システムＬＳＩ設計技術　109－4
(2003. 3. 10)

組込みシステム用 イーサネットインタフェースに関する研究

阿部　司†　吉村　斎†　稲川　清†

†苫小牧工業高等専門学校情報工学科 〒059-1275 北海道苫小牧市錦岡 443

E-mail: †(abe, yosi, inagawa)@jo.tomakomai-ct.ac.jp

あらまし　組込みシステムは，汎用計算機にはない組込みシステム独自の特性を持っている．従って，組込みシステム用の TCP/IP プロトコルスタックとネットワークインタフェースを実装するには，これらの特性を十分に考慮しなければならない．また，本研究で使用した ITRON TCP/IP API 仕様にも，プロトコルスタックに求められる性質がある．その中で，本研究では，メモリ容量等の厳しいリソース制約への対応を目標として，組込みシステム用 TCP/IP プロトコルスタックとイーサネットインタフェースの実装を行った．また，WWW サーバ等の各種サーバとクライアントプログラムを実装して，応用プログラムの実現性を確認した．

キーワード　組込みシステム，TCP/IP，ITRON，イーサネット

Study of Ethernet Interface for Embedded Systems

Tsukasa ABE† Hitoshi YOSHIMURA† and Kiyoshi INAGAWA†

† Department of Computer Science and Engineering, Tomakomai National College of Technology

443 Nishikioka, Tomakomai-shi, Hokkaido, 059-1275 Japan

E-mail: †(abe, yosi, inagawa)@jo.tomakomai-ct.ac.jp

Abstract　An embedded system has the original characteristic which is not in a general-purpose computer. Therefore, in order to implement the TCP/IP protocol stack and network interface for embedded systems, one must fully take these characteristics into consideration. Also, in the ITRON TCP/IP API specification used by this study, there are characters for which a protocol stack is asked. In this study, we implemented the TCP/IP protocol stack for embedded systems and an Ethernet interface for achieving the adaptability about the resource restrictions with severe memory capacity, etc. Various servers and client programs, such as a WWW server, were implemented, and the implementability of an application program was checked.

Keyword　Embedded System，TCP/IP，ITRON，Ethernet

1. はじめに

常時接続の普及などにより，生活の中で必要不可欠な媒体になりつつあるインターネットに，組込みシステムを接続するためは TCP/IP プロトコルスタックとネットワークインタフェースが必要である．

一方，組込みシステムの特性，リアルタイム OS（以下 RTOS）の制約，応用プログラムとのインタフェースとなる ITRON TCP/IP API 仕様[1]で求められている性質も考慮しなければならない．このため，本研究では以下の事項を考慮し，TCP/IP プロトコルスタックと，ネットワークインタフェースとしてイーサネットインタフェースの実装を行った．

(1) 組込みシステムには，メモリ容量，重量，サイズ及び消費電力等の厳しいリソースの制約という汎用計算機にはない組込みシステム独自の特性がある．本研究では，特に，メモリ容量の制約に対する対応を重点的に行った．

(2) ITRON TCP/IP API 仕様では，組込みシステム用のプロトコルスタックに求められる性質が挙げられている．本研究では，最小のコピー回数と動的メモリ管理の排除を優先して行った．

(3) RTOS では，実時間性の制約から，ハードウェア割込みに費やす時間をできるだけ短縮しなければならない．本研究においても，ネット

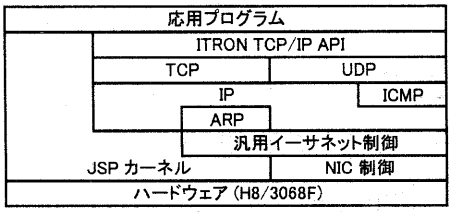


図 1 TCP/IP プロトコルスタックの構成

ワークインタフェースは，この制約を考慮して実装を行った．

2. TCP/IP プロトコルスタックの概要

組込みシステムに使用されているプロセッサの規模にも幅があり，メモリ容量が数 K バイトの 8 ビット CPU から，汎用プロセッサ並みの仮想記憶システムを備えた 32 ビットや 64 ビットのプロセッサまで存在する.また,あまりにリソース制約の厳しいプロセッサに，TCP/IP プロトコルスタックを実装することは現実的ではない．本研究では，以下に示す規模の組込みシステムを対象としている．

(1) 必要なメモリ量は，ROM が 128K バイト，RAM が 32K バイト程度．
(2) TCP/IP の各ヘッダは，32 ビット単位で処理すると効率が良いため，C 言語の int が 32 ビットで，レジスタと ALU も 32 ビットが望ましい．

