

(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2010-286935

(P2010-286935A)

(43) 公開日 平成22年12月24日(2010.12.24)

(51) Int.Cl. F I テーマコード(参考)  
**G06K 17/00 (2006.01)** G06K 17/00 F 5B058

審査請求 未請求 請求項の数 5 O L (全 26 頁)

<p>(21) 出願番号 特願2009-138776 (P2009-138776)                  (22) 出願日 平成21年6月10日 (2009.6.10)</p>	<p>(71) 出願人 000005108                  株式会社日立製作所                  東京都千代田区丸の内一丁目6番6号                  (74) 代理人 100100310                  弁理士 井上 学                  (74) 代理人 100098660                  弁理士 戸田 裕二                  (72) 発明者 山本 暖                  神奈川県川崎市麻生区王禅寺1099番地                  株式会社日立製作所システム開発研究所                  内                  (72) 発明者 金田 悦雄                  東京都港区港南二丁目15番2号 株式会                  社日立製作所情報制御システム事業部内</p>
--	--

最終頁に続く

(54) 【発明の名称】 ICタグへのデータ書き込みシステム及び方法

(57) 【要約】

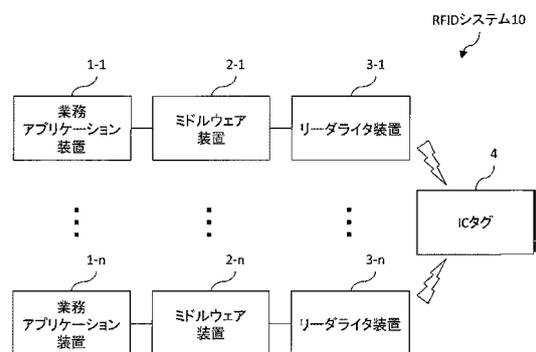
【課題】

ICタグの仕様変更に伴う業務アプリケーションの改修頻度や、業務アプリケーション間で共有する情報の更新頻度を抑えながら、計算能力やメモリ容量に制限のあるICタグに対して、データ書込処理を高速化すること。

【解決手段】

ICタグに対するデータ書込コマンドを発行する前に、直前にICタグから読み込んだデータと、これから書き込もうとするデータの差分を取り、書き換えの発生したデータに限定して書込要求コマンドを発行することで、書込要求コマンドの総実行回数を削減する。また、ICタグ内の記憶領域を、基本領域と追記領域の2つに分割し、新たにデータを書き込む際には、書き込み対象のデータへ識別のための符号を付加した後、追記領域の末尾へ連続するように書込を行うことで、書込要求データを一回の連続領域書込コマンド(ISO18000-6 Type CのBlockWrite相当のコマンド)で実現する。

【図1】



**【特許請求の範囲】****【請求項 1】**

ＩＣタグへのデータ書き込みシステムにおいて、  
前記ＩＣタグのデータ記憶部は、予め複数の種類のデータが格納された基本領域と追記領域とを有し、

ＩＣタグへ書き込むデータと前記ＩＣタグのデータ記憶部に格納されたデータとを比較し、変更が生じたデータを抽出するミドルウェア装置と、

前記抽出された変更が生じたデータを、当該データの種類に応じた識別符号と共に前記追記領域に格納するリーダライタ装置と、  
を有することを特徴とするデータ書き込みシステム。

10

**【請求項 2】**

請求項 1 に記載のデータ書き込みシステムにおいて、

前記リーダライタ装置は、前記追記領域に格納する複数のデータを連続して位置するように書き込むことを特徴とするデータ書き込みシステム。

**【請求項 3】**

請求項 1 または 2 に記載のデータ書き込みシステムにおいて、

前記ミドルウェア装置は、

前記ＩＣタグ内に格納する前記識別符号と、前記ミドルウェア装置間で共有する前記対応管理テーブルと前記符号管理テーブルの組み合わせと、を利用し識別符号を付加することを特徴とする、データ書き込みシステム。

20

**【請求項 4】**

ＩＣタグへのデータ書き込み方法において、

前記ＩＣタグのデータ記憶部は、予め複数の種類のデータが格納された基本領域と追記領域とを有しており、

ミドルウェア装置により、ＩＣタグへ書き込むデータと前記ＩＣタグのデータ記憶部に格納されたデータとを比較し、変更が生じたデータを抽出するステップと、

リーダライタ装置により、前記抽出された変更が生じたデータを、当該データの種類に応じた識別符号と共に前記追記領域に格納するステップと、  
を有することを特徴とするデータ書き込み方法。

30

**【請求項 5】**

請求項 4 に記載のデータ書き込み方法において、

前記リーダライタ装置により前記追記領域に格納するステップは、前記追記領域に格納する複数のデータを書き込む場合、当該複数のデータを連続した位置に格納されるように書き込むことを特徴とするデータ書き込み方法。

**【発明の詳細な説明】****【技術分野】****【0001】**

本発明は、ＩＣタグに対するデータの書込処理に関する。

**【背景技術】****【0002】**

近年、物流分野をはじめとする種々の業界において、ＩＣタグを利用した効率的な業務処理システムや物品管理システムの導入が検討されている。ＩＣタグは、データが書き込まれたＩＣチップとアンテナで構成されており、ＵＨＦ帯などの電波を利用して、接触することなくデータの送受信が可能である。バーコードよりも多くの情報を表現することが可能であり、かつ離れた位置から情報を読み取ることが可能であるため、物流業界などでは、従来のバーコードに代わる管理手段として検討が進められている。また、ＩＣカードに比べて低価格での生産が可能であるため、電車の乗車券や娯楽施設の入場券など、使い捨て可能なチケットへの利用可能性も検討されている。

40

**【0003】**

こうした業務での利用においては、改札機などの業務アプリケーションが、ＩＣタグか

50

ら読み込んだデータの一部を変更し、変更後のデータをICタグへ書き込む処理が多く発生すると想定される。例えば、ICタグを電車の乗車券に利用する場合、乗客の入場時には、自動改札機がICタグ内のデータを読み込み、乗車券の有効性を確認した後に、入場駅の情報等を格納したデータでタグ内のメモリを更新する。また、乗客の出場時には、自動改札機がICタグ内のデータを読み込み、乗客が有効な入場駅、入場時刻から入場済みであることを確認した後、出場情報を書き込んだデータでタグ内のメモリを更新する。

【0004】

ところで、ICタグへデータを書き込む場合には、ICタグに対してデータの書き込みを要求する特定のコマンドを送信する必要がある。ICタグの仕様を定めるISO18000-6 Type Cでは、データの書込を指示する手段として、2バイトのデータを指定のアドレスへ書き込むためのWriteコマンドと、連続したデータを指定のアドレスへ書き込むためのBlockWriteコマンドの2つが規定されているが、当該コマンドの使用には、以下に述べる制限が加えられている。Writeコマンドを用いてデータを書き込む場合には、一度のコマンド/レスポンスで2バイトしか書き込むことができない。また、BlockWriteコマンドは2バイト以上のデータを書き込むことが可能だが、一度のコマンド/レスポンスでは、メモリ上で連続したデータしか書き込むことができない。したがって、メモリ上で分散配置されたデータを書き込む場合、複数回のコマンド/レスポンスが発生し、処理の完遂までに長い時間を要することが懸念される。

10

【0005】

また、ICタグはそれ自体に駆動電源を持たず、リーダ/ライタによる電磁誘導によって電力を供給されて駆動する。そのため、ICチップ内のメモリとしてはEEPROM (Electrically Erasable and Programmable ROM) 等の不揮発性メモリが用いられる。この種の不揮発性メモリは、一般に消去処理や書込処理に時間がかかることが知られている。

20

【0006】

上述した業務処理では、高速な処理が要求されることが多い。例えば、ICタグを電車の乗車券として用いた場合、自動改札機で乗客を混雑させないために、数百ミリ秒での高速処理が要求される。また、ライン上を流れる物品に貼り付けられたタグへ、完了した処理の情報を追記していく場合にも、同様の高速性が要求される。したがって、ICタグに対する書込処理の長さは、ICタグを利用した業務処理や物品管理などに支障を及ぼす可能性がある。

30

【0007】

ICタグへの書込処理を高速化するために、業務アプリケーションが、書き換えが必要なデータに限って書込処理を要求することで、書込コマンド (Write コマンドやBlockWriteコマンドなど) の実行回数を減らし、処理完遂に要するコマンド/レスポンスの回数を削減する方法が知られている。

【0008】

また、メモリへの書き込み時間を短縮するために、メモリの書込特性を利用する方法が知られている (例えば、特許文献1参照)。EEPROMなどの不揮発性メモリは、複数バイトからなるページ単位ならば、消去書き込み動作を一括して行うことが可能である、という特性をもつ。特許文献1では、メモリ上で分散配置されたデータに対して書込要求が発生した場合に、メモリ上のデータ配置を並び換えることで、可能な限りページ単位でデータを書き込み、処理の回数を削減する技術が提案されている。当該技術では、書込要求を受けたアドレス (論理アドレス) と、メモリ上の実際のアドレス (実アドレス) の対応表を管理することで、書込の度にデータの配置を変更することを可能としている。

40

【先行技術文献】

【特許文献】

【0009】

【特許文献1】特開2006-107379

【発明の概要】

50

## 【発明が解決しようとする課題】

## 【0010】

業務アプリケーションが、書き換えが必要なデータに限って書込処理を要求する場合には、書込データの位置や範囲を指定するために、ICタグ内のメモリに、データがどのように配置されているかを把握している必要がある。このため、ICタグの仕様が変更され、メモリのデータ配置が変更された場合に業務アプリケーションの改修を要し、システム全体の管理に係る負担が増大する可能性がある。

## 【0011】

また、特許文献1に記載の技術では、ICタグ毎に論理アドレスと実アドレスを対応付けるためのリンク・テーブルと呼ばれる対応表が無ければ、ICタグ内のデータを正しく読み書きすることができない。リンク・テーブルの管理をタグ自体に行わせることは、ICカードに比べて計算能力やメモリ容量に制限のあるICタグの場合には困難である。一方、リンク・テーブルを業務アプリケーションで管理する場合、各タグのリンク・テーブルを、そのタグを処理する可能性のある全ての業務アプリケーションで共有しなければならない。また、ICタグのメモリ内データ配置が変更される度に、共有されている当該タグのリンク・テーブル全てを更新する必要が発生する、という課題がある。

10

## 【課題を解決するための手段】

## 【0012】

本発明は、上記の事情を鑑みてなされたものであり、ICタグの仕様変更に伴う業務アプリケーションの改修頻度や、業務アプリケーション間で共有する情報の更新頻度を抑えながら、計算能力やメモリ容量に制限のあるICタグに対して、データ書込処理を高速化するための装置、方法、プログラムを提供する。

20

## 【0013】

具体的には、ICタグに対するデータ書込コマンドを発行する前に、直前にICタグから読み込んだデータと、これから書き込もうとするデータの差分を取り、書き換えの発生したデータに限定して書込要求コマンドを発行することで、書込要求コマンドの総実行回数を削減する。

## 【0014】

そして、ICタグ内の記憶領域を、各データの初期値を格納する基本領域と、新たに書き換えられたデータを格納する追記領域の2つに分割し、新たにデータを書き込む際には、書き込み対象のデータそれぞれに識別のための識別符号を付加した後、メモリ上で連続するように追記領域の末尾へ書込を行うことで、書込要求データを一回の連続領域書込コマンド(ISO18000-6 Type CのBlock Write相当のコマンド)で実現。

30

## 【0015】

ここで、ICタグ内の追記領域に格納された実データと、業務データとの対応付けを管理するために、本発明では、ICタグ内に格納する識別符号と、各ミドルウェア装置間で共有する対応管理テーブルと符号管理テーブルの組み合わせを利用する。

40

## 【発明の効果】

## 【0016】

本発明によれば、ICタグの仕様変更に伴う業務アプリケーションの改修頻度や、業務アプリケーション間で共有する情報の更新頻度を抑えることができ、データ書込処理を高速化できる。

