

(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2016-81300

(P2016-81300A)

(43) 公開日 平成28年5月16日(2016.5.16)

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

審査請求 未請求 請求項の数 27 O L (全 25 頁)

(21) 出願番号 特願2014-211982 (P2014-211982)
(22) 出願日 平成26年10月16日 (2014.10.16)(71) 出願人 000129253
株式会社キーエンス
大阪府大阪市東淀川区東中島1丁目3番1
4号
(74) 代理人 100076428
弁理士 大塚 康德
(74) 代理人 100112508
弁理士 高柳 司郎
(74) 代理人 100115071
弁理士 大塚 康弘
(74) 代理人 100116894
弁理士 木村 秀二
(74) 代理人 100131886
弁理士 坂本 隆志

最終頁に続く

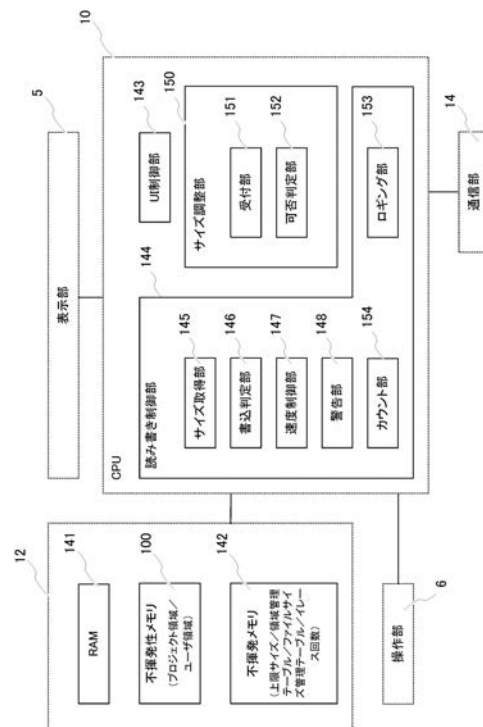
(54) 【発明の名称】 プログラマブル・ロジック・コントローラ、プログラム作成支援装置、制御方法およびプログラム

(57) 【要約】 (修正有)

【課題】 PLCにおいてユーザプログラムが記憶される記憶手段の未使用領域を有効活用する。

【解決手段】 不揮発性メモリ100はユーザプログラムを記憶し、その空き領域にユーザデータを記憶する。不揮発性メモリ100において、ユーザプログラムを記憶するために許容されるプロジェクト領域の上限サイズと、ユーザデータを記憶するために許容されるユーザ領域の上限サイズとのうちいずれかを確定するための情報が不揮発性メモリ142の領域管理テーブルなどに保持される。読み書き制御部144は、不揮発性メモリ142や領域管理テーブルに保持されている情報によって確定する上限サイズを超えない範囲でユーザプログラムまたはユーザデータを不揮発性メモリ100に書き込む。

【選択図】 図12



【特許請求の範囲】**【請求項 1】**

周期的に繰り返し実行されるユーザプログラムとユーザデータとを記憶する記憶手段と

、

前記記憶手段において、前記ユーザプログラムを記憶するために許容される記憶領域の上限サイズと、前記ユーザデータを記憶するために許容される記憶領域の上限サイズとのうちいずれかを確定するための情報を保持する保持手段と、

前記保持手段に保持されている前記情報によって確定する上限サイズを超えない範囲で前記ユーザプログラムまたは前記ユーザデータを前記記憶手段に書き込む書き込み手段とを有することを特徴とするプログラマブル・ロジック・コントローラ。

10

【請求項 2】

前記保持手段は、前記情報として、前記記憶手段において前記ユーザプログラムを格納可能な第 1 記憶領域と、前記ユーザデータを格納可能な第 2 記憶領域との境界となる記憶位置を保持していることを特徴とする請求項 1 に記載のプログラマブル・ロジック・コントローラ。

【請求項 3】

前記第 1 記憶領域と前記第 2 記憶領域との境界となる記憶位置の変更要求を受け付ける受付手段と、

前記受付手段により受け付けられた変更要求にしたがって前記前記第 1 記憶領域と前記第 2 記憶領域との境界となる記憶位置を変更する変更手段と

20

をさらに有することを特徴とする請求項 2 に記載のプログラマブル・ロジック・コントローラ。

【請求項 4】

前記保持手段は、前記ユーザプログラムを記憶するために許容される記憶領域の上限サイズと、前記ユーザデータを記憶するために許容される記憶領域の上限サイズとの両方を保持していることを特徴とする請求項 1 に記載のプログラマブル・ロジック・コントローラ。

【請求項 5】

前記記憶手段の全記憶領域のうち、前記ユーザプログラムを記憶するために許容される第 1 記憶領域以外の残りの第 2 記憶領域内に前記ユーザデータが記憶されることを特徴とする請求項 1 に記載のプログラマブル・ロジック・コントローラ。

30

【請求項 6】

前記記憶手段において前記ユーザプログラムを記憶するために消費されている記憶領域の消費サイズを取得する取得手段と、

前記記憶手段における全記憶領域のサイズから前記ユーザプログラムについての消費サイズを減算して得られる残りの記憶領域のサイズを超えない範囲で前記ユーザデータの記憶領域のサイズを設定する設定手段と

をさらに有することを特徴とする請求項 1 に記載のプログラマブル・ロジック・コントローラ。

【請求項 7】

40

前記記憶手段において前記ユーザプログラムと前記ユーザデータを記憶するために実際に消費されている記憶領域の消費サイズを取得する取得手段と、

前記消費サイズと前記ユーザデータを記憶するための記憶領域についての拡張サイズとの和が、前記記憶手段における全記憶領域のサイズ以下であるかどうかを判定する判定手段と、

前記和が前記全記憶領域のサイズ以下である場合に、前記ユーザデータの記憶領域のサイズを前記拡張サイズに応じて拡張する拡張手段と

をさらに有することを特徴とする請求項 1 に記載のプログラマブル・ロジック・コントローラ。

【請求項 8】

50

前記保持手段に保持されている前記ユーザプログラムを記憶するための第 1 記憶領域の上限サイズと、前記ユーザデータを記憶するための第 2 記憶領域の上限サイズとの変更を受け付ける受付手段と、

前記記憶手段において前記ユーザプログラムと前記ユーザデータを記憶するために実際に消費されている記憶領域の消費サイズを取得する取得手段と、

前記受付手段によって受け付けられた変更要求に基づく前記第 1 記憶領域の変更後の上限サイズを前記ユーザプログラムについての前記消費サイズが超えず、かつ、前記受付手段によって受け付けられた変更要求に基づく前記第 2 記憶領域の変更後の上限サイズを前記ユーザデータについての前記消費サイズが超えないかどうかに応じて前記第 1 記憶領域の上限サイズと前記第 2 記憶領域の上限サイズとの変更の可否を判定する判定手段と、

前記判定手段が前記第 1 記憶領域の上限サイズと前記第 2 記憶領域の上限サイズとの変更が可能と判定すると、前記第 1 記憶領域の上限サイズおよび前記第 2 記憶領域の上限サイズの変更を反映させる領域設定手段と

をさらに有することを特徴とする請求項 1 に記載のプログラマブル・ロジック・コントローラ。

【請求項 9】

前記保持手段に保持されている前記ユーザプログラムを記憶するための第 1 記憶領域の上限サイズと、前記ユーザデータを記憶するための第 2 記憶領域の上限サイズの変更を受け付ける受付手段と、

前記記憶手段への書き込み対象となっているユーザプログラムとユーザデータを記憶するために必要となる記憶領域の必要サイズを取得する取得手段と、

前記受付手段によって受け付けられた変更要求に基づく前記第 1 記憶領域の変更後の上限サイズを前記ユーザプログラムについての前記必要サイズが超えず、かつ、前記受付手段によって受け付けられた変更要求に基づく前記第 2 記憶領域の変更後の上限サイズを前記ユーザデータについての前記必要サイズが超えないかどうかに応じて前記第 1 記憶領域の上限サイズと前記第 2 記憶領域の上限サイズとの変更の可否を判定する判定手段と、

前記判定手段が前記第 1 記憶領域の上限サイズと前記第 2 記憶領域の上限サイズとの変更が可能と判定すると、前記第 1 記憶領域の上限サイズおよび前記第 2 記憶領域の上限サイズの変更を反映させる領域設定手段と

をさらに有することを特徴とする請求項 1 に記載のプログラマブル・ロジック・コントローラ。

【請求項 10】

前記判定手段は、前記記憶手段へのユーザプログラムとユーザデータの書き込みが要求されたときに判定処理を実行することを特徴とする請求項 9 に記載のプログラマブル・ロジック・コントローラ。

【請求項 11】

前記受付手段は、前記ユーザプログラムが実行されているときにも前記第 1 記憶領域の上限サイズと前記第 2 記憶領域の上限サイズの変更を受け付けることを特徴とする請求項 9 または 10 に記載のプログラマブル・ロジック・コントローラ。

【請求項 12】

前記受付手段は、前記プログラマブル・ロジック・コントローラに前記ユーザプログラムを転送するプログラム作成支援装置から送信される前記変更要求を受け付けることを特徴とする請求項 9 ないし 11 のいずれか 1 項に記載のプログラマブル・ロジック・コントローラ。

【請求項 13】

前記書き込み手段は、前記ユーザプログラムを実行することで生成されるロギングデータを一定のサンプリング数を単位として前記記憶手段に書き込むことを特徴とする請求項 1 ないし 12 のいずれか 1 項に記載のプログラマブル・ロジック・コントローラ。

【請求項 14】

前記書き込み手段は、前記記憶手段への情報の書き込み速度を前記記憶手段の性能上の

10

20

30

40

50

上限速度よりも低速な書き込み速度に制限することを特徴とする請求項 1 ないし 13 のいずれか 1 項に記載のプログラマブル・ロジック・コントローラ。

【請求項 15】

前記書き込み手段は、前記記憶手段への情報の書き込み速度を当該情報の種別に応じて制限することを特徴とする請求項 14 に記載のプログラマブル・ロジック・コントローラ。

【請求項 16】

前記書き込み手段は、前記ユーザプログラムの実行によって発生する情報の書き込み速度を制限し、前記プログラマブル・ロジック・コントローラの起動時、メンテナンス時またはプログラム作成支援装置からのユーザプログラムまたはユーザデータの転送時には前記書き込み速度を制限しないことを特徴とする請求項 15 に記載のプログラマブル・ロジック・コントローラ。

【請求項 17】

前記記憶手段のセクタごとイレース回数または書き換え回数をカウントするカウント手段と、

前記記憶手段におけるいずれかセクタについての前記カウント手段によるカウント値が閾値を超えると警報を出力する出力手段と

をさらに有することを特徴とする請求項 1 ないし 16 のいずれか 1 項に記載のプログラマブル・ロジック・コントローラ。

【請求項 18】

ユーザプログラムとユーザデータとを記憶する記憶手段を備えたプログラマブル・ロジック・コントローラに接続され、前記ユーザプログラムを作成するプログラム作成支援装置であって、

前記記憶手段において前記ユーザプログラムを記憶するために許容される第 1 記憶領域の上限サイズと、前記ユーザデータを記憶するために許容される第 2 記憶領域の上限サイズとのうちいずれかを設定する設定手段

を有することを特徴とするプログラム作成支援装置。

【請求項 19】

前記第 1 記憶領域の上限サイズと前記第 2 記憶領域の上限サイズとを表示する表示手段と、

前記表示手段に表示された前記第 1 記憶領域の上限サイズと前記第 2 記憶領域の上限サイズとを調整する調整手段と

をさらに有することを特徴とする請求項 18 に記載のプログラム作成支援装置。

【請求項 20】

前記記憶手段において前記ユーザプログラムと前記ユーザデータを記憶するために実際に消費されている記憶領域の消費サイズを取得する取得手段をさらに有し、

前記表示手段は、

前記取得手段により取得された前記ユーザプログラムについての消費サイズと前記ユーザデータについての消費サイズとをさらに表示することを特徴とする請求項 19 に記載のプログラム作成支援装置。

