

様々な通信規格に対応可能な マルチプロトコルRFIDリーダライタの提案

小林 真輔^{†,††} 諸 隈 立志^{††} 坂 村 健^{†,††}

本論文では、異なる RFID 通信プロトコルに動的変更可能でかつ複数プロトコルを同時に扱うことが可能である RFID リーダライタアーキテクチャを提案する。様々な規格が存在する RFID タグに対して、リーダライタ側であらゆる RFID タグに対応することで、ユーザ側で意識することなく、様々な RFID タグを利用することが可能となる。提案方式は、高速 CPU によるソフトウェア無線や動的再構成ハードウェアによる方法ではなく、プロトコル処理をパラメータ化し、専用ハードウェアを制御するアーキテクチャをとる。これにより、携帯端末に組み込めるように小型化、低消費電力化を実現する。提案方式のプロトタイプ実装を行い、評価を行った。13.56 MHz 帯の RFID タグである ISO 15693, ISO 18092 (NFCIP-1), 2.45 GHz 帯である ISO 18000-4 Mode 1 の複数のタグに対応できることを確認し、規格の異なるタグの ID を読み出すことが確認できた。また、他方式と比較して、消費電力は約 43%に改善された。

A Proposal of Retargettable Multi-protocol RFID Reader/Writer

SHINSUKE KOBAYASHI,^{†,††} TATSUSHI MOROKUMA^{††}
and KEN SAKAMURA^{†,††}

In this paper, we propose a novel RFID reader/writer architecture that dynamically adopts a wide variety protocols of RFID by changing hardware configuration software. The proposed architecture is not software defined radio based on high performance CPU, but parameterized hardware that can be specialized RFID protocols. This architecture consumes lower power than software defined radio, because the proposed architecture does not need high performance CPUs. In experiments, we have designed a prototype that can read RFID tags: ISO 15693, ISO 18092 (NFCIP-1), and ISO 18000-4. From experimental results, we showed the prototype read tag IDs which include several kinds of RFIDs. Power consumption of the proposed system was two out of five lower than that of the others.

1. はじめに

身の回りのあらゆるところにコンピュータを埋め込み、それらが協調して、人々の生活を向上させるユビキタスコンピューティングの研究がさかに行われている¹⁾。ユビキタスコンピューティングを実現するためのキーデバイスの1つがRFID (Radio Frequency Identification) タグである。RFID タグは、パッシブタグと呼ばれタグリーダから発信される電磁エネルギーを電力として動作し応答する RFID タグとアクティブタグと呼ばれる電源を内蔵し自立的に電波を発信して通信する RFID タグがある。それぞれのタグ

は、用途によって用いられる場面が異なる。本研究では、中でも比較的安価に製造可能なパッシブタグに注目する。

現在までにパッシブタグで用いられる周波数帯は様々であり、周波数帯ごとに各種の規格が提案されている。利用周波数帯の特性にはそれぞれ一長一短があり、応用により異なる種類のタグが利用されている。たとえば、2.45 GHz 帯の RFID タグは、周波数が比較的高いために、アンテナの長さを短くすることができる。そのため、対象物が小さい場合においても使いやすい。しかし、GHz 帯の電波は水分に吸収されやすいため、水分が中に含まれているペットボトルや瓶などに付けて用いる場合、通信性能が低下する。一方、13.56 MHz 帯の RFID タグは、波長が比較的最長い分、アンテナの長さは長くなるが、水分に対しては 2.45 GHz に比べて特性が良い。また、これは制度上の問題であるが、電波の利用に関しては許認可事項で

[†] 東京大学大学院情報学環

Interfaculty Initiative in Information Studies, The University of Tokyo

^{††} YRP ユビキタスネットワークング研究所

YRP Ubiquitous Networking Laboratory

あり、国ごとに使用可能な周波数帯域が異なる。このようなことから将来のコピキタスコンピューティング環境は、1種類のタグが使われるということではなく、様々な周波数帯のタグが混在する環境と考えるべきであろう²⁾。しかし、複数種のタグに対してそれぞれ異なるリーダライタを用意するのは使い勝手からもコスト面からも問題が多い。そこで本研究では、1つのRFIDタグのリーダを様々な規格に対応させられるようなアーキテクチャを提案する。RFIDタグには複数の規格が存在し、従来、ユーザはそれぞれのタグが異なることを意識して、リーダを使い分けなければならなかった。リーダを使い分けることは、処理の煩雑性を招くだけでなく、携帯性も損なわれる。また、タグの規格は国によって使用できる規格とそうでない規格があり、利用国の電波法に合わせる必要がある。1つのリーダライタで複数タグに柔軟に対応できるマルチリーダ機能は、新たな規格のタグや独自プロトコルのタグなどが提案された場合にも、ハードウェアを変更することなく柔軟に対応できる。マルチリーダを実現する方法として、ソフトウェア無線や動的再構成ハードウェアによる方法が考えられるが、我々の目指す携帯機器に内蔵可能なマルチリーダを実現するには、消費電力や小型化の面で問題がある。マルチリーダの小型化は以下のようにコピキタスコンピューティングの応用に必須である。RFIDタグのアプリケーションを考慮した場合、様々なモノにつけられたタグをあらゆる場所で読めることが、食品トレーサビリティなどのコピキタスコンピューティングの応用から要求されている²⁾。食品トレーサビリティでは、一般の消費者が家庭でその食品が安全に作られていることをトレースできるようになることが期待されている。また、医薬品や飲食物の飲み合わせなどを調べる場合にも、医薬品や飲食物の成分などがその場で分かる必要がある。いつでも、どこでも利用できるようにするための1つの手段がマルチリーダを携帯機器に組み込むことである。現在でも携帯電話にFeliCaのRFIDリーダが組み込まれている機種がみられるが、マルチリーダも携帯機器に組み込み、ユーザがどこにおいても利用できるようにすることが、コピキタスコンピューティングを実現するうえで重要である。

