した処理のリアルタイム制御を行う方法であって、

外部タイマを用いて、リアルタイム処理を実行するのに必要な繰り返し間隔で、定期的に割り込み信号を発生させる外部割り込み発生ステップと、

前記外部割り込み発生ステップにおいて発生された割り込み信号に同期して、イベントを検出し、該イベントに対応する処理が実行されるタスクへ切替えることを前記汎用オペレーティングシステムに指示するイベント駆動処理を行うイベント駆動ステップを有することを特徴とする。

【００１８】

この様にすると、外部タイマによって、イベントの発生にリアルタイムに対応するのに十分な繰り返し間隔毎に外部割り込み信号を発生させることが出来る。従って、該割り込み信号に同期してイベント駆動処理を行うことにより、前記繰り返し間隔をタイムスライスやシステムタイマによる内部割り込み等によって確保出来ない汎用的なパソコン及びパソコン用の汎用ＯＳを用いた場合であっても、イベントに対応する処理がリアルタイムに実行されるよう制御することができる。

【００１９】

また、請求項１において、

前記タスク切替手段は、

前記繰り返し間隔毎にタイムスライスを行うタイムスライス手段をさらに有し、

前記イベント駆動手段は、

前記タイムスライス手段による各タイムスライス毎に強制的にイベント駆動処理を行うよう構成してもよい。

【００２０】

また、請求項８において、

前記イベント駆動ステップは、

前記繰り返し間隔毎に前記汎用オペレーティングシステムが行うタイムスライスの各タイムスライス毎に前記イベント駆動処理を行うよう構成してもよい。

【００２１】

この様にすると、前記イベント駆動処理を各タイムスライス毎に強制的に処理されるタスクとして実現することが出来る。従って、簡単な構成で、イベントに対応する処理がリアルタイムに実行されるよう制御することができる。

【００２２】

請求項３記載の発明は、
請求項１～２のいずれかにおいて、
前記リアルタイム制御手段は、
前記イベントの発生に対応する処理を行うプログラムが、前記イベントの発生を待つ旨の登録を行うイベント登録手段をさらに有し、
前記イベント駆動手段は、
前記イベント登録手段によって登録されたイベントの発生を検出したときは、前記イベント登録手段によって登録されたイベントの発生を待つプログラムが実行されるタスクへの切替を前記タスク切替手段に指示することを特徴とする。

【００２３】

請求項８記載の発明は、
請求項７において、
前記イベントの発生に対応する処理を行うプログラムが、前記イベントの発生を待つ旨の登録を行うイベント登録ステップをさらに有し、
前記イベント駆動ステップは、
前記イベント登録ステップにおいて登録されたイベントの発生を検出したときは、前記イベント登録ステップにおいて登録されたイベントの発生を待つプログラムが実行されるタスクへの切替を前記汎用オペレーティングシステムに指示することを特徴とする。

【００２４】

この様にすることにより、イベントと対応する処理を行うプログラムとの動的な対応をとることが出来、システム資源の有効活用を図ることが出来る。また、イベント駆動処理が終了した時点でイベント待ちの解除を行うことにより、リアルタイムにイベント待ち及びイベント待ちの解除を管理することができる。

【００２５】

請求項４記載の発明は、
請求項１～３のいずれかにおいて、
前記リアルタイム制御手段は、
前記ロボットの有するハードウエアリソースの変化と、マニピュレータ動作又は周辺装置を制御するプログラムで同期、通信を行うための出力データ群との少なくとも一方をイベントとして扱うことを特徴とする。

【００２６】

請求項９記載の発明は、
請求項７、８のいずれかにおいて、
前記ロボットの有するハードウエアリソースの変化と、マニピュレータ動作又は周辺装置を制御するプログラムで同期、通信を行うための出力データ群との少なくとも一方をイベントとして扱うことを特徴とする。

【００２７】

前記ハードウエアリソースとしてはＩ／Ｏポートやメモリにマッピングされた入出力ポートやマニピュレータ本体、ドライブボックスに設置されたシステムＩ／Ｏ、ＩＳＡバスに接続された特殊な基板類等がある。また、前記プログラムとは、マニピュレータ動作や周辺装置制御を行うためのユーザータスクやコントローラの内部状態等を監視するためのシステムタスク等で実行されるプログラムをさす。

【００２８】

この様に、ロボットで発生する各種状態をイベントとして扱うことで、これらの様々なイベントに対応した処理をリアルタイムで行うことが可能となる。

【００２９】

請求項５記載の発明は、
請求項４において、
前記リアルタイム制御手段は、
複数のタスクから参照更新可能な共有メモリ領域に、前記ハードウエアリソースの変化と

、マニピュレータ動作又は周辺装置を制御するプログラムで同期、通信を行うための出力データ群との少なくとも一方を管理するためのイベントリソース状態を記憶するイベントリソース状態記憶手段と、
前記マニピュレータ動作と前記周辺装置を制御するプログラムの同期、通信を行うための出力データ群との少なくとも一方に基づき、前記イベントリソース状態記憶手段に記憶されたイベントリソース状態を更新するイベントリソース状態更新手段とを有し、
前記イベント駆動手段は、
前記イベントリソース状態更新手段が更新したイベントリソース状態に基づき、ハードウエアリソースを更新するハードウエアリソース更新手段とを含むことを特徴とする。

【００３０】

請求項１０記載の発明は、
請求項９において、
複数のタスクから参照更新可能な共有メモリ領域に設定されたイベントリソーステーブルに記憶された、前記ハードウエアリソースの変化と、マニピュレータ動作又は周辺装置を制御するプログラムで同期、通信を行うための出力データ群との少なくとも一方を管理するためのイベントリソース状態を、
前記マニピュレータ動作と前記周辺装置を制御するプログラムの同期、通信を行うための出力データ群との少なくとも一方に基づき更新するイベントリソース状態更新ステップをさらに有し、
前記イベント駆動ステップは、
前記イベントリソース状態更新ステップにおいて更新されたイベントリソース情報に基づき、ハードウエアリソースを更新するハードウエアリソース更新ステップとを含むことを特徴とする。

