

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

PPP 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

(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 ...	22
第4章 ネットワーク接続機能 ...	29
第5章 API関数 ...	41
第6章 RX-NET (PPP) 依存部 ...	56
索引 ...	111

V850 シリーズ, V853, V850/SA1, V850/SB1, V850/SB2, 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。

本製品は外国為替および外国貿易管理法の規定により規制貨物等（または役務）に該当しますので、日本国外に輸出する場合には、同法に基づき日本国政府の輸出許可が必要です。

- 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
ネットワーク接続機能
API関数
RX-NET (PPP) 依存部

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・エミュレータ)	U11595J	U11595E

資料名	資料番号		
	和文	英文	
IE-V850E-MC (V850E/IA1™,V850E/IA2™用インサーキット・エミュレータ), IE-V850E-MC-A (V850E/MA1™,V850E/MA2™用インサーキット・エミュレータ)	U14487J	U14487E	
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
ID850 Ver.2.50 統合ディバग्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
	インストレーション編	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
	インストレーション編	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	このマニュアル	-	
RX-NET ネットワーク・ライブラリ (DNS)	U15304J	-	
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ग्ガ	U13737J	U13737E	
RD850 Pro Ver.3.01 タスク・ディバग्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-FP3 フラッシュ・メモリ・プログラマ	U13502J	U13502E	
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	20
第 3 章 システム構築	22
3.1 概要	22
3.2 CF 定義ファイルの記述	23
3.3 情報ファイルの生成	23
3.4 RX850 Pro 依存部の記述	24
3.5 RX-NET(TCP/IP) 依存部の記述	25
3.6 RX-NET(PPP) 依存部の記述	25
3.7 処理プログラムの記述	26
3.8 オブジェクト・ファイルの生成	26
3.9 アーカイブ・オブジェクトの生成	27
3.10 リンク・ディレクティブ・ファイルの記述	27
3.11 ロード・モジュールの生成	27
第 4 章 ネットワーク接続機能	29
4.1 概要	29
4.2 認証プロトコル	29
4.3 接続形態	30
4.4 通信オプション	32
4.4.1 LCP 通信設定要求	32
4.4.2 LCP 通信設定要求応答	32
4.4.3 IPCP 通信設定要求	32
4.4.4 IPCP 通信設定要求応答	33
4.5 ネットワーク接続機能 API 関数	34
4.5.1 Point-to-Point プロトコルによる接続の確立 / 遮断	34
4.5.2 パスワード・データの登録	37
4.5.3 Challenge メッセージの登録	38
4.5.4 IP アドレスの獲得	39
第 5 章 API 関数	41
5.1 概要	41
5.2 API 関数の呼び出し	41

5.3	データ・マクロ	42
5.3.1	データ・タイプ	42
5.3.2	接続条件	42
5.3.3	戻り値	43
5.4	データ構造体	44
5.4.1	モデム制御シーケンス情報	44
5.5	API 関数解説	46
5.5.1	外部インタフェース仕様	48
第 6 章	RX-NET(PPP) 依存部	56
6.1	概要	56
6.2	基本情報	56
6.2.1	デバイス・ドライバ・エントリ・テーブル情報	56
6.3	データ・マクロ	57
6.3.1	データ・タイプ	57
6.4	ドライバ関数解説	59
6.4.1	外部インタフェース仕様	61
索引	68

目次

図 1-1	RX-NET(PPP) の位置付け	13
図 1-2	RX-NET(PPP) 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)	20
図 3-1	システム構築手順	22
図 3-2	RX850 Pro 依存部の処理の流れ	24
図 4-1	ヌルモデム接続	30
図 4-2	モデム接続	31

表目次

表 2-1	RX-NET(PPP) の提供形式	16
表 3-1	ドライバ関数	25
表 5-1	データ・タイプ	42
表 5-2	接続条件	42
表 5-3	戻り値	43
表 5-4	API 関数	48
表 6-1	データ・タイプ	57
表 6-2	ドライバ関数	61

第1章 概説

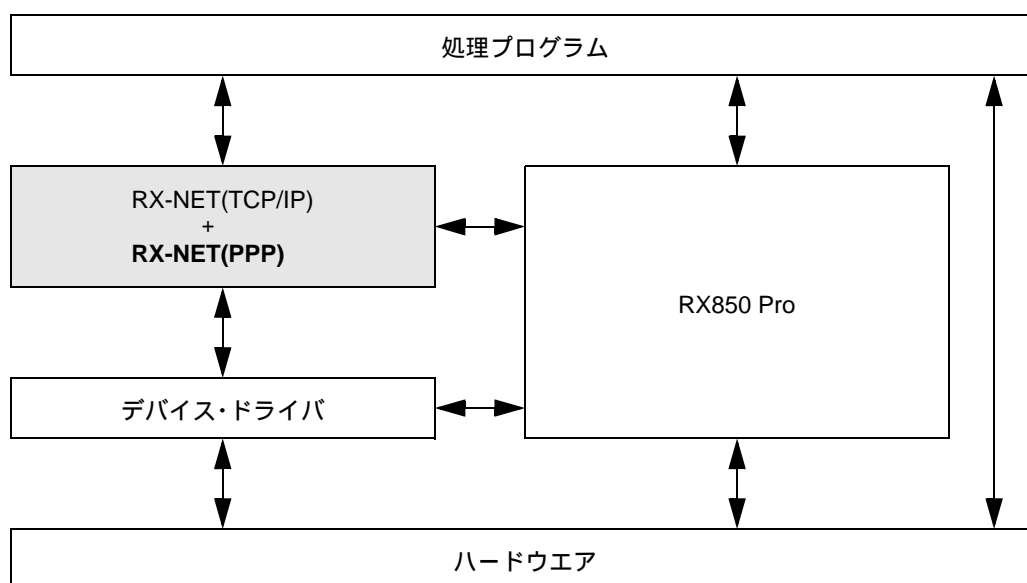
1.1 概要

RX-NET(PPP) 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、シリアル回線 (電話回線、RS-232C クロス・ケーブルなど) 上に確立された Point-to-Point リンクを介したネットワーク接続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(PPP) が提供する API 関数を利用することにより、Point-to-Point プロトコルを利用したネットワーク接続を実現することが可能となります。

図 1-1 に、RX-NET(PPP) の位置付けを示します。

図 1-1 RX-NET(PPP) の位置付け



1.2 特徴

以下に、RX-NET(PPP) の特徴を示します。

- RFC に準拠

RX-NET(PPP) では、RFC に準拠した設計が行われています。

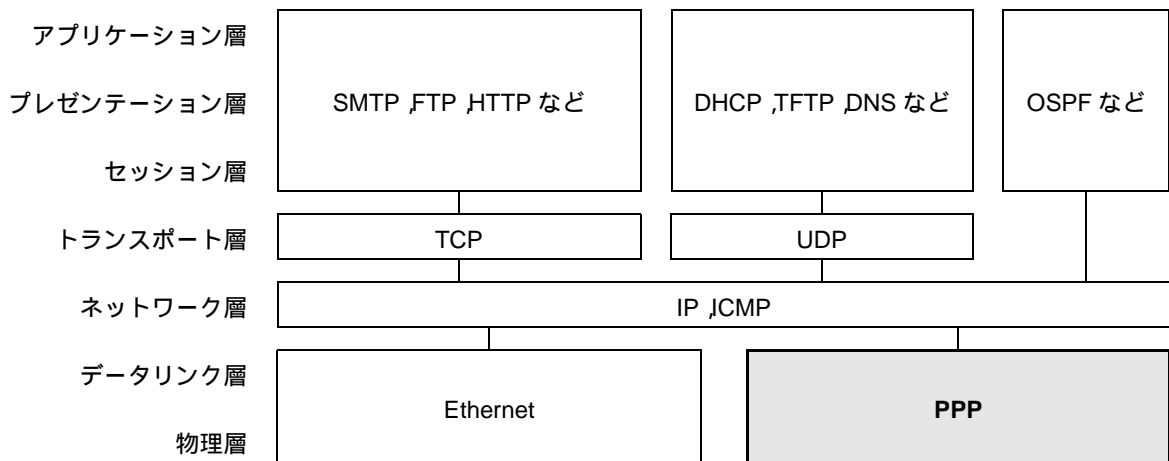
なお、RFC とは、インターネットに関する研究開発機関 IETF (Internet Engineering Task Force) が取りまとめた公開技術文書であり、電子メール、ファイル転送などのプロトコル仕様（情報交換を行う際に必要な手順、および、規約）の他にも、各種サービス、ガイドラインといった多岐に渡った情報（ネットワーク技術の実装と運用に主眼を置いた情報）が記載されています。