【請求項 21】

前記調整手段によって調整された前記第 1 記憶領域の上限サイズを前記ユーザプログラムについての前記消費サイズが超えず、かつ、前記調整手段によって調整された前記第 2 記憶領域の上限サイズを前記ユーザデータについての前記消費サイズが超えないかどうかに応じて前記第 1 記憶領域の上限サイズおよび前記第 2 記憶領域の上限サイズの調整を前記プログラマブル・ロジック・コントローラに反映させることが可能かどうかを判定する判定手段と、

前記設定手段は、前記判定手段が前記第 1 記憶領域の上限サイズおよび前記第 2 記憶領域の上限サイズの調整を前記プログラマブル・ロジック・コントローラに反映させることが可能と判定すると、前記第 1 記憶領域の上限サイズおよび前記第 2 記憶領域の上限サイ

10

20

30

40

50

ズの調整を前記プログラマブル・ロジック・コントローラに反映させることを特徴とする請求項 20 に記載のプログラム作成支援装置。

【請求項 22】

前記表示手段は、前記ユーザデータについての消費サイズをバーグラフとして表示し、前記調整手段は、前記第 2 記憶領域の上限サイズを前記バーグラフに沿って調整することを特徴とする請求項 19 ないし 21 のいずれか 1 項に記載のプログラム作成支援装置。

【請求項 23】

前記表示手段は、前記ユーザデータについての消費サイズと、前記ユーザプログラムを含むプロジェクトデータの消費サイズとをそれぞれバーグラフとして表示し、

前記調整手段は、前記第 2 記憶領域の上限サイズを前記バーグラフに沿って調整することを特徴とする請求項 19 ないし 21 のいずれか 1 項に記載のプログラム作成支援装置。

【請求項 24】

周期的に繰り返し実行されるユーザプログラムとユーザデータとを記憶する記憶手段を有するプログラマブル・ロジック・コントローラの制御方法であって、

前記記憶手段において、前記ユーザプログラムを記憶するために許容される記憶領域の上限サイズと、前記ユーザデータを記憶するために許容される記憶領域の上限サイズとのうちいずれかを確定するための情報を保持手段に保持させる保持工程と、

前記保持手段に保持されている前記情報によって確定する上限サイズを超えない範囲で前記ユーザプログラムまたは前記ユーザデータを前記記憶手段に書き込む書き込み工程とを有することを特徴とするプログラマブル・ロジック・コントローラの制御方法。

【請求項 25】

周期的に繰り返し実行されるユーザプログラムとユーザデータとを記憶する記憶手段を備えたプログラマブル・ロジック・コントローラに接続され、前記ユーザプログラムを作成するプログラム作成支援装置の制御方法であって、

前記記憶手段において前記ユーザプログラムを記憶するために許容される第 1 記憶領域の上限サイズと、前記ユーザデータを記憶するために許容される第 2 記憶領域の上限サイズとのうちいずれかを設定する設定工程を有することを特徴とするプログラム作成支援装置の制御方法。

【請求項 26】

請求項 24 に記載のプログラマブル・ロジック・コントローラの制御方法の各工程を当該プログラマブル・ロジック・コントローラに実行させるプログラム。

【請求項 27】

請求項 25 に記載のプログラム作成支援装置の制御方法の各工程を当該プログラム作成支援装置に実行させるプログラム。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、プログラマブル・ロジック・コントローラ、プログラム作成支援装置、制御方法およびプログラムに関する。

【背景技術】

【0002】

特許文献 1 が示すようにプログラマブル・ロジック・コントローラ (PLC) の CPU ユニットには、ラダープログラム等のユーザプログラムを記憶するプログラムメモリが設けられている。CPU はユーザプログラムを実行することでデータの読み書きを行いながら拡張ユニットなどを制御するが、制御の高速化の観点からデータは高速かつ不揮発性のメモリに格納される。プログラムメモリは、ユーザプログラムが書き込まれるだけであるため低速なメモリで十分である。このような理由からユーザプログラムが格納されるメモリとユーザデータが格納されるメモリは別々に用意されていた。

【先行技術文献】

【特許文献】

10

20

30

40

50

【 0 0 0 3 】

【特許文献 1】特開 2 0 1 1 - 0 3 4 3 7 8 号公報

【発明の概要】

【発明が解決しようとする課題】

【 0 0 0 4 】

ところで、不揮発性メモリの高速化および大容量化に伴い、上述したプログラムメモリの大容量化が進みつつある。その一方で、ユーザプログラムのサイズの増加はそれほど進んではない。よって、プログラムメモリには広大な未使用領域が出現することがある。したがって、プログラムメモリの未使用領域の一部にユーザデータを記憶できるようにすれば、プログラムメモリとユーザデータ用のメモリとを個別に用意しなくて済むようになる。ただし、1つのメモリ内でプログラムメモリとユーザデータを自由に記憶させるようにすると、ユーザプログラムを改変したときに、ユーザプログラムのサイズが増大し、ユーザプログラムをプログラムメモリに書き込めなくなってしまうおそれがある。これはユーザがユーザデータを任意に作成できるため、プログラムメモリの大半をユーザデータが占めてしまう可能性があるからである。よって、プログラムメモリをユーザプログラムとユーザデータとで共有する場合は、予めユーザプログラムの上限サイズとユーザデータの上限サイズとのうち少なくとも一方を設定すればよいだろう。

10

【 0 0 0 5 】

そこで、本発明は、P L Cにおいてユーザプログラムが記憶される記憶手段の未使用領域を有効活用することを目的とする。

20

【課題を解決するための手段】

【 0 0 0 6 】

本発明は、たとえば、

周期的に繰り返し実行されるユーザプログラムとユーザデータとを記憶する記憶手段と

、
前記記憶手段において、前記ユーザプログラムを記憶するために許容される記憶領域の上限サイズと、前記ユーザデータを記憶するために許容される記憶領域の上限サイズとのうちいずれかを確定するための情報を保持する保持手段と、

前記保持手段に保持されている前記情報によって確定する上限サイズを超えない範囲で前記ユーザプログラムまたは前記ユーザデータを前記記憶手段に書き込む書き込み手段とを有することを特徴とするプログラマブル・ロジック・コントローラを提供する。

30

【発明の効果】

【 0 0 0 7 】

本発明によれば、P L Cにおいてユーザプログラムが記憶される記憶手段の未使用領域にユーザデータを格納することで、未使用領域を有効活用することが可能となる。また、ユーザプログラムやユーザデータの上限サイズを超えない範囲でユーザプログラムやユーザデータを記憶手段に書き込むようにしたので、ユーザプログラムの改変時の問題も緩和できるようになる。

【図面の簡単な説明】

【 0 0 0 8 】

40

【図 1】P L Cシステムの一例を示す図

【図 2】ユーザプログラムの一例を示す図

【図 3】プログラム作成支援装置の一例を示す図

【図 4】P L Cの一例を示す図

【図 5】スキャンタイムを説明するための図

【図 6】ベースレス・ビルディングタイプのP L Cの一例を示す図

【図 7】複数の記憶領域の上限サイズを管理する手法の一例を示す図

【図 8】上限サイズを設定するためのユーザインタフェースの一例を示す図

【図 9】上限サイズを設定するためのユーザインタフェースの一例を示す図

【図 10】上限サイズを設定するためのユーザインタフェースの一例を示す図

50

【図 1 1】管理テーブルの一例を示す図

【図 1 2】基本ユニットの C P U の機能の一例を示す図

【図 1 3】設定プログラムによって実現される機能の一例を示す図

【図 1 4】上限サイズの設定処理を示すフローチャート

【発明を実施するための形態】

【0009】

以下に本発明の一実施形態を示す。以下で説明される個別の実施形態は、本発明の上位概念、中位概念および下位概念など種々の概念を理解するために役立つであろう。また、本発明の技術的範囲は、特許請求の範囲によって確定されるのであって、以下の個別の実施形態によって限定されるわけではない。

10

【0010】

はじめにプログラマブル・ロジック・コントローラ（P L C、単にプログラマブルコントローラと呼ばれてもよい）を当業者にとってよりよく理解できるようにするために、一般的な P L C の構成とその動作について説明する。

【0011】

図 1 は、本発明の実施の形態によるプログラマブル・ロジック・コントローラシステムの一構成例を示す概念図である。図 1 に示すように、このシステムは、ラダープログラムなどのユーザプログラムの編集を行うためのプログラム作成支援装置 1 と、工場等に設置される各種制御装置を統括的に制御するための P L C（プログラマブル・ロジック・コントローラ）2 とを備えている。P L C 2 は、C P U が内蔵された基本ユニット 3 と 1 つないし複数の拡張ユニット 4 を備えている。基本ユニット 3 に対して 1 つないし複数の拡張ユニット 4 が着脱可能となっている。基本ユニット 3 は基本ユニットと呼ばれることもある。

20

【0012】

基本ユニット 3 には、表示部 5 及び操作部 6 が備えられている。表示部 5 には、基本ユニット 3 に取り付けられている各拡張ユニット 4 の動作状況などを表示することができ、表示部 5 の表示内容は、操作部 6 を操作することにより切り替えることができる。表示部 5 には、通常、P L C 2 内のデバイスの現在値（デバイス値）や P L C 2 内で生じたエラー情報などが表示される。なお、デバイスとは、デバイス値を格納するために設けられたメモリ上の領域を指す名称である。デバイス値とは、入力機器からの入力状態、出力機器への出力状態およびユーザプログラム上で設定される内部リレー（補助リレー）、タイマー、カウンタ、データメモリ等の状態を示す情報である。

30

【0013】

拡張ユニット 4 は、P L C 2 の機能を拡張するために用意されており、基本ユニット 3 に対して側方から取り付けられる。1 つ目の拡張ユニット 4 は、基本ユニット 3 に対して側方から直接的に取り付けられる。2 つ目以降の拡張ユニット 4 は、既に取り付けられている拡張ユニット 4 に対して、側方から直列的に取り付けられる。たとえば、基本ユニット 3 の右側面と拡張ユニット 4 の左側面とが連結面になっている。同様に、1 つ目の拡張ユニット 4 の右側面の形状等は基本ユニット 3 の右側面とほぼ同じであるため、1 つ目の拡張ユニット 4 の右側面に 2 つ目の拡張ユニット 4 の左側面が連結される。このような連結方式は、数珠つなぎ方式とかデジチェーン方式と呼ばれてもよい。各連結面にはコネクタが設けられており、通信や電力供給を行うためのバスもコネクタを介して連結される。このようにして、基本ユニット 3 と複数の拡張ユニット 4 が直列的に取り付けられると、各拡張ユニット 4 内に備えられた配線（例：バス）を介して、各拡張ユニット 4 が基本ユニット 3 に対して通信可能に接続される。各拡張ユニット 4 には、その拡張ユニット 4 の機能に対応する被制御装置 1 6（図 4）が接続され、これにより、各被制御装置 1 6 が拡張ユニット 4 を介して基本ユニット 3 に接続される。被制御装置 1 6 には、センサなどの入力装置や、アクチュエータなどの出力装置が含まれる。

