

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第5384306号
(P5384306)

(45) 発行日 平成26年1月8日(2014.1.8)

(24) 登録日 平成25年10月11日(2013.10.11)

(51) Int. Cl.

F I

G06F	11/28	(2006.01)	G06F	11/28	315A
G05B	19/05	(2006.01)	G06F	11/28	340C
			G05B	19/05	A

請求項の数 6 (全 22 頁)

(21) 出願番号 特願2009-269085 (P2009-269085)
 (22) 出願日 平成21年11月26日(2009.11.26)
 (65) 公開番号 特開2011-113304 (P2011-113304A)
 (43) 公開日 平成23年6月9日(2011.6.9)
 審査請求日 平成24年5月14日(2012.5.14)

(73) 特許権者 591016253
 発紘電機株式会社
 石川県白山市上柏野町238番地
 (74) 代理人 100074099
 弁理士 大菅 義之
 (72) 発明者 前川 勉
 石川県白山市上柏野町890番地1 発紘
 電機株式会社内
 審査官 多胡 滋

最終頁に続く

(54) 【発明の名称】 表示／制御・シミュレートシステム、そのプログラマブル表示器、シミュレータ装置

(57) 【特許請求の範囲】

【請求項1】

プログラマブル表示器とシミュレータ装置とが接続されたシステムであって、
 前記プログラマブル表示器は、
 各種マクロブロックより成るマクロデータが付随された画面データを記憶する画面データ記憶手段と、

該画面データを用いて任意のイベントに応じた処理を実行し、該処理に係わり前記マクロブロックのプログラムを実行する際、1行毎にその行番号を前記シミュレータ装置に通知し、実行許可が返信されてくると当該行のプログラムを実行する処理実行手段とを有し、

前記シミュレータ装置は、
 前記画面データ記憶手段に記憶されているマクロデータと同じマクロデータを記憶する記憶手段と、

該記憶手段に記憶されているマクロデータのマクロブロックのプログラムの任意の行をブレークポイントに設定させて、該設定内容をブレーク位置情報として記憶するブレークポイント設定支援手段と、

前記処理実行手段から前記行番号の通知がある毎に、前記ブレーク位置情報を参照して、該行番号の行が前記ブレークポイントであるか否かを判定し、ブレークポイントではない場合には前記実行許可を返信し、ブレークポイントである場合には前記実行許可を返信せず実行再開待ち状態とすると共にその後所定の指示があった場合に前記実行許可を

返信するマクロ実行制御手段と、

を有することを特徴とする表示 / 制御・シミュレートシステム。

【請求項 2】

前記シミュレータ装置は、前記実行再開待ち状態におけるシミュレート結果を表示するシミュレート結果表示手段を更に有することを特徴とする請求項 1 記載の表示 / 制御・シミュレートシステム。

【請求項 3】

前記マクロ実行制御手段は、前記所定の指示としてステップ実行があった場合には、前記通知された行番号の次の行をブレークポイントに設定したうえで、前記実行許可を返信することを特徴とする請求項 1 または 2 記載の表示 / 制御・シミュレートシステム。

10

【請求項 4】

前記プログラマブル表示器は、プログラマブル表示器の実機、または該実機の動作をエミュレートするエミュレート装置であることを特徴とする請求項 1 ~ 3 の何れかに記載の表示 / 制御・シミュレートシステム。

【請求項 5】

プログラマブル表示器とシミュレータ装置とが接続されたシステムにおける該プログラマブル表示器であって、

各種マクロブロックより成るマクロデータが付随された画面データを記憶する画面データ記憶手段と、

該画面データを用いて任意のイベントに応じた処理を実行し、該処理に係わり前記マクロブロックのプログラムを実行する際、1 行毎にその行番号を前記シミュレータ装置に通知し、実行許可が返信されてくると当該行のプログラムを実行する処理実行手段と、

20

を有することを特徴とするプログラマブル表示器。

【請求項 6】

プログラマブル表示器とシミュレータ装置とが接続されたシステムにおける該シミュレータ装置であって、

前記プログラマブル表示器側に記憶されているマクロデータと同じマクロデータを記憶する記憶手段と、

該記憶手段に記憶されているマクロデータのマクロブロックのプログラムの任意の行をブレークポイントに設定させて、該設定内容をブレーク位置情報として記憶するブレークポイント設定支援手段と、

30

任意のマクロブロックを実行中の前記プログラマブル表示器からこれから実行する行の行番号の通知がある毎に、前記ブレーク位置情報を参照して、該行番号の行が前記ブレークポイントであるか否かを判定し、ブレークポイントではない場合には実行許可をプログラマブル表示器へ返信し、ブレークポイントである場合には前記実行許可を返信せずに行再開待ち状態とすると共にその後所定の指示があった場合に前記実行許可を返信するマクロ実行制御手段と、

を有することを特徴とするシミュレータ装置。

【発明の詳細な説明】

40

【技術分野】

【0001】

本発明は、プログラマブル表示器に関するシミュレータ装置を有するシステムに関する。

【背景技術】

【0002】

従来より、例えば、PLC (プログラマブル・ロジック・コントローラ) 本体等である外部機器と、プログラマブル表示器とが接続された表示 / 制御システム構成が知られている。このようなシステムでは、一般的に、プログラマブル表示器は、外部機器のメモリからデータを読み出して表示したり、外部機器のメモリにデータを書き込むことで外部機器に

50

対して任意の指令を与えたり、自己の画面の表示内容の制御やスクリーン切換え等を行う。

【 0 0 0 3 】

上記プログラマブル表示器の処理は、予め記憶される基本プログラムが、ダウンロードされた画面データを用いて行う。画面データには、各種マクロデータ（マクロブロック）が含まれている。マクロデータ（マクロブロック）は、例えば、画面上に配置される各アイテム（スイッチ、メータ、ボタン等）に対応付けられて、このアイテムに応じた処理を実行するプログラム等である。あるいは、アイテムに対応付けられておらず、例えば定期的に実行されるプログラム等である。

【 0 0 0 4 】

上記表示 / 制御システムにおいて、更に、作画エディタを有するパソコン等（作画エディタ装置という）が、プログラマブル表示器に接続された構成となっている場合がある。上記画面データは、この作画エディタ装置上でユーザが任意に作成して、プログラマブル表示器にダウンロードする。

【 0 0 0 5 】

プログラマブル表示器では、上記マクロデータにより、独自の演算処理を行う機能としてのマクロ機能を有することになる。このマクロ機能によって、演算処理以外にもプログラマブル表示器の動作や外部機器のメモリ操作を任意のタイミングで行うことができる。このマクロ機能は、上記画面データ（マクロデータ）を変更することで、容易に変更（カスタマイズ）できる。このように容易にカスタマイズできる点から、プログラマブル表示器に複雑な動作を行わせたい場合に、マクロデータが多用される傾向にある。

【 0 0 0 6 】

ここで、画面データ（各種マクロデータ含む）は、上記の通りユーザが作成するものであり、当然何らかのミスが生じる可能性があり、特にマクロデータに関してミスや勘違い等によりユーザが望んだ動作とならない場合が有り得る。よって、新たな画面データを作成してプログラマブル表示器にダウンロードした後、実際に運用を開始する前に、特にマクロデータ（プログラム）に関する動作確認を行うことが望ましい。

【 0 0 0 7 】

この為、従来より、シミュレータ装置により画面データの動作確認を行うことが提案されている。このシミュレータ装置は、上記 P L C 等である外部機器の機能・動作を模擬するものである。このシミュレータ装置は、上記作画エディタ装置に、当該作画エディタ装置の機能の 1 つとして搭載されている場合が少なくないが、この例に限るものではない。

【 0 0 0 8 】

何れにしても、シミュレータ装置は、少なくとも上記動作確認の際にはプログラマブル表示器に接続されて、プログラマブル表示器と通信を行いながら、擬似的な外部機器として動作する。また、シミュレーション結果の表示等を行う。

【 0 0 0 9 】

すなわち、従来の上記シミュレータ装置は、例えば以下に記す機能を有する。

まず、外部機器の機能・動作をシミュレートする機能を有する。この機能は、例えばプログラマブル表示器からのメモリの読み込み / 書き込み要求に応じて、シミュレータ装置上に擬似的に構築している外部機器メモリ（以下、擬似メモリと表記）について応答 / 書き換えを行なうことで、外部機器をシミュレートしている。

【 0 0 1 0 】

また、シミュレート中に擬似メモリの内容を一覧で表示し、シミュレータ装置上の G U I にてユーザが任意に指定した擬似メモリ・アドレスの読み込み / 書き込みを行う機能を有する。更に、ユーザが任意に指定した擬似メモリ・アドレスに関して任意に指定した単位 / 範囲 / 時間でインクリメント / デクリメントする機能等も有している。