【００３１】

本発明では各タスクから参照更新可能な共有メモリ領域にイベントリソースの状態を記憶するイベントリソーステーブルを設けている。そして、各タスクで処理されるプログラムの出力により変化するイベントリソースの状態に対応する前記イベントリソーステーブルの該当エリアが更新される。従って、前記イベントリソーステーブルを参照することにより各プログラムの出力によるイベントを検出することが出来るため、効率よくイベントを検出することができる。

【００３２】

また、イベントリソーステーブルのイベントリソースの状態の変化に基づき現実のハードウエアリソースの更新することが出来るため、各タスクで処理されるプログラムは現実のハードウエアリソースの更新を行う必要はない。したがって、ハードウエアリソースの更新一元化により、処理速度の向上を図ることが出来る。

【００３３】

請求項６記載の発明は、
請求項３又は請求項３に従属する請求項４、５のいずれかにおいて、
イベントの発生を待つ旨の登録を行ったプログラムを、所与の間隔で定期的に起動することをタスク切替手段に指示することで、該プログラムのスワップアウトを防止するスワップアウト防止手段を含むことを特徴とする。

【００３４】

請求項１１記載の発明は、
請求項８又は請求項８に従属する請求項９、１０のいずれかにおいて、
イベントの発生を待つ旨の登録を行ったプログラムを、所与の間隔で定期的に起動することを前記汎用オペレーティングシステムに指示することで該プログラムのスワップアウトを防止するスワップアウト防止ステップを含むことを特徴とする。

【００３５】

通常イベントの発生を待つ旨の登録を行ったプログラムは、イベントの発生に対応してリアルタイムに処理を行う必要がある。ところが、プログラムが一旦スワップアウトされて

しまうと再びロードされるまで数１０～数１００msecかかってしまう。そこで、この様にすると、イベントの発生を待つプログラムは、現実に処理要求があるか否かにかかわらず定期的に処理が起動されるため、プログラムのスワップアウトを防止することが出来、処理のリアルタイム性を確保することが出来る。

【００３６】

【発明の実施の形態】

以下、本発明の好適な実施の形態について、図面を参照して詳細に説明する。

【００３７】

本実施の形態のロボット用コントローラは、マニピュレータ動作或いは周辺装置制御を行うものである。この様なマニピュレータ動作或いは周辺装置制御はユーザーが行いたい制御にあわせてカスタマイズ出来る必要がある。このため、ユーザーは本コントローラに接続されたマニピュレータ等で実行させたい動作を本コントローラで実行可能なロボット制御用言語等で記述した各種アプリケーションプログラムを作成する。本コントローラは、この様なユーザーがマニピュレータ等の制御用に作成した各種アプリケーションプログラムが実行されるユーザータスクや、コントローラの内部状態を監視するためのシステムの処理を行ういくつかのタスクがマルチタスクで実行されるよう構成されている。

【００３８】

図３は、本コントローラ１０が行うマルチタスク処理の例を示した概念図である。同図に示すように、本コントローラ１０はマニピュレータの制御処理１２０－１、コンベアの制御処理１２０－２…等を制御するためのアプリケーションプログラムを含む各種プログラムが、それぞれタスク１１０－１、１１０－２…で、マルチタスク実行されるよう構成されている。すなわちＣＰＵを時分割して、前記複数のタスクに割り当てることで、見かけ上、複数のタスクを同時に実行させるよう構成されている。

【００３９】

この様な、ロボット用コントローラにおいては、前記各種プログラムで実行されるマニピュレータ等や周辺装置の制御はリアルタイム性が要求されることから、イベントが起こってからそのイベントの処理をするタスクが起動されるまでの時間が長時間であれば実用にならないという問題がある。

【００４０】

本実施の形態のロボット用コントローラ１０は、前記イベントに対応した処理をリアルタイムに実行するために以下のような構成を採用している。

【００４１】

まず、前提となる本システムのハードウエア構成を説明する。図２は本システムのハードウエア構成を表した図である。

【００４２】

同図に示すように、本コントローラ１０は、メインＣＰＵ２１０、ＨＤＤ２２０、ＲＡＭ２３０、インターフェイスボード２４０がＣＰＵバス２８０により接続されている。そしてマニピュレータ等２７０、システムＩ／Ｏ２６２等が設置されたドライブボックス２６０、外部タイマ２５２等を含む特殊な基板類２５０がインターフェイスボード２４０と、拡張バス２９０によって接続されている。

【００４３】

ＲＡＭ２３０には、本コントローラを制御するためのオペレーティングシステム２３２と、前記イベントに対応した処理をリアルタイムに実行するためコントロールプログラム２３４と、ユーザーがロボット制御用に作成したアプリケーションプログラム２３６を含む各種プログラム及びデータ等が格納されている。

【００４４】

前記オペレーティングシステム２３２は、汎用的なパソコン用のＯＳを用いて構成されている。該パソコン用汎用ＯＳは、プリエンプティブなマルチタスク機能や同期イベント機能などの標準的な機能を有する汎用的なパソコン用ＯＳであるが、いわゆるリアルタイムＯＳではない。リアルタイムＯＳでないとは、ＯＳによるタスク切替が低速でかつ、切替

時間の短縮設定が不可能なため、リアルタイム処理ができないＯＳをいう。なお、本実施の形態では、前記パソコン用汎用ＯＳとして、マイクロソフト社製のＷＩＮＤＯＷＳ９５（商品名）を利用している。

【００４５】

前記コントロールプログラム２３４とは、後述するイベント駆動部４０、ハードウエアリソース更新部４４、イベント登録部６０、イベントリソース状態記憶部７０、イベントリソース状態更新部８０、スワップアウト防止部９０の機能を実現するための各種プログラム及び記憶エリアを含んで構成されている。

【００４６】

なお、後述するタスク切替部３０については、主に前記オペレーティングシステム２３２のタスク切替機能を利用している。また、イベント駆動部４０及びイベント登録部６０は一部前記オペレーティングシステム２３２の同期イベント機能を利用している。

【００４７】

前記アプリケーションプログラム２３６とは、前述したユーザーがマニピュレータ等で実行させたい動作をロボット制御用言語等で記述したプログラムを指す。

【００４８】

また、イベントとは、マニピュレータ動作又は周辺装置等を制御する上で発生する各種事象をさす。具体的には、マニピュレータ等２７０やドライブボックス２６０で発生したシステムＩ／Ｏ２６２の変化に対応する事象や、基板類２５０等のハードウエアリソースの変化に対応する事象や、各プログラム２３２、２３４、２３６がタスク間での同期、通信を行うたの出力データ群により発生する事象がある。これらイベントを発生させる要因となる前記システムＩ／Ｏ、前記ハードウエアリソース、前記出力データ群等をイベントリソースという。

