

平成18年度  
研究開発成果報告書

超小型汎用コミュニケーション端末のための  
基盤技術の研究開発

委託先： (株)横須賀テレコムリサーチパーク

平成19年4月

情報通信研究機構

# 平成18年度 研究開発成果報告書 (一般型)

「超小型汎用コミュニケーション端末のための基盤技術の研究開発の研究開発」

## 目次

1	研究開発課題の背景	2
2	研究開発の全体計画	4
2-1	研究開発課題の概要	4
2-2	研究開発目標	4
2-2-1	最終目標	4
2-2-2	中間目標	6
2-3	研究開発の年度別計画	7
3	研究開発体制	8
3-1	研究開発実施体制	8
4	研究開発実施状況	9
4-1	超小型マルチプロトコル RFID R/W 装置の実現	9
4-1-1	パッシブ型タグ	9
4-1-2	アクティブ型タグ	13
4-1-3	まとめ	19
4-2	サーバー・クライアントの協調処理型ブラウザの研究開発	20
4-2-1	研究開発内容	20
4-2-2	ユーザインタフェース	20
4-2-3	ブラウザ・ソフトウェア構成	22
4-2-4	プラグイン・マネージャ	25
4-2-5	UI マネージャ	26
4-2-6	HTML プラグイン	28
4-2-7	SVG プラグイン	34
4-2-8	動画プラグイン	35
4-2-9	スクリプト言語	36
4-2-10	まとめ	36
4-3	小型化・省電力化のための S/W・H/W アーキテクチャの研究開発	37
4-3-1	ハードウェアプラットフォーム	37
4-3-2	ソフトウェアプラットフォーム	46
4-3-3	まとめ	59
4-4	総括	59
5	参考資料・参考文献	60
5-1	研究発表・講演等一覧	60

## 1 研究開発課題の背景

20世紀後半から、情報通信技術（ICT: Information and Communication Technology）の急速な進展と広範な普及によって、我々の社会は大きく変革し、いわゆる情報社会へと突入した。特に近年、小型化されたコンピュータや通信ノードを身の回りのあらゆるところに埋め込み、現実世界の状況情報を獲得しながら、国民の生活の質の向上に寄与するという、新しいICTのパラダイムとして、ユビキタスコンピューティング及びユビキタスネットワークを日本は世界に対して提案し、世界を牽引している。

### （1）先行研究の状況とユビキタスコンピューティング技術の進展

2001～2005年度の5年間において、当社では、民間基盤技術研究促進制度に基づいた「ユビキタスコンピューティング環境を実現するための基盤ネットワークプロトコルの研究開発」の事業を実施し、世界で始めてユビキタスコンピューティング環境の技術体系の構築に取り組み、高い成果を上げた。具体的には、基盤となる通信プロトコルを核として、それを実現するためのハードウェアやソフトウェアアーキテクチャ、ユーザインタフェース、セキュリティーなどの基盤技術とともに、物流支援や安心安全の実現応用、ロケーション依存情報サービスなどの様々な応用に即した研究開発を実施した。これによって、ユビキタスコンピューティング環境を実現するために不可欠な基盤ハードウェアアーキテクチャや基盤ソフトウェア、基盤通信プロトコルなどを確立することに成功し、現在それらの技術のうち既に開発段階を終えたものについては普及段階になっている。

例えば、ユビキタスコンピューティング環境を構成するノードを構築するためのオープンソースの次世代組込リアルタイムカーネルであるT-Kernelは、既に2000以上のライセンス契約を結び、実製品に組み込まれている。また、ユビキタスコンピューティング環境をセキュアにする統合フレームワークeTRON (Entity and Economy TRON)は、耐タンパー性にあるスマートカードタグとして、実製品が出荷されてきた。超小型リアルタイム通信ノードnT-Engine (Nano-T エンジン)も完成し、工業制御用機器として展開を図っている。そのほかにも、T-Engineというユビキタスコンピューティングノードの開発用ハードウェアのオープンプラットフォームも国内各社より二十種類以上もの製品が出荷され現在もバリエーションを増やしており、ユビキタスコンピューティングノード向けにJavaによるソフトウェア開発環境もリリースしている。

### （2）ユビキタスコンピューティング環境におけるユーザ端末の機能

上記の研究開発事業の中で、ユビキタス情報サービスを提供するために、ユビキタスコンピューティング環境を構成する諸機能を実現するユーザ端末（ユビキタスコミュニケーター：UC）の試作も行ってきた。UCは、ユビキタスコンピューティング環境と人間が対話するために利用する携帯型の汎用コミュニケーション端末であり、ユビキタスコンピューティング環境内のモノや場所に設置された、RFIDやアクティブタグ、アクティブマーカ等から識別番号等の情報を受け取り、それをきっかけとしてモノや場所に関する情報サービスを提供する。

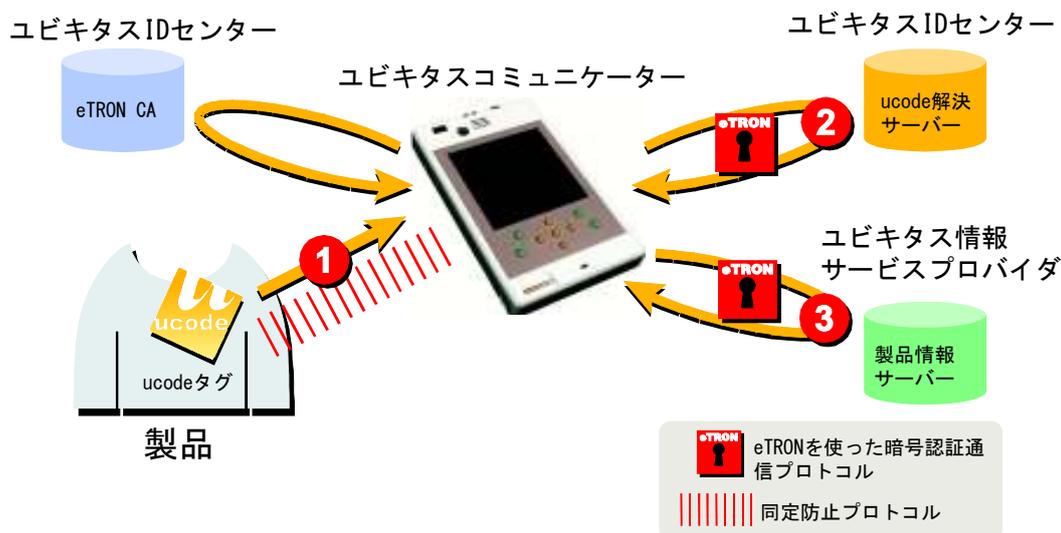


図1：ユビキタスコンピューティング環境の情報配信アーキテクチャ

従って、UCには、様々な種類のRFID等の電子タグ、センサーネットワークなどのアクティブ型通信ノード、赤外線などのビーコン、BluetoothやZigbeeのようなPANレベルの無線通信、WiFiや携帯電話網などの広域網へ接続するための広帯域通信といった多様な通信機能を搭載しつつ、初心者ユーザに対しても魅力的なユーザインタフェースやマルチメディア機能といった機能を欠かすことはできない。

### (3) 究極の超小型ユーザ端末とその機能要求

このように、2005年度までの先行研究によって、ユビキタスコンピューティング環境を実現する基盤となる諸技術を確認することができた。我々は、更にこの成果の上へステップアップして、真のユビキタスコンピューティング技術を浸透させることを狙っている。そのためには、以下の三点点が重要であると考えている。第一に、ユビキタスコンピューティング環境とユーザの間のインタフェースが重要である。特にそれらの間のインタフェースとなるコミュニケーション端末（今までの当社の研究ではUCと呼んできたもの）は、究極まで小型軽量化・省電力化をする必要がある。なぜならば、既に人々の間では携帯電話やiPodのような音楽プレイヤーなど、非常に小さい端末機器が普及している。ユビキタスコンピューティング環境とのインタフェースとなる機器の機能は、これら既存の機器とは比べ物にならないほど高いものであったとしても、現在普及している機器よりも大きい機器は、恐らく利用者には受け入れられないと考えられるからだ。

第二に、使いやすい利用者インタフェースを提供しなければならない。こうした端末で提供する情報サービスは、情報配信や情報提供であり、そのインタフェースとなるソフトウェアはブラウザである。このブラウザが、使いやすく、かつ上記で述べた究極の超小型端末のハードウェアで軽快に動作しなければならない。

第三に、サービスのユビキタス性（遍在性）が大切である。利用者の満足を得るためには、いつでもどこでも情報サービスが受けられることが重要である。現在携帯電話の通話地域はかなりユビキタスになってきているのと同様に、RFIDや電子タグを読み取るサービスについてでも、いつでもどこで、どのタグでも一つの端末でアクセスできるユビキタス性が不可欠である。

このように極めて豊富な機能を小さい端末に入れ込むためには、単なる既存技術をチューニングしたり、実装技術のノウハウの蓄積だけで実現することは不可能であり、小型軽量化を実現するためのハードウェア技術とソフトウェア技術の両面からのブレークスルーが必須である。

## 2 研究開発の全体計画

### 2-1 研究開発課題の概要

現在当社の研究開発事業によって、ユビキタスコンピューティング環境を実現するために不可欠な基盤ハードウェアアーキテクチャや基盤ソフトウェア、基盤通信プロトコルなどを確立した。現在既に開発段階を終えたものは普及段階になっている。更にこの成果の上へステップアップし、真のユビキタスコンピューティング技術を浸透させるためには、次の三点が重要である第一に、ユビキタスコンピューティング環境とユーザの間の I/F 端末が重要である。特に、インタフェース機器の究極の小型化が重要である。第二に、使いやすい利用者インタフェースが重要である。第三に、サービスのユビキタス性（遍在性）が大切であり、特にいつでもどの RFID タグでも一つの端末でアクセスできるユビキタス性が不可欠である。

そこで、本研究開発課題では、タバコ箱大までに究極に小型化された超小型汎用コミュニケーション端末を実現するために必要な、基盤ハードウェア及び基盤ソフトウェア技術の研究開発を目的とする。こうした端末を究極まで小型化するためには、単なる既存技術をチューニングしたり、実装技術のノウハウの蓄積だけで実現することは不可能である。ハード・ソフトの両面から、従来とは全く異なる方式によるブレークスルーが必須である。本研究開発課題は3つのサブテーマからなる。①ソフトウェア制御型の RFID マルチプロトコル R/W の研究開発。②ヒューマンフレンドリなユーザインタフェースをコンパクトかつ高性能で実現するために、サーバ・クライアント間で不可分散が実行環境に応じて動的にできるブラウザの研究開発。③これらの要素技術を一つのプラットフォームに統合するためのプラットフォームアーキテクチャの研究開発。

本研究によって、複数種類のパッシブ RFID やアクティブ型のマーカー・RFID を同時に読み取ることのできる小型携帯端末が開発される。これは、真のユビキタスネットワーク環境を構築するためには、きわめて重要な要素技術であり、ユビキタス情報社会への影響度はきわめて高い。また、ユーザフレンドリなブラウザが高性能で小型携帯端末に搭載可能になることの影響度のきわめて大きい。

また、本研究の成果は、現時点でも、年間 4400 万台出荷されている携帯電話端末に組み込みことが予想される技術であり、かつ今後あらゆる場面で利用される RFID やアクティブマーカータグとのインタフェースのための機器である。また、こうした機器やメカニズムを利用したユビキタス型の情報配信サービスにも波及効果があり、これらのことから、本研究開発成果の波及性はきわめて高い。

### 2-2 研究開発目標

#### 2-2-1 最終目標（平成 23 年 3 月末）

##### (1) 超小型汎用コミュニケーション端末の実現(全体)

タバコの箱大の超小型汎用コミュニケーション端末を実現する。この端末は、以下の機能を備える。

1. 近距離通信モジュールとして、サブテーマ 1 で開発した超小型マルチプロトコル RFID R/W 装置
2. 広域網通信モジュールとして、IEEE 802.11b または PHS モジュール
3. カラー液晶またはカラーの有機 EL ディスプレイによるグラフィック表示機能
4. タッチパネルによるヒューマンインタフェース機能

5. サブテーマ2で開発したブラウザ
6. 基盤ソフトウェアとして、サブテーマ3で開発したソフトウェアスタック及びチップセット LSI を備える

## (2)超小型マルチプロトコル RFID R/W 装置の実現(サブテーマ1)

1. パラメータをソフトウェアで切り替えることによって、パッシブ型 RFID に関して以下のプロトコルに対応する。
  - (ア) ISO/IEC 15693 (13.56MHz)
  - (イ)  $\mu$ チップ (日立社製) (2.45GHz)
  - (ウ) ISO/IEC 18000-4 (2.45GHz)
  - (エ) eTRON/16 (13.56MHz)
  - (オ) ISO/IEC 18000-6 (900MHz)
2. 小型化・低消費電力化のため、制御機構をシングルチップ LSI で実現する。
3. モジュール全体のサイズをアンテナを含め、50mm×50mm で実現する。
4. 消費電力は、通信時で 350mA@3.3 V 以下とする。
5. 上記の5つのプロトコルについて、自動調整機能を備える。
6. 端末本体と接続する方法として、機器内部に組み込むための有線接続を持った機器と、PAN を用いた無線接続をもった機器を実現する。
7. アクティブ型の RFID に関しては、複数のプロトコルに対応した、耐環境仕様のタグを開発する。また設置容易性を向上させるために、視覚的に電波状況が読み取れるモニター、数値補正用のパラメータ算出の機能を備える。
8. 平成 21 年 1 月までに試作したモジュールの評価結果を用いて、最終的に小型無線受信モジュールを実現する。PDA 型の小型機器に組み込み、アクティブ型 RFID タグとの通信が可能であることを確認する。また、設置容易性を実現するためのクティブ型 RFID タグの電波状況などが読み取れるモニター機能に関しても実装する。

## (3)ブラウザ(サブテーマ2)

1. バイトコード言語によって、描画プリミティブ API を操作できるプラグインインタフェースを備える。
2. 一次元、二次元データを閲覧するためのブラウザプラグインモジュールを備える。
3. サーバー側で、上記のプラグインと同等な機能を実行することが可能であり、ブラウザ処理をサーバー側でもクライアント側でも実行できる手段を提供する。

## (4)チップセット(サブテーマ3)

1. LSI IP を成果とする。また、チップセットをあるいは LSI IP を内包した ASIC を搭載したユーザ端末を実現する。
2. ユビキタスイベントとして、少なくともアクティブ RFID および赤外線マーカをサポートできるものとする。

## (5)ソフトウェアアーキテクチャ(サブテーマ3)

端末をトータルで動作させるための API セットを持ったマネージャー群を実現する。その際、本端末を特徴つける以下の機能は必ず含むものとする。

1. ブラウザインタフェース (プラグイン機能インタフェース)
2. RFID マネージャー
3. アクティブタグマネージャー
4. セキュリティーマネージャー (VPN 機能を持つ)

## 5. コンテキストマネージャー

### 2-2-2 中間目標（平成 21 年 9 月末）

#### (1) 超小型汎用コミュニケーション端末の実現(全体)

電子手帳程度のサイズの超小型汎用コミュニケーション端末を実現し、機能検証を実施する。この端末は、以下の機能を備える。

1. 近距離通信モジュールとして、RFID R/W 装置と無線受信モジュールの双方を備える
2. 広域網通信モジュールとして、IEEE 802.11b または PHS モジュール
3. カラー液晶またはカラーの有機 EL ディスプレイによるグラフィック表示機能
4. タッチパネルによるヒューマンインタフェース機能

#### (2) 超小型マルチプロトコル RFID R/W 装置の実現(サブテーマ1)

1. パッシブ型の RFID タグについては、パラメータをソフトウェアで切り替えることによって、2 つ以上の RFID のプロトコルに対応する。
2. 小型化・低消費電力化のため、制御機構をシングルチップ LSI で実現する。
3. 設置容易性・出力調整機能を備えたアクティブ型 RFID タグの試作機の開発を行う。また、発電素子を一体化したモジュールの試作を行い、設置時の特性や性能評価を行う。
4. アクティブ型の RFID タグの受信装置の小型化を行うための回路検討を行い、試作機の開発を行う。回路の共通化や部品の共通化、アンテナの共通化など可能な部分の検討を進める。

#### (3) ブラウザ(サブテーマ2)

1. バイトコード言語及び描画プリミティブ API の仕様を開発する。
2. 上記仕様に基づいた API とバイトコード言語の仮想マシンの試作を完了し、機能評価を実施する。
3. 一次元、二次元データを閲覧するためのブラウザプラグの仕様開発を終了し、機能評価のための試作を完了する（試作のプラットフォームは機能評価に十分なものの上で行う）。

#### (4) チップセット(サブテーマ3)

1. 機能およびアルゴリズムを開発、検証できる評価ボードを、FPGA および汎用 CPU で実装する。ユビキタスイベントとして、アクティブ型 RFID および赤外線マーカを対象とする。スクリーニング機能では、電波強度の評価は固定閾値としたアルゴリズムを実装する。イベント選択の状況が評価できる試験装置を制作しアルゴリズムの評価が行えるようにする。

#### (5) ソフトウェアアーキテクチャ(サブテーマ3)

端末をトータルで動作させるための API セットに含まれる以下のマネージャーの設計と試作による機能評価を完了する。

1. RFID マネージャー
2. アクティブタグマネージャー
3. セキュリティーマネージャー（VPN 機能を持つ）
4. コンテキストマネージャー

## 2-3 研究開発の年度別計画

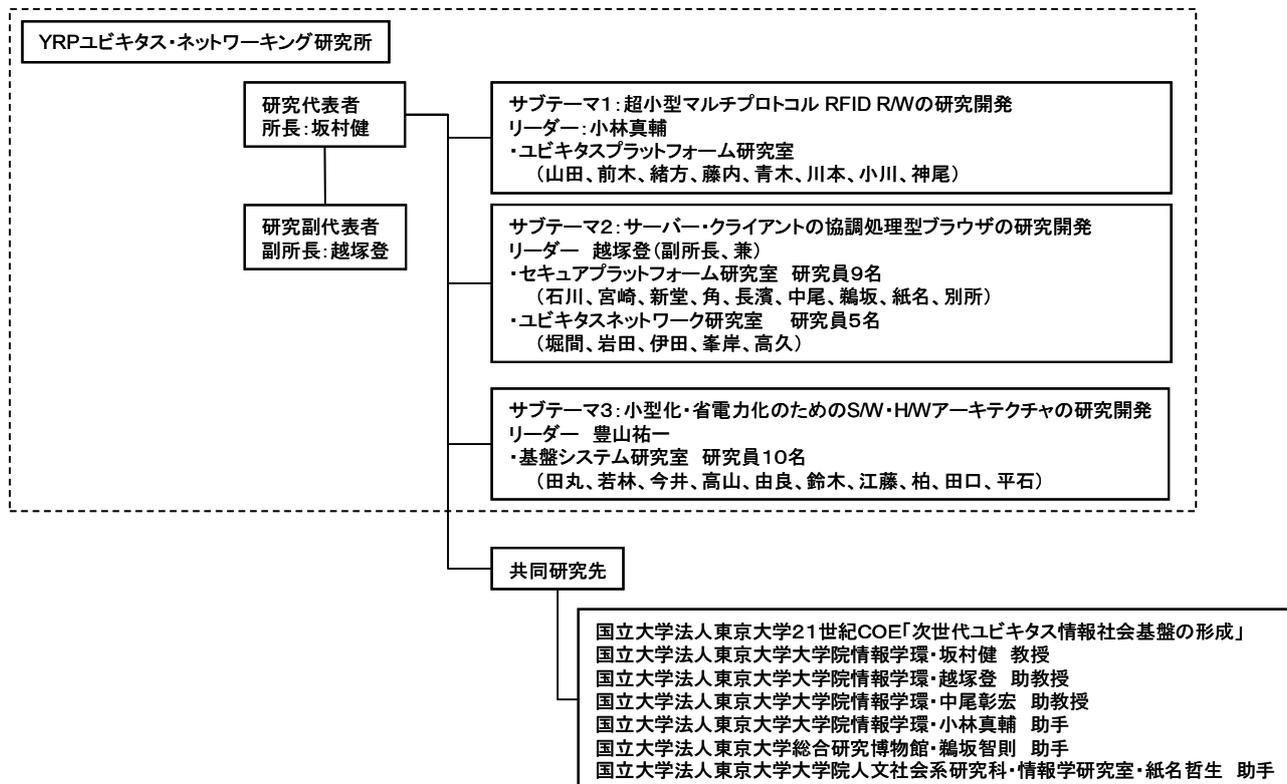
金額は非公表

研究開発項目	18年度	19年度	20年度	21年度	22年度	計	備考
マルチプロトコルRFID R/Wの研究開発(パッシブ) [サブテーマ1]	一次試作	一次試作評価・ 端末組込・ 二次試作	二次試作評価・ 端末組込・ 自動調整ソフトウェア試作	三次試作評価・ 自動調整ソフトウェア改良	端末組込・ 評価実験用量産・ 評価実験		
マルチプロトコルRFID R/W及び自動調整機能つき タグの研究開発(アクティブ) [サブテーマ1]	設計	一次試作	評価・ パラメーター調整 ソフトウェア試作	二次試作	評価・ パラメーター調整 ソフトウェア改良・ 評価実験		
サーバークライアント間協調ブラウザの研究開発 [サブテーマ2]	グラフィック基本 機能開発・ 協調処理言語機構 開発	一次元ブラウザ機 能開発	一次元ブラウザ機 能開発・ 二次元ブラウザ機 能開発	二次元ブラウザ機 能開発	システム全体チェ ーニング・ 評価実験		
超小型端末用チップセットの研究開発 [サブテーマ3]	端末機能抽出・設計	一次試作	評価・端末埋込・二 次試作	二次試作・評価	端末埋込・ 評価実験		
端末アーキテクチャの研究開発 [サブテーマ3]	0次試作 (ハード)・ ソフト設計	0次試作改良 (ハード)・ ソフト試作	一次試作 (ハード)・ ソフト試作計	一次試作改良 二次試作 (ハード)・ ソフト試作	二次試作改良 (ハード)・ ソフト調整・ 評価実験		
間接経費額(税込み)							
合計							