本研究では，以上のような規模の組込みシステムに求められる通信機能を，次のように設定した．

(1) 応用層の基本プロトコルは HTTP．
(2) 応用層とのインタフェースは ITRON TCP/IP API.
(3) TCP のオプションは MSS の通知のみ．
(4) IP での，ルーティングと分割・再構成は行わない．またオプションは未実装．
(5) ネットワークでの配置は，単一リンクの終端ノード．
(6) データリンク・物理層は，単一のネットワークインタフェース．

3. リアルタイム OS とターゲットプロセッサ

リアルタイム OS (以下 RTOS)として豊橋技術科学大学の組込みリアルタイムシステム研究室を中心とした TOPPERS プロジェクト[2]で開発された JSP カーネルを使用した．JSP カーネルは μITRON 4.0 仕様準拠[3]でスタンダードプロファイルを実装している．

実装ターゲットプロセッサは，日立製 H8/300H シリーズの H8/3068F[4] または H8/3069F[5] である．H8/300H シリーズは，16 ビット CPU であるが，レジスタと ALU が 32 ビットであり，本研究で対象とした組込みシステムの規模に合致している．H8/3068F と H8/3069F の内蔵 ROM は,それぞれ 384K バイトと 512K バイトで容量に余裕があるが，内蔵 RAM は 16K バイトで小容量なため外部に RAM を増設している．

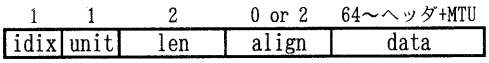


図2 net_buf 構造体

イーサネット制御 LSI は，NE2000 互換の RealTek RTL8019AS[6]を使用している．

TCP/IP プロトコルスタック及びイーサネットインタフェースは，FreeBSD バージョン 3.4[7]をベースに実装を行った．

4. TCP/IP プロトコルスタックの実装

4.1. TCP/IP プロトコルスタックの構成

本研究における TCP/IP プロトコルスタックの構成を**図 1** に示す.本研究おける TCP/IP プロトコルスタックは RTOS 内部に組込むのではなく，RTOS と応用プログラムの中間のミドルウェアとして位置付けており，TCP/IP プロトコルスタック内でも，RTOS が提供する各種サービスを利用して動作している．以下に，各構成要素の概要を示す．

(1) ITRON TCP/IP API では，機能レベルでは基本機能を実装しており，組込みシステム特有のノンブロッキングコール，省コピー API，コールバック関数も実装している．ITRON TCP/IP API と TCP のインタフェースは TCP 受付口 ID，TCP 通信端点 ID により行う.また,UDP は UDP 通信端点 ID である．この受付口と通信端点は，FreeBSD におけるソケット，プロトコル制御ブロック，TCP 制御ブロックに相当する構造体である．
(2) IP では，ネットワークでの配置は終端ノードのため，ルーティングは実装していない．また，分割・再構成も実装していない．これは，TCP に関しては MSS オプションにより分割は回避可能であり，UDP に関しても，512 オクテット以上のデータグラムを送信する応用プログラムは稀であり，未実装でも問題がないと判断した．
(3) ICMP では，単一リンクの終端ノードに必要と考えられる ICMP エコー機能，配送エラーの受信と報告機能のみ実装している．
(4) IP からイーサネットへの出力前に，経路表から，配送すべきノードの IP アドレスを決定する．ベースとなった FreeBSD では，ARP も含

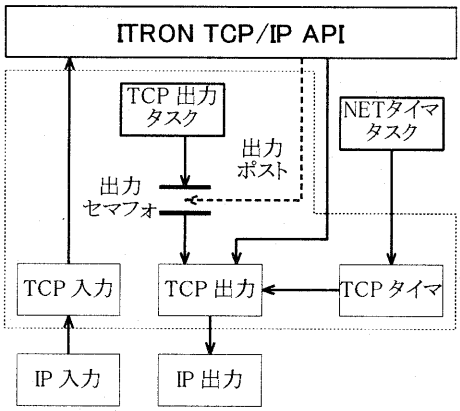


図 3 TCP の制御とデータの流れ

んだ大規模なルーティング表を検索することで IP アドレスを決定しているが，本研究での実装では，経路表は，エントリがターゲットノードの IP アドレス，サブネットマスクとルータから構成される単純で静的な表としている．

(5) ARP では，経路表から決定したノードに IP データグラムを配送するために必要な MAC アドレス解決と，他のノードから要求される MAC アドレス解決要求に対する応答のみ実装している．