## 【図面の簡単な説明】

## 【0017】

【図1】RFIDシステムの概略図。

【図2】RFIDシステムを構成する装置群の概略図。

【図3】業務データ書込バッファの概略図。

【図4】対応管理テーブルの概略図。

50

- 【図5】符号管理テーブルの概略図。
- 【図6】装置のハードウェア構成図。
- 【図7】初期化処理のシーケンス図。
- 【図8】ミドルウェア装置が初期化処理を実行する際のフローチャート。
- 【図9】読込処理および更新処理のシーケンス図。
- 【図10】ミドルウェア装置が読込処理を実行する際のフローチャート。
- 【図11】ミドルウェア装置が読込処理を実行する際のフローチャート。
- 【図12】ミドルウェア装置が書込処理を実行する際のフローチャート。
- 【図13】ICタグを乗車券として利用する場合のシステムの一例を示した図。
- 【図14】対応管理テーブルおよび符号管理テーブルの一例を示した図。 10
- 【図15】データ記憶部の一例を示した図。
- 【図16】業務データ読込バッファ，業務データ書込バッファ，業務データ差分バッファの一例を示した図。
- 【図17】データ記憶部の一例を示した図。
- 【図18】データ記憶部の一例を示した図。
- 【図19】データ記憶部の一例を示した図。
- 【発明を実施するための形態】
- 【0018】
- 以下，図面を用いて本発明の実施の形態について説明する。なお，これにより本発明が 20  
限定されるものではない。
- 【実施例1】
- 【0019】
- 図1は，本発明の一実施形態であるRFIDシステム10の概略図である。
- 【0020】
- 図示するように，RFIDシステム10は，業務アプリケーション装置1-1～1-n（nは1以上の自然数）と，ミドルウェア装置2-1～2-nと，リーダ/ライタ装置3-1～3-nと，ICタグ4と，を備える。リーダライタ装置3-i（iは1以上n以下の自然数）とICタグ4は，UHF帯などの無線を用いて相互に通信を行う。また，ミドルウェア装置2-iとリーダライタ装置3-iは，USBやネットワークを介して相互に通信を行う。また，業務アプリケーション装置1-iとミドルウェア装置2-iは，USBやネットワークを介して相互に通信を行って良い。また，ミドルウェア装置2-iがソフトウェアとして業務アプリケーション装置1-iの内部に含まれる場合には，装置内部の信号線を用いて相互に通信を行っても良い。 30
- 【0021】
- 図2は，業務アプリケーション装置1と，ミドルウェア装置2と，リーダライタ装置3と，ICタグ4と，の機能構成の例を示す概略図である。ここで，業務アプリケーション装置1と，ミドルウェア装置2と，リーダライタ装置3と，は，それぞれ業務アプリケーション装置1-1～1-nと，ミドルウェア装置2-1～2-nと，リーダライタ装置3-1～3-nに共通する構造を記している。
- 【0022】 40
- 図示するように，業務アプリケーション装置1は，業務処理部101と，ミドルウェア通信部102と，を備える。
- 【0023】
- 業務処理部101は，業務アプリケーションの担当する業務を処理する。例えば，自動改札機では，ICタグから読み取った乗車券データの検証や，改札時の時刻，設置される駅などの情報に基づいて，データの上書きを指示する，といった処理が考えられる。
- 【0024】
- ミドルウェア通信部102は，ミドルウェア装置2との間でのデータの送受信を制御する。例えば，ミドルウェア通信部102としては，TCP/IP通信を可能とするプロトコルスタックや，プロセス間通信を可能とするモジュールなどが考えられる。 50

## 【 0 0 2 5 】

ミドルウェア装置 2 は、アプリケーション通信部 2 0 1 と、アプリケーションコマンド処理部 2 0 2 と、データ変換部 2 0 3 と、差分計算部 2 0 4 と、追記領域末尾ポインタ 2 0 5 と、業務データ差分バッファ 2 0 6 1 と、業務データ書込バッファ 2 0 6 2 と、業務データ読込バッファ 2 0 6 3 と、基本領域書込バッファ 2 0 6 4 と、基本領域読込バッファ 2 0 6 5 と、追記領域書込バッファ 2 0 6 6 と、追記領域読込バッファ 2 0 6 7 と、対応管理テーブル 2 0 7 と、符号管理テーブル 2 0 8 と、I C タグコマンド処理部 2 0 9 と、リーダライタ通信部 2 1 0 と、を備える。

## 【 0 0 2 6 】

アプリケーション通信部 2 0 1 は、業務アプリケーション装置 1 との間でのデータの送受信を制御する。例えば、アプリケーション通信部 2 0 1 としては、T C P / I P 通信を可能とするプロトコルスタックや、プロセス間通信を可能とするモジュールなどが考えられる。

10

## 【 0 0 2 7 】

アプリケーションコマンド処理部 2 0 2 は、業務アプリケーション装置 1 からアプリケーション通信部 2 0 1 を介して受信したアプリケーションコマンドのデコード処理や、業務アプリケーション装置 1 へ送信するアプリケーションレスポンスのエンコード処理を実行する。アプリケーションコマンドは、業務アプリケーション装置 1 が I C タグ 4 に対してデータの読み書き等を指示するために用いるコマンドで、アプリケーションレスポンスは、アプリケーションコマンドの実行結果や、I C タグから読み取ったデータの内容を含み、業務アプリケーション装置 1 に対して送信されるメッセージである。

20

## 【 0 0 2 8 】

データ変換部 2 0 3 は、アプリケーションコマンド処理部 2 0 2 によってデコードされたアプリケーションコマンドの内容に応じて、各種のバッファ 2 0 6 1 ~ 2 0 6 7 に対するデータの読み書きや、対応管理テーブル 2 0 7 および符号管理テーブル 2 0 8 からのデータ取得や、差分計算部 2 0 4 に対する差分計算の実行要求などを行った後、I C タグコマンド処理部 2 0 9 に対して I C タグコマンドの実行を要求し、I C タグ 4 に対するデータの読み書きを実行する。

## 【 0 0 2 9 】

また、データ変換部 2 0 3 は、I C タグコマンド処理部 2 0 9 から、リーダライタ 3 経由で I C タグ 4 からの応答が受信されたことを通知されると、応答の内容に応じて各種のバッファ 2 0 6 1 ~ 2 0 6 7 に対するデータの読み書きや、対応管理テーブル 2 0 7 および符号管理テーブル 2 0 8 からのデータ取得などを行った後、アプリケーション通信部 2 0 1 を介して、業務アプリケーション装置 1 に対してアプリケーションレスポンスを送信する。

30

## 【 0 0 3 0 】

差分計算部 2 0 4 は、業務データ書込バッファ 2 0 6 2 に格納されたデータと、業務データ読込バッファ 2 0 6 3 に格納されたデータと、の差分（異なっているデータの抽出）を計算し、結果を業務データ差分バッファ 2 0 6 1 へ格納する。

## 【 0 0 3 1 】

追記領域末尾ポインタ 2 0 5 は、I C タグ 4 内の追記領域 4 0 3 2 の末尾のアドレスを含み、I C タグ毎にその値を対応付けて格納する。

40

## 【 0 0 3 2 】

業務データ差分バッファ 2 0 6 1 は、上述したように、差分計算部 2 0 4 によって計算された差分データを格納するための記憶領域である。バッファの構造は後述する業務データ書込バッファ 2 0 6 2 と同様であるため、説明を省略する。

## 【 0 0 3 3 】

業務データ書込バッファ 2 0 6 2 は、業務アプリケーション装置 1 から受信した書込アプリケーションコマンドに含まれる書込対象業務データを一時的に格納するための記憶領域である。業務データ書込バッファ 2 0 6 2 の一例について、図 3 を用いて説明する。

50

## 【 0 0 3 4 】

図 3 は、業務データ書込バッファ 2 0 6 2 の一例を示した図である。図 3 に示すように、業務データ書込バッファ 2 0 6 2 はテーブル型のデータ構造を取る。各レコード（行）には、「データ名」フィールド 2 0 6 2 - 1 と、「データ」フィールド 2 0 6 2 - 2 が定義されている。「データ名」フィールド 2 0 6 2 - 1 には、業務アプリケーション装置 1 から受信した書込対象業務データの名前を含み、「データ」フィールド 2 0 6 2 - 2 には、当該業務データの内容が格納される。なお、以後、業務データ書込バッファ 2 0 6 2 内で「データ名」フィールド 2 0 6 2 - 1 に値 X を持つ行を、業務データ書込バッファ 2 0 6 2 の X 行、と呼ぶこととする。

## 【 0 0 3 5 】

業務データ読込バッファ 2 0 6 3 は、読込アプリケーションコマンドを実行した後、業務アプリケーション装置 1 へレスポンスとして返却する業務データを、一時的に格納するための記憶領域である。業務データ読込バッファ 2 0 6 3 のデータ構造は、業務データ書込バッファ 2 0 6 2 と同様のため、説明を省略する。

## 【 0 0 3 6 】

基本領域書込バッファ 2 0 6 4 は、I C タグ 4 のデータ記憶部 4 0 3 内の基本領域 4 0 3 1 へ書き込む予定のデータを一時的に格納するための記憶領域である。

## 【 0 0 3 7 】

基本領域読込バッファ 2 0 6 5 は、I C タグ 4 のデータ記憶部 4 0 3 内の基本領域 4 0 3 1 から読み込んだデータを一時的に格納するための記憶領域である。

## 【 0 0 3 8 】

追記領域書込バッファ 2 0 6 6 は、I C タグ 4 のデータ記憶部 4 0 3 内の追記領域 4 0 3 2 へ書き込む予定のデータを一時的に格納するための記憶領域である。

## 【 0 0 3 9 】

追記領域読込バッファ 2 0 6 7 は、I C タグ 4 のデータ記憶部 4 0 3 内の追記領域 4 0 3 2 から読み込んだデータを一時的に格納するための記憶領域である。

## 【 0 0 4 0 】

各バッファ 2 0 6 1 ~ 2 0 6 7 に格納されるデータの値は、後述する処理フローを実施する中で、ミドルウェア装置 2 - 1 ~ 2 - n それぞれによって異なるものとなる。

## 【 0 0 4 1 】

対応管理テーブル 2 0 7 は、業務アプリケーション装置 1 が扱う業務データと、その業務データが基本領域 4 0 3 1 上のどの位置に記載されるかを示すオフセットと、その業務データのサイズ、の 3 つの値を対応付けるテーブルである。対応管理テーブル 2 0 7 に含まれる値は、ミドルウェア装置 2 - 1 ~ 2 - n 間で同じものを、R F I D システム 1 0 の運用開始前に共有しておく。対応管理テーブル 2 0 7 の一例について、図 4 を用いて説明する。

## 【 0 0 4 2 】

図 4 は、対応管理テーブル 2 0 7 の一例を示した図である。対応管理テーブル 2 0 7 は、各レコード（行）に対して、「データ名」フィールド 2 0 7 - 1 と、「オフセット」フィールド 2 0 7 - 2 と、「データ長」フィールド 2 0 7 - 3 と、が定義されている。「データ名」フィールド 2 0 7 - 1 には、業務アプリケーション装置 1 が扱う業務データの名前が格納され、同じレコード（行）内の「オフセット」フィールド 2 0 7 - 2 には、当該業務データが基本領域 4 0 3 1 内のどの位置に格納されるかを示す物理アドレス値が格納される。また、同じレコード内の「データ長」フィールド 2 0 8 - 3 には、当該業務データがデータ記憶部 4 0 3 内で占めるデータサイズが格納される。

## 【 0 0 4 3 】

符号管理テーブル 2 0 8 は、業務アプリケーション装置 1 が扱う業務データと、その業務データのデータ長と、追記領域 4 0 3 2 へ当該業務データを書き込む際に利用する識別符号と、の組を対応付けるテーブルである。符号管理テーブル 2 0 8 に含まれる値は、ミ

10

20

30

40

50

ドルウェア装置 2 - 1 ~ 2 - n 間で同じものを，RFIDシステム 10 の運用開始前に共有しておく。符号管理テーブル 208 の一例について，図 5 を用いて説明する。