- 注) 1 経費は研究開発項目毎に消費税を含めた額で計上。また、間接経費は直接経費の30%を上限として計上(消費税を含む)。  
 2 備考欄に再委託先機関名を記載  
 3 年度の欄は研究開発期間の当初年度から記載。

### 3 研究開発体制

#### 3-1 研究開発実施体制



## 4 研究開発実施状況

### 4-1 超小型マルチプロトコル RFID R/W 装置の実現

#### 4-1-1 パッシブ型タグ

ユビキタスコミュニケーターのような端末を究極まで小型化するためには、単なる既存技術をチューニングや、実装技術のノウハウの蓄積だけで実現することは不可能である。小型化を実現するうえで問題となるのは、RFID や無線通信モジュールの小型化や、高機能ブラウザを実現する上での機能分散処理などが上げられる。これらの問題を解決するには、小型軽量化を実現するためのハードウェア技術とソフトウェア技術の両面から、従来とは全く異なる方式を導入する、ブレイクスルーが必須である。それは、キーとなるいくつかの要素技術の確立と、及びそれらを組み合わせるためのシステム化するための斬新で新奇な構造（アーキテクチャ）の確立が不可欠である。

まず、RFID のサービスをユビキタス性を持たせるためには、複数の種類の RFID を同時に読むことが可能なマルチプロトコル・R/Wが必要である。サポートする RFID のエアインタフェースの数だけハードウェアを搭載しては、究極の小型化を実現することはできない。そこで、ハードウェア部品は一つに集約し、ソフトウェアによってエアインタフェースを切り替えることができる、ソフトウェア制御型のマルチプロトコルR/Wを開発する。これが「サブテーマ1：超小型マルチプロトコル RFID R/W の研究開発」である。

小型する上で、ハードウェア上で最も困難な点は、非常に多くの種類の物理通信プロトコルをサポートしなければならない点である。その最も代表的な課題が、RFID である。RFID はコストや応用の条件に応じて、様々な通信プロトコルが用いられている。それぞれ合理性があり、単一のプロトコルに無理やり標準化することは非現実的である。実際に、標準化は進められているものの、ISO 18000 で規定されている物流用の RFID でも十種類以上の規格が並存している。

従来は、これらの RF のプロトコルはそれぞれ異なるハードウェアで実装されてきたため、例えば、5種類のプロトコルをサポートするためには、5台の R/W 装置が必要であった。5台を乗せる方針をとっている限りは小型軽量化には限界があり、究極まで小型化軽量化させることはできない。そこで、本研究開発課題では、単一のハードウェア装置でソフトウェアの制御によって、複数のRF周波数、プロトコルで通信のできる、マルチ周波数、マルチプロトコルの RFID R/W 装置を開発する。

パッシブ型の RFID タグについては、パラメタをソフトウェアで切り替えることによって、2つ以上の RFID のプロトコルに対応する。小型化・低消費電力化のため、制御機構をシングルチップ LSI で実現する。パッシブ RFID の R/W から電波を送信し、送信した電波の電力から回路を駆動し、その後、RFID より送信されてきた電波を受信する。図 4-1-1-1、図 4-1-1-2 に本年度設計したデジタル回路、アナログ回路のブロック図をそれぞれ示す。デジタル部は、メイン制御ブロック、送信用ブロック、RF 制御ブロック、受信ブロック、メモリコントローラ、シリアル ROM I/F、バスアービタ、シリアル I/F から構成される。送信ブロックから電波を送信し、受信ブロックにて電波の受信処理をする。図 4-1-1-2 にしめすデジタル部と図 4-1-1-1 にしめす送信ブロック、受信ブロックが接続されて動作をする。アナログ部は、13.56MHz, 2.45GHz, 900MHz 帯の 3 周波数帯に対してそれぞれ独立した回路を持っている。

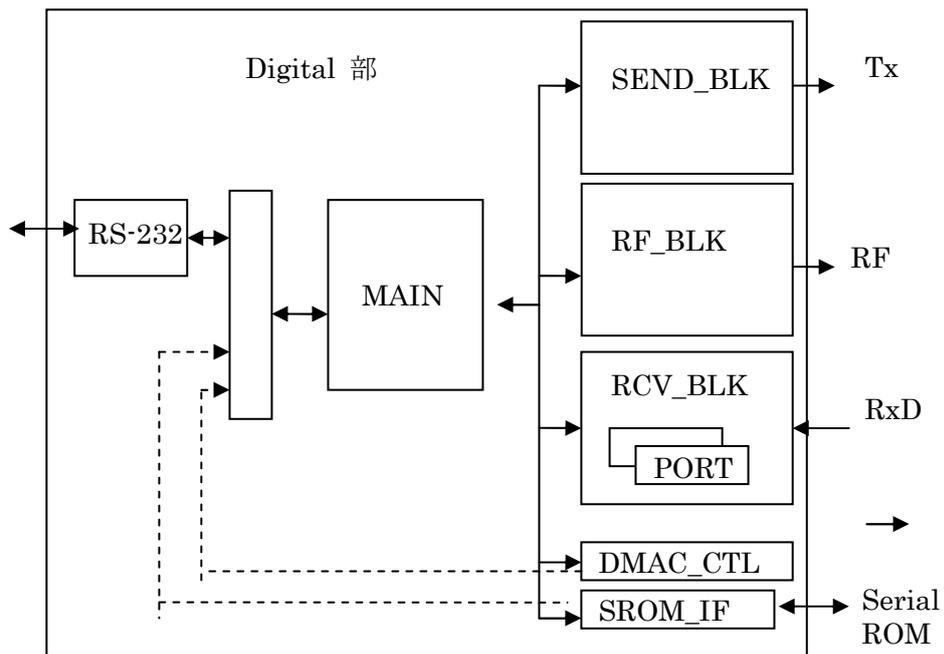
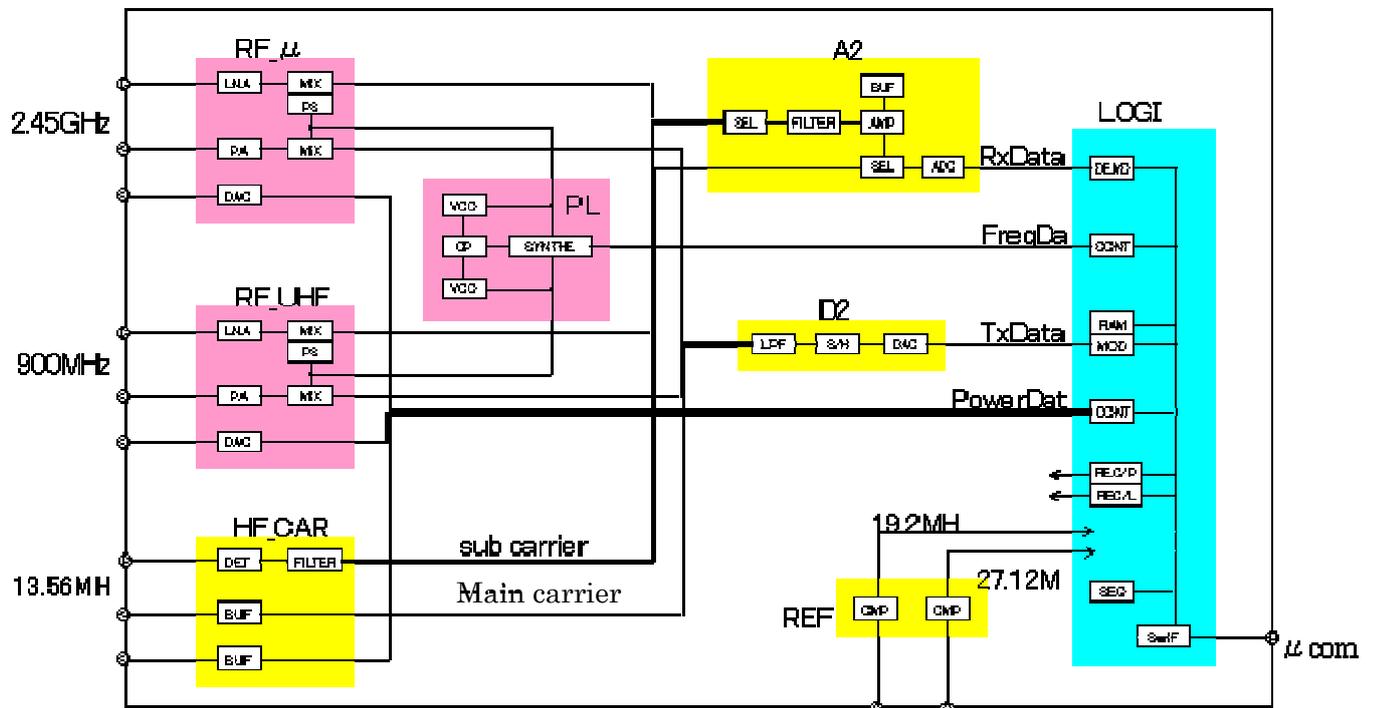


図 4-1-1-1 : デジタル回路のブロック図

RF 帯アナログ回路

HF 帯アナログ回路

ロジック回路



基準クロックモジュール

図 4-1-1-2 : アナログ回路のブロック図

図 4-1-1-1, 図 4-1-1-2 に本年度設計したデジタル回路, アナログ回路のブロック図をそれぞれ示す. 本研究で開発するマルチプロトコル R/W の特徴は, それぞれの処理ブロックにおいてパラメタによって処理内容を変更することでブロックの共通化を図っている点である. 送信ブロック, 受信ブロックにおいて, 符号化・復号化, プリアンブル検出などの共通の処理が必要な部分をパラメタ化している. デジタル側の入力, RS232C を用いてシリアル通信を行う形となっている. これは, さまざまな機器に接続可能な構成を作るうえで, 共通のインタフェースとして利用可能なシリアルインタフェースとしたためである. メインブロックがそれぞれの個別のブロックの制御を行い動作する. また, アナログ回路においては, それぞれの周波数帯ごとに独立して回路を作成している. 周波数帯にあわせたフィルタ回路を含んでいる. 変調などは DAC (Digital to Analog Converter) にて出力時に数値を変化させることによって実現している.

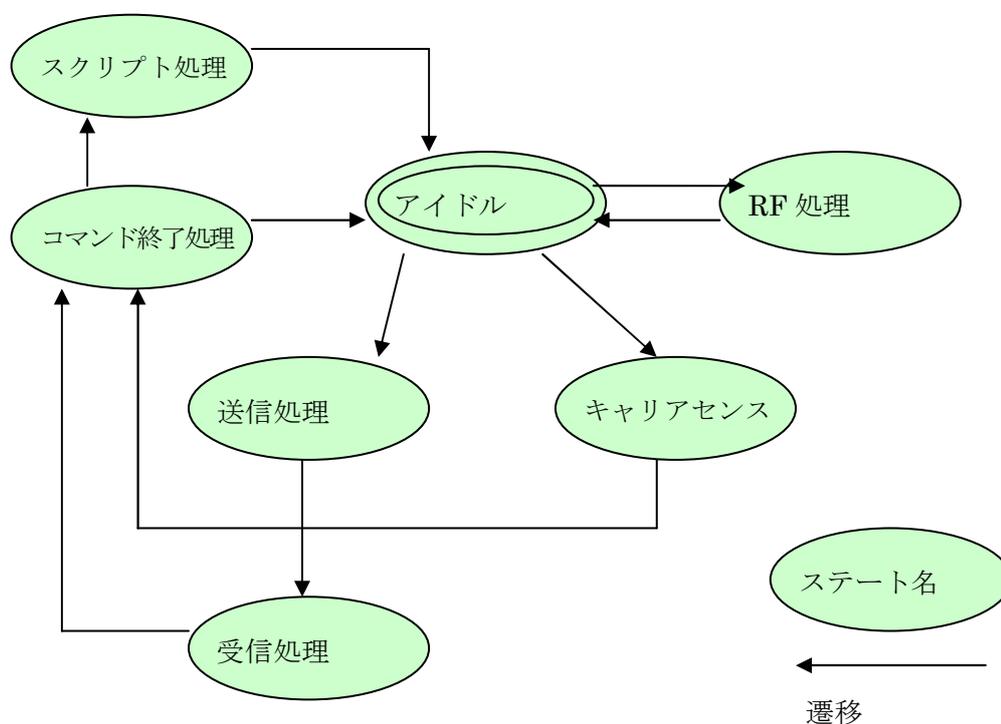


図 4-1-1-3 : メインブロック制御フロー

表 4-1-1-1 : 全体制御フローステート

状態名	状態
アイドル	処理要求待ち
RF処理	RF PLL 設定処理
キャリアセンス	キャリアセンス処理
送信処理	送信制御ブロックでフレーム生成 & フレーム送信
送信処理	受信制御ブロックでフレーム受信
コマンド終了処理	コマンド終了報告 & フレーム送信不可状態
スクリプト処理	DMA実行状態



図 4-1-1-5 は送信ブロックのブロック図である。送信ブロックは通信データバッファに格納されているデータを符号化し、送信を行う。CRC などのエラーチェックを入れる場合には CRC 生成ブロック、パリティ生成ブロックにて処理を行い、その後パターン読み出し制御部内でその送信データの符号化を行う。符号化はパターンバッファに格納されているパターンにあわせて符号化を行う。パターン生成ブロックにて実際に符号化されたデータを生成し、出力する。

### 受信制御ブロック

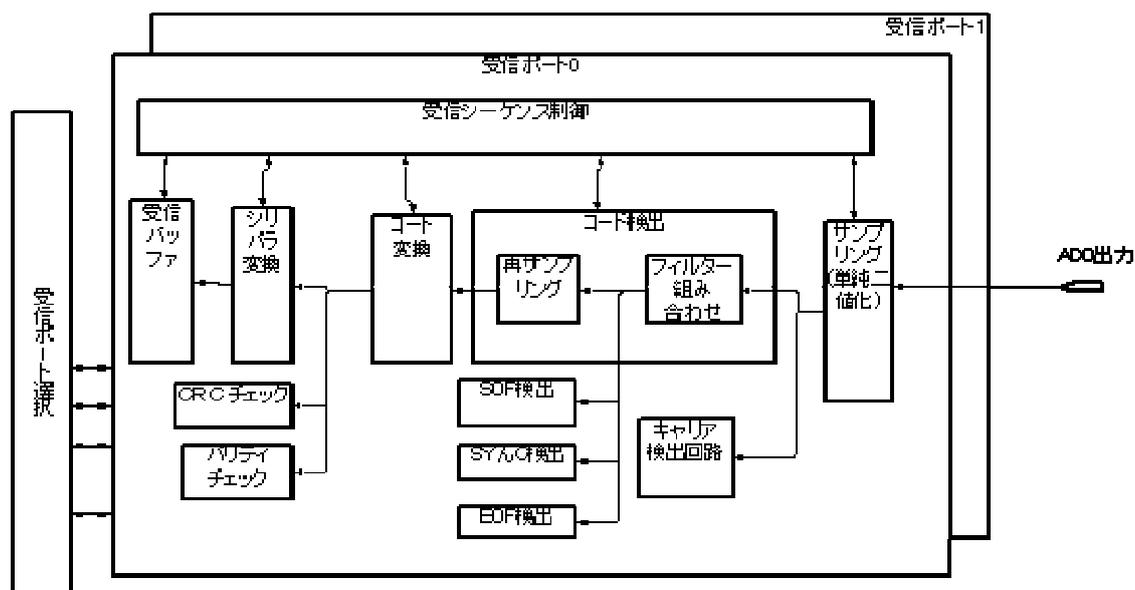


図 4-1-1-6 : 受信制御ブロック

図 4-1-1-6 は受信制御ブロックである。受信時はサンプリングによって波形をデジタル回路にて検出し、コード検出ブロックでノイズ除去用のフィルタを解して、必要なコードを取得する。その後、プリアンプルなどの検出を行う。検出後に続くデータをコード変換ブロックにて処理をし、その後、シリアルからパラレルデータに変換し受信を完了する。

今期は上記のようにマルチプロトコル R/W の詳細設計と回路設計までを行った。来年度にレイアウト、マスク設計を行い、チップの試作を行う。

## 4-1-2 アクティブ型タグ

アクティブ型 RFID は自らが電源を搭載しているため、パッシブ型 RFID と比較して、RFID R/W と数 m から数十 m 離れた距離で通信を行うことができる。この特長を活かして、ユニークな固有 ID を繰り返し発信するアクティブ型 RFID を建物や街路灯など「場所」に設置して、その R/W を搭載したユビキタスコミュニケータを持った利用者が、付近を通過することで固有 ID を取得し、固有 ID に関連する情報を提示する応用を試行している。

注:アクティブ型の場合、通信機の構成で実装されるため、タグと RFID R/W は送受信機能をもった同一のもの、あるいは一方が送信をし、他方が受信をする送信機と受信機といった形になる。以下混乱を起こさないように必要に応じて名称を定義して利用することにする。

場所に設置して、固有 ID を定期的に発信する機能を持つアクティブ型タグを「マーカ」と呼ぶ。

マーカは、通信媒体として電波あるいは光(可視光、赤外線)が用いられる。電波を媒体とするタグは主に広い領域に対してサービスするのに適し、光を媒体とするタグは数 m 程度の狭い領域に限定してサービスするのに適する。

#### 4-1-2-1 電波式アクティブタグ

ユビキタスコンピューティング環境では、「場所」にとりつけられた非常に多数のアクティブ型タグが動作することになり、それらの相互干渉の問題が発生する。この問題は、光を媒体とするマーカでは、光の直進性が強いいため、隣接するマーカとの間の干渉を考えるのみでよく、比較的容易に解決ができる。一方電波を媒体とするマーカでの相互干渉は、電波の到達距離が長く、反射や回り込みなどの現象により、問題が複雑である。

電波を媒体とするマーカは、電波の相互干渉を避けるために、電波を発信する前にキャリアセンスを行い、他の電波がでてきている場合は、少し待ってから、再度試みを行う。待ち時間(再送信時間)の設定と発信する情報(固有 ID など)の伝送時間、伝送間隔の関係により、タイムスロットが形成され、複数のマーカが順次自分のスロット時間で信号を出すようになる。たとえば、1 秒に 1 回信号を出すマーカを 10 スロットで割り当てられるように再送信時間を設定したとする。キャリアを検知できる範囲に 10 個以内のマーカしか存在しない場合は、割り当てられたタイムスロットで、定期的にそれぞれのマーカが発信するように安定する。ところが、10 個以上のマーカが存在すると、スロット数が不足するため衝突が起こり、発信できないマーカがでてきてしまう。この問題を改善するには、以下のような方法が考えられる。

##### (1) 時間的改善

- (a) 伝送速度を上げ 1 スロット時間を短くする
- (b) 発信間隔を長くしてスロット数を増やす

##### (2) 多重化による改善

- (c) 複数周波数チャネルによる多重化
- (d) 変調方式の工夫による多重化

(a)は評価を行ったと特定小電力無線マーカでは、4 値 FSK による 14.4kbps の性能をもちこれ以上の改善は回路が複雑化する。(d)も同様である。(b)は発信間隔が長くなるため、ユビキタスコミュニケータで移動しながら受信することを考えると、情報がとりにくくなるという問題があり、自由に長くすると応用の要求仕様に適合しなくなる。従って(c)の複数周波数チャネルによる多重化により改善することについて検討を行うこととした。

マーカ側は、複数周波数チャネルを利用することとする。ユビキタスコミュニケータがこれを受信することを想定し、機能的にマーカを受信する回路を、「受信モジュール」と呼ぶこととする。受信モジュールは、複数チャネルのキャリアをしらべ、最も強いチャネルを自動的に選択をする自動チューニング機能を実現する。

18 年度では、本自動チューニング機構を特定省電力無線モジュールにソフトウ

フェア的に実現して、評価を行った。

電波マーカは 3 台用意し、以下の 3 チャンネルを割り当て、それぞれ 0.5 秒間隔で、1024 ビットのデータを繰り返し発信するようにした。周波数の異なる 3 台のマーカは、それぞれ同期をしていないので、同時に電波がでていいる時間が存在する。

