

お客様各位

---

## カタログ等資料中の旧社名の扱いについて

---

2010年4月1日を以ってNECエレクトロニクス株式会社及び株式会社ルネサステクノロジが合併し、両社の全ての事業が当社に承継されております。従いまして、本資料中には旧社名での表記が残っておりますが、当社の資料として有効ですので、ご理解の程宜しくお願ひ申し上げます。

ルネサスエレクトロニクス ホームページ (<http://www.renesas.com>)

2010年4月1日  
ルネサスエレクトロニクス株式会社

【発行】ルネサスエレクトロニクス株式会社 (<http://www.renesas.com>)

【問い合わせ先】 <http://japan.renesas.com/inquiry>

## ご注意書き

1. 本資料に記載されている内容は本資料発行時点のものであり、予告なく変更することがあります。当社製品のご購入およびご使用にあたりましては、事前に当社営業窓口で最新の情報をご確認いただきますとともに、当社ホームページなどを通じて公開される情報に常にご注意ください。
2. 本資料に記載された当社製品および技術情報の使用に関連し発生した第三者の特許権、著作権その他の知的財産権の侵害等に関し、当社は、一切その責任を負いません。当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
3. 当社製品を改造、改変、複製等しないでください。
4. 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。お客様の機器の設計において、回路、ソフトウェアおよびこれらに関連する情報を使用する場合には、お客様の責任において行ってください。これらの使用に起因しお客様または第三者に生じた損害に関し、当社は、一切その責任を負いません。
5. 輸出に際しては、「外国為替及び外国貿易法」その他輸出関連法令を遵守し、かかる法令の定めるところにより必要な手続を行ってください。本資料に記載されている当社製品および技術を大量破壊兵器の開発等の目的、軍事利用の目的その他軍事用途の目的で使用しないでください。また、当社製品および技術を国内外の法令および規則により製造・使用・販売を禁止されている機器に使用することができません。
6. 本資料に記載されている情報は、正確を期すため慎重に作成したのですが、誤りが無いことを保証するものではありません。万一、本資料に記載されている情報の誤りに起因する損害がお客様に生じた場合においても、当社は、一切その責任を負いません。
7. 当社は、当社製品の品質水準を「標準水準」、「高品質水準」および「特定水準」に分類しております。また、各品質水準は、以下に示す用途に製品が使われることを意図しておりますので、当社製品の品質水準をご確認ください。お客様は、当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途に当社製品を使用することができません。また、お客様は、当社の文書による事前の承諾を得ることなく、意図されていない用途に当社製品を使用することができません。当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途または意図されていない用途に当社製品を使用したことによりお客様または第三者に生じた損害等に関し、当社は、一切その責任を負いません。なお、当社製品のデータ・シート、データ・ブック等の資料で特に品質水準の表示がない場合は、標準水準製品であることを表します。  
標準水準： コンピュータ、OA 機器、通信機器、計測機器、AV 機器、家電、工作機械、パーソナル機器、産業用ロボット  
高品質水準： 輸送機器（自動車、電車、船舶等）、交通用信号機器、防災・防犯装置、各種安全装置、生命維持を目的として設計されていない医療機器（厚生労働省定義の管理医療機器に相当）  
特定水準： 航空機器、航空宇宙機器、海底中継機器、原子力制御システム、生命維持のための医療機器（生命維持装置、人体に埋め込み使用するもの、治療行為（患部切り出し等）を行うもの、その他直接人命に影響を与えるもの）（厚生労働省定義の高度管理医療機器に相当）またはシステム等
8. 本資料に記載された当社製品のご使用につき、特に、最大定格、動作電源電圧範囲、放熱特性、実装条件その他諸条件につきましては、当社保証範囲内でご使用ください。当社保証範囲を超えて当社製品をご使用された場合の故障および事故につきましては、当社は、一切その責任を負いません。
9. 当社は、当社製品の品質および信頼性の向上に努めておりますが、半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。また、当社製品は耐放射線設計については行っておりません。当社製品の故障または誤動作が生じた場合も、人身事故、火災事故、社会的損害などを生じさせないようお客様の責任において冗長設計、延焼対策設計、誤動作防止設計等の安全設計およびエージング処理等、機器またはシステムとしての出荷保証をお願いいたします。特に、マイコンソフトウェアは、単独での検証は困難なため、お客様が製造された最終の機器・システムとしての安全検証をお願いいたします。
10. 当社製品の環境適合性等、詳細につきましては製品個別に必ず当社営業窓口までお問合せください。ご使用に際しては、特定の物質の含有・使用を規制する RoHS 指令等、適用される環境関連法令を十分調査のうえ、かかる法令に適合するようご使用ください。お客様がかかる法令を遵守しないことにより生じた損害に関し、当社は、一切その責任を負いません。
11. 本資料の全部または一部を当社の文書による事前の承諾を得ることなく転載または複製することを固くお断りいたします。
12. 本資料に関する詳細についてのお問い合わせその他お気付きの点等がございましたら当社営業窓口までご照会ください。

注 1. 本資料において使用されている「当社」とは、ルネサスエレクトロニクス株式会社およびルネサスエレクトロニクス株式会社とその総株主の議決権の過半数を直接または間接に保有する会社をいいます。

注 2. 本資料において使用されている「当社製品」とは、注 1 において定義された当社の開発、製造製品をいいます。



ユーザース・マニュアル

RX-NET

ネットワーク・ライブラリ

DNS Ver.1.30

---

対象デバイス

V850シリーズ™

対象リアルタイムOS

RX850 Pro Ver.3.15

資料番号 U15304JJ3V0UM00 (第3版)

発行年月 September 2002 CP(K)

© NEC Corporation 2001 , 2002

(メモ)

# 目次要約

|                   |    |
|-------------------|----|
| 第1章 概説 ...        | 13 |
| 第2章 インストレーション ... | 16 |
| 第3章 システム構築 ...    | 20 |
| 第4章 アドレス解決機能 ...  | 26 |
| 第5章 API関数 ...     | 32 |
| 索引 ...            | 45 |

V850シリーズ, V853, V850/SA1, V850/SB1, V850/SB2, V850/SC1, V850/SC2, V850/SC3, V850/SF1, V850/SV1, V850E/MS1, V850E/MS2, V850E/MA1, V850E/MA2, V850E/IA1, V850E/IA2は, 日本電気株式会社の商標です。  
PC/ATは米国IBM社の商標です。

Windows, Windows NTは, 米国Microsoft Corporationの米国およびその他の国における登録商標または商標です。  
Green Hills Softwareは米国Green Hills Software, Inc.の商標です。

その他, 記載の会社名/製品名は, 各社の商標, または, 登録商標です。

COPYRIGHT© 1983 to 2000 PACIFIC SOFTWARES INC. ALL RIGHT RESERVED.

COPYRIGHT© 2000 to 2002 NEC CORPORATION ALL RIGHT RESERVED.

本製品はパシフィックソフトウェア社のTCP/IPネットワーキングソフトウェアとそれに関連するソフトウェア製品をベースに開発したものです。

- **本資料の内容は予告なく変更することがありますので、最新のものであることをご確認の上ご使用ください。**
- 文書による当社の承諾なしに本資料の転載複製を禁じます。
- 本資料に記載された製品の使用もしくは本資料に記載の情報の使用に際して、当社は当社もしくは第三者の知的財産権その他の権利に対する保証または実施権の許諾を行うものではありません。上記使用に起因する第三者所有の権利にかかわる問題が発生した場合、当社はその責を負うものではありませんのでご了承ください。
- 本資料に記載された回路、ソフトウェア、及びこれらに付随する情報は、半導体製品の動作例、応用例を説明するためのものです。従って、これら回路・ソフトウェア・情報をお客様の機器に使用される場合には、お客様の責任において機器設計をしてください。これらの使用に起因するお客様もしくは第三者の損害に対して、当社は一切その責を負いません。

M7A 98.8

巻末にアンケート・コーナーを設けております。このドキュメントに対するご意見をお気軽にお寄せください。



# はじめに

**対象者** このマニュアルは、V850シリーズの応用システムを設計、開発するユーザを対象とします。

**目的** このマニュアルは、次の構成に示すRX-NETの機能をユーザに理解していただくことを目的としています。

**構成** このマニュアルは、大きく分けて次の内容で構成しています。

概 説  
インストレーション  
システム構築  
アドレス解決機能  
API関数

**読み方** このマニュアルの読者には、電気、論理回路、マイクロコンピュータ、C言語、アセンブリ言語に関する一般知識が必要です。

V850シリーズのハードウェア機能、命令機能を知りたいとき  
各製品のユーザズ・マニュアルを参照してください。

**凡 例** 注 :本文中につけた注の説明  
注意 :気をつけて読んでいただきたい内容  
備考 :本文の補足説明  
数の表記 :2進数 ...XXXX または B'XXXX  
10進数...XXXX  
16進数...0xXXX または H'XXXX  
2のべき数を示す接頭語(アドレス空間、メモリ容量):  
K(キロ) : $2^{10} = 1024$   
M(メガ) : $2^{20} = 1024^2$   
G(ギガ) : $2^{30} = 1024^3$

**関連資料** このマニュアルを使用する場合は、次の資料もあわせてご覧ください。  
関連資料は暫定版の場合がありますが、この資料では「暫定」の表示をしておりません。  
あらかじめご了承ください。

## 開発ツールに関する資料(ユーザズ・マニュアル)

(1/2)

| 資料名                                                                                                                          | 資料番号    |         |
|------------------------------------------------------------------------------------------------------------------------------|---------|---------|
|                                                                                                                              | 和文      | 英文      |
| IE-703002-MC (V853™, V850/SA1™, V850/SB1™, V850/SB2™, V850/SC1™, V850/SC2™, V850/SC3™, V850/SF1™, V850/SV1™用インサートキット・エミュレータ) | U11595J | U11595E |
| IE-V850E-MC (V850E/IA1™, V850E/IA2™用インサートキット・エミュレータ),<br>IE-V850E-MC-A (V850E/MA1™, V850E/MA2™用インサートキット・エミュレータ)              | U14487J | U14487E |

開発ツールに関する資料 (ユーザーズ・マニュアル)

(2/2)

| 資料名                                                                                  | 資料番号                    |         |         |
|--------------------------------------------------------------------------------------|-------------------------|---------|---------|
|                                                                                      | 和文                      | 英文      |         |
| IE-703003-MC-EM1 (V853用インサーキット・エミュレータ・オプション・ボード)                                     | U11596J                 | U11596E |         |
| IE-703017-MC-EM1 (V850/SA1用インサーキット・エミュレータ・オプション・ボード)                                 | U12898J                 | U12898E |         |
| IE-703037-MC-EM1 (V850/SB1,V850/SB2用インサーキット・エミュレータ・オプション・ボード)                        | U14151J                 | U14151E |         |
| IE-703040-MC-EM1 (V850/SV1用インサーキット・エミュレータ・オプション・ボード)                                 | U14337J                 | U14337E |         |
| IE-703079-MC-EM1 (V850/SF1用インサーキット・エミュレータ・オプション・ボード)                                 | U15447J                 | U15447E |         |
| IE-703102-MC (V850E/MS1™,V850E/MS2™用インサーキット・エミュレータ)                                  | U13875J                 | U13875E |         |
| IE-703102-MC-EM1, IE-703102-MC-EM1-A( V850E/MS1 ,V850E/MS2用インサーキット・エミュレータ・オプション・ボード) | U13876J                 | U13876E |         |
| IE-703107-MC-EM1 (V850E/MA1,V850E/MA2用インサーキット・エミュレータ・オプション・ボード)                      | U14481J                 | U14481E |         |
| IE-703116-MC-EM1 (V850E/IA1用インサーキット・エミュレータ・オプション・ボード)                                | U14700J                 | U14700E |         |
| CA850 Ver.2.50 Cコンパイラ・パッケージ                                                          | 操作編                     | U16053J | 作成予定    |
|                                                                                      | C言語編                    | U16054J | U16054E |
|                                                                                      | PM plus編                | U16055J | 作成予定    |
|                                                                                      | アセンブリ言語編                | U16042J | U16042E |
| リンク・ディレクティブ・ジェネレータ                                                                   | U16352J                 | -       |         |
| ID850 Ver.2.50 統合ディバッガ                                                               | 操作編 Windows®ベース         | U15181J | U15181E |
| SM850 Ver.2.50 システム・シミュレータ                                                           | 操作編 Windowsベース          | U15182J | U15182E |
| SM850 Ver.2.00以上 システム・シミュレータ                                                         | 外部部品ユーザ・オープン・インタフェース仕様編 | U14873J | U14873E |
| RX850 Ver.3.13以上 リアルタイムOS                                                            | 基礎編                     | U13430J | U13430E |
|                                                                                      | インストール編                 | U13410J | U13410E |
|                                                                                      | テクニカル編                  | U13431J | U13431E |
| RX850 Pro Ver.3.13 リアルタイムOS                                                          | 基礎編                     | U13773J | U13773E |
|                                                                                      | インストール編                 | U13774J | U13774E |
|                                                                                      | テクニカル編                  | U13772J | U13772E |
| RX-NET ネットワーク・ライブラリ (TCP/IP) Ver.1.30                                                | U15083J                 | -       |         |
| RX-NET ネットワーク・ライブラリ (PPP) Ver.1.30                                                   | U15303J                 | -       |         |
| RX-NET ネットワーク・ライブラリ (DNS) Ver.1.30                                                   | このマニュアル                 | -       |         |
| RX-NET ネットワーク・ライブラリ (DHCP) Ver.1.30                                                  | U15382J                 | -       |         |
| RX-NET ネットワーク・ライブラリ (SMTP)                                                           | U15505J                 | -       |         |
| RX-NET ネットワーク・ライブラリ (POP)                                                            | U15539J                 | -       |         |
| RX-NET Ver.1.00 ネットワーク・ライブラリ (telnet)                                                | U16085J                 | -       |         |
| RX-NET Ver.1.10 ネットワーク・ライブラリ (FTP)                                                   | U15946J                 | -       |         |
| RX-NET Ver.1.00 ネットワーク・ライブラリ (WebServer)                                             | U16294J                 | -       |         |
| RD850 Ver.3.01 タスク・ディバッガ                                                             | U13737J                 | U13737E |         |
| RD850 Pro Ver.3.01 タスク・ディバッガ                                                         | U13916J                 | U13916E |         |
| AZ850 Ver.3.10 システム・パフォーマンス・アナライザ                                                    | U14410J                 | U14410E |         |
| PG-FP4 フラッシュ・メモリ・プログラマ                                                               | U15260J                 | U15260E |         |

