遠隔監視向け端末のためのブロック型アーキテクチャの提案と評価

松本典剛[†] 遠藤浩通[†] 山田勉[†]
中三川哲明[†] 齊藤雅彦[†]

各種I/O機能をモジュール化し，複数のモジュールを組み合わせることで組込み計算機に必要な機能を実現するUniversal Block Architecture（UBA）を提案する．一応用例として，遠隔監視システム用端末を取り上げ，UBAのアーキテクチャを取り入れたブロック型超小型ユビキタスサーバを開発した．本ブロック型超小型ユビキタスサーバは，Ethernet，無線LAN，PHS（Personal Handy-phone System），DoPa（Docomo Packet），Bluetoothなどの各種ネットワークインフラのうち，設置場所の環境に最適な通信方式を選択できる．また，組込みLinuxを採用したことにより，各種通信機能と，USB（Universal Serial Bus）や，FPGA（Field Programmable Gate Array）などのI/Oとを連携して動作させるアプリケーションを容易に構築できることが特徴である．UBAでは，モジュール間のインタフェースが性能ボトルネックとなりうるが，メモリカードと無線LANを同時に使用しても，約5Mbpsの通信スループットを得ることができ，大幅な性能低下が発生しないことを確認した．これにより，超小型化，および，開発コストの削減を実現しながらも，遠隔監視サービスに適用可能な性能を確保することができる．

Development and Evaluation of the Embedded Computer with Universal Block Architecture for Remote Monitoring Services

Noritaka Matsumoto,[†] Hiromichi Endo,[†] Tsutomu Yamada,[†]
Tetsuaki Nakamikawa[†] and Masahiko Saito[†]

This paper describes the Universal Block Architecture (UBA) which can implement an embedded computer of various I/O functions by combining modules of the computer. We have developed "Blocked Super Small Ubiquitous Server" which utilizes the UBA for remote monitoring services. By using the "Blocked Super Small Ubiquitous Server", we can select the appropriate communication network according to the environment such as Ethernet, Wireless-LAN, PHS (Personal Handy-phone System), and DoPa (Docomo Packet). Also, we uses the embedded Linux for the system software so that the user can combine any communication network with the I/Os, such as USB and FPGA. Even with the combination of a memory card and Wireless-LAN, we can obtain the performance of the 5 Mbps throughput. Therefore, the UBA not only is applicable to the remote monitoring services, but also can reduce its size and development cost.

1. はじめに

ADSL（Asymmetric Digital Subscriber Line），光ファイバによる常時接続環境や，無線LAN，PHS，携帯電話といった無線データ通信の普及により，どこにいても情報を得ることができるユビキタス情報社会が実現しつつある[1),2)]．たとえば，携帯電話から終電の時刻を調べる，現在地に近いレストランを調べるといった，情報提供サービスをいつでもどこからでも利用できるようになってきた．かつて，Weiserは，"Ubiquitous Computing"が実現された社会では計算機が身の回りに意識せず浸透し，1人を複数の計算機が奪い合うと予測した[3)]．計算機が身の回りに『浸透』してきている例として，全国各地の天気情報や，道路の渋滞情報がセンサにより自動的に取得されている例があげられる．このようなコンテンツは，サービスの高付加価値化や新たなサービス創設に欠かせない情報となってきている．

しかしながら，情報サーバへのコンテンツの登録は，これまでは主に人手や他の情報源からの変換により実現されてきた．また，フィールドからの情報取得と情

† 株式会社日立製作所日立研究所
Hitachi Research Laboratory, Hitachi Ltd.

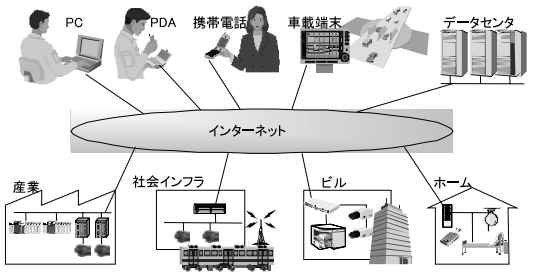

図 1 ユビキタス情報社会
Fig. 1 Ubiquitous information society.