チャンネル A-1	429. 2500MHz
チャンネル A-2	429. 4125MHz
チャンネル A-3	429. 5750MHz

また、同時に周波数チャンネルの異なる次の二つの電波マーカを用意し、自動チューニング機構に影響がないかを評価することとした。

チャンネル E-2	429. 4625MHz
チャンネル E-3	429. 6250MHz

受信モジュールの自動チューニング機構は、まず、あらかじめ設定されているチャンネル(設定グループと呼ぶ)A-1, A-2, A-3 のキャリアをキャリアセンスする。このモードを「通信チャンネルサーチモード」と呼ぶ。このときキャリア信号のレベルを測定し、十分な強度のチャンネルがみつければ、「固定チャンネル通信モード」に移行する。「固定チャンネル通信モード」では、通信チャンネルサーチモードで定められたチャンネルのマーカのみを受信する。しかし、移動したり周りの環境が変わり、キャリア信号レベルが閾値より低い強度しかない時間が一定時間続くと、「通信チャンネルサーチモード」に遷移する。

図 4-1-2-1 に示すように、マーカは一定周期で信号を発信している。受信モジュールは、通信 ch サーチモードでは、三つのチャンネルを順に切り替え、電波強度を測定する。これで、もし十分な強度である閾値を超え、かつ一番強いチャンネルが A-2 だとすると、A-2 チャンネルを固定的に受信する、固定 ch 通信モードに移行する。A-2 チャンネルのマーカからの信号が安定して受信でき、閾値を下回らない間は、A-2 チャンネルを受信しつづける。もし A-2 チャンネルの信号が閾値より低い状態が一定時間以上つづくと、受信モジュールは、通信 ch サーチモードに移行して、A-1, A-2, A-3 の電波のうち十分強い電波が見つかるまでを繰り返しサーチする。

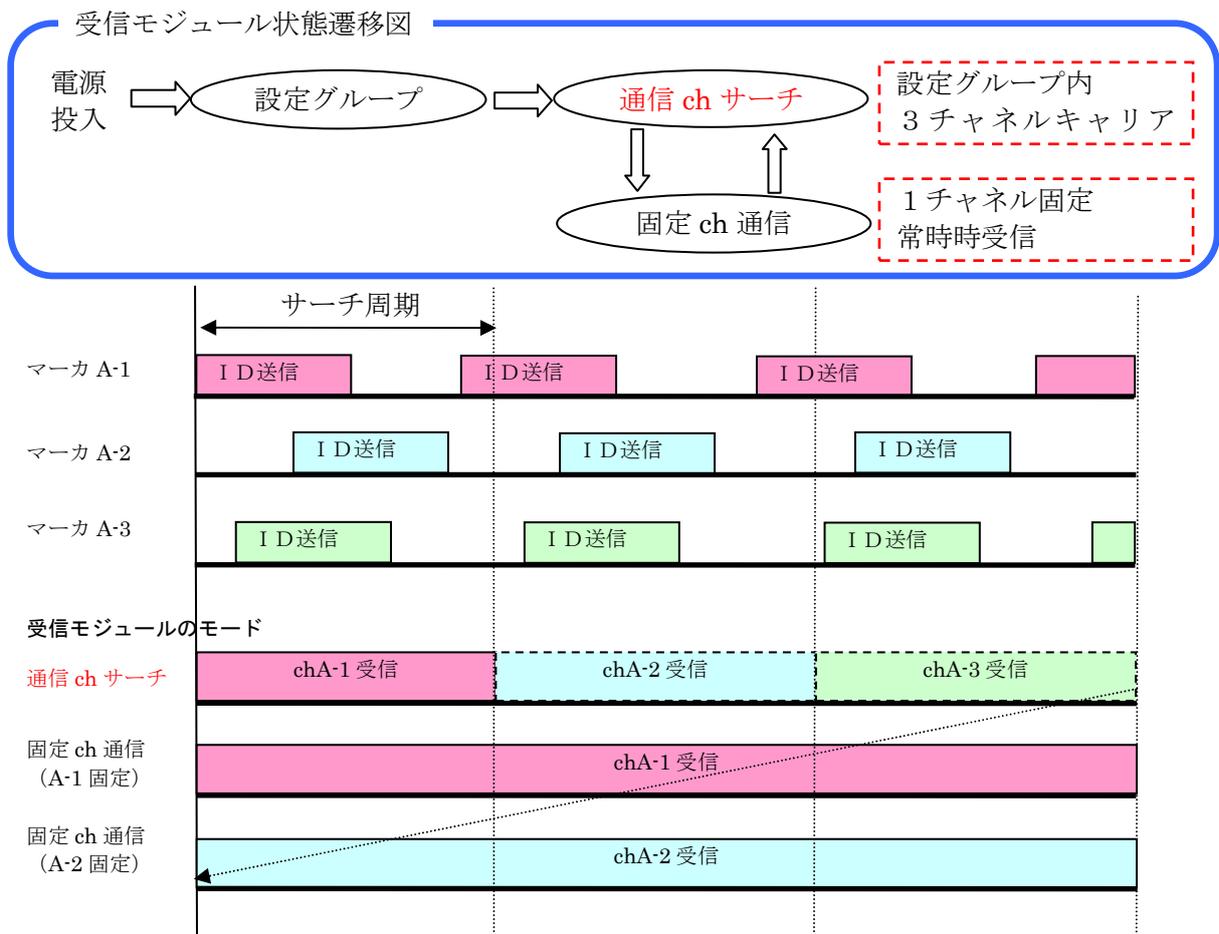


図 4-1-2-1：自動チューニング方式

本方式の評価を行うため、多種の小物品を扱う物流センターにシステムを設置して、実験を行った。対象の物流センターは、70m×50m 天井高さは約 3m である。図 4-1-2-2 はその平面図である。金属の網の張られた棚が林立し、それを取り囲むグレーで塗りつぶしてある領域はベルトコンベアラインである。ここに特定小電力 10mW の電波マーカを 5 台設置した。チャンネルは先に示した A-1, A-2, A-3 と E-2, E-3 である。429MHz 帯の 10mW 電波は、自由空間でおおよそ 60m 程度到達するので、棚による傷害があるとはいえ、各マーカの電波はかなりの領域が重なる。

図中で、A-1 取得所望領域等と記載した領域は、その領域を移動する受信モジュールが A-1 のマーカを受信するのが適当と想定した領域である。

判定を容易にするためにマーカからの信号により、受信モジュールにつけた LED を制御するようにした。マーカからは、1024 ビットのデータを 0.5 秒間隔で送信し、各ビットは受信モジュールの No. と対応づけする。たとえば No. 11 の受信モジュールは、11 ビット目の状態を認識するようにする。もし 1 であれば LED を点灯し、0 であれば消灯する。マーカに制御コンピュータを接続して各ビットの値を制御できるようにしておく。たとえば、No. 11 の受信モジュールのために 0.5 秒間隔でビット 11 を ON/OFF を交互に繰り返すようにすると、受信モジュールの LED は正常に受信していれば 1 秒の周期で点滅することになる。

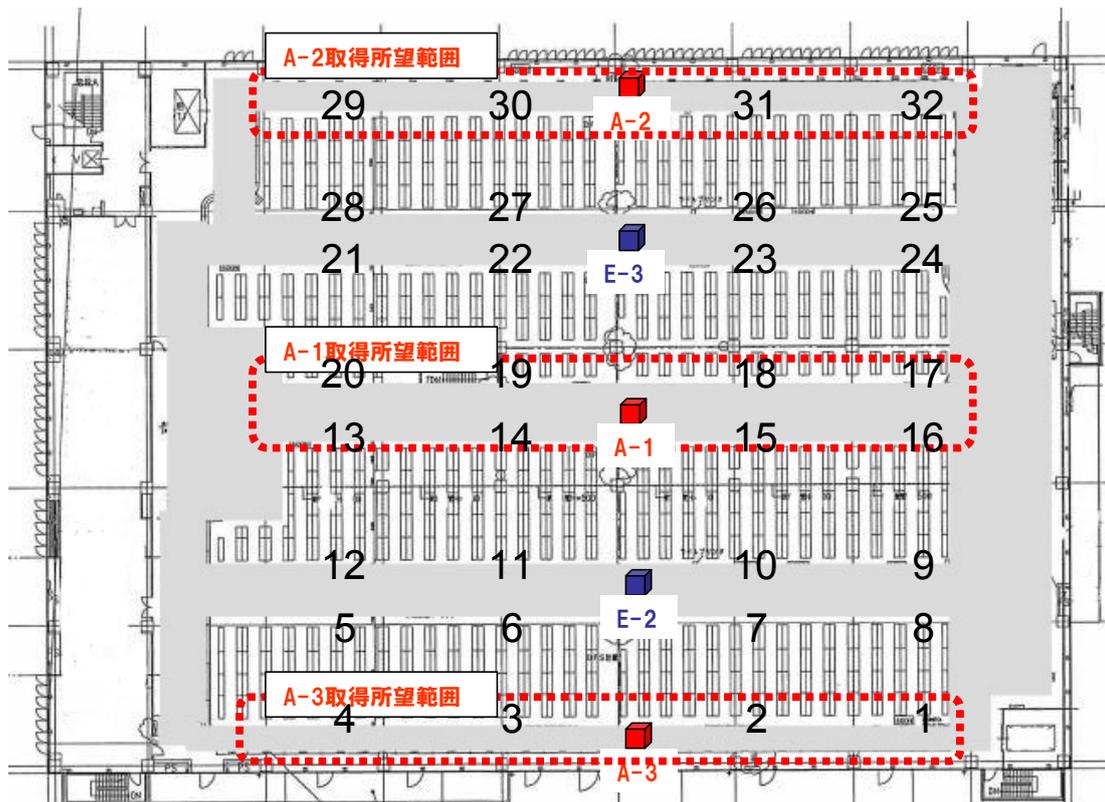


図 4-1-2-2: マーカの設置状況

実験においては、ch サーチ周期を 1 秒とし、閾値以下となる時間が 10 秒を超えると通信 ch サーチモードに移行するように設定した。A-1, A-2, A-3, A-4, A-5 から信号を発信し、複数の受信モジュールを移動させて動作を確認した。受信モジュールは A-1, A-2, A-3 の受信グループに設定してある。移動により、電波強度が弱まると通信 ch サーチモードに移行するが、A-1, A-2, A-3 の取得所望範囲に入り固定 ch 通信モードに移行すると以降、安定して受信する。所望範囲においては、床付近や、棚やコンベアの陰となる領域でも安定して受信できた。

表 4-1-2-1 は、図 4-1-2-2 の各測定点における、各マーカの電波強度の測定結果である。水色で示した部分が、その測定点における所望のマーカである。これに対して、横方向に赤、ピンク、黄色で示した部分は、所望の電波強度に近い強度の他チャンネル電波が届いていることを示す。たとえば測定位置 1 では、A-3 が所望の電波であるが、 $-85.13\text{dBm}$  である。この位置で A-1 の電波は  $-82.51\text{dBm}$  のため、A-1 取得所望領域から移動してきた場合や、周囲のノイズなどで A-1 の強度を高く判定した場合は、A-3 取得所望領域にもかかわらず、A-1 を受信する固定 ch 通信モードとなる場合がある。このようなケースは、複数階の建物にマーカを設置した場合で、階毎にチャンネルの設定を変えておくと、通常はフロアの床天井などが遮蔽効果が高いため同じフロアは固定 ch 通信モードで受信する。ところが、エスカレータや吹き抜け空間のような場合は、上下階のマーカの電波強度の差が少ない場合があり、自動チューニング機構により、別の階のマーカを受信する可能性がある。電波を使う場合、このような現象は避けられないが、このようなケースでは、階を異動する場所に階を指定する赤外線式のマーカを併用する方法等により、解決することができる。

「場所」にアクティブタグを設置する応用において、通信速度やスロット数の制限により、所望設置数に対して、設置可能数の制限があったが、本方式による自

動チューニング機構をもつ電波式アクティブタグによると設置可能なマーカ数を大幅に増やすことが可能となることの確証がえられた。

表 4-1-2-1: 電波強度測定データ

測定位置	A1 429.2500MHz	A2 429.4125MHz	E2 429.4625MHz	A3 429.5750MHz	E3 429.6250MHz
1	-82.51	-99.87	-69.26	-85.13	-99.65
2	-63.8	-88.2	-60.84	-69.32	-84.37
3	-72.43	-96.15	-62.82	-68.61	-87.7
4	-71.39	-99.22	-75.32	-88.87	-92.15
5	-72.36	-92.55	-65.06	-101.79	-88.03
6	-70.02	-88	-49.98	-69.25	-87.69
7	-60.96	-88.26	-51.96	-68.99	-88.16
8	-80.04	-87.66	-69.22	-81.42	-91.77
9	-73.04	-90.24	-66.96	-92.64	-84.9
10	-62.48	-80.2	-58.25	-85.74	-73.46
11	-63.42	-85.28	-44.59	-76.22	-81.62
12	-70.35	-79.96	-59.51	-90.82	-79.39
13	-50.3	-65.9	-65.56	-93.23	-71.05
14	-47.71	-71.43	-52.16	-93.51	-62.7
15	-53.47	-68.99	-65.43	-90.36	-59.98
16	-59.68	-88.87	-74.28	-94.99	-76.54
17	-60.68	-71.24	-73.29	-99.91	-69.58
18	-38.53	-63.1	-55.25	-94.19	-61.58
19	-42.55	-68.65	-67.79	-92.23	-53.85
20	-62.06	-77.66	-86.8	-99.87	-75.07
21	-73.3	-71.63	-91.94	-112.23	-57.43
22	-68	-53.84	-80.93	-102.65	-49.54
23	-56.99	-54.18	-79.43	-98.27	-48.19
24	-70.97	-66.1	-81.86	-113.84	-68.77
25	-82.34	-57.15	-94.8	-109.12	-64.34
26	-66.69	-53.29	-79.1	-110.06	-56.91
27	-63.95	-51.29	-75	-105.44	-42.86
28	-77.62	-61.07	-81.19	-108.69	-54.24
29	-84.78	-53.47	-89.95	-113.59	-65.27
30	-71.62	-51.55	-95.68	-113.39	-56.87
31	-75.93	-54.02	-92.02	-112.62	-52.93
32	-79.02	-62.47	-89.4	-122.19	-67.82

色	意味合い
青	所望電波
赤	5dB 以内の同グループ電波
紫	10dB 以内の同グループ電波
黄	15dB 以内の同グループ電波
白	同グループだが、影響は薄い
灰	別グループで干渉無し

#### 4-1-2-2 光学式アクティブタグ

光学式アクティブタグとして、従来家庭用リモコンと同等な変調方式による光学式アクティブタグを採用していたが、伝送速度が低く、消費電力が高いという問題があった。また、伝送速度が遅いため、R/Wを搭載したユビキタスコミュニケータが移動しながら受信すると受信しにくいという問題があった。このため、伝送速度をあげて消費電力を下げる光学式アクティブタグを試作した。

方式としては赤外線通信に使われる IrDA-SIR の変調方式に基づくもので、ゆ 880nm、115.2kbps ベースバンドで RZI 変調方式をとる。

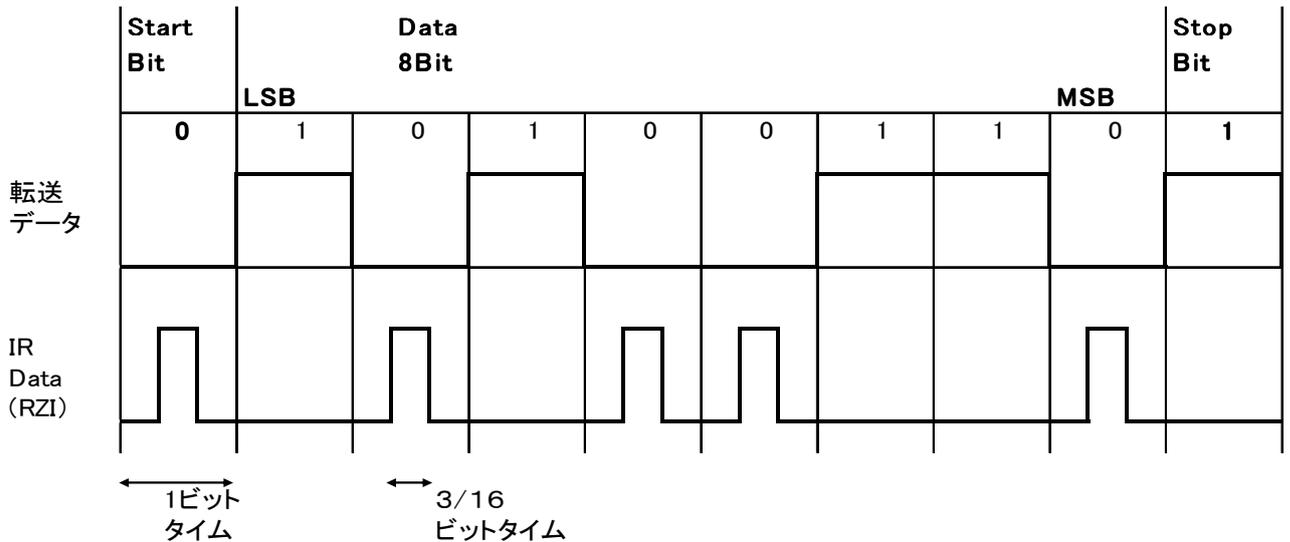


図 4-1-2-3: 光学式アクティブタグの変調方式

赤外線 LED は 9 個搭載し、3 個ずつ角度を変えることにより、おおよそ長辺側 60 度、短辺側 30 度の長方形の領域をカバーするようにした。ユビキタスコミュニケータに組み込む受信モジュール(R/W)では周辺で 4m、中央部で 5m 程度の距離で安定受信できる。このため 4m 以下の天井高での設置に適用可能である。128bit の情報を毎秒 5 回発信可能である。

受信モジュールは、IrDA 用の受信用部品を利用できこれをシリアルデータに変換して処理する。この回路は、4-3-1 で述べるハードウェアプラットフォームに搭載した。

#### 4-1-3 まとめ

パッシブ型 RFID のマルチプロトコル R/W については、第一次試作のための設計までは完了したものの、そのチップ試作には至っていない。第一次試作で開発するチップのアナログ部分のチューニングを行い、第一次試作 LSI の完成度の高さを上げるために、設計時間を費やしたことがチップ試作にいたっていない理由である。第 1 次試作は平成 19 年末には完了し、モジュールとしての評価を行う予定で進めている。今後の研究計画によって本遅れは挽回することは十分可能であると考えている。

アクティブ型 RFID については、複数ある R/W に対して移動型のアクティブ型タグが強度の高い R/W と通信するように周波数・チャンネルをチューニングする機能を

既存アクティブ型タグをソフトウェア的に実現した。このソフトウェア機能設計は次年度のブレッドボード開発のための論理的設計として設計を完了した。また、光学式アクティブ型のタグ及びその R/W の第一次試作を完了した。R/W 側はサブテーマ 3-1 で構築したハードウェアプラットフォーム上で稼動している。

## 4-2 サーバー・クライアントの協調処理型ブラウザの研究開発

### 4-2-1 研究開発内容

本開発は、超小型汎用コミュニケーション端末においてヒューマンフレンドリなユーザインタフェースをコンパクトかつ高性能で実現するために、サーバ・クライアント間で不可分散が実行環境に応じて動的にできるブラウザ(以下、本ブラウザと称する)の研究開発である。

ブラウザが対象とするコンテンツは、一般のインターネット上の WWW サービス(HTML)から、専用の XML 記述、動画などのマルチメディア・データ、など多岐にわたる。また、将来的にも様々なコンテンツに用意に対応可能であるよう、コンテンツの種別毎に、プラグイン・モジュールを用意し、それを動的に変更可能とするプラグイン構造とする。

また、必要なプラグイン・モジュールのみをロードし実行することにより、省資源でコンパクトなブラウザが実現できると考える。

以上をふまえて本ブラウザの基本方針を以下の通りとする。

4. バイトコード言語によって、描画プリミティブ API を操作できるプラグインインタフェースを備える。
5. 一次元、二次元データを閲覧するためのブラウザ・プラグインモジュールを備える。
6. サーバー側で、上記のプラグインと同等な機能を実行することが可能であり、ブラウザ処理をサーバー側でもクライアント側でも実行できる手段を提供する。

プラグイン・モジュールを用意するコンテンツの種別は、まず現在広く一般的に使用されているものより、以下を選択した。

1. HTML プラグイン(WWW)
2. SVG プラグイン(ベクトルグラフィックス)
3. 動画プラグイン(MPEG2)

クライアント側とサーバー側の協調動作をコンテンツ・レベルにて記述できるよう、TCP/IP ネットワークに対応したスクリプト言語を導入する。

コンテンツ・データやスクリプト言語に、今期はまず一般的な仕様を用いることとする。これは現段階において、それらのコンテンツ・データそのものや情報、ツールが豊富に存在するからである。ただし、今後、より超小型汎用コミュニケーション端末に適したコンテンツ・データやスクリプト言語の検討は必要である。

