

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第4387288号
(P4387288)

(45) 発行日 平成21年12月16日(2009.12.16)

(24) 登録日 平成21年10月9日(2009.10.9)

(51) Int.Cl.

F I

G 0 5 B 19/05 (2006.01)

G 0 5 B 19/05

N

請求項の数 7 (全 18 頁)

(21) 出願番号	特願2004-322687 (P2004-322687)	(73) 特許権者	000134109
(22) 出願日	平成16年11月5日(2004.11.5)		株式会社デジタル
(65) 公開番号	特開2006-134099 (P2006-134099A)		大阪府大阪市住之江区南港東8丁目2番5
(43) 公開日	平成18年5月25日(2006.5.25)		2号
審査請求日	平成18年10月17日(2006.10.17)	(74) 代理人	110000338
			特許業務法人原謙三国際特許事務所
		(74) 代理人	100080034
			弁理士 原 謙三
		(72) 発明者	清水 建二
			大阪府大阪市住之江区南港東8-2-52
			株式会社デジタル内
		審査官	佐藤 彰洋

最終頁に続く

(54) 【発明の名称】 制御用表示装置、エディタ装置、プログラムおよび記録媒体

(57) 【特許請求の範囲】

【請求項1】

予め定められた出力先へ、指定されたデバッグ情報を出力するデバッグ関数を含むスクリプト情報を格納する記憶手段と、

予め定められた事象が発生したときに、上記記憶手段に格納されたスクリプト情報の示すスクリプトを実行する制御手段とを有し、制御対象としてのデバイスの状態を表示する制御用表示装置において、

上記記憶手段には、当該デバッグ情報出力の有効/無効を示すフラグ情報が記憶されていると共に、

外部機器にて生成された上記スクリプト情報およびフラグ情報を受け取って上記記憶手段に書き込む書き込み手段を備え、

上記制御手段は、上記スクリプト情報の示すスクリプト中のデバッグ関数を実行するにあたって、上記記憶手段に格納されたフラグ情報が有効を示している場合にのみ、上記予め定められた出力先へ、当該デバッグ関数によって指定されたデバッグ情報を出力することを特徴とする制御用表示装置。

【請求項2】

上記スクリプト情報には、互いに異なる複数のトリガに、それぞれ対応するスクリプトを示す情報が含まれており、

上記フラグ情報には、全てのスクリプトに関連する第1フラグ情報と、上記各トリガに対応する第2フラグ情報とが含まれており、

10

20

上記制御手段は、上記複数のトリガのいずれかの発生を検出した場合、当該トリガに対応するスクリプト情報の示すスクリプトを実行すると共に、上記第1フラグ情報と、当該トリガに対応する第2フラグ情報との双方が有効を示している場合に、フラグ情報が有効であると判断することを特徴とする請求項1記載の制御用表示装置。

【請求項3】

上記フラグ情報には、上記外部機器によって設定される外部設定フラグ情報と、自機器で変更可能な内部設定フラグ情報とが含まれており、

上記制御手段は、上記内部設定フラグ情報および外部設定フラグ情報の双方が有効を示している場合にのみ、上記フラグ情報が有効であると判断すると共に、

予め定められた操作を受け付けた場合、操作に応じて、上記内部設定フラグ情報を変更する入力手段を備えていることを特徴とする請求項1記載の制御用表示装置。

【請求項4】

請求項1記載の制御用表示装置のフラグ情報およびスクリプト情報を生成するエディタ装置であって、

スクリプトの入力、および、上記デバッグ情報出力の有効/無効の入力を受け付ける受付手段と、

上記受付手段が入力を受け付けたスクリプトを示すスクリプト情報、および、上記受付手段が入力を受け付けた上記デバッグ情報出力の有効/無効を示すフラグ情報を出力する出力手段とを備えていることを特徴とするエディタ装置。

【請求項5】

請求項2記載の制御用表示装置のフラグ情報およびスクリプト情報を生成するエディタ装置であって、

複数のスクリプトと、それぞれに関連するトリガと、上記全てのスクリプトを実行する際のデバッグ情報出力の有効/無効と、上記各トリガに関連するスクリプトを実行する際のデバッグ情報出力の有効/無効との入力を受け付ける受付手段、

並びに、上記受付手段への入力に基づいて、上記各スクリプトを示すスクリプト情報と、それぞれに関連するトリガの情報と、上記全てのスクリプトを実行する際のデバッグ情報出力の有効/無効を示す第1フラグ情報と、上記各トリガに関連するスクリプトを実行する際のデバッグ情報出力の有効/無効を示す第2フラグ情報とを出力する出力手段を備えていることを特徴とするエディタ装置。

【請求項6】

請求項1～5のいずれか1項に記載の各手段として、コンピュータを動作させるプログラム。

【請求項7】

請求項6記載のプログラムが記録された記録媒体。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、スクリプトを実行可能な制御用表示装置、および、そのスクリプトを示すスクリプト情報を生成するエディタ装置に関するものである。

【背景技術】

【0002】

従来から、制御システムのHMI(Human Machine Interface)としてプログラマブル表示器が広く使われている。当該プログラマブル表示器は、例えば、後述の特許文献1に示すように、画面上の領域と、当該領域への表示や入力に対応するデバイスとの対応関係を示すタグを組み合わせて構成された画面データに基づいて、入力および画面表示を制御している。また、プログラマブル表示器の操作画面等を設計する設計者は、エディタ装置によって、制御システムの実情に合わせた画面データを作成できる。

【0003】

したがって、当該画面データに基づいて、上記プログラマブル表示器を動作させること

10

20

30

40

50

によって、プログラマブル表示器は、制御システムのターゲットシステムに含まれるデバイスの状態を的確にオペレータに通知し、当該オペレータによる操作を受け付けることができる。

【 0 0 0 4 】

また、よりきめ細かな制御を実現するために、上記特許文献 1 記載のプログラマブル表示器は、プログラマブル表示器専用提供されたスクリプト言語で記述されたスクリプトを実行することができる。

【 0 0 0 5 】

一方、後述する特許文献 2 には、予めデバッグ時に必要なデバッグ情報を表示し、かつ、プログラムを一時停止されるデバッグ関数を組み込んだプログラムをコンパイルして計算機にセットし、この計算機でデバッグを行うか否かを選択するデバッグスイッチを OFF 状態としてプログラムを実行開始したときには、プログラム中のデバッグ関数をスキップして通常のプログラムとして使用し、デバッグスイッチを on 状態としてプログラムを実行開始したときには、デバッグ情報を表示すると共に、デバッグストップ関数のレベル値が設定範囲内であるときには、プログラムを続行ボタンが押下されるまで一時停止させる構成が記載されている。

【特許文献 1】特開 2 0 0 1 - 0 5 6 7 0 4 号公報（公開日：2001年 2 月 2 7 日）

【特許文献 2】特開平 6 - 1 4 9 6 1 6 号公報（公開日：1994年 5 月 3 1 日）

【発明の開示】

【発明が解決しようとする課題】

【 0 0 0 6 】

しかしながら、上記特許文献 1 の構成では、プログラマブル表示器がスクリプトを実行可能になり、プログラマブル表示器の操作画面等の設計者が、当該スクリプトによって、上記タグでは指定できないような動作を、プログラマブル表示器に指示できるようになった反面で、スクリプトに不具合（バグ）が見つかった場合は、基本的に、設計者がスクリプトをデバッグする必要がある。