報サーバへの情報蓄積は，これまでは比較的高価な計算機資源を利用し，一般に，各環境に依存した独自インタフェースで実現されている．『あまねく存在する情報』を様々なサービスで利用するためには，フィールドの情報を柔軟かつ安価に伝達する計算機の存在が不可欠となる．

以上のような社会状況を背景に，筆者らはユビキタスコントローラの研究を行ってきた[4]．ユビキタスコントローラとは，産業分野で培った制御技術・高信頼化技術を中心として，新たに，情報系標準技術（インターネット標準技術）を取り入れた組込み向け計算機システムである．適用対象として，発電所や工場といった産業機器，鉄道や上下水といった社会インフラ，ビル管理やホーム家電などの遠隔監視・制御があげられる（**図 1**）．インターネットなどの公衆網を介してこのような機器を遠隔から監視・制御するサービスは，情報の（バーチャルな）世界と，物の（リアルな）世界とを結び付けるサービスとして，あらゆる分野において注目されている[5),6)]．

そこで，このような遠隔監視サービスを安価に実現するためのキーコンポーネントとして，情報系標準インタフェースを有し，設置場所に合わせて最適なネットワークを選択できるフィールド設置型の組込み計算機システム（ブロック型超小型ユビキタスサーバ）を開発した．本稿では，ブロック型超小型ユビキタスサーバに適用するために考案したアーキテクチャを中心にして説明する．また，本開発のブロック型超小型ユビキタスサーバの性能について，特に実運用上の構成を想定した性能評価を行った．

2. 再構成可能アーキテクチャ（Universal Block Architecture）の提案

2.1 従来の組込み計算機の問題点

従来，監視用コントローラを開発する際には，標準開発環境に付加機能を追加して開発評価を行い，量産時には基板設計をやり直して，必要な部品のみで構成される製品を再開発するという方法をとってきた．この方法には，量産品を安価に製造できるというメリットがある一方，設計工数や設計コストが大きくなる，量産品の再評価が必要，などのデメリットがあった．そのほか，組込み用パソコンに機能を追加して開発・評価を行い，そのまま量産品として製造するという方法も用いられることがある．この場合，設計・評価を標準化できるという利点があるが，単品のコストが大きいことや，使用されない機能が多くなることなどが問題としてあげられる．

また，汎用の PC アーキテクチャを使用しながらも，電源，CPU，メモリ，各種 I/O などの主要な機能をワンボード化することによって小型化を実現した組込み計算機が製品化されてきている．しかし，この場合，欲しい機能が足りないために結局オプションボードが必要になったり，逆に，必要のない機能が余分についていたりする場合が多々ある．したがって，今後ユビキタス情報社会において，様々な場面で組込み計算機を活用する際に，必要かつ十分な機能を簡単に提供できるシステムが求められると考えた．

2.2 再構成可能アーキテクチャ

前節で述べたように，組込み計算機の汎用的な構成を低コストで実現することが求められている．また，センサネットワークやフィールド装置への組込みを考えると，小型でフレキシブルに実装可能な構成が必要である[7),8)]．

このような課題をふまえ，筆者らは次のような方法を提案する．第 1 に，コントローラを機能ごとにモジュール形式に分割し，開発時には，個々のモジュールを組み合わせてコントローラの評価を行う．量産時には用途に応じてモジュールを取捨選択し，必要な部品のみでコントローラを構成する．この方法を採用することにより，量産品を低価格化すると同時に，設計・評価も容易とし，手戻りをなくすことができる．

以上のような方法を実現するための技術として，プロセッサ，メモリ，電源回路を必須コンポーネントとし，他の拡張機能をすべてモジュール形式で追加，変更することが可能な Universal Block Architecture（UBA）を考案した．ここでは，特に，ネットワーク機能（Ethernet☆，無線 LAN，PHS，DoPa☆☆，Bluetooth☆☆☆）を柔軟に取捨選択できるアーキテクチャの実現を狙っている．このため，我々は，ハードウェアアーキテクチャとして UBA，ソフトウェアとして組込

☆ Ethernet は（株）Xerox の商標です．
☆☆ DoPa は（株）NTT ドコモの商標です．
☆☆☆ http://www.bluetooth.com/

みLinux☆を搭載したユビキタスコントローラ「ブロック型超小型ユビキタスサーバ “Ubiquitous Cube”」を開発した．開発した “Ubiquitous Cube” は，以下のコンセプトを有するコントローラである．

- 必要に応じて機能を取捨選択することにより，複数の事業にまたがって利用可能．
- TCP/IP，HTTP などの通信プロトコルや，HTML，XML などのマークアップ言語を用いて，情報と制御の連携を実現．
- 超小型・低消費電力ながらも遠隔監視に必要な通信性能を確保するハード・ソフトウェア．

我々は，前節に示したUBAのモジュール構成をバスアーキテクチャで実現するため，『UBAシステムバス』を実現した．また，モジュール間接続に関する設定を皆無とするため，割込みやチップセレクトを各モジュールに適切に振り分ける『リソースルーティング機能』，および，システムを構成するモジュールを検出するための『プラグアンドプレイ機能』を開発した．以下では，これらの機能について原理および詳細を説明する．

2.3 UBAシステムバス

UBAシステムバスは，低コスト，かつ，多様なデバイスを接続できることが必須条件である．多種のデバイスを接続するためのインタフェースとして，PCI（Peripheral Component Interconnect）[9),10)]，PC/104バス[11)]などがある．

