

RF タグへの書き込みシーケンスの検討

米澤 祐紀^{†1} 江原 正規^{†2} 星 徹^{†2}
高 汐一 紀^{†3} 東 康弘^{†4} 中村 雅之^{†4}
半田 隆夫^{†4} 徳田 英幸^{†1}

現在 RF タグは、高速処理が要求される場面で利用が検討されている。しかし RF タグへのアクセスは、外部環境の影響により途中でアクセスできなくなってしまう場合があり、従来手法では、データが途中で書き込めなくなった場合に誤ったデータが検知されてしまうという問題を抱えていた。またそれを間違ったデータだと検知することはできない。そこで本論文では、データのエンドから書き込みを行うバックライト手法の提案を行った。この提案手法により書き込んだデータに信頼性を持たせることが可能となった。

An Examination of Writing Sequence to RF tag

YUKI YONEZAWA,^{†1} MASAKI EHARA,^{†2}
TOHRU HOSHI,^{†2} KAZUNORI TAKASHIO,^{†3}
YASUHIRO HIGASHI,^{†4} MASAYUKI NAKAMURA,^{†4}
TAKAO HANDA,^{†4} and HIDEYUKI TOKUDA^{†1}

UHF RFID tags become to be used for traceability in logistic fields or baggage transportation system in airports, etc. However it can not be always access the data of RFID tag. If it can not access to RFID tag, existing method can not identify that it is correct data or not. Therefore, in this paper we propose the method of back write that writes from the end of data. Then, we can provide the reliable data with this method.

1. はじめに

近年、大容量の FRAM (Ferroelectric Random Access Memory) を搭載した RFID が開発され、ユーザが読み書き可能な大きなユーザメモリを活かしたアプリケーションの提案や開発が進んでいる。RFID の貼付対象のモビリティに着目すると、これらは2つに大別することができる。

モビリティを有する対象の例としては航空機向けアプリケーションがあり、これは航空機の多くの部品情報や整備情報をメモリに格納することによって、高品質なメンテナンスを実現しようするものである。一方、モビリティを有しない例としては、測量用基準点への埋め込み (インテリジェント基準点) といった試みがある。このようにモビリティの有無では用途は一見異なるものの、従来の流通での利用法 (工場、倉庫、店舗等) と比して、公共ネットワークにアクセスができないような場合・状況でも、屋外でのデータキャリアとして保守や管理への活用が期待されていることが分かる。このような利用環境では、RFID タグ自体の堅牢性や耐環境性が求められることは勿論であるが、RFID リーダライタがハンディタイプになることから、作業上、タグとリーダライタ間の通信の安定性が担保されづらい状況も想定される。そのため、ID を読み取るだけの場合と違い、重要なデータを扱うことから確実な読み取り/書き込みが要求される。

特に UHF 帯 RF タグでは、1word^{*1} ずつの書き込みしかできない点が問題となる。複数 word を一度に書き込むコマンドは用意されているが、RF タグと RFID リーダライタの両方の対応が必要となる。さらに、UHF 帯 RF タグのエアインタフェースの国際標準である ISO/IEC 18000-6TypeC¹ では、複数 word の書き込みコマンドは、オプション扱いである。そのため、1word 単位での書き込みが必然となる。

そこで我々は、UHF 帯 RF タグへの書き込み処理の高速化を行うために、RF タグと RFID リーダライタ間のセッションマネージャの提案や、書き込みデータのパディングを行うことで高速化する手法の提案を行ってきた²⁾³⁾。

しかし、これらの手法は、RF タグへのアクセスが最初から最後まで行える理想的な環境

^{†1} 慶應義塾大学大学院 政策・メディア研究科

^{†2} 東京工科大学 コンピュータサイエンス学部

^{†3} 慶應義塾大学 環境情報学部

^{†4} 日本電信通話

*1 1word=16bit

を想定している。そのため、上記のように RF タグへのアクセスが途中で切れてしまう場合の処理が考慮されていない。アクセスが途中で遮断されてしまった場合の問題として、データの信頼性が失われるという問題がある。例えば、書き込み側が「4911111000014」を RF タグ内に保存しようとしたときに、その書き込み中に書き込みができなくなった場合、別のシステムのデコーダが「4911110999808」というデータにデコードしてしまう問題が起こる。デコーダは、デコードされたデータを書き込み途中のデータであると認識できず、そのままアプリケーションへ値を返すことになる。