【 0 0 0 7 】

ここで、以下の手順、すなわち、デバッグの際は、例えば、図 1 1 に示すように、スクリプト中に、例えば、その時点でのデバイスの状態など、デバッグに入用な情報（デバッグ情報）を表示するための命令を挿入してデバッグすると共に、デバッグに成功したときには、スクリプトから、これらの命令を削除するという手順で、デバッグすると、デバッグの終了時の手間が増加してしまうという問題を生じる。また、特許文献 2 の構成は、プログラマブル表示器に適用したものではない。なお、図 1 1 では、一例として、一行からなるスクリプトに、1 行からなる表示命令を追加した場合を示しているが、一般に、スクリプトは、複数の行からなり、適切にデバッグ情報を表示するためには、複数箇所に、デバッグ情報を表示するための命令を挿入することが多い。したがって、これらの命令を削除する際の手間は大きくなりがちである。

【 0 0 0 8 】

本発明は、上記の問題点に鑑みてなされたものであり、その目的は、デバッグ効率の高い制御用表示装置、および、そのためのスクリプト情報およびフラグ情報を生成するエディタ装置を実現することにある。

【課題を解決するための手段】

【 0 0 0 9 】

本発明に係る制御用表示装置は、上記課題を解決するために、スクリプトを示すスクリプト情報を格納する記憶手段と、予め定められた事象が発生したときに、上記記憶手段に格納されたスクリプト情報の示すスクリプトを実行する制御手段とを有する制御用表示装置において、上記スクリプトには、予め定められた出力先へ、指定されたデバッグ情報を出力するデバッグ関数を含ませることができ、上記記憶手段には、当該デバッグ情報出力の有効／無効を示すフラグ情報が記憶されていると共に、上記制御手段は、上記スクリプト情報の示すスクリプト中のデバッグ関数を実行するにあたって、上記記憶手段に格納さ

10

20

30

40

50

れたフラグ情報が有効を示している場合にのみ、上記予め定められた出力先へ、当該デバッグ関数によって指定されたデバッグ情報を出力することを特徴としている。

【 0 0 1 0 】

上記構成の制御用表示装置は、スクリプトを実行できる制御手段を備えているので、デバイスの状態を表示する際に、より柔軟に表示動作を制御できる一方で、スクリプトに不具合（バグ）があると、その不具合を解消するためにスクリプトを修正（デバッグ）する必要がある。

【 0 0 1 1 】

上記構成では、スクリプトに予め定められた出力先へ、指定されたデバッグ情報を出力するデバッグ関数を含ませることができ、上記制御手段は、記憶手段に格納されたフラグ情報が、上記デバッグ関数によるデバッグ情報出力が有効であることを示している場合には、上記予め定められた出力先へ、当該デバッグ関数によって指定されたデバッグ情報を出力する。これとは逆に、記憶手段のフラグ情報が無効を示している場合、上記制御手段は、記憶手段に格納されているスクリプト情報の示すスクリプトがデバッグ関数を含んでいる場合であっても、当該デバッグ関数によるデバッグ情報出力を行わない。

10

【 0 0 1 2 】

したがって、制御用表示装置は、上記フラグ情報を変更するだけで、スクリプト情報を変更することなく、デバッグ情報を出力するか否かを切り換えることができ、制御用表示装置におけるスクリプトのデバッグ効率を向上できる。

【 0 0 1 3 】

20

また、上記構成に加えて、外部装置にて生成された上記スクリプト情報およびフラグ情報を受け取って上記記憶手段に書き込む書き込み手段を備えていてもよい。さらに、上記構成に加えて、上記記憶手段は、外部装置にて生成された上記スクリプト情報およびフラグ情報を格納した記憶媒体を装着可能であり、当該記憶媒体から、上記スクリプト情報およびフラグ情報を読み出してもよい。

【 0 0 1 4 】

これらの構成では、制御用表示装置は、上記外部装置にて生成されたフラグ情報に応じて、デバッグ情報を出力するか否かを変更できる。したがって、制御用表示装置のスクリプトのデバッグが終了した時点において、外部装置（例えば、スクリプト情報およびフラグ情報を生成するエディタ装置など）のユーザは、無効を示すフラグ情報を生成し、制御用表示装置の記憶手段に書き込ませるか、あるいは、記憶手段に読み取らせることによって、制御用表示装置におけるデバッグ情報出力を禁止できる。これにより、例えば、制御用表示装置のオペレータが表示されているデバイスの状態を確認したり、操作したりしている時点のように、不所望な時点で制御用表示装置にデバッグ情報が表示されてしまうという不具合を防止でき、制御用表示装置の安全性を向上できる。

30

【 0 0 1 5 】

また、上記構成に加えて、上記スクリプト情報には、互いに異なる複数のトリガに、それぞれ対応するスクリプトを示す情報が含まれており、上記フラグ情報は、全てのスクリプトに関連する第1フラグ情報と、上記各トリガに対応する第2フラグ情報とが含まれており、上記制御手段は、上記複数のトリガのいずれかの発生を検出した場合、当該トリガに対応するスクリプト情報の示すスクリプトを実行すると共に、上記第1フラグ情報と、当該トリガに対応する第2フラグ情報との双方が有効を示している場合に、フラグ情報が有効であると判断してもよい。

40

【 0 0 1 6 】

当該構成では、第1および第2フラグ情報の組み合わせによって、デバッグ情報を出力するか否かが決定されるので、第1フラグ情報のみによって決定する場合よりも柔軟に、しかも、第2フラグ情報のみによって決定する場合よりも簡単に、デバッグ情報を出力するか否かを設定できる。

【 0 0 1 7 】

さらに、上記構成に加えて、上記フラグ情報には、外部機器によって設定される外部設

50

定フラグ情報と、自機器で変更可能な内部設定フラグ情報とが含まれており、上記制御手段は、上記内部および外部フラグ情報の双方が有効を示している場合にのみ、上記フラグ情報が有効であると判断すると共に、予め定められた操作を受け付けた場合、操作に応じて、上記内部設定フラグ情報を変更する入力手段とを備えていてもよい。

【0018】

当該構成では、制御手段は、上記内部および外部フラグ情報の双方が有効を示している場合には、デバッグ情報を出力し、いずれかが無効を示している場合には、デバッグ情報を出力しない。したがって、外部フラグ情報が有効に設定されている場合は、制御用表示装置の入力手段によって、内部設定フラグ情報を変更することによって、デバッグ情報を出力するか否かを変更でき、デバッグ効率を向上できる。また、外部設定フラグ情報を無効に設定することによって、デバッグ情報の出力を禁止できる。これらの結果、デバッグ効率の向上と安全性の向上との双方を達成できる。

10

【0019】

一方、本発明に係るエディタ装置は、上記課題を解決するために、上記制御用表示装置のフラグ情報およびスクリプト情報を生成するエディタ装置であって、スクリプトの入力、および、上記デバッグ情報出力の有効/無効の入力を受け付ける受付手段と、上記受付手段が入力を受け付けたスクリプトを示すスクリプト情報、および、上記受付手段が入力を受け付けた上記デバッグ情報出力の有効/無効を示すフラグ情報を出力する出力手段とを備えている。また、本発明に係るエディタ装置は、上記課題を解決するために、上記第1および第2フラグ情報を参照して動作する制御用表示装置のフラグ情報およびスクリプト情報を生成するエディタ装置であって、複数のスクリプトと、それぞれに関連するトリガと、上記全てのスクリプトを実行する際のデバッグ情報出力の有効/無効と、上記各トリガに関連するスクリプトを実行する際のデバッグ情報出力の有効/無効との入力を受け付ける受付手段、並びに、上記受付手段への入力に基づいて、上記各スクリプトを示すスクリプト情報と、それぞれに関連するトリガの情報と、上記全てのスクリプトを実行する際のデバッグ情報出力の有効/無効を示す第1フラグ情報と、上記各トリガに関連するスクリプトを実行する際のデバッグ情報出力の有効/無効を示す第2フラグ情報とを出力する出力手段を備えている。

