

(19) 日本国特許庁(JP)

(12) 公開特許公報(A)

(11) 特許出願公開番号

特開2012-243136

(P2012-243136A)

(43) 公開日 平成24年12月10日(2012.12.10)

(51) Int.Cl. F I テーマコード (参考)
G05B 19/05 (2006.01) G05B 19/05 B 5H220

審査請求 未請求 請求項の数 2 O L (全 11 頁)

<p>(21) 出願番号 特願2011-113585 (P2011-113585) (22) 出願日 平成23年5月20日 (2011.5.20)</p>	<p>(71) 出願人 502129933 株式会社日立産機システム 東京都千代田区神田練堀町3番地 (74) 代理人 110000350 ポレール特許業務法人 (72) 発明者 須藤 栄一 新潟県胎内市富岡46番地1 株式会社日立産機システム内 Fターム(参考) 5H220 BB12 CC07 CX01 CX06 DD04 JJ12 JJ24</p>
--	--

(54) 【発明の名称】 ラダープログラミングツール

(57) 【要約】

【課題】

従来は、プログラマブルコントローラに用いるシーケンスプログラムをデバッグする装置で、入出力接点の状態が時系列にどのように変化したかが分かるデバッガを提供するものがあった。

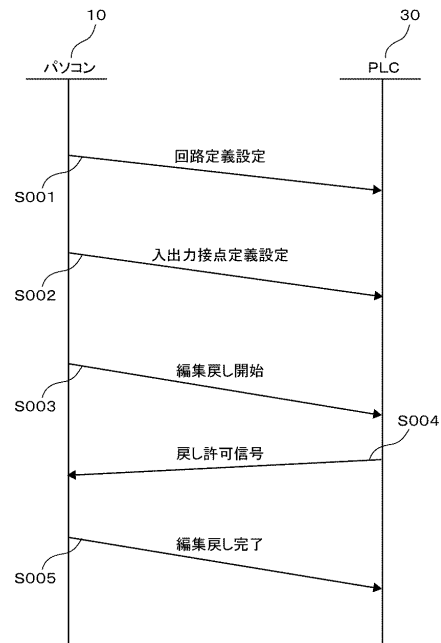
また、ラダープログラムを編集し直すことなく、着目した回路に限定して動作を確認することができるデバッグ装置およびプログラム確認方法があった。

【解決手段】

本発明は、プログラマブルコントローラのためのラダープログラミングツールにおいて、ユーザが作成したラダープログラムをデバッグするとき、ラダープログラムの変更を前記プログラマブルコントローラの記憶素子に記憶し、ユーザの要望に応じて、変更部分を前記パソコンの表示装置に遡って表示し、ラダープログラムの編集を行うことを特徴とする。

【選択図】 図1

図1



【特許請求の範囲】

【請求項 1】

パソコンと、
 該パソコンと接続され、パソコンの内容を表示する表示装置と、
 前記パソコンとプログラマブルコントローラとを接続し、
 該プログラマブルコントローラは、中央演算装置、通信制御回路、記憶素子、入出力制御回路、入出力装置より構成され、
 前記パソコンと前記プログラマブルコントローラの中央演算装置は、前記通信制御回路を介して情報を送受信可能な構成で行うラダープログラミングツールにおいて、
 ユーザが作成したラダープログラムをデバッグするとき、ラダープログラムの変更を前記プログラマブルコントローラの記憶素子に記憶し、
 ユーザの要望に応じて、変更部分を前記パソコンの表示装置に遡って表示し、ラダープログラムの編集を行うことを特徴とするラダープログラミングツール。

10

【請求項 2】

請求項 1 記載のラダープログラミングツールにおいて、
 ラダープログラムの変更部分はすべて前記プログラマブルコントローラの記憶素子に記憶することを特徴とするラダープログラミングツール。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、プログラマブルコントローラのラダープログラミングツールにおいて、編集戻し機能を備えたラダープログラミングツールに関する。

20

【背景技術】

【0002】

プログラマブルコントローラは、一般に CPU やメモリなどを内蔵したコントローラであり、センサやスイッチなどの入力信号を入力し、この入力信号のデータのプログラム演算を行い、この演算結果を出力データとして出力し、電磁弁、モータ又は表示灯などを ON - OFF して制御するものである。