【００４９】

次に、本コントローラ１０がイベントに対応した処理をリアルタイムに実行するための機能について説明する。

【００５０】

図１は本コントローラ１０の機能ブロック図である。本コントローラ１０は、プリエンプティブにタスクを切り替えるタスク切替部３０と、外部タイマを用いて所定の繰り返し間隔毎に割り込み信号を発生させる外部割り込み発生部５０と、前記外部割り込み発生部５０が発生した割り込み信号に同期してイベント駆動処理を行うイベント駆動部４０と、前記イベントの発生に対応する処理を行うアプリケーションプログラム２３６が、前記イベントの発生を待つ旨の登録を行うイベント登録部６０と、イベントの発生等を管理記憶するためのイベントリソース状態を記憶するイベントリソース状態記憶部７０と、前記イベントリソース状態記憶部７０に記憶されたイベントリソース状態を更新するイベントリソース状態更新部８０と、イベントの発生を待っているアプリケーションプログラム２３６のスワップアウトを防止するスワップアウト防止部９０を含んで構成されている。

【００５１】

なお、外部割り込み発生部５０と、イベント駆動部４０と、イベント登録部６０と、イベントリソース状態記憶部７０と、イベントリソース状態更新部８０がリアルタイム制御手段２０として機能し、前記タスク切替部３０にタスクの切替を指示することにより、イベントの発生に対応する処理がリアルタイムに実行されるよう制御する。

【００５２】

さらに、前記イベント駆動部４０は、前記イベントリソース状態更新部８０が更新した情報に基づき、後述する所定のハードウエアリソースを更新するハードウエアリソース更新部４４とを含んで構成されている。

【００５３】

前記外部割り込み発生部５０は、拡張バス２９０に接続された基板類２５０に実装されている外部タイマ２５２を用いて、所定の繰り返し間隔毎に定期的に割り込み信号を発生させている。ここにおいて所定の繰り返し間隔とは、イベントの発生を検出して、該イベン

トに対応するプログラムをリアルタイムに起動するのに必要な短い間隔をさし、本実施の形態では１msecに設定している。

【００５４】

前述したように、本コントローラ１０のＲＡＭ２３０に格納されたオペレーティングシステム２３２はいわゆるリアルタイムＯＳでないため、システム内部でイベントに対応するプログラムをリアルタイムに検出するのに必要な短い間隔でタイムスライスを行うことが出来ない。従って本実施の形態では、この様に外部タイマ２５２を用いることにより、前記所定の繰り返し間隔毎に定期的に割り込み信号を発生させ、この割り込み信号を利用することにより、１msec間隔のタイムスライスが実現したのと同様の効果を得ることができる。

【００５５】

イベント登録部６０は、前記イベントの発生に対応する処理を行うアプリケーションプログラム２３６のイベント待ち登録要求により、前記アプリケーションプログラム２３６が、イベントの発生を待つ旨の登録を行う。前記アプリケーションプログラム２３６とは、図３にしめすマニピュレータの制御処理１２０－１、コンベアの制御処理１２０－２…等を制御するためにマルチタスクで実行されるプログラムのことであり、図２のＲＡＭ２３０に格納されている。また、イベントの発生とは、前述したような様々なイベントリソースの変化をいう。

【００５６】

マルチタスクで実行されているアプリケーションプログラム２３６がイベントリソースの変化に応じて起動されるためには、イベントリソースとその変化を待つ前記各種プログラムを対応づけておく必要がある。従って各アプリケーションプログラム２３６はイベントリソースの変化を待つ状況が発生した時に、イベント待ちの登録をイベント登録部６０に要求する。

【００５７】

イベント登録部６０は、この要求を受けて、イベントリソースをイベントオブジェクトに対応づけてイベント待ちの登録を行うのである。なお、ここにおいてイベントとは前述したような各種事象をさし、イベントオブジェクトとは、システムが前記各種事象を管理する際の単位となるものである。すなわち、システムが管理するのはイベントオブジェクトであり、各イベントオブジェクトは、自己の識別ＩＤとしてイベントハンドル有している。

【００５８】

なお、システムがイベントをイベントオブジェクトとして管理する機能はオペレーティングシステム２３２の同期イベント機能を利用することにより実現される。

【００５９】

この登録は具体的には、以下のようにして行われる。すなわちイベント登録部６０は、システム初期化時に予め適当な個数のイベントオブジェクトを生成しておく。

【００６０】

前記アプリケーションプログラム２３６からイベント登録要求が発生した場合には、使われていない（イベントリソースと対応していない）イベントオブジェクトをイベントリソースに割り当てる。既に登録されているイベントリソースに対して別のアプリケーションプログラム２３６から登録要求が発生した場合には、改めてイベントオブジェクトを割り当てるのではなく、既に登録されているイベントリソースに対応したイベントオブジェクトが用いられる。なお、ここにおいて用いられるとは、後述するようにイベントハンドルをアプリケーションプログラムに戻す際に、該イベントオブジェクトのイベントハンドルが前記別のアプリケーションプログラムに戻されるという意味である。また、予め生成しておいたイベントオブジェクトが足りなくなった場合には、その時点でイベントオブジェクトを新たに生成する。

【００６１】

また、後述するように、イベント待ちのアプリケーションプログラム２３６のスワップア

ウトを防止するために、ダミイ起動を行う際に使用するイベントオブジェクトへの登録も同時に行う。これは、アプリケーションプログラム毎に予め独立に生成されるイベントオブジェクトを使用する。イベント待ちの登録要求を行ったアプリケーションプログラム２３６には、このイベントオブジェクトを割り当て、ダミイ起動を実現させる。

【００６２】

そして登録が終わるとイベント登録部６０は、前記イベントオブジェクトの識別ＩＤであるイベントハンドルを、登録を要求したアプリケーションプログラム２３６に戻す。このとき、アプリケーションプログラム２３６は、イベントリソースに対応したイベントオブジェクトのイベントハンドルと、ダミイ起動用のイベントオブジェクトのイベントハンドルを受け取ることになる。