20

【0020】

これらの構成では、入力に応じたフラグ情報あるいは第1および第2フラグ情報を出力することによって、上記制御用表示装置において、デバッグ情報が出力されるか否かを制御できる。したがって、デバッグ情報の出力するか否かを制御可能で、しかも、デバッグ効率の高い制御用表示装置の安全性を向上させることができる。

30

【0021】

ところで、上記制御用表示装置およびエディタ装置は、ハードウェアで実現してもよいし、プログラムをコンピュータに実行させることによって実現してもよい。具体的には、本発明に係るプログラムは、上記制御用表示装置またはエディタ装置の各手段としてコンピュータを動作させるプログラムであり、本発明に係る記録媒体には、当該プログラムが記録されている。

【0022】

これらのプログラムがコンピュータによって実行されると、当該コンピュータは、上記制御用表示装置あるいはエディタ装置として動作する。したがって、上記制御用表示装置およびエディタ装置と同様の効果が得られる。

40

【発明の効果】

【0023】

本発明に係る制御用表示装置は、スクリプト中のデバッグ関数を実行するにあたって、記憶手段に格納されたフラグ情報が有効を示している場合にのみ、予め定められた出力先へ、当該デバッグ関数によって指定されたデバッグ情報を出力する制御手段を備えているので、制御用表示装置のデバッグ効率を向上できる。また、本発明に係るエディタ装置は、制御用表示装置のフラグ情報を生成できるので、制御用表示装置のデバッグ効率および

50

安全性を向上させることができる。

【発明を実施するための最良の形態】

【0024】

本発明の一実施形態について図1ないし図10に基づいて説明すると以下の通りである。すなわち、本実施形態に係る制御システムは、スクリプトの記述を変えることなく、デバッグ用のメッセージを、制御システムのHMIとしての制御用表示装置に表示するか否かを、一括して変更可能とすることによって、スクリプトのデバッグ効率を向上できるため、スクリプトを実行可能な制御用表示装置が備えられた制御システムに広く好適に用いることができる。

【0025】

以下では、制御用表示装置の一例としてのプログラマブル表示器の構成について説明する前に、制御システム全体の概略構成および動作について説明する。

【0026】

すなわち、本実施形態に係る制御システム1は、図1に示すように、ターゲットシステム2のデバイス2aを制御するためのシステムであって、例えば、ベルトコンベアー式の自動組付機などのデバイス2aの制御に好適に用いられている。

【0027】

上記制御システム1には、上記デバイス2aを制御する制御装置としてのPLC11と、多くの場合、制御対象近傍に配されると共に、制御システム1のHMIとして、デバイス2aの状態を表示し、オペレータによるデバイス2aへの操作を受け付けるプログラマブル表示器13とを備えている。

【0028】

本実施形態に係る制御システム1では、各プログラマブル表示器13は、イーサネット（登録商標）などのLAN（ローカルエリアネットワーク）15によって互いに接続されている。さらに、上記LAN15には、多くの場合、プログラマブル表示器13よりも離れた場所から、制御システム1全体を管理する制御用ホストコンピュータ17が接続されている。

【0029】

また、上記各プログラマブル表示器13は、シリアルケーブルなどの通信路を介して、それぞれに対応するPLC11と接続されている。なお、図1では、説明の便宜上、LAN15にプログラマブル表示器13が2台接続されると共に、各プログラマブル表示器13に1台ずつのPLC11、各PLC11に1台ずつのデバイス2aがそれぞれ接続されている場合を例示しているが、当然ながら、それぞれの接続台数は任意に設定できる。

【0030】

また、デバイスは、デバイスアドレスにより特定可能で、しかも、状態を取得したり、制御（変更）できるものであれば、例えば、デバイス2a自体であってもよいし、例えば、PLC11やプログラマブル表示器13の記憶装置など、制御システム1に設けられた記憶装置の一領域を示してもよい。

【0031】

ここで、上記制御システム1では、制御システム1に必須の構成であり、しかも、HMIとして動作するため、演算能力に余力のあるプログラマブル表示器13が通信の大半を処理するように構成されている。さらに、各プログラマブル表示器13は、自らに接続されているPLC11の機種に固有の専用プロトコルと、LAN15での共通プロトコルとを変換して、他のプログラマブル表示器13や制御用ホストコンピュータ17と、PLC11との通信を中継する。なお、共通プロトコルと専用プロトコルとの間のプロトコル変換には、同じ指示に同じコードが割り当てられるように予め定められた共通のコマンドと上記共通のコマンドに対応するPLC11固有のコマンドとの間の変換や、データやアドレスの表現方法の変換、デバイスアドレスと、当該デバイスアドレスに対応し、しかも、デバイスアドレスとは異なる値に設定可能な変数や変数の名称（変数名）との変換なども含まれる。

10

20

30

40

50

【 0 0 3 2 】

これにより、プログラマブル表示器 1 3 および制御用ホストコンピュータ 1 7 は、他のプログラマブル表示器 1 3 に接続されている P L C 1 1 の機種に拘らず、L A N 1 5 を介して共通のプロトコルで通信できる。この結果、互いに異なる機種の P L C 1 1 の混在する制御システム 1 が比較的容易に実現されている。

【 0 0 3 3 】

上記プログラマブル表示器 1 3 は、後述する画面データに基づいて、デバイスの状態を画面表示する際の動作や、画面への操作に応じてデバイスの状態を制御する際の動作を特定するものであって、P L C 1 1 と通信する P L C ・ I F 部 3 1 と、上記 L A N 1 5 に接続するための L A N ・ I F 部 3 2 と、例えば、液晶表示装置などからなるディスプレイ 3 3 と、ディスプレイ 3 3 の画面上に配されたタッチパネルなどからなる入力装置 3 4 と、上記各部材 3 1 ~ 3 4 を制御する H M I 処理部 3 5 と、当該 H M I 処理部 3 5 によって参照され、上記画面データおよび上記変数が格納される記憶部 3 6 とを備えている。

10

【 0 0 3 4 】

本実施形態に係るプログラマブル表示器 1 3 は、例えば、プログラマブル表示器であって、上記画面データは、画面上の領域と、当該領域への表示や入力に対応するデバイスとの対応関係を示すタグを組み合わせて構成されている。本実施形態では、H M I 処理部 3 5 が複数の単位画面（ベース画面）を切り換え表示可能であり、上記タグは、当該タグが有効となる単位画面を示すファイル番号と、単位画面上で実行すべき動作内容を特定する事象名と、各事象毎に参照される参照情報とを含んでいる。

20

【 0 0 3 5 】

例えば、上記タグが所定の画面領域（表示座標範囲）へ所定のデバイスの状態に応じた部品図形を表示する表示タグの場合、上記参照情報には、表示座標範囲と、デバイスを特定可能な変数と、例えば、部品図形がスイッチの場合、O N を示す図形のファイルおよび O F F を示す図形のファイルなど、表示時に参照するファイル番号とが含まれる。さらに、タグが入力タグの場合、参照情報として、有効入力座標範囲と、入力結果が書き込まれるデバイスの変数とが含まれる。