我々の提案する低消費電力、小型化を実現可能とする新たな方式は、RFIDタグの通信に必要なそれぞれの処理ブロックごとにパラメータ化を行うことでソフトウェア処理の柔軟性を確保しつつ、動的再構成デバイスなどの特別な回路技術を用いずに実現できる。また、符号化に関しては様々な方式に対応可能なよう

に柔軟に指定可能な方式を実現した。さらに、受信時のフィルタ処理においても、組合せを自由に変更可能なアーキテクチャを採用することで様々なRFIDに対して柔軟に対応可能とした。このことにより、消費電力を低くおさえつつ、様々な規格に対応できる。

2. RFID タグ

RFIDタグは、電波もしくは電磁誘導を用いて非接触通信を行い、半導体書き込まれている情報の読み書きを行うことができる媒体のことである。代表的なRFIDタグの処理手順を図1に示す。まず、送信を行うためのパケットを生成する。このパケットはコマンドと送信したいデータなどが含まれる。次に誤り訂正符号を生成する。次に符号化を行う。その後、変調を行い、アンテナから出力する。受信時は、タグより送られてきた電波を増幅し、フィルタでの雑音除去などを行った後に、復調を行う。そして、復号を行う。次に、受信したデータが正しいことを確認するために誤り検出、誤り訂正を行い、その後、送られてきたデータの内容を取り出す。

現在までにRFIDタグは多数の規格が提案されている。主に国際規格団体であるISO (International Standard Organization)、IEC (International Electrotechnical Commission)において策定されている規格が、現在多く使われている。ISO/IEC 14443、15693 (13.56 MHz)、ISO/IEC 18000-2 (135 kHz)、18000-3 (13.56 MHz)、18000-4 (2.45 GHz)、18000-6 (915 MHz)、18000-7 (400 MHz) など様々な周波数帯で動作するRFIDタグの規格がある。

各規格において、各処理の共通点と非共通点を次に整理する。

通信方式：通信方式は、全2重と半2重の両方が考えられるが、RFIDタグに限って考えるとほとんどすべての規格が半2重での通信方式で規定されている。

周波数帯：周波数帯は、先に述べたとおり、規格ごとに様々な周波数帯が使用されている。主に使われているのは、13.56 MHz、915 MHz、2.45 GHzである。アンテナの大きさや通信特性などが様々に変化するため、一概にどの周波数帯が良いとはいえない。

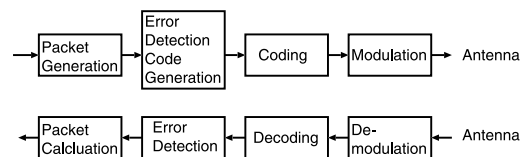


図1 RFIDタグの通信処理手順

Fig. 1 Communication flow of RFID tags.

変調方式：変調方式は比較的単純な ASK (Amplifier Shift Keying), FSK (Frequency Shift Keying) もしくは, PSK (Phase Shift Keying) を用いているのがほとんどである。変調度に関しては, ASK に関していえば, 100%である場合もあれば, そうでない場合もある。FSK に関していえば, 周波数帯によって変化させる周波数も異なる。PSK に関しては, BPSK (Binary Phase Shift Keying) を用いる場合がほとんどである。

符号化方式：符号化方式は, マンチェスタ, FM0, NRZ, ミラー, 拡張ミラー, 差分符号など様々な符号化方式がある。各規格によって採用している方式の組合せは異なる。

誤り検出：誤り検出符号は基本的に CRC (Cyclic Redundancy Check) を採用している。規格によってはパリティチェックを組み合わせている場合もある。ただし, 生成多項式は規格ごとに異なる。

コマンド群, 輻輳制御：それぞれの規格ごとに基本的には異なる。同じ方式で輻輳制御を行う場合もあるが, コマンドのフォーマットなどの細かい部分では異なっている。

フレームフォーマット：通信時のフレームフォーマットは共通点があり, Preamble, データ領域, CRC もしくはパリティチェックという順番となっている。規格によっては, Preamble の後に同期をとるためのフレームが存在する場合もある。データ領域は規格によって内容が異なる。ID の後にコマンドを続ける場合もあれば, ID のみを送信する場合もある。

3. 提案アーキテクチャ

3.1 従来方式の問題点

従来のマルチリーダは, 同一周波数帯で異なる規格もしくは異なるモードのタグ用アンテナに対して, それぞれ独立した回路を付加することで実現していた。この方法では, 確実に想定しているタグに対して対処することができるが, 新規のタグに柔軟に対応することはむずかしい。柔軟に対応できるようにソフトウェアですべての処理を行うソフトウェア無線^{(3)~(7)}を使い, マルチリーダを実現することも考えられる。実際に, いくつかはソフトウェア無線により開発されている^{(8)~(10)}。しかし, この方式の場合, フィルタリングなどを行うために高性能な MPU (Micro Processing Unit) や DSP (Digital Signal Processor) が必要となり, 消費電力が大きくなる。ソフトウェア無線をリコンフィギュラブルロジックを用いて実現する手法^{(11)~(16)}も提案されている。リコンフィギュラブル

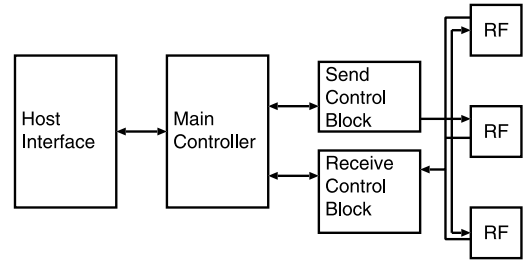


図 2 全体図
Fig. 2 Overview.