つまり、従来手法で書き込みを行ったデータは、データ自体の信頼性に欠ける。現状で書き込みが遮断されてしまった場合、書き込みを行った RFID リーダライタが利用しているシステムしかそのことを検知することはできない。RF タグは、複数のマーケット・システムが関連する分野での利用が考えられているため、書き込み途中の誤ったデータが他のシステム利用されるため問題になると考えられる。そこで、本論文では書き込み途中で書き込みができなくなった場合に、誤ったデータを認識させない書き込み方式の提案を行う。

本論文の構成は以下のとおりである。第 2 章では書き込み中断の状況について述べ、第 3 章では、バックライト書き込み手法の提案を行う。第 4 章では、提案手法の実装について述べ、第 5 章では、提案手法の有効性の評価を行う。そして、第 6 章で本論文のまとめを述べる。

2. データの書き込みについて

RF タグ内部にデータを保存するための国際標準として、ISO/IEC 15962⁴⁾ が存在する。ISO/IEC 15962 は、Linux などのオペレーションシステムが利用するファイルシステムと同じように、データを保存する論理フォーマットが定められている。Linux では、ファイルの実態をデータブロック領域とそれを指し示すポインタである inode に分割することで管理している。Linux では、inode をハードディスクに書き込みを行い、そのあとに実データをデータブロック領域に書き込みを行う。これは、ファイルシステムにパーミッションという概念が存在するため、ファイルの移動中でもパーミッションを紐付けておく必要があるからだと考えられる。一方、RFID のデータフォーマットでは、パーミッションという概念が存在せず、ユーザは自分の所属するアプリケーションに対応する RF タグであれば全てのデータに対し、読み込み、書き込みを行うことができる。そのため、Linux のようなオペレーティングシステムとは別の書き込み手順が必要になると考えられる。

書き込みを行ったデータの信頼性を向上させる技術として、第三者の RFID リーダライ

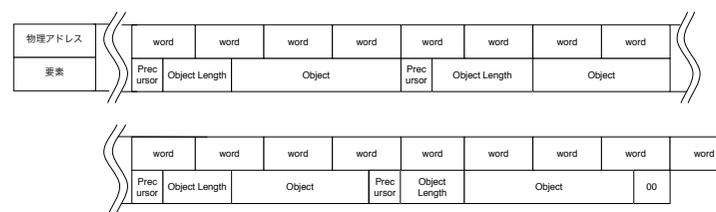


図 1 データフォーマット
 Fig.1 Data Format

タを検知のために、RFID ReaderID、RF タグの Cache データと Cache Check Byte を利用した手法が提案されている⁵⁾。この手法により、書き込み途中であるということが検知できるが、独自拡張機能となるため、他のシステムが介入してきた際に利用ができなくなる。また、我々は電子署名を利用して、データの改ざんを検知する研究も行っている⁶⁾。この提案手法を用いることにより、データの改ざんを検知することができる。しかし、書き込み途中のデータの処理が不十分であるため、書き込み途中の電子署名が RF タグの中に残ってしまい、電子署名に対する信頼性が失われるという問題が存在する。

2.1 エンコーダ・デコーダについて

RF タグへデータを格納する際のエンコードについて述べる。エンコーダは、保存するデータを Object、Object Length、Precursor の大きく 3 つの要素に分けてエンコードを行う。図 1 に ISO/IEC 15962 で用いられる Non-Directory 方式のエンコード済みのデータの並び方を示す。Object は、保存するデータを示す。保存するデータは、エンコーダによって圧縮される。Object Length は、その Object のデータ長を示す。Precursor は、圧縮した Object の圧縮方式と Object を管理する Object Identifier を一緒に格納する。Precursor、Object Length、Object を一組として、これらを Data-Set と呼び、Data-Set の最後にデータエンドを示す「00」を追加する。