【 0 0 3 6 】

また、記憶部 3 6 には、各変数について、変数の名称（変数名）と、変数に対応するデバイス 2 a または内部メモリを特定するための情報（例えば、アドレスなど）と、変数の内容との組み合わせが記憶されている。なお、本実施形態では、変数に対応するデバイス 2 a の機種に拘わらず、変数の内容を格納する際の表現方法（例えば、ワード長や符号の有無、あるいは、B C D / 2 進表記など）が予め定められた表現方法に統一されており、変数が実在のデバイス 2 a に対応する場合、記憶部 3 6 には、実際の機種での表現方法も格納されている。この場合、H M I 処理部 3 5 は、P L C ・ I F 部 3 1 を介してデバイス 2 a の状態を取得あるいは制御する際、表現方法を形式変換して、格納時の表現方法を統一する。

30

【 0 0 3 7 】

一方、上記 H M I 処理部 3 5 は、所定の時間間隔で、記憶部 3 6 に格納された画面データから、ベース画面のファイル番号が、現在表示中のベース画面である表示タグを抽出する。さらに、H M I 処理部 3 5 は、記憶部 3 6 に格納された変数を参照して、タグに対応する変数の内容を読み出し、内容に応じた部品図形をディスプレイ 3 3 に表示する。ここで、P L C ・ I F 部 3 1 に接続された P L C 1 1 により制御されるデバイス 2 a に、上記変数が対応している場合、H M I 処理部 3 5 は、P L C ・ I F 部 3 1 による P L C 1 1 との通信によって、デバイス 2 a の状態を取得し、上記変数の内容を当該状態に応じて更新している。これにより、ディスプレイ 3 3 には、例えば、図 2 に示すように、デバイス 2 a の状態が表示される。

40

【 0 0 3 8 】

なお、変数が他のプログラマブル表示器 1 3 に接続された P L C 1 1 によって制御されるデバイス 2 a の場合、H M I 処理部 3 5 は、L A N ・ I F 部 3 2、L A N 1 5 および他

50

のプログラマブル表示器 13 を介して当該 PLC 11 と通信するなどして、デバイスの状態を取得し、それに応じて変数の内容を更新する。

【0039】

また、入力装置 34 への押し操作など、オペレータの入力操作を受け付けると、HMI 処理部 35 は、上記画面データから、現在表示中のベース画面に対応し、当該入力操作にマッチする入力タグを検索すると共に、入力結果に応じて、タグに対応する変数の内容を更新する。さらに、HMI 処理部 35 は、デバイス 2a の状態を取得する場合と略同様に、PLC 11 やプログラマブル表示器 13 と通信するなどして、上記変数の内容に応じて、デバイス 2a の状態を制御させる。ここで、入力操作の後も、HMI 処理部 35 は、デバイス 2a の状態を画面表示するので、操作結果が画面表示に反映される。

10

【0040】

なお、HMI 処理部 35 は、例えば、他のプログラマブル表示器 13 や制御用ホストコンピュータ 17 など、LAN 15 に接続された機器から自らに接続された PLC 11 のデバイス 2a への制御指示を受け取った場合や、これとは逆に、上記機器へ報告すべきデバイス 2a の状態を自らの PLC 11 から受け取った場合には、上述したプロトコル変換によって、LAN 15 での通信と PLC 11 相手の通信との間を中継できる。

【0041】

一方、上記制御用ホストコンピュータ 17 は、制御システム 1 の監視制御を行う HMI 処理部 71 と、LAN 15 に接続され、上述の共通プロトコルで通信する LAN・IF 部 72 と、LAN・IF 部 72 および LAN 15 を介して各プログラマブル表示器 13 と通信して、HMI 処理部 71 などの要求に応じるサーバ部 73 とを備えており、ターゲットシステム 2 や PLC 11 あるいはプログラマブル表示器 13 の状態を表示したり制御できる。

20

【0042】

ここで、プログラマブル表示器 13 が上記タグを組み合わせて構成された画面データに基づいて動作する場合、詳細は後述するように、デバイスの状態をオペレータへ通知し、オペレータからの操作を受け付ける操作画面を設計する設計者は、各タグに、画面上の領域およびデバイスを関連付けるだけで、プログラマブル表示器 13 の動作（例えば、プログラマブル表示器 13 が操作画面を表示する動作、操作画面への操作を受け付ける動作、および、他の機器へデバイス 2a の状態変更を指示する動作など）を指定できるので、これらの動作に応じてプログラマブル表示器 13 のプログラムを変更する場合よりも容易に、これらの動作を指定できる。

30

【0043】

ところが、プログラマブル表示器 13 に要求される動作は、例えば、制御対象とするターゲットシステム 2 や、制御システム 1 のユーザが集めたいと考えているデータなどの相違によって、制御システム 1 毎に互いに異なっていることが多い。したがって、プログラマブル表示器 13 の動作全てを、タグによって表現できるように、各種タグを予め用意しておくことは難しい。

【0044】

したがって、上記プログラマブル表示器 13 は、上記タグによって指定された動作だけではなく、図 3 に示すように、スクリプト（プログラム）によって指定された動作も実行できるように構成されている。ここで、スクリプト中では、式を評価する動作や、評価結果に応じて、スクリプトのうち、次に実行する部分を変更する動作を記述できるだけではなく、デバイスの状態を評価する動作や、デバイスの状態を評価結果に応じて変更する動作も記述できる。これにより、スクリプトを実行するプログラマブル表示器 13 は、デバイスの状態を、上記操作画面の設計者に指示された評価方法で評価し、それに応じて、デバイスの状態を変更することができる。なお、プログラマブル表示器 13（より詳細には、HMI 処理部 35）は、テキストで表現されたスクリプトの構成要素（例えば、文や式など）を逐次変換（インタープリット）して実行してもよいが、以下では、スクリプトを、中間コードに、予め翻訳（コンパイル）しておく場合について説明する。

40

50

【 0 0 4 5 】

具体的には、本実施形態において、上記記憶部 3 6 のうち、画面データを格納する記憶領域には、図 1 に示すように、スクリプトを記憶する記憶領域 M S が設けられており、当該記憶領域 M S には、図 3 に示すように、テキストで表現されたスクリプトに対応した中間コード群が格納されている。中間コードは、図 4 に示すように、中間コード（図の例では、V C 1）の機能を示す O P コード（機能コード；図の例では、O P 1）を含んでいる。また、中間コードには、当該 O P コード以外にも、必要に応じて、1 以上のオペランド（図の例では、O P R 1 ~ O P R 3）を含ませることができ、H M I 処理部 3 5 は、中間コードにオペランドが含まれている場合、当該オペランドの値を参照しながら、O P コードの示す機能を実行できる。

10

【 0 0 4 6 】

例えば、図 3 に示す文 S T 1 は、あるデバイスアドレス「L S 0 1 0 0」で特定されるデバイスの内容を、1 増加させることを示している。また、文 S T 1 では、「w :」と記述することによって、当該デバイスがワード長のデータ領域を持ったデバイスであること（デバイスの内容がワード長であること）を示している。