40

【0014】

プログラム作成支援装置 1 は、たとえば、携帯可能ないわゆるノートタイプやタブレッ

50

トタイプのパーソナルコンピュータであって、表示部 7 及び操作部 8 が備えられている。P L C 2 を制御するためのユーザプログラムの一例であるラダープログラムは、プログラム作成支援装置 1 を用いて作成され、その作成されたラダープログラムは、プログラム作成支援装置 1 内でニモニックコードに変換される。そして、プログラム作成支援装置 1 を、U S B (Universal Serial Bus) などの通信ケーブル 9 を介して P L C 2 の基本ユニット 3 に接続し、ニモニックコードに変換されたラダープログラムをプログラム作成支援装置 1 から基本ユニット 3 に送ると、そのラダープログラムが基本ユニット 3 内でマシンコードに変換され、基本ユニット 3 に備えられたメモリ内に記憶される。

【 0 0 1 5 】

なお、図 1 では示していないが、プログラム作成支援装置 1 の操作部 8 には、プログラム作成支援装置 1 に接続されたマウスなどのポインティングデバイスが含まれていてもよい。また、プログラム作成支援装置 1 は、U S B 以外の他の通信ケーブル 9 を介して、P L C 2 の基本ユニット 3 に対して着脱可能に接続されるような構成であってもよい。

【 0 0 1 6 】

図 2 は、ラダープログラムの作成時にプログラム作成支援装置 1 の表示部 7 に表示されるラダー図 1 7 の一例を示す図である。図 2 に示すように、P L C 2 を制御するためのラダープログラムは、プログラム作成支援装置 1 の表示部 7 にマトリックス状に表示される複数のセル 1 8 内に仮想デバイスのシンボル 1 9 を適宜配置して、視覚的なリレー回路を表すラダー図 1 7 を構築することにより作成される。

【 0 0 1 7 】

ラダー図 1 7 には、たとえば、1 0 列 × N 行 (N は任意の自然数) のセル 1 8 が配置されている。そして、各行のセル 1 8 内に、図 2 に示す左側から右側に向かって、時系列的に仮想デバイスのシンボル 1 9 を適宜配置することにより、視覚的なリレー回路を作成することができる。作成されるリレー回路は、1 行で表される直列的なリレー回路であってもよいし、複数行に並列的に表されたリレー回路を互いに結合することにより作成された、並列的なリレー回路であってもよい。

【 0 0 1 8 】

図 2 に示すリレー回路は、入力装置からの入力信号に基づいてオン / オフされる 3 つの仮想デバイス (以下、「入力デバイス」と呼ぶ。) のシンボル 1 9 a , 1 9 b , 1 9 c と、出力装置の動作を制御するためにオン / オフされる仮想デバイス (以下、「出力デバイス」と呼ぶ。) のシンボル 1 9 d とが適宜結合されることにより構成されている。

【 0 0 1 9 】

各入力デバイスのシンボル 1 9 a , 1 9 b , 1 9 c の上方に表示されている文字 (「 R 0 0 0 1 」 , 「 R 0 0 0 2 」 及び 「 R 0 0 0 3 」) は、その入力デバイスのデバイス名 (アドレス名) 2 1 を表している。各入力デバイスのシンボル 1 9 a , 1 9 b , 1 9 c の下方に表示されている文字 (「 フラグ 1 」 , 「 フラグ 2 」 及び 「 フラグ 3 」) は、その入力デバイスに対応付けられたデバイスコメント 2 2 を表している。出力デバイスのシンボル 1 9 d の上方に表示されている文字 (「 原点復帰 」) は、その出力デバイスの機能を表す文字列からなるラベル 2 3 である。

【 0 0 2 0 】

図 2 に示す例では、デバイス名 「 R 0 0 0 1 」 及び 「 R 0 0 0 2 」 にそれぞれ対応する 2 つの入力デバイスのシンボル 1 9 a , 1 9 b が直列的に結合されることにより、AND 回路が構成されている。また、これらの 2 つの入力デバイスのシンボル 1 9 a , 1 9 b からなる AND 回路に対して、デバイス名 「 R 0 0 0 3 」 に対応する入力デバイスのシンボル 1 9 c が並列的に結合されることにより、OR 回路が構成されている。すなわち、このリレー回路では、2 つのシンボル 1 9 a , 1 9 b に対応する入力デバイスがいずれもオンした場合、又は、シンボル 1 9 c に対応する入力デバイスがオンした場合にのみ、シンボル 1 9 d に対応する出力デバイスがオンされるようになっている。

【 0 0 2 1 】

図 3 は、図 1 のプログラム作成支援装置 1 の電氣的構成について説明するためのブロッ

10

20

30

40

50

ク図である。図 3 に示すように、プログラム作成支援装置 1 には、CPU 24、表示部 7、操作部 8、記憶装置 25 及び通信部 26 が備えられている。表示部 7、操作部 8、記憶装置 25 及び通信部 26 は、それぞれ CPU 24 に対して電氣的に接続されている。記憶装置 25 は、少なくとも RAM を含む構成であり、ラダープログラム記憶部 25a と、編集ソフト記憶部 25b とを備えている。

【0022】

ユーザは、編集ソフト記憶部 25b に記憶されている編集ソフトを CPU 24 に実行させて、操作部 8 を通じてラダープログラムを編集する。ここで、ラダープログラムの編集には、ラダープログラムの作成及び変更が含まれる。編集ソフトを用いて作成されたラダープログラムは、ラダープログラム記憶部 25a に記憶される。また、ユーザは、必要に応じてラダープログラム記憶部 25a に記憶されているラダープログラムを読み出し、そのラダープログラムを、編集ソフトを用いて変更することができる。通信部 26 は、通信ケーブル 9 を介してプログラム作成支援装置 1 を基本ユニット 3 に通信可能に接続するためのものである。

【0023】

図 4 は、PLC 2 の電氣的構成について説明するためのブロック図である。図 4 に示すように、基本ユニット 3 には、CPU 10、表示部 5、操作部 6、記憶装置 12 及び通信部 14 が備えられている。表示部 5、操作部 6、記憶装置 12、及び通信部 14 は、それぞれ CPU 10 に電氣的に接続されている。記憶装置 12 は、RAM や ROM、メモリカードなどを含んでもよく、ラダープログラムなどが記憶される。記憶装置 12 には、プログラム作成支援装置 1 から入力されたラダープログラムやユーザデータが上書きして記憶される。

【0024】

図 5 は、本発明の実施の形態に係るプログラマブルコントローラの基本ユニット 3 でのスキャンタイムの構成を示す模式図である。図 5 が示すように 1 つのスキャンタイム T は、入出力のリフレッシュを行うためのユニット間通信 201、プログラム実行 202、END 処理 204 により構成されている。ユニット間通信 201 で、基本ユニット 3 は、ラダープログラムを実行して得られた出力データを基本ユニット 3 内の記憶装置 12 から外部機器などに送信するとともに、受信データを含めた入力データを基本ユニット 3 内の記憶装置 12 に取り込む。プログラム実行 202 で、基本ユニット 3 は、更新された入力データを用いてプログラムを実行（演算）する。基本ユニット 3 はプログラムの実行によりデータを演算処理する。なお、END 処理とは、プログラム作成支援装置 1 や基本ユニット 3 に接続された表示器（図示せず）等の外部機器とのデータ通信、システムのエラーチェック等の周辺サービスに関する処理全般を意味する。

【0025】

このように、プログラム作成支援装置 1 はユーザの操作に応じたラダープログラムを作成し、作成したラダープログラムを PLC 2 に転送する。PLC 2 は、入出力リフレッシュ、ラダープログラムの実行および END 処理を 1 サイクル（1 スキャン）として、このサイクルを周期的、すなわちサイクリックに繰り返し実行する。これにより、各種入力機器（センサ等）からのタイミング信号に基づいて、各種出力機器（モータ等）を制御する。よって、PLC 2 は汎用のパーソナルコンピュータ（PC）とは全く異なる動きをする。

【0026】

< 連結構造 >

図 6 (A) は PLC 2 を構成する基本ユニット 3、拡張ユニット 4 およびエンドユニット 11 が連結される前の状態を示している。この例では、基本ユニット 3 の右方向に 1 つ以上の拡張ユニット 4 が連結される。最も右に位置する拡張ユニット 4 にはエンドユニット 11 が連結される。エンドユニット 11 は、最も右に位置する拡張ユニット 4 の右側面を保護するものであり、バスの終端機能を備えていてもよい。

【0027】

10

20

30

40

50

図 6 (B) は P L C 2 を構成する基本ユニット 3、拡張ユニット 4 およびエンドユニット 1 1 が連結された状態を示している。この P L C 2 はベースレスのビルディングタイプであるため、ベースを使用せずに、基本ユニット 3、拡張ユニット 4 およびエンドユニット 1 1 が連結される。工場において制御盤に設置しやすくするために、P L C 2 の背面には D I N レールなどの基準レール 1 3 が取り付けられるようになっている。なお、基本ユニット 3 および拡張ユニット 4 の連結状態を維持するためにロック機構が採用されてもよい。

【 0 0 2 8 】

< メモリの管理 >

上述したように P L C 2 の記憶装置 1 2 は揮発性メモリ (R A M など) と不揮発性メモリ (R O M、E E P R O M、フラッシュメモリ、S D メモリカードなど) を有している。一般に、プログラム作成支援装置 1 によって作成されたプロジェクトデータ (ラダープログラム、ユニット構成情報 (拡張ユニットの識別情報など)、ラベル (変数など)) は不揮発性メモリ (フラッシュメモリ) に格納され、そこから R A M にロードされて使用される。また、P L C の制御に関連しないユーザデータ (ラダープログラムを実行することで作成されたロギングファイル、P L C の取扱説明書など) は不揮発性メモリ (S D メモリカード) に格納される。近年になり、プロジェクトデータを格納するのに使用されていた不揮発性メモリの容量が増大し、未使用領域も増加することになった。この未使用領域にユーザデータを記憶すれば、未使用領域を有効活用することが可能となる。また、ユーザデータを格納するための専用のメモリを省略できる利点もある。ただし、プロジェクトデータとユーザデータとを単一のメモリに混在させると、プロジェクトデータを改変したときに、プロジェクトデータを格納できなくなってしまう恐れがある。これは未使用領域にユーザデータを格納した結果、未使用領域がほとんどなくなってしまうことで発生するだろう。よって、プロジェクトデータが改変されてサイズが増加することも考慮して、プロジェクトデータの記憶領域を確保しておき、その記憶領域にはユーザデータを格納しないような工夫が必要となる。反対にプロジェクトデータが膨大となり、ユーザデータを格納できなくなってしまうことも考えられる。よって、プロジェクトデータの記憶する記憶領域やユーザデータを記憶する記憶領域については上限サイズの管理が必要となろう。

【 0 0 2 9 】

図 7 (A) ないし図 7 (D) は上限サイズを管理する手法の一例を示す図である。記憶装置 1 2 には上述したように書き換え可能な不揮発性メモリ 1 0 0 が設けられている。図 7 (A) は最も単純な例を示しており、不揮発性メモリ 1 0 0 にはプロジェクトデータ 1 0 2 を記憶するプロジェクト領域 1 0 1 とユーザデータ 1 0 4 を記憶するユーザ領域 1 0 3 とが存在する。プロジェクト領域 1 0 1 のうち実際にプロジェクトデータ 1 0 2 によって消費されている記憶領域は斜線の部分である。同様に、ユーザ領域 1 0 3 のうち実際にユーザデータ 1 0 4 によって消費されている記憶領域は斜線の部分である。境界 1 1 1 は、プロジェクト領域 1 0 1 とユーザ領域 1 0 3 との境界を示している。境界 1 1 1 は、プロジェクト領域 1 0 1 の上限サイズとユーザ領域 1 0 3 の上限サイズとを示している。境界 1 1 1 はアドレスとして管理されてもよいし、プロジェクト領域 1 0 1 の上限サイズとユーザ領域 1 0 3 の上限サイズとを示す情報によって管理されてもよい。なお、この例は最も単純な例であるため、不揮発性メモリ 1 0 0 の全記憶領域のサイズは、プロジェクト領域 1 0 1 の上限サイズとユーザ領域 1 0 3 の上限サイズとの和に等しい。つまり、不揮発性メモリ 1 0 0 の全記憶領域のサイズとプロジェクト領域 1 0 1 の上限サイズとの差から、ユーザ領域 1 0 3 の上限サイズが算出されてもよい。反対に、不揮発性メモリ 1 0 0 の全記憶領域のサイズとユーザ領域 1 0 3 の上限サイズとの差から、プロジェクト領域 1 0 1 の上限サイズが算出されてもよい。このように、プロジェクト領域 1 0 1 の上限サイズとユーザ領域 1 0 3 の上限サイズとのうち一方を保持しておけば、境界 1 1 1 を管理できる。ユーザは、プログラム作成支援装置 1 や P L C 2 の基本ユニット 3 を操作することで境界 1 1 1 (プロジェクト領域 1 0 1 の上限サイズやユーザ領域 1 0 3 の上限サイズ) を変更する。