ロジックは, MPU よりは低消費電力を実現でき, さらに専用ロジックに近いレベルでの処理性能が実現できるが, 多くの場合, 通常の専用ロジックと比較すると消費電力は高く, 回路規模も大きくなる。そのため, 携帯機器に取り入れるのは不向きで, 利用に制限が生じる問題点がある。そこで, 柔軟性を残しつつ, さらに高性能な MPU などを用いず, 消費電力が大きくならないような実現方式, アーキテクチャが要求されている。

3.2 アプローチ

低消費電力, 小型化の観点から, ソフトウェア無線のようにすべてソフトウェアでの処理とはせず, ハードウェア上で柔軟に処理可能な方式を開発する。特に, フィルタ処理や符号化処理など CPU で処理を行うには不向きな処理は基本的にハードウェアでの対応とする。また, ハードウェアで実現する場合に不向きになると考えられるプロトコル制御 (コマンド選択や輻輳制御処理など) は, ソフトウェアで行う。そして, これはソフトウェア無線の方式でも共通であるが, 極力デジタル処理可能とするためにアナログ部分は最小限にとどめる。このことにより, デジタル処理において柔軟に他の方式への対応が可能となる。

3.3 全体構成

図 2 に全体の構成図を示す。メインコントローラより RF (Radio Frequency) 回路制御ブロック, 送信制御ブロック, 受信制御ブロックの制御を行う。RF 回路はそれぞれの周波数帯ごとに用意し, 選択する周波数帯に応じて駆動する方式をとる。RF 回路制御ブロックは, それぞれの RF 回路のパラメータを設定するための回路である。送信制御ブロックは送信用のフレームを生成し, 符号化を行って出力する。また, 受信制御ブロックは, 受信したデータを解析し, 復号してデータを取り出すための回路である。以降では, 送信制御ブロック, 受信制御ブロックに関して説明を行う。



SOF; Start of Frame
 SYNC; Synchronization
 DATA; Data including commands and parameters
 CRC; CRC or parity
 EOF; End of Frame

図 3 送信フレームフォーマット

Fig. 3 Send frame format.

3.4 送信制御ブロック

通常、RFID タグのコマンド転送に用いられるフレームは、プリアンブルを用いて同期をとり、その後データを送信する。無線でのデータ転送時にはエラーが入ることが多いため、データが正しく転送されていることを確認するために、CRC などの誤り検出符号を入れることが多い。以上のようなことをふまえて、提案方式では図 3 に示すフォーマットで送信フレームを作成する。SOF (Start of Frame) は、フレームの開始を示す。SYNC は、データ部分の始まりを示す。DATA はデータ部分を示す。CRC は CRC もしくはパリティを示す。EOF (End of Frame) はフレームの終点を示す。送信フレームはそれぞれ RFID の仕様によって異なる。仕様の表現によっては、SYNC と呼ばれる部分が存在しないなどあるが、ここでは単純にフレームの開始を示すビット列を SOF と呼び、データ部分の開始を示す直前のビット列を SYNC と呼ぶ。また、データ部分は RFID タグの仕様によってコマンドが最初に入ったり、コマンドのパラメータが入ったりするが、ここでは特にコマンドの内容やパラメータや実際のユーザのデータに関する順番は問わない。単純にユーザ指定のデータ領域という意味で使用することによって、様々な種類の RFID タグのフレームフォーマットを共通に扱うことができるようになる。

送信制御ブロックの構成を図 4 に示す。規格で定められている送信コマンドとパラメータを含んだ送信データをソフトウェアによって生成する。その後、送信データをいったん送信データバッファに格納する。次に、CRC 生成ブロック、パリティ生成ブロックにおいて、CRC もしくは、パリティ生成を行う。そして、パターン読み出し制御ブロックで、パターンバッファに格納されている各符号のパターンと送信データを照らし合わせ、パターン生成ブロックで最終的に符号を生成する。パターン読み出し制御ブロックは、パターンバッファに格納されている符号化パターンの読み出しとパターン生成ブロックへの符号出力の指示を行う。符号化対象となるデータのビット列を読み出し

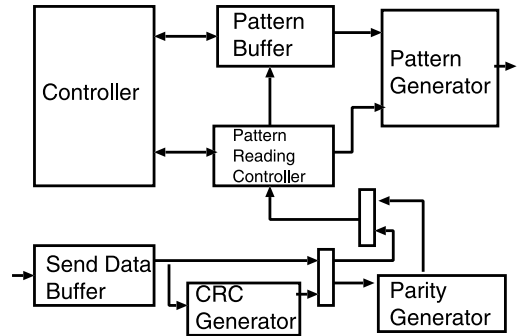


図 4 送信制御ブロック

Fig. 4 Send control block.