【 0 0 1 1 】

ここで、たとえば、特許文献 1、特許文献 2 には、プログラマブル・ロジック・コントローラ（以下 P L C と表記）内の動作をシミュレートする方法が提示されている。

10

20

30

40

50

上記のシミュレータ装置によると、P L CのシーケンスをシミュレートすることによりP L Cのデバッグ効率を向上させることができる。

【 0 0 1 2 】

また、特許文献3には、プログラマブル表示器とシミュレータ装置間の汎用的なI / Fと、シミュレータ装置内部に持つ擬似メモリの制御方法について開示されている。

上記のシミュレータ装置によると、プログラマブル表示器とのI / F部分をシミュレートすることで、実際の外部機器を接続していなくても、プログラマブル表示器の動作を確認することができる。

【 0 0 1 3 】

また、特許文献4には、グラフィック操作パネル（表示器）が、制御用ホストコンピュータとP L Cとに接続された構成で、画面データ実行時に、制御用ホストコンピュータ上でP L Cの模擬動作を行うこと等が開示されている。更に、グラフィック操作パネル（表示器）に接続することなく、スタンドアロンの状態にある制御用ホストコンピュータ上で、画面データのテスト表示・デバッグを行えることも開示されている。

10

【 0 0 1 4 】

また、特許文献5には、パソコン上にて、表示器のシミュレータ装置と、P L Cのシミュレータ装置とを用いて、P L Cの表示器のハードウェアを用いずに、表示器の表示画面データのシミュレーションを行えることが開示されている。

【 0 0 1 5 】

この様に、従来のシミュレータ装置としては、P L Cの模擬動作のみを行って表示器に関しては実機に接続するタイプと、P L C、表示器の両方の模擬動作を行うことによりスタンドアロンでデバッグ実現可能なタイプとが存在する。尚、表示器の動作の模擬機能に関しては、表示器をエミュレートするエミュレート装置という場合もある。本説明では、この様なエミュレート装置もシミュレータ装置に含まれるものとする。

20

【 0 0 1 6 】

また、特許文献6には、シミュレータ装置に係わり、シミュレーション用画面の表示例等が開示されている。

【 先行技術文献 】

【 特許文献 】

【 0 0 1 7 】

【 特許文献1 】 特開平4 - 2 1 5 1 0 5号公報

【 特許文献2 】 特開平5 - 2 9 7 9 1 5号公報

【 特許文献3 】 特許第3 5 6 2 7 8 3号公報

【 特許文献4 】 特開2 0 0 1 - 7 5 6 3 4号公報

【 特許文献5 】 特開2 0 0 6 - 1 5 5 1 5 8号公報

【 特許文献6 】 特開平1 1 - 1 3 4 2 1 8号公報

【 発明の概要 】

【 発明が解決しようとする課題 】

【 0 0 1 8 】

しかしながら、上記特許文献1～3に記載の方法においては、主な目的がP L Cのデバッグ効率の向上であり、マクロブロックのデバッグ効率の向上に関しては不十分というのが実情であった。

40

【 0 0 1 9 】

上述した通り、上記画面データ（各種マクロブロックが付随されている）を変更することで、上記マクロ機能を容易に変更（カスタマイズ）でき、便利である。

しかしながら、一方で、従来のプログラマブル表示器用の外部機器シミュレータ装置では、カスタマイズしたマクロデータの設定に何らかの問題がある為にユーザが望んだ通りの動作とならない場合に、各種マクロデータのうち問題があるマクロデータを特定することは容易であるが、当該マクロブロック内において問題がある箇所（行）を特定する為の作業が、非常に手間が掛かるという問題があった。つまり、マクロブロックのデバッグ効

50

率が悪いという問題があった。

【0020】

これについて、以下、より詳しく説明する。

まず、任意のマクロデータ（マクロブロックともいう）は、例えば「mov」、「Fill」等のマクロコマンドを用いた複数行のプログラムリスト（例えば後に図5等に一例を示す）となっている。各行は、例えば、上記マクロコマンドと、このマクロコマンドの実行対象（例えばリード/ライトするメモリアドレス等）から成る。

【0021】

このマクロブロックの内容は、ユーザが任意に作成するものであるため、当然、ミスが生じる可能性はある。また、ミスがなくても、ユーザが望んだ通りの動作とならない場合も有り得る。

【0022】

何れにしても、何らかの問題が生じた場合でも、マクロブロック全体に問題があるわけではなく、その一部に問題がある場合が殆どである。よって、複数行から成るマクロブロックの何行目に問題があるのかを特定することで、問題がある行を修復する必要がある。

【0023】

ここで、上記何らかの問題があるか否かは、ユーザが判断するものである。これは、上記のようにシミュレータ装置の機能としてシミュレート中の擬似メモリとその値を一覧で表示する機能がある。ユーザは、この表示されるメモリ値を見ることで、問題があるか否かを判断することができる。

【0024】

しかしながら、このメモリ値は、マクロブロック単位で処理実行した結果しか表示することができない。例えば、仮に、100行から成るマクロブロックを例にするならば、100行全てを実行した結果としてのメモリ値が表示されることになる。

【0025】

この為、このメモリ値が期待した値ではない等の判断理由で、このマクロブロックに問題があると分かった場合でも、このマクロブロック内のどの行に問題があるのか分からなかった。これより、従来では、例えば以下に説明する作業を行って問題の行を特定していた。

【0026】

すなわち、上記のように処理実行はマクロブロック単位で行われ、途中で処理を止めることができないので、問題となるマクロブロックの内容を修正して処理実行して結果（メモリ値）を確認する作業を繰り返し行っていた。例えば、上記100行から成るマクロブロックの例の場合、例えばまず上記作画エディタ装置上でユーザが、1行目～10行目までを残して残り（11行～100行）は削除したマクロブロックを作成する。そして、作成した（修正した）マクロブロックをプログラマブル表示器にダウンロードする。その後、シミュレーションを行う。

【0027】

この例の場合、上記修正したマクロブロックを実行すると、1行目～10行目のみが実行されることになり、その結果がメモリ値として反映されることになり、ユーザは表示されるメモリ値を参照することで、1行目～10行目に問題があるか否かを判断できる。

【0028】

そして、1行目～10行目に問題がなければ、今度は1行目～20行目のみから成るマクロブロックを作成して、上記1行目～10行目のマクロブロックと同様の処理を行う。これを、100行目まで繰り返し行うことになる。

【0029】

上記の例では、仮に、1行目～20行目のマクロブロックのときに問題有りだと判断したならば、11行目～20行目の何れかの行に問題があることになる。よって、今度は、1行単位でマクロブロックを作成して検証していく。最初は、1行目～11行目までのマクロブロック、次は1行目～12行目のマクロブロックというように、順次、“修正したマ

10

20

30

40

50

クマクロブロック”を作成して検証していく。

【0030】

上記マクロブロックの修正方法は、一例であり、この例に限るものではない。例えば、最初に1行目～50行目のマクロブロックを作成して検証すれば、100行のマクロブロックの前半と後半のどちらに問題があるのかが分かり、そこから更に対象を狭めていく等という方法も考えられる。あるいは、最初から1行単位で検証していく方法も考えられる。つまり、最初は1行目のみ、次は1～2行目、次は1～3行目というように、1行ずつ増やしていくようにしてもよい。

【0031】

あるいは、上記のような規則的な方法ではなく、デバッグ作業者が、経験や勘等により、例えば怪しいと思う行を推測して検証する等という、試行錯誤を繰り返して行うことも考えられる。しかし、デバッグ作業者の経験や勘に頼る部分が多く、また作業効率が悪いものであった。あるいは、マクロブロック中にパッチコマンドを挿入する方法等もあるが、これも作業効率が悪いものであった。

10

【0032】

何れにしても、従来では、問題があるマクロブロックに関して当該マクロブロック内の問題箇所を特定する為に、試行錯誤を繰り返し、逐一マクロブロックを修正してこれを表示器にダウンロードしてシミュレーションを掛けるという、非常に手間が掛かる作業が必要であった。この為、作業効率が悪いものであり、デバッグ作業に非常に時間が掛かっていた。

20

【0033】

尚、この問題は、上記特許文献4, 5のようなエミュレート装置も含むシミュレータ装置の場合でも、同様に生じるものである。

本発明の課題は、プログラマブル表示器用のシミュレータ装置において、任意のマクロブロック実行中に任意のタイミングで実行を一旦停止する機能を備えさせたことで、手間が掛からず短時間でマクロブロック内の問題箇所を特定することができ、以ってマクロブロックのデバッグ効率の向上を図ることができる表示/制御・シミュレートシステム、そのプログラマブル表示器、シミュレータ装置等を提供することである。