10

20

30

40

50

【 0 0 3 0 】

図 7 (B) は、プロジェクト領域 1 0 1 とユーザ領域 1 0 3 とを管理する手法の他の例を示している。この例では、プロジェクト領域 1 0 1 の上限サイズ 1 1 2 とユーザ領域 1 0 3 の上限サイズ 1 1 3 とが個別に管理されることを示している。ユーザは、プログラム作成支援装置 1 や P L C 2 の基本ユニット 3 を操作することでプロジェクト領域 1 0 1 の上限サイズ 1 1 2 やユーザ領域 1 0 3 の上限サイズ 1 1 3 を変更する。ただし、プロジェクト領域 1 0 1 の上限サイズ 1 1 2 やユーザ領域 1 0 3 の上限サイズ 1 1 3 との和は、不揮発性メモリ 1 0 0 の全記憶領域のサイズを超えることはできない。

【 0 0 3 1 】

図 7 (C) は、プロジェクト領域 1 0 1 、ユーザ領域 1 0 3 およびその他の用途の領域 1 0 5 を管理する手法の他の例を示している。上述した例は 2 つの記憶領域が例示されているが、本発明は 3 つ以上の記憶領域についても適用可能である。ユーザは、プログラム作成支援装置 1 や P L C 2 の基本ユニット 3 を操作することでプロジェクト領域 1 0 1 の上限サイズ 1 1 6 やユーザ領域 1 0 3 の上限サイズ 1 1 7 、その他の用途の領域 1 0 5 の上限サイズ 1 1 8 を変更する。ただし、プロジェクト領域 1 0 1 の上限サイズ 1 1 6 とユーザ領域 1 0 3 の上限サイズ 1 1 7 とその他の用途の領域 1 0 5 の上限サイズ 1 1 8 との和は、不揮発性メモリ 1 0 0 の全記憶領域のサイズを超えることはできない。

【 0 0 3 2 】

1 つの不揮発性メモリ 1 0 0 を複数の記憶領域に分割する方法はいくつか存在する。1 つはファイルシステムを用いて不揮発性メモリ 1 0 0 に複数のドライブを構築したり、1 つのドライブに複数のパーティションを構築したりする手法がある。しかし、これらの場合は上限サイズを変更するためにフォーマットやディスクイメージの書き込みなど、複雑な作業が発生してしまう。また、その時点で記憶領域に保存されているファイルのバックアップとリストアも必要になってしまい、非常に長い作業時間が必要になってしまう。

【 0 0 3 3 】

そこで、1 つのドライブに複数のフォルダ (ディレクトリ) を作成し、各フォルダに格納されるファイルのサイズの合計値が上限サイズを超えないように管理する手法が柔軟性やサイズの変更時間の点で有利であろう。また、この手法はファイルシステムを使用しなくても実現できる。ファイル (ファイルシステムを使用しない場合はファイルという概念は必須ではないが、便宜上 1 つのデータをファイルと呼ぶことにする) の種別 (プロジェクトデータに属するファイルやユーザデータに属するファイル) ごとにファイルのサイズの合計値を算出し、各合計値がそれぞれの上限サイズを超えないように書き込みを制限する手法が考えられる。ファイルシステムでは読み書きはクラスタと呼ばれる最小単位に基づき実行されるため、小さなデータであっても、少なくとも 1 つのクラスタを占有してしまう。これは無駄である。一方で、ファイルシステムを使用しなければ、このような問題は生じないといった利点もある。

【 0 0 3 4 】

< 記憶領域の上限サイズの調整方法 >

上限サイズの調整はプログラム作成支援装置 1 から実行されてもよいし、プログラム作成支援装置 1 を用いずに P L C 2 で実行されてもよい。

【 0 0 3 5 】

図 8 は上限サイズを設定するためのユーザインタフェース (U I) 1 2 0 の一例を示している。ここでは最も簡単な例である 2 つの記憶領域の境界 1 2 6 を変更することで、2 つの記憶領域の上限サイズを調整する手法について説明する。

【 0 0 3 6 】

ユーザインタフェース 1 2 0 はプログラム作成支援装置 1 の表示部 7 に表示され、操作部 8 の操作に連動する。C P U 2 4 はユーザによる操作部 8 の操作に応じてポインタ 6 4 の表示位置を変更して、表示部 7 に表示する。スライダバー 1 2 1 は、プロジェクト領域の上限サイズとユーザ領域の上限サイズを確定する境界 1 2 6 を調整するためのコントロールである。C P U 2 4 は、ポインタ 6 4 によってスライダバー 1 2 1 が右方向に移

10

20

30

40

50

動されると、これに連動して境界 1 2 6 を右方向に移動し、境界 1 2 6 の移動量または位置に応じて、プロジェクト領域の上限サイズを拡張（増加）し、ユーザ領域の上限サイズを縮小（削減）する。また、CPU 2 4 は、ポインタ 6 4 によってスライダバー 1 2 1 が左方向に移動されると、これに連動して境界 1 2 6 を左方向に移動し、境界 1 2 6 の移動量または位置に応じて、プロジェクト領域の上限サイズを縮小（削減）し、ユーザ領域の上限サイズを拡張（増大）する。

【0037】

CPU 2 4 は、PLC 2 に問い合わせ取得したプロジェクト領域における実際の消費サイズと、ユーザ領域における実際の消費サイズとについてそれぞれ表示してもよい。図 8 では消費サイズがグラフ（斜線部のバーグラフで表示されているが、円グラフなど他のグラフ形式でもよい）で表示されるとともに、表 1 2 3 にも数値（％表示等であってもよいし、％表示等も併記されてもよい）で表示されている。これにより、ユーザは各記憶領域にどの程度の未使用領域があるかを視覚的に把握でき、上限サイズの決定と調整に役立つであろう。表 1 2 3 には、各記憶領域の上限サイズ、プログラム作成支援装置 1 に保持されている現在のプロジェクトデータを PLC 2 に格納する上で必要となるサイズ（現在の必要サイズ）、プログラム作成支援装置 1 に保持されている現在のユーザデータを PLC 2 に格納する上で必要となるサイズ（現在の必要サイズ）、プロジェクトデータを PLC 2 に格納する上で実際に消費されているサイズ（PLC 2 内の消費要サイズ）、ユーザデータを PLC 2 に格納する上で実際に消費されているサイズ（PLC 2 内の消費要サイズ）といった情報が含まれている。CPU 2 4 は、OK ボタン 1 2 4 が押下されると、スライダバー 1 2 1 によって調整された上限サイズを PLC 2 に反映させる。これは変更要求を PLC 2 に送信し、PLC 2 が保持している上限サイズを示す情報を書き換えることで実現される。CPU 2 4 は、キャンセルボタン 1 2 5 が押下されると、スライダバー 1 2 1 によって調整された上限サイズを破棄し、PLC 2 に反映させずに、ユーザインタフェース 1 2 0 を閉じる。

【0038】

図 9 は上限サイズを設定するためのユーザインタフェース 1 2 0 ' の一例を示している。ユーザインタフェース 1 2 0 ' は、上述したユーザインタフェース 1 2 0 を簡素化したものであり、プログラム作成支援装置 1 の表示部 7 に表示されてもよいし、PLC 2 の表示部 5 に表示されてもよい。ここでは後者について説明する。

【0039】

PLC 2 の基本ユニット 3 は市場から小型化が求められているため、表示部 5 や操作部 6 も小型化が求められる。よって、表示部 5 に表示可能な情報量は少なくなり、操作部 6 から入力可能な指示も少なくなりやすい。この例では、操作部 6 に選択キー、決定キー、右キー、左キー、キャンセルキーなどが含まれているものとする。CPU 1 0 は、選択キーによって上限サイズの調整を選択されると、ユーザインタフェース 1 2 0 ' を表示部 5 に表示する。CPU 1 0 は、記憶装置 1 2 の不揮発性メモリ 1 0 0 に保持されている上限サイズの情報や、各記憶領域の消費サイズの情報を読み出す。CPU 1 0 は上限サイズの情報に基づき境界 1 2 6 の位置を決定して表示する。また、CPU 1 0 は各記憶領域の消費サイズの情報に基づき、バーグラフを表示する。

【0040】

CPU 1 0 は、右キーが押下されると、これに連動して境界 1 2 6 を右方向に移動し、境界 1 2 6 の移動量または位置に応じて、プロジェクト領域の上限サイズを拡張（増加）し、ユーザ領域の上限サイズを縮小（削減）する。また、CPU 1 0 は、左キーが押下されると、これに連動して境界 1 2 6 を左方向に移動し、境界 1 2 6 の移動量または位置に応じて、ユーザ領域の上限サイズを拡張（増加）し、プロジェクト領域の上限サイズを縮小（削減）する。

【0041】

また、CPU 1 0 は、各記憶領域について、実際の消費サイズと上限サイズとを数値（％表示等であってもよいし、％表示等も併記されてもよい）で表示してもよい。

【 0 0 4 2 】

消費サイズや必要サイズなどの記憶サイズ（記憶容量）に関する％（百分率）値は、CPU 10が記憶サイズを上限サイズで除算し、その商に100を乗算することで算出されてもよい。同様に上限サイズに関する％値は、CPU 10が不揮発性メモリ100の全記憶容量で上限サイズを除算し、その商に100を乗算することで算出されてもよい。

【 0 0 4 3 】

図10（A）は上限サイズを調整するためのユーザインタフェース120の他の例を示す図である。この例では、プロジェクト領域の上限サイズを調整するためのコントロールとしてテキストボックスが採用されている。同様に、ユーザ領域の上限サイズを調整するためのコントロールとしてもテキストボックスが採用されている。

10

【 0 0 4 4 】

図10（B）は上限サイズを調整するためのユーザインタフェース120の他の例を示す図である。この例では、プロジェクト領域の上限サイズを調整するためのコントロールとしてスピンコントロール（スピンボックス）が採用されている。同様に、ユーザ領域の上限サイズを調整するためのコントロールとしてもスピンコントロール（スピンボックス）が採用されている。なお、CPU 24は、一方の記憶領域の上限サイズがテキストボックスに入力されたり、スピンコントロールによって変更されたりすると、他方の記憶領域の上限サイズを算出して変更してもよい。不揮発性メモリ100の全記憶容量から一方の記憶領域の上限サイズを減算することで他方の記憶領域の上限サイズを算出できるからである。このように一方の記憶領域の上限サイズと他方の記憶領域の上限サイズとが関連している場合は、予めこれらの関連性を示す関数を用意しておくことで、CPU 10は一方の記憶領域の上限サイズから他方の記憶領域の上限サイズを算出してよい。