【 0 0 4 7 】

ここで、一例として、上記文 S T 1 が 1 つの中間コードに対応している場合について説明すると、図 3 に示す分 S T 1 に対応する中間コード V C 1 は、図 4 に示すように、加算を示す O P コード A D D と、加算の対象を示す 2 つのオペランド O P R 1 ・ O P R 2 と、加算結果の格納場所を示すオペランド O P R 3 とを含んでいる。本実施形態では、各オペランドは、値自体だけではなく、その値が格納されたアドレスを示すことができ、中間コード V C 1 には、各オペランドがいずれであるかを示すフラグ（図示せず）も含まれている。この例では、第 1 および第 3 のオペランド O P R 1 ・ O P R 3 は、アドレスを示しており、双方の値は、アドレス「L S 0 1 0 0」を示す値に設定されている。また、第 2 のオペランド O P R 2 は、値自体を示しており、その値は、加算対象の値自体「1」に設定されている。

20

【 0 0 4 8 】

一方、H M I 処理部 3 5 は、上記中間コード V C 1 を実行する場合、その O P コードに基づいて、処理が加算処理であることを特定し、オペランド O P R 1 の示す値とオペランド O P R 2 の示す値との加算結果を、オペランド O P R 3 に示すアドレスに格納する。ここで、上記の場合、オペランド O P R 1 は、アドレスを示しており、オペランド O P R 2 が値自体を示しているので、H M I 処理部 3 5 は、オペランド O P R 1 の示すアドレスから読み出した値と、オペランド O P R 2 の示す値とを加算する。

30

【 0 0 4 9 】

なお、上記では、一例として、1 つの文 S T 1 が、1 つの中間コード V C 1 に対応している場合を例にして説明したが、これに限るものではなく、テキストで表現されたスクリプト中の 1 つの文または 1 つの式が、複数の中間コードに対応していてもよい。また、上記では、オペランドがデバイスのアドレスを示している場合を例にして説明したが、デバイスを特定できれば、例えば、上記の変数を示していてもよい。

【 0 0 5 0 】

ここで、図 3 の例では、1 つのスクリプトが、1 つの文から構成されている場合を例示したが、一般には、スクリプトは、文あるいは式の複数からなり、一連の処理毎に別のスクリプトとして格納されている。さらに、本実施形態では、各スクリプトに、トリガを設定することができ、プログラマブル表示器 1 3（H M I 処理部 3 5）は、各トリガが発生しているか否かを監視すると共に、トリガのいずれかが発生した場合、当該トリガに関連付けて記憶されているスクリプトを実行できる。

40

【 0 0 5 1 】

なお、図 3 の例では、トリガとして、例えば、「特定の単位画面に切り換えた」ときが設定されており、プログラマブル表示器 1 3 は、現在表示中の単位画面が、当該特定の単位画面に切り換えられる度に、図 3 に示すスクリプトを実行して、上記アドレス「L S 0

50

「100」に格納されている値を1増加させる。したがって、当該アドレスの内容を参照することによって、上記特定の単位画面が表示された回数を把握できる。

【0052】

このように、本実施形態に係るプログラマブル表示器13は、画面データに含まれるタグによって特定された動作を行うだけではなく、画面データに含まれるスクリプトの示す動作も実行できる。したがって、上記タグでは表現できないような動作も、プログラマブル表示器13に実行させることができ、タグで特定された動作のみを実行する構成と比較して、プログラマブル表示器13の動作を、より柔軟に制御できる。

【0053】

ところで、上記のように、設計者がスクリプトを記述（プログラミング）してプログラマブル表示器13に、その動作を指定する場合、そのスクリプトに不具合（バグ）があったときには、設計者自身がスクリプトをデバッグして、不具合を解消する必要がある。

【0054】

本実施形態では、効率よくデバッグするため、スクリプトにおいて、指定した情報（例えば、特定のデバイスの状態や特定の文字列など）を、予め指定された出力先（例えば、操作画面や特定の記録媒体／送信先の機器など）へ出力するためのデバッグ関数が、新たに定義されており、画面データを生成するエディタ装置としての制御用ホストコンピュータ17（後述）、および、上記プログラマブル表示器13は、デバッグ関数を処理できるように構成されている。これにより、デバッグ関数を実行するプログラマブル表示器13は、上記指定した情報を、操作画面に関係なく表示したり、特定の記録媒体／送信先の機器に記憶したりすることができる。なお、以下では、当該指定した情報をデバッグ情報と称する。

【0055】

さらに、制御用ホストコンピュータ17およびプログラマブル表示器13は、プログラマブル表示器13または制御用ホストコンピュータ17を操作する設計者の指示に応じて、デバッグ関数の有効／無効を、一括して、あるいは、トリガ毎に制御できるように構成されている。

【0056】

具体的には、本実施形態では、図5に示すように、デバッグ関数「__debug（）」が定義されている。この例では、上記デバッグ関数は、上記デバッグ情報を特定するための引数argを持つことができる。本実施形態では、引数argとして、例えば、任意の文字列（改行および復帰以外）と、デバイスを特定するための情報（例えば、アドレスなど）と、改行または復帰とのいずれかを設定することができる。なお、図5の例では、引数argとして、図3に示す文ST1と同じデバイスを特定するための情報（「w:L S 0100」）が設定されている。

【0057】

また、デバッグ関数に対応する中間コードVC11は、図6に示すように、デバッグ関数であることを示すOPコードOP11と、引数argに応じた内容のオペランドOPR11とを含んでいる。なお、上記のように、本実施形態では、引数argは、任意の文字列と、デバイスを特定するための情報と、改行または復帰とのいずれかの種類に大別できるので、中間コードVC11には、オペランドOPR11の種類を識別するフラグ（図示せず）も含まれている。また、オペランドとして文字列を含んでいてもよいが、本実施形態に係る中間コードには、引数argが文字列の場合、オペランドOPR11は、当該文字列が格納されたアドレスを示しており、プログラマブル表示器13は、オペランドOPR11の示すアドレスから文字列を読み出すように構成されている。

【0058】

一方、上記画面データには、当該画面データに含まれるデバッグ関数の有効／無効を一括して設定するためのフラグ情報F1が含まれており、図1に示すように、記憶部36のうち、画面データを記憶する記憶領域には、当該フラグ情報Faを格納するための記憶領域MF1が設けられている。

10

20

30

40

50

【 0 0 5 9 】

さらに、本実施形態に係る画面データには、当該画面データに含まれるデバッグ関数の有効／無効を、それぞれに対応するトリガ毎に設定するために、各トリガに関連して設定されるフラグ情報 F 2 も含まれており、記憶部 3 6 のうち、画面データを記憶する記憶領域には、各トリガに対応するフラグ情報 F 2 を格納するための記憶領域 M F 2 ... も設けられている。

【 0 0 6 0 】

加えて、本実施形態に係るプログラマブル表示器 1 3 は、画面データにて有効と設定されたデバッグ関数であっても、プログラマブル表示器 1 3 への操作に応じて有効／無効を変更できるように構成されており、上記記憶部 3 6 には、有効／無効を示すフラグ情報 F 3 を格納するための記憶領域 M F 3 も設けられている。

10

【 0 0 6 1 】

一方、本実施形態に係るプログラマブル表示器 1 3 (より詳細には、H M I 処理部 3 5) は、記憶部 3 6 の記憶領域 M S に格納されているスクリプトを実行するにあたって、当該記憶領域 M S に格納されている中間コードを順次実行しているが、次に実行すべき中間コードが、デバッグ関数に対応する中間コードである場合、上記フラグ情報 F 1 および F 3 と、現在実行中のスクリプトに対応するフラグ情報 F 2 とが全て有効を示している場合に、上記デバッグ関数の指定する動作を行って、デバッグ情報を上記出力先に出力する。これとは逆に、フラグ情報 F 1 ~ F 3 のいずれかが無効を示している場合、H M I 処理部 3 5 は、このような出力動作を行わず、当該デバッグ関数に対応する中間コードの次に実行すべき中間コードの処理を開始することができる。