4.2. ネットワークバッファ(net_buf)

本研究でベースとした FreeBSD に実装されているプロトコルスタックでは，プロトコルスタック内における各層間のデータの受け渡しに使用されるメモリブロックは，mbuf と呼ばれる固定的なメモリブロックによるリスト構造で実装している．この mbuf は，プロトコルスタック内での処理中に動的メモリ操作が行われる等の問題がある[8]．それに対し，本研究における実装では，**図 2** に示すように，固定長メモリプールから割り当てる net_buf を用いてメモリブロックを実装している．

入力においては「NIC 入力」で，NIC の受信バッファに記憶されているフレーム長から net_buf を割当てる．出力においては，「TCP 出力」で，TCP 通信端点から算出したサイズの net_buf を割当てる．入出力いずれも，割当て前にデータサイズは確定しているため，これ以降，net_buf の動的メモリ操作は行わない．

4.3. TCP の実装

図 3 に TCP の制御とデータの流れを示す．ベースした FreeBSD の実装では，「IP 入力」からの「TCP 入力」の呼出しと，応用プログラムの API であるソケットインタフェースからの「TCP 出力」の呼出しは，プロトコルスイッチ構造体を経由して行っている．

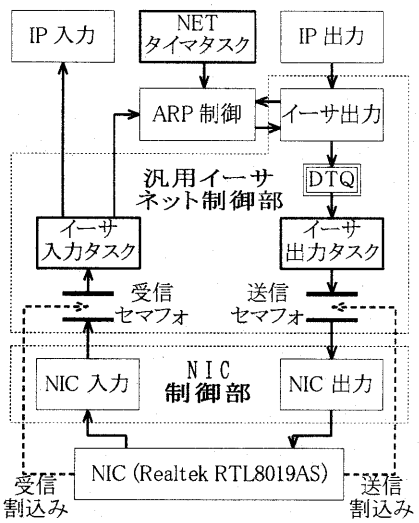


図 4 イーサネットの制御とデータの流れ

それに対して，本研究での実装では，「IP 入力」から「TCP 入力」の呼出しは，IP ヘッダのプロトコルフィールドを参照して，直接関数を呼出し，関数呼出しのオーバヘッドを削減している．

ITRON TCP/IP API からは，直接「TCP 出力」を呼出す方式と，TCP 通信端点に出力指示フラグをポストし，「TCP 出力タスク」を経由して間接的に「TCP 出力」を呼出す方式を実装した．

ITRON TCP/IP API では，組込みシステムを対象としているため，プロトコルスタック内で実行がブロックされる可能性のある API では，タイムアウトを指定できる．タイムアウトには，ミリ秒単位のタイムアウト以外に特殊な指定が可能で，その中にプロトコルスタック内でタスクの実行をブロックしない「ノンブロッキング」指定がある．

本研究での実装において，ミリ秒単位のタイムアウトが指定された場合は，直接「TCP 出力」を呼出すが，「ノンブロッキング」が指定された場合，TCP 通信端点に出力指示フラグをポストだけして，API を呼出した応用タスクに戻る．従って，この応用タスクは「TCP 出力タスク」と平行して動作を継続することができる．

「ノンブロッキング」指定による API の終了は，TCP 通信端点に指定されたコールバック関数を経由して応用タスクに通知され，応用タスクとの同期は，応用タスクに任されている．

本研究での実装において，「ノンブロッキング」指定の情報は，送信系 API と受信系 API で個別に管理しているため，API の終了を待たずに同系の呼出すことはできないが，別系の API を呼出すことは可能で

ある．

TCP のオプションは，MSS オプションのみ実装しており，FreeBSD の TCP に実装されている性能向上のための拡張機能は実装していない．しかし，BSD の基本制御アルゴリズムはそのまま実装しており，基本機能としては十分である．

5. イーサネットインタフェースの実装

5.1. イーサネットインタフェースの構成

イーサネットインタフェースの構成を図 4 に示す．イーサネットインタフェースは，NIC に依存しない汎用イーサネット制御部と，NIC に依存する NIC 制御部から構成している．この構成により，新たな NIC を組込むためには，NIC 制御部を実装するだけでよい．

「IP 出力」は，net_buf にイーサネットヘッダと IP データグラムを設定して，「イーサ出力」を呼出し，データキュー(DTQ)に投入する．このとき，DTQ に空きがない場合，API で指定されたタイムアウトに従って待ち状態に入る．