【００６３】

そして、アプリケーションプログラム２３６は、取得したイベントハンドルで示されるイベントオブジェクトの変化によって起動するように、オペレーティングシステム２３２に依頼する。これにより、前記各種プログラムは前記イベントオブジェクトとして登録されたイベントリソースの変化又はスワップアウト防止用にダミイ起動が起こるまで待機することになる。

【００６４】

なお、イベントハンドルがアプリケーションプログラムに戻されると、イベントリソースとアプリケーションプログラムの対応付けが行われたことになるが、この対応付けは、オペレーティングシステムが標準で有している同期イベント機能を利用している。すなわち、当該同期イベント機能を利用すると、イベントオブジェクトをシグナル状態にすることにより、対応するイベントハンドルを有する全てのプログラムのタスクを起動することができるのである。

【００６５】

図４は、イベント登録の具体的イメージを示した説明図である。前述したようにイベント登録部６０は、システム初期化時に予め、イベントを識別するためのイベントオブジェクト（対応するイベントハンドルがＥＨ１、ＥＨ２、ＥＨ３…）、を適当な個数と、ダミイ起動用のイベントオブジェクト（対応するイベントハンドルがＤＨ１、ＤＨ２…）を各アプリケーションプログラム毎に生成しておく（１）。

【００６６】

アプリケーションプログラムＡＰ２のタスクが実行されているとき、イベントリソースＥＲ２の変化を待つという事象が発生すると、該アプリケーションプログラムＡＰ２はイベント登録部６０に登録要求を出す（２）。このときイベントリソースＥＲ２を引数として、イベント登録部６０に渡す。

【００６７】

イベント登録部６０は、使われていないイベントオブジェクト、この場合、イベントオブジェクト（当該イベントオブジェクトのイベントハンドルはＥＨ１）がイベントリソースＥＲ１に割り当てられているため（３）、他のイベントオブジェクト（当該イベントオブジェクトのイベントハンドルはＥＨ２）をイベントリソースＥＲ２に割り当て（４）、該イベントオブジェクトのイベントハンドルＥＨ２、及びダミイ処理用のイベントオブジェクトのイベントハンドルＤＨ２をアプリケーションプログラムＡＰ２に戻す（５）。アプリケーションプログラムＡＰ２は、イベント登録部６０からもどされたイベントハンドルＥＨ２及びＤＨ２で示されるイベントオブジェクトがシグナル状態になるまで、待機するようにオペレーティングシステム２３２に依頼する（６）。

【００６８】

しかし、スワップアウト用のダミイ起動に関しては、スワップアウトしないようダミイ処理が行われるのみなので、登録、実行に関しては、ユーザーがアプリケーションプログラムを作成する際には意識しなくてよいよう構成されている。

【００６９】

なお、各アプリケーションプログラムにおいて、同時に複数のイベントリソースを登録す

ることも可能である。例えばアプリケーションプログラムにおいて複数のイベントリソース（ＥＲ１、ＥＲ３、ＥＲ４）の変化を待つという事象が発生した場合は、これらを引数としてイベント登録部６０に登録要求を出すと、以下のような処理がおこなわれる。

【００７０】

イベントリソースＥＲ１には既にイベントオブジェクトＥＨ１が対応しているため、あらたな対応づけは行われない。また、イベントリソースＥＲ３には、使われていないイベントオブジェクトＥＨ３（当該イベントオブジェクトのイベントハンドル）が割り当てられる。また、イベントリソースＥＲ４に割り当てるイベントオブジェクトが無い場合は、イベントオブジェクトＥＨ４（当該イベントオブジェクトのイベントハンドル）が新たに生成して割り当てられる。

【００７１】

そして、各割り当てられたイベントオブジェクトのイベントハンドルＥＨ１、ＥＨ３、ＥＨ４とダミイ用のイベントオブジェクトのイベントハンドルＤＨが前記アプリケーションプログラムに戻される。

【００７２】

イベントリソース状態記憶部７０は、各タスクから参照更新可能な共有メモリ領域に設けられており、各種イベントリソースの状態を記憶するイベントリソーステーブルが記憶されている。

【００７３】

該イベントリソーステーブルには、前述したように、マニピュレータ等２７０やドライブボックス２６０で発生したシステムＩ／Ｏ２６２、基板類２５０等のハードウエアリソースの状態や、各プログラム２３２、２３４、２３６がタスク間での同期、通信を行うための出力データ群の状態を記憶するよう構成されている。

【００７４】

なお、該イベントリソーステーブルのイベントリソースの状態は、後述するようにイベント駆動部４０により一定時間毎に、又イベントリソース状態更新部８０により必要に応じて更新される。

【００７５】

イベントリソース状態更新部８０は、主にアプリケーションプログラム２３６からの要求により、出力ポートの状態及びタスク間での同期、通信を行うための出力データ群の状態を更新する。

【００７６】

前記要求は、各アプリケーションプログラム２３６が、その実行中にイベントリソース状態更新部８０を起動する関数を呼び出すという形式で行われ、その際の引数として、出力データ群を渡す。

【００７７】

通常、各アプリケーションプログラム２３６は、前記出力データ群を、各アプリケーションプログラム２３６間で個別にやりとりする。しかし、本実施の形態では、各アプリケーションプログラム間でやりとりが必要な出力データ群が生じた場合は、前記関数によりイベントリソース状態更新部８０にその旨の要求を行うことになる。この要求をうけて、イベントリソース状態更新部８０は、イベントリソース状態記憶部７０の該当するイベントリソース状態の更新をおこなうのである。

【００７８】

従って、各アプリケーションプログラム２３６は、現実に出力データ群が生じ場合に行う出力ポート２４２の更新を行わず、これらは、後述するようにハードウエアリソース更新部４４が一元化して行うよう構成されている。

【００７９】

イベント駆動部４０は、前記外部割り込み発生部５０が発生させた割り込み信号に同期してイベント駆動処理を行う。イベント駆動処理とは、リアルタイム処理を実行するのに必要な繰り返し間隔で定期的にイベントを検出し、該イベントに対応する処理が実行される