# 目次

|                                 |    |
|---------------------------------|----|
| 第 1 章 概説 .....                  | 13 |
| 1.1 概要 .....                    | 13 |
| 1.2 特徴 .....                    | 14 |
| 1.3 実行環境 .....                  | 15 |
| 1.4 開発環境 .....                  | 15 |
| 第 2 章 インストール .....              | 16 |
| 2.1 概要 .....                    | 16 |
| 2.2 インストール手順 .....              | 16 |
| 2.2.1 Windows ベース .....         | 16 |
| 2.2.2 UNIX ベース .....            | 17 |
| 2.3 ディレクトリ構成 .....              | 18 |
| 2.3.1 CA850 対応版 .....           | 18 |
| 2.3.2 CCV850E 対応版 .....         | 19 |
| 第 3 章 システム構築 .....              | 20 |
| 3.1 概要 .....                    | 20 |
| 3.2 CF 定義ファイルの記述 .....          | 21 |
| 3.3 情報ファイルの生成 .....             | 21 |
| 3.4 RX850 Pro 依存部の記述 .....      | 22 |
| 3.5 RX-NET(TCP/IP) 依存部の記述 ..... | 23 |
| 3.6 処理プログラムの記述 .....            | 23 |
| 3.7 オブジェクト・ファイルの生成 .....        | 24 |
| 3.8 アーカイブ・オブジェクトの生成 .....       | 24 |
| 3.9 リンク・ディレクティブ・ファイルの記述 .....   | 24 |
| 3.10 ロード・モジュールの生成 .....         | 25 |
| 第 4 章 アドレス解決機能 .....            | 26 |
| 4.1 概要 .....                    | 26 |
| 4.2 アドレス解決機能 API 関数 .....       | 26 |
| 4.2.1 RX-NET(DNS) の初期化 .....    | 26 |
| 4.2.2 ネーム・サーバの登録 .....          | 27 |
| 4.2.3 ドメイン・サーチ・リストの登録 .....     | 28 |
| 4.2.4 ホスト情報の獲得 .....            | 29 |
| 4.2.5 ホスト情報獲得処理の強制終了 .....      | 31 |
| 第 5 章 API 関数 .....              | 32 |
| 5.1 概要 .....                    | 32 |
| 5.2 API 関数の呼び出し .....           | 32 |
| 5.3 データ・マクロ .....               | 33 |
| 5.3.1 データ・タイプ .....             | 33 |
| 5.3.2 戻り値 .....                 | 33 |
| 5.4 データ構造体 .....                | 35 |
| 5.4.1 ホスト情報 .....               | 35 |
| 5.5 API 関数解説 .....              | 36 |
| 5.5.1 外部インタフェース仕様 .....         | 38 |

索引 ..... 45

# 目次

|       |                              |    |
|-------|------------------------------|----|
| 図 1-1 | RX-NET(DNS) の位置付け .....      | 13 |
| 図 1-2 | RX-NET(DNS) の階層的な位置付け .....  | 14 |
| 図 2-1 | ディレクトリ構成 (CA850 対応版) .....   | 18 |
| 図 2-2 | ディレクトリ構成 (CCV850E 対応版) ..... | 19 |
| 図 3-1 | システム構築手順 .....               | 20 |
| 図 3-2 | RX850 Pro 依存部の処理の流れ .....    | 22 |

# 表目次

|       |                         |    |
|-------|-------------------------|----|
| 表 2-1 | RX-NET(DNS) の提供形式 ..... | 16 |
| 表 5-1 | データ・タイプ .....           | 33 |
| 表 5-2 | 戻り値 .....               | 33 |
| 表 5-3 | API 関数 .....            | 38 |

# 第1章 概説

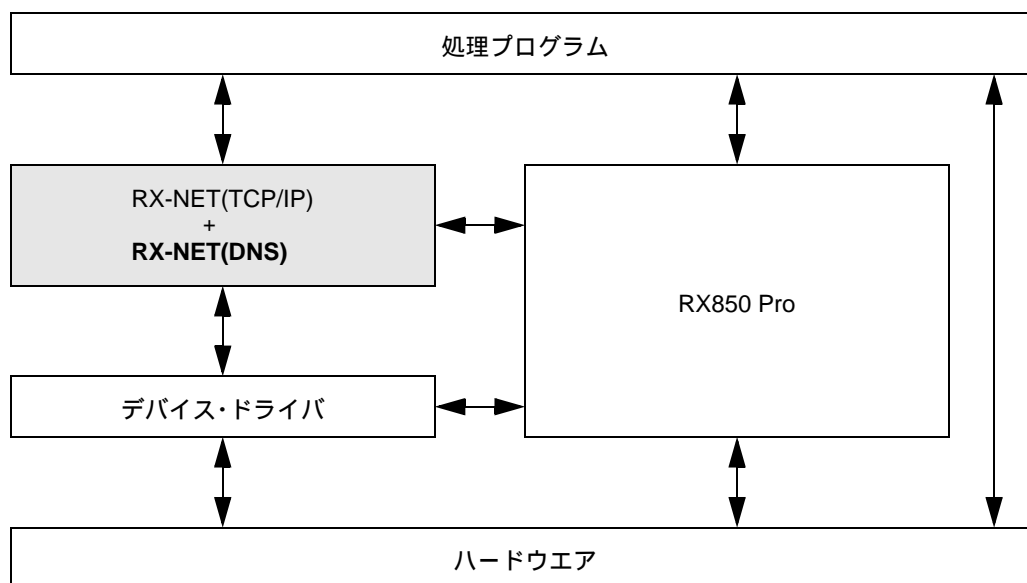
## 1.1 概要

RX-NET(DNS) は、組み込み型制御用リアルタイム・オペレーティング・システム RX850 Pro (μITRON3.0 仕様準拠, NEC 製) 上で動作する TCP/IP ライブラリ RX-NET に対し, DNS (Domain Name System) によるアドレス解決を行うためのアプリケーション・プログラム・インタフェース関数 (Application Program Interface 関数) を提供しています。

したがって, ユーザは, RX-NET(DNS) が提供する API 関数を利用することにより, ホスト名から IP アドレスを, IP アドレスからホスト名を獲得することが可能となります。

図 1-1 に, RX-NET(DNS) の位置付けを示します。

図 1-1 RX-NET(DNS) の位置付け



## 1.2 特徴

以下に，RX-NET(DNS) の特徴を示します。

- RFC に準拠

RX-NET(DNS) では，RFC に準拠した設計が行われています。

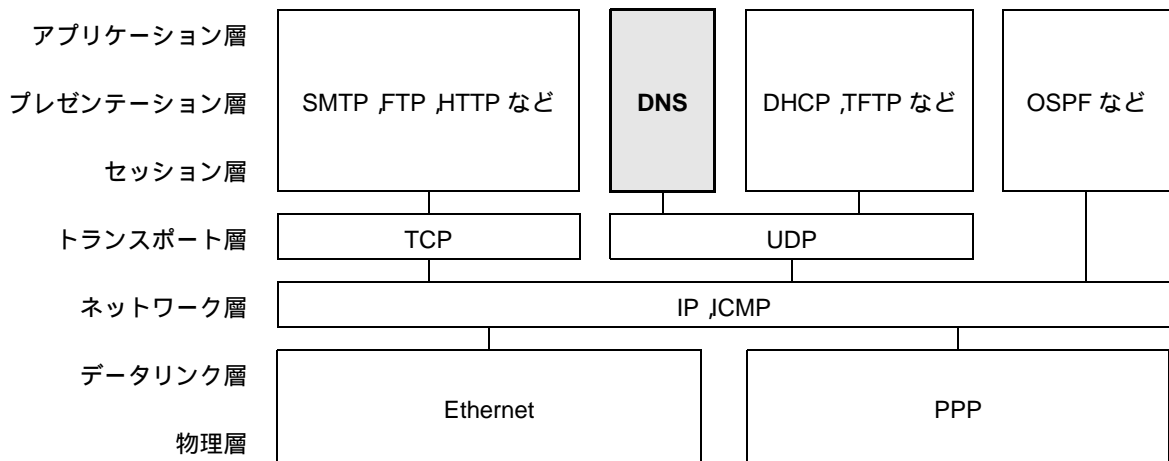
なお，RFC とは，インターネットに関する研究開発機関 IETF (Internet Engineering Task Force) が取りまとめた公開技術文書であり，電子メール，ファイル転送などのプロトコル仕様（情報交換を行う際に必要な手順，および，規約）の他にも，各種サービス，ガイドラインといった多岐に渡った情報（ネットワーク技術の実装と運用に主眼を置いた情報）が記載されています。

- アドレス解決機能をサポート

RX-NET(DNS) では，インターネット上で利用されるアドレス解決機能を提供しています。したがって，RX-NET(DNS) を組み込んだ機器は，インターネットに接続が可能となります。

図 1-2 に，RX-NET(DNS) の階層的な位置付けを示します。

図 1-2 RX-NET(DNS) の階層的な位置付け





## 1.3 実行環境

以下に、RX-NET(DNS) が処理を実行するうえで必要となるハードウェアを示します。

- プロセッサ

以下に、RX-NET(DNS) が処理を実行するうえで必要となるプロセッサを示します。

V850 シリーズ V850E/xxx, または, V850ES/xxx

- 周辺コントローラ

RX-NET(DNS) では、処理を実行するうえで、特定の周辺コントローラは必要ありません。

- メモリ容量

以下に、RX-NET(DNS) が処理を実行するうえで必要となるメモリ容量を示します。

RX-NET(DNS) のテキスト領域 : 約 20 K バイト

RX-NET(DNS) のデータ領域 : 約 10 K バイト

## 1.4 開発環境

以下に、RX-NET(DNS) を使用した処理プログラムを開発するうえで必要となるハードウェア、および、ソフトウェアを示します。

- ハードウェア

- ホスト・マシン (下記のいずれか)

PC-9821 シリーズ : Windows 2000, 98, NT 4.0

PC-98NX シリーズ : Windows 2000, 98, Me, XP, NT 4.0

IBM-PC/AT 互換機 : Windows 2000, 98, Me, XP, NT 4.0

SPARC station : Solaris Rel.2.5.x

- ソフトウェア

- リアルタイム OS

RX850 Pro Ver.3.15 以上 : NEC 製

- ネットワーク・ライブラリ

RX-NET(TCP/IP) Ver.1.30 以上 : NEC 製

- C コンパイラ・パッケージ (下記のいずれか)