### 4-2-2 ユーザインタフェース

本ブラウザの動作プラットフォームである超小型汎用コミュニケーション端末は、パーソナルコンピュータと比べて、ユーザインタフェースを実現するハードウェアの制約が非常に大きくなる。

ディスプレイは、せいぜい VGA(640×480 ピクセル)または QVGA(320×240 ピクセル)程度であり、一時に表示することのできる情報は少ない。少ない表示領域で効率よく情報を提供できるユーザインタフェースが必要であり、パーソナルコンピュータの OS のようなマル

チウィンドウ・システムは使い勝手は良くないことが予想される。

入力デバイスは、数個のボタン・スイッチとホイール程度であり。フルキーボードの使用を前提とすることはできない。その代わり、ディスプレイはタッチパネルとして使用できる場合がある。ボタン、ホイールとタッチパネルは同時に使用可能なハードウェアがあれば、片方だけのハードウェアもありうる。よって、両方で操作が可能なユーザインタフェースが望まれる。

以上を踏まえて、本ブラウザのユーザインタフェースの基本方針を以下のように定める。

1. 一コンテンツにつき一画面を基本とする。
2. 各コンテンツの画面は簡単な操作で切り替え可能とする。
3. 切り替えの操作は、ボタンやホイール、タッチパネルから操作可能なことを考え、タブ方式とする。

以上の基本方針に基づき、本ブラウザのユーザインタフェースの画面構成は図. 4-2-2-1 に示す。

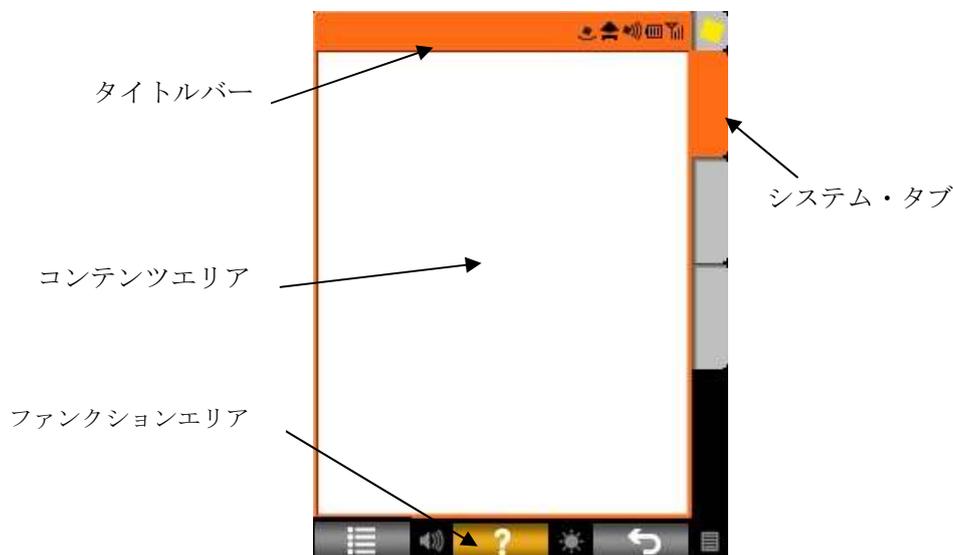


図. 4-2-2-1 ブラウザ画面構成

画面上の各部位の説明を以下に記す。

#### (1) タイトルバー

表示中のコンテンツのタイトルを表示する。またタイトルバーの右端には、システムの状態表示を行なうアイコンを配する。アイコンで示されるシステムの状態には、電池の充電状態、無線 LAN、電波マークなどの電波強度、ネットワークの接続の状態、などが含まれる。

#### (2) コンテンツエリア

コンテンツの内容が実際に表示される領域である。例えばコンテンツ・データが HTML であれば、ここにその HTML の記述に基づいたページが表示される。もし、表示がコンテンツエリアの大きさに収まらなかった場合のスクロール処理、または拡大縮小処理はコンテンツ種別毎に決められる。ただし一つのコンテンツは一画面に収まるのが望ましい。

#### (3) システム・タブ

コンテンツの切り替えを行なうためのタブである。ある状態において、表示可能なコンテ

ンツが全てタブとして表示される。タブを選択することにより、そのタブに対応したコンテンツがコンテンツエリアに表示される。

#### (4) ファンクションエリア

システムとして標準的に提供されるソフトウェア・ボタンを表示する領域である。ファンクション・ボタンはコンテンツの種別や動作状態において変更されることが許される。ただし、同一のコンテンツ、同一の状態においては同じボタン構成となるようにする。ボタンの中で特殊なものとして『戻る』ボタンがある。このボタンを押すと原則として一つ前のユーザ操作をキャンセルし、状態を戻す。この『戻る』ボタンと同一の機能を持つハードウェアボタンが通常は割り当てられる。どちらを使用しても動作は同じである。

ユーザインタフェースの操作は、ボタンやホイールによるキー操作とタッチパネルによる操作に別れ、どちらからも操作可能とする。

キー操作の場合は、操作中は画面上に注目点を表示し、それをボタンやホイールで移動させ、最終的に決定ボタンで動作を実行する。これはタッチパネルでその注目点を直接タップするのと同じ機能とする。

### 4-2-3 ブラウザ・ソフトウェア構成

本ブラウザのソフトウェアは以下から構成される。

#### (1) 各プラグイン・モジュール

コンテンツの種別毎に基本的に一つのプラグイン・モジュールが存在する。複数のプラグインの中で現在表示中のコンテンツに対応するプラグインのみがアクティブとなり、画面上のコンテンツエリアの描画と、それに対するユーザ操作を担当する。

プラグイン・モジュールは独立したプロセスであり、後述のプラグイン・マネージャにより起動され、動作を管理される。

#### (2) プラグイン・マネージャ

プラグイン・モジュールの管理を行なうソフトウェアであり、ブラウザに一つだけ存在する。プラグイン・マネージャは、各種マークや IC タグなどの情報 (イベント) を取得し、それに対応するコンテンツを決定する。そしてそのコンテンツの種別に応じて、対応するプラグイン・モジュールを起動し、表示したいコンテンツの所在を渡す。もしくは起動済みであれば、そのプラグインをアクティブとし、表示したいコンテンツの所在を渡す。

また、後述の UI マネージャと連携し、タブ操作などでコンテンツが切り替わる際に、プラグイン・モジュールの切り替えを行なう。

#### (3) UI (ユーザインタフェース) マネージャ

ブラウザ標準のユーザインタフェース操作を担当するソフトウェアであり、ブラウザに一つだけ存在する。UI マネージャが担当するのは画面上のタイトルバー、システム・タブ、ファンクションエリアに対する各種のユーザ操作である。コンテンツエリアに対するユーザ操作はそのまま対応するプラグイン・モジュールにユーザ操作のイベントを引き渡す。

#### (4) 共通プレイヤー

本ブラウザからは、MP3 や WAVE などの音声プレイヤー、MPEG などの動画プレイヤーなど様々なマルチメディア・プレイヤーが使用される。これらのプレイヤーをプラグイン・モジュールから単一のプレイヤーとして使用可能にするのがこの共通プレイヤーである。

各プラグイン・モジュールは再生したいデータを共通プレイヤーに渡すだけで、プレイヤーの種別を意識することなく、メディアの再生が可能である。また、複数のプラグイン・モジュール間の排他制御も共通プレイヤーにより処理される。

各ソフトウェアは原則として一つのプロセスとして実装され、プロセス間通信により連携する。次頁の図. 4-2-3-1 にブラウザ・ソフトウェアの関連を示す。

ブラウザ・ソフトウェア関連図

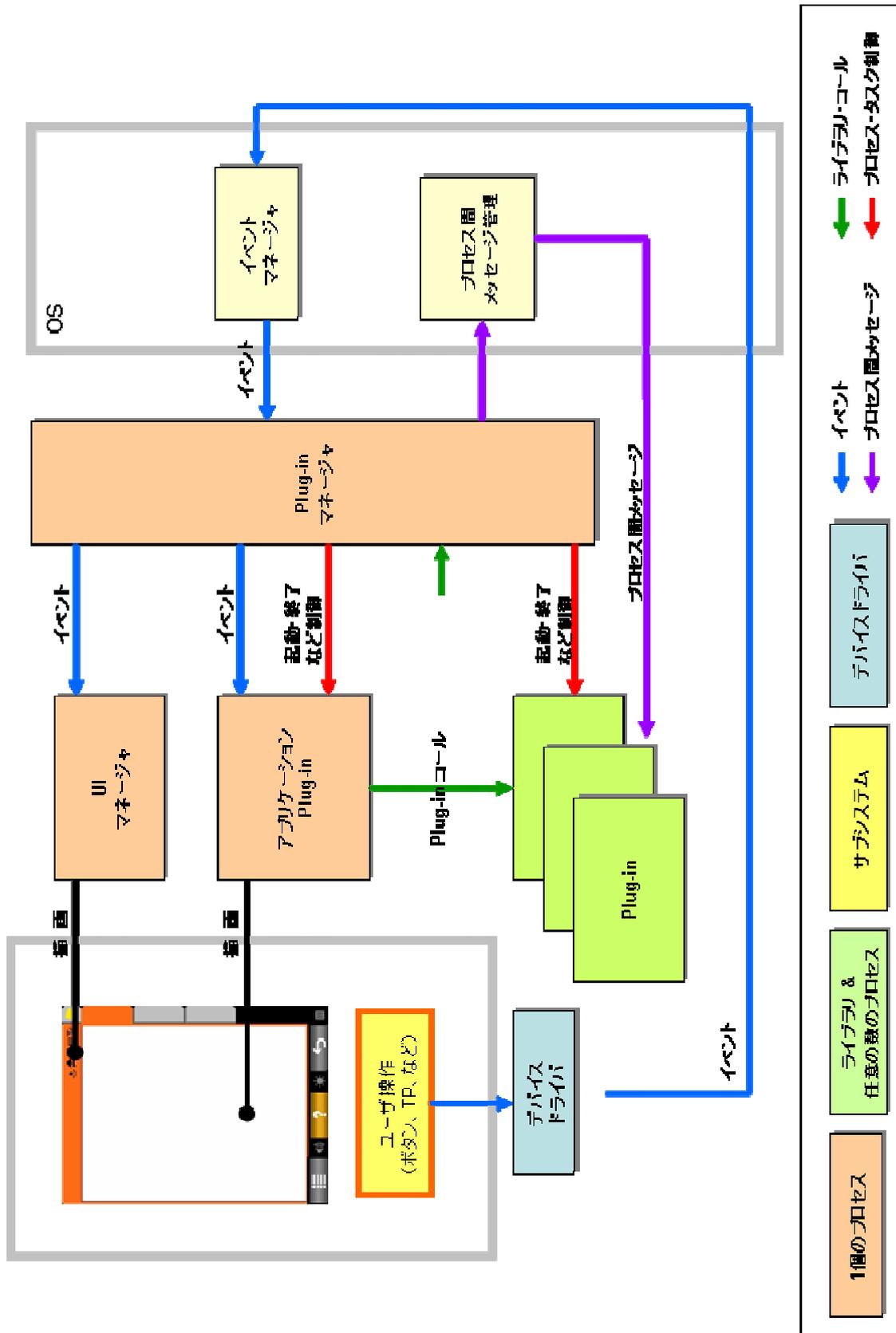


図. 4-2-3-1 ブラウザ・ソフトウェア関連図

#### 4-2-4 プラグイン・マネージャ

プラグインマネージャは、プラグイン・モジュールの管理を行なうソフトウェアであり、ブラウザに一つだけ存在する。システムからブラウザへのイベントは、まずプラグイン・マネージャが取得する。イベントが各種マーカやタグからの通知(ucode)であった場合、プラグイン・マネージャは ucode マネージャに問い合わせを行ない、そのイベントに応じたコンテンツ・データの所在を取得する。そしてそのコンテンツ・データの種別に応じたプラグインの起動、切り替えを行なう。

取得したイベントがプラグイン・マネージャと関係の無いものであった場合（ユーザの操作関係のイベントなど）は、UI マネージャにイベントを引き渡し、以降そのイベントについては関知しない。

プラグイン・マネージャと各プラグインは、どれもプロセスとして実装される。各プラグインはプラグイン・マネージャから起動されるので、各プラグインはプラグイン・マネージャの子プロセスであり、プラグイン・マネージャは全てのプラグインの親プロセスとなる。よって、子プロセスであるプラグインの終了などのイベントはOSの機能により、プラグイン・マネージャに通達され、管理を行なうことができる。

図. 4-2-4-1 にプラグイン・マネージャと他のソフトウェアの関連を示す。

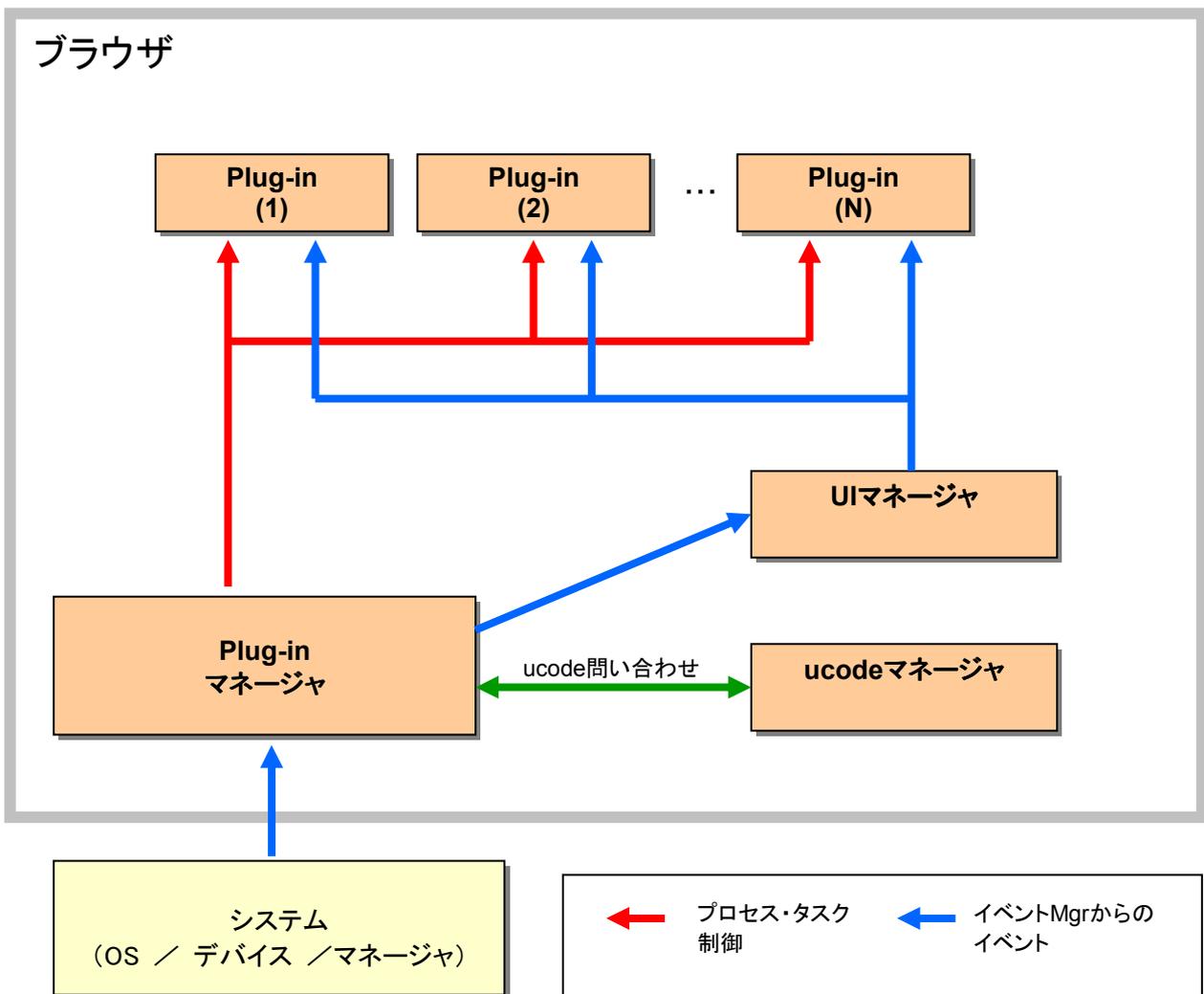


図. 4-2-4-1 プラグイン・マネージャと他のソフトの関連図

## 4-2-5 UI マネージャ

UI マネージャは、ブラウザ標準のユーザインタフェース操作を担当するソフトウェアでありブラウザに一つだけ存在する。

ユーザインタフェースは4-2-2項で述べた GUI ベースのものである。超小型汎用コミュニケーション端末のような限られたハードウェアの資源において、使い勝手の良いグラフィックベースのユーザインタフェースを提供するには、ハードウェアの機能を活用する必要がある。

超小型汎用コミュニケーション端末のグラフィック機能では、4枚の表示レイヤを重ね合わせて一つのイメージとして表示することができる。レイヤ毎に画像のフォーマットが規定でき、また透明色の設定も可能である。また、最上位の表示レイヤのみ特別にハードウェア・ウィンドウの機能を持つ。

そこで、本ブラウザではハードウェアの各表示レイヤにそれぞれの役割を表.4-2-5-1に示すように決定した。

表.4-2-5-1：表示レイヤの種類

優先度	種別	用途
1	アプリケーション・レイヤ	プラグインの表示
2	マネージャ・レイヤ	動画、バルーンなど
3	システム・レイヤ	タブ、タイトルバーなど
4	ダイアログ・レイヤ	ダイアログ、リスト

システム・レイヤは、タブやタイトルバー、ファンクションエリアなどの描画を行なうレイヤであり、通常は画面の最上位に位置する。UI マネージャがその描画と制御を担当する。

アプリケーション・レイヤは、各プラグインが描画を担当するレイヤであり、画面の最下位に位置する。このレイヤについては、UI マネージャは関知しない。

マネージャ・レイヤは、動画などの特別な表示を行なうレイヤであり、システム・レイヤとアプリケーション・レイヤの中間に位置する。それぞれ特定のシステムのマネージャが担当する。

ダイアログ・レイヤは特別な場合に、UI マネージャがダイアログを表示するレイヤであり、システム・レイヤの上位に位置する。ダイアログは、ハードウェアにより実現されるウィンドウの機能を使用して実現する。

各レイヤの位置は固定であり変更されることはない。アプリケーション・レイヤは最下位のレイヤとして常に表示される。システム・レイヤもプラグインがフル画面表示を要求したなどの特別な場合をのぞいて常に表示される。

マネージャ・レイヤとダイアログ・レイヤは必要に応じて表示される。

各レイヤの表示の関係について図.4-2-5-1に示す。

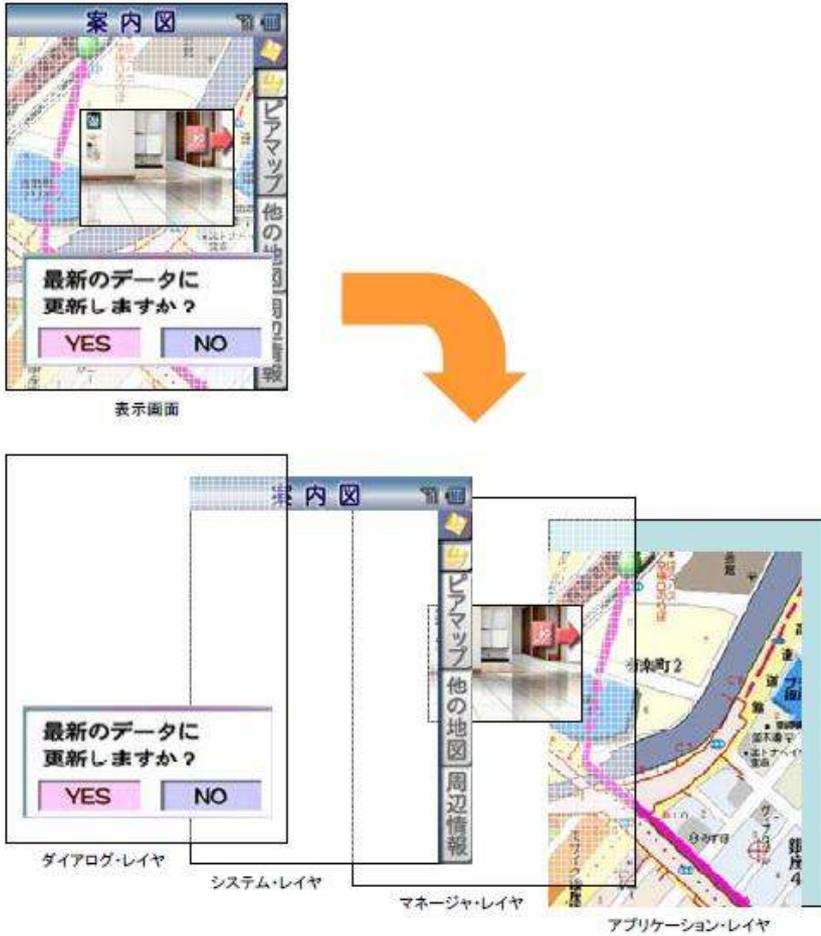


図. 4-2-5-1 画面の表示レイヤの関係

UI マネージャは、ユーザによる各種操作（ボタンの押下、タッチパネルのタップなど）のイベントを、プラグイン・マネージャ経由で取得する。その操作の対象が、システム・レイヤ、またはダイアログ・レイヤの表示に対してのものであった場合、その操作に対応した処理を実行する。それ以外のレイヤに対するイベントは、その時点で動作中のプラグインに受け渡される。