「イーサ出力タスク」は DTQ から net_buf を取り出し，「送信セマフォ」で，送信完了割込みの待合わせ後，「NIC 出力」を呼出し，NIC の送信バッファにフレームを書込み，イーサネットフレームの送信を起動する．送信完了割込みにより次のフレームの送信を起動する．

FreeBSD における実装では，「NIC 入力」に相当する部分は，NIC からのフレームを入力して IP 層のキューに投入するまで割込みコンテキストで実行する．しかし，RTOS のミドルウェアとしての実装では，割込みコンテキスでの実行時間をできるだけ短縮することが重要であるため，本研究における実装では，割込みコンテキスト内では，「受信セマフォ」により「イーサ入力タスク」と同期を取り，起床された「イーサ入力タスク」は，「NIC 入力」を呼出し，NIC の受信バッファから net_buf にフレームを入力する．

5.2. 汎用イーサネット制御関数の呼び出し

FreeBSD における実装では，IP 層からの汎用イーサネット制御に相当する関数の呼び出しは，ifnet 構造体により行われており，多数のネットワークインタフェースの中から出力経路に一致するネットワークインタフェースを選択するため，IP 層からの出力では，経路選択関数により間接的にネットワークインタフェースを選択するように実装している．

それに対し，本研究における実装では，単一リンクの終端ノードで，単一ネットワークインタフェースのみ想定しているため，IP 以上の階層から汎用イーサネット制御の各関数を直接呼出すことが可能であり，関数呼出しのオーバヘッドを削減することができる．ただし，ネットワークインタフェースを抽象化し，コンフィギュレーションによりネットワークインタフェースを選択可能なように，ネットワークインタフェース毎に，汎用イーサネット制御関数を 11 種類のマクロにより定義している．

例えば，「IP 出力」では，ネットワークインタフェースへの出力を以下のように記述している．

```
IF_SET_PROTO(output, IF_PROTO_IP);
error = IF_OUTPUT(output, gw, tmout);
```

この記述が，ネットワークインタフェース毎に定義されているマクロによりネットワークインタフェースに依存した関数や定数に展開される．上記の例は，イーサネットインタフェースでは以下のように展開される．

(1) IF_SET_PROTO はネットワークインタフェースのヘッダに上位層のプロトコルを設定する関数で，

```
(GET_ETHER_HDR(output)->type=
                    htons(IF_PROTOIP))
```

に展開される．

(2) IF_PROTO_IP は IP を表すネットワークインタフェースのプロトコル番号で，

```
ETHER_TYPE_IP
```

に展開される．

(3) IF_OUTPUT はネットワークインタフェースの出力関数で，

```
ether_output(output, gw, tmout);
```

に展開される．

5.3. NIC 制御関数の呼び出し

汎用イーサネット制御同様に，本研究における実装では，単一リンクの終端ノードのみ想定しているため，NIC 制御も，汎用イーサネット制御と同様に，汎用イーサネット制御から NIC 制御の各関数を直接呼び出すことが可能であり，NIC 毎に，NIC 制御関数を 8 種類のマクロにより定義している．

例えば，「イーサ出力タスク」では，NIC への送信を以下のように記述している．

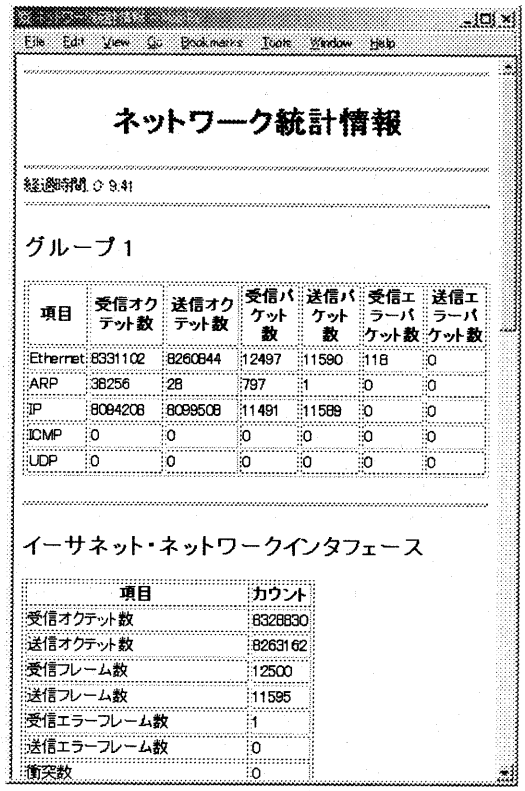
```
IF_ETHER_START(sc, output);
```

このマクロは，NE2000 互換 NIC では，

```
ed_start(sc, output);
```

に展開される．

NIC を制御するために必要となる変数は，「ソフトウェア制御構造体」にまとめて定義している．この構造体には，NIC の種類に依存しないメンバと，依存するメンバを分けて定義している．依存しないメンバは