【課題を解決するための手段】

【0034】

30

本発明の表示/制御・シミュレートシステムは、プログラマブル表示器とシミュレータ装置とが接続されたシステムであって、前記プログラマブル表示器は、各種マクロブロックより成るマクロデータが付随された画面データを記憶する画面データ記憶手段と、該画面データを用いて任意のイベントに応じた処理を実行し、該処理に係わり前記マクロブロックのプログラムを実行する際、1行毎にその行番号を前記シミュレータ装置に通知し、実行許可が返信されてくると当該行のプログラムを実行する処理実行手段とを有し、前記シミュレータ装置は、前記画面データ記憶手段に記憶されているマクロデータと同じマクロデータを記憶する記憶手段と、該記憶手段に記憶されているマクロデータのマクロブロックのプログラムの任意の行をブレークポイントに設定させて、該設定内容をブレーク位置情報として記憶するブレークポイント設定支援手段と、前記処理実行手段から前記行番号の通知がある毎に、前記ブレーク位置情報を参照して、該行番号の行が前記ブレークポイントであるか否かを判定し、ブレークポイントではない場合には前記実行許可を返信し、ブレークポイントである場合には前記実行許可を返信せずに実行再開待ち状態とすると共にその後所定の指示があった場合に前記実行許可を返信するマクロ実行制御手段とを有する。

40

【0035】

また、上記構成の表示/制御・シミュレートシステムにおいて、前記シミュレータ装置は、前記実行再開待ち状態におけるシミュレート結果を表示するシミュレート結果表示手段を更に有する。

【0036】

50

上記構成の表示 / 制御・シミュレートシステムによれば、マクロブロック実行中に、ユーザが任意に設定したブレークポイントの行で一時停止させることができる。これによって、マクロブロック実行の途中までのシミュレート結果を表示することができ、ユーザは、この表示を見ることで一時停止時点までの処理に問題があったか否かを判断できる。

【 0 0 3 7 】

従来では、マクロブロックを最後まで実行した時点でシミュレート結果を表示するので、途中までのシミュレート結果を表示したい場合には上述したように、マクロブロックを修正（任意の途中の行まで記述されたもの）し、修正する毎に修正したマクロブロックを表示器に転送・格納させて実行させるといふ、非常に手間・時間が掛かる作業が必要であった。本手法では特に修正・転送が必要なくなることから、手間が掛からず短時間でマクロブロック内の問題箇所を特定することができる。

10

【発明の効果】

【 0 0 3 8 】

本発明の表示 / 制御・シミュレートシステム、そのプログラマブル表示器、シミュレータ装置等によれば、任意のマクロブロック実行中に任意のタイミングで実行を一旦停止する機能を備えさせたことで、手間が掛からず短時間でマクロブロック内の問題箇所を特定することができ、以ってマクロブロックのデバッグ効率の向上を図ることができる。

【図面の簡単な説明】

【 0 0 3 9 】

【図 1】本例の表示 / 制御システムの概略構成図である。

20

【図 2】プログラマブル表示器のハードウェア構成図である。

【図 3】図 1 に示す表示 / 制御システムのソフトウェア構成図である。

【図 4】本例の表示 / 制御システムの処理フローチャート図である。

【図 5】マクロデータの一例を示す図である。

【図 6】マクロビューアによるブレークポイント設定画面の一例である。

【発明を実施するための形態】

【 0 0 4 0 】

以下、図面を参照して、本発明の実施の形態について説明する。

図 1 に、本例の表示 / 制御システム（プログラマブル表示器を含む P L C システム等）の概略構成図を示す。

30

【 0 0 4 1 】

図 1 において、表示 / 制御システムは、プログラマブル表示器 1 と、プログラマブル・ロジック・コントローラ本体（以下、P L C 本体と表記）等である外部機器 4 と、作画エディタ装置 5 を有する。

【 0 0 4 2 】

P L C 本体（外部機器 4）には、従来と同様に、プログラマブル表示器 1 と、各種制御対象機器（モータ、温調器等）が接続されている。P L C 本体は、一般的なものであってよく、特に詳細には説明しないが、よく知られているように、汎用的なワークメモリや、各制御情報等をデバイス名とアドレス等で識別するメモリ（制御用メモリと呼ぶものとする）と、それらを制御する機能（マイコン）等によって実現されるのが一般的である。

40

【 0 0 4 3 】

また、本例では、シミュレータ装置の機能が作画エディタ装置 5 に搭載されている例を用いるものとする。

プログラマブル表示器 1 には、従来で説明したように、作画エディタ装置 5 上で任意に作成された画面データ（各種マクロブロックが付随されている）が、作画エディタ装置 5 からダウンロードされて、これを表示器 1 内の記憶部（メモリ等）に格納する。そして、表示器 1 内の基本プログラムが、この画面データを用いて各種処理を実行する。その際、例えば後述するマクロデータ 2 6（マクロブロック）を実行することで、上記制御用メモリの特定のアドレスのデータを読み出して表示したり、特定のアドレスにデータを書き込む。P L C 本体は、書き込まれたデータに基づいて、このアドレスに対応する制御対象機

50

器（またはPLC本体自身）を制御する。

【0044】

尚、上記外部機器4は、PLC本体だけでなく制御対象機器も含まれてもよい。そして、プログラブル表示器1は、マクロブロックを実行することで、この様な外部機器4の状態を表示したり、外部機器4を制御するものである。

【0045】

また、作画エディタ装置5は、上記プログラブル表示器1用の画面データを、ユーザに任意に作成させる作画エディタ機能を有する、例えばパソコン等である。また、本例では、作画エディタ装置5は、シミュレータ装置としての機能も有する。このシミュレータ装置としての機能は、上述した従来の機能も有するが、更に後述する新たな機能も備える。尚、この例に限らず、作画エディタ装置5とは別にシミュレータ装置を設けるようにしてもよい。

10

【0046】

尚、従来のシミュレータ装置の機能は、既に説明してあり、ここでは特に詳細には説明しないが、簡単に説明するならば、例えば、作画エディタ装置5内のメモリを、擬似的な上記制御用メモリとし、プログラブル表示器1に上記制御用メモリではなくこの擬似メモリにアクセスさせる。そして、この擬似メモリの内容を読み出して表示する等の処理を行うものである。

【0047】

プログラブル表示器1は、複数の通信インタフェース2を有し、各通信インタフェース2には各通信ライン3が接続されている。プログラブル表示器1は、各通信ライン3を介して、外部機器4、作画エディタ装置5と通信可能となっている。例えば、プログラブル表示器1は、通信インタフェース2と通信ライン3を介して外部機器4と接続することにより、例えば上記制御用メモリにアクセスしてデータのリード/ライトを行うことで、外部機器4の制御や外部機器4の状態を表示することが出来る。

20

【0048】

この様な外部機器4の制御の為の操作（スイッチ操作等）をユーザに行わせたり状態表示を行う為の操作/表示用画面が、プログラブル表示器1のディスプレイに表示される。この画面表示は、画面データに基づいて行われる。この操作/表示用画面上には各種アイテム画像（スイッチ、ランプ、グラフ、数値等）が表示される。ユーザは、所望のアイテム画像の表示位置をタッチする等して、このアイテム画像のアイテムに予め割り当てられている機能（マクロブロック）を実行させることで、外部機器4の制御等が行われる。

30

【0049】

あるいは、グラフ、数値等に関しては、ユーザが操作しなくても、例えば定周期で実行されるマクロブロックによって、上記制御用メモリの所定アドレスのデータが読み出されることで、外部機器4の状態をグラフ表示/数値表示等することになる。

【0050】

尚、画面データは、どのような表示を行うか、どのような制御を行うか、どのような接続機種かといったプログラブル表示器の動作を規定するデータである。

例えば、作画エディタ装置5における画面データの作成においては、表示器1においてユーザによるオペレーションを受け付ける為のスイッチや、データの表示を行う数値表示などの各種部品のシンボル画を含む部品リストが、不図示の画面データ作成支援画面上に表示される。ユーザは、この中から所望の部品を選択して、支援画面上の任意位置に配置する。更に、ユーザは、配置した部品に対して、任意の制御対象機器のデバイス名とアドレス（上記制御用メモリにおけるアドレス）等を対応付ける。また、各部品には、予め所定のマクロブロックが割り当てられている。あるいは、ユーザが、各部品に、任意のマクロブロックを割り当てる。また、マクロブロックの内容は、ユーザが任意に編集可能であり、更に例えば従来の課題で説明したように100行のマクロブロックを10行とするような修正を行うこともできる。

40

【0051】

50

この様にして作成された画面データを、表示器 1 にダウンロードする。そして、後述するように表示器 1 における基本プログラムである本体プログラム 2 1 が、この画面データを用いた表示 / 制御処理を行う。これによって、例えば、グラフ、数値等の部品に関しては、自動的に、外部機器 4 の状態をグラフ表示 / 数値表示等することになる。また、スイッチ等の部品に関しては、ユーザが操作した部品に応じた制御処理が実行されることになる。何れにしても、上記各部品に対応付けられたアドレス等に基づいて、上記制御用メモリにおける所定アドレスにアクセスして、そのデータをリード / ライトすることになる。