20

【 0 0 4 5 】

< 記憶領域とファイルパスの管理 >

ユーザに対しては複数の記憶領域をそれぞれ上限サイズを設定された個別の記憶領域であることを認識してもらうことで、プロジェクトデータやユーザデータの消費サイズに注意してもらうことが可能となろう。しかし、1つのドライブにおいて記憶領域ごとに複数のフォルダを作成する手法では、複数の記憶領域をそれぞれ上限サイズを設定された個別の記憶領域であることが分かりにくい。そこで、プログラム作成支援装置1から見たときの論理パスと、PLC 2における実パスとを変えることで、複数の記憶領域がそれぞれ上限サイズを設定された個別の記憶領域であることを明示してもよい。

30

【 0 0 4 6 】

図11（A）は、領域管理テーブルの一例を示す図である。領域管理テーブル130は、記憶領域の名称である領域名と、プログラム作成支援装置1から見たときの記憶領域のパスである論理パスと、PLC 2における記憶領域の実際のパスである実パスと、記憶領域に設定されている上限サイズと、実際の消費サイズとを示す情報が含まれてもよい。基本ユニット3のCPU 10は、領域管理テーブル130を参照することで、プログラム作成支援装置1から指定される論理パスを実パスに変更し、ファイルの読み書きを実行する。なお、CPU 10は、上述した上限サイズの調整が実行されると、領域管理テーブル130に保持されている上限サイズを更新する。

40

【 0 0 4 7 】

図11（B）はファイルサイズ管理テーブルの一例を示す図である。CPU 10は、各記憶領域に記憶されているファイルのサイズとその合計値（消費サイズ）を管理するためにファイルサイズ管理テーブル131を作成して維持している。基本ユニット3が起動すると、CPU 10は、各記憶領域に記憶されているファイルをスキャンしてサイズの情報を取得し、ファイルサイズ管理テーブル131に登録する。また、CPU 10は、ファイルの新規作成、更新、削除などのファイル操作が発生する度に、ファイルサイズを取得して、ファイルサイズ管理テーブル131を更新する。CPU 10は、ファイルサイズ管理テーブル131を参照して各記憶領域に記憶されているファイルのサイズの合計値（消費サイズ）を算出し、領域管理テーブル130の消費サイズを更新する。CPU 10は、消

50

費サイズと上限サイズを比較することで、各記憶領域に設定された上限サイズを超えるようなファイルの書き込みが要求されると、警告を出力（表示または音声出力）したり、当該書き込みを禁止したりする。

【 0 0 4 8 】

< P L C の機能 >

図 1 2 は P L C 2 の基本ユニット 3 に搭載されている C P U 1 0 がファームウェアを実行することで実現する機能の一例を示している。なお、すべての機能は必須というわけではなく、オプションであってもよい。記憶装置 1 2 は、揮発性メモリである R A M 1 4 1 と 2 つの不揮発性メモリ 1 0 0、1 4 2 を有している。上述したように不揮発性メモリ 1 0 0 には複数の記憶領域が設けられ、プロジェクト領域にはプロジェクトデータが記憶され、ユーザ領域にはユーザデータが記憶される。もう一つの不揮発性メモリ 1 4 2 には、各記憶領域の上限サイズを示す情報や領域管理テーブル 1 3 0 やファイルサイズ管理テーブル 1 3 1、イレース回数などが保持される。イレース回数は不揮発性メモリ 1 0 0 に対して実行されたイレース処理をカウントするための変数である。なお、不揮発性メモリ 1 0 0 の一部にこれらを記憶する記憶領域が確保されてもよい。なお、イレース回数に代えて書き換え回数がカウントされてもよい。

10

【 0 0 4 9 】

U I 制御部 1 4 3 は、表示部 5 に対するユーザインタフェース 1 2 0 ' の表示や操作部 6 から操作の入力を受け付けるユニットである。読み書き制御部 1 4 4 は、記憶装置 1 2 に対する読み書きを担当するユニットである。読み書き制御部 1 4 4 には様々な機能が含まれる。サイズ取得部 1 4 5 は、不揮発性メモリ 1 0 0 の各記憶領域における実際の消費サイズを取得するユニットであり、上述した領域管理テーブル 1 3 0 やファイルサイズ管理テーブル 1 3 1 を管理している。書き込み判定部 1 4 6 は、不揮発性メモリ 1 0 0 に確保されている記憶領域に対するファイルの書き込み要求を実行できるかどうかを、上限サイズと消費サイズとに基づいて判定するユニットである。速度制御部 1 4 7 は不揮発性メモリ 1 0 0 に対するデータの書き込み速度（例：所定時間当たりの平均書き込み速度）を制御するユニットである。警告部 1 4 8 は、上限サイズの変更が許容されない場合やファイルの書き込みが禁止されるべき場合に警告を表示部 5 やスピーカー（不図示）から出力するユニットである。読み書き制御部 1 4 4 は、ファイル転送プロトコル（ F T P など）などの通信プログラムが実装されており、通信部 1 4 を通じて外部機器と通信してデータの読み書きを実行してもよい。

20

30

【 0 0 5 0 】

サイズ調整部 1 5 0 は、不揮発性メモリ 1 0 0 に確保されている複数の記憶領域のそれぞれの上限サイズを調整するユニットである。受付部 1 5 1 は、操作部 6 やプログラム作成支援装置 1 から上限サイズの変更要求を受け付けるユニットである。可否判定部 1 5 2 は、操作部 6 やプログラム作成支援装置 1 によって要求された上限サイズの調整を反映できるかどうかを判定するユニットである。サイズ調整部 1 5 0 は、上限サイズの調整を反映可能と判定すると、領域管理テーブル 1 3 0 に保持されている上限サイズを調整後の上限サイズでもって更新する。

【 0 0 5 1 】

ここでは P L C 2 が主体となって上限サイズを調整する例を示しているが、プログラム作成支援装置 1 が主体となってもよい。とりわけ、プログラム作成支援装置 1 が記憶装置 1 2 にアクセスできる場合、読み書き制御部 1 4 4 やサイズ調整部 1 5 0 の機能の大半はプログラム作成支援装置 1 の C P U 2 4 によって実現されてもよい。

40

【 0 0 5 2 】

< プログラム作成支援装置の機能 >

図 1 3 はプログラム作成支援装置 1 に搭載されている C P U 2 4 が設定プログラム 1 6 1 を実行することで実現する機能の一例を示している。なお、すべての機能は必須というわけではなく、オプションであってもよい。記憶装置 2 5 の編集ソフト記憶部 2 5 b には、上限サイズを調整して設定する設定プログラム 1 6 1 が記憶されている。また、記憶装

50

置 2 5 のラダープログラム記憶部 2 5 a には、プロジェクトデータ 1 6 2 が記憶されている。また、記憶装置 2 5 にはユーザデータ 1 6 3 が記憶されている。

【 0 0 5 3 】

C P U 2 4 は例えば次のような機能を実現する。サイズ取得部 1 6 4 は、記憶装置 2 5 に記憶されているプロジェクトデータ 1 6 2 のサイズとユーザデータのサイズを取得するユニットである。これらは上述した必要サイズである。これらは P L C 2 に対して書き込みの対象となっているデータである。また、サイズ取得部 1 6 4 は、通信部 2 6 を介して P L C 2 と通信し、各記憶領域の上限サイズや消費サイズを取得してもよい。サイズ設定部 1 6 5 は、スライダバー 1 2 1 の移動量やテキストボックスまたはスピンボックスに入力された数値に基づき、各記憶領域の上限サイズを算出し、算出した上限サイズを指定した上限サイズの変更要求を P L C 2 に対して通信部 2 6 を介して送信するユニットである。U I 制御部 1 6 6 はユーザインタフェース 1 2 0 の表示を制御したり、操作部 8 からの入力を受け付けたりするユニットである。U I 制御部 1 6 6 は、サイズ取得部 1 6 4 によって取得された上限サイズや消費サイズに応じてバーグラフを作成したり、境界 1 2 6 の位置を確定したり、操作部から入力された指示に応じてスライダバー 1 2 1 の表示位置を変更したりする。また、U I 制御部 1 6 6 は、サイズ取得部 1 6 4 を通じて取得した各記憶領域の上限サイズ、必要サイズおよび消費サイズに基づき表 1 2 3 を作成して、ユーザインタフェース 1 2 0 に表示してもよい。可否判定部 1 6 7 は、サイズ設定部 1 6 5 により設定された上限サイズを P L C 2 に対して反映させることができるかどうかを判定するユニットである。可否判定部 1 6 7 は、たとえば、消費サイズと上限サイズとを比較してもよいし、消費サイズと上限サイズとの比較結果を P L C 2 の可否判定部 1 5 2 から取得し、比較結果が上限サイズを変更可能であることを示しているのかそれとも不可能であることを示しているのかを判定してもよい。

10

20

【 0 0 5 4 】

< フローチャート >

図 1 4 は、P L C 2 の C P U 1 0 が実行する上限サイズの設定処理を示すフローチャートである。P L C 2 の操作部 6 やプログラム作成支援装置 1 の操作部 8 を通じて上限サイズの設定処理の開始が指示されると C P U 1 0 は以下の処理を実行する。なお、上限サイズの設定処理が開始されると、読み書き制御部 1 4 4 は、不揮発性メモリ 1 0 0 に対するデータの書き込みを禁止または制限してもよい。上限サイズの設定処理は、プロジェクトデータに含まれているラダープログラムが実行されている期間においても実行可能であり、実行されていない期間においても実行可能である。

30

【 0 0 5 5 】

S 1 1 で C P U 1 0 (サイズ取得部 1 4 5) は不揮発性メモリ 1 0 0 の各記憶領域における消費サイズを取得する。たとえば、サイズ取得部 1 4 5 は、ファイルサイズ管理テーブル 1 3 1 を参照して各ファイルのサイズを取得し、取得したサイズを加算して合計値を算出する。なお、これから書き込まれるファイルについて必要サイズが必要な場合は、プログラム作成支援装置 1 から必要サイズを取得する。

【 0 0 5 6 】

S 1 2 で C P U 1 0 (受付部 1 5 1) は上限サイズの調整を受け付ける。受付部 1 5 1 は操作部 6 やプログラム作成支援装置 1 から指定される上限サイズの調整量を受け付ける。

40

【 0 0 5 7 】

S 1 3 で C P U 1 0 (可否判定部 1 5 2) は上限サイズの調整を P L C 2 に対して反映可能かどうかを判定する。可否判定部 1 5 2 は受付部 1 5 1 により受け付けられた各記憶領域の上限サイズとサイズ取得部 1 4 5 によって取得された消費サイズに基づき、上限サイズの調整を反映可能かどうかを判定する。なお、これからプログラム作成支援装置 1 によって作成されたプロジェクトデータ 1 6 2 やユーザデータ 1 6 3 が書き込まれるタイミングでは、可否判定部 1 5 2 は、上限サイズと必要サイズとに基づき、上限サイズの調整を反映可能かどうかを判定する。調整を反映可能でない場合、可否判定部 1 5 2 は警告部

50

148に警告の出力を指示し、S14に進む。

【0058】

S14でCPU10(警告部148)は上限サイズを調整できないことを示す警告メッセージを表示部5に出力する。一方で、調整を反映可能であればS15に進む。

【0059】

S15でCPU10(サイズ調整部150)は、上限サイズの調整を反映させる。たとえば、サイズ調整部150は、読み書き制御部144に対して領域管理テーブル130などによって管理されている各記憶領域の上限サイズを、指定した上限サイズで更新するよう指示する。読み書き制御部144は、指定された上限サイズで領域管理テーブル130などによって管理されている各記憶領域の上限サイズを書き換える。

10

【0060】

S16でCPU10(読み書き制御部144)は不揮発性メモリ100に対するデータの書き込みを再開する。読み書き制御部144は、たとえば、プログラム作成支援装置1から送信されてきた新しいプロジェクトデータをプロジェクト領域に書き込み、領域管理テーブル130とファイルサイズ管理テーブル131を更新する。同様に、読み書き制御部144は、プログラム作成支援装置1から送信されてきた新しいユーザデータをユーザ領域に書き込み、領域管理テーブル130とファイルサイズ管理テーブル131を更新する。