例えば、画面上をタップした場合、UI マネージャはそのイベントを取得し、それがタブの上であればタブの切り替え動作を行なう。それに伴ってプラグインの切り替え、コンテンツ表示の指示はプラグイン・マネージャに対して行なわれる。ファンクションエリア上のボタンであれば、そのボタンに割り当てられて処理を実効する。タブやタイトルバー、ファンクションエリア以外の場所をタップされた場合はそのイベントはプラグインへ渡されUI マネージャは以降は関知しない。

#### 4-2-6 HTML プラグイン

HTML プラグインは、インターネットの WWW サービスで使用されるデータを解釈・表示を行なうプラグインである。つまり、パーソナルコンピュータなどで使用される Web ブラウザとほぼ同等の機能を持つ。

超小型汎用コミュニケーション端末では画面の表示サイズが限られているので、大きな Web ページはスクロールをさせる事となる。ただし、スクロールバーはさらに表示面積を犠牲となり、また操作性も期待できないので採用しないこととした。スクロールは縦方向のみホイールの操作で可能とし、横方向は画面サイズに幅を固定とする。

図. 4-2-6-1 に、HTML プラグインによる表示例を示す。



図. 4-2-6-1 HTML プラグインによる表示例

HTML プラグイン Web ブラウザとしてのスペックを表. 4-2-6-1 に示す。

表. 4-2-6-1 HTML プラグインの Web ブラウザ仕様

通信プロトコル	HTTP1.0/1.1 FTP
サポート言語	HTML4.01 XHTML1.0 CompactHTML (i-mode HTML5.0) XHTML Basic CSS1/2 JavaScript1.5 DOM CORE DOM HTML DOM CSS
画像データ	GIF/Interlaced GIF/Animated GIF JPEG/Progressive JPEG PNG (Version1.0) XBM (X11Bitmap) BMP
文字コード	Shift JIS ISO-2022-JP (JIS) 日本語 EUC Unicode (UTF-8、UTF-16) ISO-8859-1 (Latin1)

HTML プラグインは通常の Web ブラウザの機能以外に、超小型汎用コミュニケーション端末のための機能拡張がされている。これは主に音声や動画などをシステム標準のプレーヤの機能を利用して効率よく再生するための拡張である。

#### 4-2-6-1 音声データ (MP3/WAV/AAC) の再生

音声ファイルは<OBJECT>タグにより指定し、共通プレーヤー (MPM&音声再生ミドル) を使って再生する。

- ・ 扱い対象メディア

MP3 データ : MIME タイプ = "audio/mp3", 拡張子 = ".mp3"

WAV データ : MIME タイプ = "audio/wav", 拡張子 = ".wav"

AAC データ : MIME タイプ = "audio/m4a", 拡張子 = ".m4a"

- ・ コンテンツ内タグの記述

1. HTML プラグインデータをメモリにロードし、共通プレーヤーにロードしたファイルパスを渡し再生する場合

```
<OBJECT
data="プラグインで扱うデータの URL" ← 絶対パス or 相対パス
type="プラグインで扱うデータの MIME タイプ MP3 : audio/mp3 WAV : audio/wav AAC :
audio/m4a">
<PARAM name=atr value="再生属性">
</OBJECT>
```

HTML プラグインでデータをロードするので指定する URL は絶対パス/相対パスともに可能。ただしローカルデータを絶対パスで指定する場合、file:///1:~ という形で指定する。このロードされたデータをプラグイン内でローカルデータの任意の場所に保存し、その URL を共通プレーヤーに渡す。

HTML プラグイン上での画面表示は行わないので、<OBJECT>タグに属性 width と height は指定しない。

この記述方式で再生できるメディアデータサイズは 1.2MByte までとする。

PARAM タグの atr で指定する再生属性は以下のとおりとする

loop : 最後まで再生し終えた後、自動で再生位置が先頭に戻り、再び再生される (ループする)

once : 最後まで再生し終えた後、ポーズではなく、自動で完全停止される

これらが無指定の場合は属性指定なしの再生となり、最後まで再生し終えた後、一時停止される

(例)

```
<OBJECT data="http://192.168.0.1/sample.mp3" type="audio/mp3">
<PARAM name=atr value="once">
</OBJECT>
```

2. HTML プラグインから共通プレイヤーに URL を渡し、共通プレイヤーでデータをメモリにロードし再生する場合

```
<OBJECT
type="プラグインで扱うデータの MIME タイプ MP3 : audio/mp3 WAV : audio/wav AAC :
audio/m4a">
<PARAM name=url value="プラグインで扱うデータの URL"> ← 絶対パス
<PARAM name=atr value="再生属性">
</OBJECT>
```

共通プレイヤーでデータをロードするので、指定する URL は絶対パスのみとする。  
また現状共通プレイヤーではネットワーク経由のデータ取得には対応しないため、  
指定可能 URL は"file://~"のみとする (※file:///1:~の形式は指定不可)  
HTML プラグイン上での画面表示は行わないので、<OBJECT>タグに属性 width と height は  
指定しない。

(例)

```
<OBJECT type="audio/mp3">
<PARAM name=url value="file:///usr/sample.mp3">
<PARAM name=atr value="once">
</OBJECT>
```

#### 4-2-6-2 動画データ (ASF/MP4) の再生

動画ファイルは<OBJECT>タグにより指定し共通プレイヤー (MPM&動画再生ミドル) を使って再生する。

- ・ 扱い対象メディア  
ASF データ : MIME タイプ="movie/asf", 拡張子=".asf"  
MP4 データ : MIME タイプ="movie/mp4", 拡張子=".mp4"
- ・ コンテンツ内タグの記述

1. HTML プラグインでデータをメモリにロードし、共通プレイヤーにロードしたファイルパスを渡し、再生する場合

```
<OBJECT
data="プラグインで扱うデータの URL" ← 絶対パス or 相対パス
type="プラグインで扱うデータの MIME タイプ MP4 : movie/mp4 ASF : movie/asf">
<PARAM name=x value="表示開始位置 X 座標">
<PARAM name=y value="表示開始位置 Y 座標">
<PARAM name=atr value="再生属性">
</OBJECT>
```

HTML プラグインでデータをロードするので指定する URL は絶対パス/相対パスともに可能  
ただしローカルデータを絶対パスで指定する場合、file:///1:/~ という形で指定する。  
HTML プラグイン上での画面表示は行わないので、<OBJECT>タグに属性 width と height は  
指定しない。

この記述方式で再生できるメディアデータサイズは 1.2MByte までとする。  
PARAM タグの atr で指定する再生属性は以下のとおりとする

loop : 最後まで再生し終えた後、自動で再生位置が先頭に戻り、再び再生される(ループする)

once : 最後まで再生し終えた後、ポーズではなく、自動で完全停止される  
これらが無指定の場合は属性指定なしの再生となり、最後まで再生し終えた後、一時停止される

(例)

```
<OBJECT data="http://192.168.0.1/sample.asf" type="movie/asf">
<PARAM name=x value="0">
<PARAM name=y value="0">
<PARAM name=atr value="once">
</OBJECT>
```

2. HTML プラグインから共通プレイヤーに URL を渡し、共通プレイヤーでデータをメモリにロードし再生する場合

```
<OBJECT
type="プラグインで扱うデータの MIME タイプ MP4 : movie/mp4 ASF : movie/asf">
<PARAM name=url="プラグインで扱うデータの URL"> ← 絶対パス
<PARAM name=x value="表示開始位置 X 座標">
<PARAM name=y value="表示開始位置 Y 座標">
<PARAM name=atr value="再生属性">
</OBJECT>
```

共通プレイヤーでデータをロードするので、指定する URL は絶対パスのみとする。  
また現状共通プレイヤーではネットワーク経由のデータ取得には対応しないため、  
指定可能 URL は"file://~"のみとする。(※file://1:~の形式は指定不可)  
HTML プラグイン上での画面表示は行わないので、<OBJECT>タグに属性 width と height は  
指定しない。

(例)

```
<OBJECT type="movie/asf">
<PARAM name=url value="file:///usr/sample.asf">
<PARAM name=x value="0">
<PARAM name=y value="0">
<PARAM name=atr value="once">
</OBJECT>
```

次頁以降に実際の HTML の記述例を示す。

## ■HTML の記述例 (1)

```

HTML4.01 の DOCTYPE 宣言
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
http://www.w3.org/TR/html4/loose.dtd >
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=Shift_JIS" />
<title></title>
<!-- スタイルシート ----->
<style type="text/css">
* {
margin: 0;
padding: 0;
}
img {
border: 0;
}
body {
background-color: #f9f7f4;
}
/* レイアウト */
#container {
width: 216px;
height: 272px;
background-image: url(image/wa081_building_history.jpg);
background-repeat: no-repeat;
}
#text {
width: 210px;
position: absolute;
top: 133px;
left: 8px;
font-size: 14px;
}
#botan1 {
width: 81px;
height: 23px;
position: absolute;
top: 244px;
left: 45px;
}
#botan2 {
width: 81px;
height: 23px;
position: absolute;
top: 244px;
left: 129px;
}
</style>
<!-- スタイルシートここまで ----->
</head>
<!-- ここからコンテンツ内容 ----->
<body>
<div id="container">
<div id="text">
銀座のシンボルである現在の時計塔のある建物は、渡辺仁監督の下建設され、1932年6月10日に落成しました。1988年には都のランドマークに指定され、銀座の顔として、時を刻み続けています。
</div>
<div id="botan1">
<a href="wa082_tokyo_olympics.html">

</a>
</div>
<div id="botan2">
<a href="wa084_second_face.html">

</a>
</div>
</div>
<!-- 音声ファイル -->
<object type="audio/mp3">
<param name=url
value="file:///usr/contents/jpn/wako/mp3/wa081_building_history.mp3">
<param name=atr value="once">
</object>
<!-- 音声ファイルここまで -->
</body>
<!-- コンテンツここまで ----->
</html>

```

※自由度の高いHTML4.01を作成する場合に文書の先頭に記述するDTD(文書定義)

背景色の設定

※テキストのアンチエイリアスを有効にする

コンテンツ領域の設定(固定)

背景画像の設定

テキスト領域の設定

テキストの文字サイズの設定

リンクボタンのサイズと位置の設定

テキスト入力箇所

リンク先の設定

リンク先の設定

ボタン画像の設定

ボタン画像の設定

音声ファイルの設定

## ■HTML の記述例(2)

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=Shift_JIS" />
<title></title>
<!-- スタイルシート ----->
<style type="text/css">
* {
    margin: 0;
    padding: 0;
}
img {
    border: 0;
}
body {
    background-color: #ebeece;
}
/* レイアウト */
#container {
    width: 216px;
    height: 272px;
    background-image: url(image/colum_bg_1.jpg);
    background-repeat: no-repeat;
}
#photo {
    width: 216px;
    height: 129px;
    position: absolute;
    top: 0px;
    left: 0px;
    margin-top: 12px;
    text-align: center;
}
#text {
    width: 204px;
    position: absolute;
    top: 134px;
    left: 6px;
    font-size: 14px;
}
</style>
<!-- スタイルシートここまで ----->
</head>
<!-- ここからコンテンツ内容 ----->
<body>
<div id="container">
<div id="photo">

</div>
<div id="text">
日本橋～新橋間を走っていた鉄道馬車。明治15年6月開業。
</div>
<!-- 音声ファイル -->
<object type="audio/mp3">
<param name=url
value="file:///usr/contents/jpn/colum_up/mp3/co0303_basha.mp3">
<param name=atr value="once">
</object>
<!-- 音声ファイルここまで -->
</body>
<!-- コンテンツここまで ----->
</html>

```

HTML4.01 の DOCTYPE 宣言

背景色の設定

※テキストのアンチエイリアスを

コンテンツ領域の設定(固定)

背景画像の設定

写真領域の設定

写真の配置位置の設定

※ 横位置は中央配置で設定し、縦位置は写真の縦幅によって位置調整する。

テキスト領域の設定

テキストの文字サイズの設定

写真の設定

テキスト入力箇所

音声ファイルの設定

## 4-2-7 SVG プラグイン

SVG プラグインは SVG のデータを表示するプラグインである。

SVG (Scalable Vector Graphics) は、XML によって記述されたベクターグラフィック言語、および SVG で記述された画像フォーマットのことであり、W3C で標準として勧告されている。

SVG は、拡張の自由度が高い XML (Extensible Markup Language) で記述されており、XML ならではの各種機能を定義した要素を持つ。SVG ではそれ自身に回転・拡大・移動などの表現を定義しているため、単体で多様な表現をする事が可能である。従来のウェブサイトでは、いわゆるインタラクティブな双方性のある画面変化を伴う表示を JavaScript や FLASH を用いてきた。HTML/XHTML に SVG を組み合わせることにより、JavaScript や FLASH を導入せずとも同様の効果が発揮される事が期待できる。

また、XML なので、専用のアプリケーションを用いることなくテキストファイルで作製・編集できる。

超小型汎用コミュニケーション端末では、以上の SVG の特徴を踏まえて、地図表示などに SVG プラグインを使用した。

地図は、ビットマップデータで表示した場合、拡大/縮小、回転などの操作を行なうことが困難である。美しい表示を行なうためには、複数のサイズや方向異なったデータをもたなければならない、データ作成の観点からも、実行時のリソースの観点からも問題が大きかった。

SVG を用いることにより、一つの地図データを作成すれば、様々なサイズや方向の地図が表示可能となった。

図. 4-2-7-1 に SVG プラグインの画面表示の例を示す。



図. 4-2-7-1 SVG プラグインの画面表示例

#### 4-2-8 動画プラグイン

動画プラグインは、動画再生を専門に行なうプラグインである。動画は MPEG4 (ASF 形式) のものが再生可能である。

同様の動画再生は、HTML プラグインでも HTML の記述中に動画を入れることにより可能である。ただし、動画プラグインは超小型汎用コミュニケーション端末の画面全体を使ったフル画面再生が可能である。フル画面再生の際には、タイトルバーやタブ、ファンクションエリアも消すことができる。

動画プラグインのコンテンツは XML にて記述される。

以下に端末のディスク上の動画ファイル /usr/contents/mov.asf を再生する XML 記述の例を示す。

```
<?xml version="1.0" encoding="Shift_JIS" ?>

<ubicontents>
<contents id="" description="interview 01" attribute="" functionarea="false"
functionurl="" tabarea="false" encoding="Shift_JIS">

  <movie url="file:///usr/contents/mov.asf" autostart="true"
    pos_x="0" pos_y="-24" width="240" height="320" attribute="ui_hide"></movie>
  <component eventtype="key" action="none" pos_x="0" pos_y="0" width="0"
    height="0" attribute="" keycode="up"></component>
  <component eventtype="key" action="none" pos_x="0" pos_y="0" width="0"
    height="0" attribute="" keycode="down"></component>
  <component eventtype="key" action="none" pos_x="0" pos_y="0" width="0"
    height="0" attribute="" keycode="center"></component>
  <component eventtype="mediaend" action="senducode" url="" pos_x="0"
    pos_y="0" width="0" height="0"
    attribute="00001C000000000000002000000022000" keycode=""></component>

</contents>
</ubicontents>
```

#### 4-2-9 スクリプト言語

クライアント側とサーバー側の協調動作をコンテンツ・レベルにて記述できるよう、TCP/IP ネットワークに対応したスクリプト言語を導入することとした。

スクリプト言語は、既存のサーバ側のソフトウェア資産が活用できるようインターネットなどで広く普及している JavaScript (ECMAScript) を今回は使用し、その評価を行う事とした。

JavaScript は、XMLHttpRequest をサポートすることによりサーバとの非同期通信を可能とした。これにより、いわゆる AJAX と呼ばれる技術が使用可能となる。

また、スクリプト言語から超小型汎用コミュニケーション端末のシステムの機能呼び出せるよう拡張を行ない、システムが持つ環境変数を JavaScript から変更可能とした。

以下にそのサンプルを示す。これは HTML に埋め込んだ JavaScript からカテゴリ code、名称 next の環境変数にアクセスするものである。