【0003】

また、プログラマブルコントローラのプログラムは、一般にラダープログラムを使用する。ラダープログラムは、リレー回路を記号化したラダー図をベースにしてロジックを記述するプログラムで、入力条件と出力の関係を視覚的にイメージし易いプログラム言語である。このラダープログラムを作成するため、専用のエディタであるラダープログラミングツールを使ってプログラムを作成している。また、本発明の公知例として、特許文献 1 (特開 2011 - 22838) 及び特許文献 2 (特開平 8 - 305418 号公報) がある。

30

【先行技術文献】

【特許文献】

【0004】

【特許文献 1】特開 2011 - 22838

40

【特許文献 2】特開平 8 - 305418 号公報

【発明の概要】

【発明が解決しようとする課題】

【0005】

特許文献 1 には、ラダープログラムを編集し直すことなく、着目した回路に限定して動作を確認することができるデバッグ装置およびプログラム確認方法を実現することについて開示されている。

【0006】

また、特許文献 2 には、プログラマブルコントローラに用いるシーケンスプログラムをデバッグする装置で、入出力接点の状態が時系列にどのように変化したかが分かるデバッ

50

ガを提供することが記載されている。しかし、入出力接点を変更することは開示されていない。

【0007】

本発明の目的は、実際にデバッグしたラダープログラムにおいて変更した箇所を記憶して格納し、ユーザは過去の変更した箇所を表示して選択して編集することができるプログラマブルコントローラのためのラダープログラミングツールを提供することにある。

【課題を解決するための手段】

【0008】

本発明は、上記目的を達成するために、パソコンと、該パソコンと接続され、パソコンの内容を表示する表示装置と、前記パソコンとプログラマブルコントローラとを接続し、該プログラマブルコントローラは、中央演算装置、通信制御回路、記憶素子、入出力制御回路、入出力装置より構成され、前記パソコンと前記プログラマブルコントローラの中央演算装置は、前記通信制御回路を介して情報を送受信可能な構成で行うラダープログラミングツールにおいて、ユーザが作成したラダープログラムをデバッグするとき、ラダープログラムの変更を前記プログラマブルコントローラの記憶素子に記憶し、ユーザの要望に応じて、変更部分を前記パソコンの表示装置に遡って表示し、ラダープログラムの編集を行うことを特徴とする。

10

【0009】

また、上記のラダープログラミングツールにおいて、ラダープログラムの変更部分はすべて前記プログラマブルコントローラの記憶素子に記憶することを特徴とする。

20

【発明の効果】

【0010】

本発明は、ユーザが実際に過去にデバッグしたラダープログラムを表示し、選択できるラダープログラミングツールを提供することができ、また、ユーザは過去にデバッグを行なった動作のラダープログラムを格納しているため、プログラマブルコントローラを使用した装置の動作の急な変更にも対応できる。また、過去にデバッグを行なったラダープログラムを再利用できるため、そのラダープログラムのデバッグの時間短縮効果を得られる。

【図面の簡単な説明】

【0011】

30

【図1】本発明の全体のシーケンスを示す図である。

【図2】本発明のパソコン側での回路定義設定のフローチャートを示す図である。

【図3】本発明のプログラマブルコントローラ側での回路定義設定のフローチャートを示す図である。

【図4】本発明のパソコン側での入出力接点変更設定のフローチャートを示す図である。

【図5】本発明のプログラマブルコントローラ側での入出力接点変更設定のフローチャートを示す図である。

【図6】本発明のパソコン側でのラダープログラムの編集戻し設定のフローチャートを示す図である。

【図7】本発明のプログラマブルコントローラ側での編集戻し設定のフローチャートを示す図である。

40

【図8】ハードウェアの構成を示す図である。

【図9】図8に示したハードウェア構成の記憶素子のデータ保管構造例を示す図である。

【図10】本発明のデバッグ対象となるラダープログラムの一例を示す図である。

【図11】本発明のデバッグ対象となるラダープログラムの別の例を示す図である。

【発明を実施するための形態】

【0012】

以下、本発明の実施の形態について図面を用いて説明する。

【0013】

図1は、本発明の全体のシーケンスを示したものである。プログラマブルコントローラ

50

では、プログラマブルコントローラ本体とパソコンを用いてユーザが作成したラダープログラムをプログラマブルコントローラで装置を動作させるためにそのラダープログラムの動作検証を行う。一般にこれをデバッグと呼んでいる。