- ネットワーク接続機能をサポート

RX-NET(PPP) では、インターネット上で利用される Point-to-Point プロトコル用の API 関数を提供しています。したがって、RX-NET(PPP) 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(PPP) の階層的位置付けを示します。

図 1-2 RX-NET(PPP) の階層的位置付け



- 認証プロトコルをサポート

RX-NET(PPP) では、ネットワーク接続の確立要求を発行しているユーザに対するアクセス権の有無を検査する機能（セキュリティ機能）として以下に示した 3 種類をサポートしています。

- LOGIN 認証
- PAP (Password Authentication Protocol) 認証
- CHAP (Challenge Handshake Authentication Protocol) 認証

- 圧縮オプションをサポート

RX-NET(PPP) では、Point-to-Point プロトコル用圧縮オプションとして以下に示した 3 種類をサポートしています。

- PPP ヘッダの Address/Control フィールド圧縮
- PPP ヘッダのプロトコル・フィールド圧縮
- TCP/IP ヘッダの Van Jacobson TCP ヘッダ圧縮

- 高い移植性

RX-NET(PPP) が処理を実行するうえで必要となるユーザの実行環境 / アプリケーション・システムに依存した処理については、RX-NET(PPP) 依存部（ユーザ・OWN・コーディング部）として切り出し、サンプル・ソース・ファイルを提供しています。

これにより、実行環境への移植性を向上させるとともに、カスタマイズ化を容易なものとしています。

- マルチタスク処理を意識した設計

RX-NET(PPP) が提供する API 関数では、マルチタスク処理を考慮した設計が行われています。このため、ユーザが処理プログラムを記述する際、API 関数の発行に伴うタスク間の排他制御などを意識する必要がありません。

1.3 実行環境

以下に、RX-NET(PPP) 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(PPP) 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

- 周辺コントローラ

RX-NET(PPP) 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(PPP) 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(PPP) のテキスト領域 : 約 30 K バイト

RX-NET(PPP) のデータ領域 : 約 5 K バイト

1.4 開発環境

以下に、RX-NET(PPP) 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(PPP)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(PPP)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(PPP)の提供形式一覧を示します。

表2-1 RX-NET(PPP)の提供形式

ホスト・マシン	提供形式	提供媒体
Windowsベース • PC-9821シリーズ • PC-98NXシリーズ • IBM-PC/AT互換機	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	CD-ROM
UNIXベース • SPARC station	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	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(PPP)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(PPP)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(PPP)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(PPP)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(PPP)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(PPP)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(PPP)の提供媒体のディレクトリRX-NET_PPP_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(PPP)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(PPP)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(PPP)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/rxnetppp.tar を、米国 Green Hills Software, Inc. 製 CCV850E の場合は圧縮ファイル ghs/rxnetppp.tar を展開します。

【 CA850 対応版の場合 】

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

【 CCV850E 対応版の場合 】

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

- 6) ファイル群の確認
ls コマンドを実行し、RX-NET(PPP)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

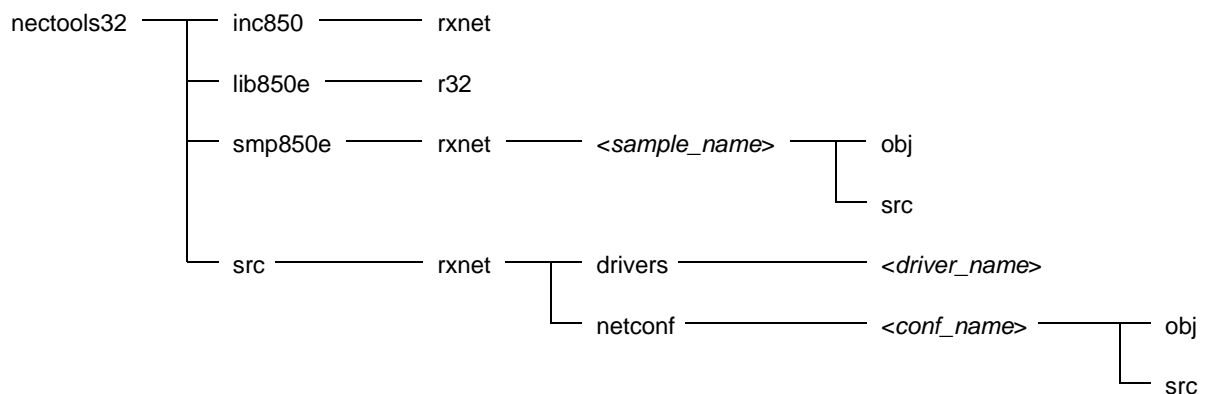
RX-NET(PPP) の提供媒体に格納されているファイル群のディレクトリ構成は、ユーザが使用する C コンパイラ・パッケージの種類 (NEC 製 CA850, 米国 Green Hills Software, Inc. 製 CCV850E) により異なります。

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(PPP) 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(PPP) 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_ppp.h : RX-NET(PPP) 用標準ヘッダ・ファイル
- 2) nectools32\inc850\rxnet
RX-NET(PPP) 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
PPP 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。
libppp.a : PPP ライブラリ
rxnetcfg_<driver_name>.o : デバイス・ドライバ・オブジェクト
- 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。
- 6) nectools32\src\rxnet\drivers\<driver_name>
ネットワーク物理層に該当するデバイス・ドライバのソース・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。
- 7) nectools32\src\rxnet\netconf\<conf_name>\obj
デバイス・ドライバ・オブジェクト (ネットワーク物理層に該当するデバイス・ドライバ、コンフィギュレーション情報依存処理部) 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、デバイス・ドライバ・オブジェクト rxnetcfg_<driver_name>.o が本ディレクトリ、および、ディレクトリ nectools32\lib850e\r32 に生成されます。

Makefile : デバイス・ドライバ・オブジェクト用メイク・ファイル

- 8) nectools32\src\rxnet\netconf*<conf_name>*\src
コンフィギュレーション情報依存処理部のソース・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。

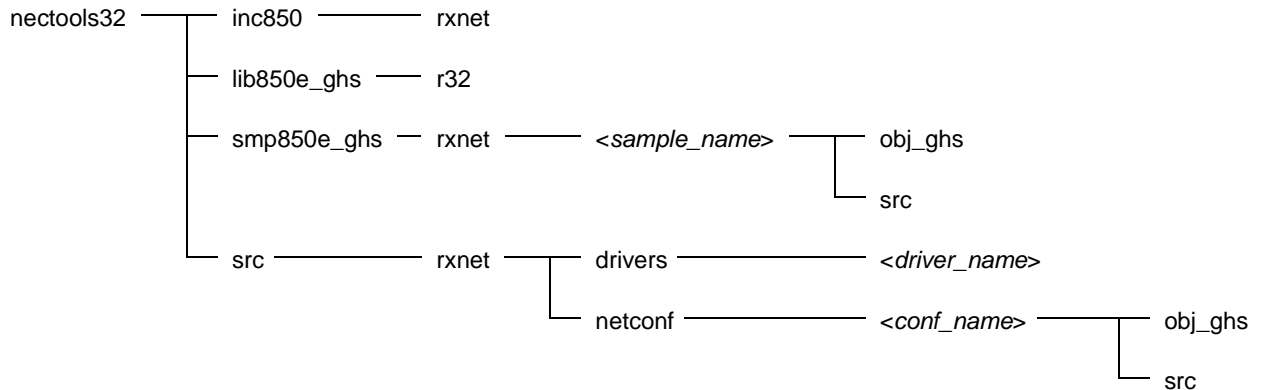
注意 *<sample_name>*, *<driver_name>*, *<conf_name>* についての詳細は, 下記に示したテキスト・ファイルを参照してください。