本研究では，バックプレーンボードを使わずに組み立てる構成により，組立てサイズを必要最小限とする方針を採用する．この場合，デバイスや実装基準，バス信号が制約されるPCIバスは不向きである．また，バックプレーンボードを使用せずに組み立てるタイプのバス方式としてPC/104がある．しかしながら，PC/104バスはサイズが大きく，ISA（Industry Standard Architecture）バス[12)]などのデスクトップPC向けのアーキテクチャに依存している．このような理由から，我々は，超小型，高速，低価格を狙うアーキテクチャとして，多くのマイコンで制御可能なSRAM（Static Random Access Memory）インタフェースを基本アーキテクチャとして採用した．また，現在，無線LAN，PHS，DoPa，Bluetoothなどが多数Compact Flash（CF）カードで提供されている理由から，SRAMインタフェースに加えて，CFカードインタフェース[13)~16)]に必要な信号を追加することとした．すなわち，UBAシステムバスの特徴は以下のとおりとなる．

- コネクタでモジュール間を接続するビルディングブロック構成．
- シングルバスマスタ・マルチスレーブのSRAMインタフェース．Compact Flashカードインタフェース装備．

ビルディングブロック構成の外観を**図2**に示す．

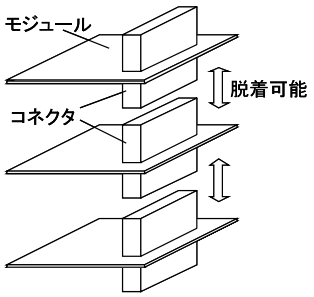

図2 ビルディングブロック構成
Fig. 2 Building block structure.

2.4 リソースルーティング機能

SRAMインタフェースにおいては，マスタデバイスがアクセス対象のスレーブデバイスを指定する場合，一般に，チップセレクト信号を用いる．逆に，スレーブデバイスからマスタデバイスへイベントを通知する場合には，一般に，割込み信号を用いる．これらの信号線（資源）は，スレーブデバイスごとに1本ずつ用意され，一意に決定される．しかしながら，スレーブデバイスを実装したモジュールを単純に開発すると，同一デバイスからは，同一信号線，すなわち，同一の資源を利用することとなる．そのため，システムバス上に2つ以上の同じモジュールを利用できない，あるいは，モジュールごとに設定を変更する仕掛け（たとえば，ジャンパーピンにより資源を切り替える方法など）が必要となる．これは，ユーザの利便性からも，また，誤設定による不具合の可能性があることからも問題となる．

そこで，UBAシステムバスでは，これらの資源を活用したバスシステムにおいて，同一モジュールを複数実装するためのリソースルーティング機能を新規に開発した．リソースルーティング機能の基本的なモデルを**図3**に示す．この図では，チップセレクト信号に関するモデルを例示するが，他の信号に関しても同様の構成を有する．各モジュール上でチップセレクト信号をたすきがけにすることが特徴である．

以下，リソースルーティング機能の動作を説明する．CPUモジュール上でのチップセレクト信号1（CS1）は，モジュールAにおいてもCS1である．モジュー

☆ Linuxは Linus B. Torvaldsらが開発したOSです．

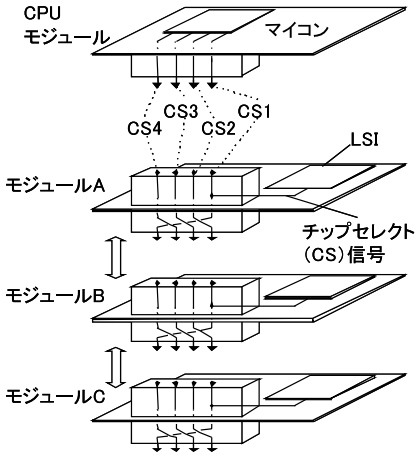

図 3 リソースルーティングモデル
Fig. 3 Resource routing model.

ル A 上の LSI は CS1 に割り付けられている．同様に，モジュール B，C においても，それぞれのモジュール上の LSI は CS1 に割り付けられている．しかしながら，各モジュールの出口部分（下のモジュールにつながるコネクタ）では，これらの CS 信号がたすきがけになっている．すなわち，各モジュールでは，同一の CS1 に割り付けておきながら，CPU 上からは，CS1〜4 に別のデバイスが割り付けられているように制御できる．モジュール A〜C が同一のハードウェアである場合でも，CPU モジュールからは独立に制御可能となる．