CA850 Ver.2.41 以上 : NEC 製

CCV850E Ver.1.8.9 Rel.4.0.2 以上 : 米国 Green Hills Software, Inc. 製

## 第2章 インストール

本章では、RX-NET(DNS)の提供媒体に格納されているファイル群をユーザの開発環境(ホスト・マシン)上にインストールする際の手順について解説しています。

### 2.1 概要

RX-NET(DNS)の提供媒体は、ホスト・マシンの種類(Windowsベース、UNIXベース)に併せて計2種類が用意されています。

表2-1に、RX-NET(DNS)の提供形式一覧を示します。

表 2-1 RX-NET(DNS)の提供形式

| ホスト・マシン                                                            | 提供形式                                               | 提供媒体   |
|--------------------------------------------------------------------|----------------------------------------------------|--------|
| Windows ベース<br>• PC-9821 シリーズ<br>• PC-98NX シリーズ<br>• IBM-PC/AT 互換機 | CA850 対応版オブジェクト・ファイル形式<br>CCV850E 対応版オブジェクト・ファイル形式 | CD-ROM |
| UNIX ベース<br>• SPARC station                                        | CA850 対応版オブジェクト・ファイル形式<br>CCV850E 対応版オブジェクト・ファイル形式 | CD-ROM |

注意 ホスト・マシンの種類別に用意された提供媒体には、2種類(CA850対応版オブジェクト・ファイル形式、CCV850E対応版オブジェクト・ファイル形式)のRX-NET(DNS)が格納されています。したがって、提供媒体からホスト・マシン上にファイル群をインストールする際には、ユーザが使用するCコンパイラ・パッケージに対応したRX-NET(DNS)をインストールする必要があります。

### 2.2 インストール手順

RX-NET(DNS)の提供媒体に格納されているファイル群のインストール手順は、ホスト・マシンの種類(Windowsベース、UNIXベース)により異なります。

そこで、以降に、ホスト・マシンがWindowsベースの場合、UNIXベースの場合のインストール手順をそれぞれに示します。

注意 RX-NET(DNS)のインストールは、RX-NET(TCP/IP)のインストール完了後に行ってください。

#### 2.2.1 Windows ベース

以下に、RX-NET(DNS)の提供媒体に格納されているファイル群をホスト・マシン(Windowsベース:PC-9821シリーズ、PC-98NXシリーズ、IBM-PC/AT互換機)上にインストールする際の手順を示します。

- 1) Windowsの起動  
ホスト・マシン、および、周辺機器などの電源を投入し、Windowsを起動します。
- 2) 提供媒体のセット  
RX-NET(DNS)の提供媒体をホスト・マシンの該当デバイス装置(CD-ROMドライブ)にセットすることにより、セットアップ・プログラムが自動実行します。  
以降、モニタ画面に表示されるメッセージに従ってインストール作業を実行します。

注意 セットアップ・プログラムが自動実行しない場合には、RX-NET(DNS)の提供媒体のディレクトリRX-NET\\_DNS\\_V850E\\_NEC\DISK1に格納されているSETUP.EXEを起動します。

- 3) ファイル群の確認  
Windowsの標準アプリケーション Explorerなどを用いて、RX-NET(DNS)の提供媒体に格納されていたファイル群がホスト・マシン上にインストールされたことを確認します。

なお、各ディレクトリについての詳細は、「2.3 ディレクトリ構成」を参照してください。

## 2.2.2 UNIX ベース

以下に、RX-NET(DNS) の提供媒体に格納されているファイル群をホスト・マシン (UNIX ベース : SPARC station) 上にインストールする際の手順を示します。

ただし、入力例中の“%”はシェル・プロンプトを、“ ”はスペース・キーの入力を、“<Enter>”はエンター・キーの入力を表しています。

- 1) ホスト・マシンへのログイン  
ホスト・マシンにログインします。

```
%
```

- 2) ディレクトリの移動  
cd コマンドを実行し、インストール用ディレクトリに移動します。  
なお、下記入力例では、インストール用ディレクトリとして /usr/local を指定しています。

注意 インストール用ディレクトリのパーミッション (read, write, execute) はインストール作業に対して許可状態である必要があります。そこで、インストール用ディレクトリのパーミッションが不許可状態であった場合には、chmod コマンドを実行し、パーミッションを不許可状態から許可状態に変更します。

```
% cd /usr/local <Enter>
```

- 3) 提供媒体のセット  
RX-NET(DNS) の提供媒体をホスト・マシンの該当デバイス装置 (CD-ROM ドライブ) にセットします。
- 4) デバイスのマウント  
mount コマンドを実行し、該当デバイス装置に対応したデバイスをマウントします。  
なお、下記入力例では、該当デバイス装置のデバイス名 (スペシャル・ファイル名) として /dev/rst8 を、マウント・ディレクトリとして /cdrom を指定しています。  
注意 ホスト・マシンによっては、“デバイスのマウント”が自動的に行われるものがあります。このような場合、mount コマンドを実行する必要はありません。

```
% mount /dev/rst8 /cdrom <Enter>
```

- 5) ファイル群のインストール  
tar コマンドを実行し、マウント・ディレクトリ /cdrom 下の圧縮ファイルをインストール用ディレクトリに展開します。  
ただし、提供媒体には、以下に示した 2 種類の圧縮ファイルが格納されています。

- CA850 対応版
- CCV850E 対応版

そこで、ユーザが使用する C コンパイラ・パッケージが NEC 製 CA850 の場合は圧縮ファイル nec/rxnetdns.tar を、米国 Green Hills Software, Inc. 製 CCV850E の場合は圧縮ファイル ghs/rxnetdns.tar を展開します。

【 CA850 対応版の場合 】

```
% tar -xvof /cdrom/RXNET/nec/rxnetdns.tar <Enter>
```

【 CCV850E 対応版の場合 】

```
% tar -xvof /cdrom/RXNET/ghs/rxnetdns.tar <Enter>
```

- 6) ファイル群の確認  
ls コマンドを実行し、RX-NET(DNS) の提供媒体に格納されていたファイル群がホスト・マシン上にインストールされたことを確認します。  
なお、各ディレクトリについての詳細は、「2.3 ディレクトリ構成」を参照してください。

```
% ls -CFR /usr/local/nertools32 <Enter>
```

## 2.3 ディレクトリ構成

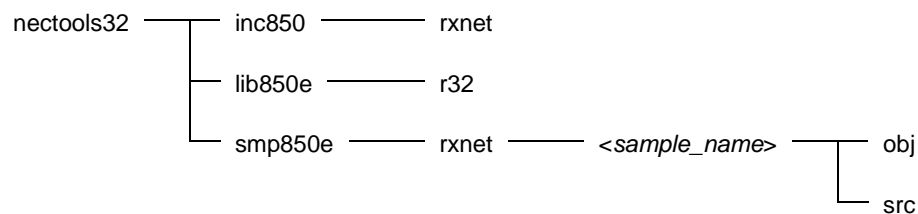
RX-NET(DNS) の提供媒体に格納されているファイル群のディレクトリ構成は、ユーザが使用する C コンパイラ・パッケージの種類 (NEC 製 CA850, 米国 Green Hills Software, Inc. 製 CCV850E) により異なります。

そこで、以降に、C コンパイラ・パッケージが CA850 の場合、CCV850E の場合のディレクトリ構成をそれぞれに示します。

### 2.3.1 CA850 対応版

図 2-1 に、RX-NET(DNS) の提供媒体 (CA850 対応版) に格納されているファイル群をホスト・マシン上にインストールした際に生成されるディレクトリ構成を示します。

図 2-1 ディレクトリ構成 (CA850 対応版)



以下に、各ディレクトリの概要を示します。

- 1) nectools32\inc850  
RX-NET(DNS) の標準ヘッダ・ファイルが格納されているディレクトリです。  
rxnet\_dns.h : RX-NET(DNS) 用標準ヘッダ・ファイル
- 2) nectools32\inc850\rxnet  
RX-NET(DNS) のヘッダ・ファイルが格納されているディレクトリです。
- 3) nectools32\lib850e\r32  
DNS リゾルバ・ライブラリ (32 レジスタ・モード) が格納されているディレクトリです。  
libdns\_rslv.a : DNS リゾルバ・ライブラリ
- 4) nectools32\smp850e\rxnet\<sample\_name>\obj  
ロード・モジュールを生成するためのメイク・ファイル Makefile が格納されているディレクトリです。  
なお、本ディレクトリにおいて、make コマンドを実行することにより、ロード・モジュール sample.out が本ディレクトリに生成されます。  
Makefile : ロード・モジュール用メイク・ファイル
- 5) nectools32\smp850e\rxnet\<sample\_name>\src  
サンプル・プログラムのソース・ファイル、および、ヘッダ・ファイルが格納されているディレクトリです。

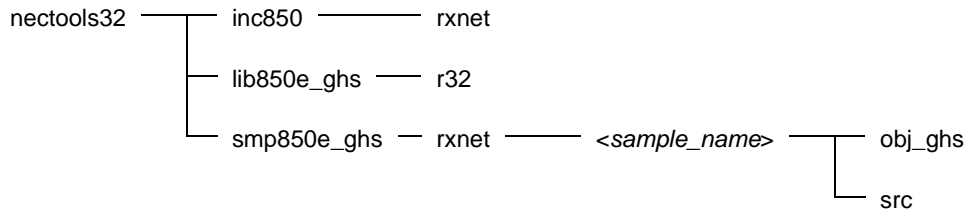
注意 <sample\_name> についての詳細は、下記に示したテキスト・ファイルを参照してください。

<sample\_name> : nectools32\smp850e\rxnet\README.DNS

## 2.3.2 CCV850E 対応版

図 2-2 に、RX-NET(DNS) の提供媒体 (CCV850E 対応版) に格納されているファイル群をホスト・マシン上にインストールした際に生成されるディレクトリ構成を示します。

図 2-2 ディレクトリ構成 (CCV850E 対応版)



以下に、各ディレクトリの概要を示します。

- 1) nectools32\inc850  
RX-NET(DNS) の標準ヘッダ・ファイルが格納されているディレクトリです。  
rxnet\_dns.h : RX-NET(DNS) 用標準ヘッダ・ファイル
- 2) nectools32\inc850\rxnet  
RX-NET(DNS) のヘッダ・ファイルが格納されているディレクトリです。
- 3) nectools32\lib850e\_ghs\r32  
DNS リゾルバ・ライブラリ (32 レジスタ・モード) が格納されているディレクトリです。  
libdns\_rslv.a : DNS リゾルバ・ライブラリ
- 4) nectools32\smp850e\_ghs\rxnet\<sample\_name>\obj\_ghs  
ロード・モジュールを生成するためのビルド・ファイル sample.bld が格納されているディレクトリです。  
なお、本ディレクトリの sample.bld を用いることにより、ロード・モジュール sample.out が本ディレクトリに生成されます。  
sample.bld : ロード・モジュール用ビルド・ファイル
- 5) nectools32\smp850e\_ghs\rxnet\<sample\_name>\src  
サンプル・プログラムのソース・ファイル、および、ヘッダ・ファイルが格納されているディレクトリです。