【0044】

図 5 は，符号管理テーブル 208 の一例を示した図である。符号管理テーブル 208 は，各レコード（行）に対して，「データ名」フィールド 208 - 1 と「識別符号」フィールド 208 - 2 と，が定義されている。「データ名」フィールド 208 - 1 には，業務アプリケーション装置 1 が扱う業務データの名前が格納され，同じレコード（行）内の「識別符号」フィールド 208 - 2 には，当該業務データが IC タグ 4 の追記領域 4032 内に書き込まれる際，当該業務データとともに書き込まれる識別のための符号が格納される。「データ名」NULL に対応する「識別符号」c0 は，追記領域内でのデータの終端を示すために使用する特別な識別符号である。

10

【0045】

本実施例では，識別符号として用いる符号の種類について制限を加えない。固定長の符号と可変長の符号のどちらを用いてもよいが，個々のデータの書き換え頻度がある程度正確に予測できている場合には，書き換え頻度に基づいた Huffman 符号などの可変長符号を利用することで，符号がデータ記憶部 403 内に占める割合を減らすことができ，有効となる可能性がある。

【0046】

IC タグコマンド処理部 209 は，データ変換部 203 から要求を受けた IC タグコマンドを実行するために，リーダライタ通信部 210 を介してリーダライタ装置 3 へ IC タグコマンドを送信する。また，リーダライタ通信部 210 を介して受信した IC タグレスポンスを，データ変換部 203 へ通知する。IC タグコマンドは，リーダライタ装置 3 が IC タグ 4 に対してデータの読み書き等を指示するために用いるコマンドで，IC タグレスポンスは，IC タグコマンドの実行結果や，IC タグから読み取ったデータの内容を含むメッセージである。

20

【0047】

リーダライタ通信部 210 は，リーダライタ装置 3 との間でのデータの送受信を制御する。例えば，USB デバイスを制御するデバイスドライバや，TCP/IP 通信を可能とするプロトコルスタックなどが考えられる。

【0048】

リーダライタ装置 3 は，ミドルウェア通信部 301 と，IC タグ通信部 302 と，を備える。

30

【0049】

ミドルウェア通信部 301 は，ミドルウェア装置 2 との間でのデータの送受信を制御する。例えば，USB デバイスを制御するデバイスドライバや，TCP/IP 通信を可能とするプロトコルスタックなどが考えられる。

【0050】

IC タグ通信部 302 は，IC タグ 4 との間でのデータの送受信を制御する。例えば，通信に使う周波数の変更や，アンテナの出力制御や，変調による IC タグコマンドの送信制御や，IC タグレスポンスの受信制御などを行う。

40

【0051】

IC タグ 4 は，リーダライタ通信部 401 と，コマンド処理部 402 と，データ記憶部 403 と，を備える。

【0052】

リーダライタ通信部 401 は，リーダライタ装置 3 との間でのデータの送受信を制御する。例えば，通信に使う周波数の変更や，アンテナの出力制御や，変調による IC タグレスポンスの送信制御や，IC タグコマンドの受信制御などを行う。

【0053】

コマンド処理部 402 は，リーダライタ装置 3 からリーダライタ通信部 401 を介して受信した IC タグコマンドのデコード処理や，リーダライタ装置 3 へ送信する IC タグレ

50

スポンスのエンコード処理を実行する。

【0054】

データ記憶部403は、ICタグ4固有のデータを格納し、コマンド処理部402によってデコードされたコマンドの内容によって、データの書込と、消去と、読込と、等を制御する。データ記憶部403は、基本領域4031と追記領域4032とで構成される。

【0055】

ここで、ミドルウェア装置2のICタグコマンド処理部209は、データ記憶部403に対して読み書きされるデータが、基本領域4031と追記領域4032のどちらに属するデータであるかを判別可能であるとする。このことは例えば、基本領域4031と追記領域4032のそれぞれがデータ記憶部403中に占める割合や、そのメモリ上の位置に関する情報を、各ミドルウェア装置2間で事前共有することで実現できる。

10

【0056】

基本領域4031は、ICタグ4の初期化時にデータを書き込まれる領域である。

【0057】

また、追記領域4032は、ICタグ4の初期化後、データの書き換えが発生した際に、データを書き込まれる領域である。基本領域4031と追記領域4032とは、データ記憶部内で異なるメモリを用いて実現されていても良いし、連続した領域を仮想的に切り離したもので良い。

【0058】

以上に記載した業務アプリケーション装置1と、ミドルウェア装置2と、リーダライタ装置3とは、例えば、図6(コンピュータ9の概略図)に示すような、CPU(Central Processing Unit)901と、メモリ902と、HDD(Hard Disk Drive)等の外部記憶装置903と、CD-ROM(Compact Disk Read Only Memory)やDVD-ROM(Digital Versatile Disk Read Only Memory)などの可搬性を有する記憶媒体904から情報を読み出す読取装置905と、キーボードやマウス等の入力装置906と、ディスプレイなどの出力装置907と、通信ネットワークに接続するためのNIC(Network Interface Card)などの送受信装置908と、を備えた一般的なコンピュータ9で実現できる。例えば、各種バッファ2061~2067やデータ記憶部403は、CPU901がメモリ902又は外部記憶装置903を利用することにより実現可能である。また、業務処理部101やデータ変換部203等、特定の処理や制御を行う機能については、外部記憶装置903に記憶されている所定のプログラムをメモリ902にロードしてCPU901で実行することで実現可能である。また、リーダライタ通信部210等、通信制御を行う機能については、CPU901が送受信装置908を利用することで実現可能である。この所定のプログラムは、読取装置905を介して記憶媒体904から、あるいは、送受信装置908を介してネットワークから、外部記憶装置903にダウンロードされ、それから、メモリ902上にロードされてCPU901により実行されるようにしてもよい。また、読取装置905を介して記憶媒体904から、あるいは、送受信装置908を介してネットワークから、メモリ902上に直接ロードされ、CPU901により実行されるようにしてもよい。

20

30

40

【0059】

次に、図7から図12を用いて、業務アプリケーション装置1-1により初期化されたICタグ4が、別の業務アプリケーション装置1-2によってデータを読み書きされる場合のシーケンスについて説明する。

【0060】

図7は、業務アプリケーション装置1-1がICタグ4の初期化を行う場合のシーケンスを記載した図である。

【0061】

ICタグを乗車券として利用する場合、当該シーケンスは、発券機(業務アプリケーション装置)による発券シーケンスに対応する。すなわち、データが書き込まれていない、

50

発券前の乗車券 I C タグに対して，その発券機が設置された駅の情報や，発券時刻や，当該乗車券の有効範囲，有効期限などを，I C タグ内のデータ記憶領域に書き込むシーケンスに対応する。

【 0 0 6 2 】

業務アプリケーション装置 1 - 1 は，ユーザから入出力装置を介して初期化要求操作を受け付けると，ミドルウェア通信部 1 0 2 が，ミドルウェア装置 2 - 1 に対して初期化要求アプリケーションコマンドを送信する（S 1 0 1）。当該コマンドには，パラメータとして，当該業務アプリケーション装置 1 - 1 に関連する設定情報が含まれる。乗車券の発券業務の場合，当該パラメータには，業務アプリケーション装置が設置された駅の情報や，発券時刻や，当該乗車券の有効範囲，有効期限などが含まれる。

10

【 0 0 6 3 】

ミドルウェア装置 2 - 1 は，装置内のアプリケーション通信部 2 0 1 が，業務アプリケーション装置 1 - 1 から送信された初期化要求アプリケーションコマンドを受信すると，受け取ったパラメータ内の業務データを，I C タグへ書き込むための処理を実行する（初期化前処理，S 1 0 2）。

【 0 0 6 4 】

以下，当該処理 S 1 0 2 の詳細について，図 8 を用いて説明する。

【 0 0 6 5 】

図 8 は，ミドルウェア装置 2 による初期化処理の詳細を表したフローチャートである。

【 0 0 6 6 】

ミドルウェア装置 2 - 1 内では，アプリケーション通信部 2 0 1 が，業務アプリケーション装置 1 - 1 から送信された初期化要求アプリケーションコマンドを受信すると（S 1 0 2 - 1），アプリケーションコマンド処理部 2 0 2 が，当該コマンドをデコードし，当該コマンドに含まれるパラメータ（業務データ）を取得しデータ変換部 2 0 3 へ受け渡す（S 1 0 2 - 2）。

20

【 0 0 6 7 】

データ変換部 2 0 3 は，まず，業務データ書込バッファ 2 0 6 2 と，基本領域書込バッファ 2 0 6 4 と，を空にする（S 1 0 2 - 3）。

【 0 0 6 8 】

次に，データ変換部 2 0 3 は，取得した業務データを，業務データ書込バッファ 2 0 6 2 へ格納する（S 1 0 2 - 4）。

30

【 0 0 6 9 】

続いて，データ変換部 2 0 3 は，C P U 9 0 1 内のレジスタまたはメモリ 9 0 2 または外部記憶装置 9 0 3 上に，4 つの変数 M ， N A M E ， O F F S E T ， D A T A 用の領域を確保し，それぞれ値 0 で初期化する（S 1 0 2 - 5）。

【 0 0 7 0 】

次に，データ変換部 2 0 3 は，対応管理テーブル 2 0 7 を参照し，当該テーブルの M 行目に値が存在するか否かを判定する（S 1 0 2 - 6）。

【 0 0 7 1 】

当該判定の結果，当該テーブルの M 行目に値が存在する場合（S 1 0 2 - 6 で Y E S），データ変換部 2 0 3 は，対応管理テーブル 2 0 7 の M 行目から，「データ名」フィールド 2 0 7 - 1 と「オフセット」フィールド 2 0 7 - 2 の値を取得し，それぞれの値で変数 N A M E と変数 O F F S E T の値を上書きする（S 1 0 2 - 7）。

40

【 0 0 7 2 】

例として，対応管理テーブル 2 0 7 が，図 4 のような値が書き込まれた状態であった場合を考える。このとき，M の値が 2 であったとすると，当該テーブルの 2 行目にはデータが格納されているため，変数 N A M E にはデータ名「d 2」が代入され，変数 O F F S E T にはオフセット値「( k ， 1 0 )」が代入される。

【 0 0 7 3 】

続いて，データ変換部 2 0 3 は，業務データ書込バッファ 2 0 6 2 の N A M E 行から，

50

「データ」フィールド 2062 - 2 の値を取得し、当該データの値で変数 DATA を上書きする (S102 - 8)。

【0074】

次に、データ変換部 203 は、基本領域書込バッファ 2064 内の、変数 OFFSET で指定される番地へ、変数 DATA 内のデータを格納する (S102 - 9)。

【0075】

続いて、データ変換部 203 は、変数 M の値をインクリメントし (S102 - 10)、対応管理テーブル 207 の M 行目に値が存在するか否かを判定する処理 (S102 - 6) へ処理を戻す。

【0076】

処理 S102 - 6 にて、対応管理テーブル 207 の M 行目に値が存在しない、すなわち、対応管理テーブル 207 内の全ての行に対する走査が完了すると、データ変換部 203 は、基本領域書込バッファ 2064 内のデータを IC タグ 4 の基本領域 4031 へ書き込むよう、リーダライタ装置 3 - 1 に対してデータ書込要求メッセージを送信し (S102 - 11)、リーダライタ装置 3 - 1 からの返信を待ち受ける状態へ遷移する (S102 - 12)。当該データ書込要求メッセージはパラメータを含み、当該パラメータによって書込位置と書き込むデータの内容を指示するものとする。ここでは書込位置として基本領域 4031 の先頭を指定し、書き込むデータの内容として基本領域書込バッファ 2064 内のデータを指定する。

【0077】

以上が、ミドルウェア装置 2 - 1 による、初期化前処理 S102 の詳細である。

【0078】

図 7 に戻り、上述の初期化前処理 S102 の結果、ミドルウェア装置 2 - 1 がリーダライタ装置 3 - 1 へデータ書込要求メッセージを送信すると (S103)、リーダライタ装置 3 - 1 は当該書込要求に従って IC タグ 4 内の基本領域 4031 先頭へ指定されたデータの書込を実行する (S104)。