本発明は、ラダープログラムのデバッグの際、修正を加え、何回もデバッグを行う。そのとき修正前に遡って入出力信号をユーザが選択できるプログラマブルコントローラのプログラミングツールを提供するものである。そのために、ユーザは、装置を動作させるため作成するラダープログラムを、その装置の機能毎に作成する。

【0014】

図1に示すように、パソコンP001とプログラマブルコントローラ(プログラマブルコントローラ01)間の全体の通信制御において、先ず、ユーザが作成するラダープログラムを機能単位で宣言する必要があり、パソコン側(P001)よりプログラマブルコントローラ側(プログラマブルコントローラ01)へ回路定義設定(ステップS001)を送信してプログラマブルコントローラ側(プログラマブルコントローラ01)に記憶する。

ここで、ラダープログラムを宣言するという事は、ラダープログラムの範囲を設定することである。また、回路とはラダープログラムのことをいう。

【0015】

次に、ユーザが実際に入出力接点を変更すると、パソコン側ではその入出力接点をプログラマブルコントローラ側に連絡する必要があるため、パソコン側よりプログラマブルコントローラ側へ入出力接点定義設定(ステップS002)を送信する。

ユーザが前のラダープログラムに戻したい状況が発生したとすると、パソコン側より編集戻し開始信号(ステップS003)をプログラマブルコントローラ側に送信し、プログラマブルコントローラ側では過去のラダープログラムに戻す準備を行う。

そして、プログラマブルコントローラ側では過去のラダープログラムに戻す準備が整うと、プログラマブルコントローラ側よりパソコン側へ戻し許可信号(ステップS004)を送信する。

そして、戻し許可信号をパソコン側で受信すると、パソコン側では過去のラダープログラムをパソコンに接続された表示器の画面に表示して、編集戻し完了信号(ステップS005)をパソコン側よりプログラマブルコントローラ側へ送信する。

以上のステップによりパソコン側とプログラマブルコントローラ側の通信制御は完了する。

【0016】

次に、本発明のハードウェア構成について、図8を用いて説明する。

図8において、10はパソコン、20はパソコン10に接続された表示装置、30はプログラマブルコントローラ、40は通信制御装置、50は中央演算装置(CPU)、60は記憶素子、70は入出力制御回路、80は入出力装置、90はパソコン10とプログラマブルコントローラ30とで通信を行う通信線である。

プログラマブルコントローラ30の中央演算装置50は、CPUで構成され、プログラマブルコントローラ内の記憶装置60、入出力装置70、通信制御回路40を制御している。通信線90はRS-232Cなどのシリアル通信で行っても良い。中央演算装置50は外部との入出力信号を制御し、入出力制御回路70は、入出力装置80と接続されている。

【0017】

次に、図2を用いて、図1におけるパソコン側よりプログラマブルコントローラ側へ送信する回路定義設定(ステップS001)について説明する。

図2はパソコン側の回路定義設定のフローチャートである。

先ず、ユーザは、パソコン側でラダープログラムの機能単位の回路番号を宣言する。

そして、最初に図8におけるプログラマブルコントローラ側の中央演算装置(CPU)50は、図2において開始回路を選択する開始回路選択判定(ステップS200)を行う。

開始回路の選択は、図10に示した回路定義開始ボタンC104を押して、回路番号C1

10

20

30

40

50

01を押すことにより実行される。

【0018】

次に、図2における開始回路番号データを送信する処理(ステップS201)で、プログラマブルコントローラ側の通信制御回路40を経由してプログラマブルコントローラ側30へ送信する。次に、ラダープログラムの終了回路をユーザが宣言する。図2の終了回路データが選択されたかを図8における中央演算装置(CPU)50は終了回路選択判定(ステップS202)を実行する。そして、終了回路が選択されると、終了回路データを終了回路データ送信処理(ステップS203)にて、通信制御回路40を経由してプログラマブルコントローラ側へ送信する。終了回路の選択は、図10に示す回路定義終了ボタンC105を押して、回路番号C101を押して実行される。