タスクへの切替を前記タスク切替部３０に指示する処理である。

【００８０】

この様なイベント駆動部４０のイベント駆動処理はＣＰＵ２１０がＲＡＭ２３０に格納されたコントロールプログラム２３４及びオペレーティングシステム２３２の同期イベント機能を実行することにより実現される。すなわち、イベント駆動部４０の機能を実現する部分にあたるコントロールプログラム２３４は、システムに常駐しており、当該処理を行うタスクは前記外部割り込み発生部５０が発生させた割り込み信号に同期して駆動されるよう構成されているのである。そして、後述するように、イベントに対応するプログラムを起動する際にオペレーティングシステム２３２の同期イベント機能を利用するよう構成されている。

【００８１】

イベント駆動部４０はイベントを検出するために、以下のようにしてイベントリソースの監視を行っている。実際のイベントリソースの変化の具体的な監視方法は各デバイス毎に異なるが、入力ポート２４２に関しては、予め監視するポートアドレス、サイズなどを定義しておき、前記割り込みに同期して、定義された入力ポートをすべて参照し、イベントリソース状態記憶部９０に記憶されたイベントリソーステーブルの該当するエリアに入力ポート２４２の現在状態を保存する。イベントの検出については、参照した入力ポート２４２の状態とイベントリソーステーブルに保存された前回の状態を比較し判断する。

【００８２】

その他のマニピュレータ等２７０、ドライブボックス２６０に設置されたシステムＩ／Ｏ、拡張バス２９０によって接続されている特殊な基板類２５０に関しては、個別に作成されたデバイスドライバをコールして監視を行い、イベントリソース状態記憶部９０に記憶されたイベントリソーステーブルの該当するエリアに前記ロボット本体２７０、システムＩ／Ｏ、特殊な基板類２５０等の現在状態を保存する。イベントの検出については、デバイスドライバのコールによる監視結果とイベントリソーステーブルに保存された前回の状態を比較し判断する。

【００８３】

各アプリケーションプログラム２３６の出力データ群に関しては、通常、出力データ群の発生は出力ポート２４４に反映されている。

【００８４】

そしてイベント駆動部４０は、検出したイベントリソースの変化の中に、イベント登録部６０に登録されたイベントリソースがあった場合、対応するイベントオブジェクトをシグナル状態にすることで、タスク切替部３０に、対応するアプリケーションプログラム２３６のタスクの起動の指示を行う。

【００８５】

なお、同一のイベントリソースの変化を複数のアプリケーションプログラムで待っていた場合、又は同じタイミングで複数のイベントリソースに変化があった場合は、対応する１又は複数のイベントオブジェクトがシグナル状態になる。そして、オペレーティングシステムの機能により各アプリケーションプログラムに、ラウンドロビンに処理が切り替わる。

【００８６】

また、前述したように各アプリケーションプログラム２３６はイベントリソーステーブルを更新した場合、現実の出力ポート２４４の更新を行わないため、この更新を行う必要がある。従って、ハードウエアリソース更新部４４は、アプリケーションプログラムに出力データ群が発生した場合、対応する現実の出力ポート２４４の更新を行う。

【００８７】

なお、更新に必要な各アプリケーションプログラム２３６の出力データ群のサイズと、その現実の出力ポート２４４の出力アドレスは、予め登録しておくよう構成されている。

【００８８】

前記タスク切替部３０は、マルチタスクで処理を実行するために所定の間隔毎に、またイ

ベント駆動部４０又はスワップアウト防止部９０が、イベントオブジェクトをシグナル状態にして、対応するアプリケーションプログラム２３６のタスクの起動の指示を行った場合にタスクへの切替をおこなう。

【００８９】

前者のタスクへの切替として、本コントローラ１０では、マルチタスクで処理を実行するために、オペレーティングシステム２３２が標準の機能として有しているタイムスライス機能により、タスクの切替を行っている。

【００９０】

また、後者のタスクへの切替は、ＣＰＵ２１０がＲＡＭ２３０に格納されたオペレーティングシステム２３２の同期イベント機能を実行することにより、シグナル状態となったイベントオブジェクトに対応するイベントハンドルを有するアプリケーションプログラム２３６のタスクを起動することで実現される。

【００９１】

前述したように、同一のイベントリソースの変化を複数のアプリケーションプログラムで待っていた場合、又は同じタイミングで複数のイベントリソースに変化があった場合は、対応する１又は複数のイベントオブジェクトをシグナル状態にすることにより、前記タスク切替部３０が対応するイベントハンドルを有するアプリケーションプログラムをラウンドロビンで起動することになる。

【００９２】

なお、イベント登録を行った際に、イベントオブジェクトとアプリケーションプログラムは、オペレーティングシステムが標準で有している同期イベント機能により対応づけられている。従って、イベントオブジェクトをシグナル状態になると、対応するイベントハンドルを有するアプリケーションプログラム２３６のタスクを起動することができるのである。

【００９３】

図５（Ａ）（Ｂ）は、本システムによりリアルタイムにイベントを駆動する手順を示したフローチャート図である。同図（Ａ）は、イベント駆動部４０が所定の間隔毎に定期的に行うイベント駆動処理の手順を示しており、同図（Ｂ）は、イベント駆動されるアプリケーションプログラム２３６の動作の手順を示している。

【００９４】

同図（Ｂ）に示すように、あるアプリケーションプログラム２３６のタスクが起動されて、処理１を実行中にイベント待ちが発生すると（ステップ１１０）、アプリケーションプログラム２３６は、前述したようにイベント登録部にイベントリソースを指定してイベント待ち登録を要求し（ステップ１２０）、イベント待機状態になる（ステップ１３０）。すなわち、前記アプリケーションプログラム２３６が変化を待つイベントリソースがイベントオブジェクトに対応して登録され、前記アプリケーションプログラム２３６はイベントオブジェクトに対応したイベントハンドルを受け取り、イベントオブジェクトがシグナル状態になるまで待機するのである。

【００９５】

一方、同図（Ａ）に示すように、イベント駆動部４０は、前述した各種ハードウエアリソースの監視をおこなっている（ステップ１０）。そして、前記イベントリソーステーブルにイベントオブジェクトとして登録されているイベントリソースであるアプリケーションプログラム（ＡＰ）による出力データ群が変化したら（ステップ２０）、イベント駆動部４０のハードウエアリソース更新部４４は、対応するハードウエアリソースである出力ポート２４４の現実の更新を行う（ステップ３０）。