<sample_name> : nectools32\smp850e\rxnet\README.PPP
<driver_name> : nectools32\src\rxnet\drivers\README.PPP
<conf_name> : nectools32\src\rxnet\netconf\README.PPP

2.3.2 CCV850E 対応版

図 2-2 に、RX-NET(PPP) 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(PPP) 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_ppp.h : RX-NET(PPP) 用標準ヘッダ・ファイル
- 2) nectools32\inc850\rxnet
RX-NET(PPP) 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
PPP 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。
libppp.a : PPP ライブラリ
rxnetcfg_<driver_name>.o : デバイス・ドライバ・オブジェクト
- 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。
- 6) nectools32\src\rxnet\drivers\<driver_name>
ネットワーク物理層に該当するデバイス・ドライバのソース・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。
- 7) nectools32\src\rxnet\netconf\<conf_name>\obj_ghs
デバイス・ドライバ・オブジェクト (ネットワーク物理層に該当するデバイス・ドライバ、コンフィギュレーション情報依存処理部) を生成するためのビルド・ファイル netconf.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 netconf.bld を用いることにより、デバイス・ドライバ・オブジェクト rxnetcfg_<driver_name>.o が本ディレクトリ、および、ディレクトリ nectools32\lib850e_ghs\r32 に生成されます。
netconf.bld : デバイス・ドライバ・オブジェクト用ビルド・ファイル
- 8) nectools32\src\rxnet\netconf\<conf_name>\src
コンフィギュレーション情報依存処理部のソース・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>、<driver_name>、<conf_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.PPP

<driver_name> : nectools32\src\rxnet\drivers\README.PPP
<conf_name> : nectools32\src\rxnet\netconf\README.PPP

第3章 システム構築

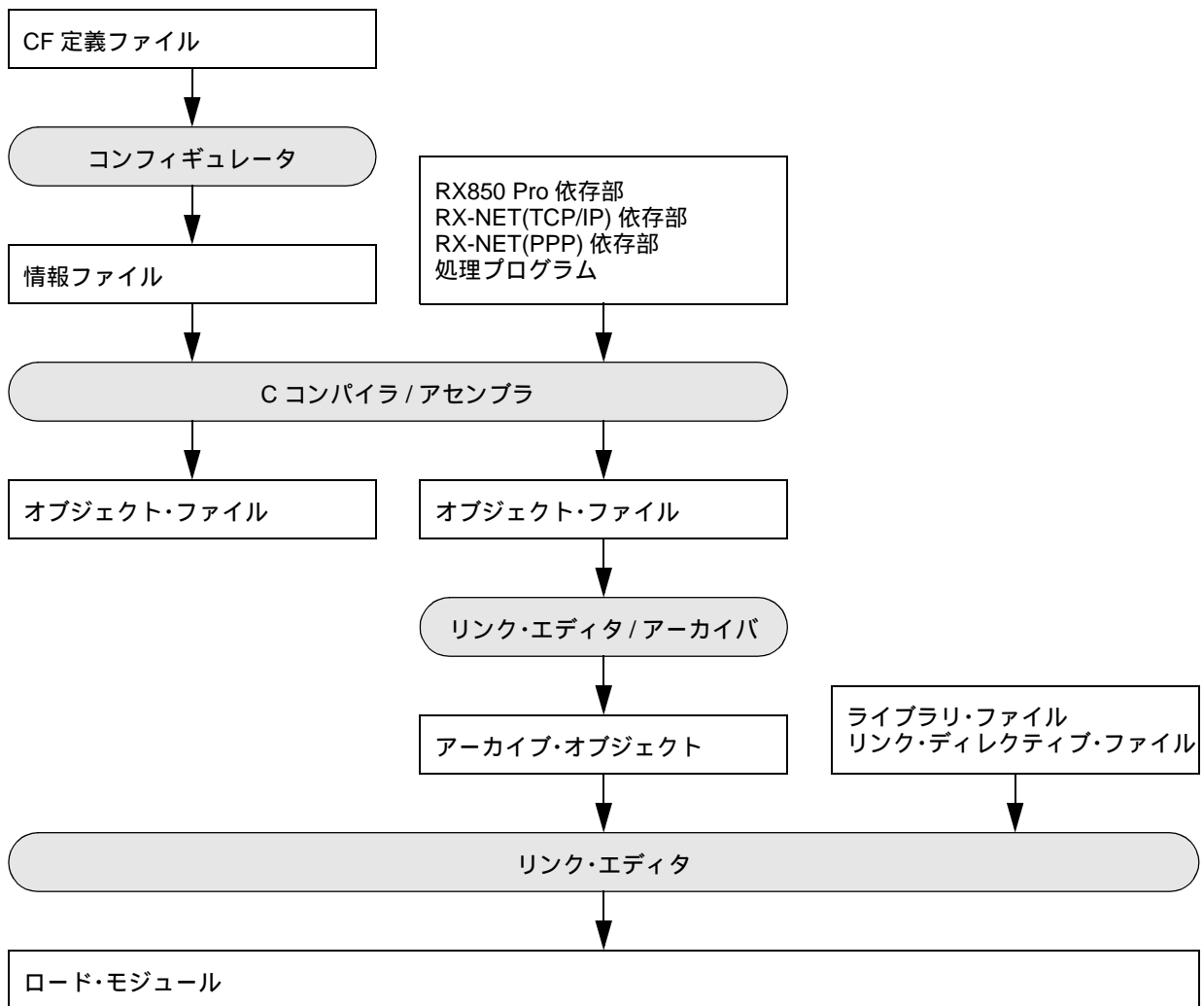
本章では、RX-NET(PPP)を使用したネットワーク・アプリケーション(ロード・モジュール)の構築手順を解説しています。

3.1 概要

システム構築とは、RX-NET(PPP)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(PPP)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。

なお, RX-NET(PPP) 依存部では 1 個のタスク, 1 個の間接起動割り込みハンドラ, 10 種類のシステム・コールを利用して各種機能を実現しています。

- SIT 情報
 - システム情報
RX-NET(PPP) 依存部用に “タスクの自動割り付け ID 番号” として 1 個分を確保。
 - システム最大値情報
RX-NET(PPP) 依存部用に “タスクの最大生成数” として 1 個分を確保。
RX-NET(PPP) 依存部用に “間接起動割り込みハンドラの最大登録数” として 1 個分を確保。
- SCT 情報
 - タスク管理機能情報
RX-NET(PPP) 依存部用に “cre_tsk, sta_tsk, ext_tsk” を定義。
 - タスク付属同期機能情報
RX-NET(PPP) 依存部用に “slp_tsk, tslp_tsk, wup_tsk” を定義。
 - 割り込み処理管理機能情報
RX-NET(PPP) 依存部用に “def_int, ret_wup, dis_int, ena_int” を定義。