【0079】

IC タグ 4 内への書込が成功すると (S105)、リーダライタ装置 3 - 1 はミドルウェア装置 2 - 1 に対して基本領域 4031 への書込に成功した旨を通知する (S106)。

【0080】

ミドルウェア装置 2 - 1 は、リーダライタ装置 3 - 1 から、基本領域 4031 への書込に成功した旨を通知されると、追記領域 4032 を初期化するための処理に移る (初期化中処理, S107)。

【0081】

以下、当該処理 S107 の詳細について、再び図 8 を用いて説明する。

【0082】

ミドルウェア装置 2 - 1 内では、リーダライタ通信部 210 がリーダライタ装置 3 - 1 からメッセージを受信すると、IC タグコマンド処理部 209 が、当該メッセージをデコードし、当該メッセージが、基本領域 4031 への書込に成功したことを意味するメッセージであると解釈する (S102 - 12 で YES)。

【0083】

すると、ミドルウェア装置 2 - 1 内のデータ変換部 203 は、追記領域の末尾を示す識別符号 c0 を IC タグ 4 の追記領域 4032 へ書き込むよう、リーダライタ装置 3 - 1 に対してデータ書込要求メッセージを送信し (S107 - 1)、リーダライタ装置 3 - 1 からの返信を待ち受ける状態へ遷移する (S107 - 2)。当該データ書込要求メッセージはパラメータを含み、当該パラメータによって書込位置と書き込むデータの内容を指示するものとする。ここでは書込位置として追記領域 4032 の先頭を指定し、書き込むデータの内容として識別符号 c0 を指定する。

【0084】

10

20

30

40

50

以上が、ミドルウェア装置 2 - 1 による、初期化中処理 S 1 0 7 の詳細である。

【 0 0 8 5 】

図 7 に戻り、上述の初期化中処理 S 1 0 7 の結果、ミドルウェア装置 2 - 1 がリーダライタ装置 3 - 1 へデータ書込要求メッセージを送信すると ( S 1 0 8 )、リーダライタ装置 3 - 1 は当該書込要求に従って IC タグ 4 内の追記領域 4 0 3 2 先頭へ指定されたデータの書込を実行する ( S 1 0 9 )。

【 0 0 8 6 】

IC タグ 4 内への書込が成功すると ( S 1 1 0 )、リーダライタ装置 3 - 1 はミドルウェア装置 2 - 1 に対して追記領域 4 0 3 2 への書込に成功した旨を通知する ( S 1 1 1 )

10

【 0 0 8 7 】

ミドルウェア装置 2 - 1 は、リーダライタ装置 3 - 1 から、追記領域 4 0 3 2 への書込に成功した旨を通知されると、初期化処理が成功したことを業務アプリケーション装置 1 - 1 へ通知するための処理に移る ( 初期化後処理、S 1 1 2 )。

【 0 0 8 8 】

以下、当該処理 S 1 1 2 の詳細について、再び図 8 を用いて説明する。

【 0 0 8 9 】

ミドルウェア装置 2 - 1 内では、リーダライタ通信部 2 1 0 がリーダライタ装置 3 - 1 からメッセージを受信すると、IC タグコマンド処理部 2 0 9 が、当該メッセージをデコードし、当該メッセージが、追記領域 4 0 3 2 への書込に成功したことを意味するメッセージであると解釈する ( S 1 0 7 - 2 で Y E S )。

20

【 0 0 9 0 】

すると、ミドルウェア装置 2 - 1 内のアプリケーションコマンド処理部 2 0 2 が、初期化要求アプリケーションコマンドの実行が成功したことを示す初期化要求アプリケーション成功応答を生成し、アプリケーション通信部 2 0 1 を介して、業務アプリケーション装置 1 - 1 へ送信する ( S 1 1 2 - 1 )。

【 0 0 9 1 】

以上が、ミドルウェア装置 2 - 1 による、初期化後処理 S 1 1 2 の詳細である。

【 0 0 9 2 】

図 7 に戻り、上述の初期化後処理 S 1 1 2 の結果、ミドルウェア装置 2 - 1 が業務アプリケーション装置 1 - 1 へ初期化要求アプリケーション成功応答を送信し ( S 1 1 3 )、IC タグ 4 に対する初期化処理は正常に完了する。

30

【 0 0 9 3 】

一方、図 8 において、S 1 0 2 - 1 2 または S 1 0 7 - 2 で、ミドルウェア装置 2 - 1 が、リーダライタ装置 3 - 1 から何らかのエラーメッセージを受信した場合には ( S 1 0 2 - 1 2 または S 1 0 7 - 2 で N O )、ミドルウェア装置 2 - 1 内のアプリケーションコマンド処理部 2 0 2 が、初期化要求アプリケーションコマンドの実行に失敗したことを示す初期化要求アプリケーション失敗応答を生成し、アプリケーション通信部 2 0 1 を介して、業務アプリケーション装置 1 - 1 へ送信する ( S 1 0 7 - 3 )。

【 0 0 9 4 】

40

図 9 は、業務アプリケーション装置 1 - 2 が IC タグ 4 からデータを読み込み、業務処理を行った後、データの更新を行う場合のシーケンスを記載した図である。

【 0 0 9 5 】

IC タグを乗車券として利用する場合、当該シーケンスは、自動改札機 ( 業務アプリケーション装置 ) による入出場シーケンスに対応する。すなわち、乗車券データが書き込まれた、入場前もしくは出場前の乗車券 IC タグに対して、その乗車券が発券された駅の情報や、発券時刻や、当該乗車券の有効範囲、有効期限などを、IC タグ内のデータ記憶領域から読み込み、有効な乗車券であることを確認した後、入場済もしくは出場済であることを示すデータでタグ内のデータを更新するシーケンスに対応する。

【 0 0 9 6 】

50

業務アプリケーション装置 1 - 2 は、ユーザから入出力装置を介して初期化要求操作を受け付けると、ミドルウェア通信部 1 0 2 が、ミドルウェア装置 2 - 2 に対して読込要求アプリケーションコマンドを送信する ( S 2 0 1 )。

【 0 0 9 7 】

ミドルウェア装置 2 - 1 は、装置内のアプリケーション通信部 2 0 1 が、業務アプリケーション装置 1 - 1 から送信された読込要求アプリケーションコマンドを受信すると、受け取ったパラメータ内の業務データを、I C タグへ書き込むための処理を実行する ( 読込前処理 , S 2 0 2 )。

【 0 0 9 8 】

以下、当該処理 S 2 0 2 の詳細について、図 1 0 を用いて説明する。

10

【 0 0 9 9 】

図 1 0 は、ミドルウェア装置 2 による読込処理の前半部を表したフローチャートである。

【 0 1 0 0 】

ミドルウェア装置 2 - 2 内では、アプリケーション通信部 2 0 1 が、業務アプリケーション装置 1 - 2 から送信された読込要求アプリケーションコマンドを受信すると、アプリケーションコマンド処理部 2 0 2 が当該コマンドをデコードし、データ変換部 2 0 3 へ受け渡す ( S 2 0 2 - 1 )。

【 0 1 0 1 】

データ変換部 2 0 3 は、まず、業務データ読込バッファ 2 0 6 3 と、基本領域読込バッファ 2 0 6 5 と、追記領域読込バッファ 2 0 6 7 を空にする ( S 2 0 2 - 3 )。

20

【 0 1 0 2 】

次に、データ変換部 2 0 3 は、I C タグ 4 の基本領域 4 0 3 1 および追記領域 4 0 3 2 に格納されたデータを読み込むよう、リーダライタ装置 3 - 2 に対してデータ読込要求メッセージを送信し ( S 2 0 2 - 3 )、リーダライタ装置 3 - 2 からの返信を待ち受ける状態へ遷移する ( S 2 0 2 - 4 )。

【 0 1 0 3 】

以上が、ミドルウェア装置 2 - 2 による、読込前処理 S 2 0 2 の詳細である。

【 0 1 0 4 】

図 9 に戻り、上述の読込前処理 S 2 0 2 の結果、ミドルウェア装置 2 - 2 がリーダライタ装置 3 - 2 へデータ読込要求メッセージを送信すると ( S 2 0 3 )、リーダライタ装置 3 - 2 は当該読込要求に従って I C タグ 4 内の基本領域 4 0 3 1 および追記領域 4 0 3 2 に格納されたデータの読込を実行する ( S 2 0 4 )。

30

【 0 1 0 5 】

I C タグ 4 からの読込が成功すると ( S 2 0 5 )、リーダライタ装置 3 - 2 はミドルウェア装置 2 - 2 に対して基本領域 4 0 3 1 および追記領域 4 0 3 2 から読み込んだデータを送信する ( S 2 0 6 )。

【 0 1 0 6 】

ミドルウェア装置 2 - 2 は、リーダライタ装置 3 - 2 から、基本領域 4 0 3 1 および追記領域 4 0 3 2 から読み込んだデータを受信すると、読込データを業務アプリケーション装置 1 - 2 へ提供するための処理に移る ( 読込後処理 , S 2 0 7 )。

40

【 0 1 0 7 】

以下、当該処理 S 2 0 7 の詳細について、再び図 1 0 を用いて説明する。

【 0 1 0 8 】

ミドルウェア装置 2 - 2 内では、リーダライタ通信部 2 1 0 がリーダライタ装置 3 - 2 からメッセージを受信すると、I C タグコマンド処理部 2 0 9 が、当該メッセージをデコードし、当該メッセージに含まれるパラメータ ( 基本領域 4 0 3 1 および追記領域 4 0 3 2 から読み込んだデータ ) を取得する。

【 0 1 0 9 】

ここで、正常にデータが取得できなかった場合、もしくは一定時間、一定回数のリトラ

50

イを経てもリーダーライタ装置 3 - 2 から応答が無かった場合は ( S 2 0 2 - 4 で N O ) , ミドルウェア装置 2 - 2 内のアプリケーションコマンド処理部 2 0 2 が , 読込要求アプリケーションコマンドの実行に失敗したことを示す読込要求アプリケーション失敗応答を生成し , アプリケーション通信部 2 0 1 を介して , 業務アプリケーション装置 1 - 2 へ送信する ( S 2 0 2 - 5 ) 。

【 0 1 1 0 】

一方 , 正常にデータが取得できた場合 ( S 2 0 2 - 4 で Y E S ) , I C タグコマンド処理部 2 0 9 は , 当該データをデータ変換部 2 0 3 へ受け渡す。データ変換部 2 0 3 は , 受け取ったデータの内 , 基本領域 4 0 3 1 から読み込まれたデータを基本領域読込バッファ 2 0 6 5 へ格納し ( S 2 0 7 - 1 ) , 追記領域 4 0 3 2 から読み込まれたデータを追記領域読込バッファ 2 0 6 7 へ格納する ( S 2 0 7 - 2 ) 。

10

【 0 1 1 1 】

まず , データ変換部 2 0 3 は , 基本領域読込バッファ 2 0 6 5 内のデータ変換処理を開始する。データ変換部 2 0 3 は , C P U 9 0 1 内のレジスタまたはメモリ 9 0 2 または外部記憶装置 9 0 3 上に , 5 つの変数 M , N A M E , O F F S E T , L E N , D A T A 用の領域を確保し , それぞれ値 0 で初期化する ( S 2 0 7 - 3 ) 。

【 0 1 1 2 】

次に , データ変換部 2 0 3 は , 対応管理テーブル 2 0 7 を参照し , 当該テーブルの M 行目に値が存在するか否かを判定する ( S 2 0 7 - 4 ) 。

【 0 1 1 3 】

当該判定の結果 , 当該テーブルの M 行目に値が存在する場合 , データ変換部 2 0 3 は , 対応管理テーブル 2 0 7 の M 行目から , 「データ名」フィールド 2 0 7 - 1 と , 「オフセット」フィールド 2 0 7 - 2 と , 「データ長」フィールド 2 0 7 - 3 と , の値を取得し , それぞれの値で変数 N A M E と変数 O F F S E T と変数 L E N の値を上書きする ( S 2 0 7 - 5 ) 。