【 0 0 5 2 】

この様な表示 / 制御処理は、上記マクロブロック実行によって実現されることになる。

そして、この様な表示器 1 の処理を、実機 (P L C 本体) に実際にアクセスすることなく擬似的に実行させる為に、シミュレータ装置が設けられている。

10

【 0 0 5 3 】

本手法は、シミュレータ装置に係わるものであり、シミュレーション動作中は外部機器 4 と接続 (アクセス) することはないので、外部機器 4 に関してはこれ以上説明しない。

本例のシミュレータ装置に係わるシミュレーション動作については、後に図 3、図 4 等を参照して説明する。

【 0 0 5 4 】

図 2 は、プログラマブル表示器 1 のハードウェア構成図である。尚、ハードウェア構成自体は、従来と同様であってよい。また、作画エディタ装置 5 は、ハードウェア的には一般的な汎用パソコンを用いればよく、ここでは図示・説明は行わない。

20

【 0 0 5 5 】

図 2 において、プログラマブル表示器 1 は、制御ユニット 1 0 と、上記通信インタフェース 2 と、タッチパネル 1 8 及びディスプレイ 1 9 を有する。

制御ユニット 1 0 は、グラフィックコントローラ 1 1、ROM 1 2、RAM 1 3、通信コントローラ 1 4、CPU 1 5、タッチパネルコントローラ 1 6 等から成り、これらが内部バス 1 7 に接続している。

【 0 0 5 6 】

グラフィックコントローラ 1 1 は、ディスプレイ 1 9 に接続しており、ディスプレイ 1 9 上に例えば上記操作 / 表示用画面を表示する。この操作 / 表示用画面の画面データは、上記のように作画エディタ装置 5 上で作成されたものが、通信ライン 3 を介してプログラマブル表示器 1 にダウンロードされたものであり、例えば ROM 1 2 に格納される。

30

【 0 0 5 7 】

ROM 1 2 には、各種プログラム (本体プログラム 2 1 等) や上記画面データ等が格納される。尚、ROM 1 2 は、例えばリード / ライト可能なタイプ (E E P R O M やフラッシュメモリ等) である。RAM 1 3 には、任意の様々なデータ (例えば、外部機器 4 から収集した機器の状態データ等) が記憶される。RAM 1 3 は、基本的に、ワークメモリとして用いられる。

【 0 0 5 8 】

通信コントローラ 1 4 は、上記通信インタフェース 2 に接続しており、上述した通信ライン 3 を介した外部機器 4 や作画エディタ装置 5 との通信を実現する。タッチパネルコントローラ 1 6 は、タッチパネル 1 8 に接続しており、タッチパネル 1 8 上におけるユーザ操作 (押下位置座標等) を検出する。

40

【 0 0 5 9 】

CPU 1 5 は、制御ユニット 1 0 全体を制御する中央処理装置である。CPU 1 5 は、例えば、ROM 1 2 に格納されている上記各種プログラムを読み出し・実行することにより、プログラマブル表示器 1 の各種処理機能を実現する。この各種処理には、例えば後述する図 4 に示す処理等も含まれる。

【 0 0 6 0 】

尚、作画エディタ装置 5 に関しては、上記汎用パソコンは、よく知られているように例えば、CPU、ハードディスク等や、メモリ、ユーザ操作部 (キーボード、マウス等)、

50

ディスプレイ、通信インタフェース等から成る。ハードディスク等には、所定のアプリケーションプログラム等が格納されている。CPUは、このプログラムを読み出し・実行することにより、上記従来の作画エディタとしての機能や、シミュレータ装置としての機能を実現する。このシミュレータ装置としての機能には、上述した従来の機能に加えて、後述する各種機能（図4で説明する機能等）が含まれる。

【0061】

図3は、上記図1に示す表示/制御システムのソフトウェア構成図である。

まず、プログラマブル表示器1における上記ROM12には、図3に示す各種プログラムや各種データが格納される。すなわち、図示の本体プログラム21、通信プログラム22、シミュレータ通信プログラム23等の各種プログラムや、シミュレータ動作判定情報24、画面データ25、マクロデータ26等の各種データが、ROM12に記憶される。

10

【0062】

マクロデータ26は、上記従来や課題で説明した各種マクロデータ（各種マクロブロック）である。よって、特に詳細には説明しないが、その一例は図5に示し後に説明する。尚、既に説明したように、マクロデータ26は、画面データ25に付随するものである。

【0063】

画面データ25は、既に説明したようにプログラマブル表示器1の操作/表示用画面の画面データである。よって、特に詳細には説明しないが、例えば各種アイテム（スイッチ、メータ、ランプ、ボタン等）がスクリーン上の任意の位置に配置されたものである。また、任意のアイテムに、任意のマクロブロックが対応付けられている。例えば、スイッチアイテムを例にすると、任意のスイッチがユーザによってON/OFF操作された場合に、このスイッチに対応するマクロブロックが実行されることになる。

20

【0064】

本体プログラム21は、プログラマブル表示器1全体動作を実現する基本的なプログラムであり、例えば表示/入力制御といった基本的な機能に関するプログラムである。例えば画面データ25を用いてディスプレイ19上に表示/制御用画面を表示する。そして、この表示/制御用画面上でのユーザによるアイテム操作に応じてマクロブロックを実行したり、あるいは定周期でマクロブロックを実行する。つまり、任意のイベントが発生する毎に、このイベントに応じたマクロブロックの処理を実行することになる。

【0065】

30

尚、図には、本体プログラム21が実現する各種機能のうち、通信選択部21aとマクロ実行部21bとを示している。上記マクロブロックの実行は、マクロ実行部21bが行う。また、通信相手の選択は、通信選択部21aが行う。選択された通信相手との通信処理は、通信プログラム22またはシミュレータ通信プログラム23を実行することで行われる。

【0066】

ここで、上記通信プログラム22及びシミュレータ通信プログラム23は、接続機種毎のそれぞれ固有のプロトコルに対応した通信制御に関するプログラムである。つまり、通信プログラム22は、外部機器4に対応した通信制御に関するプログラムである。一方、シミュレータ通信プログラム23は、作画エディタ装置5に対応した通信制御に関するプログラムである。

40

【0067】

また、例えばプログラマブル表示器1においては、ユーザが任意にモード変更できる。つまり、表示器1においては、外部機器4に接続して上述した処理を行う通常モードと、作画エディタ装置5（シミュレータ装置）に接続して擬似的な動作を行うシミュレーションモードとがあり、ユーザは、これらモードを選択できる。

【0068】

そして、通常モードにおいては、本体プログラム21は、通信プログラム22を用いて、外部機器4との通信を行う。一方、シミュレーションモードにおいては、シミュレータ通信プログラム23を用いて通信を行う。尚、上記ユーザによるモード選択結果は、図示

50

のシミュレータ動作判定情報 2 4 として記憶され、本体プログラム 2 1 はシミュレータ動作判定情報 2 4 を参照することで、現在のモードを認識できる。

【 0 0 6 9 】

尚、上記本体プログラム 2 1 や、各通信プログラム 2 2 、 2 3 は、予め（出荷時等に）プログラマブル表示器 1 に記憶されていてもよいし、任意のときに作画エディタ装置 5 等からプログラマブル表示器 1 にダウンロードされて記憶するものであってもよい。尚、これらプログラム 2 1 , 2 2 , 2 3 や、画面データ 2 5 （マクロデータ 2 6 含む）は、ROM 1 2 に記憶される。

【 0 0 7 0 】

また、図 3 に示すように、作画エディタ装置（パソコン；PC）5 は、プログラマブル表示器用作画ソフト 3 0 を有する。プログラマブル表示器用作画ソフト 3 0 は、作画エディタ装置 5 内の不図示の記憶装置（ハードディスク等）に、予め記憶されている。作画エディタ装置 5 内の不図示の CPU 等は、このアプリケーションプログラムを読み出し・実行することにより、例えば図示のシミュレータ 3 1 、マクロビューア 3 2 、通信処理部 3 6 の各種機能部の処理を実現する。また、図示していないが、作画ソフト 3 0 によって、上述した画面データ作成支援機能等も実現される。これによって、上述した画面データ作成支援画面を表示して、ユーザに任意の画面データ（マクロデータ含む）を作成させる。

【 0 0 7 1 】

本例では、作画エディタ装置 5 は、作画エディタ機能だけでなく、シミュレータ装置の機能も備える。そして、本例のシミュレータ装置を構成する機能として、図示のシミュレータ 3 1 、マクロビューア 3 2 、通信処理部 3 6 を示している。