注意 <sample\_name> についての詳細は、下記に示したテキスト・ファイルを参照してください。

<sample\_name> : nectools32\smp850e\_ghs\rxnet\README.DNS

# 第3章 システム構築

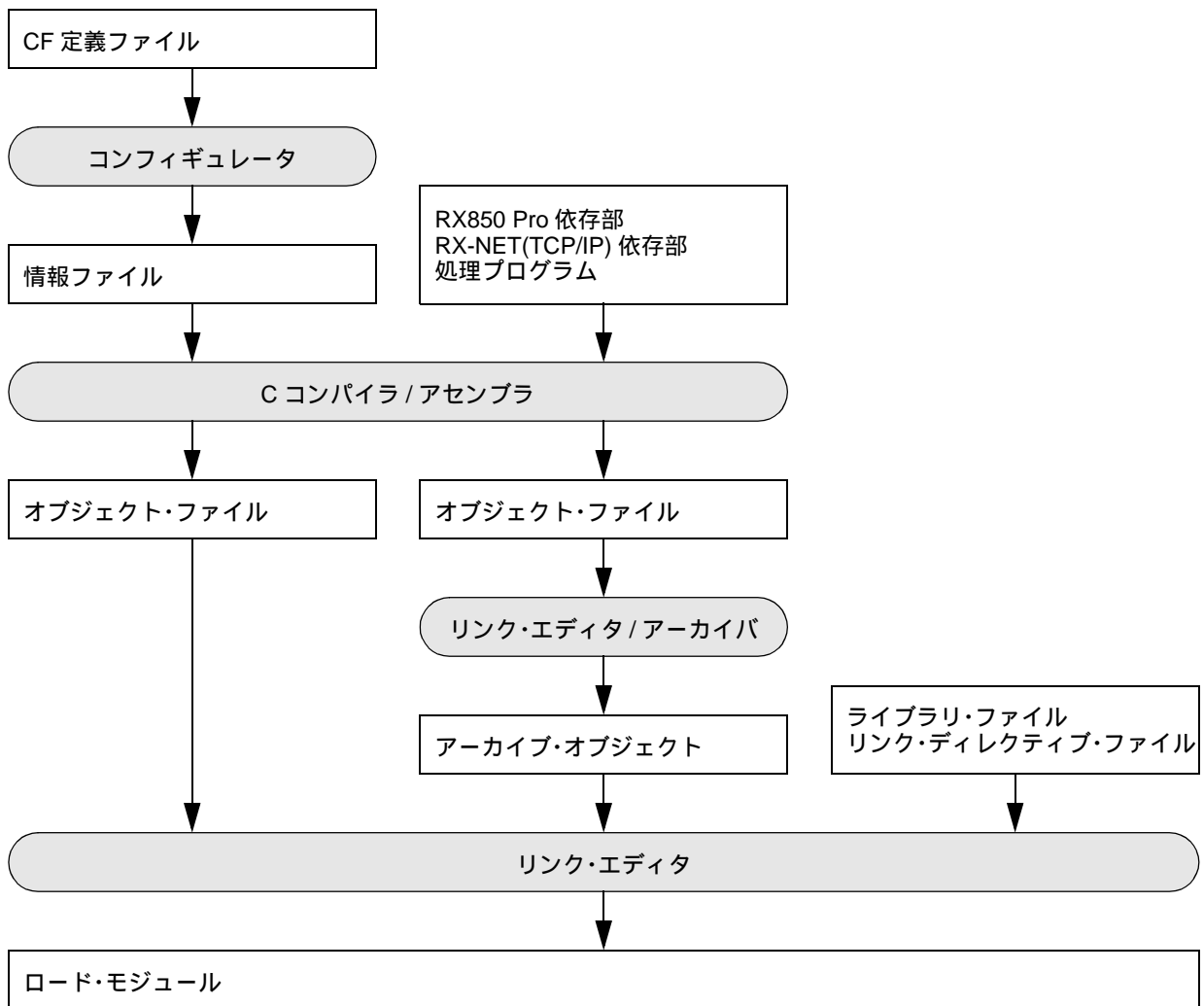
本章では、RX-NET(DNS)を使用したネットワーク・アプリケーション(ロード・モジュール)の構築手順を解説しています。

## 3.1 概要

システム構築とは、RX-NET(DNS)の提供媒体からユーザの開発環境(ホスト・マシン)上にインストールしたファイル群を用いてネットワーク・アプリケーション(ロード・モジュール)を生成することです。

図3-1に、RX-NET(DNS)のシステム構築手順を示します。

図3-1 システム構築手順



## 3.2 CF 定義ファイルの記述

組み込み型制御用リアルタイム・オペレーティング・システム RX850 Pro(μITRON3.0 仕様準拠, NEC 製)の管理下で動作する処理プログラムを作成する場合, RX850 Pro に提供するコンフィギュレーション情報(リアルタイム OS 情報, SIT 情報, SCT 情報)を保持した CF 定義ファイルが必要となります。

- 注意 1 CF 定義ファイルを記述する際の注意事項, および, コンフィギュレーション情報についての詳細は, 「**RX850 Pro ユーザーズ・マニュアル インストレーション編**」を参照してください。
- 注意 2 RX-NET(DNS) では, CF 定義ファイルのサンプル・ソース・ファイルを提供しています。

### 【 CA850 対応版の場合 】

```
nectools32\smp850e\rxnet\\src
sys.cf          : CF 定義ファイル
```

### 【 CCV850E 対応版の場合 】

```
nectools32\smp850e_ghs\rxnet\\src
sys.cf          : CF 定義ファイル
```

## 3.3 情報ファイルの生成

「3.2 CF 定義ファイルの記述」で作成された CF 定義ファイルに対して RX850 Pro が提供するユーティリティ・ツール(コンフィギュレータ cf850pro)を実行し, 情報ファイル(システム情報テーブル, システム・コール・テーブル, システム情報ヘッダ・ファイル)を生成します。

以下に, シェル・プロンプトのコマンド・ラインから cf850pro を実行する際の入力例(CF 定義ファイル sys.cf を読み込んだのち, システム情報テーブル sit.s, システム・コール・テーブル svc.s, システム情報ヘッダ・ファイル sys.h を出力)を示します。

ただし, 入力例中の “C>” はシェル・プロンプトを, “ ” はスペース・キーの入力を, “<Enter>” はエンター・キーの入力を表しています。

```
C> cf850pro -i sit.s -c svc.s -d sys.h sys.cf <Enter>
```

- 注意 1 コンフィギュレータ cf850pro の起動オプション, および, 実行方法についての詳細は, 「**RX850 Pro ユーザーズ・マニュアル インストレーション編**」を参照してください。
- 注意 2 RX-NET(DNS) では, 情報ファイルを生成するためのサンプル・コマンド・ファイルを提供しています。

### 【 CA850 対応版の場合 】

```
nectools32\smp850e\rxnet\\obj
Makefile      : ロード・モジュール用メイク・ファイル
```

### 【 CCV850E 対応版の場合 】

```
nectools32\smp850e_ghs\rxnet\\obj_ghs
sample.bld    : ロード・モジュール用ビルド・ファイル
```

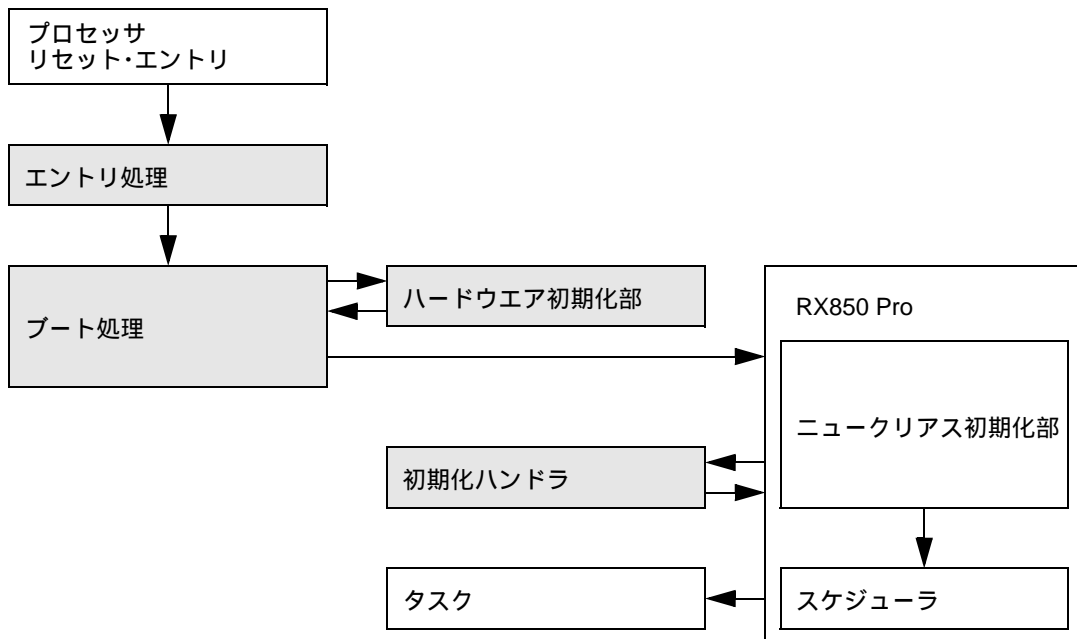
### 3.4 RX850 Pro 依存部の記述

RX-NET(DNS) では、RX850 Pro が提供する機能を利用して各種機能を実現しています。また、ユーザが記述した処理プログラムは、RX850 Pro の管理下でその処理を実行することになります。

したがって、RX850 Pro を正常に動作させるうえで必要となる RX850 Pro 依存部 ( ユーザ・OWN・コーディング部 ) の記述が必要となります。

図 3-2 に、RX850 Pro 依存部の処理の流れを示します。

図 3-2 RX850 Pro 依存部の処理の流れ



以下に、RX850 Pro 依存部の一覧を示します。

- エントリ処理**  
 割り込みが発生した際にプロセッサが強制的に制御を移すハンドラ・アドレスに対して該当処理(ブート処理、RX850 Pro が提供する割り込み処理管理機能、直接起動割り込みハンドラ)への分岐処理を割り付けるために用意された処理ルーチンです。
- ブート処理**  
 RX850 Pro が処理を実行するうえで必要となる最低限の初期化処理を行うために用意された処理ルーチンであり、エントリ処理(プロセッサのリセット・エントリに割り付けられた分岐処理)から呼び出されます。
- ハードウェア初期化部**  
 RX850 Pro が処理を実行するうえで必要となるハードウェアの初期化処理を行うために用意された処理ルーチンであり、ブート処理から呼び出されます。  
 なお、RX850 Pro では、一定周期で発生するタイマ割り込みを利用して時間管理を行っています。そこで、RX850 Pro が時間管理に利用するタイマ割り込みを発生するハードウェア(リアルタイム・パルス・ユニット、または、タイマ・コントローラ)に対しては、CF 定義ファイル作成時にシステム情報で定義した基本クロック周期でタイマ割り込みが発生するような設定を行う必要があります。
- 初期化ハンドラ**  
 ユーザの実行環境/アプリケーション・システムに依存した初期化処理を行うために用意された処理ルーチンであり、ニュークリアス初期化部から呼び出されます。  
 なお、RX850 Pro では、初期化ハンドラを“タスク”として位置付けています。

注意 1 RX850 Pro 依存部を記述する際の注意事項についての詳細は、「**RX850 Pro ユーザーズ・マニュアル インストラクション編**」を参照してください。

注意 2 RX-NET(DNS) では、初期化ハンドラのサンプル・ソース・ファイルを提供しています。



## 【 CA850 対応版の場合 】

```
nectools32\smp850e\rxnet\<sample_name>\src
varfunc.c          : 初期化ハンドラ
```

## 【 CCV850E 対応版の場合 】

```
nectools32\smp850e_ghs\rxnet\<sample_name>\src
varfunc.c          : 初期化ハンドラ
```

### 3.5 RX-NET(TCP/IP) 依存部の記述

RX-NET(DNS) では、RX-NET(TCP/IP) が提供する機能を利用して各種アドレス解決機能を実現しています。したがって、RX-NET(DNS) を正常に動作させるうえで必要となる RX-NET(TCP/IP) 依存部 ( ユーザ・OWN・コーディング部 ) の記述が必要となります。

注意 RX-NET(TCP/IP) 依存部を記述する際の注意事項についての詳細は、「**RX-NET(TCP/IP) ユーザーズ・マニュアル**」を参照してください。

### 3.6 処理プログラムの記述

ネットワーク・アプリケーションとして実現すべき処理 ( 処理プログラム ) を記述します。なお、処理プログラムは、用途別に以下のように分類 / 区別されています。

- タスク  
RX850 Pro の管理下で実行可能な処理プログラムの最小単位です。
- 直接起動割り込みハンドラ  
割り込みが発生した際、RX850 Pro を介在させることなく起動される割り込み処理専用ルーチンです。  
なお、RX850 Pro では、直接起動割り込みハンドラを“タスク”とは独立したもの ( 非タスク ) として位置付けています。このため、割り込みが発生した際には、システム内で最高優先度を持つタスクが実行中であっても、その処理は中断され、直接起動割り込みハンドラに制御が移ります。
- 間接起動割り込みハンドラ  
割り込みが発生した際に RX850 Pro による割り込み前処理 ( レジスタの退避、スタックの切り替えなど ) を行わせただのちに起動される割り込み処理専用ルーチンです。  
なお、RX850 Pro では、間接起動割り込みハンドラを“タスク”とは独立したもの ( 非タスク ) として位置付けています。このため、割り込みが発生した際には、システム内で最高優先度を持つタスクが実行中であっても、その処理は中断され、間接起動割り込みハンドラに制御が移ります。
- 周期起動ハンドラ  
一定の時間が経過した際に起動される周期処理専用ルーチンです。  
なお、RX850 Pro では、周期起動ハンドラを“タスク”とは独立したもの ( 非タスク ) として位置付けています。このため、一定の時間が経過した際には、システム内で最高優先度を持つタスクが実行中であっても、その処理は中断され、周期起動ハンドラに制御が移ります。
- 拡張 SVC ハンドラ  
ユーザが記述した関数を拡張システム・コールとして RX850 Pro に登録した処理ルーチンです。  
なお、RX850 Pro では、拡張 SVC ハンドラを“拡張 SVC ハンドラを呼び出した処理プログラム ( タスク、非タスク ) の延長線”として位置付けています。
- 拡張 SVC ハンドラ用インタフェース・ルーチン  
処理プログラム ( タスク、非タスク ) から 4 個以上の引き継ぎデータを持った拡張 SVC ハンドラを呼び出す際に必要となるインタフェース・ルーチンです。