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(PPP) 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(PPP) 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
```

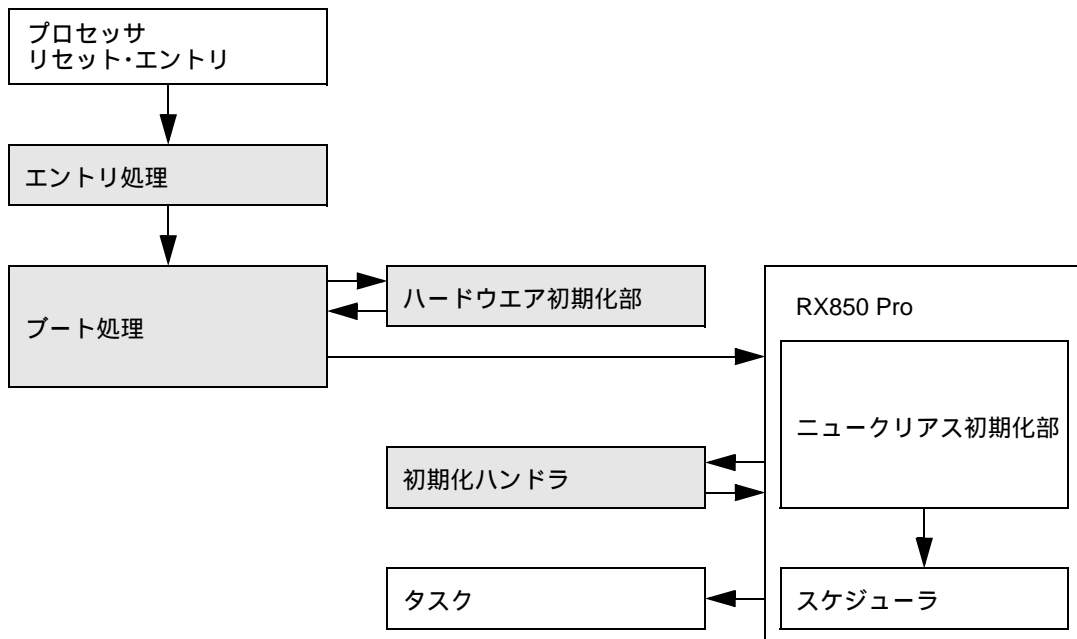
3.4 RX850 Pro 依存部の記述

RX-NET(PPP) 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(PPP) 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\\src
 varfunc.c : 初期化ハンドラ

【CCV850E 対応版の場合】
 nectools32\smp850e_ghs\rxnet\\src
 varfunc.c : 初期化ハンドラ

3.5 RX-NET(TCP/IP) 依存部の記述

RX-NET(PPP) では、RX-NET(TCP/IP) が提供する機能を利用して各種ネットワーク接続機能を実現しています。したがって、RX-NET(PPP) 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・オウン・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 RX-NET(PPP) 依存部の記述

RX-NET(PPP) では、RX-NET(PPP) が提供する機能を実現する際に必要となる基本情報、および、ドライバ関数を RX-NET(PPP) 依存部 (ユーザ・オウン・コーディング部) として切り出しています。

したがって、RX-NET(PPP) が提供する機能を利用した処理プログラムを作成する場合、RX-NET(PPP) を正常に動作させるうえで必要となる RX-NET(PPP) 依存部 (ユーザ・オウン・コーディング部) の記述が必要となります。

以下に、RX-NET(PPP) 依存部の一覧を示します。

- 基本情報

RX-NET(PPP) では、RX-NET(PPP) が提供する機能を実現する際に必要となる基本情報 (デバイス・ドライバ・エントリ・テーブル情報など) をユーザ・オウン・コーディング部として切り出しています。

以下に、RX-NET(PPP) のユーザ・オウン・コーディング部として切り出される基本情報の一覧を示します。

- デバイス・ドライバ・エントリ・テーブル情報

- ドライバ関数

RX-NET(PPP) では、Point-to-Point プロトコルによる接続の確立処理 / 遮断処理、RX-NET(PPP) の送信処理 / 受信処理などを実行した際に呼び出すドライバ関数をユーザ・オウン・コーディング部として切り出しています。

表 3-1 に、ユーザ・オウン・コーディング部として切り出されるドライバ関数の一覧を示します。

表 3-1 ドライバ関数

ドライバ関数名	機能概要
serial_updown	ppp_connect , または , ppp_disconnect を発行した際に呼び出される開始 / 終了関数
serial_write	RX-NET(PPP) が送信処理を実行した際に呼び出される送信関数
serial_read_buf	RX-NET(PPP) が受信処理を実行した際に呼び出される受信関数
serial_task	ネットワーク割り込みハンドラから呼び出されるネットワーク・タスク
serial_intr	受信割り込み, 送信完了割り込み, エラー割り込みが発生した際に呼び出されるネットワーク割り込みハンドラ

注意 1 RX-NET(PPP) 依存部についての詳細は、「第 6 章 RX-NET(PPP) 依存部」を参照してください。

注意 2 RX-NET(PPP) では、RX-NET(PPP) のサンプル・ソース・ファイルを提供しています。

【CA850 対応版, CCV850E 対応版の場合】
 nectools32\src\netconf\
 rxnetconf.c : 基本情報
 nectools32\src\rxnet\drivers\

`devnameapi.c` : ドライバ関数
`devnameisrsub.c` : ドライバ関数

3.7 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) を行わせる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(PPP) 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
```

3.8 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」～「3.7 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、リロケート可能なオブジェクト・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ーズ・マニュアルを参照してください。
- 注意 2 RX-NET(PPP) 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\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.bld : ロード・モジュール用ビルド・ファイル

3.9 アーカイブ・オブジェクトの生成

「3.8 オブジェクト・ファイルの生成」において作成したり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ーズ・マニュアルを参照してください。
- 注意 2 RX-NET(PPP) 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\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.bld : ロード・モジュール用ビルド・ファイル

3.10 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ーズ・マニュアルを参照してください。
- 注意 2 RX-NET(PPP) 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\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.lx : リンク・ディレクティブ・ファイル

3.11 ロード・モジュールの生成

「3.8 オブジェクト・ファイルの生成」 ~ 「3.9 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, 「3.10 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(PPP) 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(PPP) 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 ネットワーク接続機能

本章では、RX-NET(PPP) が提供しているネットワーク接続機能について解説しています。

4.1 概要

RX-NET(PPP) では、シリアル回線 (RS-232C クロス・ケーブル、電話回線など) 上に確立された Point-to-Point による接続の確立処理 / 遮断処理を “ネットワーク接続機能” として提供しています。

4.2 認証プロトコル

RX-NET(PPP) では、Point-to-Point による接続の確立要求を発行しているユーザに対するアクセス権の有無を検査する機能 (セキュリティ機能) として以下に示した 3 種類の認証プロトコルをサポートしています。

- LOGIN 認証

ネットワーク接続を確立する際に行われる接続設定処理 (LCP 通信設定処理、IPCP 通信設定処理など) を実行する以前に “確立要求を発行しているユーザに対するアクセス権の有無” を検査する認証プロトコルです。

なお、LOGIN 認証では、ネットワーク接続した際、サーバ側からユーザ名、および、パスワードの入力を促す文字列が送信されます。そこで、ユーザ名、および、パスワードを入力すると、サーバ側は受信したユーザ名、および、パスワードから該当ユーザの認証処理を行います。

注意 UNIX では、ユーザ名の入力を促す文字列として “login: ” が、パスワードの入力を促す文字列として “Password: ” が伝統的に使用されていますが、実際にどのような文字列が送信されてくるのかは不明であり、事前の確認が必要となります。

- PAP (Password Authentication Protocol) 認証

LOGIN 認証とは異なり、接続設定処理 (LCP 通信設定処理、IPCP 通信設定処理など) を実行中に “確立要求を発行しているユーザに対するアクセス権の有無” を検査する認証プロトコルです。

なお、PAP 認証では、ターゲット側からユーザ名、および、パスワードを暗号化されていないクリア・テキスト (平文) のままサーバ側に送信し、サーバ側は受信したユーザ名、および、パスワードから該当ユーザの認証処理を行います。

- CHAP (Challenge Handshake Authentication Protocol) 認証

PAP 認証と同様に、接続設定処理 (LCP 通信設定処理、IPCP 通信設定処理など) を実行中に “確立要求を発行しているユーザに対するアクセス権の有無” を検査する認証プロトコルです。