【 0 0 7 2 】

上記画面データ作成支援機能によって任意に作成された画面データ（マクロデータ含む）は、上記ハードディスク等に記憶され、これが図示の画面データ 3 3 及びマクロデータ 3 4 である。そして、ユーザによるダウンロード指示に応じて、これら画面データ 3 3 及びマクロデータ 3 4 が、プログラマブル表示器 1 にダウンロードされて記憶される。これが、図示の画面データ 2 5 及びマクロデータ 2 6 である。

【 0 0 7 3 】

つまり、表示器 1 側が保持する画面データ 2 5 及びマクロデータ 2 6 と同じものを、作画エディタ装置 5 側でも保持している。そして、後述するように、本手法では特にマクロデータ 3 4 （各種マクロブロック）を用いて、ユーザに任意のブレークポイントの設定を行わせる。

【 0 0 7 4 】

シミュレータ 3 1 は、上述したシミュレータ装置としての機能を実現する機能部であり、基本的には上述した既存のシミュレーション処理を実行するものであるが、本手法では更に後述するブレーク位置情報 3 5 を用いた処理（図 4 に示す処理等）を実行する。また、プログラマブル表示器 1 においても、このシミュレータ 3 1 の処理と連携して、後述する図 4 の処理等を実行する。

【 0 0 7 5 】

また、マクロビューア 3 2 は、マクロデータ 3 4 等に基づいて、ユーザに後述するブレークポイントの設定を行わせる。そして、この設定に応じてブレーク位置情報 3 5 を生成する。詳しくは後述する。尚、ブレーク位置情報 3 5 は、作画エディタ装置 5 内のハードディスク等に記憶される。

【 0 0 7 6 】

上記マクロビューア 3 2 と、シミュレータ 3 1 （及び、これに連携するプログラマブル表示器 1 の処理）が、本手法の主な特徴部分である。

尚、本例のシミュレータ装置は、シミュレータ 3 1 とマクロビューア 3 2 とから成るものと考えても良い。

【 0 0 7 7 】

図 4 は、本例の表示 / 制御システムの処理フローチャート図である。つまり、上記プロ

10

20

30

40

50

グラマブル表示器 1 及び作画エディタ装置 5 の処理フローチャート図である。

まず、プログラマブル表示器 1 側においては、本体プログラム 2 1 が、画面データ 2 5 を用いて、上述した状態表示や制御等を行っている。そして、任意のマクロブロックを実行する毎に、マクロ実行部 2 1 b がこのマクロブロックの処理を実行する。

【 0 0 7 8 】

マクロ実行部 2 1 b は、まず、シミュレータ動作判定情報 2 4 を参照して、現在、通常の動作（上記通常モード）であるか、シミュレータ装置と連動して動作しているか（上記シミュレーションモードであるか）を判定する（ステップ S 1 1）。

【 0 0 7 9 】

通常の動作（シミュレータ装置と連動していない）の場合（ステップ S 1 1 , N O）、既存の一般的な処理を行うことになる。すなわち、任意に設定したタイミングに従い、実行対象のマクロブロック（複数行から成る）を、当該マクロブロックの先頭行から順番に 1 行ずつ実行していき（ステップ S 1 1 , N O でステップ S 1 4 実行後ステップ S 1 5 , N O によるループ処理）、最終行まで処理を実行したら（ステップ S 1 5 , Y E S）、当該マクロブロックの処理を終了する。そして、例えば、次のマクロブロックの処理に遷移する。

【 0 0 8 0 】

尚、図では、ステップ S 1 5 の判定が N O の場合（未だ、最終行まで実行していない場合）、ステップ S 1 1 に戻るが、この例に限らず、ステップ S 1 4 に移行するようにしてもよい。

【 0 0 8 1 】

一方、上記シミュレーションモードの場合すなわちシミュレータ装置と連動して動作している場合（ステップ S 1 1、Y E S）、任意に設定したタイミングに従い、基本的には上記通常の動作と同様に当該マクロブロックの先頭行から順番に 1 行ずつ実行していくことになるが、1 行毎に、その行の処理（マクロコマンド等）を実行する前に、作画エディタ装置 5 のシミュレータ 3 1 に対して、今から実行するマクロブロックのブロック番号と行番号を通知する（ステップ S 1 2）。

【 0 0 8 2 】

尚、ブロック番号は、各マクロブロックを識別する為に各マクロブロックに予め割り当てた識別番号である。行番号は、マクロブロック内の先頭から何番目の行であるかを示すものである。

【 0 0 8 3 】

シミュレータ 3 1 は、通知されたブロック番号に対応するマクロブロックのブレーク位置情報 3 5 を取得する。そして、このブレーク位置情報 3 5 と通知された行番号とに基づいて、通知された行番号の行が、ブレーク対象であるか否かを判定する（ステップ S 2 1）。

【 0 0 8 4 】

ここで、上記ブレーク位置情報 3 5 について説明する。

上記の通り、ブレーク位置情報 3 5 は、マクロビューア 3 2 によってユーザが任意にブレークポイントの設定を行うことで作成される。

【 0 0 8 5 】

上記の通り作画エディタ装置 5 側では、プログラマブル表示器 1 側で保持・実行する画面データ 2 5 及びマクロデータ 2 6 と同じデータを保持している。すなわち、画面データ 3 3 及びマクロデータ 3 4 を保持している。

【 0 0 8 6 】

これより、本例のシミュレータ装置の 1 機能（アプリケーション）であるマクロビューア 3 2 は、例えばシミュレーション開始前の任意のときにユーザが任意に指定したマクロブロックの内容を、作画エディタ装置 5 のディスプレイに表示する。あるいは、上記ステップ S 1 2 による通知があったときに、通知されたブロック番号のマクロブロックの内容を表示する。何れにしても、例えば、図 6 に示すようなブレークポイント設定画面 4 0 内

10

20

30

40

50

に、マクロブロックの内容（全ての行）が表示される。

【0087】

ここで、図5に、マクロデータ34（26）の一例を示す。図示の通り、マクロデータ34（26）は、複数のマクロブロック（1～n）より成るものである。そして、各マクロブロックは、複数行のプログラム（例えばマクロコマンドを用いたプログラム文）より成る。図示のマクロブロック1の例の場合、行番号が‘0’～‘m’までの“m+1”行のプログラムとなっている。

【0088】

マクロビューア32は、例えば図6に示すブレークポイント設定画面40を、ディスプレイ上に表示する。このブレークポイント設定画面40には、任意のマクロブロックの内容（プログラムリスト）が表示される。図示の例では、上記図5に示すマクロブロック1の内容が表示されている。

10

【0089】

ユーザは、このブレークポイント設定画面40上において、表示されるマクロブロックのプログラムリストにおける任意の“行”を指定することができる。例えば、マウス操作等により所望の“行”の表示位置をクリックすることで、指定することができる。

【0090】

また、ブレークポイント設定画面40上には、図示の「ブレークポイント設定」ボタン41、「ブレークポイント解除」ボタン42、「実行」ボタン43の3種類の操作ボタンが表示される。

20

【0091】

ユーザは、所望の“行”をブレークポイントとして設定したい場合には、まず上記のように所望の“行”を指定した後、「ブレークポイント設定」ボタン41を操作する。この設定結果は、マクロビューア32によって、ブレーク位置情報35に反映される。ブレーク位置情報35は、例えば各マクロブロックのブロック番号に対応付けて、そのマクロブロックに関して上記ブレークポイントとして設定された“行”の行番号が記憶されるものである。

【0092】

ここで、ブレークポイントとは、マクロブロックの実行を一旦停止する位置（行）を示すものである。

30

また、ブレークポイント設定画面40上においては、上記ブレークポイントとして設定された“行”がユーザに分かるような表示が行われる。図示の例では設定された“行”の左側に黒丸が表示される。図示の例では、3行目（行番号=2）の行が、ブレークポイントとして設定されていることが分かる。尚、ブレークポイントとして設定する行の数は、図示の例では1つだけだが、複数の行をブレークポイントとして設定してもよい。極端な例では全ての行をブレークポイントとして設定することも可能である。

【0093】

また、ユーザは、ブレークポイントとして設定された“行”のなかで任意の行の設定を解除したい（取り消したい）場合には、設定済みの“行”のなかで所望の“行”を指定した後、「ブレークポイント解除」ボタン42を操作する。これによって、この“行”の上記黒丸表示が消えると共に、ブレーク位置情報35においてこの“行”の行番号が削除される。

40

【0094】

上記ブレークポイント設定画面40は、ユーザが任意のときにマクロビューア32を起動して表示させることもできるが、シミュレータ31が動作しているときには（図4の処理を実行中）自動的に表示される。ユーザは、予めブレークポイント設定画面40上で任意のマクロブロックに関して上記ブレークポイントを設定してもよいし、シミュレーション動作中にブレークポイントを設定してもよい。尚、シミュレーション動作中は、基本的に、上記ステップS12で通知されたブロック番号のマクロブロックの内容が、ブレークポイント設定画面40上に表示されることになる。