リソースルーティングにより，たすきがけで入れ替える本数分だけ同一のモジュール（同一チップセレクトを利用するハードウェア）を同時に利用可能である．本開発では，UBA システムバスにおいて，CS2（16 ビットデバイス用）と CS4（8 ビットデバイス用）に関して，それぞれ，2 本用意した．すなわち，同一のモジュールが同時に 2 つまで，バス上に共存できるようにした．さらに，CF カードに関しては，リソースルーティング機能により，チップセレクトのほかに，カードリセット，カード検出，電圧検出といった資源をルーティングしている．これにより，CF カードモジュールを同時に 2 枚まで実装可能となり，無線 LAN，PHS，DoPa，Bluetooth といったネットワーク系 CF カードだけでなく，CF メモリカードなども組み合わせて利用できるようになった．

2.5 プラグアンドプレイ機能

ユーザの利便性と誤設定による不具合を防ぐために，プラグアンドプレイ（PnP）機能と呼ばれる技術を実装した．PnP を実現するためには，主に下記の機能が必要である．

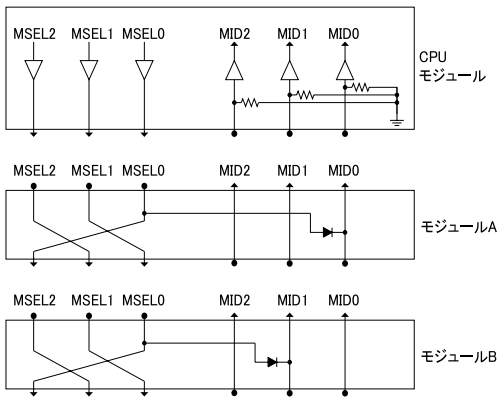

図 4 モジュール識別手段モデル
Fig. 4 Module identification model.

- ハードウェアを識別する機能
- 識別したハードウェアと設定値とを対応付ける機能

一般に，前者の機能は PCI バスや USB などのハードウェアと OS により実現され，後者の機能は OS により実現される．組込み計算機システムにおいては，後述する組込み Linux のほかに，μITRON[☆]，VxWorks[☆☆]など，用途に応じて各種の OS が用いられる．そこで，我々は UBA システムバスにおいて，ハードウェアのドライバ開発や，OS の移植を容易にするため，OS に依存しないハードウェア識別機能を実装することが望ましいと考えた．

接続デバイスを識別するためには，一般に，個々のデバイスに識別のための固有値を付与する必要がある．たとえば PCI バスであれば，コンフィギュレーションレジスタ中のベンダ ID やデバイス ID などが該当する[17]．

しかしながら，組込み向け計算機では，厳密な識別より，低コスト化に重点が置かれる．したがって，PCI デバイスと異なり，本来識別手段を持たないデバイスに対しても，モジュールとしての識別手段を提供する必要がある．そこで，UBA ではモジュールごとに ID を割り当てる簡易モジュール識別手段を提供する．モジュール識別手段の概略回路を**図 4** に示す．図中，MSELx はそれぞれモジュール選択信号である．モジュール選択信号は，前述のリソースルーティング機能を活用して駆動される．動作原理を図 4 に従って説明する．

CPU モジュールは，モジュール選択信号（MSELx）

☆ μITRON は，Micro Industrial TRON の略称です．TRON は，The Realtime Operating system Nucleus の略称です．
☆☆ VxWorks は Wind River Systems 社の商標です．

を通常 Low に駆動している．CPU モジュールは，起動後に，一番近いモジュールを識別するため，MSEL0 信号を High に駆動する．すると，モジュール A の MSEL0 が High に駆動され，その結果 MID0 のみ High が出力される．CPU モジュールは，MID[2:0] を観測しており，結果として，モジュール ID “001” を得る．そこで，CPU モジュールは，一番近いモジュールをモジュール A であると判定する．同様に，CPU モジュールから 2 番目に近い，または 3 番目に近いモジュールを識別するためには，それぞれ MSEL1 または MSEL2 を High に駆動する．すると，2 番目，3 番目のモジュールからは “010”，“011” が得られ，それぞれ，モジュール B またはモジュール C であると認識できる．なお，図 4 では，実装を簡単にするために MSELx と MIDx をダイオードで結合しているが，たとえば，シリアル ROM の出力を MID に出力するよう構成すれば，より多くの ID をモジュールに持たせることが可能となる．

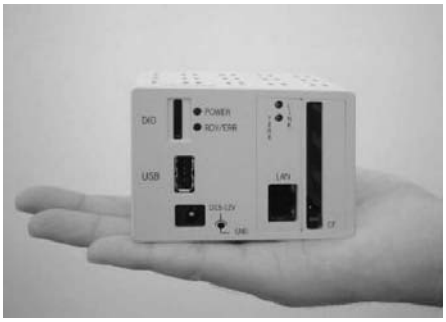

図 5 “Ubiquitous Cube” 概観
Fig. 5 “Ubiquitous Cube”.