20

【 0 1 1 4 】

例として , 対応管理テーブル 2 0 7 が , 図 4 のような値が書き込まれた状態であった場合を考える。このとき , M の値が 2 であったとすると , 当該テーブルの 2 行目にはデータが格納されているため , 変数 N A M E にはデータ名 「 d 2 」 が代入され , 変数 O F F S E T にはオフセット値 「 ( k , 1 0 ) 」 が代入され , 変数 L E N にはデータ長 「 6 」 が代入される。

30

【 0 1 1 5 】

続いて , データ変換部 2 0 3 は , 基本領域読込バッファ 2 0 6 5 から , O F F S E T 番地を始点として長さ L E N の領域に格納されたデータを取得し , 変数 D A T A へ代入する ( S 2 0 7 - 6 ) 。

【 0 1 1 6 】

次に , データ変換部 2 0 3 は , 業務データ読込バッファ 2 0 6 3 の N A M E 行へ , 変数 D A T A 内のデータを格納する ( S 2 0 7 - 7 ) 。

【 0 1 1 7 】

続いて , データ変換部 2 0 3 は , 変数 M の値をインクリメントし ( S 2 0 7 - 8 ) , 対応管理テーブル 2 0 7 の M 行目に値が存在するか否かを判定する処理 ( S 2 0 7 - 4 ) へ戻る。

40

【 0 1 1 8 】

処理 S 2 0 7 - 4 にて , 対応管理テーブル 2 0 7 の M 行目に値が存在しない , すなわち , 対応管理テーブル 2 0 7 内の全ての行に対する走査が完了すると , データ変換部 2 0 3 は , 追記領域読込バッファ 2 0 6 7 内のデータ変換処理に移る ( S 2 0 7 - 9 ) 。

以下 , 当該処理の詳細について , 図 1 1 を用いて説明する。

【 0 1 1 9 】

図 1 1 は , ミドルウェア装置 2 による読込処理の後半部を表したフローチャートである

50

## 【0120】

ミドルウェア装置 2 - 2 内のデータ変換部 203 は、まず、変数 M の値を 0 にリセットするとともに、CPU 901 内のレジスタまたはメモリ 902 または外部記憶装置 903 上に、変数 CODE 用の領域を確保し、値 0 で初期化する (S 207 - 10)。

## 【0121】

次に、データ変換部 203 は、変数 M の値が、追記領域 4032 の最大サイズを示す MAX\_\_APPEND を超えているか否かを確認する (S 207 - 11)。M の値が MAX\_\_APPEND を超えている場合 (S 207 - 11 で YES)、M に MAX\_\_APPEND と同じ値を代入し (S 207 - 20)、読込終了処理 (S 207 - 21, S 207 - 22) に移る。当該処理 (S 207 - 21, S 207 - 22) については後述する。

10

## 【0122】

一方、M の値が MAX\_\_APPEND 以下であった場合 (S 207 - 11 で NO)、データ変換部 203 は、追記領域読込バッファ 2067 の M 番地からデータ (識別符号) を取得し、変数 CODE へ代入する (S 207 - 12)。ここで、取得する CODE のデータ長は識別符号として使用する符号の種類に依存する。例えば、識別符号として固定長の符号を用いる場合は、常にその符号長のデータを取得しても良いし、可変長の符号を用いる場合には、符号が決定されるまでデータを読み込むこととしても良い。

## 【0123】

続いて、データ変換部 203 は、変数 CODE の値が、追記領域の終端を示す識別符号 c0 と一致するか否かを確認する (S 207 - 13)。CODE の値が c0 と一致する場合 (S 207 - 13 で YES)、読込終了処理 (S 207 - 20, S 207 - 21) に移る。当該処理 (S 207 - 20, S 207 - 21) については後述する。

20

## 【0124】

一方、CODE の値が c0 以外であった場合 (S 207 - 13 で NO) には、まず、変数 M の値を CODE の長さ分増加させる (S 207 - 14)。続いて、データ変換部 203 が、符号管理テーブル 208 内で「識別符号」フィールド 208 - 2 が変数 CODE と同じ値を持つ行を選択し、当該行の「データ名」フィールド 208 - 1 の値を変数 NAME へ代入する (S 207 - 15)。

## 【0125】

次に、データ変換部 203 は、対応管理テーブル 207 内で「データ名」フィールド 207 - 1 が変数 NAME と同じ値を持つ行を選択し、当該行の「オフセット」フィールド 207 - 2 と、「データ長」フィールド 207 - 3 の値をそれぞれ変数 OFFSET と LEN へ代入する (S 207 - 16)。

30

## 【0126】

続いて、データ変換部 203 は、追記領域読込バッファ 2067 内の、M 番地から長さ LEN の範囲でデータを取得し、変数 DATA へ代入する (S 207 - 17)。

## 【0127】

次に、データ変換部 203 は、業務データ読込バッファ 2063 の NAME 行へ、変数 DATA の値を格納する (S 207 - 18)。このとき、当該業務データ読込バッファ 2063 の当該位置 (NAME 行) へ既にデータが書き込まれている場合には、既存データを上書きすることで新データを格納することとする。

40

## 【0128】

続いて、データ変換部 203 は、変数 M の値を、M に LEN の値を加えた値へ更新し (S 207 - 19)、M の値が MAX\_\_APPEND を超えているか否かを判定する処理 (S 207 - 11) へ戻る。

## 【0129】

上述した通り、処理 S 207 - 11 または S 207 - 13 でそれぞれ条件が満たされた場合 (S 207 - 11 または S 207 - 13 で YES)、読込終了処理 (S 207 - 21, S 207 - 22) に移る。

## 【0130】

50

まず、データ変換部 203 は、変数 M の値を追記領域末尾ポインタ 205 へ格納する (S 207 - 21)。追記領域末尾ポインタ 205 への格納は、タグを一意に識別するための識別子 (タグ ID) との組として実行する。

【0131】

次に、ミドルウェア装置 2 - 2 内のアプリケーションコマンド処理部 202 が、読込要求アプリケーションコマンドに対する応答として、業務データ読込バッファ 2063 内のデータをパラメータとして含む読込要求アプリケーション成功応答を生成し、アプリケーション通信部 201 を介して、業務アプリケーション装置 1 - 2 へ送信する (S 207 - 22)。

【0132】

以上が、ミドルウェア装置 2 - 2 による、読込後処理 S 207 の詳細である。

【0133】

追記領域読込バッファ 2067 に格納されるデータは、データ名 (NAME) が同じ場合、基本領域読込バッファ 2065 に格納されるデータよりも新しいデータとなることを利用して、上に記述した読込後処理 S 207 のフローの代わりに以下の順序で処理を実行することで、読込処理の高速化を実現することが可能である。まず、S 207 - 1, S 207 - 2 の処理を実行する。続いて、追記領域読込バッファ 2067 から読み込んだデータを業務データ読込バッファ 2063 へ格納する処理 (S 207 - 10 から S 207 - 21 までの一連の処理) を実行する。ここで、業務データ読込バッファ 2063 へ格納されたデータ名 (NAME) を「更新済データ種別」として記憶する。次に、基本領域読込バッファ 2065 から読み込んだデータを業務データ読込バッファ 2063 へ格納する処理 (S 207 - 3 から S 207 - 9 までの一連の処理) を実行するが、ここで、処理 S 207 - 5 と処理 S 207 - 6 の間に、データ名 (NAME) が「更新済データ種別」に含まれるデータに関しては、基本領域読込バッファ 2065 からの読込処理 (S 207 - 6, S 207 - 7) をスキップする。最後に、処理 S 207 - 22 を実行し、読込後処理 S 207 を終了する。図 9 に戻り、上述の読込後処理 S 207 の結果、ミドルウェア装置 2 - 2 が業務アプリケーション装置 1 - 2 へ読込要求アプリケーション成功応答を送信する。当該成功応答には、IC タグ 4 から読み取ったデータが、業務データ名とその値の組として含まれている (S 208)。

【0134】

業務アプリケーション装置 1 - 2 は、ミドルウェア通信部 102 がミドルウェア装置 2 - 2 からデータを受信すると、業務処理部 101 が当該データを用いて業務処理を実行し、データの一部書き換え等の処理を行う (S 209)。

【0135】

続いて、業務アプリケーション装置 1 - 2 は、業務処理部 101 がミドルウェア通信部 102 を介して、書き換えられたデータの更新要求メッセージを、ミドルウェア装置 2 - 2 へ送信する (S 210)。当該メッセージには、書き換えられたデータに限らず、業務データ全体が含まれていても良い。

【0136】

ミドルウェア装置 2 - 2 は、装置内のアプリケーション通信部 201 が、業務アプリケーション装置 1 - 2 から送信された更新要求アプリケーションコマンドを受信すると、受け取ったパラメータ内の業務データを、IC タグへ書き込むための処理を実行する (書込前処理, S 211)。

【0137】

以下、当該処理 S 211 の詳細について、図 12 を用いて説明する。

【0138】

図 12 は、ミドルウェア装置 2 による書込処理の詳細を表したフローチャートである。

【0139】

ミドルウェア装置 2 - 2 内では、アプリケーション通信部 201 が、業務アプリケーション装置 1 - 2 から送信された更新要求アプリケーションコマンドを受信すると (S 21

10

20

30

40

50

1 - 1 ) , アプリケーションコマンド処理部 2 0 2 が , 当該コマンドをデコードし , 当該コマンドに含まれるパラメータ ( 業務データ ) を取得しデータ変換部 2 0 3 へ受け渡す ( S 2 1 1 - 2 ) 。

【 0 1 4 0 】

データ変換部 2 0 3 は , まず , 業務データ書込バッファ 2 0 6 2 と , 業務データ差分バッファ 2 0 6 1 と , 追記領域書込バッファ 2 0 6 6 と , を空にする ( S 2 1 1 - 3 ) 。

【 0 1 4 1 】

次に , データ変換部 2 0 3 は , 取得した業務データを , 業務データ書込バッファ 2 0 6 2 へ格納する ( S 2 1 1 - 4 ) 。

【 0 1 4 2 】

続いて , 差分計算部 2 0 4 が , 業務データ書込バッファに格納された値と , 業務データ読込バッファに格納された値と , を比較し , 値が異っているデータだけを抽出し , 業務データ差分バッファへ格納する ( S 2 1 1 - 5 ) 。

【 0 1 4 3 】

次に , データ変換部 2 0 3 は , C P U 9 0 1 内のレジスタまたはメモリ 9 0 2 または外部記憶装置 9 0 3 上に , 4 つの変数 M , N A M E , C O D E , D A T A 用の領域を確保し , それぞれ値 0 で初期化する ( S 2 1 1 - 6 ) 。

【 0 1 4 4 】

続いて , データ変換部 2 0 3 は , 業務データ差分バッファにデータが存在するか否かを判定する ( S 2 1 1 - 7 ) 。

【 0 1 4 5 】

当該判定の結果 , 当該バッファにデータが存在する場合 ( S 2 1 1 - 7 で Y E S ) , データ変換部 2 0 3 は , 業務データ差分バッファ 2 0 6 1 から 「データ名」フィールド 2 0 6 1 - 1 と 「データ」フィールド 2 0 6 1 - 2 の値を取得し , それぞれの値で変数 N A M E と変数 D A T A の値を上書きする。また値の取得後には , 取得した値を含む行のデータを , 業務データ差分バッファ 2 0 6 1 から削除し , 同じ値が二度以上取得されないようにする ( S 2 1 1 - 8 ) 。

【 0 1 4 6 】

次に , データ変換部 2 0 3 は , 符号管理テーブル 2 0 8 内で 「データ名」フィールド 2 0 8 - 1 が変数 N A M E と同じ値を持つ行を選択し , 当該行の 「識別符号」フィールド 2 0 8 - 2 の値を変数 C O D E へ代入する ( S 2 1 1 - 9 ) 。

【 0 1 4 7 】

続いて , データ変換部 2 0 3 は , 変数 C O D E と変数 D A T A に含まれる値を , C O D E , D A T A の順で結合し , 結合後のデータを追記領域書込バッファ 2 0 6 6 の末尾へ追加する ( S 2 1 1 - 1 0 ) 。