なお、CHAP 認証では、PAP 認証がクリア・テキスト (平文) のまま送信しているのに対し、ユーザ名、および、パスワードを暗号化したのち、サーバ側に送信します。

このため、CHAP 認証は、PAP 認証と比較してセキュリティに対する信頼性が高い認証プロトコルとなっています。

4.3 接続形態

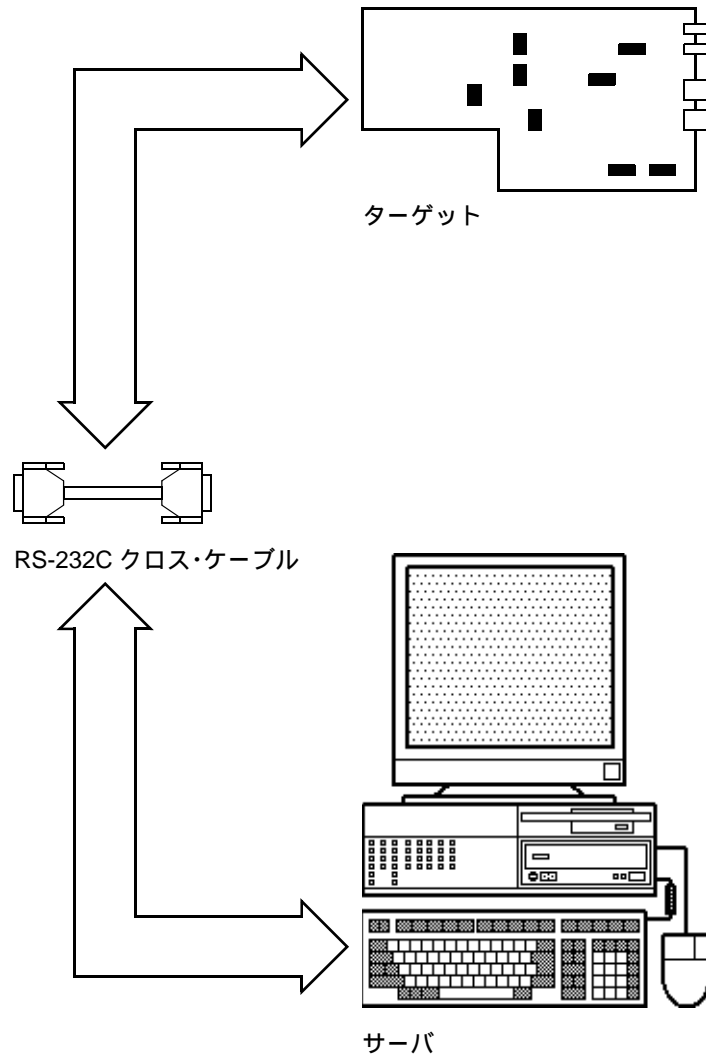
RX-NET(PPP)では、ネットワーク接続の接続形態として以下に示した2種類をサポートしています。

- ヌルモデム接続

モデムを使用することなく、シリアル回線(RS-232Cクロス・ケーブルなど)経由でターゲットとサーバをネットワーク接続する形態です。

表4-1に、ヌルモデム接続におけるネットワーク接続イメージを示します。

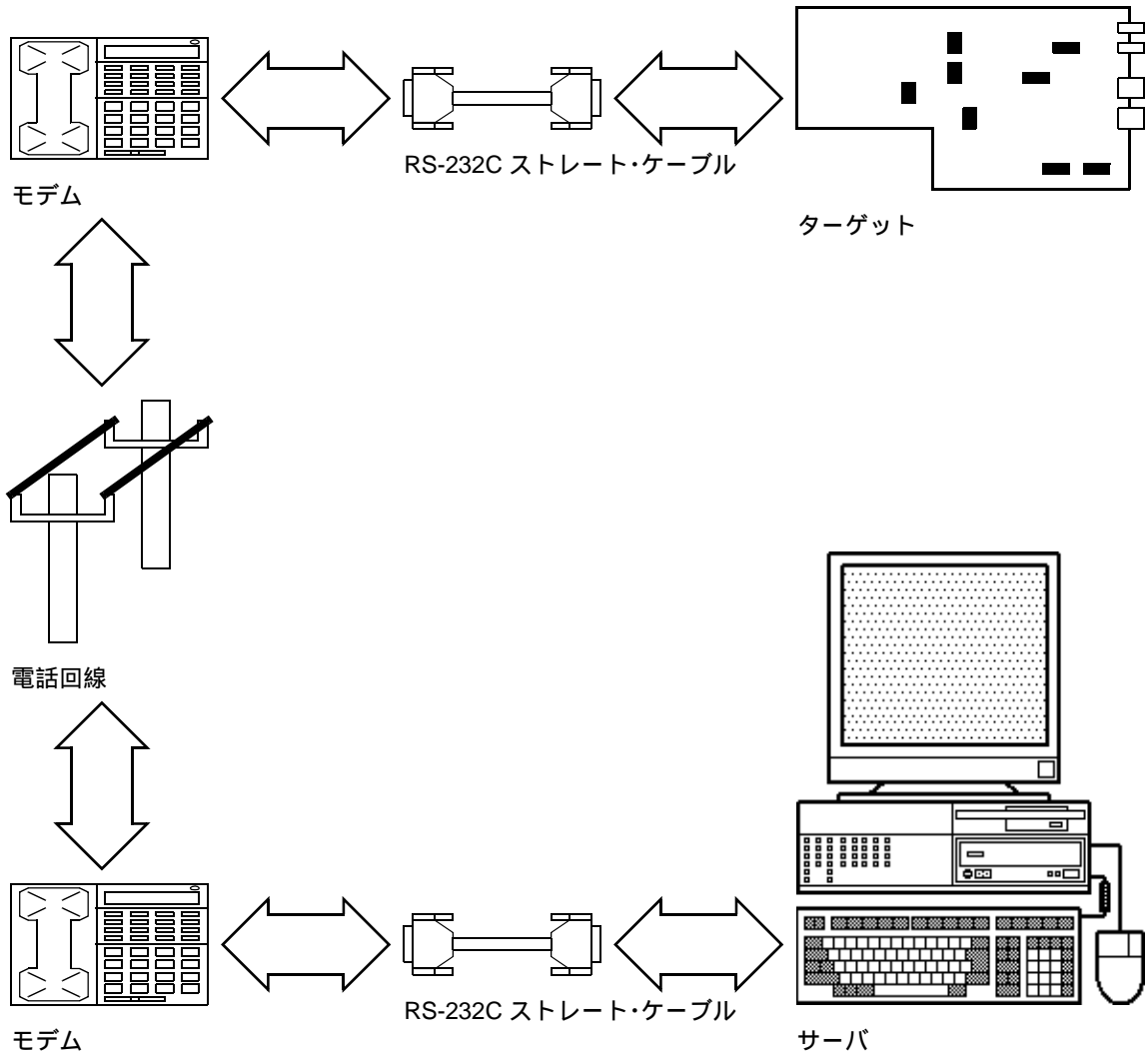
図4-1 ヌルモデム接続



• モデム接続

モデムを使用し、シリアル回線（電話回線など）経由でターゲットとサーバをネットワーク接続する形態です。表 4-2 に、モデム接続におけるネットワーク接続イメージを示します。