次に、RF タグ内に格納されたデータを解析・解凍するデコーダについて述べる。デコーダは、RF タグ内のデータを先頭から解析を行う。Precursor から圧縮方式と Object Identifier を取得し、次に Object Length 分の Object を取得し、データのデコードを行う。そして、Precursor の位置に「00」が現れるとデコード終了になる。

2.2 書き込みのエラータイプ

従来手法では、書き込みを行う Data-Set と一緒に渡される書き込みスタートアドレスに

先頭から書き込みを行うのが一般的である。その場合の、RF タグへの書き込みには次の3つのデータデコードエラーと誤認識が発生する。

- (1) Precursor の書き込みができた場合
- (2) Precursor, Length の書き込みができた場合
- (3) Precursor, Length, Object の途中まで書き込みができた場合

上記に挙げた3つの書き込みエラーが発生した場合、書き込みを行ったシステムは、その書き込んだデータがまだ途中であることが検知できる。そして、別のシステムがそのデータを読み取った場合、(1) は Length がないため、エラーであることが認識できる。しかし、(2), (3) はエラーであることが認識できない。そのため、書き込み途中のデータを正しいデータだとして扱ってしまう。

この問題を解決する手法として、デコーダの性能を上げることにより、上記の問題を解決することが可能である。しかし、エンコーダ・デコーダは最低限の機能要件だけが標準化で決められているため、開発元によりデコーダの性能が異なることが考えられる。そのため、エンコーダ・デコーダの性能によらない解決手法が必要であると考えられる。

3. 提案手法

本研究では、データの書き込みの際に、Precursor を最後に書き込む手法を提案する。本提案手法は、デコーダが先頭から逐次論理データを解析する性質を利用することで、Object, Object Length が書き終わるまで、データを成立させないことができる。そのため、書き込みエラーが発生した際、デコーダはその書き込み途中のデータを検知せずにデコードすることが可能となる。書き込みに失敗したデータは、次に書き込みを行う際に、上書きされる。

Precursor を最後にすることによって、書き込み途中のデータを認識させることはできなくなる。しかし、第1章で述べたように UHF 帯 RF タグは、1word 単位でしか書き込みしかできないため、図2に示すように、どの要素から書き込み、その要素の先頭か後ろのどちらから書き込むかを定める必要がある。

図3に ISO/IEC 15961⁷⁾ に規定されている RFID へのデータフォーマット書き込みの実装概要図を示す。図3は、次の3つの層から構成されている。

- アプリケーション層
 アプリケーション層とは、RF タグを利用するアプリケーションが実装される部分である。アプリケーション層には、アプリケーションインタフェースである ISO/IEC 15961