ながら、必要なパターンを読み出し、パターン生成ブロックへ入力して処理を行う。CRC 生成は、生成多項式をパラメータとして選択可能にすることでそれぞれの CRC に対応する。パリティ生成に関しては、偶数パリティもしくは奇数パリティの選択を可能とする。符号化のパターンに関しては、次に示すようにパターンを指定可能にすることで様々な符号化に対応可能とした。

提案手法では、バイナリの 0 を指定する場合と 1 を指定する場合でそれぞれ符号化パターンを指定する方式を用いる。図 5 (a) に符号化パターンの指定フォーマットと例を示す。サイクル数、繰返し回数、最終ワードを示すフラグ、反転を指定するフラグ、初期値の順番で指定する。サイクル数では、全体の繰り返すサイクル数を指定する。繰返し回数では、ビットの繰返しを何回繰り返すかを指定する。最終ワードフラグは、指定したパターンで最終であることを示す。反転指定は、繰り返す場合にビット反転を行うかどうかを指定する。初期値では最初に始める値を指定する。このようなパターンをパターンバッファに格納し、符号化を行う場合に、値が 0 か 1 かでパターンを選択して符号を生成する。ただし、FM0 (Bi-Phase space) のように直前の送信波形に依存する場合には、符号化パターンで指定するのは別に、ハードウェアパラメータとして、直前の値、もしくはその反転をした値を用いることを指定する。図 5 (b), (c) の例で示している Code selector がその値を示しており、00 は直前の値は使用しない、01 は直前の値を用いる、10 は直前の値を用いてかつ反転させることを指定可能としている。また、図 3 にある SOF, SYNC, EOF に関しても、同様にパターンを指定する。ただし、SOF, SYNC, EOF はすべてのフレームで共通であるため、ハードウェアパラメータの初期設定時にフォーマットを指定するのみとし、通常時は無条件でフレームを作る場合に付加す

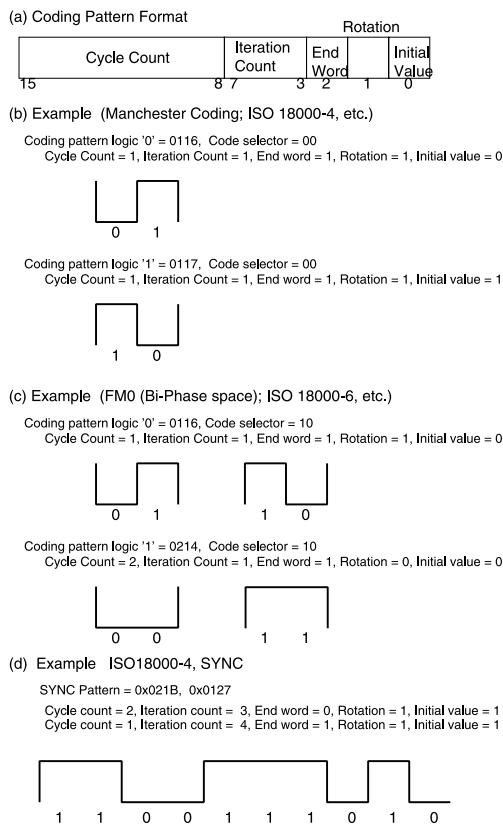


図 5 送信用符号化パターン

Fig.5 Coding pattern for sending data.

る．例として，Manchester，FM0 の 2 種類の符号化をあげる．Manchester は，ISO 18000-4，ISO 18092 などで使用されている符号化方式であり，FM0 は ISO 18000-4 の受信時や ISO 18000-6 などで使用されている符号化方式である．Manchester の場合は，‘0’ 値の場合は，サイクル数を 1，繰返し回数を 1，反転を行い，初期値を 0 として指定する．‘1’ の場合は，その逆に初期値に 1 を指定する．FM0 の場合は，直前のデータを反転して用いるため，Code selector の値を 10 として指定する．

従来ハードウェアで処理を行う場合は，ある決められた符号化方式に対応した回路を用いている．複数の符号化方式に対応するためには，それぞれ独立した回路を用いて処理を行っていた．以上のアルゴリズムを採用することで，符号化処理をハードウェアで柔軟に対応することが可能となる．ソフトウェアで処理を行う場合でも柔軟に対応可能であるが，タグとの通信を行うリアルタイム処理を考えた場合，ハードウェアで実現する場合よりも高い周波数で動作させることが必要となる．そのため，ハードウェアで実現するほうが低消費電力化が期待できる．

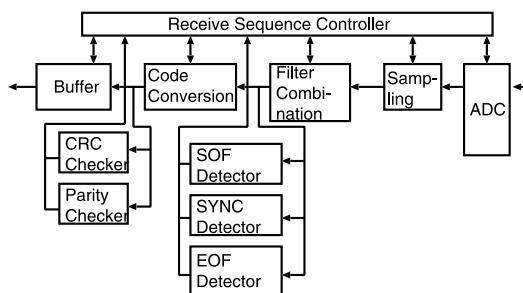


図 6 受信制御ブロック

Fig.6 Receive control block.

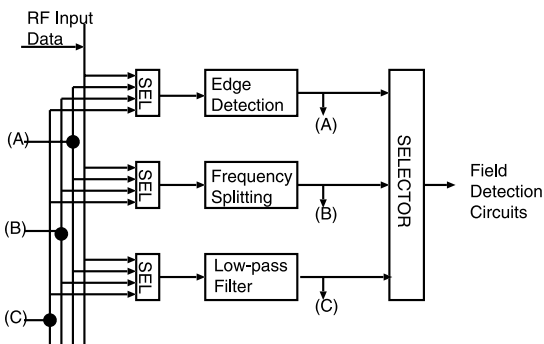


図 7 フィルタ組合せブロック

Fig.7 Filter combination block.