10

なお、上記の終了回路データ送信処理(ステップS203)は、通信線90を経由してプログラマブルコントローラ側の通信制御回路40で終了回路データが受信処理されて、中央演算装置(CPU)50にて記憶素子60に終了回路データを記憶する。そして、この一連の回路定義かが完了したことを定義終了判定(ステップS205)にて判定し、定義終了なら回路定義処理(ステップS206)を実行してプログラマブルコントローラ側へ送信する。

【0019】

次に、図3を用いて、プログラマブルコントローラ側の回路定義設定について説明する。

図3において、中央演算装置(CPU)40で、先ず開始回路データを受信したかを判断する(ステップS300)。受信していなければこの処理自体を終了し、受信していれば、次に開始回路データを記憶する処理を実行する(ステップS301)。開始回路データ記憶処理において、図9に示す記憶素子内のデータ保管構造で、開始回路データを開始回路格納データレジスタM100に書き込む。次に、パソコン側より通信制御回路40経由で送信された終了回路データを受信したかを判断する(ステップS302)。終了回路データが受信されたら、図9に示すデータ保管構造で、終了回路格納データレジスタM102に終了回路データ記憶処理(ステップS303)を実行して書き込まれる。そして、その後、中央演算装置(CPU)40は、記憶処理(ステップS304)を実行し、記憶素子60内のアドレスポインタM100の0000番地のアドレスポインタ1を書き込む。この操作は、図6に説明する変更回路をこの記憶素子に書き込むために必要な記憶素子内のアドレスを指し示す必要があるために利用する。

20

30

【0020】

次に、図4を用いて、パソコン側での入出力接点の変更をプログラマブルコントローラ側に送信するフローチャートを説明する。

図4において、先ず、図10に示す開始回路番号C101を選択してから、開始回路選択ボタンC102を押す。そしてその後、終了回路番号C107を選択して、終了回路選択ボタンC103を押すと、開始回路番号データと終了回路番号データが、開始回路/終了回路送信処理(ステップS403)を実行し、パソコン側よりプログラマブルコントローラ側に送信される。

次に、ユーザが入出力接点を変更したかをチェック(ステップS404)する。

40

図10において、例えば、出力コイルの出力接点マークC200をY100からY200に変更すると、その接点であるC201とC202もY200に変わる。

この変更した入出力接点データは、入出力接点を変更されたデータ送信処理(ステップS403)を実行し、プログラマブルコントローラ側へ送信する。

そこで、ユーザが編集を完了するときは、もう一度図10の編集ボタンC100を押す。編集ボタンC100が押されると、編集完了が判断され(ステップS404)、編集完了信号がプログラマブルコントローラ側に送信され(ステップS405)、終了となる。

また、編集が完了していないときは、ユーザがまだ変更したい回路データがあるので、変更対象機能番号データの処理(ステップS401)へ戻る。

【0021】

50

次に、プログラマブルコントローラ側において、変更された入出力接点を判定、受信し、記憶素子60に書き込むフローチャートを説明する。

図5において、先ず、パソコン側より送信された編集開始信号を受信したかを判断する(ステップS500)。編集開始信号を受信すると、次にプログラマブルコントローラ側の中央演算装置(CPU)50は、パソコン側より送信された開始回路番号データと終了回路番号データを受信したかを判断する(ステップS501)。開始回路番号データと終了回路番号データを受信すると、図3に示すフローチャートで登録した開始回路格納レジスタM100と、終了回路格納レジスタM102と同じ記憶素子60内のエリアを割り出し、開始回路/終了回路書き込み処理を中央演算装置(CPU)40で実行する(ステップS502)。次に、変更された入出力接点の変更データを受信し、判断する(ステップS503)。そして、1回前の変更入出力接点として、図9に示す記憶素子60の入出力接点データレジスタM200に、入出力接点変更データ書き込み処理を実行する(ステップS500)ことにより書き込まれる。

また、2回目からの記憶は、同様に記憶素子60内のアドレスポインタM0001の0001番地に1を書き込むと、入出力接点データレジスタM201に変更した入出力接点を書き込まれる。

同様に、n回目も記憶素子60内のアドレスポインタM0001の000n番地に1を書き込むと、入出力接点データレジスタM20nに変更した入出力接点を書き込まれる。

【0022】

次に、パソコン側より送信された編集完了信号を受信したかを判断する(ステップS505)。編集完了信号が受信されたら、一連の入出力接点の変更は終了する。編集完了信号が受信されなかったら、ユーザはまだ変更が発生するので開始回路番号データと終了回路番号データを受信し、判断するステップS501に戻り、動作を継続する。