注意 1 処理プログラムを記述する際の注意事項についての詳細は、「**RX850 Pro ユーザーズ・マニュアル 基礎編**」を参照してください。

注意 2 RX-NET(DNS) では、処理プログラムのサンプル・ソース・ファイルを提供しています。

## 【 CA850 対応版の場合 】

```
nectools32\smp850e\rxnet\\src
task.c          : タスク
```

## 【 CCV850E 対応版の場合 】

```
nectools32\smp850e_ghs\rxnet\\src
task.c          : タスク
```

### 3.7 オブジェクト・ファイルの生成

「3.2 CF 定義ファイルの記述」～「3.6 処理プログラムの記述」で作成された C 言語ソース・ファイル / アセンブリ言語ソース・ファイルに対して C コンパイラ / アセンブラを実行し、リロケートブルなオブジェクト・ファイルを生成します。

- 注意 1 C コンパイラ / アセンブラの起動オプション, および, 実行方法についての詳細は, 使用する C コンパイラ・パッケージのユーザズ・マニュアルを参照してください。
- 注意 2 RX-NET(DNS) では, オブジェクト・ファイルを生成するためのサンプル・コマンド・ファイルを提供しています。

## 【 CA850 対応版の場合 】

```
nectools32\smp850e\rxnet\\obj
Makefile       : ロード・モジュール用メイク・ファイル
```

## 【 CCV850E 対応版の場合 】

```
nectools32\smp850e_ghs\rxnet\\obj_ghs
sample.bld     : ロード・モジュール用ビルド・ファイル
```

### 3.8 アーカイブ・オブジェクトの生成

「3.7 オブジェクト・ファイルの生成」において作成したリロケートブルなオブジェクト・ファイルのうち, 1 個のオブジェクトとしてまとめることが可能なものについてはリンク・エディタ / アーカイバを実行し, アーカイブ・オブジェクトを生成します。

- 注意 1 リンク・エディタ / アーカイバの起動オプション, および, 実行方法についての詳細は, 使用する C コンパイラ・パッケージのユーザズ・マニュアルを参照してください。
- 注意 2 RX-NET(DNS) では, アーカイブ・オブジェクトを生成するためのサンプル・コマンド・ファイルを提供しています。

## 【 CA850 対応版の場合 】

```
nectools32\smp850e\rxnet\\obj
Makefile       : ロード・モジュール用メイク・ファイル
```

## 【 CCV850E 対応版の場合 】

```
nectools32\smp850e_ghs\rxnet\\obj_ghs
sample.bld     : ロード・モジュール用ビルド・ファイル
```

### 3.9 リンク・ディレクティブ・ファイルの記述

リンク・エディタが行うアドレス割り付けをユーザが固定化するためのファイル (リンク・ディレクティブ・ファイル) を記述します。

- 注意 1 リンク・ディレクティブ・ファイルを記述する際の注意事項についての詳細は, 使用する C コンパイラ・パッケージのユーザズ・マニュアルを参照してください。
- 注意 2 RX-NET(DNS) では, リンク・ディレクティブ・ファイルのサンプル・ソース・ファイルを提供しています。

## 【 CA850 対応版の場合 】

```
nectools32\smp850e\rxnet\\src
sample.dir     : リンク・ディレクティブ・ファイル
```

## 【CCV850E 対応版の場合】

```
nectools32\smp850e_ghs\rxnet\<sample_name>\obj_ghs
sample.lx          : リンク・ディレクティブ・ファイル
```

### 3.10 ロード・モジュールの生成

「3.7 オブジェクト・ファイルの生成」～「3.8 アーカイブ・オブジェクトの生成」で作成したオブジェクト・ファイル, アーカイブ・オブジェクト, 「3.9 リンク・ディレクティブ・ファイルの記述」で作成したリンク・ディレクティブ・ファイルの他に, C コンパイラ・パッケージ, RX850 Pro, RX-NET(TCP/IP), RX-NET(DNS) などが提供しているライブラリ・ファイルに対してリンク・エディタを実行し, ロード・モジュールを生成します。

- 注意 1 リンク・エディタの起動オプション, および, 実行方法についての詳細は, 使用する C コンパイラ・パッケージのユーザーズ・マニュアルを参照してください。
- 注意 2 RX-NET(DNS) では, ロード・モジュールを生成するためのサンプル・コマンド・ファイルを提供しています。

## 【CA850 対応版の場合】

```
nectools32\smp850e\rxnet\<sample_name>\obj
Makefile          : ロード・モジュール用メイク・ファイル
```

## 【CCV850E 対応版の場合】

```
nectools32\smp850e_ghs\rxnet\<sample_name>\obj_ghs
sample.bld        : ロード・モジュール用ビルド・ファイル
```

## 第4章 アドレス解決機能

本章では、RX-NET(DNS) が提供しているアドレス解決機能について解説しています。

### 4.1 概要

RX-NET(DNS) では、RX-NET(DNS) の初期化処理、ネーム・サーバの登録処理、ドメイン・サーチ・リストの登録処理の他に、ホスト情報の獲得処理などを“アドレス解決機能”として提供しています。

### 4.2 アドレス解決機能 API 関数

#### 4.2.1 RX-NET(DNS) の初期化

RX-NET(DNS) の初期化は、以下に示した API 関数を処理プログラム (タスク) から発行することにより実現されます。

- RslvInit

RX-NET(DNS) が提供する機能を実現するうえで必要となる各種初期化処理を実行します。

なお、RX-NET(DNS) では、RX-NET(DNS) の初期化処理として、RX-NET(DNS) が提供している機能を実現する際に必要となる各種資源の生成、および、メモリ領域の確保を行っています。

以下に、本 API 関数の記述例を示します。

なお、記述例中の `so_initialize` は RX-NET(TCP/IP) が提供している API 関数、`ext_tsk` は RX850 Pro が提供しているシステム・コールです。

```
#include <stdrx85p.h> /* RX850 Pro 用標準ヘッダ・ファイルの定義 */
#include <rxnet.h> /* RX-NET(TCP/IP) 用標準ヘッダ・ファイルの定義 */
#include <fnsconfig.h> /* 静的設定情報ヘッダ・ファイルの定義 */
#include <rxnet_dns.h> /* RX-NET(DNS) 用標準ヘッダ・ファイルの定義 */

void
func_task ( INT stacd ) {
    so_initialize ( ); /* RX-NET(TCP/IP) の初期化 */
    RslvInit ( ); /* RX-NET(DNS) の初期化 */
    ext_tsk ( ); /* タスクの終了処理 */
}
```

注意 本API関数の発行は、RX-NET(TCP/IP)が提供するAPI関数`so_initialize`の処理完了後に行う必要があります。

## 4.2.2 ネーム・サーバの登録

ネーム・サーバの登録は、以下に示した API 関数を処理プログラム (タスク) から発行することにより実現されます。

- SetNameServers

パラメータ *dns\_addr1* および、パラメータ *dns\_addr2* で指定されたネーム・サーバを RX-NET(DNS) に登録します。以下に、本 API 関数の記述例を示します。

なお、記述例中の *inet\_addr* は RX-NET(TCP/IP) が提供している API 関数、*ext\_tsk* は RX850 Pro が提供しているシステム・コールです。

```
#include      <stdrx85p.h>           /* RX850 Pro 用標準ヘッダ・ファイルの定義 */
#include      <rxnet.h>             /* RX-NET(TCP/IP) 用標準ヘッダ・ファイルの定義 */
#include      <fnsconfig.h>        /* 静的設定情報ヘッダ・ファイルの定義 */
#include      <rxnet_dns.h>        /* RX-NET(DNS) 用標準ヘッダ・ファイルの定義 */

void
func_task ( INT stacd ) {
    u32          dns_addr1 = inet_addr ( "192.168.0.1" );      /* 変数の宣言, 初期化 */
    u32          dns_addr2 = inet_addr ( "133.202.215.254" ); /* 変数の宣言, 初期化 */

    SetNameServers ( dns_addr1 , dns_addr2 ); /* ネーム・サーバの登録 */
    ext_tsk ( ); /* タスクの終了処理 */
}
```

注意 セカンダリ・ネーム・サーバが存在しない場合には、パラメータ *dns\_addr2* に 0x0 を設定します。

### 4.2.3 ドメイン・サーチ・リストの登録

ドメイン・サーチ・リストの登録は、以下に示した API 関数を処理プログラム (タスク) から発行することにより実現されます。

- SetSearchList

パラメータ *domain\_list* で指定されたドメイン・サーチ・リスト (最大検索要素数: 0x6) を RX-NET(DNS) に登録します。

以下に、本 API 関数の記述例を示します。

なお、記述例中の *ext\_tsk* は RX850 Pro が提供しているシステム・コールです。

```
#include <stdrx85p.h> /* RX850 Pro 用標準ヘッダ・ファイルの定義 */
#include <rxnet.h> /* RX-NET(TCP/IP) 用標準ヘッダ・ファイルの定義 */
#include <fnconfig.h> /* 静的設定情報ヘッダ・ファイルの定義 */
#include <rxnet_dns.h> /* RX-NET(DNS) 用標準ヘッダ・ファイルの定義 */

char domain_list [] = { /* 変数の宣言, 初期化 */
    "sample1.nec.co.jp",
    "sample2.nec.co.jp",
    "sample3.nec.co.jp",
    "sample4.nec.co.jp",
    "sample5.nec.co.jp",
    "sample6.nec.co.jp",
    0x0
};

void
func_task ( INT stacd ) {
    SetSearchList ( &domain_list ); /* ドメイン・サーチ・リストの登録 */
    ext_tsk (); /* タスクの終了処理 */
}
```

注意 1 RX-NET(DNS) では、ドメイン・サーチ・リスト *domain\_list* に設定されている検索要素を利用する際、ポインタ参照しています。このため、ドメイン・サーチ・リスト *domain\_list* は、外部変数、または、静的変数として定義する必要があります。

注意 2 ドメイン・サーチ・リスト *domain\_list* の設定例を以下に示します。  
 なお、ドメイン・サーチ・リスト *domain\_list* の終端には、ドメイン・サーチ・リストの終端を意味する 0x0 を設定する必要があります。

【 検索要素数が 3 個の場合 】

```
char domain_list [] = {
    "sample1.nec.co.jp", /* 第 1 検索要素 */
    "sample2.nec.co.jp", /* 第 2 検索要素 */
    "sample3.nec.co.jp", /* 第 3 検索要素 */
    0x0 /* 終端 */
};
```

【 検索要素数が 6 個の場合 】

```
char domain_list [] = {
    "sample1.nec.co.jp", /* 第 1 検索要素 */
    "sample2.nec.co.jp", /* 第 2 検索要素 */
    "sample3.nec.co.jp", /* 第 3 検索要素 */
    "sample4.nec.co.jp", /* 第 4 検索要素 */
    "sample5.nec.co.jp", /* 第 5 検索要素 */
    "sample6.nec.co.jp", /* 第 6 検索要素 */
    0x0 /* 終端 */
};
```

## 4.2.4 ホスト情報の獲得

ホスト情報の獲得は、以下に示した API 関数を処理プログラム (タスク) から発行することにより実現されます。

- GetHostByName

パラメータ *name* で指定されたホスト名に対応したホスト情報を獲得します。

なお、本 API 関数の処理が正常終了した際には、“対応するホスト情報へのポインタ”が戻り値として返されます。以下に、本 API 関数の記述例を示します。

なお、記述例中の *wai\_sem*、*sig\_sem*、*ext\_tsk* は RX850 Pro が提供しているシステム・コールです。

```
#include <stdrx85p.h> /* RX850 Pro 用標準ヘッダ・ファイルの定義 */
#include <rxnet.h> /* RX-NET(TCP/IP) 用標準ヘッダ・ファイルの定義 */
#include <fnsconfig.h> /* 静的設定情報ヘッダ・ファイルの定義 */
#include <rxnet_dns.h> /* RX-NET(DNS) 用標準ヘッダ・ファイルの定義 */

void
func_task ( INT stacd ) {
extern ID          semid; /* 変数の宣言 */
struct hostent    hostinfo; /* データ構造体の宣言 */
char              name [ ] = "www.sample.nec.co.jp"; /* 変数の宣言, 初期化 */
int               errp; /* 変数の宣言 */

    wai_sem ( semid ); /* 資源の獲得 ( 排他制御の開始 ) */
    &hostinfo = GetHostByName ( &name , &errp ); /* ホスト情報の獲得 */

    ..... /* ホスト情報の参照 */
    .....
    .....

    sig_sem ( semid ); /* 資源の返却 ( 排他制御の終了 ) */
    ext_tsk ( ); /* タスクの終了処理 */
}
```

