

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第3661732号

(P3661732)

(45) 発行日 平成17年6月22日(2005.6.22)

(24) 登録日 平成17年4月1日(2005.4.1)

(51) Int.Cl.⁷

G05B 19/048

G05B 19/05

F I

G05B 19/05

G05B 19/05

D

J

請求項の数 3 (全 9 頁)

(21) 出願番号	特願平9-243795	(73) 特許権者	000006507
(22) 出願日	平成9年9月9日(1997.9.9)		横河電機株式会社
(65) 公開番号	特開平11-85220		東京都武蔵野市中町2丁目9番32号
(43) 公開日	平成11年3月30日(1999.3.30)	(72) 発明者	守 伸有
審査請求日	平成15年1月14日(2003.1.14)		東京都武蔵野市中町2丁目9番32号 横河電機株式会社内
		審査官	森林 克郎
		(56) 参考文献	特開平08-278804(JP, A)
			特開平05-189277(JP, A)
			特開平08-014955(JP, A)
		(58) 調査した分野(Int.Cl. ⁷ , DB名)	
			G05B19/04-19/05、
			G05B23/00-23/02

(54) 【発明の名称】 プログラマブルコントローラ

(57) 【特許請求の範囲】

【請求項1】

プログラマブルコントローラにおいて、
 入出力回路と、
 アプリケーションプログラムが格納される記憶回路と、
 前記入出力回路及び前記記憶回路を制御する制御回路と、
 前記アプリケーションプログラムが1回実行される時間を測定する時間計測用カウンタ回路と、
予め設定された時間範囲に対応して設けられた複数の分布計数用カウンタ回路と、
前記時間計測用カウンタ回路の測定時間が前記予め設定された時間範囲のうちどの時間範囲に該当するかを判断して該当する前記分布計数用カウンタ回路のカウント値をインクリメントする制御回路とから構成されたデバッグ回路と
を備えたことを特徴するプログラマブルコントローラ。

10

【請求項2】

プログラマブルコントローラにおいて、
 入出力回路と、
 アプリケーションプログラムが格納される記憶回路と、
 前記アプリケーションプログラムが1回実行される時間を測定する時間計測用カウンタ回路と、
予め設定された時間範囲に対応して設けられた複数の分布計数用カウンタ回路と、
前記入出力回路及び前記記憶回路を制御すると共に前記時間計測用カウンタ回路の測定

20

時間が前記予め設定された時間範囲のうちどの時間範囲に該当するかを判断して該当する前記分布計数用カウンタ回路のカウント値をインクリメントする制御回路とを備えたことを特徴するプログラマブルコントローラ。

【請求項 3】

接続されたアプリケーションプログラム開発ツールに前記分布計数用カウンタ回路で計数されたカウント値に基づき前記アプリケーションプログラムが 1 回実行される時間の分布図を表示させることを特徴とする

特許請求の範囲請求項 1 及び請求項 2 記載のプログラマブルコントローラ。

【発明の詳細な説明】

【0001】

10

【発明の属する技術分野】

本発明は、プログラマブルコントローラに関し、特にアプリケーションプログラムのスキヤンタイムの分布を測定することが可能なプログラマブルコントローラに関する。

【0002】

【従来の技術】

従来のプログラマブルコントローラは、センサからのデータ収集や機器の制御等を予め格納されたアプリケーションプログラムに基づき実行する。

【0003】

また、プログラムの実行方式として予め格納されたアプリケーションプログラムを何回も連続して実行するスキャン方式がある。

20

【0004】

例えば、センサ等から圧力信号及び温度信号を収集して、圧力信号を温度信号で補償して出力するアプリケーションプログラムを連続して実行することにより、プログラマブルコントローラからは温度補償された圧力信号が定期的に出力されることになる。

【0005】

但し、一般にプログラマブルコントローラでアプリケーションプログラムを 1 回実行する時間（以下、スキヤンタイムと呼ぶ。）は常に一定ではなくプログラムの実行時の条件により変動する。

【0006】

例えば、入力される前記圧力信号や前記温度信号の値の変動や、前記温度補償の演算内容等の諸条件により、アプリケーションプログラムの全命令が実行されるのではなく実行されずにスキップされたり、あるいは、何回も実行される命令が生じたりするのでスキヤンタイムは常時変動する。

30

【0007】

特に、短いスキヤンタイムが要求される場合にはアプリケーションプログラムの作成時にスキヤンタイムの短縮を図る工夫をする必要があり、この場合にはプログラマブルコントローラで実行されるアプリケーションプログラムのスキヤンタイムに関する情報を得る必要がある。