20

【 0 0 6 2 】

また、上記 H M I 処理部 3 5 は、タッチパネル 3 4 などの入力装置を介して、予め定められた特定の操作を受け付けた場合、上記記憶領域 M F 3 へアクセスして、フラグ情報 F 2 を操作に応じて変更できる。一例として、本実施形態に係る H M I 処理部 3 5 は、特定の操作として、複数の特定箇所 (例えば、上隅と、その対角の下隅など) が順番に押されたことを検出すると、フラグ情報 F 2 を反転できる。また、他の例として、例えば、設定画面などに、有効／無効の設定を受け付ける領域を表示させると共に、上記特定の操作として、当該領域への操作を受け付けると、有効を示す領域が操作されたか、無効を示す領域が操作されたかによって、フラグ情報 F 3 を変更してもよい。

30

【 0 0 6 3 】

さらに、本実施形態では、各デバッグ関数の出力先を一括して指定できるように構成されている。具体的には、上記画面データには、デバッグ関数の出力先を示す出力先データ O 1 が含まれており、上記記憶部 3 6 のうち、画面データを格納する記憶領域には、当該出力先データ O 1 を格納する記憶領域 M O が設けられている。一方、プログラマブル表示器 1 3 (より詳細には、H M I 処理部 3 5) は、上記デバッグ関数の実行によって、上記デバッグ情報を出力する場合、当該記憶領域 M O に記憶された出力先へ出力できる。なお、プログラマブル表示器 1 3 は、受け付けた操作 (例えば、設定画面における出力先の指定操作など) に応じて、当該記憶領域 M O の内容を更新してもよい。

【 0 0 6 4 】

40

本実施形態では、上記出力先の一例として、操作画面上と、予め定められた記憶媒体上 (例えば、メモリカードなど) と、ネットワーク経由で指定した送信先とを設定可能である。なお、操作画面上に出力する場合、本実施形態では、例えば、H M I 処理部 3 5 が操作画面をディスプレイ 3 3 に描画するために書き込んでいる画面メモリ (ビデオメモリ) 上に、上記デバッグ情報を直接描画するなどして、当該デバッグ情報を操作画面上に重畳して表示している。これにより、例えば、図 7 に示すように、操作画面のレイアウトを変更することなく、デバッグ情報を順次表示できる。

【 0 0 6 5 】

以下では、図 1 および図 8 を参照しながら、上述した画面データを作成するための作画エディタ装置の構成を説明する。すなわち、本実施形態に係る画面データ生成装置は、図

50

１に示すように、制御用ホストコンピュータ１７の一機能ブロックとして実現されており、制御用ホストコンピュータ１７には、プログラマブル表示器１３の画面データを格納する画面データ記憶部７４と、ユーザの指示に応じて、当該画面データを作成し、当該画面データ記憶部７４に格納する作画処理部７５とが設けられている。

【００６６】

上記作画処理部７５は、図８に示すように、画面データに対応する情報が一時的に格納される作業用のメモリ８１と、ユーザの指示を受け付けて、当該メモリ８１の内容を書き換える編集処理部（受付手段）８２と、編集処理部８２によって編集されるメモリ８１の内容を図示しないディスプレイに表示させる表示処理部８３と、画面データ記憶部７４に格納された画面データを当該メモリ８１に展開したり、編集後のメモリ８１の内容に応じた画面データを画面データ記憶部７４へ書き込むデータ入出力処理部（生成手段）８４とを備えている。

10

【００６７】

ここで、上記画面データは、上述したように、画面上の領域と、当該領域への表示や入力に対応するデバイスを特定可能なアドレスとの対応関係を示すタグを組み合わせて構成されており、作画処理部７５の編集処理部８２は、例えば、タグのパレットを表示し、パレット中のタグを選択して画面上に所望のタグを配置するようにユーザへ促す。また、編集処理部８２は、ユーザによるタグの配置操作を受け付けると、当該操作に応じて、編集中の画面データが上記配置されたタグを含むように、メモリ８１の内容を変更し、表示処理部８３は、メモリ８１の内容に基づいて、当該タグを指定された座標に表示する。また、編集処理部８２は、ユーザのドラッグ＆ドロップなどの操作に応じてメモリ８１の内容を変更して、当該タグの表示座標を調整し、表示処理部８３は、変更後の位置にタグを表示する。また、編集処理部８２は、タグに関連するアドレスや変数を入力するようにユーザへ促すと共に、入力されたアドレスや変数が当該タグに関連するように、メモリ８１の内容を変更し、表示処理部８３は、ユーザの指示に応じて、各タグに関連つけられているアドレスや変数を表示する。

20

【００６８】

これらの結果、設計者は、作画処理部７５を操作して、画面上の所望の位置にタグを配置すると共に、各タグとデバイスのアドレスとを対応付けるだけで、画面データを作成できる。したがって、プログラマブル表示器１３の表示プログラムを修正する場合に比べて容易に、各プログラマブル表示器１３の表示や操作を決定（変更）でき、制御システム１のユーザ（制御用ホストコンピュータ１７を操作する設計者）は、ターゲットシステム２の実情や、プログラマブル表示器１３のオペレータの習熟度、あるいは、ユーザの好みに合わせることができる。

30

【００６９】

さらに、本実施形態に係る編集処理部８２は、上記タグの編集操作だけではなく、スクリプトの編集操作も受け付けることができる。具体的には、編集処理部８２は、例えば、図３に示すように、メモリ８１に格納されたスクリプトを、文字列として表示できる。また、編集処理部８２は、当該文字列に対する編集操作を受け付けて、上記メモリ８１に格納されたスクリプトの情報を、操作に応じて更新できる。これにより、設計者は、タグだけではなく、スクリプトも編集することができる。

40

【００７０】

また、上記編集処理部８２は、スクリプト毎にトリガを関連付けるための操作も受け付けることができる。例えば、編集処理部８２は、トリガの関連付けを指示する操作を受け付けた場合、例えば、選択可能なトリガの一覧を表示するなどして、トリガの入力を促すと共に、入力されたトリガを、スクリプトに関連付けて、メモリ８１に格納できる。

【００７１】

さらに、上記編集処理部８２は、上記トリガ毎、および、全スクリプトを対象にして、有効／無効を設定できる。具体的には、編集処理部８２は、全スクリプトに対する有効／無効の設定を指示する操作を受け付けた場合、有効／無効を示す情報（フラグ情報Ｆ１に

50

対応する情報)をメモリ81に格納できる。また、編集処理部82は、トリガに対する有効/無効の設定を指示する操作を受け付けた場合、当該トリガに関連付けて、有効/無効を示す情報(フラグ情報F2に対応する情報)を、メモリ81に格納できる。

【0072】

一方、編集終了指示を受け付けると、データ入出力処理部84は、メモリ81の内容に応じた画面データを生成して、画面データ記憶部74へ書き込む。これにより、編集されたタグとスクリプトの情報と、上記フラグ情報F1およびF2を含む画面データが生成される。なお、サーバ部73は、画面データ記憶部74に格納された画面データを、LAN15を介して、プログラマブル表示器13へ送信し、プログラマブル表示器13の記憶部36に格納させる。