50

【 0 0 9 5 】

尚、図 6 には示していないが、任意のマクロブロック実行中にその内容がブレークポイント設定画面 4 0 上に表示されている場合には、更に、どの行まで実行されたのかがユーザが分かるような何らかの表示を行うようにしてもよい。

【 0 0 9 6 】

ここで、図 4 のフローチャートの説明に戻る。

上記の通り、ステップ S 2 1 の説明に関して、図 5、図 6 を参照して、ブレークポイントの設定と、この設定に応じたブレーク位置情報 3 5 について説明した。この説明により、ステップ S 2 1 の処理は、ブレーク位置情報 3 5 において上記通知されたブロック番号のマクロブロックに関するブレークポイントの情報を参照することで行われる。つまり、上記の通り、ブレークポイントとして設定された 1 または複数の“行”の行番号が登録されているので、その中に上記通知された行番号があるか否かを判定する。

10

【 0 0 9 7 】

例えば、図 6 に示す例では、ブレーク位置情報 3 5 においてマクロブロック 1 に関しては、行番号 = ‘ 2 ’ が登録されているので、もし、表示器 1 側で実行中のマクロブロックがマクロブロック 1 であったならば、ステップ S 1 2 の処理により通知された行番号が ‘ 2 ’ のときに、ステップ S 2 1 の判定が Y E S となることになる。

【 0 0 9 8 】

最初は、先頭行の行番号 (‘ 0 ’) が通知されるので、ステップ S 2 1 の判定は N O となり、表示器 1 のマクロ実行部 2 1 b に対して所定の返信を行う (ステップ S 2 3)。ここでは、実行許可を返信するものとする。

20

【 0 0 9 9 】

表示器 1 のマクロ実行部 2 1 b は、上記ステップ S 1 2 の通知を行った後は、実行許可待ち状態 (待機状態) となっている (ステップ S 1 3)。そして、上記ステップ S 2 3 により実行許可が返信されてきたら (ステップ S 1 3 , Y E S)、ステップ S 1 2 で通知した“行”の処理を実行する (ステップ S 1 4)。最初は、先頭行の処理を実行することになる。

【 0 1 0 0 】

そして、ステップ S 1 4 の処理を実行完了したら、当該実行した“行”が最終行であるか否かを判定する (ステップ S 1 5)。最終行であれば (ステップ S 1 5 , Y E S)、本処理を終了する。最終行でなければ (ステップ S 1 5 , N O)、ステップ S 1 1 の処理に戻る。尚、図ではステップ S 1 1 に戻るが、ステップ S 1 2 に戻るようにしてもよい。

30

【 0 1 0 1 】

そして、ステップ S 1 2 において、次の実行すべき“行”の行番号等を、シミュレータ 3 1 に通知する。今度は、行番号 = ‘ 1 ’ が通知されるので、上記の例ではステップ S 2 1 の判定は N O となり、上記行番号 = ‘ 0 ’ の場合と同様の処理が実行されることになる。

【 0 1 0 2 】

続いて、行番号 = ‘ 2 ’ が通知されると、上記の例では今度はステップ S 2 1 の判定は Y E S となり、マクロ実行再開待ち状態となる (ステップ S 2 2)。そして、ユーザが上記ブレークポイント設定画面 4 0 上で「実行」ボタン 4 3 を操作すると、ステップ S 2 3 の処理に移行する。この場合も、ステップ S 2 3 で実行許可を返信することになる。

40

【 0 1 0 3 】

ここで、上記ステップ S 2 2 のマクロ実行再開待ち状態において、シミュレータ装置は、上述した従来の既存機能により (よって、特に図示や説明はしない)、上記擬似メモリの内容をディスプレイ上に表示する。ユーザは、シミュレート装置中の擬似メモリの内容を一覧表示で確認することにより、マクロが期待した通りの動作を行っているかを判断することが出来る。つまり、ユーザは、上記ブレークポイントまでの処理に異常があるか否かを判断できる。

【 0 1 0 4 】

50

上記の例において、ユーザが異常ありと判定したならば、それまでに実行した行の中に（本例では行番号 = ' 0 ' ~ ' 1 ' の 2 行の中に）、問題がある行が存在することになる。これより、ユーザは、上記マクロビューア 3 2 の機能により、今度は例えば行番号 = ' 0 ' ~ ' 1 ' の 2 行全てを、ブレークポイントとして設定する。そして、ユーザは、表示器 1 側で所定の操作を行う等して、上記マクロブロックを、再度、先頭行から順次実行させる。これによって、問題がある行を特定することができる。

【 0 1 0 5 】

一方、ユーザは、上記確認作業によって問題が無いと判断した場合には、「実行」ボタン 4 3 を操作する。これによって、ステップ S 2 2 のマクロ実行再開待ち状態が解除され、ステップ S 2 3 によって実行許可を返信することで、表示器 1 側では上記行番号 = ' 2 ' の行の処理を実行することになる。そして、4 行目以降についても、最終行の処理を実行するまで、各行毎に上述した処理を実行していくことになる。図 6 には示していないが、4 行目以降にもブレークポイント行があったならば、一時的に実行停止して、ユーザが上記一覧表示に基づいて異常判断を行うことになる。

10

【 0 1 0 6 】

また、図 6 には示していないが、上記ブレークポイント設定画面 4 0 上に更に不図示の「ステップ実行」ボタンを設けるようにしてもよい。この場合、ユーザは、上記ステップ S 2 2 のマクロ実行再開待ち状態において、再開すべきと判断した場合、上記「実行」ボタン 4 3 と「ステップ実行」ボタンの何れか一方を操作する。

【 0 1 0 7 】

20

そして、「ステップ実行」ボタンが操作された場合には、シミュレータ装置は、上記通知された行の次の行に対して、ブレークポイントの設定を行ったうえで、上記ステップ S 2 3 の返信処理を行う。これによって、ユーザによる設定内容に関係なく、必ず次の行においてもステップ S 2 1 の判定が Y E S となり、マクロ実行が一時停止することになる。

【 0 1 0 8 】

この様に、「ステップ実行」ボタンを設けたことにより、ユーザは、デバッグ中に問題がありそうな部分を見つけたならば（経験や勘に基づき判断する）、「ステップ実行」ボタンを操作することで、この部分を集中的に検証することが可能となる。

【 0 1 0 9 】

以上説明したように、本手法によれば、ユーザは予めまたはデバッグ作業中に、任意の 1 または複数の行をブレークポイントとして設定する作業を行うだけで、マクロブロック内における問題となる行を見つけて修正する作業（デバッグ作業）を実現することができ、効率的なデバッグ作業を行えるようになる。これによって、デバッグ作業に掛かる時間を、従来よりも短縮することができる。また、ユーザのデバッグ作業負担を軽減することができる。

30

【 0 1 1 0 】

この様な本手法の時間短縮効果について、以下、具体的な一例を用いて説明する。

ここでは、例えば、1 0 0 行から成るマクロブロックを例にして、1 0 0 行目に何らかの問題があるものとする。

【 0 1 1 1 】

40

この例の場合、従来のシステムでは、例えば以下に述べる手順によりデバッグ作業を行う。

マクロブロックの修正 画面転送 確認を、1 ~ 1 0 0 行目まで繰り返し行う。

【 0 1 1 2 】

すなわち、既に従来課題で一例を述べた通り、作画エディタ装置 5 において、ユーザが、マクロブロックの内容を修正する。修正方法は、既に述べた通り様々であってよいが、ここでは分かり易いように、先頭行から 1 行ずつ検証していく方法を例にする。よって、ここでは、最初はマクロブロックの内容が先頭の 1 行目のみと成るように修正し、次は 1 ~ 2 行目のみとなるように修正し、その次は 1 ~ 3 行目のみと成るように修正する。これを問題の行が見つかるまで、繰り返し行うが、上記の通り本例では 1 0 0 行目に問題が

50

あるので、結局、1～100行から成るマクロブロックを作成して検証するまで、終わらないことになる。

【0113】

尚、実際には、まず、複数のマクロブロックのなかで問題があるマクロブロックを抽出し、その後問題があるマクロブロック毎に問題がある行を特定する作業を行うことになる場合が多いと考えられる。つまり、元のマクロブロック(100行から成るもの)について動作確認したら、異常があった為に、上記問題行を特定する為のデバッグ作業を行うことになる場合が多いと考えられる。よって、上記の例では、1～99行から成るマクロブロックを作成して検証して異常が無いことを確認できた時点で、100行目に問題があることは推定可能である。しかし、これは推定であるので、ここでは、1～100行から成るマクロブロックを作成して検証するまでは、問題の行を特定できないものとして説明する。