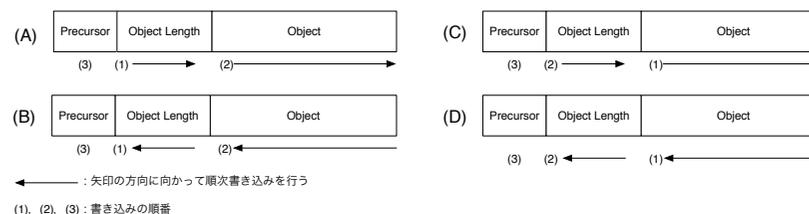


図2 各要素の書き込み順番
 Fig.2 Write Sequence

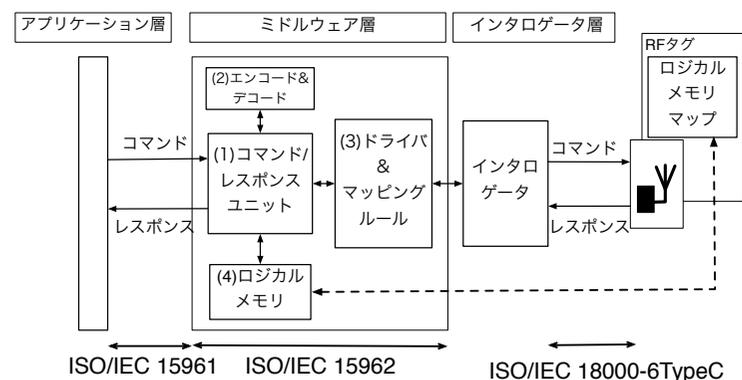


図3 実装概要図
 Fig.3 Implement abstract

を利用する。

- ミドルウェア層
 ミドルウェア層とは、アプリケーション層から送られてきたアプリケーションコマンドに応じて、データのエンコードやデコードを行い、インタロゲータを操作する部分である。この部分は、データフォーマット規定である ISO/IEC 15962 を利用する。
- インタロゲータ層
 インタロゲータ層とは、ミドルウェア層のドライバに従い、RF タグとのやり取りを行う部分である。本論文では、数メートルの範囲で、RF タグと通信が可能のため、物流での利用が検討されている UHF 帯の電波を用いる。UHF 帯に関するコマンドは、

ISO/IEC 18000-6TypeC に規定されている。

そして、ミドルウェア層は、次の4つで構成されている。

- (1) コマンド・レスポンスユニット
受け取ったアプリケーションコマンドに応じて(2)、(3)、(4)を利用してRFタグ内に格納するData-Setを作成し、(3)を利用して、読み込み、書き込みを行う。そして、その結果をレスポンスとしてアプリケーションに返す操作を行う。
- (2) エンコード&デコード
RFタグに格納する際には、圧縮エンコードを行い、取り出す際には、デコードを行う。
- (3) ドライバ&マッピングルール
ドライバはインタロゲータが用意している独自コマンドなどのラッピングを行い、読み書き操作の統一を行う。マッピングルールは、利用するRFタグに応じて格納する場所が異なる場合のそれらをラッピングして利用できるようにした物である。
- (4) ロジカルメモリ

ロジカルメモリは、格納される各Data-Setを連結した形で表されるデータである。

上記に示すように、Data-Setはコマンド・レスポンスユニットで作られる。そして、ドライバ・マッピングルール(以下、ドライバ)がその書き込みを行う。本提案手法は、Data-Setの書き方の手順が重要となってくるため、書き込み制御を行っているドライバに実装することを提案する。そして、図2の(A)、(B)、(C)は各要素を認識するための機能が必要となるため、ドライバの役割を大きく超えた役割が必要となる。しかし(D)では、後ろから順次書き込みを行うことができるので、既存のドライバの役割を変えないで存在することが可能である。そこで、本提案手法では、後ろからの書き込みを行うことを提案する。

本提案手法を用いることにより、第2章であげた問題点の3つのエラーは発生しなくなる。しかし次の2つのエラー状況が発生する。

- (1) Objectのみ書き込みができた場合
- (2) Object, Length of Objectの書き込みができた場合

上記の問題に対して、デコーダは、Precursorが「00」であるため、まだデータが書き込まれていないと判断することが可能となる。そのため、第2章で挙げた問題意識が解決されたこととなると考えられる。

そして、一度に複数のData-Setの書き込みを行う場合も存在する。物流の分野で、吉田

らは、GLN^{*1}とタイムスタンプを一組にして格納を行っている⁸⁾。また、データの改ざんを防止するために、電子署名をデータと一緒に書き込むことも考えられる⁹⁾。そのような場合は、次の2つの方法が考えられる。

- (1) 先頭のData-Setから書き込みを行う
- (2) 後ろのData-Setから書き込みを行う

(1)の順番にすることにより、データの追加を優先して書き込みを行うことが可能となる。そして(2)では、書き込みが終了するまで、両方のデータが成立しないという特徴がある。複数のデータを追加する際には、複数のデータが初めて揃ったときに意味があることが多いため、後ろのObject, Object Length, Precursorの順番に書き込みを行い、そして、先頭のObject, Object Length, Precursorと書き込みを行う。これを行うことによって、組として必要なデータをすべて書き込みを行うことが可能とあるため、(2)の手順を利用することにした。