【0008】

一般に、プログラマブルコントローラにはこのような機能が付加されているものがあり、図 5 はこのような従来のプログラマブルコントローラの一例を示す構成ブロック図である。

40

【0009】

図 5 において 1 は入出力回路、2 は制御回路、3 はアプリケーションプログラムが格納される記憶回路、4 はデバッグ回路、100 は入力信号、101 は出力信号である。

【0010】

入力信号 100 は入出力回路 1 を介して制御回路 2 に入力され、制御回路 2 から出力は入出力回路 1 を介して出力信号 101 として外部に出力される。また、制御回路 2 は記憶回路 3 及びデバッグ回路 4 と相互に接続される。

【0011】

50

ここで、図 5 に示す従来例の動作を説明する。制御回路 2 は記憶回路 3 に予め格納されたアプリケーションプログラムの命令を順次読み出して実行する。

【 0 0 1 2 】

一方、デバッグ回路 4 は前記アプリケーションプログラムを 1 回実行するのに要するスキヤンタイムに関する情報を測定する。

【 0 0 1 3 】

具体的には、デバッグ回路 4 では最新スキヤンタイム、スキヤンタイムの最大値及び最小値及び平均スキヤンタイムを測定して記憶する。

【 0 0 1 4 】

また、図 6 はデバッグ回路 4 の具体例を示す構成ブロック図であり、図 6 において 5 は時間計測用カウンタ回路、6 は制御回路、7 は記憶回路、102 はクロック信号である。

【 0 0 1 5 】

クロック信号 102 は時間計測用カウンタ回路 5 に入力され、時間計測用カウンタ回路 5 の出力が制御回路 6 に接続される。また、制御回路 6 と記憶回路 7 が相互に接続される。

【 0 0 1 6 】

ここで、図 6 に示すデバッグ回路 4 の動作を図 7 を用いて説明する。図 7 はデバッグ回路 4 の動作を説明するフロー図である。

【 0 0 1 7 】

図 7 (a) において制御回路 6 は時間計測用カウンタ回路 5 をリセットすると共にカウント動作を開始させ、図 7 (b) においてアプリケーションプログラムの実行の終了を検出する。

【 0 0 1 8 】

図 7 (b) においてアプリケーションプログラムの終了が検出されると制御回路 6 は図 7 (c) において時間計測用カウンタ回路 5 のカウント値を読み出し、図 7 (d) において最新スキヤンタイムとして記憶回路 7 に格納する。

【 0 0 1 9 】

また、図 7 (e) において制御回路 6 は平均スキヤンタイムを計算すると共にその結果を記憶回路 7 に格納する。

【 0 0 2 0 】

図 7 (f) において制御回路 6 は前記カウント値を前回までのスキヤンタイムの最大値と比較して大きいかどうかを判断する。

【 0 0 2 1 】

もし、前記カウント値が前回までの最大値よりも大きければ、制御回路 6 は図 7 (g) において前記カウント値を最大値として記憶回路 7 に格納し、図 7 (j) の処理を行う。

【 0 0 2 2 】

同様に、図 7 (h) において制御回路 6 は前記カウント値を前回までのスキヤンタイムの最小値と比較して小さいかどうかを判断する。

【 0 0 2 3 】

もし、前記カウント値が前回までの最小値よりも小さければ、制御回路 6 は図 7 (i) において前記カウント値を最小値として記憶回路 7 に格納し、図 7 (j) の処理を行う。

【 0 0 2 4 】

最後に、図 7 (j) において制御回路 6 はスキヤンタイム等の測定終了か否かを判断し、再測定であれば図 7 (a) の処理に戻り、測定終了であれば処理を終了する。

【 0 0 2 5 】

また、プログラマブルコントローラとアプリケーションプログラム開発ツールとを接続し、このアプリケーションプログラム開発ツールでデバッグ回路 4 で測定されたスキヤンタイムに関する情報を読み出して表示画面等に表示させることが可能になる。

【 0 0 2 6 】

一般にはプログラマブルコントローラでアプリケーションプログラムを実行させながら前記情報をモニタすることにより、スキヤンタイムの短縮を図る工夫をすることになる。

10

20

30

40

50

【 0 0 2 7 】

この結果、デバッグ回路 4 によりプログラマブルコントローラのアプリケーションプログラムの作成に必要なスキャンタイムに関する情報を得ることが可能になる。

【 0 0 2 8 】

【 発明が解決しようとする課題 】

しかし、図 5 等 に示す従来例ではスキャンタイムの最大値及び最小等の個々の情報は分かるものの、例えば、スキャンタイムの最大値が突発的に生じたのか、或いは、定常的に生じるものなのかを区別することが出来ない。