3.5 受信制御ブロック

受信制御ブロックの構成を図 6 に示す．受信制御ブロックは ADC (Analog to Digital Converter) からの出力をサンプリングし，フィルタ組合せブロックでフィルタリングを行う．フィルタリングを行った後に，SOF 検出，SYNC 検出，EOF 検出などの各パターンの検出をそれぞれ SOF 検出ブロック，SYNC 検出ブロック，EOF 検出ブロックで行うか，コード変換ブロックで復号化を行う．復号化は 3 値で表現された受信パターンに適合するかどうかを比較し，適合するパターンによって 0 もしくは 1 であると解釈する．結果を CRC チェックブロック，パリティチェックブロックで誤り検出や検出などを行い，正しいと判断されたデータを受信バッファに格納する．

波形の抽出処理をフィルタ組合せブロックで行う．このデジタルフィルタの処理順やデジタルフィルタに与えるパラメータを変更可能にすることで，所望の波形を取り出すことが可能となる．図 7 にフィルタ組合せブロックを示す．フィルタ組合せブロックは，アナログ回路より入力されてきた波形をフィルタリングし，必要な部分の波形だけ抽出するための回路である．複数のフィルタを選択的に使用することができる．エッジ検出フィルタ，周波数分離フィルタ，Low パスフィ

Ex. SOF of ISO18000-4

Pattern Register : 0xA911

Expected Value : 10101xxx ('x': don't care)

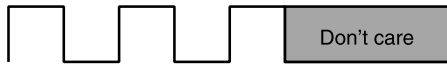


図 8 受信波形指定の例

Fig. 8 An example of wave form pattern assignment.

ルタをセレクタによって切り替えることで選択的に使用可能である。ここで各フィルタを選択的に使用するとき各セレクタでは、RF からの入力データ、それぞれのフィルタの出力から選択することが可能である。このようにすることで、カスケードに接続することができるため、フィルタの組合せを自由に選択することが可能である。

ソフトウェアでも柔軟に処理することは可能であるが、RFID タグよりの応答をリアルタイムで処理する場合に、数百 MHz 以上の CPU での処理が必要となる。そのため、提案する方式のようにハードウェアによるフィルタを柔軟に組み合わせることでフィルタ処理を実現するメリットは大きいといえる。

送信の場合と同様に、受信の場合も待ち受けるビットパターンを指定する必要がある。ただし、待ち受けるビットパターンは送信の場合の指定方法とは異なり、1, 0, Don't care の 3 値による指定を行う。3 値を導入する理由としては、受信は送信の場合と異なり、どのような信号が入力されるかが不定であるため、あらゆる入力を考慮しなければならない。あらゆる入力を考慮する場合、パターンの数が多くなりすぎる。2 値で指定する場合よりも 3 値で指定する場合の方が減らすことが可能となることから、3 値で指定する方式を採用する。3 値をビットで表現し、1 は“01”，0 は“00”，Don't care は“11”もしくは“10”となる。図 8 に示している例を用いて説明する。10101xxx (x は Don't care) を指定した場合を考える。受信パターンの指定としては、“0xA911”を指定すると 10101xxx を指定したことになる。0xA911 を下位の 2 ビットずつ順番にみた場合、“01”，“00”，“01”，“00”，“01”，“10”，“10”，“10”とみることができ、それぞれ、1, 0, 1, 0, 1, x, x, x と指定できていることが分かる。

3.6 RF 回路

RF 回路で行う処理はなるべく少なくし、デジタル側で処理することで柔軟性を向上させる方針を採用しているため、RF 回路は単純な構成を採用している。図 9 に RF 回路ブロックを示す。送信側は、送信制御

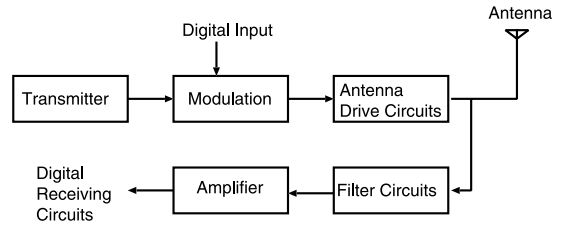


図 9 RF 回路ブロック

Fig. 9 RF block.

ブロックから出力された信号を変調し、アンテナ駆動回路を経て、アンテナから出力される。また、受信信号はフィルタ回路を経て、信号を増幅し、デジタル側の受信制御ブロックに出力する。RF 回路は各周波数帯で用意し、それぞれの変調方式、変調度に関してはパラメータとして変更できるようにする。このことにより、方式の異なる RFID タグにおいても対応可能となる。

4. 評価実験

提案方式の評価結果について述べる。2.45 GHz 帯である ISO/IEC 18000-4, 13.56 MHz 帯である ISO/IEC 15693, ISO/IEC 18092 (NFCIP-1) について実装し、評価した結果について述べる。

4.1 ハードウェアアーキテクチャ

RF 回路ブロックは 2.45 GHz, 13.56 MHz の 2 つの周波数ごとに実装を行った。ホスト CPU としては、SH7727 (SH3-DSP, 96 MHz)、ボードとして T-Engine SH7727 (フラッシュメモリ 8 MB, SDRAM 32 MB) を用いて実装を行った。デジタル部は、FPGA (Xilinx Virtex-II2000, XC2V2000-4FG676C) を用いて実装を行った。今回の実装では、CRC に関しては、CRC-5 ($X^5 + X^2 + X^1$), CRC-24 ($X^6 + X^2 + X^1$), CRC-16 ($X^{16} + X^{15} + X^2 + 1$), CRC CCITT ($X^{16} + X^{12} + X^5 + 1$) の 4 種類を選択可能として実装を行った。また、アンテナ部は 2.45 GHz に関しては $12 \times 12 \times 7$ mm のサイズのセラミックアンテナを用い、出力を 300 mW とした。13.56 MHz に関しては 58×58 mm の方形コイルアンテナを用い、出力を 93 mW とした。