ネットワーク統計情報

経過時間 0 9:41

グループ1

項目	受信オクテット数	送信オクテット数	受信パケット数	送信パケット数	受信エラーパケット数	送信エラーパケット数
Ethernet	8331102	8260844	12497	11590	118	0
ARP	38256	28	797	1	0	0
IP	8094208	8099508	11491	11589	0	0
ICMP	0	0	0	0	0	0
UDP	0	0	0	0	0	0

イーサネット・ネットワークインタフェース

項目	カウント
受信オクテット数	8328830
送信オクテット数	8263162
受信フレーム数	12500
送信フレーム数	11595
受信エラーフレーム数	1
送信エラーフレーム数	0
衝突数	0

図 5 ネットワーク統計情報ページ

NIC 制御の外部で参照する変数であり，依存しないメンバとしては，「イーサ入力タスク」，「イーサ出力タスク」と同期を取るために使用されているセマフォの ID などを定義している．依存するメンバは，NIC 制御のみで参照する変数であり，依存するメンバとしては，NIC のレジスタのアドレスやバッファのサイズなどを定義している．

6. TCP/IP プロトコルスタックの評価

6.1. 応用プログラムの実装

実装した TCP/IP プロトコルスタックとイーサネットインタフェースを評価するために WWW サーバを実装した．実装した WWW サーバがサポートする HTTP のバージョンは 1.0 (RFC 1945[9])で，要求・応答毎に接続・切断を行う方式である．転送可能なページは 2 ページで，トップページの転送容量は約 1,100 オクテット，それからリンクし，動的に生成されるネットワーク統計情報ページの転送容量は約 4,000 オクテットである．**図 5 にブラウザで表示したネットワーク統計情報のページを示す．ネットワーク統計情報は** TCP/IP プロトコルスタックに組込まれている各プロトコルのカウンタの値を出力している．

表 1 WWW サーバのメモリ必要量

	RAM	ROM	計	比 率
WWW	10,932	10,460	21,392	19.4%
TCP	1,056	27,906	28,962	26.2%
IP	48	920	968	0.9%
ICMP	20	1,468	1,488	1.3%
ARP	160	2,270	2,430	2.2%
inet 共通	0	1,808	1,808	1.6%
汎用ネット	1,460	1,994	3,454	3.1%
net_buf	5,504	408	5,912	5.3%
イーサネット	2,146	4,112	6,258	5.7%
カーネル	5,748	32,128	37,876	34.3%
合 計	27,074	83,474	110,548	

WWW サーバタスクは 2 個実装している．ITRON TCP/IP API 仕様から，コネクションの切断が完了するまで，WWW サーバタスクは，ITRON TCP/IP API の tcp_cls_cep で，標準的に約 1 分間の待ち状態に入る．このため，応答性を向上するために複数のタスクを実装した．

評価のため，WWW サーバ以外にも，ECHO サーバ等を実装し，各種応用プログラムを実現可能なことを確認している．

6.2. メモリ必要量の評価

表 1 に WWW サーバのメモリ必要量を示す．「2. TCP/IP プロトコルスタックの概要」で示した，対象とする組込みシステムの規模に十分余裕を持って収まっていることがわかる．

WWW サーバは，前節で述べたように 2 個のタスクから構成され，WWW サーバタスク 1 個当たり，スタック 2K バイト，送受信バッファそれぞれ 1K バイト，計 4K バイトの RAM が必要である．従って，ROM 部分はプログラムと固定データのため削減はできないが，WWW サーバタスクを減らすことで，RAM の必要量を削減することも可能である．

TCP/IP プロトコルスタック（表では TCP から inet 共通），汎用ネット，イーサネット及びカーネルは WWW サーバの機能向上や，net_buf に割当てるメモリ容量の見直しには影響しないメモリ必要量であり，ほぼ固定分とみなすことができる．

net_buf 管理の RAM 部分の大部分は，固定メモリプールである．この net_buf の種類とその個数はコンフィギュレーションで設定することができる．従って，通信量やデータ長等の通信状況によりチューニング可能である．