【００９６】

そして、登録されたイベントリソースに変化があった場合は（ステップ４０）、イベント登録部６０により登録されたイベントリソースに対応するイベントオブジェクトをシグナル状態にする。（ステップ５０）。タスク切替部３０に相当するオペレーティングシステム２３２は、イベントオブジェクトがシグナル状態になったのを受けて、前記対応するイ

ベントハンドルを有するアプリケーションプログラムが実行されるタスクを起動する（ステップ６０）。

【００９７】

すると同図（Ｂ）に示すように、イベントハンドルによりイベントを待機していたアプリケーションプログラム２３６は、処理２の実行を開始する（ステップ１４０）。

【００９８】

スワップアウト防止部９０は、リアルタイム性が要求される処理を行うアプリケーションプログラムが待機中である場合は、定期的に該アプリケーションプログラムを起動し、該アプリケーションプログラムがスワップアウトされないようしている。

【００９９】

リアルタイム性が要求される処理を行うアプリケーションプログラムとは、イベント待ち状態になっているアプリケーションプログラムをいい、これらがスワップアウトされてしまうと、イベントが発生して駆動される際、プログラムの再ロードに数１０～数１００ｍｅｓｃかかってしまい、処理のリアルタイム性が損なわれるからである。

【０１００】

従って、スワップアウト防止部９０は、所定の間隔毎にイベント登録部６０にダミイ起動用のイベントオブジェクトに対応するアプリケーションプログラムを起動するようタスク切替部３０に指示する。ここにおいて所定の間隔はシステムタイマを利用している。また、タスク切替部３０への指示は、以下のようにして行っている。

【０１０１】

各アプリケーションプログラムが、あるイベントリソースの変化を待つ旨の登録をイベント登録部６０に要求すると、前述したように前記アプリケーションプログラムに対してダミイ処理用のイベントオブジェクトのイベントハンドルが戻される。スワップアウト防止部９０では、定期的にこのダミイ処理用のイベントオブジェクトをシグナル状態にすることにより、タスク切替部３０に対応するイベントハンドルを有するアプリケーションプログラムのタスクの起動を指示する。

【０１０２】

アプリケーションプログラム側では、スワップアウト防止部９０によって駆動されたのかイベント駆動部４０によって駆動されたかはイベントオブジェクトの種類によって識別可能であるため、スワップアウト防止部９０によって駆動された場合、ダミイ処理を行い、再びイベント待ちを行うよう構成されている。なお、アプリケーションプログラム側で行うこの様な処理は、アプリケーションプログラムがイベント待ちの登録を行う際に呼び出す関数の行う処理として、コントロールプログラム側で用意しておくことが好ましい。この様にすると、ユーザーはアプリケーションプログラムを作成する際、そのようなことを特に意識することなく、該関数を利用するだけでよいからである。

【０１０３】

本実施の形態では、パソコン用汎用ＯＳとしてマイクロソフト社製のＷＩＮＤＯＷＳ９５（商品名）を用いた場合を例にとり説明したが他のパソコン用汎用ＯＳを使用してもよい。

【０１０４】

なお、本発明は、上記の実施例で説明したものに限らず、種々の変形実施が可能である。

【０１０５】

例えば、リアルタイム性の確保出来るパソコン用汎用ＯＳを用いた場合のロボット用コントローラ１０の実施の形態を説明する。

【０１０６】

リアルタイム性が確保出来るとは、ＯＳによるタスクの切替が高速に行えるオペレーティングシステムをいう。従ってこの様なオペレーティングシステムにおいては、十分に短い間隔をシステムタイマによって発生させることや、またリアルタイム制御を行うのに十分に短い間隔でタイムスライスを行うことが可能となる。

【０１０７】

前者の場合は、前記実施の形態の外部割り込み発生部５０において、外部タイマを用いて割り込みを発生されていたのを、システムタイマによって行うことにより、他は前記実施の形態と同様にして実現することが出来る。

【０１０８】

また後者の場合は、前記実施の形態の外部割り込み発生部５０において外部タイマを利用して割り込みを発生されていたのを、タイムスライスによって行うことにより、他は前記実施の形態と同様にして実現することが出来る。

【０１０９】

なお、いづれの場合も、システム内部でリアルタイム制御を行うのに十分に短い間隔を設定できるため、図２のハードウエア構成図における外部タイマは不要となる。

【０１１０】

ここでは、後者の場合を例に取り説明する。

【０１１１】

図６は、リアルタイム性の確保できるオペレーティングシステムを用いたロボット用コントローラ１０の機能ブロック図である。図１と異なるのは、外部割り込み発生部５０が無くなり、タスク切替部３０がタイムスライス部３２を含んでいる点である。他の同様な部分は図１と同名称、同番号を付してある。また、同様な機能についての説明は省略する。

【０１１２】

タイムスライス部３２は、リアルタイム制御を行うのに十分に短い間隔、例えば１msecでタイムスライスを行い、各タスクにラウンドロビン方式で、ＣＰＵを割り当て、各タスクの起動を行う。このとき各タイムスライス毎に、各タスクの実行に先立ち、イベント駆動部４０の駆動を行うよう構成されている。

【０１１３】

すなわちイベント駆動部４０の機能を実現するコントロールプログラム２３４は各タイムスライス毎に強制的に処理されるタスクとして、実行されることになる。そして、イベント駆動部４０によるイベント駆動処理は、１msec以下で終了するため、もしイベントが発生していない場合は、タイムスライスの残った時間は、通常のタイムスライスによる各タスクの起動がおこなわれることになる。

【０１１４】

この様にすることにより、それ以外の処理は、第一の実施の形態と全く同様にして、ロボット用コントローラの機能を実現することができる。

【０１１５】

なお、本実施の形態ではシステムで標準で用意されている同期イベント機能を利用して構成した場合を例に取り説明したが、同期イベント機能と同様の効果を奏するものであればよい。すなわち、システムで実行可能な各種関数やシステムコール等でもよい。また、同期イベント機能と同様の効果を奏するコントロールプログラムを作成してもよい。

【０１１６】

また、本コントローラ１０のオペレーティングシステム２３２及びコントロールプログラム２３４は、ＲＡＭ２３０に格納されている場合を例に取り説明したが、着脱自在に形成された外部記憶媒体に格納されている場合でもよいし、外部記憶媒体から、内部記憶媒体にロードして用いる場合でもよいし、外部から通信で内部記憶媒体にロードして用いる場合でもよい。