10

【0114】

そして、既に従来課題で述べた通り、上記マクロブロックの修正を行う毎に、修正したマクロブロックをプログラマブル表示器1にダウンロード(転送)して、プログラマブル表示器1側で保持するマクロブロックを更新させる。そして、プログラマブル表示器1において、このマクロブロックを実行させる。

【0115】

そして、このマクロブロックが最後まで実行された時点で、ユーザは、擬似メモリの内容を一覧表示させる等して実行結果を参照することができるので、問題があるか否かを確認する。問題がなければ、マクロブロックを修正して、再び上記転送、実行、確認を行うことになる。

20

【0116】

ここで、上記マクロブロック修正、画面転送、確認に掛かる時間を、それぞれ以下の通りと仮定する。

(a) 画面作画(マクロブロック修正) : 20秒

(b) 画面転送 : 20秒

(c) 動作確認 : 20秒

よって、この例では、デバッグ作業に掛かる時間(ここでは問題の行を特定するまでに掛かる時間)は、以下の通りとなる。

30

【0117】

$$((a) + (b) + (c)) \times 100 (\text{行}) = 6000 \text{秒} (\text{約}100 \text{分})$$

一方、本例のシステムによれば、上記(a)画面作画と(b)画面転送が必要なくなり、(c)動作確認に掛かる時間と、ブレークポイント設定に掛かる時間だけで済む。上記の例の場合、全ての行をブレークポイントとする設定を行うことになる。あるいは、先頭行のみブレークポイント設定しておき、デバッグ作業中は必ず上記「ステップ実行」ボタンを操作するようにしてもよい。何れの方法でも、先頭行から1行ずつ動作確認を行うことになる。また、特に上記の例の場合、ブレークポイントとすべき行をどこにすべきか考える必要がなく、先頭行または全行指定で設定が完了するので、設定作業に掛かる時間は極めて短くて済む(例えば、数十秒程度)。

40

【0118】

(c)動作確認に掛かる時間は、従来と同じである。

すなわち、マクロのステップ実行による1行毎の動作確認に掛かる時間は20秒であるので、

$$20 \text{秒} \times 100 \text{行} = 2000 \text{秒} (\text{約}33 \text{分})$$

となる。これに、上記設定作業に掛かる時間を加えたとしても、従来(100分)に比べて、非常に短い時間で済む(ほぼ1/3程度で済む)。

【0119】

このように、画面作画及び画面転送が不要になるため、作業効率が非常に向上し、デバッグ作業に掛かる時間が大きく短縮できる(1/3程度)。

50

画面データのサイズが大きい場合や、マクロブロック内の行数が多いほど/マクロブロックの数が多ければ、作業効率の向上(時間短縮)の効果は更に大きくなるものと考えられる。

【0120】

また、以上説明した一例は、プログラマブル表示器に関しては実機を用いるが、PLC等に関しては実機を用いずにパソコン上でシミュレータ装置によって擬似的な動作を行う場合を前提にしたものであるが、この例に限るものではない。

【0121】

上述したように、従来技術において、パソコン上等において、プログラマブル表示器の機能・動作を擬似的に実現することも行われている。つまり、プログラマブル表示器の動作をエミュレートするエミュレート装置が、従来より存在している。そして、既に述べた通り、従来より、パソコン上等において上記シミュレータ装置とエミュレート装置の両方を動作させることで、プログラマブル表示器の画面データのデバッグを行うことが知られている。

10

【0122】

本手法は、このような構成に対しても適用可能である。この場合、図3には示さないが、作画エディタ装置5のプログラマブル表示器用作画ソフト30に、更に上記エミュレート装置の機能を備えさせるようにする。そして、このエミュレート装置と上記シミュレータ装置とで連携処理を行って、上述したデバッグ作業を実現させる。尚、この場合、エミュレート装置には、従来の処理機能に加えて更に、例えば図4に示すマクロ実行部21bの処理と略同様の処理を行う機能も備えるようにする。勿論、この場合には、シミュレータ装置とエミュレート装置は、同じパソコン上(作画エディタ装置5)に構築されているので、通信プログラムによる通信ライン2を介した通信処理等は必要ない。

20

【0123】

また、この場合、上記モードの設定や設定モードに応じたステップS11の処理は、行ってもよいし、行わなくてもよい。モード設定を行う場合は、作画エディタ装置5上でユーザが設定する。例えば、シミュレータ装置と連携動作させずにエミュレート装置単独で動作させたい場合には、上記通常モードに設定することで、ステップS11の判定がNOとなる処理を実行させることになる。

【0124】

この様に、本手法は、プログラマブル表示器が実機のケース、その動作を模擬するエミュレート装置のケースの何れのケースに対しても、適用可能であり、上述した効果を奏するものとなる。

30

【0125】

尚、上記表示/制御システムにおいて、プログラマブル表示器1の通信処理部は、作画エディタで設定された外部機器によりそれぞれ固有の通信プログラムに差し替えて動作するような汎用的な構成にしておくことが実用的である。この場合、本体プログラム21と通信プログラム22間のインタフェースは、外部機器固有の通信プロトコルに依存しない汎用的な共通インタフェースにしておくことが理想的である。

【0126】

最後に、上記表示/制御システムにおいて、既存技術に関することについて、以下に捕捉説明しておく。

40

* 従来のシミュレータ機能の概要

表示器1の動作を確認する場合、表示器1と実際の外部機器(実機;外部機器4)を接続して動作確認する方法と、例えば特許文献3のように、表示器とシミュレータ装置(パーソナルコンピュータ等によって構成される、外部機器(実機)をシミュレートするシミュレータ装置)を接続して、動作確認する方法が提案されている。また、パーソナルコンピュータ等によって構成される、表示器をエミュレートするエミュレート装置と、同パーソナルコンピュータ上に構成されるシミュレータソフトとの連動により、動作確認する方法が考えられる。

50

【 0 1 2 7 】

上記表示 / 制御システムにおいて、表示器 1 - 外部機器 4 間の連携は、通信ライン 3 を介して行うコマンドによって構築され、外部機器 4 の制御はコマンドによるデバイスメモリの操作によって行われる。これは通信ラインを介して行うコマンドで連携できれば外部機器は実機（外部機器 4）である必要がないとも考えられる。

【 0 1 2 8 】

プログラブル表示器 1 のシミュレータ装置は、上記の様な考え方に基づき、シミュレートするデバイスメモリを擬似的に用意し、シミュレータ専用のコマンドで表示器 1 との連携を行う。この様にすることで、表示器 1 が擬似的に外部機器を接続した状態にして、表示器 1 の動作を確認することができるようにしたものである。尚、表示器 1 の動作とは、主に本体プログラム 2 1 が画面データ 2 5 を用いて実行する処理のことであり、シミュレータ装置を用いることで画面データの動作確認を行うものである。そして、特に画面データ 2 5 に付随する各マクロブロックの動作確認を行うものである。

10

【 0 1 2 9 】

例えば従来技術では、図 3 のように表示器 1 とシミュレータ装置によって画面データの動作を確認する場合、表示器 1 の通信プログラム 2 2 をシミュレータ通信用の通信プログラム 2 3 に切換え、実際に表示器 1 と通信する外部機器をシミュレータ装置に変更することで、あたかも実機（外部機器 4）に接続しているように表示器 1 の動作を確認できる。エミュレータ装置と連動する場合にも同様の方法により、シミュレータソフトは表示器 1 と通信する場合とエミュレータと連動する場合とで、通信インタフェースの経路を差し替えることで、容易に双方と通信可能な構造になっている。

20

【 0 1 3 0 】

また、既に述べた通り、従来より、シミュレータ装置上に構築された擬似メモリの内容を表示 / 編集する GUI 手段が存在している。これにより、ユーザは、擬似メモリの内容を参照することで上記表示器の動作に異常があるか否か判断できたり、あるいは擬似メモリの内容を容易に変更することができる。

【 0 1 3 1 】

シミュレータ装置は、シミュレート対象の画面データファイルと実際に表示器が表示している画面により、どのような擬似メモリを表示するのかを解析し、その画面で使用するメモリを一覧で表示する等の機能も有する。

30

【 0 1 3 2 】

また、以下に既存のマクロデータ（各種マクロブロック）の具体的を列挙しておく。

尚、マクロデータは、図 5 で示す通り画面データの一部であり、プログラブル表示器内とシミュレータソフト内に同じ内容のデータを持つ。各種マクロブロックを実行することによって、演算処理や、プログラブル表示器の動作や外部機器のメモリ操作等を、任意のタイミングで行うことができる。この任意のタイミングについても、具体例を以下に示している。尚、以下の具体例では、各種マクロブロック（オープンマクロ、ONマクロ等）を、例えば[スクリーン]、[スイッチ]等のようにグループ分けしたうえで示している。また、具体的なマクロブロック名称を示さず、概略的な説明のみとしている場合もある。