対象とした組込みシステムの規模に対して，本実装によるメモリ必要量には，まだ十分に余裕がある．net_buf 管理やコンフィギュレーションに影響されるが，RAM の約 5K バイト，ROM の約 37K バイトを WWW サーバタスクに使用することができるため，さらに機能向上やページの追加を行うことが可能である．

7. おわりに

本研究では，組込みシステムに求められる通信機能を検討した上で，ITRON TCP/IP API仕様のTCP/IPプロトコルスタックと，それに組込んで使用するイーサネットインタフェースの実装に関するの研究を行った．評価で述べたように対象とした組込みシステムの制約条件として設定したリソース規模に収まることを示した．また，本論文で詳細は示さなかったがWWWサーバ含む各種クライアントとサーバを実装し，応用プログラムの実現性も確認している．

TCP/IPプロトコルスタック及びイーサネットインタフェースの実装における研究課題と，今後の研究課題について以下に示す．

(1) 本研究で示したイーサネットインタフェースの他に，組込みシステム用のプロセッサに内蔵されているシリアルインタフェースを利用し，PPP[10]も実装している．更に今後はbluetoothや無線LANなど多様なネットワークインタフェースの実装を予定している．

(2) 本研究では，データリンク・物理層は単一のネットワークインタフェースの実装に限定することで各種の簡単化を実現できたが，複数のノードと直接通信できることが必要な応用もある．また，現在未使用であるが，net_bufのunitフィールドはインタフェース番号を設定することを想定しており，今後は，これを活用した複数のネットワークインタフェースの実装に関する研究を行う．ただし，イーサネットディバイスドライバの変更も必要であり，これらの変更によるメモリ必要量の増加が妥当かどうかの検討も今後の課題である．

(3) これからのインターネットではIPv6への対応が欠かせない．最近のメモリの低廉化により，IPv6の実装は多くのメモリを必要とすることを容認している．しかし，組込みシステムでは厳しいリソース制約が依然としてあると考えられ，非PC系ディジタル機器への適用に向けたIPv6最小要求仕様の検討[11]が行われている．本研究の実装で行ったように，組込みシステムに求められるIPv6の通信機能を検討し，厳しいリソース制約に対応した実装の研究を行う予定である．

謝辞 本研究は，(財) みやぎ産業振興機構が実施している「組込みシステム・オープンプラットフォームの構築とその実用化開発」の受託研究により行われた．この場を借りて，各関係機関の皆様に感謝する．

文　献

[1] Embedded TCP/IP 技術委員会／ITRON 専門委員会，ITRON TCP/IP AIP 仕様 1.00.01，高田広章編，トロン協会，東京，1998.

[2] TOPPERS/JSP プロジェクトホームページ：http://www.ertl.ics.tut.ac.jp/TOPPERS/

[3] μITRON4.0 仕様 4.01.00，高田広章編，トロン協会，東京，2001.

[4] (株) 日立小平セミコン技術ドキュメントグループ，H8/3068F シリーズ H8/3068F-ZTAT™ ハードウェアマニュアル，(株) 日立製作所，東京，2001.

[5] (株) 日立小平セミコン技術ドキュメントグループ，H8/3069F シリーズ H8/3069F-ZTAT™ ハードウェアマニュアル，(株) 日立製作所，東京，2001.

[6] REALTEK SEMI-CONDUCTOR CO., LTD. , RTL8019AS Realtek Full-Duplex Ethernet Controller wih Plug and Play Function (RealPNP) SPECIFICATION: REALTEK SEMI-CONDUCTOR CO., LTD, TAIWAN, 2001.

[7] FreeBSD プロジェクトホームページ：http://www.freebsd.org/

[8] 高田広章，“TCP/IP プロトコルスタックの問題点と ITRON TCP/IP API 仕様”，Interface，Vol.24, pp. 86-97，Oct. 1998.

[9] Berners-Lee, T., Fielding, R. and Frystyk, H.: Hypertext Transfer Protocol – HTTP/1.0, RFC 1945, May 1996.

[10] 阿部司，組込みシステム用ネットワークインタフェースに関する研究，SWEST4 第 4 回組込みシステム技術に関するサマーワークショップ予稿集，pp.77-84，July 2002.

[11] 岡部宣夫，石山政浩，井上淳，筱島雅之，坂根昌一，佐治木次郎，野口敬，宮田宏，“非 PC 系ディジタル機器への適用に向けた IPv6 最小要求仕様の検討”，情報処理，Vol. 42, pp. 920-925，Sep. 2001.