```
<HTML>
<HEAD>
<TITLE>
環境変数アクセスサンプル
</TITLE>
<SCRIPT>
function GetNextUcode () {
    ret = document.env.getenv( "/usr/etc/env.ini", "code", "next" );
    confirm( ret ) ;
}
function SetNextUcode () {
    ret = document.env.setenv( "/usr/etc/env.ini", "code", "next", "0", "UBP" );
    confirm( ret ) ;
}
</SCRIPT>
</HEAD>
<BODY>
<OBJECT classid="ml_HBU_PluginEnv" name="env">
</OBJECT>
<FORM>
<table align=center>
<tr><td><INPUT type=button value="次は?" onClick="GetNextUcode();"></td></tr>
<tr><td><INPUT type=button value="次へ" onClick="SetNextUcode();"></td></tr>
</table>
</FORM>
</BODY>
</HTML>
```

#### 4-2-10 まとめ

平成18年度の開発計画は、サーバ・クライアントの協調処理型ブラウザの実

現に向けて、平成 18 年度はその基盤となるモジュールを開発することであった。すでに、一次元、二次元データを閲覧するブラウザプラグインの仕様開発及び第一次試作は完了した。描画プリミティブも試作が完成している。協調処理のための仮想マシン型言語については、仕様評価のためにブラウザ上に ECMA Script を移植することによって動作させるところまできている。順調に進捗しており、平成 21 年 9 月には、中間目標を予定通り達成できる予定である。

#### 4-3 小型化・省電力化のための S/W・H/W アーキテクチャの研究開発

##### 4-3-1 ハードウェアプラットフォーム

###### 4-3-1-1 0 次試作機全体構成

超小型汎用コミュニケーション端末の実現に向けて、平成 18 年度はチップセット機能の抽出、イベント処理に関連する機能、処理能力を評価するためのベースハードウェアを 0 次試作として開発した。

超小型コミュニケーション端末にはユビキタスコンピューティング環境のための必須ソフトウェアを搭載する必要がある。これらはサブテーマ 2 で研究開発を進めているブラウザや、サブテーマ 3 で研究開発を進めているマネージャ群がふくまれる。また、実証実験により評価していくことが想定されているので音声、画像、動画などについての表現力も豊かである必要である一方評価に耐えるような小型軽量化も必要である。

ユビキタスコンピューティング環境から発生する無線式あるいは赤外線式のマーカによるイベント、RFID R/W(最終的にはサブテーマ 1 で研究開発を進めているマルチプロトコル R/W に置き換わる)からの外部イベントを処理して、アプリケーションに渡す構造となる。そしてこのイベント処理機能部分がサブテーマ 3 のチップセットの対象である。

このような点から考えハードウェアプラットフォームは、本体部と外部イベントに係わる部分からなり、特に外部イベントに係わる部分は今後の研究開発を進める上で複数種類、複数方式を試行する必要があるため、本体部と分離できるような構造を適用する。本体部に対して分離できる外部イベントに係わる部分をアタッチメントと呼ぶ。

本体部にはメイン CPU に加え、サブ CPU を設ける。サブ CPU は電源制御やリアルタイムクロックなど、メイン CPU を停止させた時でも稼働する必要な機能を担う低消費電力の CPU である。また、このサブ CPU は、アタッチメントの CPU と通信できるようになっている。アタッチメント CPU はアタッチメントに内蔵するイベントの受信装置やその他の周辺を制御する。

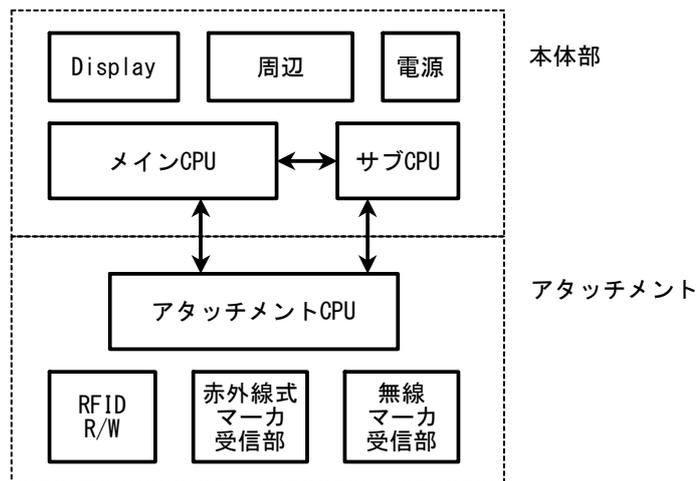


図 4-3-1-1: 全体構成図



図 4-3-1-2: 0次試作機(アタッチメントタイプB 接続状態)

#### 4-3-1-2 本体部

本体部は、大型の表示部を中心とした、PDA 様の形状をしている。タッチパネル付きのカラーディスプレイとわずかなスイッチ類が配置されている。本体下部は 50 ピンのコネクタがあり、これでアタッチメントと信号のやりとりや電源の供給を行う。本体内蔵のバッテリー充電も 50 ピンコネクタから行うことができる。音声入出力信号も 50 ピンコネクタに配置されていて、アタッチメントにヘッドホンやマイクを接続する。



図 4-3-1-3: 本体部

(1) メイン CPU 及びメモリ

小型軽量化、省電力、高性能の観点から、現在の高機能携帯電話のアプリケーションプロセッサとして利用されている CPU を利用した。動作クロックは 216MHz である。この CPU は同一パッケージ内に 64MB の SDRAM(メモリ)を内蔵している。

また、本 CPU には、種々の I/O 以外に動画あるいは静止画の MPEG/JPEG を展開するためのアクセラレータを内蔵している。

(2) 不揮発性ストレージ

不揮発性ストレージとして、8MB の Flash メモリと miniSD カードが利用できる。miniSD カードスロットはバッテリーケース内にあり、miniSD カードを交換することによりプログラムやデータを入れ替えることができる。また、アタッチメント接続用の 50 ピンコネクタの USB 端子にホストコンピュータを接続すると、miniSD カードがディスクストレージに見えるようになり、外部から書替えが可能である。ホストコンピュータとの接続は充電台で充電するときに USB が接続され、充電台に接続されたホストからアクセスできるようになっている。

(3) ディスプレイおよびタッチパネル

ディスプレイは、タッチパネル付き 3.5 インチ 320×240ドットの有機 EL ディスプレイを採用した。有機 EL ディスプレイは自己発光型のデバイスで、バックライト付き LCD のようにフィルターを必要としないため発光効率がよく低消費電力となる。また、輝度も高く屋外での利用でも有利である。有機 EL の特長として LCD に比べて応答速度が速く、動画再生時に残像のような現象が起きない。

(4) 無線

無線 LAN(802.11b)および近傍無線(Bluetooth)を搭載する。無線 LAN は、RFID や無線あるいは赤外線マーカから取得したユビキタス ID を解決して関連情報を取得したり、一括で情報をダウンロードするなどに用いられる。なお、街中など公共空間ではすでに 802.11b 等の一般的な無線 LAN の電波が管理されないで利用されているため、ユビキタスコンピューティング環境を実用化するためには、管理された無線通信手段が必要であり、今後の動向にあわせて適切なデバイスに置き換えていく。

近傍無線は本体の周辺で使う周辺機器と無線で通信するための物である。たとえば

ヘッドセット、RFID R/W あるいはアタッチメント相当の外部イベント受信機能などである。

#### (5) セキュリティ機能

eTRON セキュリテーアーキテクチャによるセキュリティモジュール SIM チップインタフェースを搭載する。課金やパーソナル情報などセキュリティを保つ必要がある情報の通信に適用できるようにする。

#### (6) モーションセンサー

ナビゲーションのための方位検知や端末の姿勢を知るために、モーションセンサーを搭載する。3 軸の地磁気と 2 軸の加速度センサーを採用した。

#### (7) サブプロセッサ

低消費電力のプロセッサで、メイン CPU がスリープ状態のときサブプロセッサはシステム監視をし、イベントによりメイン CPU をアクティベートする。機能としては、以下である。

- ・ バッテリ監視および充電制御
- ・ リアルタイムクロック監視
- ・ 電源制御(基板上の電源パート毎の入り切り)
- ・ モーションセンサー処理
- ・ タッチパネル処理
- ・ アタッチメントからのイベント処理

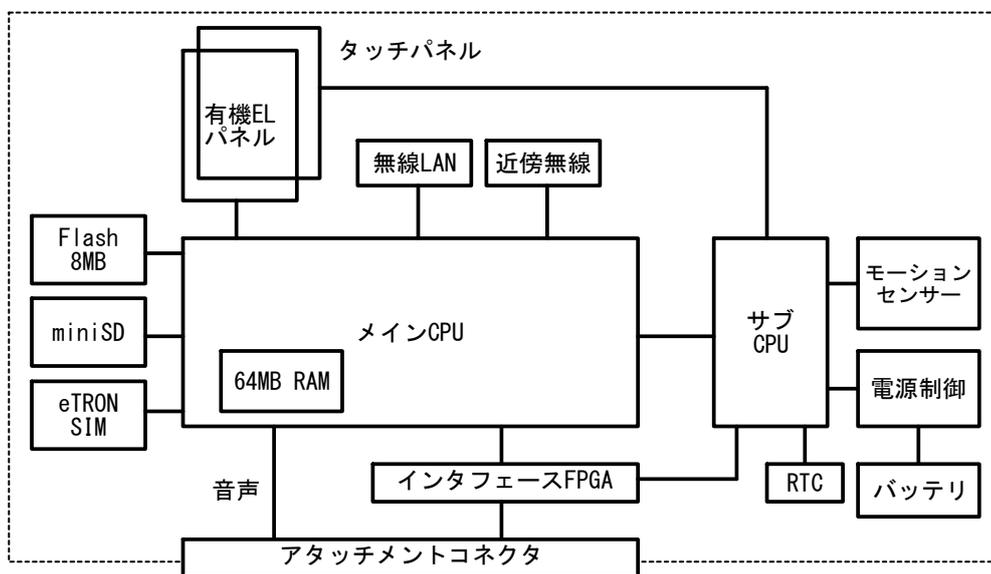


図 4-3-1-4 : 本体部のブロック図

### 4-3-1-3 アタッチメント

アタッチメントは外部イベント受信する複数のユニットを搭載し、本体部と結合することができるようになっている。0 次試作では、タイプ A、タイプ B の二種

類のアタッチメントを開発した。

タイプ A は、特定省電力無線マーカ受信部および赤外線式マーカ受信部を持つ基板を作成した。音声端子を持ちヘッドセットを接続する。

図中右先端部、および本体部接続コネクタ付近に赤外線式マーカ受信部がある。左側の白いケース中の無線マーカ受信部は、位置を評価するためケーブルで接続してある。右下の円盤状のものは通知用の振動モータである。



図 4-3-1-5: アタッチメントタイプ A

タイプ B はタイプ A に RFID R/W を追加したアタッチメントである。ヘッドホンはネックストラップに収納し、また、RFID タグを読むために読み取り部をタグに近づけられるように筐体には伸縮機構を設けた。

図中表裏にある黒い楕円状の部分が赤外線式マーカ受信部である。無線式マーカ受信部は赤外線式マーカ受信部の上部に内蔵されている。また、先端には RFID R/W が内蔵されている。



図 4-3-1-6: アタッチメントタイプ B

(1) 特定省電力無線マーカ受信部

429MHz 帯特定省電力無線式のコビキタス ID(ucode) マーカを受信する。ucode は 128bit で、一つのマーカからは 1 秒に 1 回程度の頻度で ucode が発信される。マ

一カは放送型で、受信部からマーカに対して応答を返すことはない。

マーカは、1 秒に 1 回の定期発信時刻に相互干渉防止のためキャリアセンスし、他のマーカが発信中の場合は、一定時間待って再度試みをする。この方式により構成される時間スロットに各マーカが割り当てられ、順に発信する。ただし、スロット数よりも近傍にあるマーカの数が多い場合は、あふれが起こり一定期間発信できないマーカがでる。

マーカ受信部は、受信したマーカの ucode と電波強度をシリアルインタフェースで出力する。また、受信したマーカの電波強度が一定以下の場合、シリアルインタフェースに出力しないように閾値を設定することができる。

#### (2) 赤外線式マーカ受信部

赤外線式マーカは、波長 880nm の IrDA-SIR 仕様をベースにした信号を発信する。変調はベースバンドで RZI (Return Zero Invert) で、値 0 のときそのビット時間の 3/16 の赤外線光パルスが発信される。赤外線式マーカから発信されるユビキタス ID (ucode) は 128 ビットである。マーカは、1 秒に 3~5 回程度 ucode を発信する。

赤外線式マーカ受信部は、880nm の赤外線信号受光部および復調部からなる。復調後はシリアル信号として出力される。

#### (3) RFID R/W

アタッチメントタイプ B にはパッシブ型、周波数 13.56MHz、ISO15693 規格タグの小型 R/W を搭載している。パッシブ型 RFID の読み取りは、利用者が意識して行うことになるので、R/W の読み込み操作はスイッチを押して行う。アタッチメントの根元にある黄色いレバーが読み取りスイッチとして働く。またこのレバーは伸縮の機械的なりリリースボタンとしても機能する。

#### (4) アタッチメント CPU

アタッチメントのイベント受信機能を制御し、イベント処理を行うことができるようにアタッチメントに CPU を搭載している。アタッチメント CPU も本体部のサブ CPU 同様消費電力の低いものを利用している。

平成 19 年度以降、この部分を利用してソフトウェア的にイベント処理の機能、能力を評価する。これに基づいて高度なイベント処理を省電力下で可能なイベント処理機能を実現するチップセットアーキテクチャを検討する。

### 4-3-1-4 イベント処理に関連するインタフェース構造

0 次試作ハードウェアプラットフォームでは、イベント処理に関する機能、性能を評価するために、イベントをメイン CPU、サブ CPU、アタッチメント CPU のどの CPU で、どのように処理するか、変化させて評価できるよう信号パスを多数設けている。

表 4-1-3-1 はアタッチメント CPU、サブ CPU、メイン CPU のそれぞれで、どのイベントの処理が可能かを示している。このうち RFID はメイン CPU のみでしか処理可能ではないが、RFID は利用者が意識的に操作するため、メイン CPU がアクティブであることを前提として問題ないからである。なお、ソフトウェア構造的にメイン CPU 以外で前処理をした方が望ましいという結果ができれば、イベント処理用のチップセットで処理に含めることは可能である。

表 4-3-1-1: 各 CPU の処理可能イベント

	アタッチメント CPU	サブ CPU	メイン CPU
赤外線式マーカ 1	○	○	○
赤外線式マーカ 2	○	○	○
無線マーカ	○	○	○
RFID			○
タッチパネル		○	○
モーションセンサー		○	○

イベント処理インタフェース構造は図 4-3-1-7 のようになっている。赤外線式マーカ受信部は 2 つあり、それぞれ復調後のシリアル信号がアタッチメント CPU の二つのシリアルインタフェースに接続される。

無線マーカ受信部は、ジャンパー接続によりアタッチメント CPU に接続するか、インタフェース FPGA 経由でメイン CPU のシリアルポートに接続するかを切り替えることができる。

アタッチメント CPU は、サブ CPU と接続され、さらにサブ CPU からメイン CPU に接続されている。このパスを利用すると、赤外線式マーカ、無線マーカのイベントをアタッチメント CPU、サブ CPU、メイン CPU のいずれでもあるいは分散した形で処理可能である。

インタフェース FPGA は、メイン CPU、サブ CPU および、アタッチメント内の RFID R/W および無線マーカ受信部の間に入っており、信号の流れを制御できる。また、必要に応じてロジックを変更することができるようになっている。

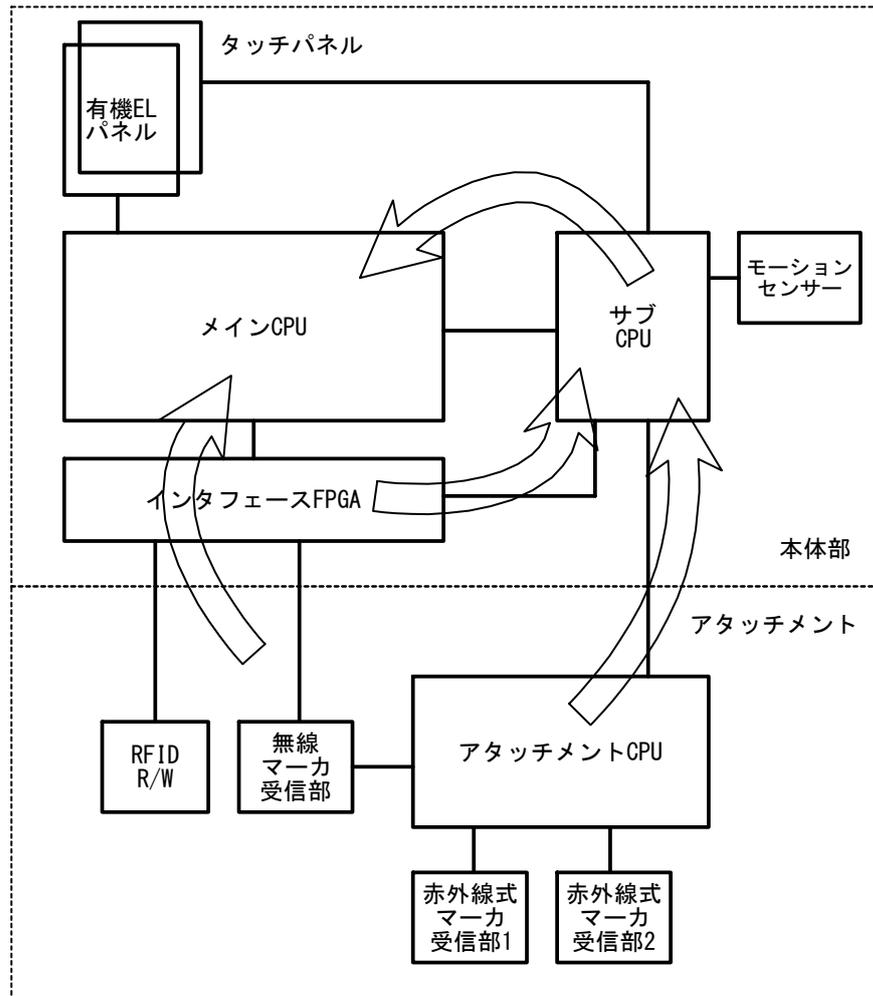


図 4-3-1-7: イベント処理インターフェース構造

#### 4-3-1-5 動作実験

ハードウェアプラットフォーム0次試作機の外部イベントの受信機能について現実のユビキタスコンピューティング環境において、ハードウェア構成の性能が十分であるか、またイベント処理のアルゴリズムとして検討すべき問題抽出の目的で、実験を行った。銀座四丁目交差点を中心に設置された特定小電力無線マーカを利用して、受信動作を確認した。図 4-3-1-8 は、無線マーカの配置図である。●印が設置箇所を示し、中央部の 100m 範囲に 36 個のマーカが設置されている。各マーカは出力 1mW とし 1 秒に 1 回 ucode を発信している。