以上のフローチャートで過去に変更した入出力接点がプログラマブルコントローラの記憶素子60に記憶される。

【0023】

次に、パソコン側での編集戻しについて、図6を用いて説明する。

図6は、今まで記憶してきた過去の入出力接点をユーザの要望により編集戻しをプログラマブルコントローラ側に送信するフローチャートである。

図10における編集戻しボタンC106を押すと、先ず、編集戻し開始信号がパソコン側よりプログラマブルコントローラ側に対して送信される(ステップS600)。これは、これまで変更などを行った過去の回路に編集を戻すことを宣言することである。

【0024】

次に、図10における開始回路番号C101を押して開始回路選択ボタンC102を押す。その後、終了回路番号C107を押して、終了回路選択ボタンC103を押すと、開始回路番号データと終了回路番号データが入力されたかを判断する(ステップS601)。入力されると、開始回路データと終了回路データを、開始回路/終了回路送信処理を実行することでプログラマブルコントローラ側へ送信処理される(ステップS602)。

次に、図10における編集戻しボタンC106を押すと、戻し判定で判断される(ステップS603)。その後、入出力接点データ送信処理(ステップS604)において、図9に示す記憶素子60内のアドレスポインタM001が1の行の入出力接点データレジスタM200の入出力接点データをプログラマブルコントローラ側へ送信する。

次に、プログラマブルコントローラ側より変更戻し許可信号が受信されたかを判断する(ステップS605)。受信していると、過去表示戻し処理を実行する(ステップS606)。すなわち、変更した入出力接点を遡って表示する。この場合、回路番号001から002の入出力接点マークC200とC201がY200よりY100に戻したということになる。しかし、回路番号002の入出力接点はY200のままである。これは回路選択をしていないためである。

以上のことより、ユーザは過去にデバッグしたラダープログラムを選択することができる。ユーザからの編集戻し操作が終了したかを判断する(ステップS607)。これは、図

10

20

30

40

50

10に示す編集ボタンC106を押すと、編集終了となり、編集戻し完了処理を実行（ステップS608）し、プログラマブルコントローラ側に編集戻し完了信号を送信する。編集戻しが完了していなければ、編集戻しを実行していると判断し、ステップS601へ戻る。

【0025】

次に、図7を用いてプログラマブルコントローラ側での編集戻しの処理について説明する。

図7は、過去の入出力接点をユーザの要望により戻すプログラマブルコントローラ側のフローチャートである。

10
20
まず、通信制御回路40で受信し、編集戻し開始信号が受信されたかを中央演算装置（CPU）50で判断する（ステップS700）。編集戻し開始信号が受信されると、次に開始回路/終了回路を受信したかを判断する（ステップS701）。開始回路/終了回路が受信されると、中央演算装置（CPU）50は、開始回路/終了回路のデータ処理を実行（ステップS702）し、記憶素子60内の開始回路格納レジスタM100と終了回路格納レジスタM102と同じ列のデータを検知処理する。次に、戻したい入出力接点データを受信したかを判断する（ステップS703）。受信されると、中央演算装置（CPU）50は、記憶素子60内のアドレスポインタM001が1の行の入出力接点データレジスタM200の入出力接点データを戻し開路実行処理（ステップS704）を実行し、検索処理を行う。また、このステップS704の処理では、図8に示すプログラマブルコントローラ側内の中央演算装置（CPU）50は、入出力制御回路70を制御して変更した入出力接点で入出力装置80を動作させる。その後、変更戻し表示許可信号の送信処理（ステップS705）を実行し、パソコン側へ送信する。また、この処理では過去のラダープログラムの回路を表示する。そして、編集戻し完了信号を受信したら終了となる。受信していないとステップS700に戻る。

【0026】

図10は、ユーザが作成するラダープログラムを機能単位で記憶する回路定義開始ボタンC104と回路定義終了ボタンC105を備えている。

ここで、回路の定義を行い、そのデータをプログラマブルコントローラ側の記憶素子60に登録する。

次に、ユーザはデバッグに移る。その時、編集ボタンC100を押して、編集開始したい回路番号C101を選択して、開始回路選択ボタンC102を押す。

次に、編集の終わり回路番号C107を選択して、終了回路選択ボタンC103を押す。このような作業により、この例では回路番号1と回路番号2が編集できる回路に選択できる。