10

【0073】

ここで、上述したように、本実施形態では、スクリプトは、予め中間コードに変換された後で、プログラマブル表示器13に格納されており、データ入出力処理部84は、画面データを生成する際、メモリ81に格納されたスクリプトを中間コードに変換した後で、当該中間コード群を示す情報を含む画面データを生成している。より詳細には、当該データ入出力処理部84は、デバッグ関数も中間コードに変換できるように構成されており、メモリ81に、デバッグ関数を示す文字列を示す情報が含まれている場合、デバッグ関数を示すOPコードと、当該文字列のうち、引数argを示す部分に対応した内容のオペランドとを持った中間コードを生成できる。

【0074】

20

上記構成において、デバッグする際の各部の動作を、図9に基づき説明すると、以下の通りである。すなわち、ステップ1(以下では、S1のように略称する)において、作画処理部75(より詳細には、編集処理部82)は、設計者の操作に応じて、操作画面を編集する。同様に、S2において、作画処理部75は、設計者の操作に応じて、スクリプトを編集する。

【0075】

上記S1およびS2において、操作画面およびスクリプトの編集が終了し、画面データの作成指示を受け付けると、S3において、作画処理部75は、編集された操作画面の情報(タグ)およびスクリプトの情報(スクリプトの中間コード)を含む画面データを生成し、画面データ記憶部74に書き込む。

30

【0076】

さらに、制御用ホストコンピュータ17(より詳細には、サーバ部73)は、S4において、設計者の指示に応じて、上記画面データをプログラマブル表示器13にダウンロードして、プログラマブル表示器13の記憶部36に格納させる。

【0077】

一方、プログラマブル表示器13(より詳細には、HMI処理部35)は、S11~S21において、画面データのタグおよびスクリプトの示す動作を行っている。具体的には、プログラマブル表示器13は、S11において、画面データのスクリプトのトリガが発生しているか否かを監視しており、トリガが発生すると(上記S11にて、YESの場合)、S12~S16において、発生したトリガに対応するスクリプトを実行する。

40

【0078】

より詳細には、プログラマブル表示器13は、S12において、次に実行すべき中間コードがデバッグ関数に対応する中間コードか否かを判断することによって、スクリプトのうち、次に処理しようとしている部分が、デバッグ関数であるか否かを判定する。

【0079】

デバッグ関数であった場合(上記S12にて、YESの場合)、プログラマブル表示器13は、S13において、上記フラグ情報F1~F3の全てが有効であるか否かを判定し、有効であった場合(上記S13にて、YESの場合)にのみ、デバッグ関数の示す出力動作を行う(S14)。

【0080】

50

一方、デバッグ関数ではなかった場合（上記 S 1 2 にて、NO の場合）、プログラマブル表示器 1 3 は、S 1 5 において、当該中間コードを実行することによって、上記部分に対応する処理を行う。

【 0 0 8 1 】

なお、上記 S 1 2 ~ S 1 5 の処理は、スクリプトが終了するまで（S 1 6 にて、YES になるまで）繰り返される。また、プログラマブル表示器 1 3 は、S 2 1 において、画面データのタグの示す情報に基づき、上述したように、デバイスの状態を表示したり、デバイスの状態への操作を受け付けたりしている。

【 0 0 8 2 】

ここで、上記構成では、フラグ情報 F 1 ~ F 3 のいずれかが無効に設定されていれば、スクリプト中にデバッグ関数が記述されていたとしても、デバッグ関数の示す出力動作を行わない。

【 0 0 8 3 】

したがって、図 1 1 に示すように、デバッグ情報を出力するための関数や文などをスクリプト中に記載しておき、デバッグ情報の出力を停止するためには、スクリプトから、これらの関数や文全てを削除する場合と異なって、設計者は、フラグ情報 F 1 ~ F 3 のいずれかを無効にするだけで、デバッグ関数の示す出力動作を停止させることができる。この結果、デバッグ情報の出力動作を停止する際に、設計者がスクリプトを変更する必要のある箇所を大幅に削減でき、デバッグ効率を向上できる。

【 0 0 8 4 】

また、本実施形態に係るプログラマブル表示器 1 3 は、プログラマブル表示器 1 3 への操作に応じて、フラグ情報 F 3 を変更できる。したがって、設計者は、デバッグ中に、プログラマブル表示器 1 3 を操作するだけで、画面データ自体を変更せずに、上記出力動作を停止したり、再開させることができる。この結果、デバッグ時において、画面データの変更回数も削減でき、デバッグ効率をさらに向上できる。

【 0 0 8 5 】

さらに、本実施形態に係るプログラマブル表示器 1 3 は、トリガに対応するフラグ情報 F 2 が無効である場合、例えば、図 1 0 に示すように、当該トリガに関連するスクリプト中のデバッグ関数によるデバッグ情報の出力を停止する。したがって、フラグ情報 F 1 のみによって決定する場合よりも柔軟に、しかも、フラグ情報 F 2 のみによって決定する場合よりも簡単に、デバッグ情報を出力するか否かを設定でき、デバッグ効率を向上できる。なお、図 1 0 では、図 7 にて表示されていた各デバッグ情報のうち、あるトリガに対応するスクリプト中のデバッグ関数によるデバッグ情報（図 7 中、第 3 行、第 4 行および第 1 1 行のデバッグ情報）が表示されていない。

【 0 0 8 6 】

加えて、本実施形態に係るプログラマブル表示器 1 3 は、画面データに含まれるフラグ情報 F 1 および F 2 のいずれかが無効である場合、デバッグ関数によるデバッグ情報の出力を停止する。したがって、設計者は、作画処理部 7 5 へ指示して、フラグ情報 F 1 および F 2 のいずれかが無効に設定された画面データを生成させることによって、プログラマブル表示器 1 3 におけるデバッグ情報の出力を禁止できる。これにより、例えば、デバッグが終了した後、プログラマブル表示器 1 3 のオペレータが操作画面によってデバイス 2 a の状態を確認したり、操作画面を操作している時点のように、不所望な時点で、プログラマブル表示器 1 3 にデバッグ情報が表示表示されてしまうという不具合を確実に防止でき、プログラマブル表示器 1 3 の安全性をさらに向上できる。

【産業上の利用可能性】

【 0 0 8 7 】

本発明によれば、制御用表示装置のデバッグ効率を向上できるので、デバイスの状態を表示する制御用表示装置、あるいは、そのためのフラグ情報を生成するエディタ装置として、広く適用できる。

【図面の簡単な説明】

【 0 0 8 8 】

【図 1】本発明の実施形態を示すものであり、制御システムの要部構成を示すブロック図である。

【図 2】上記制御システムに設けられたプログラマブル表示器の画面例を示す図面である。

【図 3】上記プログラマブル表示器にて実行されるスクリプトを示す図面である。

【図 4】上記スクリプトを示す中間コード群を構成する中間コードを示す図面である。

【図 5】上記スクリプトに挿入されるデバッグ関数を示す図面である。

【図 6】上記デバッグ関数に対応する中間コードを示す図面である。

【図 7】上記プログラマブル表示器の画面を示すものであり、デバッグ関数によってデバッグ情報が表示されている状態を示す図面である。 10

【図 8】上記制御システムに設けられた作画処理部の要部構成を示すブロック図である。

【図 9】上記制御システムの各部の動作を示すフローチャートである。

【図 10】上記プログラマブル表示器の画面を示すものであり、上記デバッグ情報のうち、特定のトリガに対応するスクリプト内のデバッグ関数によるデバッグ情報のみが表示されていない状態を示す図面である。