4.2 ソフトウェアアーキテクチャ

図 10 にソフトウェア構成を示す。それぞれの RFID タグに対応したデバイスドライバとライブラリを T-Kernel¹⁷⁾ 上に実装した。アプリケーションからは、ライブラリを起動することでハードウェアを駆動する構成とした。デバイスドライバではレジスタ値を設定するための最低限の関数群を実装し、ライブラリ中で

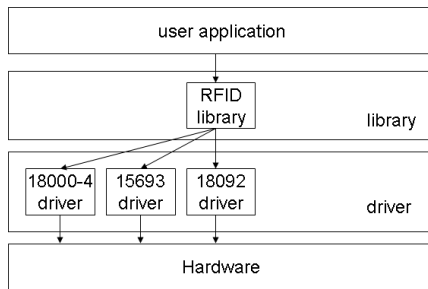


図 10 ソフトウェア構成
Fig. 10 Software architecture.

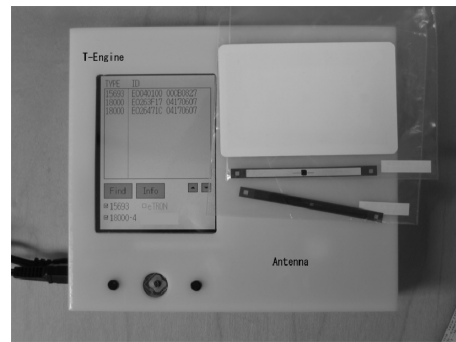


図 12 プロトタイプハードウェア
Fig. 12 A prototype hardware.

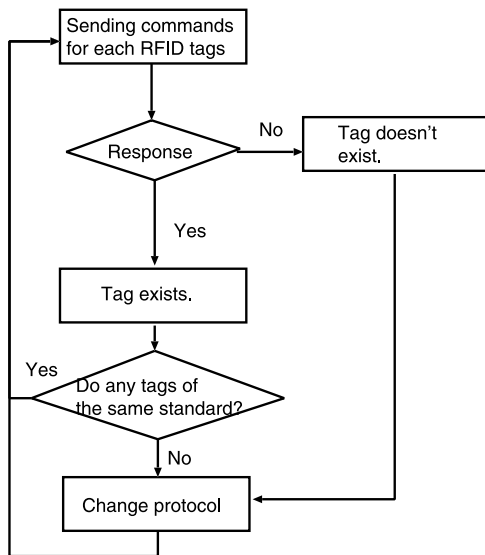


図 11 処理フロー
Fig. 11 Control flow.

はそれらの関数群を用いて輻輳制御やプロトコル制御を実現した。

複数のタグの読み取り評価用のソフトウェアを実装して評価を行った。図 11 に複数の RFID タグを読み出す場合の処理フローを示す。それぞれのタグの規格に応じたコマンドを順に送信する。タグの読み取りを行い、レスポンスがあった場合にはタグが存在するとして、ID を読み出し、そうでない場合は、該当する規格のタグが存在しないと判断する。ID 読み出しが成功した場合は、同じ規格の他のタグが存在するかを調べるために同じ規格のコマンドをもう一度送りなおす。これをすべての規格のタグの検出を行うまで、時分割で行う。RFID タグはそれぞれプロトコルや通信時のパラメータが異なるため、たとえ同じ周波数帯を用いても異なる規格のタグの場合、同時に応答することはない。したがって、すべてのタグについてコマンドを送信することで検出を行うことが可能である。

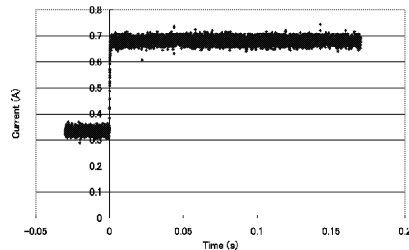


図 13 FPGA ボードの消費電流
Fig. 13 Current of FPGA board.

4.3 結果

上記のような仕様に従って、プロトタイプを作成した。図 12 は、動作させたときの写真である。実験では、ISO 18000-4 のタグを 2 枚、ISO 15693 のタグを 1 枚、ISO 18092 のタグを 1 枚用いた。これらのタグをアンテナの上ののせた状態でタグの ID を読み出すソフトウェアを起動させて動作を確認した。

FPGA 上に実装されたデジタル回路の規模は、4 入力 LUT (Look Up Table) が 8,948 個であり、フリップフロップ数は 5,958 個であった。また、動作周波数は RFID のベースバンド周波数の中で最も低い 13.56 MHz で設計した。上記で説明したようにタグをアンテナの上ののせてタグの ID を読み込ませたところ、すべてのタグの ID を読み取ることを確認できた。また、タグの ID を読み込んだ場合の電流を調べた。図 13 は、タグ読み込み時の FPGA ボードの消費電流であり、図 14 は、タグ読み込み時の T-Engine ボードの消費電流である。図 14 においては、プロトコルを切り替えるタイミングで表示の更新とプロトコルの切替えが行われている。約 20 ms の処理時間のほとんどは表示の切替えに要している時間である。プロトコルの切替えは今回の実装では 1 クロックサイクルで実現できているため、図には明確に現れていない。T-Engine ボードは平均 419 mA (DC 5.6 V)、提案方式を実装

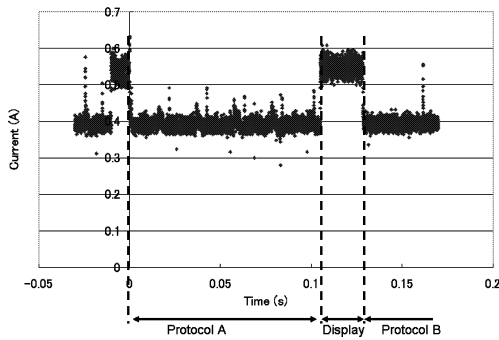


図 14 T-engine ボードの消費電流
Fig. 14 Current of T-engine board.