マーカの取り付け場所は、植栽の地面近く、および街路灯の高さ約 2.5m 付近である。取り付け場所あるいは、その周囲の金属による障害がみられるが、目的のマーカから概ね 5m 程度の距離で受信可能であった。車両の反射などにより遠方のマーカが受信できることもあり、5~8 程度のマーカを受信する。

このように複数受信しているマーカの中から、たとえば最も近いマーカの ucode をメイン CPU に受け渡すことがイベント処理部の機能となる。複数のマーカからの信号が逐次受信されるので、どれを採用するかアルゴリズムが必要となる。マルチパスや周辺ノイズの影響で単純に特定の電波強度を閾値ととして採用するとい

う方法では、目的でないマーカを採用してしまう場合がある。また、マーカの取り付け状況により、マーカから等距離であっても電波強度にはばらつきが生じる。

これらの現象を加味すると、平均化処理、履歴処理、特定場所におけるマーカ出力の補正值などを利用した処理が必要となる。また、どのようなアプリケーションで利用するかによってもこの処理アルゴリズムは変える必要がある。たとえばナビゲーションを想定するなら、近傍のマーカだけではなく、さらに離れた位置のマーカとの関係や、移動して受信してきた履歴を使った処理が必要となると考えられる。

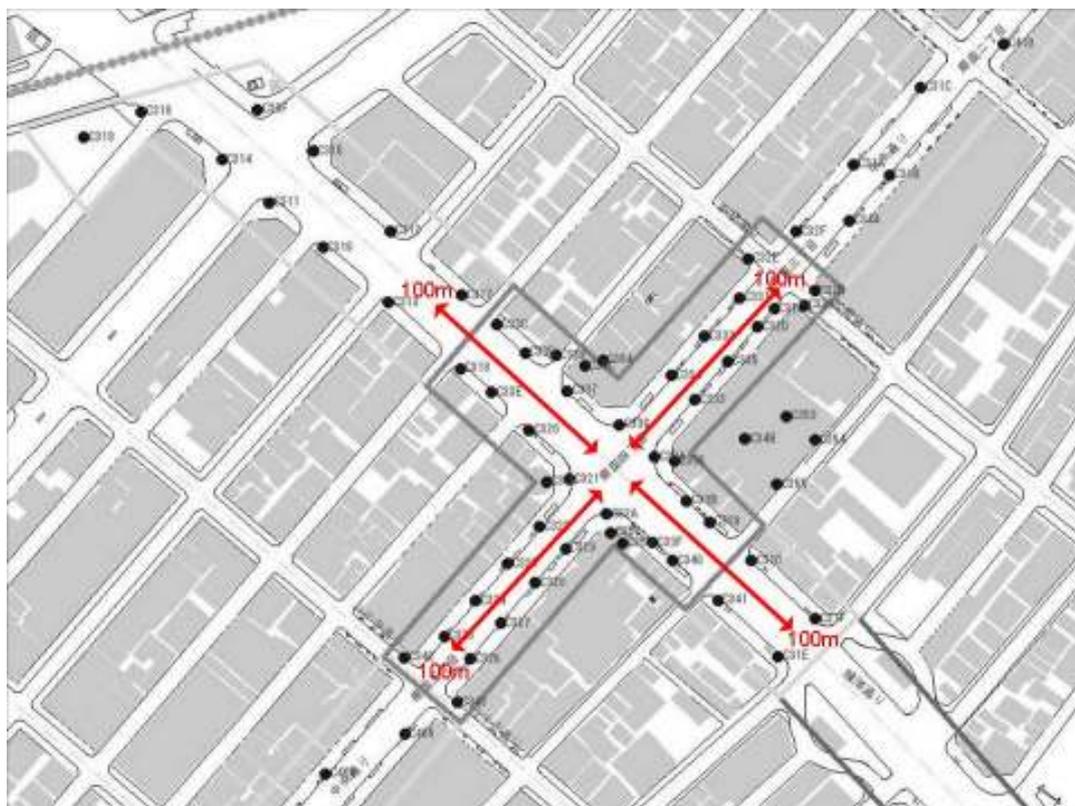


図 4-3-1-8: 屋外無線マーカ設置図

図 4-3-1-9 は、地下街に設置した無線マーカの受信状況の一例である。小さい青丸印で示されたのが無線マーカである。赤丸のついた T019 の ucode をどこで受信できたかを大きい薄目のブルーの丸印で示している。天井高さの低い地下街では、反射の影響で、地上よりも遠いマーカを受信することがわかる。

地下街の天井には、赤外線式のマーカを設置してあり、1秒間に3回発信させている。無線マーカの信号をうけながら、赤外線の信号を受信できるかについても実験を行い、信号をメイン CPU に上げることが確認できた。

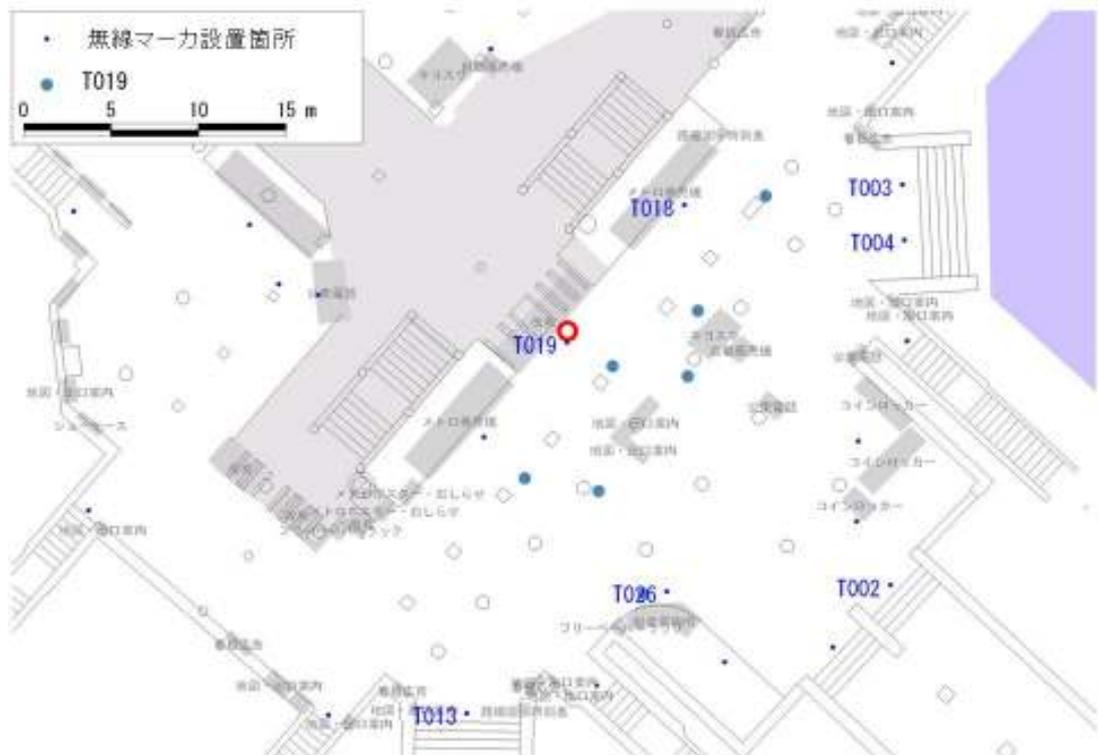


図 4-3-1-9: 地下街設置の無線マーカ (天井に赤外線マーカ設置)

しかしながら、メイン CPU でこれを処理すると 1 秒間に 3 回の赤外線と 1 秒に 1 回の複数の無線マーカを処理することになる。メイン CPU は、目的の ucode を受信したら、関連する情報を音声、画像、動画などで提示する処理を行う必要があり、外部イベントの処理はメイン CPU 外で行う必要がある。どの信号をうけてどのような情報を提示したかは、アプリケーションレベルでも履歴として必要であるが、同じ位置にいたときに、1 秒間に 3 回受ける赤外線式マーカの信号をマスクするためにはイベント処理部にも履歴を処理する機能や、上位のアプリケーションからその履歴情報をどの程度保持するかなどのやりとりが必要であることが考えられる。

これらのイベント処理機能を検討する上でのハードウェアプラットフォームは、18 年度の成果として完成することができた。

## 4-3-2 ソフトウェアプラットフォーム

### 4-3-2-1 研究開発内容

本開発は、超小型汎用コミュニケーション端末のアプリケーション・ソフトウェアの基盤となるソフトウェアプラットフォーム(以下、本プラットフォーム)の研究開発である。

超小型汎用コミュニケーション端末は RFID やアクティブタグなど様々なデバイスからの情報に対応し、各種のメディア情報を扱うため、リアルタイム性能が要求される。そこで本プラットフォームはリアルタイム OS を中核としたリアルタイムシステムでなくてはな

らない。

そこで、リアルタイム OS 上に、超小型汎用コミュニケーション端末のアプリケーション・ソフトウェアであるサーバー・クライアント協調処理型ブラウザを動作させるための API セットを持ったマネージャ群を実現する。

その際、本端末を特徴付ける以下の機能を含まなくてはならない。

1. ブラウザインタフェース (プラグイン機能インタフェース)
2. RFID 管理
3. アクティブタグ管理
4. セキュリティ管理 (VPN機能を持つ)
5. コンテキスト管理

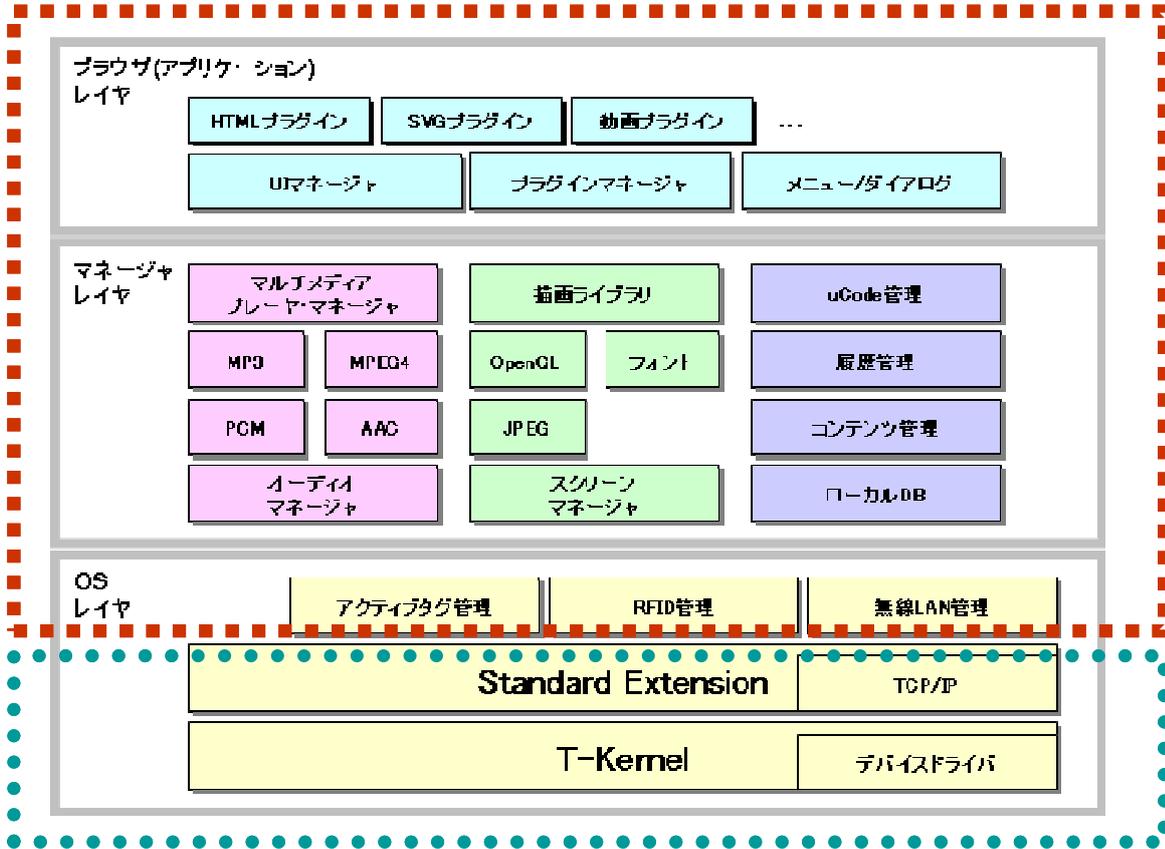
現在本プラットフォームは、OS やデバイスドライバなど基本的なシステムの実装が完了し、ブラウザインタフェースとなる UI マネージャとプラグインマネージャ、そして RFID やアクティブタグなどを管理するイベントマネージャや ucode マネージャが動作するに至っている。

コンテキスト管理については、現在は端末内に内蔵したデータベースを用いたローカルな環境で実験・検証を進めている。今後、セキュリティ管理機能と共にネットワーク対応を本格的に進めていく計画である。

## 4-3-2-2 ソフトウェア全体構成

ソフトウェアが大きく三つのレイヤに別けることができる。図4-3-2-1に構成図を示す。

### 本研究開発で研究開発した部分



### 既存の成果の微修正で構築した部分

図 4-3-2-1 ソフトウェア全体構成図

#### (1) OS レイヤ

OS の最も中心であるカーネルにはリアルタイム OS である T-Kernel を使用する。各デバイス（周辺ハードウェア）に対応したデバイスドライバは、この T-Kernel により管理される。

OS の上位層として T-Kernel/Standard Extension を用いる。メモリ管理やファイル管理はここで行う。メモリ管理は MMU を用いた多重論理空間モデルを採用する。また、Standard Extension を拡張し、TCP/IP プロトコルスタックを実装した。本プロトコルスタックは BSD に準拠した標準的なものである。

OS レイヤの最上位に、アクティブタグ管理、RFID 管理、無線 LAN 管理などのデバイスの上位の管理を行なうマネージャ群を設ける。

イベント管理の基本的な機能の提供も本レイヤで行なう。イベント管理機能を実現するイベントマネージャは Standard Extension の機能として実現される。デバイスからの非同期的な事象通知や、アクティブタグ、RFID から取得された固有番号 (ucode) も全てイベントとして一元的に管理される。

## (2) マネージャレイヤ

超小型汎用コミュニケーション端末のアプリケーション・ソフトウェアであるサーバー・クライアント協調処理型ブラウザを動作させるための API セットを持ったマネージャ群をこのレイヤで実現する。

マネージャは大きく以下の三つに分かれる。

第一に、オーディオや動画データの再生を行なうマルチメディア系のマネージャ群である。このマネージャ群は、マルチメディアプレーヤ・マネージャが全体を統括する。デバイス層とのインタフェースはオーディオマネージャが行なう。

第二に、グラフィック関係のマネージャ群である。アプリケーションへの API は描画ライブラリが受け持ち、その下位にフォントマネージャや OpenGL ライブラリが存在する。デバイス層とのインタフェースはスクリーンマネージャが行なう。また、スクリーンマネージャはマルチメディアプレーヤの動画系のプレーヤとも連携を行なう。

第三は、ucode 関係のマネージャ群である。ucode は無線マーカや赤外線マーカなどのアクティブタグや RFID などのパッシブタグから取得される。ucode に結び付けられたコンテンツデータの情報を得るのが ucode マネージャである。また、取得した ucode の時間的管理を行なうのが履歴マネージャである。今回のプラットフォームではコンテンツ・データは主にローカルなディスク上に置かれるが、ネットワーク経由での取得も可能である。その際にデータのキャッシュ管理などをコンテンツマネージャがおこなう。

またそれぞれのマネージャはローカルのデータベースを用いて各種の管理を行なう。

## (3) ブラウザ (アプリケーション) レイヤ

超小型汎用コミュニケーション端末のアプリケーション・ソフトウェアであるサーバー・クライアント協調処理型ブラウザが本レイヤに相当する。アプリケーションの各機能は、プラグインとして実装され、それをプラグインマネージャにより管理する。

GUI をベースとしたユーザインタフェースは、UI マネージャが担当する。また、ダイアログやメニューなど、様々なユーザインタフェースを実現するマネージャが存在する。

### 4-3-2-3 デバイス管理

周辺ハードウェア（デバイス）の管理は、T-Kernel の管理下で各デバイスドライバが行なう。デバイスドライバの仕様は基本なものは T-Kernel で規定されているので、汎用性や互換性を考慮し、極力それにしたがるものとする。

また複数の異なったデバイスを上位のソフトより統合的に扱うために、上位の論理的なデバイスドライバを導入する。

たとえば、アクティブタグは Dice(無線マーカ)や赤外線マーカなど複数の種類が存在する。さらに、RFID などのパッシブなタグも上位のソフトから見れば、同じ ucode を発するデバイスである。そこで、これらアクティブタグやパッシブタグの上位デバイスドライバを設け、統合的に扱うことを可能とする。

以下、表 4-3-2-1 に本プラットフォームの主なデバイスドライバの一覧を示す。

表 4-3-2-1 主なデバイスドライバー一覧

名称	対象デバイス	備考
ビデオ	LCD 表示	QVGA サイズ
オーディオ	本体音声入出力	
PD	キー、タッチパネル	
ディスク	miniSD カード	
シリアル	本体 RS232C	
USB	USB ファンクション	ホストとの接続に使用
eTRON	eTRON SIM I/F	
方位・加速度	内蔵方位/加速度センサー	
Bluetooth	内蔵 Bluetooth	SPP, A2DP に対応
無線 LAN	内蔵無線 LAN	
赤外線	外部赤外線受信 I/F	ポート接続
RFID	外部 RFID リーダ I/F	シリアル接続
Dice	外部 Dice 受信機 I/F	シリアル接続
ucode	RFID, Dice, 赤外線, 白杖等の上位ドライバ	シリアルドライバの上位ドライバ
バイブレータ	外部バイブレータ	ポート接続
Clock	内蔵 RTC	
電源管理	電源管理・省電力制御	

#### 4-3-2-4 イベント管理

デバイスからの非同期の事象通知は、OS が提供するイベント機能を用いて、イベントとしてアプリケーション(ブラウザ)に通知される。

イベントは、ブラウザ内で、プラグインマネージャ、UI マネージャ、アクティブなプラグインの順で通達され、それぞれが自身に関連するイベントを取得し、それに応じた処理を実行する、

イベントは、アクティブタグやRFID タグの ucode を伝える ucode イベントと、ボタンやタッチパネル等のユーザ操作を伝えるボタン/PD イベントがある。

図 4-3-2-2 に ucode イベントの処理の流れを示す。

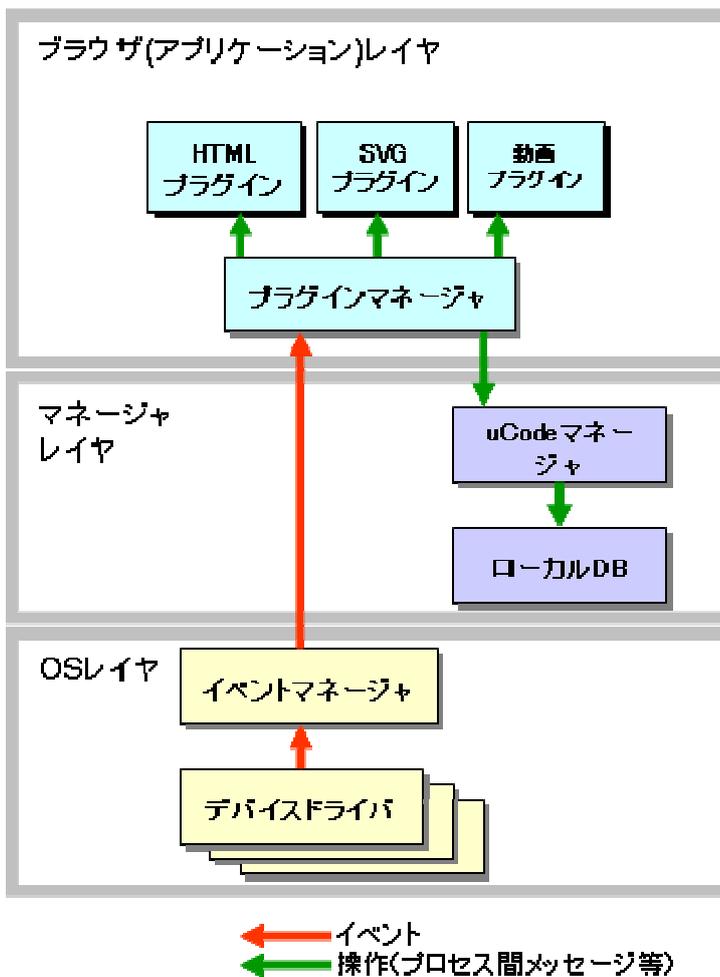


図 4-3-2-2 ucode イベントの処理の流れ

ucode イベントはプラグインマネージャにて処理される。プラグインマネージャは取得した ucode から ucode マネージャを介して対応するコンテンツ・データを得る。そしてそのコンテンツ・データの種別に応じて、プラグイン切り替えなどの操作を行なう。

図 4-3-2-3 にボタン/PD イベントの処理の流れを示す。

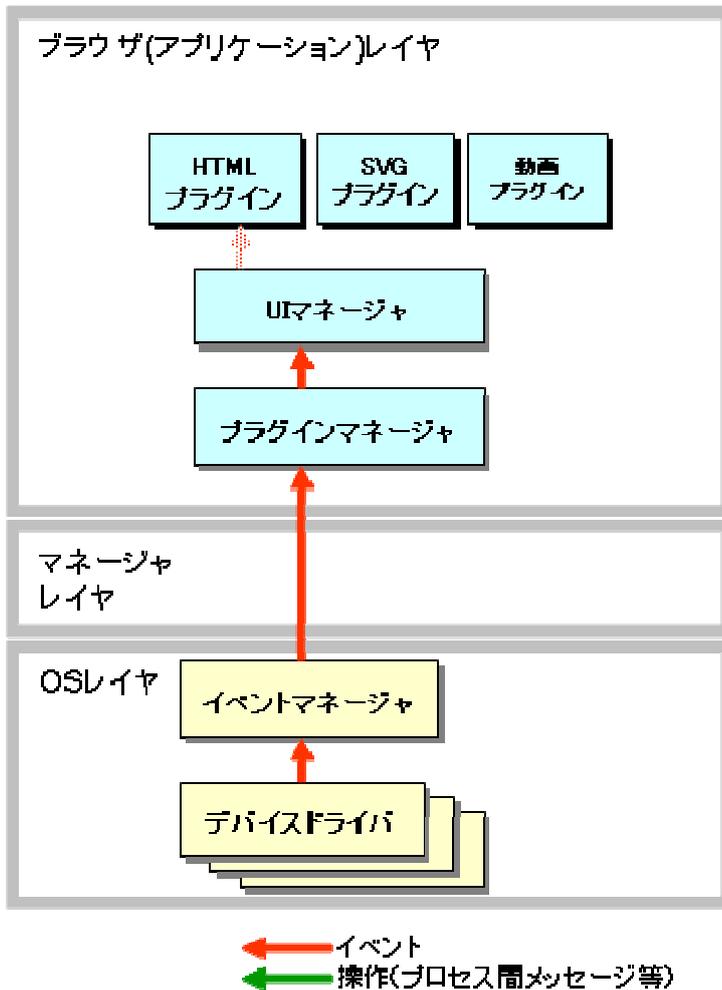


図 4-3-2-3 ボタン/PD イベントの処理の流れ

ボタン/PD イベントはプラグインマネージャを経由してUI マネージャに受け渡される。UI マネージャはそれがシステムの UI に対しての操作であれば、それに応じた処理を実行する。そうでない場合は、現在アクティブな状態のプラグインへイベントを引き渡す。以降、システムはイベントの処理を関知しない。それぞれのアプリケーションに応じた処理をイベントを取得したプラグインが実行する。もし、そのイベントにプラグインが対応していない場合は、イベントは破棄される。

#### 4-3-2-5 マルチメディア管理

本端末では、マルチメディア・データとして、MP3 や PCM などの音声データ、MPEG4 の動画データを扱う。

これらのマルチメディア・データを上位のソフトから統一的に扱えるようにする仕組みがマルチメディアプレーヤである。マルチメディアプレーヤは、再生、停止、一時停止などのいわゆるメディア再生として扱う事のできるデータを扱うことができる。

つまり、上位のソフトはデータの所在(ファイル)を指定し、再生/停止/一時停止のコールを行なうだけで、そのデータが MP3 なのか PCM なのか、または動画なのか意識することなく操作をすることができる。

以下、図 4-3-2-3 にメディアプレーヤの構成を示す。

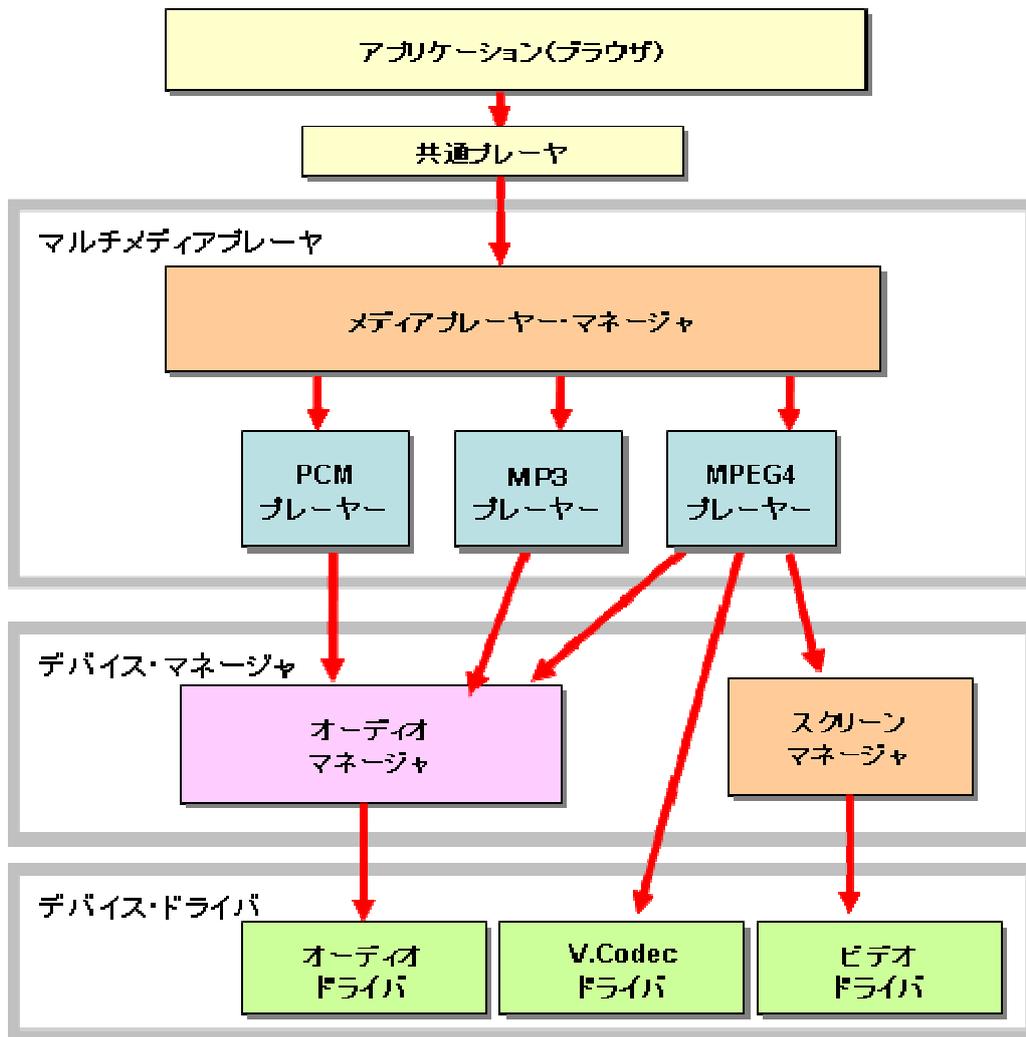


図 4-3-2-3 メディアプレーヤ・マネージャ構成図

マルチメディア管理の中心となるのはマルチメディアプレーヤ・マネージャである。実際の個々のデータの再生などを行うのは、データ種別に対応したメディアプレーヤである。本プラットフォームでは、MP3 プレーヤ、PCM プレーヤ、MPEG4 プレーヤなどのメディアプレーヤが存在する。

アプリケーションは、個々のプレーヤを意識することはなく、マルチメディアプレーヤ・マネージャを介して、再生したいメディア種別の論理的なプレーヤを生成することができる。

ただし、本プラットフォームのアプリケーションであるブラウザは、その上でさらに複数のプラグインを実行する。個々のプラグインが独立にメディアプレーヤを生成した場合、リソースの衝突やそれを回避するための排他制御が煩雑になる、などの問題が生じる。

そこでブラウザに一つの共通プレーヤを設け、各プラグインはそのプレーヤを共用する事とする。共通プレーヤは、各プラグインからの要求を時系列に処理し、必要な排他制御を行なう。

アプリケーションやプラグインからみれば、共通プレーヤに対し自由に再生したいメディア・データの操作を要求するだけとなる。

以下の表 4-3-2-2 に共通プレーヤの機能一覧を示す。

表 4-3-2-2 共通プレーヤ機能一覧

名称	機能	備考
MP_PLAY	再生	
MP_STOP	停止	
MP_PAUSE	一時停止	
MP_FF	早送り	優先度低し
MP_FW	巻き戻し	優先度低し
MP_NC	次チャプタ	優先度低し
MP_PC	前チャプタ	優先度低し
MP_REPLAY	最初から再生	

#### 4-3-2-6 グラフィック管理

画像表示は、スクリーンと呼ぶ仮想的な画面を単位とする。スクリーンはスクリーンマネージャにより管理される。アプリケーションはスクリーンマネージャを介して任意のスクリーンを生成し、そのスクリーンに対して画像の描画を行なうことができる。