追加後 , データ変換部 2 0 3 は , 業務データ差分バッファ 2 0 6 1 にまだデータが残っているかどうかを確認する処理 ( S 2 1 1 - 7 ) に戻る。

【 0 1 4 8 】

処理 S 2 1 1 - 7 にて , 業務データ差分バッファ 2 0 6 1 にデータが残っていない , すなわち , 書き換える必要のある全ての業務データに対して , 追記領域書込バッファ 2 0 6 6 への追記が完了すると ( S 2 1 1 - 7 で N O ) , データ変換部 2 0 3 は , 追記領域書込バッファ 2 0 6 6 の末尾へ識別符号 c 0 を書き込む ( S 2 1 1 - 1 1 ) 。

【 0 1 4 9 】

次に , データ変換部 2 0 3 は , 追記領域末尾ポインタ 2 0 5 から , 書込対象の電子タグ 4 のタグ I D に一致する行を検索し , 追記領域 4 0 3 2 の末尾アドレスを取得し , 変数 M へ格納する ( S 2 1 1 - 1 2 ) 。

【 0 1 5 0 】

続いて , データ変換部 2 0 3 は , 追記領域書込バッファ 2 0 6 6 内のデータを , I C タグ 4 の追記領域 4 0 3 2 内 , 変数 M で指定される番地へ書き込むよう , リーダライタ装置 3 - 2 に対してデータ書込要求メッセージを送信し ( S 2 1 1 - 1 3 ) , リーダライタ装置 3 - 2 からの返信を待ち受ける状態へ遷移する ( S 2 1 1 - 1 4 ) 。

当該データ書込要

10

20

30

40

50

求メッセージはパラメータを含み、当該パラメータによって書込位置と書き込むデータの内容を指示するものとする。ここでは書込位置として追記領域 4 0 3 2 の M 番地を指定し、書き込むデータの内容として追記領域書込バッファ 2 0 6 6 内のデータを指定する。

【 0 1 5 1 】

以上が、ミドルウェア装置 2 - 2 による、書込前処理 S 2 1 1 の詳細である。

【 0 1 5 2 】

図 9 に戻り、上述の書込前処理 S 2 1 1 の結果、ミドルウェア装置 2 - 2 がリーダライタ装置 3 - 2 へデータ書込要求メッセージを送信すると ( S 2 1 2 )、リーダライタ装置 3 - 2 は当該書込要求に従い、IC タグ 4 内の追記領域 4 0 3 2 における M 番地へ、指定されたデータの書込を実行する ( S 2 1 3 )。

10

【 0 1 5 3 】

IC タグ 4 内への書込が成功すると ( S 2 1 4 )、リーダライタ装置 3 - 2 はミドルウェア装置 2 - 2 に対して追記領域 4 0 3 2 への書込に成功した旨を通知する ( S 2 1 5 )。

【 0 1 5 4 】

ミドルウェア装置 2 - 2 は、リーダライタ装置 3 - 2 から、追記領域 4 0 3 2 への書込に成功した旨を通知されると、書込の成功を業務アプリケーション装置 1 - 2 へ通知するための処理に移る ( 書込後処理、S 2 1 6 )。

【 0 1 5 5 】

以下、当該処理 S 2 1 6 の詳細について、再び図 1 2 を用いて説明する。

20

【 0 1 5 6 】

ミドルウェア装置 2 - 2 内では、リーダライタ通信部 2 1 0 がリーダライタ装置 3 - 2 からメッセージを受信すると、IC タグコマンド処理部 2 0 9 が、当該メッセージをデコードし、当該メッセージが、追記領域 4 0 3 2 への書込に成功したことを意味するメッセージであると解釈する ( S 2 1 1 - 1 4 で Y E S )。

【 0 1 5 7 】

すると、ミドルウェア装置 2 - 2 内のアプリケーションコマンド処理部 2 0 2 が、更新要求アプリケーションコマンドの実行が成功したことを示す更新要求アプリケーション成功応答を生成し、アプリケーション通信部 2 0 1 を介して、業務アプリケーション装置 1 - 2 へ送信する ( S 2 1 6 - 1 )。

30

【 0 1 5 8 】

一方、S 2 1 1 - 1 4 において、ミドルウェア装置 2 - 2 が、リーダライタ装置 3 - 2 から何らかのエラーメッセージを受信した場合や、一定時間、一定回数のリトライを経てもリーダライタ装置 3 - 2 から応答が無かった場合は ( S 2 1 1 - 1 4 で N O )、ミドルウェア装置 2 - 2 のアプリケーションコマンド処理部 2 0 2 が、更新要求アプリケーションコマンドの実行に失敗したことを示す更新要求アプリケーション失敗応答を生成し、アプリケーション通信部 2 0 1 を介して、業務アプリケーション装置 1 - 2 へ送信する ( S 2 1 6 - 2 )。

【 0 1 5 9 】

以上が、ミドルウェア装置 2 - 1 による、書込後処理 S 2 1 6 の詳細である。

40

【 0 1 6 0 】

図 9 に戻り、上述の書込後処理 S 2 1 6 の結果、ミドルウェア装置 2 - 2 が業務アプリケーション装置 1 - 2 へ更新要求アプリケーション成功応答を送信すると ( S 2 1 7 )、IC タグ 4 に対するデータ更新処理は正常に完了する。

【 0 1 6 1 】

なお、ここでは説明のために、IC タグ 4 に対して初期化処理を行った業務アプリケーション装置と、初期化後の IC タグ 4 に対して読込処理と書込処理を行う業務アプリケーション装置を異なるものとして記載したが、両者は必ずしも異なるものでなくとも良い。

【 0 1 6 2 】

また、追記領域 4 0 3 2 の空き容量が小さくなった場合には、以下の手順により、追記

50

領域 4 0 3 2 のクリーニングを行うことで、空き容量を最大化することが可能である。まず、業務アプリケーション装置 1 は、ミドルウェア装置 2 に対して読込要求メッセージを送信し、IC タグ 4 から最新の業務データを取得する。続いて、当該取得したデータをパラメータとする初期化要求メッセージを、ミドルウェア装置 2 に対して送信することで、IC タグ 4 内の基本領域 4 0 3 1 のデータが最新のデータに更新され、追記領域 4 0 3 2 には識別符号 c 0 だけが格納されることとなる。

【 0 1 6 3 】

ここで、追記領域 4 0 3 2 のクリーニングを行うまでは、それまでに IC タグ 4 に対して行われた全ての更新内容が、追記領域内のデータという形で残されているため、当該領域内の以前のデータを参照し、最新のデータとして再度書き込みを行うことで、処理のやり直し（アンドゥ操作）を実現することが可能である。

10

【 実施例 2 】

【 0 1 6 4 】

以上の実施形態に基づき、IC タグを乗車券として利用した場合を例に取って、IC タグ（乗車券）へのデータの書込が行われる過程をより具体的に説明する。

【 0 1 6 5 】

図 1 3 は、本実施例のシステム構成の概略図である。

【 0 1 6 6 】

図示するように、本実施例では、業務アプリケーション装置として、発券装置 1 - 1 と、入場管理装置 1 - 2 と、出場管理装置 1 - 3 と、精算装置 1 - 4 と、の 4 つを備える。当該装置はそれぞれミドルウェア装置 2 - 1 ~ 2 - 4 と相互に通信を行い、各ミドルウェア装置はそれぞれリーダーライタ装置 3 - 1 ~ 3 - 4 と相互に通信を行う。IC タグ 4 は乗車券として利用される。

20

【 0 1 6 7 】

また、図 1 4 は、本実施例において、各ミドルウェア装置 2 - 1 ~ 2 - 4 に格納される対応管理テーブル 2 0 7 と符号管理テーブル 2 0 8 の一例を示した図である。例えば対応管理テーブル 2 0 7 においては、データ名として「乗車駅」を持つデータは、IC タグ 4 上の基本領域 4 0 3 1 ではオフセット（k + 2, 1 0）で指定される箇所に書き込まれ、その長さは 8 であることが示されている。また、符号管理テーブル 2 0 8 においては、データ名として「乗車駅」を持つデータは、IC タグ 4 上の追記領域 4 0 3 2 へ書き込まれる際には、識別符号 c 9 と結合された状態で書き込まれることが示されている。

30

【 0 1 6 8 】

乗車客が駅構内に設置された発券装置 1 - 1（例えば券売機）に対して運賃の投入や行き先の範囲選択などを指示すると、発券装置 1 - 1 はミドルウェア装置 2 - 1 に対し、図 7 および図 8 で示したような初期化処理の実行要求を送信する。図 7 に示すシーケンスおよび図 8 に示すフローが滞りなく完了すると、IC タグ 4 のデータ記憶部 4 0 3 には図 1 5 のようにデータが書き込まれる。

【 0 1 6 9 】

図 1 5 は、発券装置 1 - 1 による発券処理が完了した後の、IC タグ 4（乗車券）内のデータ記憶部 4 0 3 の一例を示した図である。発券駅、種別、改札通過日時、有効期間、有効区間、入出場情報（図 1 5 では「入出」と記載）、発券日時、乗車駅、といったデータが基本領域 4 0 3 1 に書き込まれる。追記領域 4 0 3 2 には、先頭に識別符号 c 0 のみが書き込まれる。

40

【 0 1 7 0 】

続いて乗車客が、発券された IC タグ 4 を入場管理装置 1 - 2（例えば自動改札機）へ挿入すると、当該入場管理装置 1 - 2 はミドルウェア装置 2 - 2 に対し、図 9 ~ 図 1 2 で示したような読込処理および更新処理の実行要求を送信する。

【 0 1 7 1 】

まず、図 9 の S 2 0 1 ~ S 2 0 7 までの処理が滞りなく完了すると、IC タグ 4 のデータ記憶部 4 0 3 から読み出されたデータがミドルウェア装置 2 - 2 内の業務データ読込バ

50

ッファ2063へ格納される。当該データが格納された直後の業務データ読込バッファ2063の一例を図16の左上部へ示す。図示するように、各業務データ名（発券駅、改札通過日時など）に対応した値が読み取られ、バッファに格納される。

#### 【0172】

入場管理装置1-2は、ミドルウェア装置2-2から、業務データ読込バッファ2063内のデータを受信すると、図9の業務処理S209に相当する処理として、例えば、発券駅や発券時刻、有効区間などを確認し、入場の可否を判定する。入場可能と判定した場合には、業務データを書き換えた後、図9のS210で示すような更新要求メッセージを送信する。このとき、書き換え後のデータは当該更新要求メッセージのパラメータとして渡され、当該パラメータは、図12のS211-1～S211-4に従い、業務データ書込バッファ2062へ格納される。当該データが格納された直後の業務データ書込バッファ2062の一例を図16の右上部へ示す。

10

#### 【0173】

続いて、ミドルウェア装置2-2は、図12のS211-5に従い、業務データ書込バッファ2062と業務データ読込バッファ2063の差分を計算し、業務データ差分バッファ2061へ格納する。当該差分データが格納された直後の、業務データ差分バッファ2061の一例を、図16の下部へ示す。

#### 【0174】

以上に示したように、本実施例では、書き換え前の業務データと、書き換え後の業務データの差分を計算することにより、更新が必要なデータの範囲を特定し、後に実行する、ICタグに対するデータ書込処理の回数を削減し、書込速度の向上が実現されている。

20

#### 【0175】

ミドルウェア装置2-2は、図12のS211-7～S211-10に従い、業務データ差分バッファ2061内の各業務データを追記領域書込バッファ2066へ書き込んだ後、当該バッファの末尾へ識別符号c0を付加し、追記領域の末尾へ追記する。当該追記が完了した直後の、ICタグ4内データ記憶部403の一例を、図17に示す。

#### 【0176】

更新処理においては、基本領域4031内のデータには変更を加えず、追記領域4032に、識別符号と当該識別符号に対応する更新後のデータとを連ねて追記する。図17では、追記領域4032内に改札通過日時、入出場情報、乗車駅情報を追記している。（図17中、\*の記号は更新後のデータであることを示す）