図 4-2 モデム接続



4.4 通信オプション

4.4.1 LCP 通信設定要求

RX-NET(PPP) では、Point-to-Point による接続の確立要求が発行された際、以下に示した通信オプションの設定を通信相手に要求しています。

- 非同期文字マップ
エスケープ送信を要求しません。
- 最大受信単位
デフォルト値 1500 (単位: バイト) を使用します。
- マジック・ナンバ
未使用。
- Address/Control フィールド圧縮
API 関数 `ppp_connect` 発行時に接続条件 (圧縮オプションの種類) として Address/Control フィールド圧縮 `PPP_CMP_ADDR` を指定した場合、Address/Control フィールド圧縮の有効要求を行います。
ただし、Nak、または、Rej の応答があった場合には、通信相手の設定に従います。
- プロトコル・フィールド圧縮
API関数`ppp_connect`発行時に接続条件(圧縮オプションの種類)としてプロトコル・フィールド圧縮`PPP_CMP_PROTOCOL`を指定した場合、プロトコル・フィールド圧縮の有効要求を行います。
ただし、Nak、または、Rej の応答があった場合には、通信相手の設定に従います。
- 認証プロトコル
API 関数 `ppp_connect` 発行時に接続条件 (認証プロトコルの種類) として PAP 認証サーバ `PPP_SV_PAP`、または、CHAP 認証サーバ `PPP_SV_CHAP` を指定した場合、PAP、または、CHAP を送信します。

4.4.2 LCP 通信設定要求応答

RX-NET(PPP) では、通信相手から要求パケットに対する応答は以下に示したようになります。

- 非同期文字マップ
0x0 ~ 0xffffffff の全ての値に対して、Ack で応答します。
- マジック・ナンバ
Ack で応答します。
- Address/Control フィールド圧縮
Ack で応答します。
- プロトコル・フィールド圧縮
Ack で応答します。
- 認証プロトコル
PAP 認証、または、CHAP 認証の場合、Ack で応答します。

4.4.3 IPCP 通信設定要求

RX-NET(PPP) では、Point-to-Point による接続の確立要求が発行された際、以下に示した通信オプションの設定を通信相手に要求しています。

- IP アドレス
API 関数 `ppp_connect` のパラメータで指定された IP アドレスの要求を行います。
ただし、Nak の応答があった場合には、通信相手が指定した IP アドレスで再要求を行います。
- Van Jacobson TCP ヘッダ圧縮
API関数`ppp_connect`発行時に接続条件(圧縮オプションの種類)としてVan Jacobson TCPヘッダ圧縮`PPP_CMP_VJ`を指定した場合、スロット数 0x10、connection-ID 圧縮の有効要求を行います。

ただし、Nak、または、Rej の応答があった場合には、通信相手の設定に従います。

- DNS アドレス

API 関数 `ppp_connect` 発行時に接続条件 (DNS アドレスの通知要求有無) として DNS アドレスの通知要求を行う `PPP_CL_MSDNS` を指定した場合、DNS アドレスの通知要求を行います。

4.4.4 IPCP 通信設定要求応答

RX-NET(PPP) では、通信相手から要求パケットに対する応答は以下に示したようになります。

- IP アドレス

API 関数 `ppp_connect` 発行時に接続条件 (認証プロトコルの種類) として PAP 認証クライアント `PPP_CL_PAP`、または、CHAP 認証クライアント `PPP_CL_CHAP` を指定した場合、Ack を送信します。

なお、API 関数 `ppp_connect` 発行時に接続条件 (認証プロトコルの種類) として PAP 認証サーバ `PPP_SV_PAP`、または、CHAP 認証サーバ `PPP_SV_CHAP` を指定した場合、パラメータで指定された IP アドレスを送信します。

ただし、Nak、または、Rej の応答があった場合には、通信相手の設定に従います。

したがって、API 関数 `ppp_connect` 発行時に IP アドレスの指定が行われていなかった場合には、通信相手の要求に従います。

- Van Jacobson TCP ヘッダ圧縮

スロット数 `0x2 ~ 0x10`、`connection-ID` 圧縮有効 / 無効のいずれの場合も Ack で応答します。

4.5 ネットワーク接続機能 API 関数

4.5.1 Point-to-Point プロトコルによる接続の確立 / 遮断

Point-to-Point プロトコルによる接続の確立 / 遮断は、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。

- ppp_connect

パラメータ *username*, *password*, *ipaddress*, *ipmask*, *param* で指定された情報をもとに、Point-to-Point プロトコルによる接続を確立します。

ただし、本 API 関数の発行からパラメータ *wait* で指定された待ち時間が経過するまでの間に接続設定処理 (LCP 通信設定処理、IP CP 通信設定処理など) が完了しなかった場合には、Point-to-Point プロトコルによる接続は行わず、戻り値として EPPP_TI MEDOUT を返しています。

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 <fnconfig.h> /* 静的設定情報ヘッダ・ファイルの定義 */
#include <rxnet_ppp.h> /* RX-NET(PPP) 用標準ヘッダ・ファイルの定義 */

void
func_task ( INT stacd ) {
    char username = NULL ; /* 変数の宣言, 初期化 */
    char password = NULL ; /* 変数の宣言, 初期化 */
    u32 ipaddress = 0x0 ; /* 変数の宣言, 初期化 */
    u32 ipmask = 0x0 ; /* 変数の宣言, 初期化 */
    int wait = 5000 ; /* 変数の宣言, 初期化 */
    int param = PPP_DIAL | PPP_LOGIN ; /* 変数の宣言, 初期化 */

    so_initialize ( ) ; /* RX-NET(TCP/IP) の初期化 */
                        /* Point-to-Point プロトコルによる接続の確立 */
    ppp_connect ( username , password , &ipaddress , ipmask , wait , param ) ;
    ext_tsk ( ) ; /* タスクの終了処理 */
}

```

注意 1 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。

注意 2 パラメータ *ipaddress* に 0x0 を指定した場合、RX-NET(PPP) は IP アドレスの割り当て処理は通信相手が行うものとして処理を行います。
なお、本 API 関数が正常終了した際には、パラメータ *ipaddress* で指定された領域に“通信相手が割り当てた IP アドレス”が格納されます。

注意 3 パラメータ *param* に PPP_DIAL を指定した場合、ダイアル名エントリ・テーブル *dial_array* に対する参照処理が実行されます。
なお、パラメータ *param* に PPP_DIAL を指定しなかった際には、接続形態は“ヌルモデム接続”となり、ダイアル名エントリ・テーブル *dial_array* に対象する参照処理は実行されません。

注意 4 パラメータ *param* に PPP_LOGIN を指定した場合、ログイン名エントリ・テーブル *login_array* に対する参照処理が実行されます。

注意 5 パラメータ *param* に PPP_CL_PAP、または、PPP_CL_CHAP を指定した場合、通信相手が PAP 認証要求、または、CHAP 認証要求を行ってきた際には、パラメータ *username*、および、パラメータ *password* で指定された値を用いて認証処理が実行されます。

注意 6 パラメータ *param* に PPP_CL_MSDNS を指定した場合、通信相手が DNS アドレスの通知を行ってきた際には、API 関数 *ppp_getNameServerIP*、または、*ppp_getNameServerIP2* を発行することにより該当情報を獲得することができます。

注意 7 本 API 関数では、Point-to-Point プロトコルによる接続の確立処理を実行する際、ドライバ関数 `serial_updown` の呼び出しを行っています。
なお、ドライバ関数 `serial_updown` についての詳細は、「第 6 章 RX-NET(PPP) 依存部 `serial_updown`」を参照してください。

- ppp_disconnect

パラメータ *ipaddress* , *ipmask* で指定された情報をもとに , Point-to-Point プロトコルによる接続を遮断します。以下に , 本 API 関数の記述例を示します。

なお , 記述例中の *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_ppp.h>       /* RX-NET(PPP) 用標準ヘッダ・ファイルの定義 */

void
func_task ( INT stacd ) {
    char      username = NULL ;    /* 変数の宣言 , 初期化 */
    char      password = NULL ;   /* 変数の宣言 , 初期化 */
    u32       ipaddress = 0x0 ;    /* 変数の宣言 , 初期化 */
    u32       ipmask = 0x0 ;      /* 変数の宣言 , 初期化 */
    int       wait = 5000 ;       /* 変数の宣言 , 初期化 */
    int       param = PPP_DIAL | PPP_LOGIN ; /* 変数の宣言 , 初期化 */

    so_initialize ( ) ;          /* RX-NET(TCP/IP) の初期化 */
                                /* Point-to-Point プロトコルによる接続の確立 */
    ppp_connect ( username , password , &ipaddress , ipmask , wait , param ) ;

    .....
    .....
    .....

    ppp_disconnect ( ipaddress , ipmask ) ; /* Point-to-Point プロトコルによる接続の遮断 */
    ext_tsk ( ) ;                 /* タスクの終了処理 */
}