【 0 0 2 9 】

このため、アプリケーションプログラムのどの部分を修正すれば良いかが分からず、アプリケーションプログラムの修正作業の効率が悪くなってしまうと言った問題点があった。従って本発明が解決しようとする課題は、スキャンタイムの長さの分布状況を得ることが可能なプログラマブルコントローラを実現することにある。

【 0 0 3 0 】

【 課題を解決するための手段 】

このような課題を達成するために、本発明の第 1 では、
プログラマブルコントローラにおいて、
入出力回路と、
アプリケーションプログラムが格納される記憶回路と、
前記入出力回路及び前記記憶回路を制御する制御回路と、
前記アプリケーションプログラムが 1 回実行される時間を測定する時間計測用カウンタ回路と、予め設定された時間範囲に対応して設けられた複数の分布計数用カウンタ回路と、前記時間計測用カウンタ回路の測定時間が前記予め設定された時間範囲のうちどの時間範囲に該当するかを判断して該当する前記分布計数用カウンタ回路のカウント値をインクリメントする制御回路とから構成されたデバッグ回路とを備えたことを特徴するものである。

【 0 0 3 1 】

このような課題を達成するために、本発明の第 2 では、
プログラマブルコントローラにおいて、
入出力回路と、
アプリケーションプログラムが格納される記憶回路と、
前記アプリケーションプログラムが 1 回実行される時間を測定する時間計測用カウンタ回路と、
予め設定された時間範囲に対応して設けられた複数の分布計数用カウンタ回路と、
前記入出力回路及び前記記憶回路を制御すると共に前記時間計測用カウンタ回路の測定時間が前記予め設定された時間範囲のうちどの時間範囲に該当するかを判断して該当する前記分布計数用カウンタ回路のカウント値をインクリメントする制御回路とを備えたことを特徴するものである。

【 0 0 3 2 】

このような課題を達成するために、本発明の第 3 では、
本発明の第 1 及び第 2 において、
接続されたアプリケーションプログラム開発ツールに前記分布計数用カウンタ回路で計数されたカウント値に基づき前記アプリケーションプログラムが 1 回実行される時間の分布図を表示させることを特徴とするものである。

【 0 0 3 4 】

【 発明の実施の形態 】

以下本発明を図面を用いて詳細に説明する。図 1 は本発明に係るプログラマブルコントローラの一実施例を示す構成ブロック図である。

【 0 0 3 5 】

図 1 において 1 , 2 及び 3 は図 5 と同一符号を付してあり , 4 a はデバッグ回路 , 1 0 0

10

20

30

40

50

aは入力信号、101aは出力信号である。

【0036】

入力信号100aは入出力回路1を介して制御回路2に入力され、制御回路1から出力は入出力回路1を介して出力信号101aとして外部に出力される。また、制御回路2は記憶回路3及びデバッグ回路4aと相互に接続される。

【0037】

ここで、図1に示す実施例の動作を説明する。制御回路2は記憶回路3に予め格納されたアプリケーションプログラムの命令を順次読み出して実行する。

【0038】

一方、デバッグ回路4aは前記アプリケーションプログラムを1回実行するのに要するスキャンタイムに関する情報を測定する。 10

【0039】

具体的には、デバッグ回路4aではスキャンタイムを測定して予め設定された時間範囲のうちどの時間範囲に該当するかを判断してその個数を計数する。言い換えれば、スキャンタイムの時間範囲での分布状況を測定して記憶する。

【0040】

また、図2はデバッグ回路4aの具体例を示す構成ブロック図であり、図2において5及び102は図6と同一符号を付してあり、6aは制御回路、8、9、10及び11は分布計数用カウンタ回路である。

【0041】

クロック信号102は時間計測用カウンタ回路5に入力され、時間計測用カウンタ回路5の出力が制御回路6aに接続される。また、制御回路6aと分布計数用カウンタ回路8、9、10及び11が相互に接続される。 20

【0042】

ここで、図2に示すデバッグ回路4aの動作を図3を用いて説明する。図3はデバッグ回路4aの動作を説明するフロー図である。

【0043】

図3(a)において制御回路6aは分布計数カウンタ回路8～11で計数する時間範囲を予め設定する。

【0044】

具体的には、測定したスキャンタイムを"Ts can"とした場合に、 30

$t_3 \quad Ts can < t_4 \quad (1)$

$t_2 \quad Ts can < t_3 \quad (2)$

$t_1 \quad Ts can < t_2 \quad (3)$

$0 \quad Ts can < t_1 \quad (4)$

(但し、 $0 < t_1 < t_2 < t_3 < t_4$)