注意 1 本 API 関数を発行する際には、マルチタスク処理を考慮した記述 (API 関数の発行に伴うタスク間の排他制御) が必要となります。

注意 2 ホスト名 *name* の設定例を以下に示します。

【ホスト名のみを設定する場合】

```
char          name [ ] = "www";
```

【ホスト名、および、ドメイン名を設定する場合】

```
char          name [ ] = "www.sample.nec.co.jp";
```

注意 3 本 API 関数の発行から 150 秒が経過するまでの間にホスト情報を獲得することが出来なかった場合、*errp* で指定された領域に ETIMEDOUT が格納されます。

注意 4 パラメータ *errp* で指定された領域には、以下に示したエラー・コードが格納されます。

|              |        |                                            |
|--------------|--------|--------------------------------------------|
| ENOERR       | (0x0)  | : 正常終了                                     |
| EINVAL       | (0x16) | : パラメータの指定が不正です                            |
| ECONNABORTED | (0x61) | : ホスト情報獲得処理を実行中に他タスクから RslvAbort が発行された    |
| ETIMEDOUT    | (0x67) | : 接続がタイムアウトしています                           |
| ECONNREFUSED | (0x68) | : ネーム・サーバとの接続処理が失敗したため、ホスト情報を獲得することが出来なかった |

注意 5 ホスト情報 *hostent* についての詳細は、「5.4.1 ホスト情報」を参照してください。

- GetHostByAddr

パラメータ *ipaddr* で指定された IP アドレスに対応したホスト情報を獲得します。

なお、本 API 関数の処理が正常終了した際には、“対応するホスト情報へのポインタ”が戻り値として返されます。

以下に、本 API 関数の記述例を示します。

なお、記述例中の *inet\_addr* は RX-NET(TCP/IP) が提供している API 関数、*wai\_sem*、*sig\_sem*、*ext\_tsk* は RX850 Pro が提供しているシステム・コールです。

```
#include <stdrx85p.h> /* RX850 Pro 用標準ヘッダ・ファイルの定義 */
#include <rxnet.h> /* RX-NET(TCP/IP) 用標準ヘッダ・ファイルの定義 */
#include <fnsconfig.h> /* 静的設定情報ヘッダ・ファイルの定義 */
#include <rxnet_dns.h> /* RX-NET(DNS) 用標準ヘッダ・ファイルの定義 */

void
func_task ( INT stacd ) {
extern ID semid; /* 変数の宣言 */
struct hostent hostinfo; /* データ構造体の宣言 */
u32 ipaddr = inet_addr ( "192.168.0.100" ); /* 変数の宣言, 初期化 */
int errp; /* 変数の宣言 */

wai_sem ( semid ); /* 資源の獲得 ( 排他制御の開始 ) */
&hostinfo = GetHostByAddr ( &ipaddr , &errp ); /* ホスト情報の獲得 */

..... /* ホスト情報の参照 */
.....
.....

sig_sem ( semid ); /* 資源の返却 ( 排他制御の終了 ) */
ext_tsk ( ); /* タスクの終了処理 */
}
```

注意 1 本 API 関数を発行する際には、マルチタスク処理を考慮した記述 (API 関数の発行に伴うタスク間の排他制御) が必要となります。

注意 2 本 API 関数の発行から 150 秒が経過するまでの間にホスト情報を獲得することが出来なかった場合、*errp* で指定された領域に ETIMEDOUT が格納されます。

注意 3 パラメータ *errp* で指定された領域には、以下に示したエラー・コードが格納されます。

|              |        |                                            |
|--------------|--------|--------------------------------------------|
| ENOERR       | (0x0)  | : 正常終了                                     |
| EINVAL       | (0x16) | : パラメータの指定が不正です                            |
| ECONNABORTED | (0x61) | : ホスト情報獲得処理を実行中に他タスクから RslvAbort が発行された    |
| ETIMEDOUT    | (0x67) | : 接続がタイムアウトしています                           |
| ECONNREFUSED | (0x68) | : ネーム・サーバとの接続処理が失敗したため、ホスト情報を獲得することが出来なかった |

注意 4 ホスト情報 *hostent* についての詳細は、「5.4.1 ホスト情報」を参照してください。



## 4.2.5 ホスト情報獲得処理の強制終了

ホスト情報獲得処理の強制終了は、以下に示した API 関数を処理プログラム (タスク, 非タスク) から発行することにより実現されます。

- RslvAbort

本 API 関数を発行した際、処理の完了していない API 関数 (GetHostByName, GetHostByAddr) を強制的に異常終了させます。

なお、本 API 関数の発行により異常終了した API 関数には、エラー・コードとして ECONNABORTED が設定されます。

以下に、本 API 関数の記述例を示します。

なお、記述例中の ext\_tsk は RX850 Pro が提供しているシステム・コールです。

```
#include <stdrx85p.h> /* RX850 Pro 用標準ヘッダ・ファイルの定義 */
#include <rxnet.h> /* RX-NET(TCP/IP) 用標準ヘッダ・ファイルの定義 */
#include <fnsconfig.h> /* 静的設定情報ヘッダ・ファイルの定義 */
#include <rxnet_dns.h> /* RX-NET(DNS) 用標準ヘッダ・ファイルの定義 */

void
func_task ( INT stacd ) {
    RslvAbort ( ); /* ホスト情報獲得処理の強制終了 */
    ext_tsk ( ); /* タスクの終了処理 */
}
```

**注意** 本 API 関数では、終了要求のキューイングが行われません。このため、ホスト情報獲得処理を実行中のタスクが存在しなかった場合には、何も処理は行わず、エラーとしても扱いません。

# 第 5 章 API 関数

本章では、RX-NET(DNS) が提供しているアプリケーション・プログラム・インタフェース関数 (API 関数) について解説しています。

## 5.1 概要

RX-NET(DNS) が提供している API 関数は、ユーザが記述した処理プログラムから RX-NET(DNS) が直接管理している資源を間接的に操作するために用意されたサービス・ルーチンです。

以下に、RX-NET(DNS) が提供している API 関数 (5 種類) を示します。

|               |                |               |               |
|---------------|----------------|---------------|---------------|
| RslvInit      | SetNameServers | SetSearchList | GetHostByName |
| GetHostByAddr | RslvAbort      |               |               |

## 5.2 API 関数の呼び出し

API 関数を C 言語、および、アセンブリ言語で記述された処理プログラムから発行する場合の呼び出し方法を以下に示します。

- C 言語  
API 関数を C 言語で記述された処理プログラムから発行する場合、通常の C 言語関数と同様の方法で呼び出しを行うことにより、API 関数のパラメータは RX-NET(DNS) に引き数として渡され、該当処理が実行されます。
- アセンブリ言語  
API 関数をアセンブリ言語で記述された処理プログラムから発行する場合、ユーザが開発環境として使用する C コンパイラ・パッケージの関数呼び出し規約に従ったパラメータ、および、戻り番地の設定を行ったのち、jarl 命令による呼び出しを行うことにより、API 関数のパラメータは RX-NET(DNS) に引き数として渡され、該当処理が実行されます。

注意 RX-NET(DNS) が提供する API 関数を処理プログラムから発行する場合、以下に示したヘッダ・ファイルの定義 (インクルード処理) を行う必要があります。

|             |   |                            |
|-------------|---|----------------------------|
| stdrx85p.h  | : | RX850 Pro 用標準ヘッダ・ファイル      |
| rxnet.h     | : | RX-NET(TCP/IP) 用標準ヘッダ・ファイル |
| fnsconfig.h | : | 静的設定情報ヘッダ・ファイル             |
| rxnet_dns.h | : | RX-NET(DNS) 用標準ヘッダ・ファイル    |

なお、stdrx85p.h は RX850 Pro が、rxnet.h、fnsconfig.h は RX-NET(TCP/IP) が提供しています。

## 5.3 データ・マクロ

RX-NET(DNS) が提供する API 関数を発行する際に使用する各種データ・マクロ (データ・タイプ, 戻り値など) について以下に示します。

### 5.3.1 データ・タイプ

表 5-1 に, API 関数を発行する際に指定する各種パラメータのデータ・タイプ一覧を示します。

なお, データ・タイプのマクロ定義は, 標準ヘッダ・ファイル `nctools32\inc850\rxnet.h` から呼び出されるヘッダ・ファイル `nctools32\inc850\rxnet\ccdep.h` で行われています。

表 5-1 データ・タイプ

| マクロ | 型              | 意味            |
|-----|----------------|---------------|
| u8  | unsigned char  | 符号無し 8 ビット整数  |
| u16 | unsigned short | 符号無し 16 ビット整数 |
| u32 | unsigned int   | 符号無し 32 ビット整数 |

注意 標準ヘッダ・ファイル `nctools32\inc850\rxnet.h`, および, ヘッダ・ファイル `nctools32\inc850\rxnet\ccdep.h` は, RX-NET(TCP/IP) が提供しています。

### 5.3.2 戻り値

表 5-2 に, API 関数からの戻り値一覧を示します。

なお, 戻り値のマクロ定義は, 標準ヘッダ・ファイル `nctools32\inc850\rxnet.h` から呼び出されるヘッダ・ファイル `nctools32\inc850\rxnet\errno.h` で行われています。

表 5-2 戻り値

| マクロ         | 数値   | 意味                              |
|-------------|------|---------------------------------|
| ENOERR      | 0x0  | 正常終了                            |
| ENOENT      | 0x2  | ファイル, または, ディレクトリの指定が不正です       |
| ENXIO       | 0x6  | ネットワーク・デバイス名, または, アドレスの指定が不正です |
| EBADF       | 0x9  | ファイル番号の指定が不正です                  |
| ENOMEM      | 0xc  | メモリ領域が確保できません                   |
| EACCES      | 0xd  | 対象ソケットへのアクセスが拒否されました            |
| EFAULT      | 0xe  | アドレスの指定が不正です                    |
| ENODEV      | 0x13 | ネットワーク・デバイス名の指定が不正です            |
| EINVAL      | 0x16 | パラメータの指定が不正です                   |
| EMFILE      | 0x17 | 最大ソケット生成数を超過しています               |
| EWOULDBLOCK | 0x50 | 対象ソケットがノンブロッキング・モードです           |
| EINPROGRESS | 0x51 | 現在, 該当処理が実行されています               |
| EALREADY    | 0x52 | 既に該当処理が実行されています                 |
| ENOTSOCK    | 0x53 | ソケットの指定が不正です                    |
| EMSGSIZE    | 0x55 | メッセージのサイズが大きすぎます                |
| EPROTOTYPE  | 0x56 | プロトコルの指定が不正です                   |

| マクロ             | 数値   | 意味                                        |
|-----------------|------|-------------------------------------------|
| ENOPROTOOPT     | 0x57 | ソケット・オプションの指定が不正です                        |
| EPROTONOSUPPORT | 0x58 | プロトコルの指定が不正です                             |
| ESOCKTNOSUPPORT | 0x59 | サービス・タイプの指定が不正です                          |
| EOPNOTSUPP      | 0x5a | ソケット・タイプの指定が不正です                          |
| EAFNOSUPPORT    | 0x5b | アドレス・ファミリの指定が不正です                         |
| EADDRINUSE      | 0x5c | アドレスの指定が不正です                              |
| EADDRNOTAVAIL   | 0x5d | 要求アドレスを割り当てるできません                         |
| ENETDOWN        | 0x5e | ネットワークがダウンしています                           |
| ENETUNREACH     | 0x5f | ネットワークに到達できません                            |
| ECONNABORTED    | 0x61 | 接続が遮断されています                               |
| ECONNRESET      | 0x62 | ピアによる接続のリセットが行われました                       |
| ENOBUFS         | 0x63 | メッセージのサイズが大きすぎます                          |
| EISCONN         | 0x64 | 既に能動モード、または、受動モードへの移行が完了しています             |
| ENOTCONN        | 0x65 | 接続が確立していません                               |
| ESHUTDOWN       | 0x66 | 接続が遮断されているため、メッセージを送受信することができません          |
| ETIMEDOUT       | 0x67 | 接続がタイムアウトしています                            |
| ECONNREFUSED    | 0x68 | IP アドレス、ポート番号の割り付け、および、受動モードへの移行が完了していません |
| EHOSTDOWN       | 0x6a | ホスト・マシンがダウンしています                          |
| EHOSTUNREACH    | 0x6b | ホスト・マシンを捕捉することができません                      |
| ENOURGENTDATA   | 0x6c | 緊急データを受信することができません                        |