30

以上に示したように、本実施例では、書き換え対象のデータが連続するように追記を行う。そのため、ISO18000-6 Type C仕様のBlockWriteコマンドに対応したICタグのように、メモリ上で連続したデータを1度のコマンドで書き込むことのできるICタグに対して、1度のコマンドで更新処理が済み、書込処理の高速化が実現されている。

#### 【0177】

続いて乗車客が電車へ乗り、所望の駅に到着後、出場するために、入場管理装置1-2を通した後のICタグ4を、出場管理装置1-3（例えば自動改札機）へ挿入すると、当該出場管理装置1-3は、ミドルウェア装置2-3に対し、図9～図12で示したような読込処理および更新処理の実行要求を送信する。

40

#### 【0178】

ここで、出場管理装置1-3による読込処理は、追記領域4032内にデータが格納されている点で、先述した入場管理装置1-2による読込処理の場合と異なる。特に、図11のS207-14～S207-19の処理が、追記領域4032に格納されているデータの分だけ実行される。例えばICタグ4内のデータ記憶部403が、図17に示したような状態を取る場合、図11の処理S207-13で追記領域読込バッファ先頭のデータがc0と比較されるが、当該バッファ先頭には識別符号c2が格納されているため、S207-14以降へ処理を進めることとなる。

#### 【0179】

50

まず、Mの値を識別符号c2の長さだけ増加させた後(S207-14)、図14に記載した符号管理テーブル208を参照し、識別符号としてc2を含む行を探し、当該行の「データ名」フィールドに含まれる「改札通過日時」を取得し、変数NAMEへ代入する(S207-15)。

【0180】

次に、対応管理テーブル207を参照し、データ名として変数NAMEすなわち「改札通過日時」を含む行を探し、当該行の「オフセット」フィールドと「データ長」フィールドの値をそれぞれ変数OFFSETおよび変数LENへ代入する。ここでは図14に記載の対応管理テーブル207に基づき、OFFSETには(k, 10)が、LENには6が、それぞれ代入される(S207-16)。

10

【0181】

続いて、追記領域読込バッファ2067内、M番地(識別符号c2を読み込んだ直後の位置)から、長さLENだけの範囲に存在するデータを読み込み、変数DATAへ代入する。ここでは、ICタグ4内のデータ記憶部403が図17に示したような構造を取るため、改札通過日時の更新後の値が取得される(S207-17)。取得した値は、業務データ読込バッファ2063の「改札通過日時」行へ格納され、基本領域4031から読み込んだ値が追記領域4032から読み込んだ値で上書きされる。

【0182】

以上に示したような処理が、追記領域4032内の残りのデータに対しても行われ、最終的に業務データ読込バッファ2063内には、最新の追記データが古いデータを上書きした状態で、業務アプリケーション(ここでは出場管理装置1-3)へ提供される。

20

【0183】

本実施例では、ICタグ内でのデータ配置方法を識別符号のみで表現し、データ長やデータ名は各ミドルウェア装置に保存することで、ICタグ内のメモリを有効利用することを可能にしている。

【0184】

出場管理装置1-3は、ICタグ4の有効性を確認すると、業務データの改札通過日時を書き換え、ミドルウェア装置2-3へ更新を要求し、更新の完了後、乗車券を回収し、改札の扉を乗車客のために開放する。改札通過日時が書き換えられた直後のICタグ4について、データ記憶部403の一例を図18に示す。図示するように、追記領域4032の末尾には、識別符号c2と、改札通過日時の二度目の更新データ(「改札通過日時\*\*」として表記)と、追記領域の末尾を示す識別符号c0と、が追記されている。

30

【0185】

精算装置1-4による精算時など、ICタグ4とミドルウェア装置2とが通信可能な時間が比較的長く確保されている場合には、追記領域4032の空き容量を拡大するために、以下のような処理を実行しても良い。まず、精算装置1-4は、ミドルウェア装置2-4に対して読込要求メッセージを送信し、ICタグ4から最新の業務データを取得する。続いて、当該取得したデータをパラメータとする初期化要求メッセージを、ミドルウェア装置2-4に対して送信することで、ICタグ4内の基本領域4031のデータが最新のデータに更新され、追記領域4032には識別符号c0だけが格納されることとなる。

40

【0186】

図19は、図18で示したICタグ4に対して、上記の処理を実行した結果、追記領域4032の空き容量が最大化された例を図示したものである。

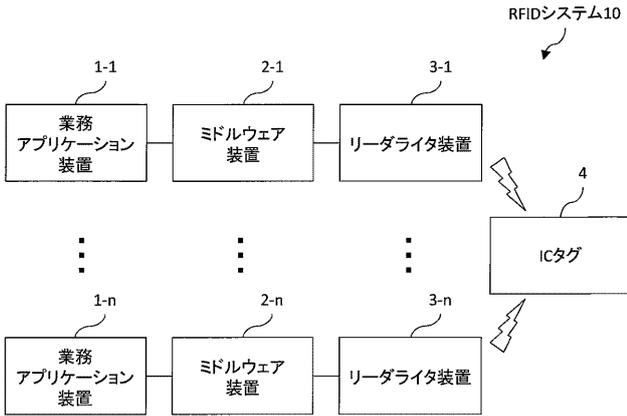
【符号の説明】

【0187】

1-1~1-n:業務アプリケーション装置, 2-1~2-n:ミドルウェア装置, 3-1~3-n:リーダライタ装置, 4:ICタグ, 9:コンピュータ, 10:RFIDシステム

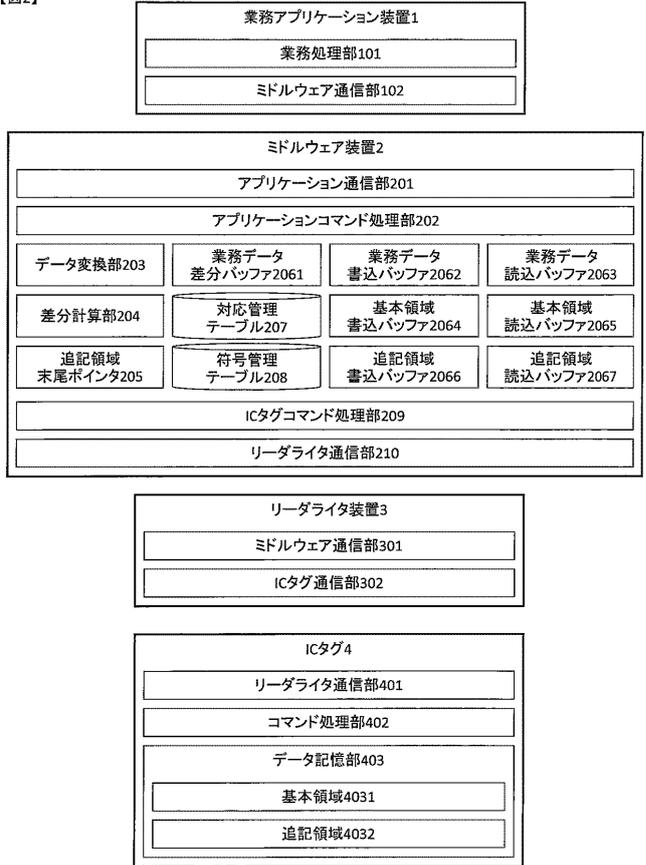
【図1】

【図1】



【図2】

【図2】



【図3】

【図3】

業務データ書込バッファ2062

データ名	データ
d1	2009-01-21
d2	18:28:32
⋮	⋮
d6	渋谷駅
⋮	⋮
d9	未入場

【図5】

【図5】

符号管理テーブル208

データ名	識別符号
NULL	c0
d1	c1
d2	c2
⋮	⋮
d6	c6
⋮	⋮
d9	c9

【図4】

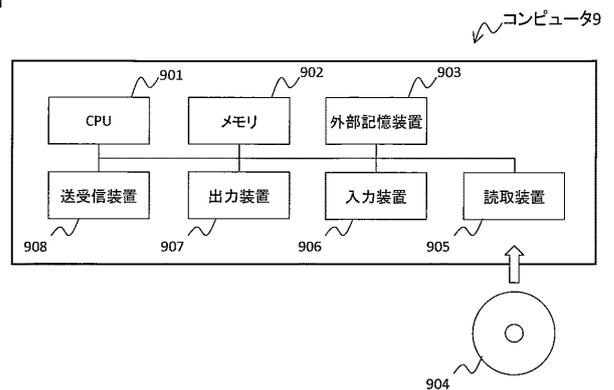
【図4】

対応管理テーブル207

データ名	オフセット	データ長
d1	(k, 0)	8
d2	(k, 10)	6
⋮	⋮	⋮
d6	(k+1, 14)	4
⋮	⋮	⋮
d9	(k+2, 10)	6