【0061】

<ユーザ領域に対するデータの書き込み単位>

20

PLC2のCPU24のロギング部153はラダープログラムなどのユーザプログラムを実行することで生成されたデータをログとして記録する。ロギング部153はこのようなロギングデータをユーザ領域に書き込む。ところで、ユーザはPLC2の電源スイッチをオフにすることでPLC2の動作電力を簡単に遮断することができる。そのため、ユーザによってロギングデータの書き込み中に電源スイッチがオフにされると、ロギングデータの書き込みが中途半端な状態で終了してしまう可能性がある。

【0062】

そこで、ロギング部153は、ユーザプログラムを実行することで生成されるロギングデータを一定のサンプリング数を単位としてユーザ領域に書き込むことにする。一般にファイルシステムではトランザクションを単位としてデータを書き込む。よって、ロギング部153が1つのトランザクションに相当するサンプリング数のデータを一塊としてユーザ領域に書き込むことで、ロギングデータの中途半端な書き込みを抑制できるようになる。たとえば、1つのトランザクションがn個のサンプルから構成される場合、ロギング部153は、n個のサンプルずつロギングデータをユーザ領域へ書き込む。

30

【0063】

<書き込み速度の制御>

不揮発性メモリ100などの記憶装置には性能上(設計上)の上限速度が存在する。よって、通常は、性能上の上限速度でデータが不揮発性メモリ100に書き込まれる。その一方で、不揮発性メモリ100をフラッシュメモリやEEPROMで構成する場合、不揮発性メモリ100にはイレース回数や書き換え回数の制限が存在する。この制限を超えると不揮発性メモリ100に書き込むことが不可能となってしまう。PLC2には設計段階で製品寿命(期待寿命)が想定され、その製品寿命を下回らないように各部品が選択される。よって、不揮発性メモリ100の寿命がPLC2の期待寿命を下回らないように、不揮発性メモリ100への書き込みを制限する必要がある。

40

【0064】

たとえば、書き込み制御部144の速度制御部147は、不揮発性メモリ100への情報の書き込み速度を不揮発性メモリ100の性能上の上限速度よりも低速な書き込み速度に制限してもよい。ここで、書き込み速度は、ある所定期間における平均書き込み速度である。速度制御部147が書き込み用のバッファに保持されているデータを不揮発性メモリ100に対して間欠的(一定の待機時間ごとに)に出力することで、ある所定期間にお

50

ける平均書き込み速度を低下させることが可能となる。これにより、ある所定期間に発生するイレース回数や書き換え回数が制限されるため、不揮発性メモリ 100 の寿命が PLC 2 の期待寿命を下回らないようになる。なお、ある所定期間における平均書き込み速度は、不揮発性メモリ 100 の寿命（イレース回数や書き換え回数の上限値）と PLC 2 の期待寿命とから算出される。つまり、期待寿命内でのイレース回数や書き換え回数が上限値以下となるように、平均書き込み速度が算出される。

【0065】

ところで、プロジェクトデータの書き換え頻度はそれほど多くないが、ユーザプログラムを実行することで発生するロギングデータなどのユーザデータの書き換え頻度は相対的に多い。そこで、速度制御部 147 は、不揮発性メモリ 100 に対する書き込みの対象となっているデータの種別（プロジェクトデータか、ユーザデータかなど）に応じて書き込み速度を制限してもよい。たとえば、速度制御部 147 は、プロジェクト領域に対するプロジェクトデータの書き込みを検知すると、書き込み速度を高速に設定する。一方で、速度制御部 147 は、ユーザ領域に対するユーザデータの書き込みを検知すると、書き込み速度を低速に設定する。このように書き換え頻度の多い種別のデータについては書き込み速度を低速にすることで、効率よく、不揮発性メモリ 100 の寿命を延ばすことができる。一方で、プロジェクトデータのような比較的サイズの大きなデータについて書き込み速度を低下させてしまうと、プログラム作成支援装置 1 から PLC 2 へのデータの転送時間が長くなり、ユーザビリティが低下する。よって、このような種別のデータについては書き込み速度を高速とし、ユーザビリティ向上させる。

【0066】

なお、このような低速化は、プログラム作成支援装置 1 からの書き込み要求には適用されず、PLC 2 内で発生した書き込み要求にだけ適用されてもよい。速度制御部 147 は、ユーザプログラムの実行によって発生する情報の書き込み速度を制限し、プログラマブル・ロジック・コントローラの起動時、メンテナンス時またはプログラム作成支援装置 1 からのユーザプログラムまたはユーザデータの転送時には書き込み速度を制限しないようにしてもよい。これにより不揮発性メモリ 100 の寿命が PLC 2 の期待寿命を下回らないようにしつつ、高速に書き込むことが必要な情報については高速に書き込むことが可能となる。つまり、PLC 2 の期待寿命内で実行されるイレース回数や読み書き回数が不揮発性メモリ 100 の寿命回数を超えないようになる。

【0067】

<イレース回数に基づく警告>

不揮発性メモリ 100 のイレース回数が不揮発性メモリ 100 の寿命回数を超えると、不揮発性メモリ 100 にはもはや情報を書き込めなくなってしまう。これは、PLC 2 の交換や修理が必要となることを意味する。工場で製造ラインが稼働中に PLC 2 の動作が停止してしまうと、製造ラインの稼働率が低下してしまう。製造ラインの稼働中における PLC 2 の動作停止を回避するためには、不揮発性メモリ 100 の寿命が近づいた時に、ユーザにそれを警告して、PLC 2 の交換や修理を事前に促すことが望ましい。

【0068】

そこで、カウント部 154 は、不揮発性メモリ 100 のセクタごとイレース回数または書き換え回数をカウントし、カウント値（イレース回数 / 書き換え回数）を不揮発性メモリ 142 に記憶させる。さらに、書き込み判定部 146 は、不揮発性メモリ 100 におけるいずれかのセクタについてのカウント値が閾値を超えると警告部 148 に警報を出力するように指示する。警告部 148 に警報を表示部 5 や音声出力部（不図示）から出力する。

【0069】

これにより不揮発性メモリ 100 の寿命が尽きる前に PLC 2 のメンテナンスをユーザに促すことが可能となる。

【0070】

<まとめ>

本実施形態では、ユーザプログラムとユーザデータとを記憶する記憶手段の一例として

不揮発性メモリ 100 について説明した。また、不揮発性メモリ 100 において、ユーザプログラムを記憶するために許容される記憶領域（プロジェクト領域）の上限サイズと、ユーザデータを記憶するために許容される記憶領域（ユーザ領域）の上限サイズとのうちいずれかを確定するための情報を保持する保持手段として不揮発性メモリ 142 や領域管理テーブル 130 について説明した。読み書き制御部 144 は、不揮発性メモリ 142 や領域管理テーブル 130 に保持されている情報によって確定する上限サイズを超えない範囲でユーザプログラムまたはユーザデータを不揮発性メモリ 100 に書き込む。このように不揮発性メモリ 100 における複数の記憶領域のうち少なくとも 1 つの領域について上限サイズを設定することで、残りの未使用領域を他の記憶領域として使用することが可能となる。たとえば、ユーザプログラムについて想定される上限（最大）サイズを管理することで、残りの未使用領域を他の記憶領域（ユーザ領域）として使用することが可能となる。また、ユーザデータの上限サイズを管理することで、残りの未使用領域を他の記憶領域（プロジェクト領域）として使用することが可能となる。いずれの場合もメンテナンス等によるユーザプログラムのサイズの拡大を想定して、十分なサイズのプロジェクト領域を確保するように、上限サイズを決定することになる。これにより、PLC においてユーザプログラムが記憶される記憶手段の未使用領域を有効活用することが可能となる。また、ユーザプログラムやユーザデータの上限サイズを超えない範囲でユーザプログラムやユーザデータを不揮発性メモリ 100 に書き込むようにしたので、ユーザプログラムの改変時の問題も緩和できるようになる。つまり、ユーザプログラムを改変してもユーザプログラムを不揮発性メモリ 100 に書き込めなくなったり、ユーザデータを書き込めなくなったりといった課題のいずれかが解決されよう。

【0071】

不揮発性メモリ 142 や領域管理テーブル 130 は、不揮発性メモリ 100 においてユーザプログラムを格納可能な第 1 記憶領域（プロジェクト領域）と、ユーザデータを格納可能な第 2 記憶領域（ユーザ領域）との境界 126 となる記憶位置を保持していてもよい。図 7 などを用いて説明したように、境界 126 となる記憶位置は、アドレスであってもよいし、プロジェクト領域の上限サイズとユーザ領域の上限サイズとを個別に管理することで、実質的に境界 126 となる記憶位置が管理されてもよい。もちろん、記憶位置を管理せずに、単純に上限サイズだけを管理してもよい。これは、記憶領域の上限サイズを実際の消費サイズが超えなければ、十分だからである。

【0072】

図 12 などを用いて説明したように、受付部 151 は、プロジェクト領域とユーザ領域との境界となる記憶位置の変更要求を受け付ける。サイズ調整部 150 は、受け付けられた変更要求にしたがってプロジェクト領域とユーザ領域との境界となる記憶位置を変更する変更手段として機能してもよい。これにより、プロジェクト領域の上限サイズやユーザ領域の上限サイズを変更することが可能となる。

【0073】

図 11 (A) を用いて説明したように、不揮発性メモリ 142 や領域管理テーブル 130 は、ユーザプログラムを記憶するために許容される記憶領域の上限サイズと、ユーザデータを記憶するために許容される記憶領域の上限サイズとの両方を保持していてもよい。上述したように、片方の上限サイズだけを保持しておいて他方の上限サイズについては算出されてもよい。不揮発性メモリ 100 の全記憶領域のサイズは既知であるため、これを利用して片方の上限サイズから他方の上限サイズは簡単に求められる。

【0074】

図 7 などを用いて説明したように、不揮発性メモリ 100 の全記憶領域のうち、ユーザプログラムを記憶するために許容される第 1 記憶領域以外の残りの第 2 記憶領域内にユーザデータが記憶される。これにより元々はプロジェクト領域専用のものであった不揮発性メモリ 100 の未使用領域を有効活用することが可能となる。

【0075】

図 12 を用いて説明したように、サイズ取得部 145 は、不揮発性メモリ 100 におい

てユーザプログラムを記憶するために消費されている記憶領域の消費サイズを取得してもよい。サイズ調整部 150 は、不揮発性メモリ 100 における全記憶領域のサイズからユーザプログラムについての消費サイズを減算して得られる残りの記憶領域のサイズを超えない範囲でユーザデータの記憶領域の上限サイズを設定する。これにより、すでに不揮発性メモリ 100 に記憶されているユーザプログラムを破棄することなく、安全に上限サイズを調整することが可能となる。

【0076】

図 12 を用いて説明したように、サイズ取得部 145 は、不揮発性メモリ 100 においてユーザプログラムとユーザデータを記憶するために実際に消費されている記憶領域の消費サイズを取得してもよい。可否判定部 152 は、消費サイズとユーザデータを記憶するための記憶領域についての拡張サイズとの和が、不揮発性メモリ 100 における全記憶領域のサイズ以下であるかどうかを判定する。サイズ調整部 150 は、この和が全記憶領域のサイズ以下である場合に、ユーザデータの記憶領域のサイズを拡張サイズに応じて拡張する拡張手段として機能する。不揮発性メモリ 100 における全記憶領域のサイズを超えるような、ユーザ領域の拡張は認められない。また、PLC2 の動作中にユーザデータやプロジェクトデータを強制的に消去してしまうと、PLC2 が誤作動してしまう。よって、ユーザプログラムとユーザデータの実際の消費サイズを考慮して、ユーザ領域の拡張を許容することで、PLC2 の誤作動を低減できよう。