注意 標準ヘッダ・ファイル `nctools32\inc850\rxnet.h`、および、ヘッダ・ファイル `nctools32\inc850\rxnet\nermo.h` は、RX-NET(TCP/IP) が提供しています。

## 5.4 データ構造体

RX-NET(DNS) が提供する API 関数を発行する際に使用する各種データ構造体 ( ホスト情報など ) について以下に示します。

### 5.4.1 ホスト情報

以下に、ホスト情報 `hostent` を示します。

なお、ホスト情報 `hostent` の定義は、標準ヘッダ・ファイル `nctools32\inc850\rxnet_dns.h` から呼び出されるヘッダ・ファイル `nctools32\inc850\rxnet\netdb.h` で行われています。

```

struct      hostent {
            char          *h_name ;                /* ホスト名 */
            char          **h_aliases ;           /* エイリアス・リスト */
            int           h_addrtype ;           /* アドレス・ファミリ */
            int           h_length ;             /* アドレス長 ( 単位 : バイト ) */
            int           **h_addr_list ;        /* IP アドレス・リスト */
};

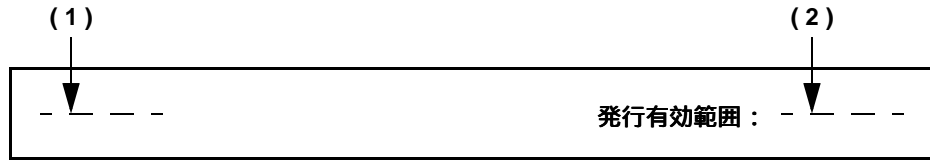
```

以下に、ホスト情報 `hostent` の詳細を示します。

- `h_name`  
“ Fully Qualified Domain Name 形式 ( ドメイン名 , サブドメイン名など省略しない形式 ) のホスト名 ” が格納されます。
- `h_aliases`  
“ エイリアス名が格納されている領域へのポインタ ” が格納されます。
- `h_addrtype`  
インターネット・アドレス `AF_INET` を意味する “ 0x2 ” が格納されます。
- `h_length`  
IPv4 仕様プロトコル・アドレス長 ( 32 ビット ) を意味する “ 0x4 ” が格納されます。
- `h_addr_list`  
“ IP アドレスが格納されている領域へのポインタ ” が格納されます。

## 5.5 API 関数解説

次項から RX-NET(DNS) が提供している API 関数について、以下の記述フォーマットに従って解説します。



(3) —▶ **概要**

-----  
 -----  
 -----

(4) —▶ **C 言語形式**

-----  
 -----  
 -----

(5) —▶ **パラメータ**

| I/O | パラメータ | 説 明 |
|-----|-------|-----|
|     |       |     |

(6) —▶ **機能**

-----  
 -----  
 -----  
 -----  
 -----

(7) —▶ **戻り値**

-----  
 -----

**( 1 ) 名称**

API 関数の名称を示しています。

**( 2 ) 発行有効範囲**

API 関数の発行が可能な処理プログラムの種別を示しています。

タスク : タスクからのみ発行可能  
 非タスク : 非タスクからのみ発行可能  
 タスク / 非タスク : タスク, 非タスクのどちらかも発行可能

**( 3 ) 概要**

API 関数の機能概要を示しています。

**( 4 ) C 言語形式**

API 関数を C 言語で記述された処理プログラムから発行する際の記述形式を示しています。

**( 5 ) パラメータ**

API 関数のパラメータを以下の形式で示しています。

| I/O | パラメータ | 説 明 |
|-----|-------|-----|
| A   | B     | C   |

**A ) パラメータの種類**

- I ... RX-NET(DNS) への入力パラメータ
- O ... RX-NET(DNS) からの出力パラメータ

**B ) パラメータのデータ・タイプ****C ) パラメータの説明****( 6 ) 機能**

API 関数の機能詳細を示しています。

**( 7 ) 戻り値**

API 関数からの戻り値をデータ・マクロ, および, 数値で示しています。

### 5.5.1 外部インタフェース仕様

表 5 に、RX-NET(DNS) が提供している API 関数の一覧を示します。

表 5-3 API 関数

| API 関数名        | 機能概要                 |
|----------------|----------------------|
| RslvInit       | RX-NET(DNS) の初期化     |
| SetNameServers | ネーム・サーバの登録           |
| SetSearchList  | ドメイン・サーチ・リストの登録      |
| GetHostByName  | ホスト名に対応したホスト情報の獲得    |
| GetHostByAddr  | IP アドレスに対応したホスト情報の獲得 |
| RslvAbort      | ホスト情報獲得処理の強制終了       |

次頁以降に、各種 API 関数の外部インタフェース仕様詳細を示します。



## RslvInit

発行有効範囲：タスク

### 概要

RX-NET(DNS) の初期化

### C 言語形式

```
int RslvInit ( void );
```

### パラメータ

なし

### 機能

RX-NET(DNS) が提供する機能を実現するうえで必要となる各種初期化処理を実行します。

なお、RX-NET(DNS) では、RX-NET(DNS) の初期化処理として、RX-NET(DNS) が提供している機能を実現する際に必要となる各種資源の生成、および、メモリ領域の確保を行っています。

注意 本 API 関数の発行は、RX-NET(TCP/IP) が提供する API 関数 `so_initialize` の処理完了後に行う必要があります。

### 戻り値

|          |      |                     |
|----------|------|---------------------|
| ENOERR   | 0x0  | 正常終了                |
| EINVAL   | 0x16 | パラメータの指定が不正です       |
| EALREADY | 0x52 | 既に本 API 関数が発行されています |
| ENETDOWN | 0x5e | ネットワークがダウンしています     |

## SetNameServers

発行有効範囲：タスク

### 概要

ネーム・サーバの登録

### C 言語形式

```
int SetNameServers ( u32 dns_addr1 , u32 dns_addr2 );
```

### パラメータ

| I/O | パラメータ                  | 説明                                         |
|-----|------------------------|--------------------------------------------|
| I   | u32 <i>dns_addr1</i> ; | プライマリ・ネーム・サーバの IP アドレス ( ネットワーク・バイト・オーダー ) |
| I   | u32 <i>dns_addr2</i> ; | セカンダリ・ネーム・サーバの IP アドレス ( ネットワーク・バイト・オーダー ) |

### 機能

*dns\_addr1* , および , *dns\_addr2* で指定されたネーム・サーバを RX-NET(DNS) に登録します。

注意 セカンダリ・ネーム・サーバが存在しない場合には , *dns\_addr2* に 0x0 を設定します。

### 戻り値

|        |      |               |
|--------|------|---------------|
| ENOERR | 0x0  | 正常終了          |
| EINVAL | 0x16 | パラメータの指定が不正です |

# SetSearchList

発行有効範囲：タスク

## 概要

ドメイン・サーチ・リストの登録

## C 言語形式

```
int SetSearchList ( char *domain_list );
```

## パラメータ

| I/O | パラメータ                       | 説明                        |
|-----|-----------------------------|---------------------------|
| I   | char * <i>domain_list</i> ; | ドメイン・サーチ・リストを格納した領域へのポインタ |

## 機能

*domain\_list* で指定されたドメイン・サーチ・リスト (最大検索要素数: 0x6) を RX-NET(DNS) に登録します。

注意 1 RX-NET(DNS) では、ドメイン・サーチ・リスト *domain\_list* に設定されている検索要素を利用する際、ポインタ参照しています。このため、ドメイン・サーチ・リスト *domain\_list* は、外部変数、または、静的変数として定義する必要があります。

注意 2 ドメイン・サーチ・リスト *domain\_list* の設定例を以下に示します。  
なお、ドメイン・サーチ・リスト *domain\_list* の終端には、ドメイン・サーチ・リストの終端を意味する 0x0 を設定する必要があります。

### 【 検索要素数が 3 個の場合 】

```
char domain_list[] = {
    "sample1.nec.co.jp", /* 第 1 検索要素 */
    "sample2.nec.co.jp", /* 第 2 検索要素 */
    "sample3.nec.co.jp", /* 第 3 検索要素 */
    0x0 /* 終端 */
};
```

### 【 検索要素数が 6 個の場合 】

```
char domain_list[] = {
    "sample1.nec.co.jp", /* 第 1 検索要素 */
    "sample2.nec.co.jp", /* 第 2 検索要素 */
    "sample3.nec.co.jp", /* 第 3 検索要素 */
    "sample4.nec.co.jp", /* 第 4 検索要素 */
    "sample5.nec.co.jp", /* 第 5 検索要素 */
    "sample6.nec.co.jp", /* 第 6 検索要素 */
    0x0 /* 終端 */
};
```

## 戻り値

|        |      |               |
|--------|------|---------------|
| ENOERR | 0x0  | 正常終了          |
| EINVAL | 0x16 | パラメータの指定が不正です |

# GetHostByName

発行有効範囲：タスク

## 概要

ホスト名に対応したホスト情報の獲得

## C 言語形式

```
struct hostent GetHostByName ( char *name , int *errp );
```

## パラメータ

| I/O | パラメータ        | 説明                   |
|-----|--------------|----------------------|
| I   | char *name ; | ホスト名を格納した領域へのポインタ    |
| O   | int *errp ;  | エラー・コードを格納する領域へのポインタ |

## 機能

*name* で指定されたホスト名に対応したホスト情報を獲得します。

なお、本 API 関数の処理が正常終了した際には、“対応するホスト情報へのポインタ”が戻り値として返されます。

- 注意 1 本 API 関数を発行する際には、マルチタスク処理を考慮した記述 (API 関数の発行に伴うタスク間の排他制御) が必要となります。
- 注意 2 ホスト名 *name* の設定例を以下に示します。

【ホスト名のみを設定する場合】

```
char name [] = "www";
```

【ホスト名、および、ドメイン名を設定する場合】

```
char name [] = "www.sample.nec.co.jp";
```

- 注意 3 本 API 関数の発行から 150 秒が経過するまでの間にホスト情報を獲得することが出来なかった場合、*errp* で指定された領域に ETIMEDOUT が格納されます。

- 注意 4 *errp* で指定された領域には、以下に示したエラー・コードが格納されます。

ENOERR (0x0) : 正常終了

EINVAL (0x16) : パラメータの指定が不正です

ECONNABORTED (0x61) : ホスト情報獲得処理を実行中に他タスクから RslvAbort が発行された

ETIMEDOUT (0x67) : 接続がタイムアウトしています

ECONNREFUSED (0x68) : ネーム・サーバとの接続処理が失敗したため、ホスト情報を獲得することが出来なかった

- 注意 5 ホスト情報 *hostent* についての詳細は、「5.4.1 ホスト情報」を参照してください。

## 戻り値

- NULL 以外 正常終了
- 対応するホスト情報へのポインタが返されます
- NULL 異常終了
- 対応するエラー・コードが *errp* で指定された領域に格納されます

# GetHostByAddr

発行有効範囲：タスク

## 概要

IP アドレスに対応したホスト情報の獲得

## C 言語形式

```
struct hostent GetHostByAddr ( u32 ipaddr , int *errp );
```

## パラメータ

| I/O | パラメータ               | 説明                          |
|-----|---------------------|-----------------------------|
| I   | u32 <i>ipaddr</i> ; | IP アドレス ( ネットワーク・バイト・オーダー ) |
| O   | int <i>*errp</i> ;  | エラー・コードを格納する領域へのポインタ        |

## 機能

*ipaddr* で指定された IP アドレスに対応したホスト情報を獲得します。

なお、本 API 関数の処理が正常終了した際には、“対応するホスト情報へのポインタ” が戻り値として返されます。

注意 1 本 API 関数を発行する際には、マルチタスク処理を考慮した記述 (API 関数の発行に伴うタスク間の排他制御) が必要となります。

注意 2 本 API 関数の発行から 150 秒が経過するまでの間にホスト情報を獲得することが出来なかった場合、*errp* で指定された領域に ETIMEDOUT が格納されます。