した FPGA ボードならびに RF 回路の消費電流の平均は 681 mA (DC 5.6 V) であった。また、1 つのタグを処理する場合に、1 プロトコルあたりの処理時間は約 100 ms であった。また、通信距離は ISO18092 は約 2 cm, ISO15693 は約 10 cm, ISO18000-4 は約 12 cm であった。

5. 考 察

提案方式を実装し、仕様の異なる複数種類のタグに関してコマンド送信可能であることを確認することができた。処理時間に関しては、時間制約がある受信処理が問題となる。受信処理の中でもサンプリング、フィルタリング処理が時間制約のある処理となるが、今回の実装ではソフトウェアによる処理ではなく、ハードウェアでのサンプリングならびにフィルタリング処理を行っているため、問題とはならなかった。また、今回の実装では、複数種類の RFID タグのパラメータをあらかじめ FPGA のメモリ上に設定しておき、その設定を読み出す方式で実現した。そのため、タグの種類切替え時の処理時間のオーバーヘッドはほとんどなかった。dynamic configuration FPGA を用いた場合、仮に Altera 社の Cyclone II を用いた場合には、40 MHz クロックで構成変更に少なくとも 50 ms 程度は必要であると考えられる¹⁸⁾。1 プロトコルあたりの処理時間が約 100 ms であったことを考慮すると、処理時間の 50% であり、無視できない時間であるといえる。

消費電力に関しては、仕様の異なるタグをソフトウェア無線で扱う方式で実装している ThingMagic の Mercury4e⁸⁾ は、ホスト CPU に Intel 製 Xscale プロセッサ (IXP420 266 MHz)、ベースバンド処理部を Texas Instruments 製の DSP (Ti5502 300 MHz) を使用しており、無線通信部分の消費電力は 8.8 W になる。受信データをリアルタイムに処理することを考えた場合に、30 から 50 μ s でフィルタ処理や復号処

理を行う必要があり、それをソフトウェアで処理することを考えると少なくとも数百 MIPS 程度の処理能力のある CPU が必要となると考えられる。本提案では、その DSP 処理を専用回路で処理することで低消費電力化を実現している。実験データより、T-Engine が 2.25 W, FPGA ボードが 3.81 W であった。Mercury4e と比較した場合、無線通信部分のみで考えると消費電力が 43% に削減できていることが分かった。ISO18000-4 用のリーダーと比較した場合、今回試作したマルチプロトコルリーダーとはほぼ同等の送信出力である PC カード型のリーダーでは、消費電力が約 2.5 W であった。通信部分が実装されている FPGA ボードとの比較で約 1.5 倍の電力になっているが、他の周波数帯のアナログ回路、デジタル回路が含まれていないことを考慮すると、妥当な結果であると考えられる。また、回路規模について考えた場合、単体の ISO18000-4 の回路の場合およそフリップフロップが 1,400 で、LUT が 2,200 程度であったことから、約 4 倍程度になっている。単純に 3 種類のプロトコルを実装した場合に面積が単体の場合の 3 倍になると考えると、専用回路の回路規模の方が小さいが、マルチプロトコルリーダーの方は、プロトコルを追加した場合においても想定しているパラメータの範囲内であれば、回路規模は増加しない。一方、専用回路の場合、基本的には他のプロトコルへの対応は柔軟にできない。専用回路として作り込んだ場合に比べ、冗長性を持たせている分、今回の実装のように 3 種類程度では回路規模や面積は不利にはなるが、柔軟性は提案方式の方がすぐれているといえる。

提案方式では、フィルタの組合せを選択できる構成とすることで、RFID タグに必要なフィルタ処理をフレキシブルに処理することができる。様々な RFID に対応するためには、ノイズ除去などをそれぞれの RFID に合わせて行う必要がある。ソフトウェア無線と比較して、ハードウェア化していることにより、動作周波数を低く抑えることが可能であり、このことが低消費電力化につながっているといえる。また、専用回路を用いる場合と比較してソフトウェア無線の場合は複数の高性能プロセッサを用いる必要があるためシステムの大型化を招くことから、携帯端末への実装は難しいといえる。本方式では、専用回路を用いる方式であるため、システムの小型化が可能である。その観点からみた場合においても、本方式の方が携帯端末に向いている方式であるといえる。そして、今回実装した回路を LSI 化することで、さらに小型化、低消費電力化を実現できる見通しを得ることができた。

6. おわりに

本論文では、規格の異なる RFID タグに動的に対応可能な方式を提案した。ソフトウェア無線方式とは異なり、RFID タグのプロトコル処理に必要な部分で演算量が多い部分は、ハードウェアをパラメータ化して実現することにより、低消費電力、小面積での実現を可能とした。送信、受信の処理をそれぞれパラメータ化することで、規格の異なる RFID タグの方式にハードウェアを変更することなく、ソフトウェアを変更するだけで対応可能である。評価実験により、提案する方式を実装し、その有効性が示された。