式(1)の条件を満たせば分布計数用カウンタ8のカウント値をインクリメントし、同様に、式(2)、式(3)及び式(4)の条件を満たせば分布計数用カウンタ9、10及び11のカウント値をインクリメントする様に設定する。

【0045】

図3(b)において制御回路6aは時間計測用カウンタ回路5をリセットすると共にカウント動作を開始させ、図3(c)においてアプリケーションプログラムの実行の終了を検出する。 40

【0046】

図3(c)においてアプリケーションプログラムの終了が検出されると制御回路6aは図3(d)において時間計測用カウンタ回路5のカウント値を読み出す。

【0047】

図3(e)において制御回路6aは前記カウント値が式(1)の条件を満たすかどうかを判断する。

【0048】

もし、式(1)の条件を満たせば図3(f)において制御回路6aは分布計数用カウンタ回路8のカウンタ値をインクリメントし、図3(1)の処理を行う。

【0049】

また、図3(g)において制御回路6aは前記カウンタ値が式(2)の条件を満たすかどうかを判断する。

【0050】

もし、式(2)の条件を満たせば図3(h)において制御回路6aは分布計数用カウンタ回路9のカウンタ値をインクリメントし、図3(1)の処理を行う。

【0051】

さらに、図3(i)において制御回路6aは前記カウンタ値が式(3)の条件を満たすかどうかを判断する。

10

【0052】

もし、式(3)の条件を満たせば図3(j)において制御回路6aは分布計数用カウンタ回路10のカウンタ値をインクリメントし、図3(1)の処理を行う。

【0053】

そして、図3(k)において制御回路6aは分布計数用カウンタ回路11のカウンタ値をインクリメントする。

【0054】

最後に、図3(1)において制御回路6aは測定終了か否かを判断し、再測定であれば図3(b)の処理に戻り、測定終了であれば処理を終了する。

20

【0055】

図4はこのようにデバッグ回路4aで測定したスキャンタイムの分布状況の一例を示す分布図である。

【0056】

具体的には、プログラマブルコントローラとアプリケーションプログラム開発ツールとを接続し、このアプリケーションプログラム開発ツールでデバッグ回路4aで測定されたスキャンタイムの分布状況に関する情報を読み出して表示画面等に分布図で表示させたものである。

【0057】

図4中"イ"、"ロ"、"ハ"及び"ニ"は分布計数用カウンタ回路11、10、9及び8で計数されたカウンタ値であり、図4においてはスキャンタイムは式(2)に示す時間範囲に多く分布していることが分かる。

30

【0058】

また、図4中"ニ"の分布が少ないことからスキャンタイムの最大値は突発的に発生していることも併せて認識することができる。

【0059】

このため、上述の場合にはアプリケーションプログラムの中で定常は実行されない特定の分岐ルーチン等に問題があると特定でき、その部分を修正すれば良いので、アプリケーションプログラムの修正作業の効率が良くなる。

【0060】

40

一方、図4中"ニ"の部分のカウンタ値が大きければ、スキャンタイムの最大値が定常的に生じるものであると認識できるので、アプリケーションプログラムの中で定常的に実行されるルーチンに問題があると特定できる。

【0061】

この結果、デバッグ回路4aによりスキャンタイムの長さを判断して、予め設定された時間範囲のうちの時間範囲に該当するかを判断してその個数を計数することにより、スキャンタイムの長さの分布状況を得ることが可能になる。

【0062】

なお、図2のデバッグ回路4aにおいては説明の簡単のために分布計数用カウンタ回路を4個例示したが、勿論、この個数に限定される訳ではなく、分布計数用カウンタ回路を多

50

く設けることにより、スキャンタイムのより詳細な分布状況を得ることができる。

【 0 0 6 3 】

また、図 2 のデバッグ回路 4 a では説明の簡単のために制御回路 6 a を図 1 中の制御回路 2 とは別個に例示したが、制御回路 6 a における処理内容を制御回路 2 に処理させても構わない。

【 0 0 6 4 】

また、アプリケーションプログラム開発ツールとしては一般にはパーソナルコンピュータやワークステーション等の制御装置が用いられる。

【 0 0 6 5 】

また、スキャンタイムがあるの時間以上になった場合にのみ発生する現象の特定に関して 10
は、スキャンタイムがあるの時間以上になる頻度と前記現象の発生頻度の関係を調べる必要があり、本実施例によればこれらの関係を調べる事が可能になる。

【 0 0 6 6 】

【 発明の効果 】

以上説明したことから明らかなように、本発明によれば次のような効果がある。
デバッグ回路によりスキャンタイムの長さを判断して、予め設定された時間範囲のうちどの時間範囲に該当するかを判断してその個数を計数することにより、スキャンタイムの長さの分布状況を得ることが可能なプログラマブルコントローラが実現できる。