注意 3 *errp* で指定された領域には、以下に示したエラー・コードが格納されます。

|              |        |                                            |
|--------------|--------|--------------------------------------------|
| ENOERR       | (0x0)  | : 正常終了                                     |
| EINVAL       | (0x16) | : パラメータの指定が不正です                            |
| ECONNABORTED | (0x61) | : ホスト情報獲得処理を実行中に他タスクから RslvAbort が発行された    |
| ETIMEDOUT    | (0x67) | : 接続がタイムアウトしています                           |
| ECONNREFUSED | (0x68) | : ネーム・サーバとの接続処理が失敗したため、ホスト情報を獲得することが出来なかった |

注意 4 ホスト情報 *hostent* についての詳細は、「5.4.1 ホスト情報」を参照してください。

## 戻り値

|         |                                                    |
|---------|----------------------------------------------------|
| NULL 以外 | 正常終了<br>- 対応するホスト情報へのポインタが返されます                    |
| NULL    | 異常終了<br>- 対応するエラー・コードが <i>errp</i> で指定された領域に格納されます |

## RslvAbort

発行有効範囲：タスク / 非タスク

### 概要

ホスト情報獲得処理の強制終了

### C 言語形式

```
void RslvAbort ( void );
```

### パラメータ

なし

### 機能

本 API 関数を発行した際、処理の完了していない API 関数 (GetHostByName, GetHostByAddr) を強制的に異常終了させます。

なお、本 API 関数の発行により異常終了した API 関数には、エラー・コードとして ECONNABORTED が設定されます。

**注意** 本 API 関数では、終了要求のキューイングが行われません。このため、ホスト情報獲得処理を実行中のタスクが存在しなかった場合には、何も処理は行わず、エラーとしても扱いません。

### 戻り値

なし

# 索引

## A

|                      |                |
|----------------------|----------------|
| API 関数 .....         | 26, 32, 38     |
| GetHostByAddr .....  | 30, 32, 38, 43 |
| GetHostByName .....  | 29, 32, 38, 42 |
| RslvAbort .....      | 31, 32, 38, 44 |
| RslvInit .....       | 26, 32, 38, 39 |
| SetNameServers ..... | 27, 32, 38, 40 |
| SetSearchList .....  | 28, 32, 38, 41 |
| 外部インタフェース仕様 .....    | 38             |
| データ構造体 .....         | 35             |
| データ・マクロ .....        | 33             |
| 呼び出し方法 .....         | 32             |

## C

|                 |    |
|-----------------|----|
| cf850pro .....  | 21 |
| CF 定義ファイル ..... | 21 |
| 情報ファイル .....    | 21 |

## D

|                      |        |
|----------------------|--------|
| DNS リゾルバ・ライブラリ ..... | 18, 19 |
| libdns_rslv.a .....  | 18, 19 |

## F

|                   |    |
|-------------------|----|
| fnsconfig.h ..... | 32 |
|-------------------|----|

## G

|                     |                |
|---------------------|----------------|
| GetHostByAddr ..... | 30, 32, 38, 43 |
| GetHostByName ..... | 29, 32, 38, 42 |

## L

|                     |        |
|---------------------|--------|
| libdns_rslv.a ..... | 18, 19 |
|---------------------|--------|

## M

|                |    |
|----------------|----|
| Makefile ..... | 18 |
|----------------|----|

## R

|                     |                |
|---------------------|----------------|
| README.DNS .....    | 18, 19         |
| RFC .....           | 14             |
| RslvAbort .....     | 31, 32, 38, 44 |
| RslvInit .....      | 26, 32, 38, 39 |
| RX850 Pro 依存部 ..... | 22             |
| エントリ処理 .....        | 22             |
| 初期化ハンドラ .....       | 22             |
| ハードウェア初期化部 .....    | 22             |

|                                  |            |
|----------------------------------|------------|
| ブート処理 .....                      | 22         |
| RX850 Pro 用標準ヘッダ・ファイル .....      | 32         |
| stdrx85p.h .....                 | 32         |
| RX-NET(DNS) .....                | 13         |
| API 関数 .....                     | 32         |
| アドレス解決機能 .....                   | 26         |
| 位置付け .....                       | 13         |
| インストール .....                     | 16         |
| 階層的位置付け .....                    | 14         |
| 開発環境 .....                       | 15         |
| システム構築 .....                     | 20         |
| 実行環境 .....                       | 15         |
| 特徴 .....                         | 14         |
| rxnet_dns.h .....                | 18, 19, 32 |
| RX-NET(DNS) 用標準ヘッダ・ファイル .....    | 18, 19, 32 |
| rxnet_dns.h .....                | 18, 19, 32 |
| rxnet.h .....                    | 32         |
| RX-NET(TCP/IP) 依存部 .....         | 23         |
| RX-NET(TCP/IP) 用標準ヘッダ・ファイル ..... | 32         |
| rxnet.h .....                    | 32         |

## S

|                      |                |
|----------------------|----------------|
| sample.blb .....     | 19             |
| SetNameServers ..... | 27, 32, 38, 40 |
| SetSearchList .....  | 28, 32, 38, 41 |
| stdrx85p.h .....     | 32             |

## あ

|                    |    |
|--------------------|----|
| アーカイブ・オブジェクト ..... | 24 |
| アドレス解決機能 .....     | 26 |

## い

|                   |    |
|-------------------|----|
| インストレーション .....   | 16 |
| UNIX ベース .....    | 17 |
| Windows ベース ..... | 16 |

## え

|              |    |
|--------------|----|
| エントリ処理 ..... | 22 |
|--------------|----|

## お

|                   |    |
|-------------------|----|
| オブジェクト・ファイル ..... | 24 |
|-------------------|----|

## か

|                                |    |
|--------------------------------|----|
| 開発環境 .....                     | 15 |
| ソフトウェア .....                   | 15 |
| ハードウェア .....                   | 15 |
| 外部インタフェース仕様 .....              | 38 |
| 拡張 SVC ハンドラ .....              | 23 |
| 拡張 SVC ハンドラ用インタフェース・ルーチン ..... | 23 |
| 間接起動割り込みハンドラ .....             | 23 |

## こ

|                 |    |
|-----------------|----|
| コンフィギュレータ ..... | 21 |
| cf850pro .....  | 21 |

## し

|                          |    |
|--------------------------|----|
| システム構築 .....             | 20 |
| CF 定義ファイル .....          | 21 |
| RX850 Pro 依存部 .....      | 22 |
| RX-NET(TCP/IP) 依存部 ..... | 23 |
| アーカイブ・オブジェクト .....       | 24 |
| オブジェクト・ファイル .....        | 24 |
| 処理プログラム .....            | 23 |
| リンク・ディレクティブ・ファイル .....   | 24 |
| ロード・モジュール .....          | 25 |
| システム・コール・テーブル .....      | 21 |
| システム情報テーブル .....         | 21 |
| システム情報ヘッダ・ファイル .....     | 21 |
| 実行環境 .....               | 15 |
| 周辺コントローラ .....           | 15 |
| プロセッサ .....              | 15 |
| メモリ容量 .....              | 15 |
| 周期起動ハンドラ .....           | 23 |
| 情報ファイル .....             | 21 |
| システム・コール・テーブル .....      | 21 |

|                                |    |
|--------------------------------|----|
| システム情報テーブル .....               | 21 |
| システム情報ヘッダ・ファイル .....           | 21 |
| 初期化ハンドラ .....                  | 22 |
| 処理プログラム .....                  | 23 |
| 拡張 SVC ハンドラ .....              | 23 |
| 拡張 SVC ハンドラ用インタフェース・ルーチン ..... | 23 |
| 間接起動割り込みハンドラ .....             | 23 |
| 周期起動ハンドラ .....                 | 23 |
| タスク .....                      | 23 |
| 直接起動割り込みハンドラ .....             | 23 |

## せ

|                      |    |
|----------------------|----|
| 静的設定情報ヘッダ・ファイル ..... | 32 |
| fnsconfig.h .....    | 32 |

## た

|           |    |
|-----------|----|
| タスク ..... | 23 |
|-----------|----|

## ち

|                    |    |
|--------------------|----|
| 直接起動割り込みハンドラ ..... | 23 |
|--------------------|----|

## て

|                   |        |
|-------------------|--------|
| ディレクトリ構成 .....    | 18     |
| CA850 対応版 .....   | 18     |
| CCV850E 対応版 ..... | 19     |
| データ構造体 .....      | 35     |
| ホスト情報 .....       | 35     |
| データ・タイプ .....     | 33     |
| データ・マクロ .....     | 33     |
| データ・タイプ .....     | 33     |
| 戻り値 .....         | 33     |
| テキスト・ファイル .....   | 18, 19 |
| README.DNS .....  | 18, 19 |

## は

|                  |    |
|------------------|----|
| ハードウェア初期化部 ..... | 22 |
|------------------|----|

## ひ

|                  |    |
|------------------|----|
| ビルド・ファイル .....   | 19 |
| sample.bld ..... | 19 |

## ふ

|             |    |
|-------------|----|
| ブート処理 ..... | 22 |
|-------------|----|

## ほ

|             |    |
|-------------|----|
| ホスト情報 ..... | 35 |
|-------------|----|



## め

|                |    |
|----------------|----|
| メイク・ファイル ..... | 18 |
| Makefile ..... | 18 |

## も

|           |    |
|-----------|----|
| 戻り値 ..... | 33 |
|-----------|----|

## り

|                        |    |
|------------------------|----|
| リンク・ディレクティブ・ファイル ..... | 24 |
|------------------------|----|

## ろ

|                 |    |
|-----------------|----|
| ロード・モジュール ..... | 25 |
|-----------------|----|

---

## — お問い合わせ先 —

### 【技術的なお問い合わせ先】

NEC半導体テクニカルホットライン  
(電話：午前 9:00～12:00，午後 1:00～5:00)

電話 : 044-435-9494  
FAX : 044-435-9608  
E-mail : [info@lsi.nec.co.jp](mailto:info@lsi.nec.co.jp)

### 【営業関係お問い合わせ先】

#### システムLSI第一営業事業部

東京 (03)3798-6106, 6107, 6108, 6155  
大阪 (06)6945-3178, 3200, 3208  
名古屋 (052)222-2375  
仙台 (022)267-8740  
水戸 (029)226-1702  
広島 (082)242-5504  
鳥取 (0857)27-5313  
松山 (089)945-4149

#### システムLSI第二営業事業部

東京 (03)3798-6110, 6111, 6112, 6151, 6156  
名古屋 (052)222-2170, 2190  
松本 (0263)35-1662  
前橋 (027)243-6060  
立川 (042)526-5981  
静岡 (054)254-4794  
金沢 (076)232-7303  
福岡 (092)261-2806

### 【資料の請求先】

上記営業関係お問い合わせ先またはNEC特約店へお申しつけください。

### 【NECエレクトロニクス デバイス ホームページ】

NECエレクトロニクスデバイスの情報がインターネットでご覧になれます。

URL(アドレス) <http://www.ic.nec.co.jp/>

## アンケート記入のお願い

お手数ですが、このドキュメントに対するご意見をお寄せください。今後のドキュメント作成の参考にさせていただきます。

[ドキュメント名] RX-NET ネットワーク・ライブラリ DNS Ver.1.30 ユーザーズ・マニュアル  
( U15304JJ3V0UM00 ( 第3版 ) )

[お名前など] ( さしつかえのない範囲で )

御社名 ( 学校名 , その他 ) ( )  
ご住所 ( )  
お電話番号 ( )  
お仕事の内容 ( )  
お名前 ( )

1. ご評価 ( 各欄に をご記入ください )

| 項 目            | 大変良い | 良 い | 普 通 | 悪 い | 大変悪い |
|----------------|------|-----|-----|-----|------|
| 全体の構成          |      |     |     |     |      |
| 説明内容           |      |     |     |     |      |
| 用語解説           |      |     |     |     |      |
| 調べやすさ          |      |     |     |     |      |
| デザイン , 字の大きさなど |      |     |     |     |      |
| その他 ( )        |      |     |     |     |      |
| ( )            |      |     |     |     |      |

2. わかりやすい所 ( 第 章 , 第 章 , 第 章 , 第 章 , その他 )

理由 [ ]

3. わかりにくい所 ( 第 章 , 第 章 , 第 章 , 第 章 , その他 )

理由 [ ]

4. ご意見 , ご要望

5. このドキュメントをお届けしたのは

NEC販売員 , 特約店販売員 , その他 ( )

ご協力ありがとうございました。

下記あてにFAXで送信いただくか , 最寄りの販売員にコピーをお渡しください。

日本電気(株) NEC エレクトロニクス  
半導体テクニカルホットライン

FAX : ( 044 ) 435-9608

2000.6