【０１１７】

さらに、本発明は、ロボットの種類、構成は問わず、様々なロボット用コントローラに適用可能である。

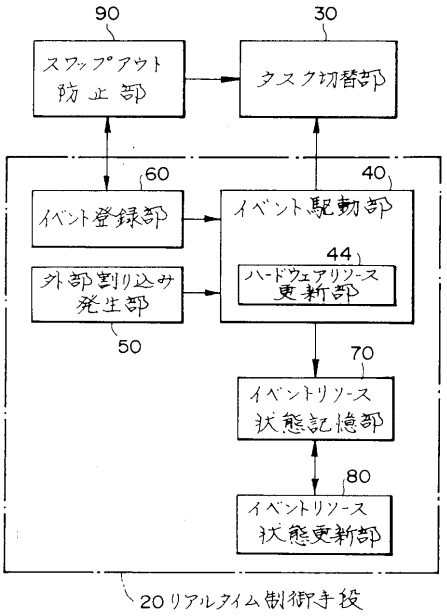
【０１１８】

また、本実施の形態ではロボット用コントローラのシステムについて説明したが、同様の構成でロボット用シーケンサも構成可能であり、本発明をロボット用シーケンサに適用した場合も本発明の技術的範囲に属する。

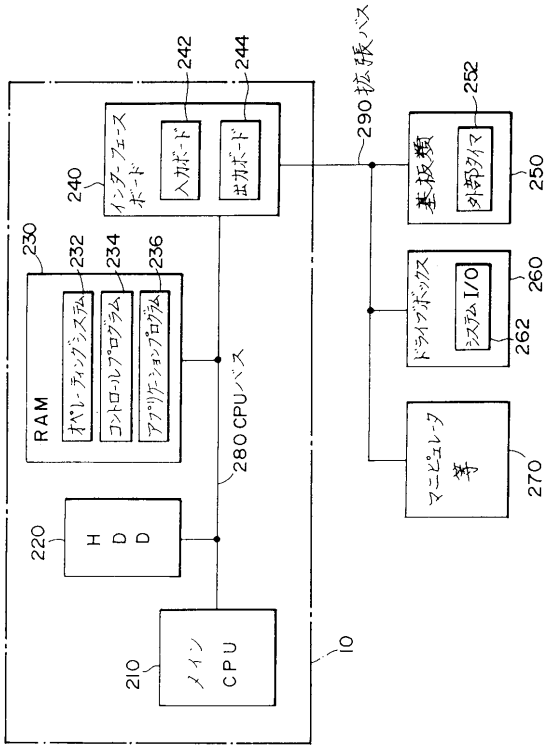
【０１１９】

【図面の簡単な説明】
【図１】本実施の形態のロボット用コントローラの機能ブロック図である。
【図２】本実施の形態のロボット用コントローラのハードウエア構成を表した図である。
【図３】本コントローラが行うマルチタスク処理の例を示した概念図である。
【図４】イベント登録の具体的イメージを示した説明図である。
【図５】同図（Ａ）（Ｂ）は、本システムによりリアルタイムにイベントを駆動する手順を示したフローチャート図である。
【図６】他の実施の形態のロボット用コントローラの機能ブロック図である。
【符号の説明】
１０　ロボット用コントローラ
２０　リアルタイム制御手段
３０　タスク切替部
３２　タイムスライス部
４０　イベント駆動部
４４　ハードウエアリソース更新部
５０　外部割り込み発生部
６０　イベント登録部
７０　イベントリソース状態記憶部
８０　イベントリソース状態更新部
９０　スワップアウト防止部
２１０　メインＣＰＵ
２２０　ＨＤＤ
２３０　ＲＡＭ
２３２　オペレーティングシステム
２３４　コントロールプログラム
２３６　アプリケーションプログラム
２４０　インターフェースボード
２４２　入力ポート
２４４　出力ポート
２５０　基板類
２５２　外部タイマ
２６０　ドライブボックス
２６２　システムＩ／Ｏ
２７０　マニピュレータ
２８０　ＣＰＵバス
２９０　拡張バス

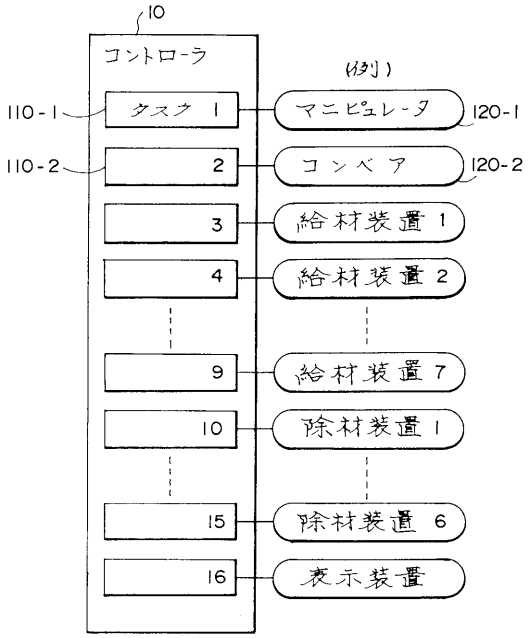
【図１】


90
スワップアウト防止部
30
タスク切替部
60
イベント登録部
40
イベント駆動部
44
ハードウェアリソース更新部
外部割り込み発生部
50
70
イベントリソース状態記憶部
80
イベントリソース状態更新部
20 リアルタイム制御手段

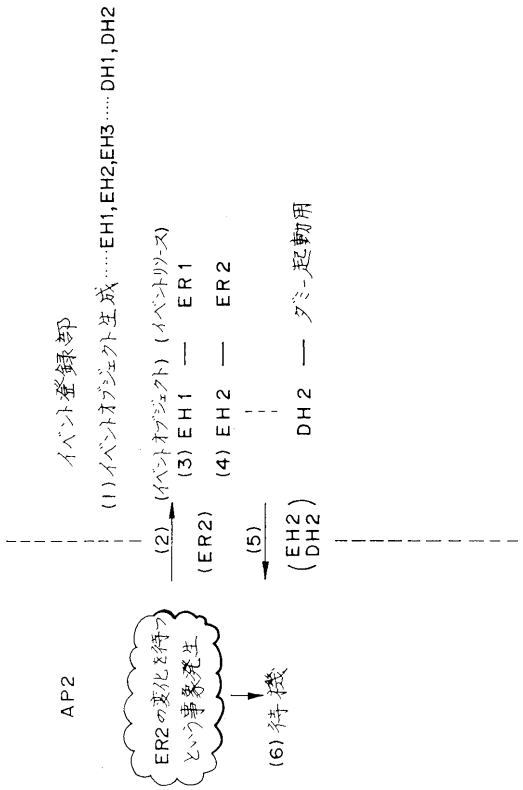
【図２】


290 拡張バス
242
244
240
インターフェースボード
入力ボード
出力ボード
252
基板類
外部メモリ
250
230
232
234
236
RAM
オペレーティングシステム
コントロールプログラム
アプリケーションプログラム
280 CPUバス
ドライブボックス
システムI/O
260
262
マニピュレータ等
270
220
HDD
210
メインCPU
10