このスクリーンはメモリ上に確保された画像領域と描画のための各種情報である。これを論理スクリーンと呼ぶ。これに対し、実際に端末の画面上に表示されるスクリーンを物理スクリーンと呼ぶ。物理スクリーンはハードウェアにより決定される VRAM であり、ビデオドライバにより制御することができる。

アプリケーションより描画可能であり、実際の端末の画面にも表示可能なスクリーンは、論理スクリーンと物理スクリーンのペアとして実現される。これを通常スクリーンと呼ぶ。物理スクリーン、論理スクリーンの関係は一般に言う VRAM とオフクスリーンの関係と同等である。

以上のようなスクリーンの制御はスクリーンマネージャにより行なわれる。ビデオドライバはスクリーンマネージャにより制御され、アプリケーションからは隠蔽される。

以下の図 4-3-2-4 にグラフィック管理の概要を示す。

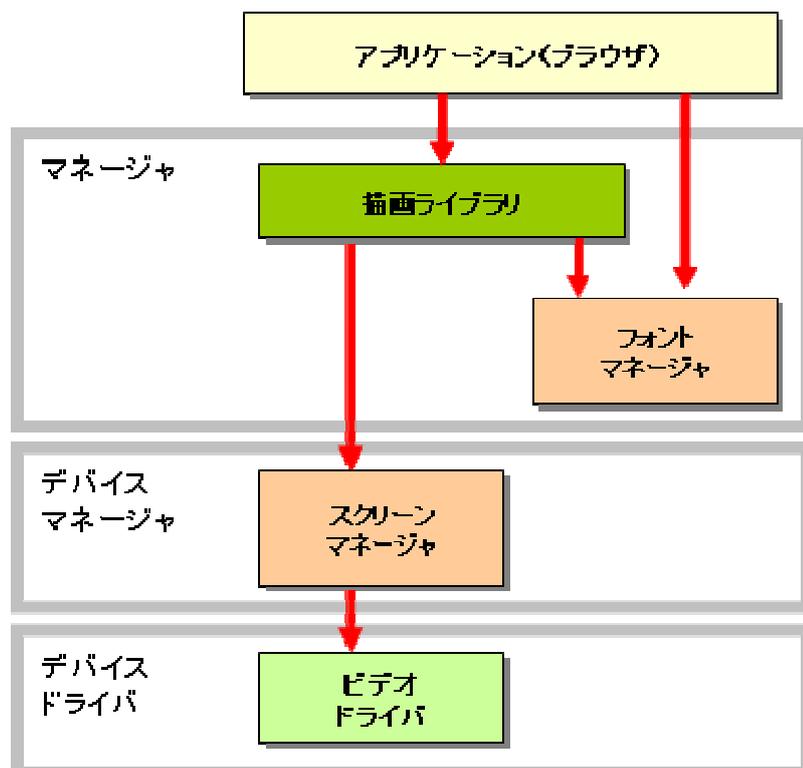


図 4-3-2-4 グラフィック管理の概要

スクリーンへの描画は、アプリケーションが直にメモリ上に操作することも出来るが、一般的な図形描画を行なう描画ライブラリが用意されている。

文字描画については、ビットマップフォントおよびアウトラインフォントに対応したフォントマネージャを用いる。なお、文字描画自体は、直接フォントマネージャを操作しなくとも、描画ライブラリを介して行なうことができる。

描画ライブラリの機能を表 4-3-2-3 に示す。描画命令の体系は、オープンな TRON のグラフィックプリミティブの仕様に準拠している。また、別に OpenGL 仕様に準拠した描画ライブラリも開発を計画している。OpenGL 仕様はハードウェアとの連携が比較的容易であり、3D 描画機能を持つなどの特徴があり、今後期待される仕様である。

表 4-3-2-3 描画ライブラリ機能一覧

コール名称	機能
<b>描画環境</b>	
qgOpenGev	描画環境のオープン
qgCloseGev	描画環境のクローズ
qgRefGev	描画環境の情報の参照
qgSetClip	クリッピング領域の設定
qgGetClip	クリッピング領域の取得
qgEnableRef	画面更新の有効化
qgDisableRef	画面更新の無効化
<b>図形描画</b>	
qgSetPen	線描画情報の設定
qgDrawPoint	点の描画
qgTestPoint	点の情報取得
qgDrawLine	直線の描画
qgDrawPline	直線列の描画
qgDrawRect	長方形の描画
qgFillRect	長方形の塗りつぶし
qgDrawOval	楕円の描画
qgFillOval	楕円の塗りつぶし
qgDrawPolygon	多角形の描画
qgFillPolygon	多角形の塗りつぶし
qgDrawArc	弧の描画
qgDrawSect	扇形の描画
qgFillSect	扇形の塗りつぶし
qgDrawCho	弓形の描画
qgFillCho	弓形の塗りつぶし
<b>文字描画</b>	
qgSetChp	文字描画位置の設定
qgGetChp	文字描画位置の取得
qgSetChc	文字描画色の設定
qgSetChd	文字描画方向・間隔の設定
qgSetScript	文字セットの指定
qgSetFont	フォントの指定
qgGetFont	フォント情報の取得
qgRefChar	文字情報の参照
qgRefString	文字列情報の参照
qgDrawString	文字列の描画
qgDrawStringP	任意位置への文字列の描画

#### 4-3-2-7 コンテンツ管理

ucode マネージャは、デバイスなどからの ucode を入力として、端末内部で保持しているコンテキスト情報を用いて、コンテンツの所在情報を取得・履歴を記録する。ucode マネージャの下位に履歴マネージャ、ローカル・データベースが存在する。

以下の図 4-3-2-5 にコンテンツ管理の概略を示す。

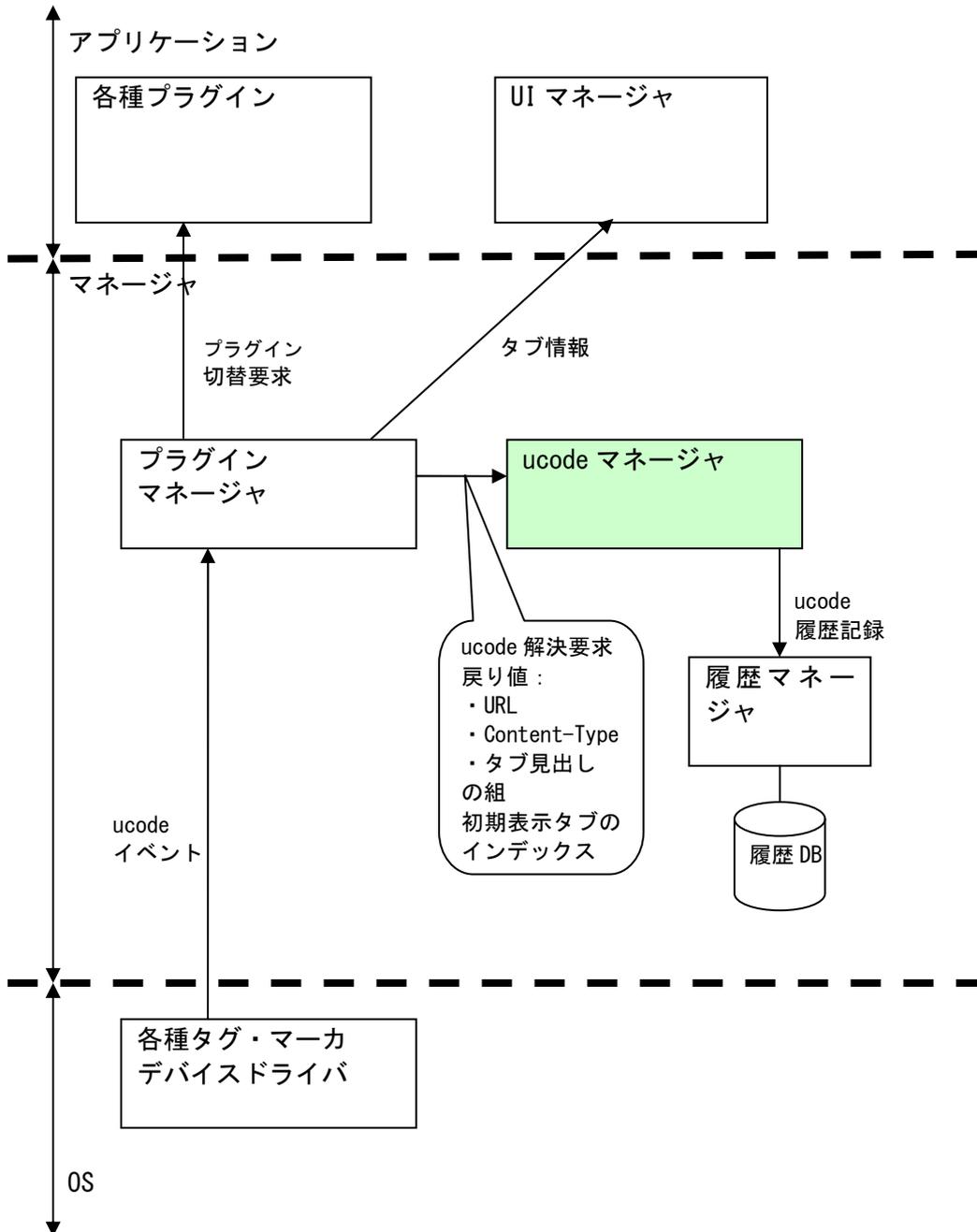


図 4-3-2-5 コンテンツ管理の概略図

コンテンツの管理において各ソフトウェアは以下の処理を行う。

(1) プラグインマネージャ：

各種ドライバから ucode イベントを受け取り, ucode マネージャへ ucode 解決要求を発行し, 解決結果としてコンテンツの URL を取得する. この URL に基づき, コンテンツ・データを表示・再生するのに最適なプラグインへ制御を切り替え, 対象のプラグインへコンテンツ・データの URL を転送する。また, ucode マネージャから取得したタブ情報を基に, UI マネージャにタブ情報の再描画要求を行なう。

(2) ucode マネージャ

プラグインマネージャからの ucode 解決要求に対し, 各種コンテキスト情報を用いて ucode 解決を行う。以下の項目を含むコンテンツの情報を解決結果として返却する。

タブ情報を表す URL・Content-Type・タブ見出し文字列のリスト  
リスト要素でシステムが最初に表示すべき項目のインデックス番号

(3) 履歴マネージャ：

ucode マネージャから ucode を受け取り, ローカル・データベースに記録する。各プラグインからコンテンツ情報を受け取り, ローカル・データベースに記録する。

また, 履歴 UI (ブラウザの機能) から履歴リスト検索要求を受け取り, ローカル・データベースから該当する履歴データを取り出す。

(4) デバイスドライバ：

アクティブタグや RFID などから ucode 情報を取得し, ucode イベントの形式でプラグインマネージャに送信する。

(5) UI マネージャ

プラグインマネージャから受け取ったタブ表示情報に基づいてタブを描画する。

(6) 各プラグイン

プラグインマネージャから渡された URL に基づいて各種コンテンツを表示・再生する。

ユーザからの ucode リンクの参照操作により, 擬似 ucode イベントをプラグインマネージャに送信する。

現状、コンテンツ・データは原則としてローカルのファイルシステム上に存在するものとする。但し、データの所在は URL で示されるので、ネットワーク上のファイルを指定することも可能である。

ネットワーク上のファイルの取得は、現状はプラグインが独自に行なう場合と、コンテンツマネージャは行なう場合がある。ただし、将来的にはコンテンツ・データの管理は全てコンテンツマネージャにより行われる事が望ましい。

また、超小型汎用コミュニケーション端末の場合、ネットワークへの接続は無線 LAN などの通信を前提としている。そのため、電波状況や、その場所のインフラの整備状況により常にネットワークに接続できるとは限らない。よって、ネットワーク上のデータであっても接続が出来ない場合は、ローカルのファイルシステム上のデータを使用する仕組みが必要である。これは、コンテンツマネージャのデータ・キャッシュ機能を用いて実現される。

コンテンツマネージャは、プラグインマネージャからネットワーク上のデータの URL を取得すると以下の手順でコンテンツ・データを取得する。

(1) ネットワークに可能であれば、URL で指定されたネットワークのサーバ上のファイルをローカルのファイルシステムにダウンロードする。

(2) (1)でダウンロードしたデータを指定されたコンテンツデータのキャッシュデータとして、データベースに記録する。

(3) ネットワークに接続が不可能であれば、ローカルのファイルシステムにコンテンツのキャッシュ・データがあるか確認する。もしあればそのデータを使用する。

コンテンツマネージャは以下の今後の課題が残されている。

(1) コンテンツデータが複数のファイルより構成されている場合 (たとえば、一つの HTML ファイルと複数の JPEG ファイルなど)、構成ファイルリストを用いて管理しているが、ファイルの所在などに制約がある。また、実際のデータとリストの管理が煩雑となる。よって、より効率的な管理方法が必要である。

(2) HTML の中で記述されたリンクを表示する場合など、HTML ブラウザ(プラグイン)のキャッシュ機能と重複、ないしは衝突する問題がある。これには HTML を解釈するプラグイン自体からコンテンツマネージャを使用するような拡張が必要である。

### 4-3-3 まとめ

ハードウェアプラットフォームについては、0次試作としてのベースハードウェアの開発を実施した。本ハードウェアの上にサブテーマ 3-2 のソフトウェアプラットフォームを稼動させ、そのプラグイン機能の中で、サブテーマ 2 にある、協調処理型ブラウザの第一次試作を完了した。

一方、ソフトウェアプラットフォームについては、各種ブラウザを組み込むためのプラグイン機能を持ったプラグインマネージャの第一次試作を行った。ユビキタス応用に最適化したユーザインタフェースを持ったプラグイン機能が実現されている。更に、赤外線、RFID、アクティブタグ等からの外部イベントを効率よく処理するためのイベントマネージャの第一次試作を行った。

### 4-4 総括

全体としては、電子手帳サイズのコミュニケーション端末の第一次試作を行った。今年

度の目標としていた、以下の1～4の機能をすべて備えている。

1. 近距離通信モジュールとして、RFID R/W 装置と無線受信モジュールの双方を備える
2. 広域網通信モジュールとして、IEEE 802.11b または PHS モジュール
3. カラー液晶またはカラーの有機 EL ディスプレイによるグラフィック表示機能
4. タッチパネルによるヒューマンインタフェース機能

今後は、ハードウェアの評価・検証を行うことで、仕様の見直しも視野にいれつつ、省電力化、性能向上等の最適化を実施する。全般的に順調に進捗しており、平成 21 年 9 月には、中間目標を予定通り達成できる予定である。

超小型マルチプロトコル RFID R/W 装置については、現在第一試作のための設計が終了した段階であり、当初の計画より多少遅れている。ただし、平成 19 年末には 1 次試作を終えられる予定であり、今後の研究開発のスピードアップを図ることで、この遅れは十分挽回可能である。アクティブ型 RFID タグに関しては、発電素子を一体化したモジュール試作はまだ行えていないが、周波数やチャンネルをチューニングする機能に関しては搭載しており、予定通り開発が進んでいる。

ブラウザについては、一次元、二次元データを閲覧するブラウザプラグインの仕様開発及び第一次試作は完了している。描画プリミティブも試作が完成している。協調処理のための仮想マシン型言語については、仕様評価のためにブラウザ上に ECMA Script を移植することによって動作させるところまでできている。順調に進捗しており、平成 21 年 9 月には、中間目標を予定通り達成できる予定である。

ハードウェアプラットフォームについては、すでに評価ボードは試作が完了している。ボードに関しては、今後評価検証し、更に最適化を行うことである。またイベント選択のスクリーニング機能の研究開発が今後の課題である。実環境におけるイベント発生を想定して最適なイベント抽出処理に必要な方式を 19 年度に検討を予定しており、これにより目標とするアーキテクチャが明確化される。最後に、ソフトウェアプラットフォームであるが、ベースとなるプラグインマネージャ、RFID やアクティブタグからのイベントを処理するイベントマネージャの試作が完了している。これらのマネージャに関しては、評価を行い、仕様及び性能に関するブラッシュアップを行っていく。順調に進捗しており、平成 21 年 9 月には、中間目標を予定通り達成できる予定である。

## 5 参考資料・参考文献

### 5-1 研究発表・講演等一覧

#### (1) 論文

1. [参考] 小林真輔, 諸隈立志, 坂村健:「様々な通信プロトコルに動的対応可能な RFID リーダライタの試作」組込みシステムシンポジウム 2006 論文集, IPSJ シンポジウムシリーズ Vol. 2006, No.9, pp. 16 - 24, 2006 年 10 月. (本論文は、平成 17 年度で終了した、前の民間基盤技術研究促進制度のプロジェクトにおける成果であり、本研究プロジェクトにつながる先行研究となったもの。査読付、「論文奨励賞受賞」)
2. [参考] Shinsuke Kobayashi, Tatsushi Morokuma, Ken Sakamura, “A Dynamic Retargettable Multi-Protocol RFID Reader/Writer,” IEEE International Symposium on Ubiquitous Computing and Intelligence (UCI07), May 21-23, 2007 (本論文は、平成 19 年度掲載分となります).

#### (2) 解説著書等

1. Ken SAKAMURA: "Intelligent House in the Age of Ubiquitous Computing - The House

for Sustainability", Proceedings of International Future Design Conference on Changing Places of Digi-log Future, 2006(会議録掲載論文).

2. 坂村健:「ユビキタスコンピューティング」, 3D 映像 三次元映像のフォーラム, 通巻 79 号, Vol.20 No.4, 2006.(会議録掲載論文).

### (3) その他資料

1. "越塚登:「各分野の第一人者が緻密に予測する、技術の未来図 20XX 年のユビキタス ロボット Web」(Tech 総研), 2006 年 11 月.  
[http://rikunabi-next.yahoo.co.jp/tech/docs/ct\\_s03600.jsp?p=000951](http://rikunabi-next.yahoo.co.jp/tech/docs/ct_s03600.jsp?p=000951)"
2. 越塚登:「ユビキタス ID 技術とユビキタスミュージアム」, 情報管理 Vol. 49, No. 8, 2006 年 11 月号, (独) 科学技術振興機構, pp. 417~424.
3. 坂村健:「UNL—トロンキーボード (2006)」, アクシス 10 月号, Vol.123, 2006(掲載記事・雑誌)
4. 坂村健:「ユビキタス社会はこうなる—『どこでもコンピュータ』の時代」, 神道時事問題研究 第 570 号, 2006, 花園神社(掲載記事・雑誌)

### (4) 報道発表(YRP UNL)

1. 2006 年 12 月 4 日 T-Kernel/Standard Extension がカーナビ製品に採用
2. 2006 年 12 月 04 日アタッチメントにより応用向けに最適構成がとれる小型軽量の新型ユビキタス・コミュニケータを開発
3. 2006 年 12 月 4 日進む ucode/uID アーキテクチャの実用化
4. 2006 年 12 月 4 日 ucode を用いた物品管理事業の開始