3. 実　　装

3.1 ハードウェア構成

前章で述べた UBA に基づき，ブロック型超小型ユビキタスサーバ “Ubiquitous Cube” を試作した．“Ubiquitous Cube” の外観を**図 5** に示す．“Ubiquitous Cube” は，複数のモジュールから構成される組込み計算機であり，図 5 に示す筐体には，CPU モジュールと電源モジュール以外に，I/O モジュールを 2 つまで選択して実装可能である．

図 6 に，“Ubiquitous Cube” を構成する各モジュールの概観を示す．また，各モジュールの特徴を**表 1** に示す．これらのモジュール 1 枚あたりの大きさは，すべて，54mm（W）× 54mm（D）× 17mm（H）である．前章で説明したリソースルーティング機能により，これらのモジュールのうち 3 枚を選択して構成可能である．たとえば，次のような構成が考えられる．

- Ethernet モジュールを 2 枚実装し，ルータを構成．
- CF（PHS カード），FPGA（外部センサ）を実装し，屋外データ収集装置を構成．
- CF（メモリカード），CF（無線 LAN カード），USB（カメラ）を実装し，画像監視装置を構成．

このように，必要最小限の部品で構成することが可能であり，各モジュールはクロック駆動によらない非同期のバスで接続されるため，全体として低消費電力のシステムを実現している．

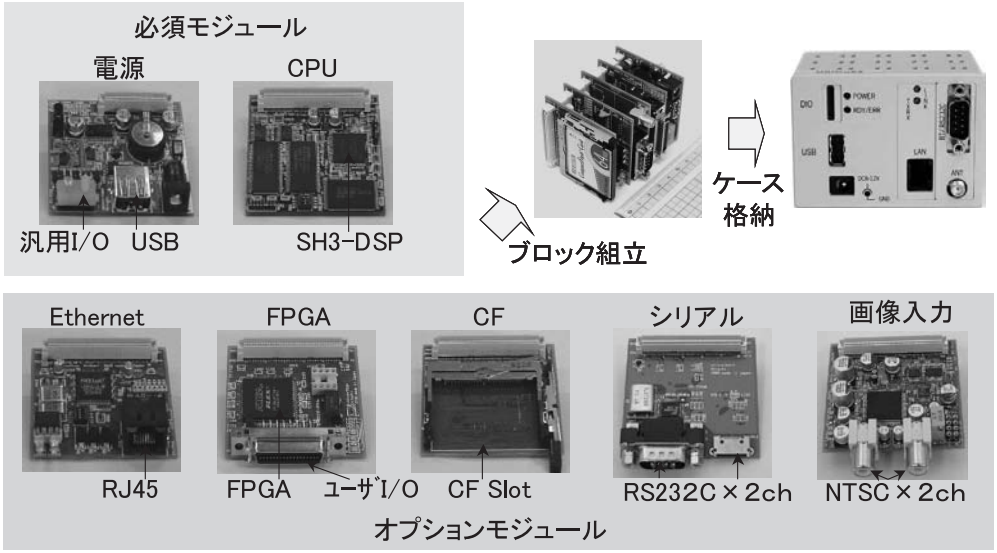

図 6 各種モジュール概観
Fig. 6 I/O modules.

表 1 構成モジュール一覧
Table 1 List of UBA modules.

モジュール	選択	特徴
電源	必須	SH，CF 用電源を生成．バックアップ電池，USB，汎用入出力機能搭載．
CPU	必須	SH3-DSP（SH7727）144 MHz，RAM（32 MB），ROM（16 MB），RTC（Real Time Clock）搭載．
Ethernet	オプション	Ethernet MAC/PHY チップ搭載（10Base-T/100Base-TX）．Ethernet 給電可能．
シリアル	オプション	UART（RS232C 準拠）2 チャンネル搭載．最大転送速度 115.2 Kbps．
CF	オプション	コンパクトフラッシュType I/II 準拠．
FPG A	オプション	TTL I/F のバス入出力可能．論理 30K ゲートプログラム可能．
画像入力	オプション	NTSC 入力 2 チャンネル，JPEG コーデック搭載．

“Ubiquitous Cube” は，設置場所に応じて，『任意のネットワークインフラ』を用いて，『任意の装置』をインターネット接続することを狙っている．そこで，デバイスドライバ・アプリケーションの作成を容易にするために，CPU モジュールとして SH3-DSP（SH7727）[18]，OS として組込み Linux である MontaVistaLinux2.1 [19]を採用した[20),21)]．しかしながら UBA は SH3-DSP などの SuperH アーキテクチャにとらわれることなく，その他任意のマイコンを CPU として採用可能である．たとえば，マイコンとして H8 シリーズ[22)]を採用することで，より低消費電力な組込み計算機システムを構築できる．