次に、変更したい入出力接点C200の内容をY100からY201に変更する。

ここで、変更した内容は、プログラマブルコントローラ側の記憶素子に記憶される。

次に、上記内容で過去にデバッグした状態に戻したい時が発生した場合、ユーザは編集戻しボタンC106を押すと、回路番号0001と回路番号0002が過去で使用していた入出力接点C200がY100、C201がY100となり過去の状態に戻すことができる。また、回路番号0003の入出力接点C202は変更されない。従って、各回路の機能単位での過去にデバッグを行ったラダープログラムを再利用できるため、そのラダープログラムのデバッグの時間短縮効果を得られ、過去にデバッグしたラダープログラムをユーザが選択できる。

【0027】

また、図11に示すように、パソコン側の画面構成は縦長方向に表示することも可能である。画面の構成については、図10と同じなので説明は省略する。

【符号の説明】

【0028】

10 パソコン 20 表示装置 30 プログラマブルコントローラ
40 通信制御装置 50 中央演算装置（CPU） 60 記憶素子

10

20

30

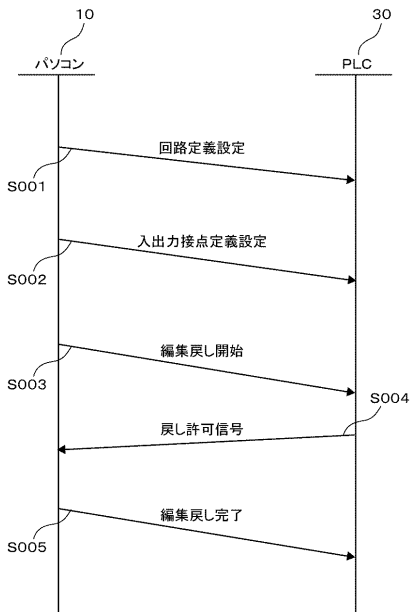
40

50

- 7 0 入出力制御回路 8 0 入出力装置
- M 0 0 1 アドレスポインタ
- M 1 0 0 開始回路格納データレジスタ
- M 1 0 2 終了回路格納データレジスタ
- M 2 0 0 1回前の回路の入出力接点データレジスタ
- M 2 0 1 2回前の回路の入出力接点データレジスタ
- M 2 0 N n回前の回路の入出力接点データレジスタ
- C 1 0 0 編集ボタン
- C 1 0 1 回路番号
- C 1 0 2 開始回路選択ボタン
- C 1 0 3 終了回路選択ボタン
- C 1 0 4 回路定義開始ボタン
- C 1 0 5 回路定義終了ボタン
- C 1 0 6 編集戻しボタン。