今後は、さらなる小型化、低消費電力化に向けて LSI 化を行い、小型端末の実現を行う予定である。

謝辞 研究を進めるにあたって様々なご助力をいただいた(株)日立超 LSI システムズの村上昌之氏、深見祐二氏に感謝する。また、本プロジェクト推進時にはユビキタスネットワークング研究所の所員であった(株)日立製作所の早川幹氏に感謝する。なお、この研究開発の一部は、情報通信研究機構の受託研究による。

参考文献

- 1) Krikke, J.: T-Engine: Japan's Ubiquitous Computing Architecture Is Ready for Prime Time, *IEEE Pervasive Computing*, pp.4-9 (2005).
- 2) 坂村 健: ユビキタス, TRON に出会う—「どこでもコンピュータの時代へ」, NTT 出版 (2004).
- 3) Bauracchini, E.: The Software Radio Concept, *IEEE Communications Magazine*, pp.138-143 (2000).
- 4) Engel, J.: DSP for RFID, *45th IEEE International Midwest Symposium on Circuits and Systems*, pp.227-230, IEEE (2002).
- 5) Mitola, J.: The Software Radio Approach, *IEEE Communications Magazine*, pp.26-38 (1995).
- 6) Baines, R.: The DSP Bottleneck, *IEEE Communications Magazine*, pp.46-54 (1995).
- 7) Walter, H.W.: Tuttlebee: Software-Defined Radio: Facets of a Developing Technology, *IEEE Personal Communications*, pp.38-44 (1999).
- 8) ThingMagic: Mercury4.
<http://www.thingmagic.com>
- 9) NEC システムテクノロジー: FlexReaderWriter.
<http://www.necst.co.jp/press/20060628/index.htm>
- 10) パナソニックコミュニケーションズ株式会社: マルチプロトコル対応可能 UHF RFID リーダライタ . <http://panasonic.co.jp/corp/news/official.data/data.dir/jn060711-1/jn060711-1.html>
- 11) Srikanteswara, S., Palat, R.C., Reed, J.H. and Athanas, P.: An Overview of Configurable Computing Machines for Software Radio Handsets, *IEEE Communication Magazine*, pp.134-141 (2003).
- 12) Brakensiek, J., Oelkrug, B., Bucker, M., Uffmann, D., Droge, A., Darianian, M. and Otte, M.: Software Radio Approach For Re-configurable Multi-standard Radios, *13th IEEE International Symposium Personal Indoor and Mobile Radio Communication*, pp.110-114, IEEE (2002).
- 13) Hauser, J.R. and Wawrzyniek, J.: Garp: A MIPS Processor with a Reconfigurable Coprocessor, *5th Annual IEEE Symposium on FPGAs for Custom Computer Machines*, pp.12-21 (1997).
- 14) Mohebbi, B., Filho, E.C. and Kurdahi, F.J.: A Case Study of Mapping a Software-Defined Radio (SDR) Application on a Reconfigurable DSP Core, *1st IEEE/ACM/IFIP International Conference on Hardware/Software Codesign and System Synthesis*, pp.103-108 (2003).
- 15) Heidari, G. and Lane, K.: Introducing a Paradigm Shift in the Design and Implementation of Wireless Devices (2003). White paper available at <http://www.qstech.com>
- 16) Taylor, M.B., Kim, J., Miller, J., Wentzloff, D., Ghodrati, F., Greenwald, B., Hoffman, H., Johnson, P., Lee, J.W., Lee, W., Ma, A., Saraf, A., Seneski, M., Shnidman, N., Strumpfen, V., Frank, M., Amarasinghe, S. and Agarwal, A.: The RAW Microprocessor: A Computational Fabric For Software Circuits and General-Purpose Programs, *IEEE Micro*, Vol.22, No.2, pp.25-35 (2002).
- 17) 坂村 健 (監修), T-Engine フォーラム (編著): T-Kernel 標準ハンドブック, パーソナルメディア (2005).
- 18) Altera Corporation: Configuration Handbook, Vol.1. <http://www.altera.co.jp/literature/lit-config.jsp>

(平成 19 年 1 月 9 日受付)

(平成 19 年 6 月 5 日採録)



小林 真輔（正会員）

2003年大阪大学大学院基礎工学研究科博士課程修了。博士（工学）。同年大阪大学大学院情報科学研究科特任教員，東京大学大学院情報学環助手，2006年より東京大学大学院情報学環特任助教授，2007年より東京大学大学院情報学環特任准教授。YRP ユビキタス・ネットワークング研究所主幹研究員を兼任。コンピュータアーキテクチャ，VLSI デザイン，LSI 設計手法，組み込みシステム，ユビキタスコンピューティング等の研究に従事。IEEE，電気情報通信学会各会員。



諸隈 立志

1979年慶應義塾大学大学院工学研究科電気工学専攻修士課程修了。YRP ユビキタス・ネットワークング研究所研究開発推進部長。ユビキタス・ネットワークングの研究開発に従事。



坂村 健（正会員）

1951年生まれ。東京大学大学院情報学環教授・工学博士。専攻はコンピュータ・アーキテクチャ（電脳建築学）。1984年からオープンなコンピュータ・アーキテクチャTRONを構築。ユビキタス環境のための組み込みOSとして世界中で多数使われている。さらに，コンピュータを使った電気製品，家具，住宅，ビル，都市，ミュージアムなど広範なデザイン展開を行っている。2002年よりYRP ユビキタス・ネットワークング研究所長を兼任。IEEE フェロー。2003年紫綬褒章。2006年日本学士院賞。『変わる国，日本へ』，『ユビキタスで作る社会情報基盤』，『グローバルスタンダードと国家戦略』等，著書多数。