40

【 0 1 3 3 】

[スクリーン]

- オープンマクロ：スクリーンを表示したとき1回実行
- クローズマクロ：スクリーンを切り替えたとき1回実行
- サイクルマクロ：スクリーンを表示中繰り返し実行

[マルチオーバーラップ]

- オープンマクロ：マルチオーバーラップを表示したとき1回実行
- クローズマクロ：マルチオーバーラップを消したとき1回実行

[スイッチ]

- ONマクロ：スイッチを押したとき1回実行

50

- OFFマクロ：スイッチから指が離れたとき1回実行
[ファンクションスイッチ]
 - ONマクロ：ファンクションスイッチを押したとき1回実行
 - OFFマクロ：ファンクションスイッチから指が離れたとき1回実行
[初期マクロ]
- コントローラと通信を開始する前にマクロブロックのマクロを1回実行。

【0134】

[グローバルマクロ]

読込エリアのマクロ実行ビットが0 1(エッジ)でマクロブロックのマクロを1回実行

10

【0135】

[イベントタイママクロ]

表示している画面に関係なく、常時設定時間ごとにマクロブロックのマクロを実行。

[インターバルタイマ]

インターバルタイマを設定したスクリーンを表示中、設定内容に従ってタイマが起動。

【0136】

タイムアップする度に、マクロブロックのマクロを実行。

[マクロモード]

マクロモードが設定されている画面を表示中、設定メモリの状態によってマクロを実行

20

【0137】

- ONマクロ：メモリのビットが0 1(エッジ)で実行
- OFFマクロ：メモリのビットが1 0(エッジ)で実行

【符号の説明】

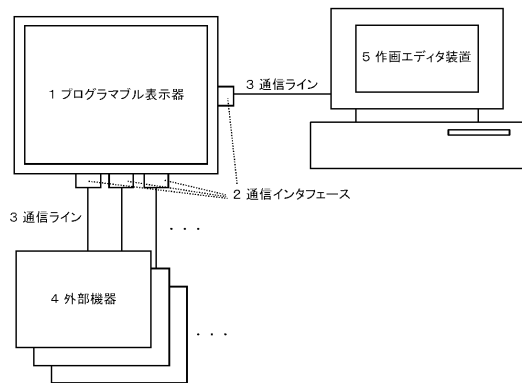
【0138】

- | | | |
|-----|------------------|----|
| 1 | プログラマブル表示器 | |
| 2 | 通信インタフェース | |
| 3 | 通信ライン | |
| 4 | 外部機器 | |
| 5 | 作画エディタ装置 | 30 |
| 10 | 制御ユニット | |
| 11 | グラフィックコントローラ | |
| 12 | ROM | |
| 13 | RAM | |
| 14 | 通信コントローラ | |
| 15 | CPU | |
| 16 | タッチパネルコントローラ | |
| 17 | 内部バス | |
| 18 | タッチパネル | |
| 19 | ディスプレイ | 40 |
| 21 | 本体プログラム | |
| 21a | 通信選択部 | |
| 21b | マクロ実行部 | |
| 22 | 通信プログラム | |
| 23 | シミュレータ通信プログラム | |
| 24 | シミュレータ動作判定情報 | |
| 25 | 画面データ | |
| 26 | マクロデータ | |
| 30 | プログラマブル表示器用作画ソフト | |
| 31 | シミュレータ | 50 |

- 3 2 マクロビューア
- 3 3 画面データ
- 3 4 マクロデータ
- 3 5 ブレーク位置情報
- 4 0 ブレークポイント設定画面
- 4 1 「ブレークポイント設定」ボタン
- 4 2 「ブレークポイント解除」ボタン
- 4 3 「実行」ボタン

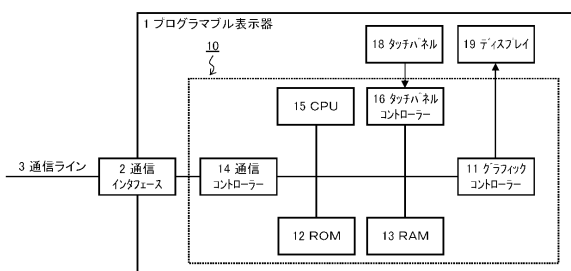
【図 1】

本例の表示制御システムの概略構成図



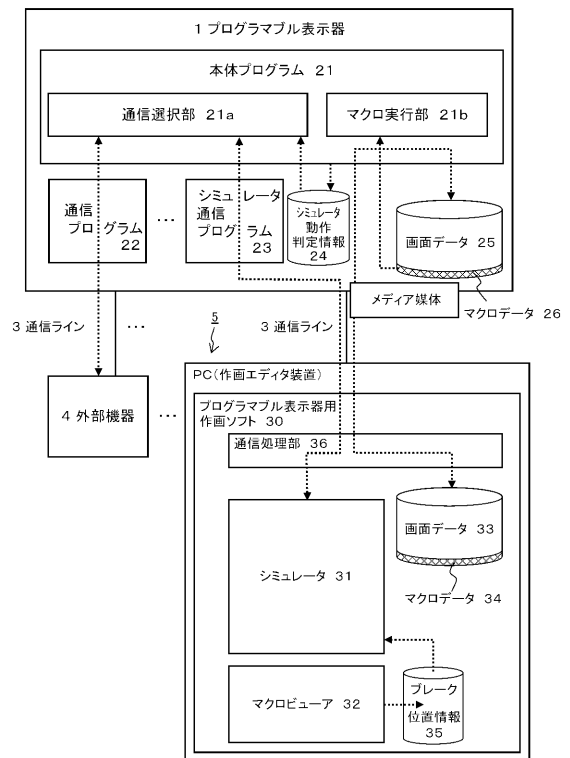
【図 2】

プログラマブル表示器のハードウェア構成図



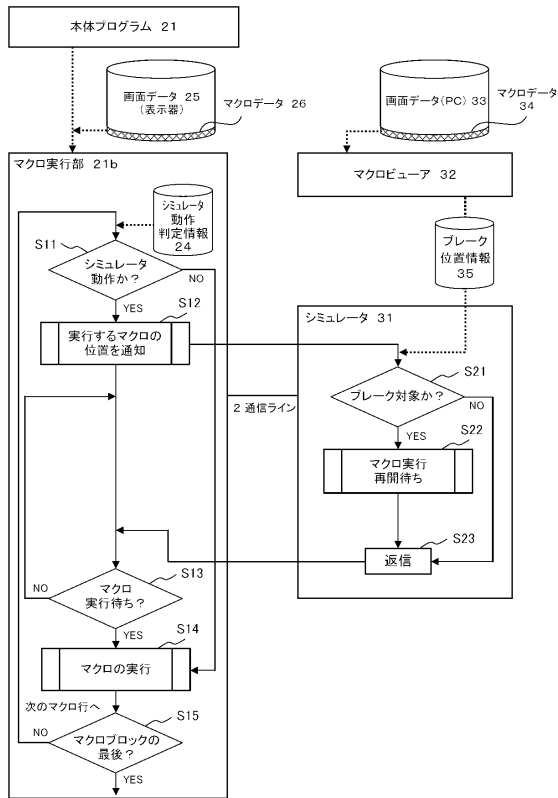
【図 3】

図1に示す表示制御システムのソフトウェア構成図



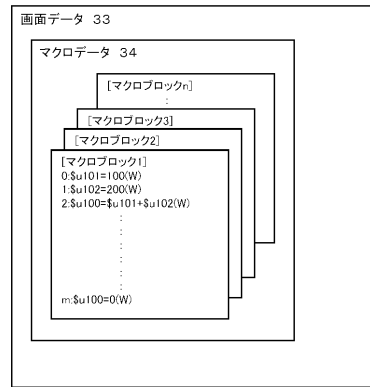
【図4】

本例の表示/制御システムの処理フローチャート図



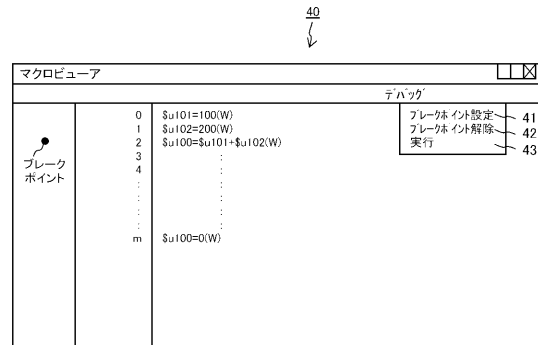
【図5】

マクロデータの一例を示す図



【図6】

マクロビューアによるブレークポイント設定画面の一例



フロントページの続き

(56)参考文献 特開2001-282326(JP,A)
特開2007-109120(JP,A)
特開2009-223471(JP,A)

(58)調査した分野(Int.Cl., DB名)

G06F 11/28
G06F 11/36
G05B 19/05