```

注意 1 IP アドレス *ipaddress* には API 関数 *ppp_connect* の発行により割り当てられた IP アドレス (ネットワーク・バイト・オーダ) を設定します。

また , ネット・マスク *ipmask* には , API 関数 *ppp_connect* を発行した際に指定したネット・マスク (ネットワーク・バイト・オーダ) を設定します。

注意 2 本 API 関数では , Point-to-Point プロトコルによる接続の遮断処理を実行する際 , ドライバ関数 *serial_updown* の呼び出しを行っています。

なお , ドライバ関数 *serial_updown* についての詳細は , 「第 6 章 RX-NET(PPP) 依存部 *serial_updown*」を参照してください。

4.5.2 パスワード・データの登録

パスワード・データの登録は、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。

- ppp_setPassDB

パラメータ *username*, *password*, *ipaddress* で指定されたパスワード・データを認証クライアントからの接続要求を受け付けた際に参照するパスワード・データ・ベースに登録します。

なお、パラメータ *number* には、パスワード・データ・ベースのエントリ番号を指定します。

以下に、本 API 関数の記述例を示します。

なお、記述例中の *so_initialize*, *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_ppp.h>              /* RX-NET(PPP) 用標準ヘッダ・ファイルの定義 */

void
func_task ( INT stacd ) {
    int          number = 0x0 ;           /* 変数の宣言, 初期化 */
    char         username = "sample_name" ; /* 変数の宣言, 初期化 */
    char         password = "sample_password" ; /* 変数の宣言, 初期化 */
    u32          ipaddress = inet_addr ( "0.0.0.0" ) ; /* 変数の宣言, 初期化 */

    so_initialize ( ) ;                  /* RX-NET(TCP/IP) の初期化 */
                                          /* パスワード・データの登録 */
    ppp_setPassDB ( number , &username , &password , ipaddress ) ;
    ext_tsk ( ) ;                        /* タスクの終了処理 */
}
```

4.5.3 Challenge メッセージの登録

Challenge メッセージの登録は、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。

- ppp_setChapChallengeMsg

パラメータ *buf*, *buf_len*, *name* で指定された情報をもとに、CHAP 認証クライアントからの接続要求を受け付けた際に CHAP 認証クライアントに対して送信する Challenge メッセージ、および、CHAP 認証サーバのマシン名を登録します。

以下に、本 API 関数の記述例を示します。

なお、記述例中の *so_initialize*, *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_ppp.h> /* RX-NET(PPP) 用標準ヘッダ・ファイルの定義 */

void
func_task ( INT stacd ) {
    int          number = 0x0 ; /* 変数の宣言, 初期化 */
    char         username = "sample_name" ; /* 変数の宣言, 初期化 */
    char         password = "sample_password" ; /* 変数の宣言, 初期化 */
    u32          ipaddress = inet_addr ( "0.0.0.0" ) ; /* 変数の宣言, 初期化 */
    char         buf = "sample_message" ; /* 変数の宣言, 初期化 */
    int          buf_len = strlen ( buf ) ; /* 変数の宣言, 初期化 */
    char         name = "CHAP_server_name" ; /* 変数の宣言, 初期化 */
    u32          ipmask = inet_addr ( "255.255.255.0" ) ; /* 変数の宣言, 初期化 */
    int          wait = 5000 ; /* 変数の宣言, 初期化 */
    int          param = PPP_SV_CHAP ; /* 変数の宣言, 初期化 */

    so_initialize ( ) ; /* RX-NET(TCP/IP) の初期化 */
                    /* パスワード・データの登録 */
    ppp_setPassDB ( number , &username , &password , ipaddress ) ;
                    /* Challenge メッセージの登録 */
    ppp_setChapChallengeMsg ( &buf , buf_len , &name ) ;

    username = NULL ; /* 変数の初期化 */
    password = NULL ; /* 変数の初期化 */
    ipaddress = inet_addr ( "10.30.180.1" ) ; /* 変数の初期化 */

                    /* Point-to-Point プロトコルによる接続の確立 */
    ppp_connect ( username , password , &ipaddress , ipmask , wait , param ) ;
    ext_tsk ( ) ; /* タスクの終了処理 */
}

```


4.5.4 IP アドレスの獲得

IP アドレスの獲得は、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。

- ppp_getNameServerIP

プライマリ・ネーム・サーバの IP アドレスを *ipaddress* で指定された領域に格納します。

以下に、本 API 関数の記述例を示します。

なお、記述例中の *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 <fnconfig.h> /* 静的設定情報ヘッダ・ファイルの定義 */
#include <rxnet_ppp.h> /* RX-NET(PPP) 用標準ヘッダ・ファイルの定義 */

void
func_task ( INT stacd ) {
    char username = NULL ; /* 変数の宣言, 初期化 */
    char password = NULL ; /* 変数の宣言, 初期化 */
    u32 ipaddress = 0x0 ; /* 変数の宣言, 初期化 */
    u32 ipmask = 0x0 ; /* 変数の宣言, 初期化 */
    int wait = 5000 ; /* 変数の宣言, 初期化 */
    int param = PPP_DIAL | PPP_LOGIN ; /* 変数の宣言, 初期化 */

    so_initialize ( ) ; /* RX-NET(TCP/IP) の初期化 */
                        /* Point-to-Point プロトコルによる接続の確立 */
    ppp_connect ( username , password , &ipaddress , ipmask , wait , param ) ;
    ppp_getNameServerIP ( &ipaddress ) ; /* IP アドレスの獲得 */
    ext_tsk ( ) ; /* タスクの終了処理 */
}
```

注意 本 API 関数の発行は、API 関数 *ppp_connect* の処理完了後に行う必要があります。

ただし、API 関数 *ppp_connect* を発行した際に接続条件として DNS アドレスの通知要求を行う *PPP_CL_MSDNS* が指定されていなかった場合、および、通信相手が DNS アドレスの通知を行わなかった場合には、IP アドレスの獲得処理は行わず、戻り値として *EPPP_NOMSDNS* が返されます。

- ppp_getNameServerIP2

セカンダリ・ネーム・サーバの IP アドレスを *ipaddress* で指定された領域に格納します。

以下に、本 API 関数の記述例を示します。

なお、記述例中の *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_ppp.h>        /* RX-NET(PPP) 用標準ヘッダ・ファイルの定義 */

void
func_task ( INT stacd ) {
    char      username = NULL;      /* 変数の宣言, 初期化 */
    char      password = NULL;     /* 変数の宣言, 初期化 */
    u32       ipaddress = 0x0;     /* 変数の宣言, 初期化 */
    u32       ipmask = 0x0;        /* 変数の宣言, 初期化 */
    int       wait = 5000;         /* 変数の宣言, 初期化 */
    int       param = PPP_DIAL | PPP_LOGIN; /* 変数の宣言, 初期化 */

    so_initialize ( );             /* RX-NET(TCP/IP) の初期化 */
                                   /* Point-to-Point プロトコルによる接続の確立 */
    ppp_connect ( username , password , &ipaddress , ipmask , wait , param );
    ppp_getNameServerIP2 ( &ipaddress ); /* IP アドレスの獲得 */
    ext_tsk ( );                  /* タスクの終了処理 */
}
```

注意 本 API 関数の発行は、API 関数 *ppp_connect* の処理完了後に行う必要があります。
 ただし、API 関数 *ppp_connect* を発行した際に接続条件として DNS アドレスの通知要求を行う *PPP_CL_MSDNS* が指定されていなかった場合、および、通信相手が DNS アドレスの通知を行わなかった場合には、IP アドレスの獲得処理は行わず、戻り値として *EPMP_NOMSDNS* が返されます。