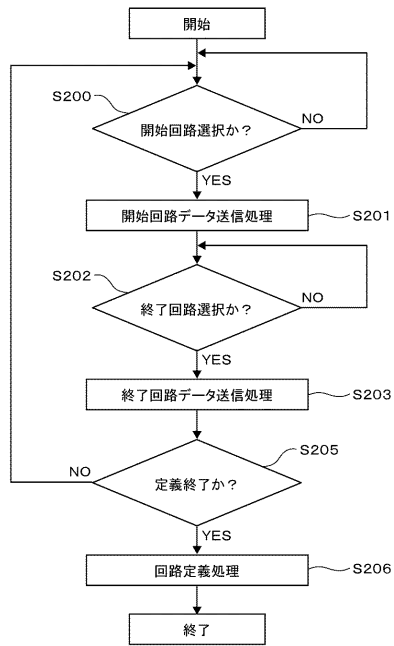
【 図 1 】

図 1



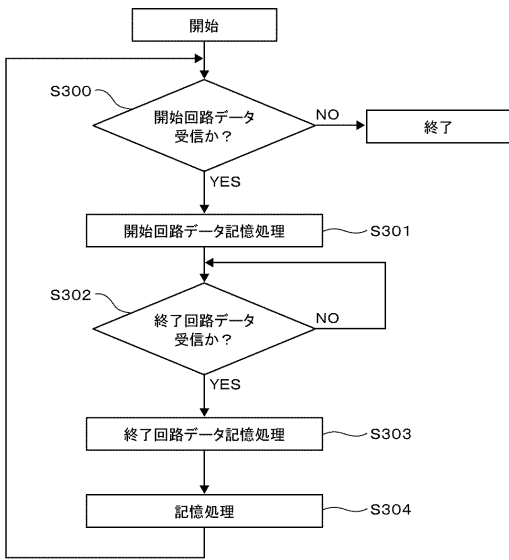
【 図 2 】

図 2



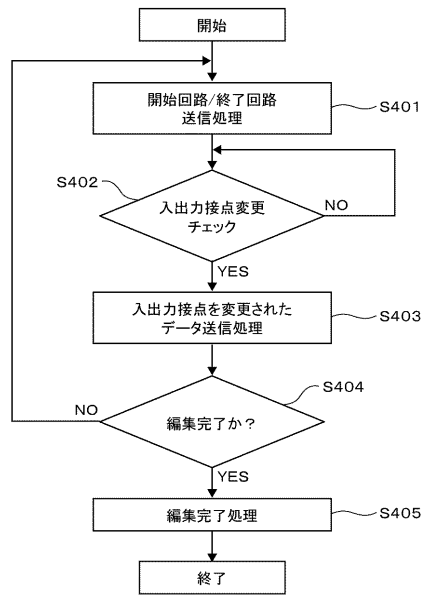
【 図 3 】

図 3



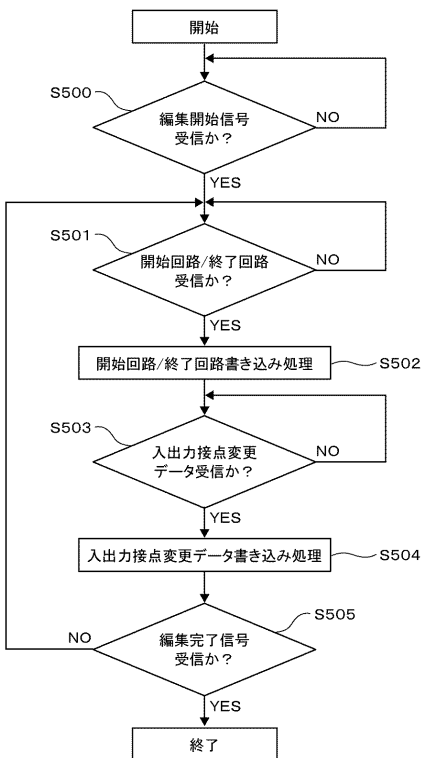
【 図 4 】

図 4



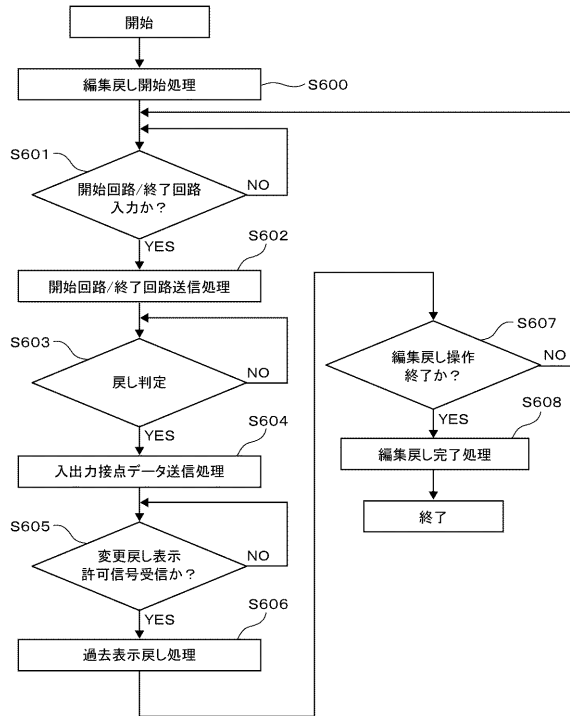
【 図 5 】