4. 実 装

本提案手法は、図3のドライバ機能の一つとして実装を行う。ドライバには、コマンド・レスポンスユニットとエンコーダによって構築されたData-Setが渡される。そして、ドライバは渡されたデータをRFタグへ書き込みを行う。

ドライバのクラス図を図4にあげる。ドライバの構成は次の3つから構成される。(1)ドライバインタフェースは、コマンドレスポンスユニットから値を渡すためのインタフェースを定義する。(2)各電波用ドライバは、(1)で定義した各インタフェースに対してリードコマンド、ライトコマンドのシーケンスの定義を行う。各電波ごとにクラスを用意することで、メモリバンクなどの違いを吸収することが可能となる。(3)独自コマンド実装クラスは、各リーダライタごとに保持しているライトコマンドの定義を行う。本提案手法は、(2)の各電波用ドライバに実装を行った。

5. 評 価

評価に利用したRFIDリーダライタ、RFタグを表1に示す。評価実験は、途中で書き込みをPrecursor, Object Lengthをそれぞれ書き込んだ直後にストップさせる。Objectは、書き込み途中でストップさせる。そして、従来手法と提案手法でそれぞれ書き込みを行った

*1 GLN:Global Location Number

(1)ドライバインタフェース

(2)各電波用ドライバ

(3)独自コマンド実装

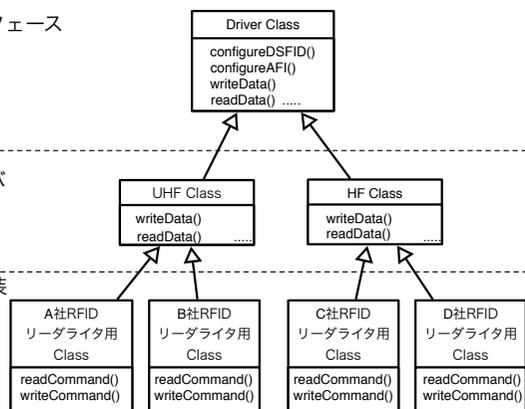


図 4 ドライバのクラス構成図
Fig. 4 Structure of Driver Class

実際のデコーダのデータ検出で評価を行う。

評価実験に用いるデータは、吉田らが利用していた GLN とタイムスタンプを一組にしたものを利用する⁸⁾。GLN とタイムスタンプの Object Identifier と Object を表 2 に示す。GLN とタイムスタンプは、Data-Set で表すと 24Byte になる。はじめに、GLN だけの書き込みの評価を行い、次に、GLN とタイムスタンプを一組にした評価を行う。

表 1 実装に利用した機器
Table 1 Used device

RF タグリーダライタ	UHF 帯 RF タグ対応 RW モジュール 評価キット ⁹⁾
RF タグ	セキュア RF タグプロトコル対応 μ-Chip Hibiki (ユーザメモリは最大 1532bit)

表 2 評価に利用したデータ
Table 2 Data used for evaluation

項目	Object-Identifier	Object
GLN	1 0 15961 9 412 1	4911111000014
タイムスタンプ	1 0 15961 9 7003 1	0901011210

表 3 書き込みエラーに対するデコーダの認識 (GLN)
Table 3 Behavior of decoder to Writing Error(GLN)

エラー場所	Precursor	Object Length	Object
従来手法	認識されない	誤ったデータの検出	誤ったデータの検出
提案手法	正しいデータを検知	検知されない	検知されない

GLN のみを書き込んだ場合の結果を表 3 に示す。従来手法では、誤ったデータを検知してしまっていたが、提案手法では、正しいデータを検知することができている。そのため、本提案手法の有効性が明らかとなった。