また，CF モジュール選択時には，市販の各種通信カードを使用することにより，無線 LAN，PHS，DoPa，Bluetooth など豊富な通信インタフェースを利用することが可能である．

図 7 に，“Ubiquitous Cube” における SuperH プロセッサのアドレスマップを示す．前述のリソースルーティング機能と本アドレスマッピングにより，I/O モジュールの順番を気にせずに使用することができる．たとえば，Ethernet モジュールを使用している場合には，リソースルーティングの機能により，CPU モジュールに近いほうの Ethernet モジュールが，エリア 2 の 16 ビットデバイス 0（アドレス 0x08000000）に自動的にマッピングされる．

以上のように，“Ubiquitous Cube” のハードウェアは，超小型，低消費電力，かつ，ファンレスといった特徴を有し，I/O，通信機能を豊富に用いることができるため，遠隔監視用の組込み計算機に最適な構成となっている．

3.2 ソフトウェア構成

“Ubiquitous Cube” のソフトウェア構成を**図 8** に示す．OS（Operating System）として，組込み Linux を採用することにより，Linux の豊富な通信プロトコルやアプリケーションをそのまま利用可能とし，デバイスドライバに関しても既存ドライバの基本構造を変

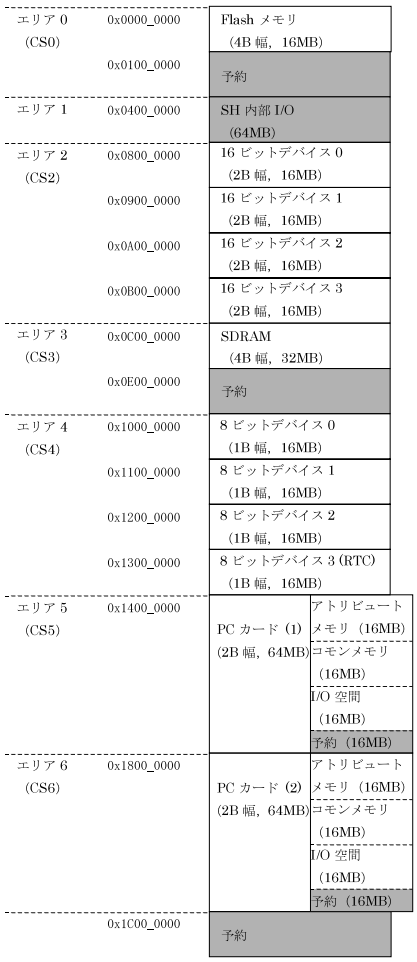

図 7 SuperH プロセッサアドレスマップ
Fig. 7 Address mapping of SuperH processor.

えずに流用することができる．また，前述のプラグアンドプレイ機能により，検出したモジュールに応じて必要なドライバを自動的にロードすることができるため，ユーザは “Ubiquitous Cube” の構成を変更しても，ドライバを意識せずに使用することができる．モニタプログラム，ブートローダ，Linux カーネル，および，Linux ファイルシステムイメージは図 7 におけ

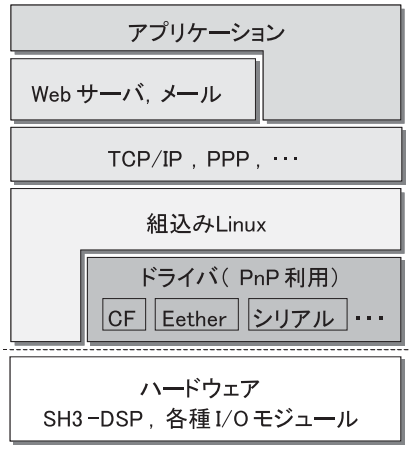

図 8 ソフトウェア構成
Fig. 8 Software structure.

るエリア 0 の Flash メモリ内に格納されている．OS の起動は以下のような手順で行われる．

(1) ブートローダは Linux カーネルをエリア 3 の SDRAM 内に展開する．
(2) SDRAM の一部を RAM ディスクとして確保する．
(3) RAM ディスク上にファイルシステムを展開する．
(4) プラグアンドプレイ機能により必要なドライバをロードする．
(5) 各種デーモンプログラム・アプリケーションを起動する．

4. 性能評価

我々は，“Ubiquitous Cube” を，主に，フィールド機器の遠隔監視を目的として開発した．そこで，遠隔監視システムとして実運用する際に，特に重要なネットワーク性能に関する評価を行った．性能測定の手順を以下に示す．

(1) 電源モジュール＋ CPU モジュール＋ Ethernet モジュール＋ CF モジュール（計 4 モジュール）で “Ubiquitous Cube” を構成する．
(2) CF モジュールにメモリカードを挿入して動作させる．
(3) “Ubiquitous Cube” を LAN に接続し，デスクトップ PC との間で FTP によりファイルの送受信を行う．
(4) RAM ディスク（SDRAM）上のファイルを送受信し，通信の実効速度を計測する．(システムバスの負荷低)
(5) 通信処理とは別にバックグラウンドで，メモリカードに対するデータアクセスを実行している最中に，(4) の計測を行う．(システムバスの負荷高)

表 2 性能評価
Table 2 Evaluation of performance.