【 図面の簡単な説明 】

【 図 1 】 本発明に係るプログラマブルコントローラの一実施例を示す構成ブロック図である。 20

【 図 2 】 デバッグ回路の具体例を示す構成ブロック図である。

【 図 3 】 デバッグ回路の動作を説明するフロー図である。

【 図 4 】 スキャンタイムの分布状況の一例を示す分布図である。

【 図 5 】 従来のプログラマブルコントローラの一例を示す構成ブロック図である。

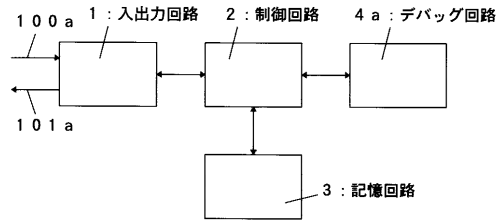
【 図 6 】 デバッグ回路の具体例を示す構成ブロック図である。

【 図 7 】 デバッグ回路の動作を説明するフロー図である。

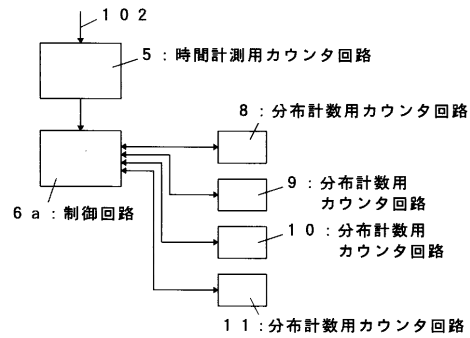
【 符号の説明 】

- 1 入出力回路
- 2 , 6 , 6 a 制御回路 30
- 3 , 7 記憶回路
- 4 , 4 a デバッグ回路
- 5 時間計測用カウンタ回路
- 8 , 9 , 1 0 , 1 1 分布計数用カウンタ回路
- 1 0 0 , 1 0 0 a 入力信号
- 1 0 1 , 1 0 1 a 出力信号
- 1 0 2 クロック信号

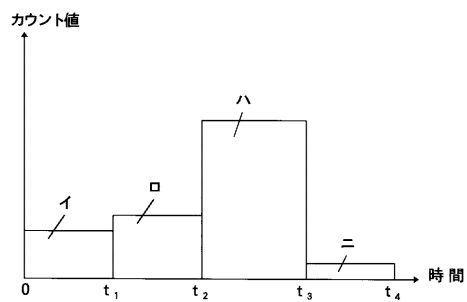
【図 1】



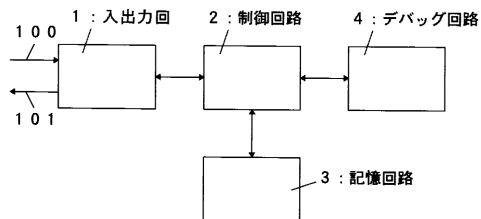
【図 2】



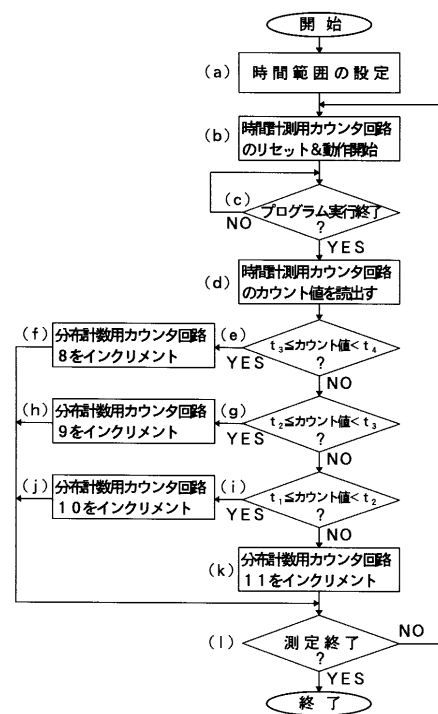
【図 4】



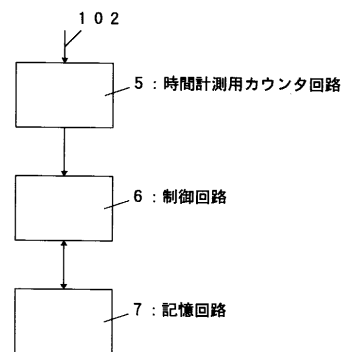
【図 5】



【図 3】



【図 6】



【図 7】