表 4 書き込みエラーに対するデコーダの認識 (GLN とタイムスタンプ)
Table 4 Behavior of decoder to Writing Error(GLN & Timestamp)

GLN			
エラー場所	Precursor	Object Length	Object
従来手法	検知されない	GLN の誤データ検知	GLN の誤データ検知
提案手法	GLN とタイムスタンプを正しく検知	検知されない	検知されない
タイムスタンプ			
エラー場所	Precursor	Object Length	Object
従来手法	GLN のみ検知	GLN とタイムスタンプの誤データ検知	GLN とタイムスタンプの誤データ検知
提案手法	検知されない	検知されない	検知されない

評価実験の結果を表 4 に示す。表 4 に示すように従来手法では、GLN の誤ったデータとタイムスタンプの誤ったデータ検知された。そのため、従来手法では、データの信頼性が失われている。しかし、提案手法では、GLN とタイムスタンプの正しいデータのみを検知することができるため、データの信頼性を上げることができていると考えられる。

6. ま と め

本論文では、RF タグへ書き込み途中に発生する問題に着目し、データ解析エンコーダの特徴を利用した、後ろから書き込む手法を提案し、実装を行った。その結果、書き込み途中の誤ったデータを検知することがなくなり、書き込みデータの信頼性を持たせることが可能となった。また、複数のデータを一度に書き込みを行う際にも、本提案手法は有効であり、その有効性を示すことができた。

今回は、Non-Directory 方式での利用を行ったが、実装をドライバに行ったため、他の

データフォーマットでも利用可能であると考えられる。そのため、今後は他のデータフォーマットでの利用を考えていきたい。また、本提案手法は、大容量のメモリーを搭載したRFタグでも有効な手法であると考えられるため、RFタグの種類に対応も行っていきたいと考えている。そして、RFタグを用いたアプリケーションとして、通信設備保守・管理支援などの分野への応用も行っていきたいと考えている。

7. 謝 辞

本研究は、「センサー IC タグ連携ソフトウェア技術の研究開発」の成果の一部である。

参 考 文 献

- 1) ISO/IEC 18000-6. Information technology - Radio frequency identification for item management - part6: Parameters for air interface communications at 860MHz to 960MHz, 2004.
- 2) 米澤祐紀, 江原正規, 井上亮文, 徳田英幸. UHF RFID タグにおけるユーザメモリのフォーマットとリードライト性能に関する検討. 情報処理学会 マルチメディア, 分散, 協調とモバイルシンポジウム, pp. 196-201, 2009.
- 3) 米澤祐紀, 陳卓, 江原正規, 井上亮文, 星徹. UHF 帯 RF タグに適したデータフォーマット変換手法の評価. 全国大会講演論文集, No.3, pp. 55-57, 2009.
- 4) ISO/IEC 15962. Information technology - Radio frequency identification(RFID) for item management - Data protocol:data encoding rules and logical memory functions, 2004.
- 5) Seok-Young Jang, Sang-Hwa Chung, and Seong-Joon Lee. An Effective Design of an Active RFID Reader Using a Cache of Tag Memory. *Lecture Notes in Computer Science*, Vol. 4819, pp. 584-595, 2009.
- 6) 山中一哉, 米澤祐紀, 宮田宙和, 江原正規, 宇田隆哉. デジタル署名を用いた rfid タグ内部データ改竄検出手法の提案. *SCIS*, 2009.
- 7) ISO/IEC 15961. Information technology - Radio frequency identification(RFID) for item management - Data protocol:application interface, 2004.
- 8) 吉田真樹, 戸田暁博, 江原正規, 井上亮文, 星徹. RF タグのユーザメモリーを用いた流通経路記録手法の提案. 全国大会講演論文集, No.3, pp. 299-301, 2009.
- 9) UHF 帯 RFID 対応 RW モジュール 評価キット. <http://www.hitachi-jten.co.jp/products/uhf-rfid/index.html>, May 2009.

誤	正
日本電信通話	日本電信電話株式会社 NTT 環境エネルギー研究所