【図 11】従来技術を示すものであり、プログラマブル表示器で実行すべきスクリプトに、デバッグに必要な情報を表示するための命令を挿入した状態を示す図面である。

【符号の説明】

【 0 0 8 9 】

2 a デバイス

1 3 プログラマブル表示器（制御用表示装置）

1 7 制御用ホストコンピュータ（外部機器）

3 2 L A N ・ I F 部（書き込み手段）

3 4 タッチパネル（入力手段）

3 5 H M I 処理部（制御手段）

3 6 記憶部（記憶手段）

7 5 作画処理部（エディタ装置）

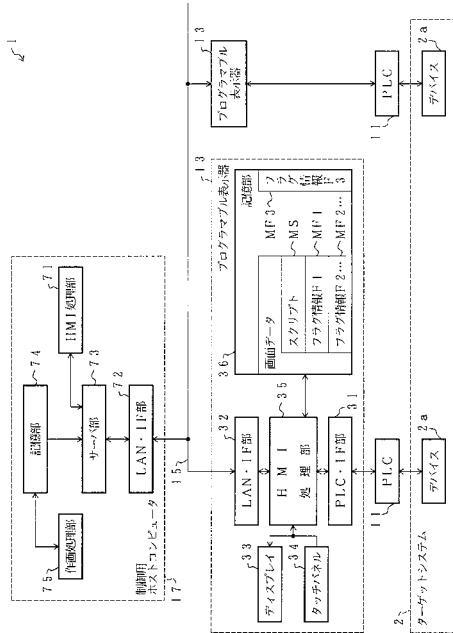
8 2 編集処理部（受付手段）

8 4 データ入出力処理部（出力手段）

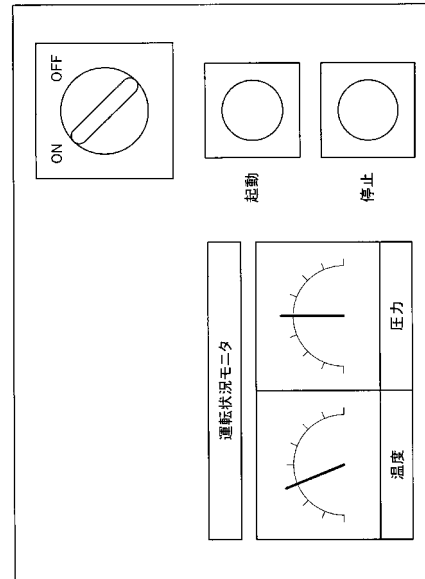
20

30

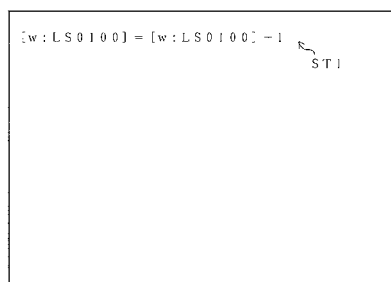
【 図 1 】



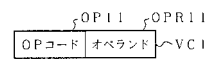
【 図 2 】



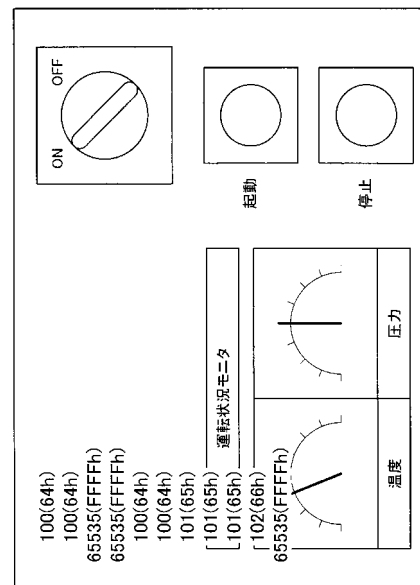
【 図 3 】



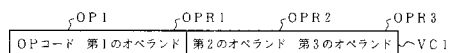
【 図 6 】



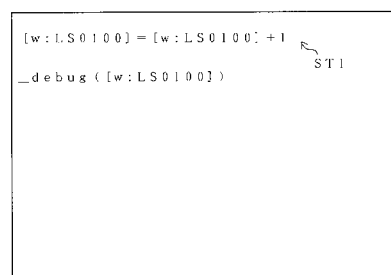
【圖 7】



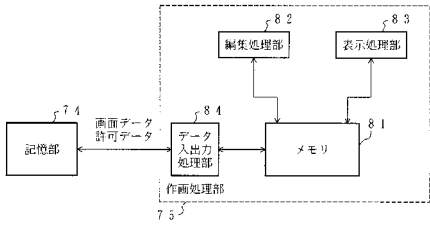
【 図 4 】



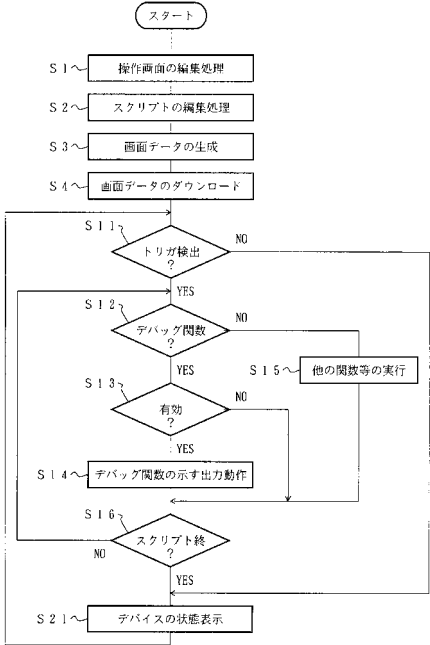
【 図 5 】



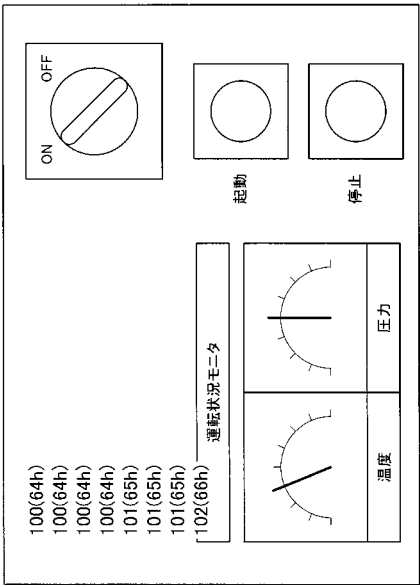
【図 8】



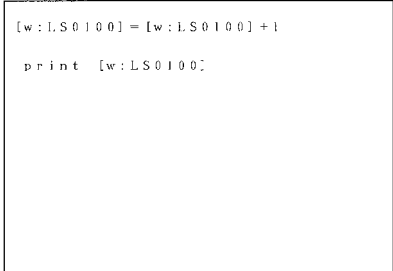
【図 9】



【図 10】



【図 11】



フロントページの続き

(56)参考文献 特開平 1 1 - 1 1 0 2 5 6 (J P , A)
特開平 0 6 - 1 4 9 6 1 6 (J P , A)
特開平 0 9 - 2 0 4 2 0 8 (J P , A)
特開 2 0 0 1 - 2 8 2 3 2 4 (J P , A)

(58)調査した分野(Int.Cl. , D B 名)
G 0 5 B 1 9 / 0 5
G 0 6 F 1 1 / 2 8