【0077】

可否判定部 152 は、受付部 151 によって受け付けられた変更要求に基づくプロジェクト領域の変更後の上限サイズをユーザプログラムについての消費サイズが超えず、かつ、変更要求に基づくユーザ領域の変更後の上限サイズをユーザデータについての消費サイズが超えないかどうかに応じてプロジェクト領域の上限サイズとユーザ領域の上限サイズとの変更の可否を判定してもよい。サイズ調整部 150 は、可否判定部 152 がプロジェクト領域の上限サイズとユーザ領域の上限サイズとの変更が可能と判定すると、プロジェクト領域の上限サイズおよびユーザ領域の上限サイズの変更を反映させる領域設定手段として機能する。図 12 を用いて説明したように、プロジェクト領域の上限サイズとユーザ領域の上限サイズとが個別に調整されてもよいし、同時にこれらが調整されてもよい。後者の場合、両方の記憶領域について上限サイズの調整が可能と判断されたときに、調整が反映されることになる。前者の場合は、一方の記憶領域のみが調整条件を満たしており、他方の記憶領域は調整条件を満たしていないことがある。この場合、サイズ調整部 150 は、一方の記憶領域のみ調整を反映してもよいし、両方の記憶領域について反映を中止してもよい。

【0078】

プロジェクトデータを改変したときに、プロジェクト領域の上限サイズの拡張が必要になることがある。この場合、改変後のプロジェクトデータ 162 を格納するための必要サイズと、ユーザデータ 163 を格納するための必要サイズがサイズ取得部 145、164 によって取得される。可否判定部 152 は、受け付けられた変更要求に基づくプロジェクト領域の変更後の上限サイズをプロジェクトデータ 162 についての必要サイズが超えず、かつ、変更要求に基づくユーザ領域の変更後の上限サイズをユーザデータ 163 についての必要サイズが超えないかどうかに応じてプロジェクト領域の上限サイズとユーザ領域の上限サイズとの変更の可否を判定してもよい。サイズ調整部 150 は、可否判定部 152 がプロジェクト領域の上限サイズとユーザ領域の上限サイズとの変更が可能と判定すると、プロジェクト領域の上限サイズおよびユーザ領域の上限サイズの変更を反映させる。これにより、プロジェクトデータを改変したときにも、改変されたプロジェクトデータを格納できる範囲内で上限サイズの変更が許容される。

【0079】

このように、可否判定部 152 は、原則として、不揮発性メモリ 100 へのユーザプログラムとユーザデータの書き込みが要求されたときに判定処理を実行してもよい。

【0080】

10

20

30

40

50

受付部 151 は、プロジェクトデータに含まれているユーザプログラムが実行されているときにもプロジェクト領域の上限サイズとユーザ領域の上限サイズの変更を受け付けてもよい。これにより PLC 2 の動作を止めることなく、つまり、製造ラインを止めることなく、上限サイズを調整することが可能となる。

【0081】

受付部 151 は、プログラマブル・ロジック・コントローラにユーザプログラムを転送するプログラム作成支援装置 1 から送信される変更要求を受け付けてもよい。これによりプログラム作成支援装置 1 から上限サイズの調整を実行することが可能となる。図 8 や図 9 を用いて説明したように、プログラム作成支援装置 1 は PLC 2 よりもリッチな（情報量の多い）GUI を提供できる。よって、ユーザは視覚的に理解しやすい状態で上限サイズを調整することが可能となろう。

10

【0082】

読み書き制御部 144 のロギング部 153 は、ユーザプログラムを実行することで生成されるロギングデータを一定のサンプリング数を単位として不揮発性メモリ 100 に書き込んでもよい。これにより停電等が発生した PLC 2 が停止しても、中途半端なロギングデータが保存されないようになる。

【0083】

読み書き制御部 144 の速度制御部 147 は、不揮発性メモリ 100 への情報の書き込み速度を不揮発性メモリ 100 の性能上の上限速度よりも低速な書き込み速度に制限してもよい。これにより不揮発性メモリ 100 の寿命を延ばすことが可能となろう。なお、読み書き制御部 144 の速度制御部 147 は、情報の書き込み速度を当該情報の種別に応じて制限してもよい。たとえばユーザプログラムとユーザデータとでは必要となる書き込み速度に違いがあるため、データの種別に応じて書き込み速度が制限されてもよい。また、速度制御部 147 は、ユーザプログラムの実行によって発生する情報の書き込み速度を制限し、プログラマブル・ロジック・コントローラの起動時、メンテナンス時またはプログラム作成支援装置 1 からのユーザプログラムまたはユーザデータの転送時には書き込み速度を制限しないように動作してもよい。これによりユーザビリティを低下させにくくしつつ、不揮発性メモリ 100 の寿命を延ばすことが可能となろう。

20

【0084】

カウント部 154 は、不揮発性メモリ 100 のセクタごとイレース回数または書き換え回数をカウントしてもよい。警告部 148 や表示部 5 は、不揮発性メモリ 100 におけるいずれかセクタについてのカウント部 154 によるカウント値が閾値を超えると警報を出力する出力手段として機能してもよい。これによりユーザは不揮発性メモリ 100 の寿命が近づいていることを把握して、PLC 2 を交換することが可能となろう。また、不揮発性メモリ 100 の寿命が尽きることによる、突然の PLC 2 の動作停止といった事態を起こりにくくすることが可能となろう。

30

【0085】

上述したように上限サイズの調整と設定は PLC 2 が主体となり、プログラム作成支援装置 1 はユーザインタフェースを提供するだけであってもよいが、図 8 や図 13 を用いて説明したように、プログラム作成支援装置 1 が主体となって上限サイズの調整と設定を実行してもよい。この場合、サイズ調整部 150 の機能はプログラム作成支援装置 1 に実装されることになる。

40

【0086】

たとえば、サイズ設定部 165 は、不揮発性メモリ 100 においてユーザプログラムを記憶するために許容されるプロジェクト領域の上限サイズと、ユーザデータを記憶するために許容されるユーザ領域の上限サイズとのうちいずれかを設定する。これにより不揮発性メモリ 100 の未使用領域にユーザデータを格納可能となり、未使用領域の有効活用を図れるであろう。

【0087】

図 8 ないし図 10 を用いて説明したように、ユーザインタフェース 120、120' お

50

よびUI制御部166は、プロジェクト領域の上限サイズとユーザ領域の上限サイズとを表示する表示手段として機能する。また、操作部8やスライダバー121、テキストボックス、スピンのボックスは、ユーザインタフェース120に表示されたプロジェクト領域の上限サイズとユーザ領域の上限サイズとを調整する調整手段として機能してもよい。

【0088】

サイズ取得部164は、不揮発性メモリ100においてユーザプログラムとユーザデータを記憶するために実際に消費されている記憶領域の消費サイズを取得してもよい。ユーザインタフェース120、120'およびUI制御部166は、取得されたユーザプログラムについての消費サイズとユーザデータについての消費サイズとをさらに表示してもよい。これによりユーザは不揮発性メモリ100の各記憶領域にどの程度のデータが記憶されており、どの程度の空き領域があるかを容易に把握できるようになる。

10

【0089】

可否判定部167は、操作部8やスライダバー121、テキストボックス、スピンのボックスによって調整されたプロジェクト領域の上限サイズをユーザプログラムについての消費サイズが超えず、かつ、調整されたユーザ領域の上限サイズをユーザデータについての消費サイズが超えないかどうかに応じて、プロジェクト領域の上限サイズおよびユーザ領域の上限サイズの調整をPLC2に反映させることが可能かどうかを判定してもよい。サイズ設定部165は、可否判定部167がこれらの調整をPLC2に反映させることが可能と判定すると、プロジェクト領域の上限サイズおよびユーザ領域の上限サイズの調整をPLC2に反映させてもよい。

20

【0090】

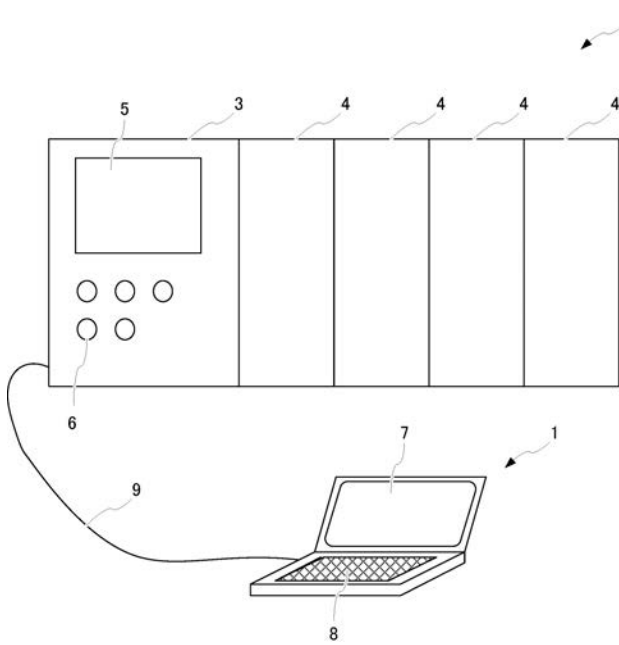
図8、図9を用いて説明したように、ユーザインタフェース120、120'およびUI制御部166は、ユーザデータについての消費サイズをバーグラフとして表示してもよい。サイズ設定部165は、ユーザ領域の上限サイズをバーグラフに沿って調整してもよい。これによりユーザは簡単な操作によりユーザ領域の上限サイズを調整できるようになる。なお、ユーザインタフェース120、120'およびUI制御部166は、ユーザデータについての消費サイズと、ユーザプログラムを含むプロジェクトデータの消費サイズとをそれぞれバーグラフとして表示してもよい。これによりユーザは不揮発性メモリ100の各記憶領域にどの程度のデータが記憶されており、どの程度の空き領域があるかを容易に把握できるようになる。

30

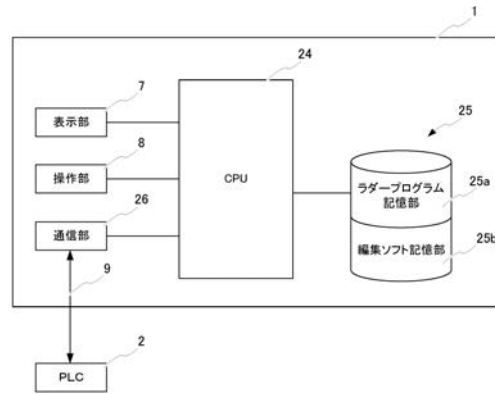
【0091】

本実施形態によれば、PLC2の制御方法の各工程をCPU10に実行させるプログラム(ファームウェア)も提供される。同様に、プログラム作成支援装置1の制御方法の各工程をCPU24に実行させる設定プログラム161も提供される。