通信方法	通信方向	バス負荷	実効速度（bps）
Ethernet	送信/受信	低	約 16 M
（100 Base-TX）	送信/受信	高	約 13 M
無線 LAN	送信/受信	低	約 5 M
（802.11b/11 Mbps）	送信/受信	高	約 4.5 M
PHS	送信	低/高	約 40 K
（128 k パケット方式）	受信		約 70 K
DoPa	受信	低/高	約 10 K
（受信 28.8 kbps）			

別の性能評価として，各種無線通信を利用して以下のような測定も行った．

(1) 電源モジュール＋ CPU モジュール＋ CF モジュール＋ CF モジュール（計 4 モジュール）で “Ubiquitous Cube” を構成する．
(2) 一方の CF カードに各種通信カード（無線 LAN, PHS, DoPa）を搭載し，もう一方にメモリカードを挿入して動作させる．
(3) 通信カードによりネットワークに接続する．無線 LAN の場合は，無線 LAN 搭載 PC と直接データ送受信を行う．PHS と DoPa の場合には，公衆網を経由してデスクトップ PC と接続する．
(4) Ethernet と同様にして，システムバスの負荷が低い場合と高い場合について，それぞれの通信方式における実効速度を測定する．

ここで，“Ubiquitous Cube” のシステムバスの動作周波数は 48 MHz である．CF の I/O アクセスは 16 サイクルで 2 B のデータを読み書きするため，ピーク時には約 6 MB/s のデータアクセスとなる．Ethernet は 5 サイクルで 2 B のデータを読み書きするため，ピーク時には約 19 MB/s のデータアクセスとなる．したがって，Ethernet 通信の測定では，高負荷時には Ethernet アクセスと CF アクセス両方の負荷がシステムバスにかかる．また，無線通信の測定では，高負荷時には CF アクセス 2 つ分の負荷がシステムバスにかかることになる．

実験の結果を**表 2** にまとめる．Ethernet による通信では，システムバスに高負荷をかけると，実効速度が 20%ほど低下している．同様に，無線 LAN による通信では，システムバスに高負荷をかけると，実効速度が 10%ほど低下している．PHS や DoPa により通

信を行った場合には，システムバスに高負荷をかけた場合でも性能の低下はほとんど見られなかった．

以上のことから，Ethernet や無線 LAN などのように，通信速度が高速になるほど，システムバスへの負荷が影響し，性能の低下が起きることが分かる．逆に PHS や DoPa などのように低速で通信を行う場合には，システムバスに高負荷をかけても性能の低下はほとんど見られない．したがって，今後は高速通信であっても性能の低下を引き起こさないようなシステム設計が課題となる．しかしながら，表 2 で得られた Ethernet や無線 LAN の実効速度は，実運用において，たとえば，監視データ中でデータ量が最も多いと考えられる画像データ（圧縮時通常数十 K バイト）を送信することなどを想定した場合に十分な性能を有する．したがって，Ethernet，および，各種の無線通信を利用した遠隔監視において，本開発の “Ubiquitous Cube” を有効に活用できることを確認した．

以上のように，“Ubiquitous Cube” では，UBA を利用し，データアクセス中に通信処理を実行するといった場合でも，十分な通信速度を保持したままデータ処理が可能である．したがって，遠隔監視システムにおいて，本研究で提案するアーキテクチャが実運用上有効であることを実証できたと考える．

5. おわりに

様々なフィールド情報を活用する遠隔監視サービスを容易に実現するための情報端末として，ブロック型超小型ユビキタスサーバ “Ubiquitous Cube” を開発した．そのためのアーキテクチャとして，柔軟かつ容易に入出力モジュールを構築する Universal Block Architecture（UBA）を提案した．UBA により，必要な機能のみを組み合わせ，設置場所に応じて最適な組込み計算機を構成できる．

“Ubiquitous Cube” の性能評価を行ったところ，UBA を利用しても，遠隔監視に必要十分な通信速度が維持できることを確認した．

したがって，従来では利用環境（配線，スペース，使用可能な通信インフラなど）の面から設置が困難であった場所にでも遠隔監視用の端末を設置することが可能となる．また，使用可能なインタフェースの違いなどで，これまで柔軟に対応できなかった装置や設備についても，UBA の利用により，端末のカスタマイズが安価かつ，容易に行えるようになる．このように，本研究により，遠隔監視サービスの適用分野を広げることができる．