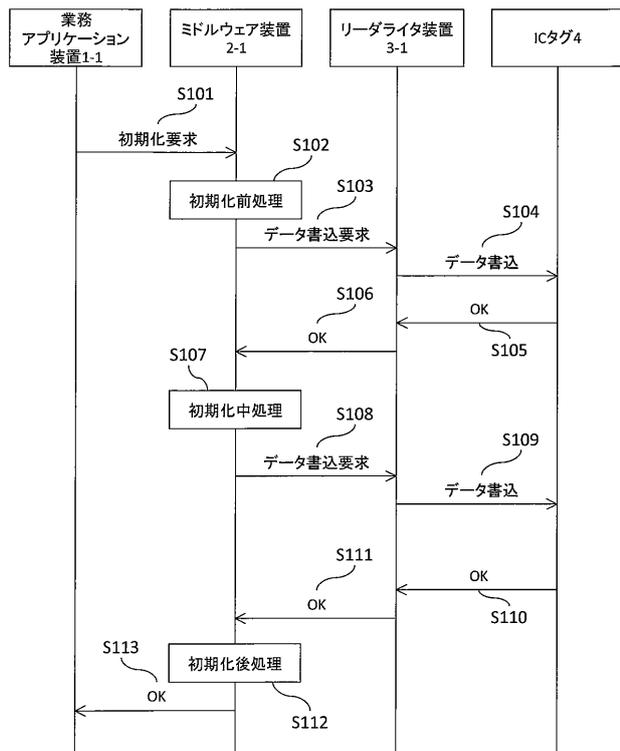
【図6】

【図6】



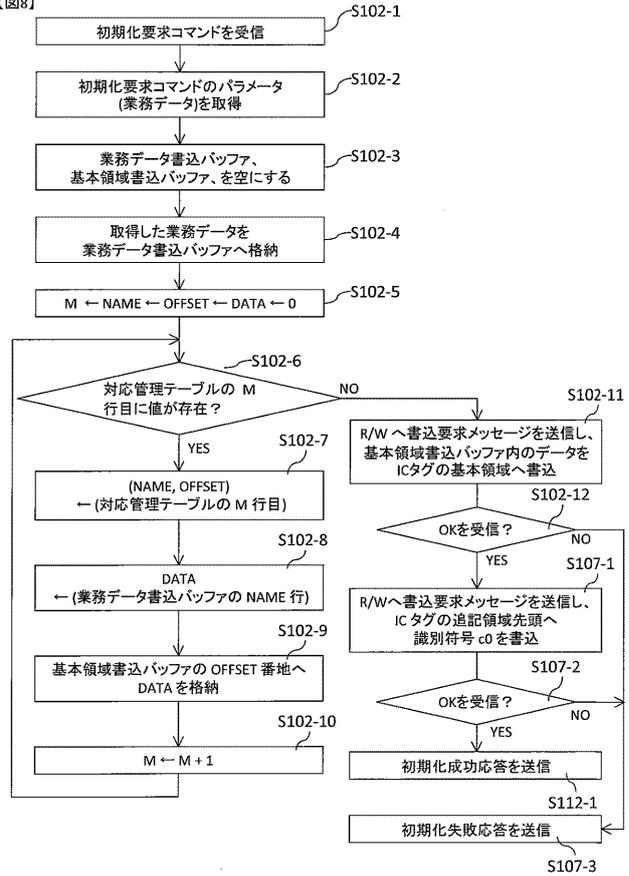
【 図 7 】

【図7】



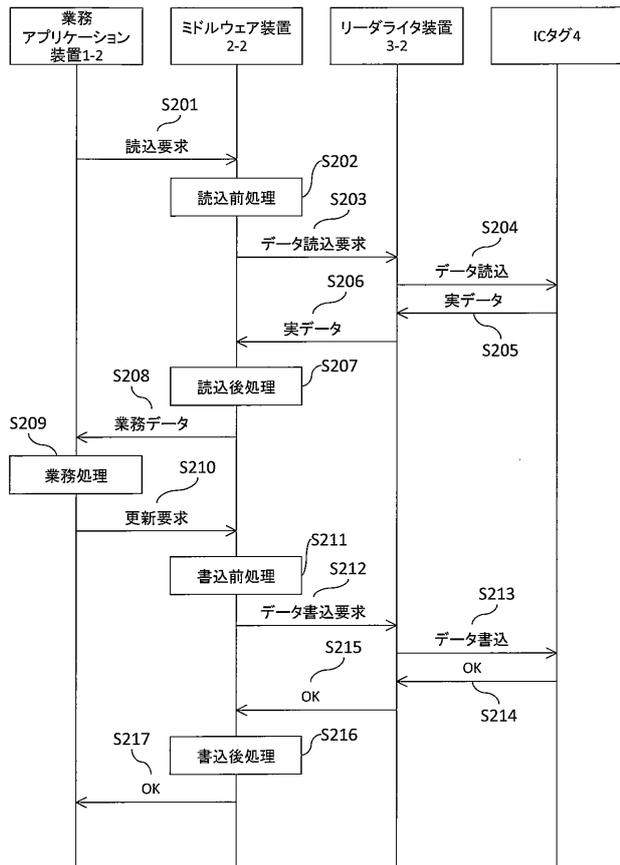
【 図 8 】

【図8】



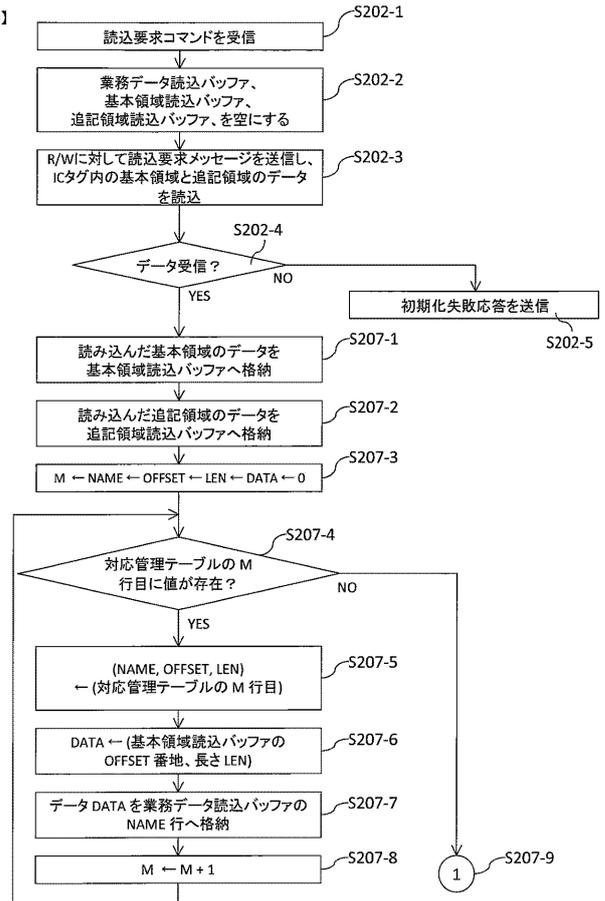
【 図 9 】

【図9】



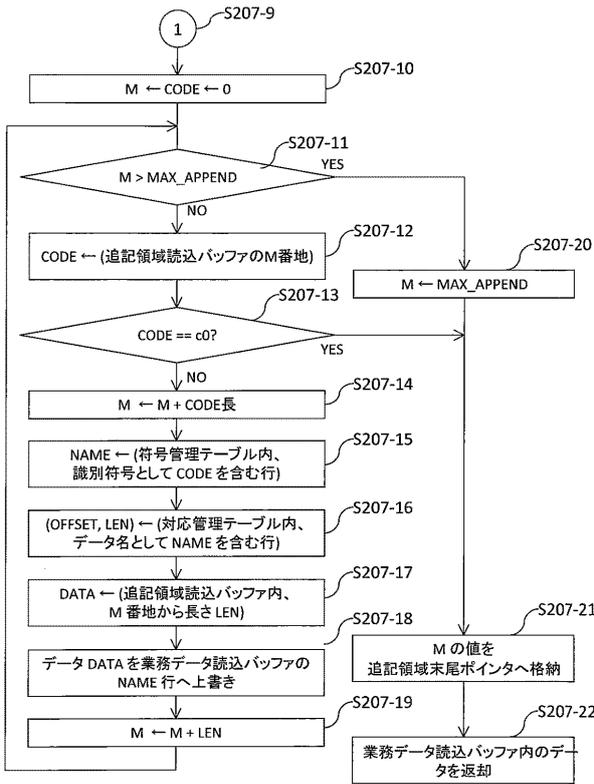
【 図 10 】

【図10】



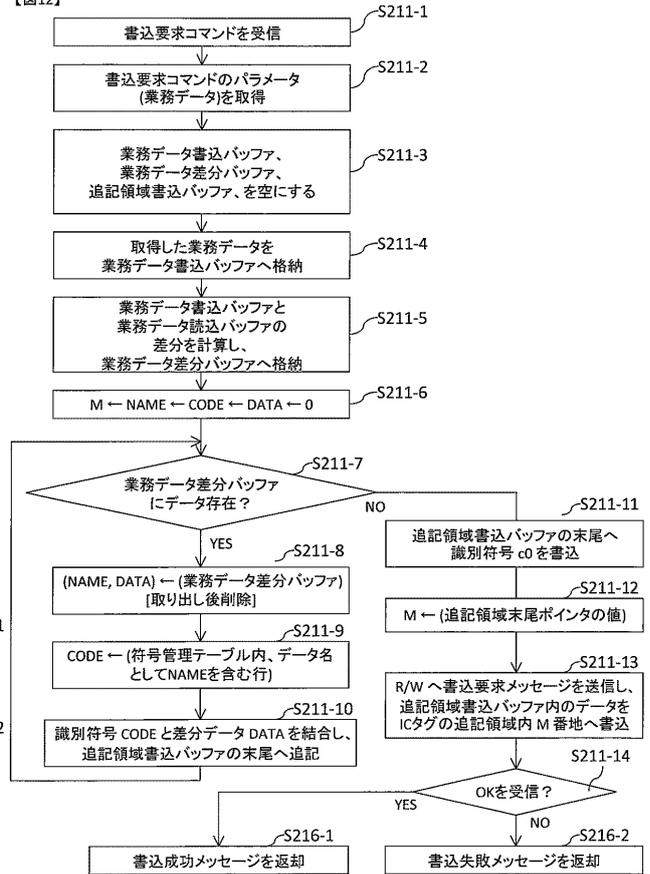
【図11】

【図11】



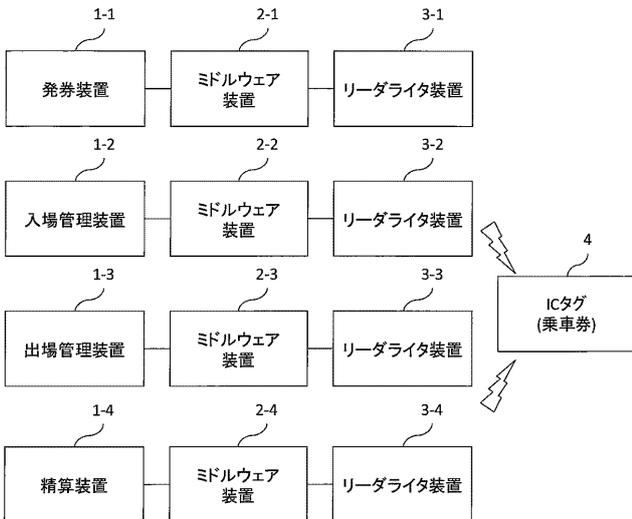
【図12】

【図12】



【図13】

【図13】



【図14】

【図14】

対応管理テーブル207

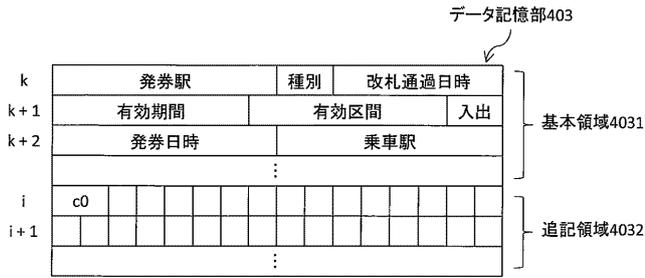
データ名	オフセット	データ長
発券駅	(k, 0)	8
改札通過日時	(k, 10)	6
⋮	⋮	⋮
入出場情報	(k+1, 14)	2
⋮	⋮	⋮
乗車駅	(k+2, 10)	8

符号管理テーブル208

データ名	識別符号
NULL	c0
発券駅	c1
改札通過日時	c2
⋮	⋮
入出場情報	c6
⋮	⋮
乗車駅	c9

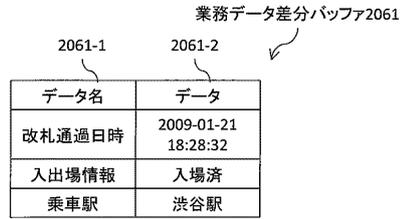
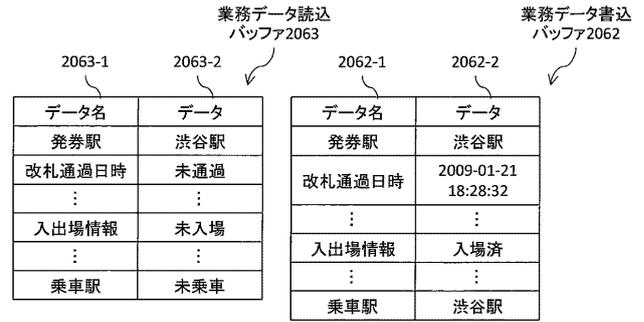
【図15】

【図15】



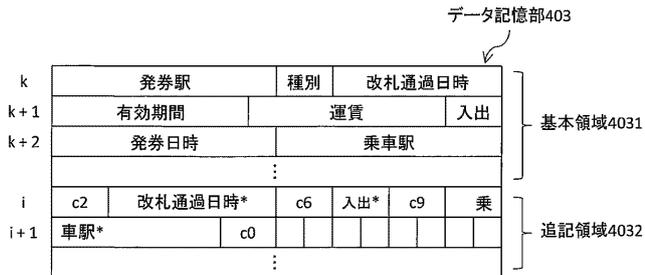
【図16】

【図16】



【図17】

【図17】



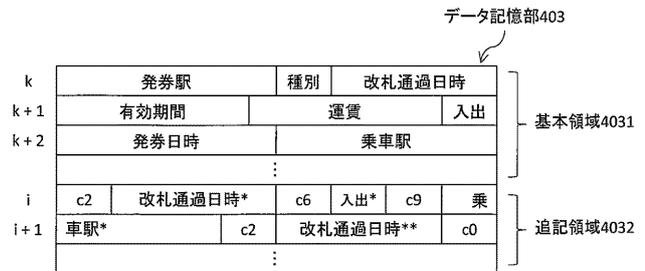
【図19】

【図19】



【図18】

【図18】



---

フロントページの続き

(72)発明者 鍛 忠司

神奈川県川崎市麻生区王禅寺 1 0 9 9 番地 株式会社日立製作所システム開発研究所内

(72)発明者 五十井 恭子

東京都港区港南二丁目 1 5 番 2 号 株式会社日立製作所情報制御システム事業部内

Fターム(参考) 5B058 CA15 KA04 KA08 KA11

【要約の続き】

【選択図】 図 1