図 5



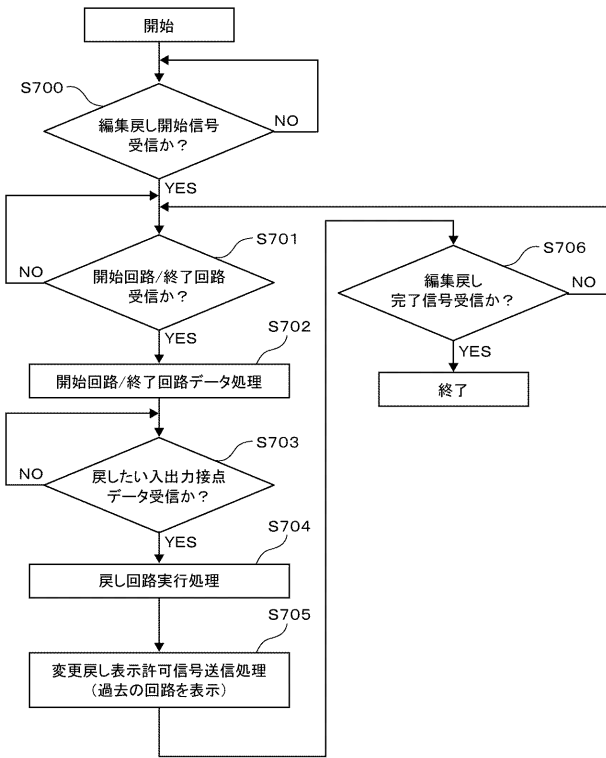
【 図 6 】

図 6



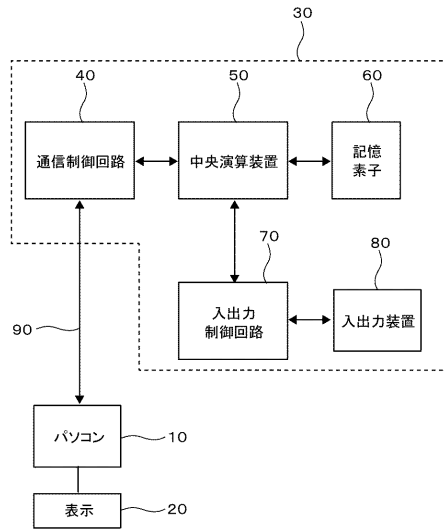
【 図 7 】

図 7



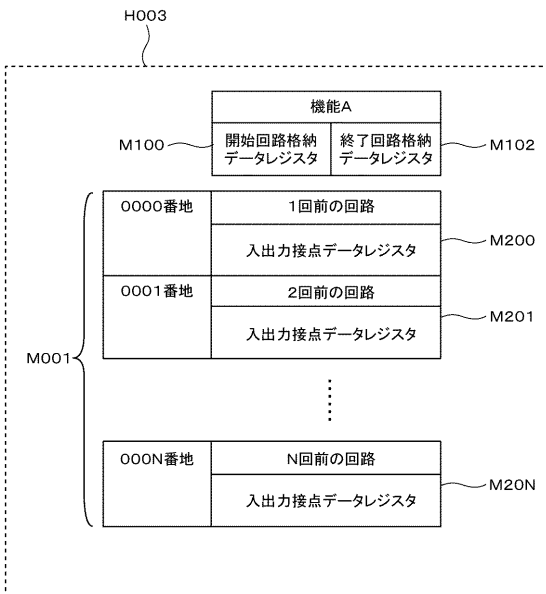
【 図 8 】

図 8



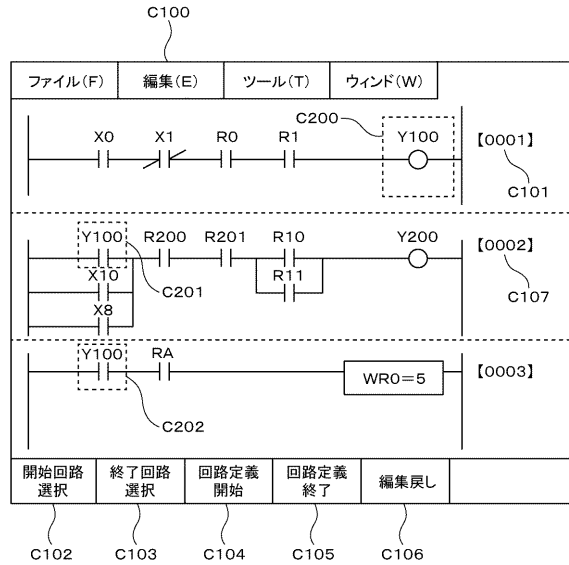
【 図 9 】

図 9



【 図 10 】

図 10



【 図 1 1 】

図 1 1