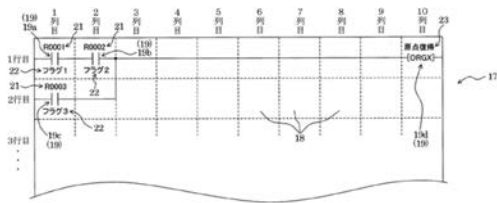
【図 1】



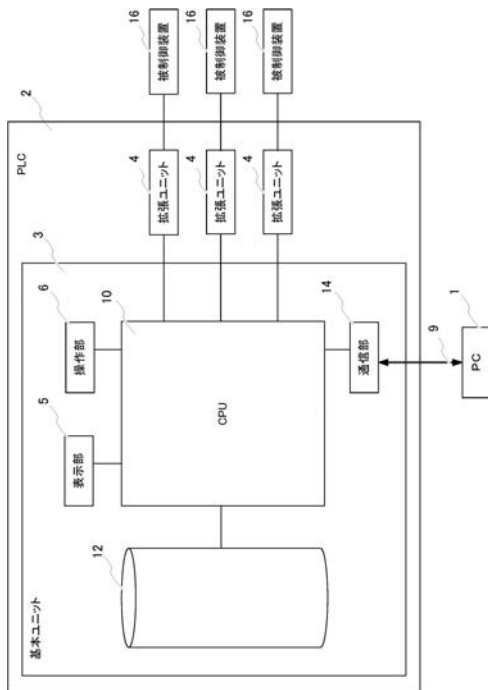
【図 3】



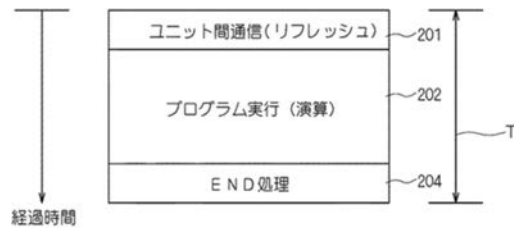
【図 2】



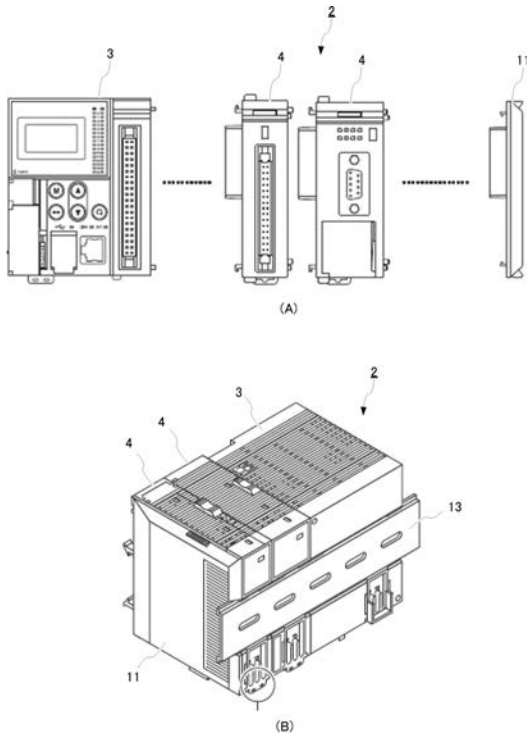
【図 4】



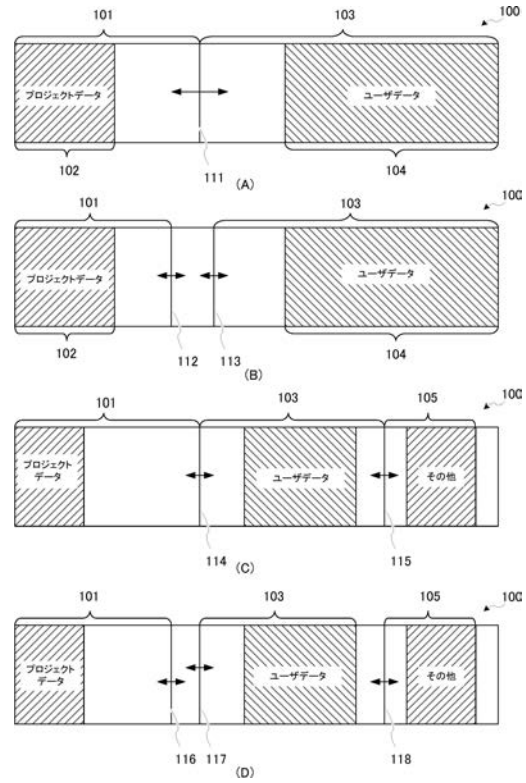
【図 5】



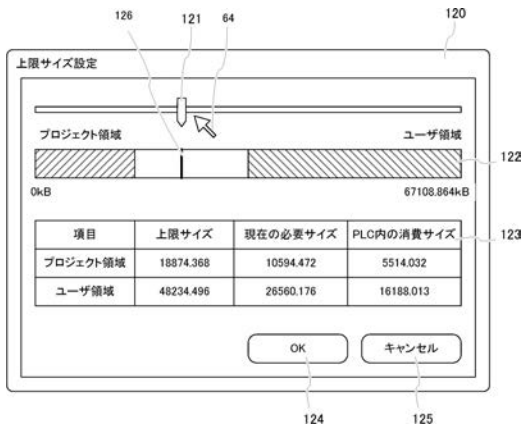
【図 6】



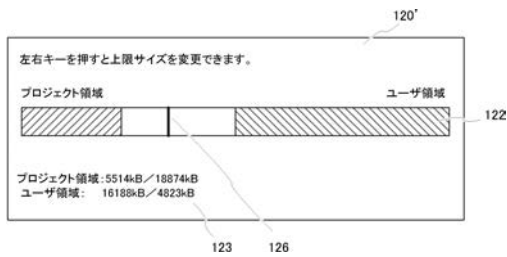
【図 7】



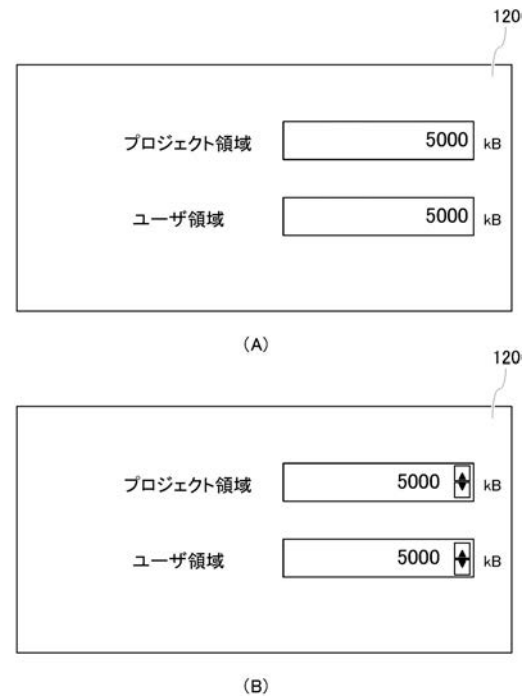
【図 8】



【図 9】



【図 10】



【図 1 1】

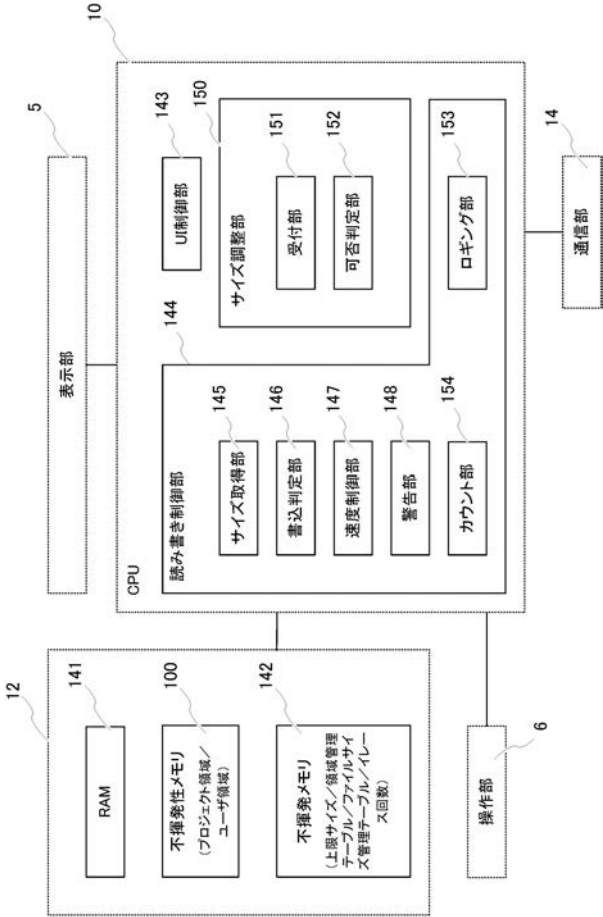
領域名	論理パス	実パス	上限サイズ	消費サイズ
プロジェクト領域	/プロジェクト領域	/ルートドライブ/プロジェクト領域	18874 kB	5514 kB
ユーザ領域	/ユーザ領域	/ルートドライブ/ユーザ領域	48234 kB	16188 kB
-	-	-	-	-

(A)

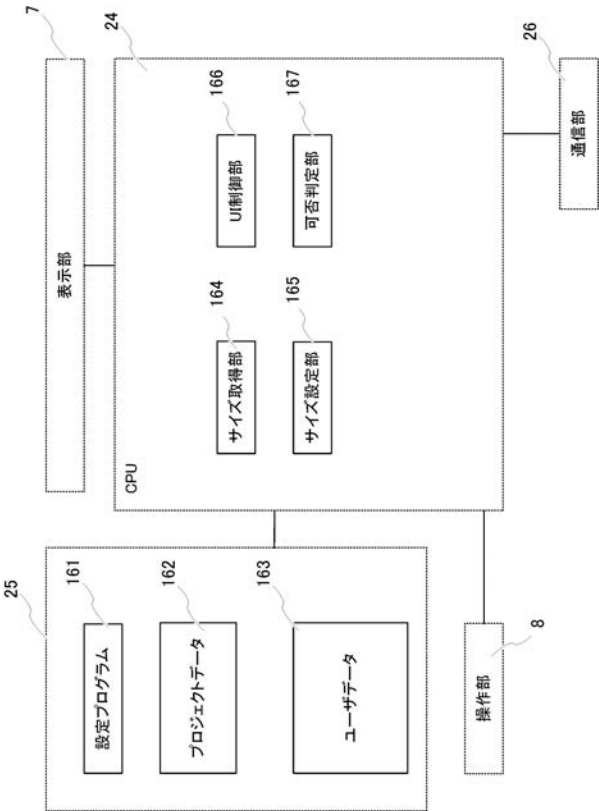
実ファイルパス	ファイルサイズ
/ルートドライブ/プロジェクト領域/aaa.txt	100 byte
/ルートドライブ/ユーザ領域/aaa.txt	500 byte
-	-

(B)

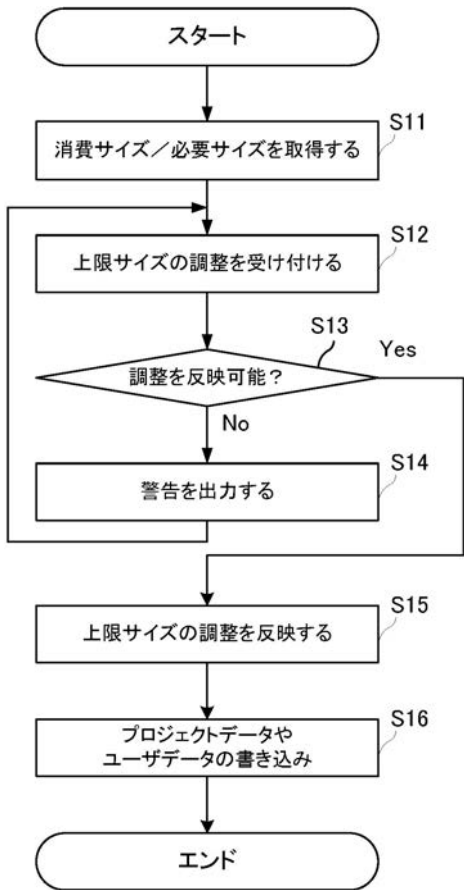
【図 1 2】



【図 1 3】



【図 1 4】



フロントページの続き

(72)発明者 立田 和俊

大阪府大阪市東淀川区東中島 1 - 3 - 1 4 株式会社キーエンス内

Fターム(参考) 5H220 BB12 CC07 CX04 DD04 EE01 JJ01 JJ12 JJ13 JJ26 JJ41