参考文献

1) 齊藤雅彦：ユビキタス情報社会とユビキタスコントローラ，電子情報通信学会誌，Vol.85, No.7, pp.519–521 (2002).
2) 森川博之，青山友紀，南　正輝：ユビキタスネットワーキングへの道，情報処理，Vol.43, No.6, pp.631–638 (2002).
3) Weiser, M.: Some Computer Science Issues in Ubiquitous Computing, *Comm. ACM*, Vol.36, No.7, pp.75–84 (1993).
4) 齊藤雅彦：ユビキタスコントローラ（IP/Web 技術活用超小型コントローラ）の開発，電気学会研究会資料，金属産業研究会 (2002).
5) 中島久雄：ユビキタス・ネットワーク時代のマーケティング戦略，知的資産創造，Vol.10, No.1, pp.84–95, 野村総合研究所 (2002).
6) 名雲俊忠：ユビキタス・ネットワーク時代の革新的事業モデル，知的資産創造，Vol.10, No.1, pp.70–83, 野村総合研究所 (2002).
7) Estrin, D.: Connecting the Physical World with Pervasive Networks, *IEEE Pervasive Computing*, Vol.1, No.1, pp.59–69 (2002).
8) 高橋史忠，蓬田宏樹：感じるネットがあったなら…，日経エレクトロニクス，No.826, pp.100–109, 日経 BP 社 (2002).
9) PCI Special Interest Group: *PCI Local Bus Specification, Revision 2.3* (2002).
10) 阿部晋樹，上野伸二：PC サーバの入出力インタフェース動向，情報処理，Vol.44, No.4, pp.417–721 (2002).
11) PC/104 Embedded Consortium: *PC/104 Specification, Version 2.4* (2001).
12) Microsoft: *Plug and Play ISA Specification, Version 1.0a* (1994).
13) PCMCIA, JEITA: PC Card Standard, Release 8, Vol.1–11 (2001).
14) Anderson, D.: *PCMCIA System Architecture 16-Bit PC Cards Second Edition*, Addison-Wesley Publishing Company (1995).
15) CompactFlash Association: *CF+ and CompactFlash Specification, Revision 1.4* (1999).
16) 西島泰介，大中邦彦：メモリカード活用技術の徹底研究．
17) インタフェース編集部：OPEN DESIGN No.7 PCI バスの詳細と応用へのステップ，CQ 出版社 (1995).
18) 日立製作所：SH7727 ハードウェアマニュアル，第 3 版 (2002).
19) MontaVista Software: *MontaVista Linux Professional Edition Reference Guide, Version 2.1* (2002).
20) 中島達夫：組み込みシステム用標準 OS としての Linux，情報処理，Vol.43, No.2, pp.148–153

(2002).
21) 田丸喜一郎：豊富な技術者を後ろ盾に組み込みOSの主流はLinuxへ，日経エレクトロニクス，No.824, pp.140–151, 日経BP社 (2002).
22) 日立製作所：H8/3006，H8/3007ハードウェアマニュアル，第3版 (1997).

(平成15年 7 月25日受付)
(平成15年12月 7 日採録)

松本 典剛（正会員）

1999年早稲田大学理工学部機械工学科卒業．2001年早稲田大学大学院理工学研究科修士課程修了．2001年（株）日立製作所日立研究所入社．産業用コントローラ，組込みLinuxの実装に関する研究に従事．電子情報通信学会会員．

遠藤 浩通（正会員）

1997年筑波大学工学システム学類卒業．1999年筑波大学大学院理工学研究科修士課程修了．1999年（株）日立製作所日立研究所入社．分散・組み込みシステム向けOSの実装に関する研究に従事．ソフトウェア科学会会員．

山田 勉

1992年京都大学工学部卒業．1994年京都大学大学院工学研究科修士課程修了．1994年（株）日立製作所日立研究所入社．産業用コントローラ向けアーキテクチャ，高信頼LSIの実装に関する研究に従事．IEEE，電子情報通信学会各会員．

中三川哲明（正会員）

1983 年宇都宮大学工学部卒業．1985 年宇都宮大学大学院工学研究科修士課程修了．1985年（株）日立製作所日立研究所入社．高信頼サーバ，産業用コントローラの研究に従事．現在は鉄道向けアプリケーションの研究に従事．IEEE，電子情報通信学会各会員．

齊藤 雅彦（正会員）

1986年京都大学工学部卒業．1988年京都大学大学院工学研究科修士課程修了．1988年（株）日立製作所日立研究所入社．並列/分散処理システム，リアルタイムOSの実装に関する研究に従事．IEEE，電子情報通信学会各会員．