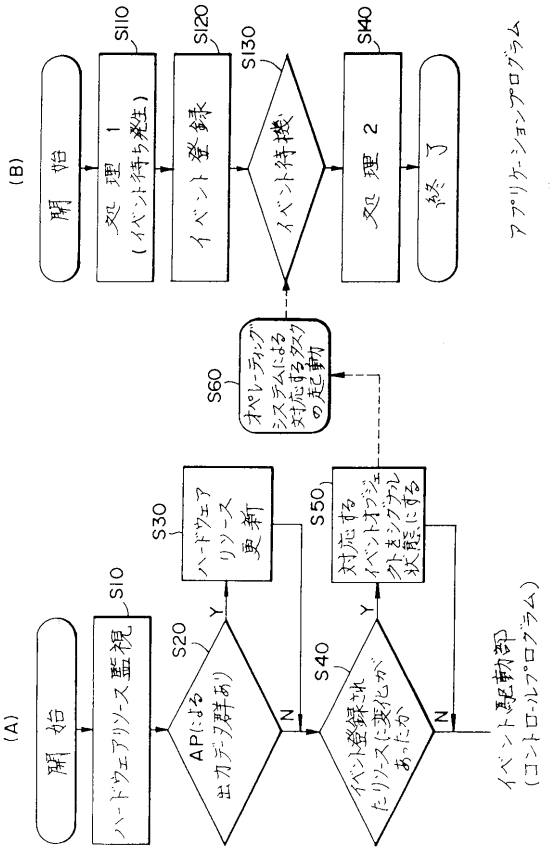
【図３】


10
コントローラ
(列)
110-1
タスク 1
マニピュレータ
120-1
110-2
2
コンベア
120-2
3
給材装置 1
4
給材装置 2
9
給材装置 7
10
除材装置 1
15
除材装置 6
16
表示装置

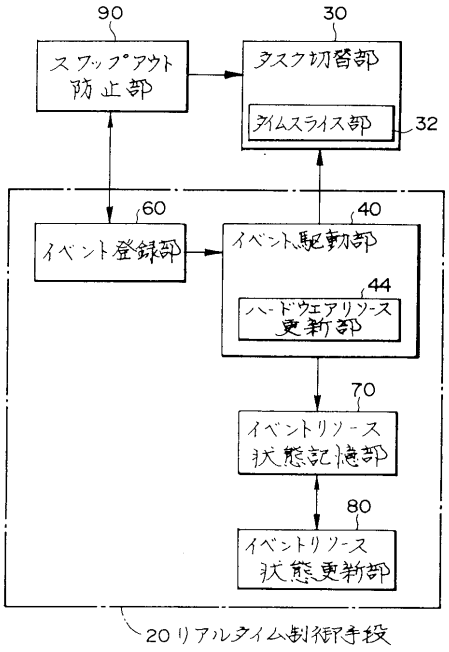
【図４】


イベント登録部
(1) イベントオブジェクト生成……EH1, EH2, EH3……DH1, DH2
(2)
(イベントオブジェクト) (イベントリソース)
(3) EH1 — ER1
(4) EH2 — ER2
DH2 — ダミー起動用
(5)
(ER2)
(EH2 DH2)
AP2
ER2の変化を待つという事象発生
(6) 待機

【図５】


(B)
開始
処理1
(イベント待ち発生)
S110
イベント登録
S120
イベント待機
S130
処理2
S140
終了
アプリケーションプログラム
S60
オペレーティングシステムによる対応するタスクの起動
S30
ハードウェアリソース更新
S50
対応するイベントオブジェクトをシグナル状態にする
(A)
開始
S10
ハードウェアリソース監視
S20
APによる出力データ有り
Y
N
S40
イベント登録されたリソースに変化があったか
Y
N
イベント起動部
(コントロールプログラム)

【図６】


90
スワップアウト防止部
30
タスク切替部
タイムスライス部
32
60
イベント登録部
40
イベント起動部
44
ハードウェアリソース更新部
70
イベントリソース状態記憶部
80
イベントリソース状態更新部
20 リアルタイム制御手段

フロントページの続き

(72)発明者　五味　一博
長野県諏訪市大和３丁目３番５号　セイコーエプソン株式会社内
(72)発明者　宮沢　比呂之
長野県諏訪市大和３丁目３番５号　セイコーエプソン株式会社内
(72)発明者　奥山　正幸
長野県諏訪市大和３丁目３番５号　セイコーエプソン株式会社内

審査官　二階堂　恭弘

(56)参考文献　特開昭５９－２０６８０８（ＪＰ，Ａ）
特開平３－７１２３４（ＪＰ，Ａ）
特開平１０－２１０９３（ＪＰ，Ａ）
特開平４－９２９０２（ＪＰ，Ａ）
特開平５－２０４６７２（ＪＰ，Ａ）
特開平８－１６４１０（ＪＰ，Ａ）
平井幸広他，レートモノトニック・スケジューリング・ポリシーに基づく倒立振子の制御について，第５０回（平成７年前期）全国大会講演論文集，日本，社団法人情報処理学会，１９９５年３月１７日，第４巻，第２４７－２４８ページ
周期スレッドを用いたマルチメディアデータの同期処理，情報処理学会研究報告，日本，社団法人　情報処理学会，１９９４年　3月　3日，Vol.94,No.19，第１－６頁

(58)調査した分野(Int.Cl.，ＤＢ名)
B25J 1/00-21/02
G05B19/18-19/46
G06F 9/46