第5章 API 関数

本章では、RX-NET(PPP) 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(PPP) 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(PPP) 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(PPP) が提供している API 関数 (6 種類) を示します。

ppp_connect	ppp_disconnect	ppp_setPassDB	ppp_setChapChallengeMsg
ppp_getNameServerIP	ppp_getNameServerIP2		

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(PPP) 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(PPP) 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(PPP) 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_ppp.h	: RX-NET(PPP) 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(PPP) 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, 接続条件など) 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 nectools32\inc850\rxnet.h から呼び出されるヘッダ・ファイル nectools32\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 nectools32\inc850\rxnet.h, および, ヘッダ・ファイル nectools32\inc850\rxnet\ccdep.h は, RX-NET(TCP/IP) が提供しています。

5.3.2 接続条件

表 5-2 に, API 関数 ppp_connect を発行する際に指定する接続条件一覧を示します。

なお, 接続条件のマクロ定義は, 標準ヘッダ・ファイル nectools32\inc850\rxnet_ppp.h から呼び出されるヘッダ・ファイル nectools32\inc850\rxnet\pppapi.h で行われています。

表 5-2 接続条件

マクロ	数値	意味
PPP_LOGIN	0x10000	LOGIN 認証
PPP_DIAL	0x20000	モデム接続
PPP_CL_PAP	0x200000	PAP 認証クライアント
PPP_CL_CHAP	0x400000	CHAP 認証クライアント
PPP_CL_MSDNS	0x800000	DNS アドレスの通知要求を行う
PPP_SV_PAP	0x2000000	PAP 認証サーバ
PPP_SV_CHAP	0x4000000	CHAP 認証サーバ
PPP_CMP_VJ	0x20000000	Van Jacobson TCP ヘッダ圧縮
PPP_CMP_ADDR	0x40000000	Address/Control フィールド圧縮
PPP_CMP_PORT	0x80000000	プロトコル・フィールド圧縮

5.3.3 戻り値

表 5-3 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_ppp.h` から呼び出されるヘッダ・ファイル `nctools32\inc850\rxnet\ppperrno.h` で行われています。

表 5-3 戻り値

マクロ	数値	意味
EPPP_NOERR	0x0	正常終了
EPPP_INVALID	0x1300	パラメータの指定が不正です
EPPP_NOPPP	0x1301	RX-NET(PPP) の初期化処理，終了処理が失敗しました
EPPP_NETDOWN	0x1302	<code>so_initialize</code> が発行されていません
EPPP_DIALFAIL	0x1303	ダイヤルアップ処理が失敗しました
EPPP_LOGINFAIL	0x1304	LOGIN 文字列処理が失敗しました
EPPP_DEVICEFAIL	0x1305	デバイス・ドライバの初期化処理，または，終了処理が失敗しました
EPPP_TIMEDOUT	0x1307	接続がタイムアウトしています
EPPP_NOMEM	0x1308	ヒープ用メモリ領域が確保できません
EPPP_CONNECT	0x1309	既に <code>ppp_connect</code> が発行されています
EPPP_NOTCONN	0x130a	既に <code>ppp_disconnect</code> が発行されています
EPPP_NOMSDNS	0x130b	DNS アドレスの獲得処理が失敗しました

5.4 データ構造体

RX-NET(PPP) 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 (モデム制御シーケンス情報など) について以下に示します。

5.4.1 モデム制御シーケンス情報

以下に、モデム制御シーケンス情報 `se_sequence` を示します。

なお、モデム制御シーケンス情報 `se_sequence` の定義は、標準ヘッダ・ファイル `nctools32\inc850\rxnet_ppp.h` から呼び出されるヘッダ・ファイル `nctools32\inc850\rxnet\ppp.h` で行われています。

```
#define          MAXSEND_EXPECT          20

typedef          struct                  se_sequence {
    char          send [ MAXSEND_EXPECT ]; /* 送信文字列 */
    char          expect [ MAXSEND_EXPECT ]; /* 応答文字列の期待値 */
    long         timeout ;                /* 待ち時間 ( 単位 : ミリ秒 ) */
    int          trys ;                    /* 再試行回数の上限值 */
} SE_SEQUENCE ;
```

以下に、Linger オプション情報 `lingeropt` の詳細を示します。

- `send []`
送信文字列を指定します。
- `expect []`
送信文字列に対する応答文字列の期待値を指定します。
- `timeout`
送信文字列の送信処理から応答文字列と期待値のマッチング検査処理を実行するまでの待ち時間 (単位 : ミリ秒) を指定します。
ただし、`SE_SEQUENCE` 型配列 `dial_array []` においては、送信文字列の送信処理からモデムの DCD 信号線が “ 1 ” になるまでの待ち時間となります。
- `trys`
応答文字列と期待値が異なっていた場合に行われる送信処理の再試行回数の上限值を指定します。

注意 RX-NET(PPP) では、以下に示した 2 種類の `SE_SEQUENCE` 型配列をユーザの処理プログラム内で定義する必要があります。

なお、モデム制御シーケンス情報の最終要素には、それが最終要素であることを RX-NET(PPP) に通知するために `{ "\0", "\0", 0x0, 0x0 }`

を記述する必要があります。

- ダイアル名エントリ・テーブル
以下に、ダイアル名エントリ・テーブル `dial_array` の記述例を示します。

```
SE_SEQUENCE dial_array [ ] = {
    { "ATDT012-345-6789\r", "OK\r", 3000L, 2 },
    { "\0", "\0", 0x0, 0x0 }
};
```

なお、上記のようにダイアル名エントリ・テーブル `dial_array` を定義した場合、次のような意味を持ちます。

- 文字列 “ ATDT012-345-6789\r ” をシリアル回線に送信 (AT コマンドでプッシュ回線に 012-345-6789 をダイアル)
- 3000 ミリ秒が経過する以前にモデムの DCD 信号線が “ 1 ” になった場合は、正常終了
- 3000 ミリ秒が経過してもモデムの DCD 信号線が “ 1 ” にならなかった場合は、計 2 回まで待ち受け処理を実行

- ログイン名エントリ・テーブル

以下に、ログイン名エントリ・テーブル login_array の記述例を示します。

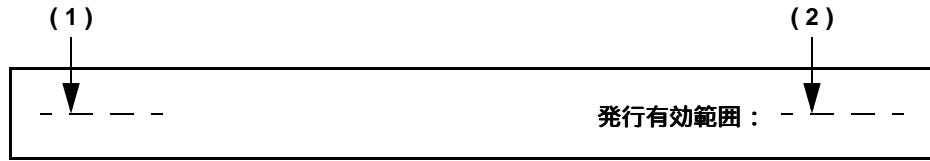
```
SE_SEQUENCE login_array [] = {
    {"\r", "ogin: ", 5000L, 5},
    {"myloginname\r", "assword: ", 3000L, 3},
    {"mypasswd\r", "\0", 1000L, 2},
    {"\0", "\0", 0x0, 0x0}
};
```

なお、上記のようにログイン名エントリ・テーブル login_array を定義した場合、次のような意味を持ちます。

- 改行文字 “\r” をシリアル回線に送信
- 5000 ミリ秒が経過した際、シリアル回線から文字列を受信していたか否かを検査し、受信文字列の末尾が “ogin: ” と一致していた場合は、文字列 “myloginname\r” をシリアル回線に送信
なお、5000 ミリ秒が経過してもシリアル回線から文字列を受信していなかった場合、および、受信文字列の末尾が “ogin: ” と一致していなかった場合は、計 5 回まで待ち受け処理を実行
- 3000 ミリ秒が経過した際、シリアル回線から文字列を受信していたか否かを検査し、受信文字列の末尾が “assword: ” と一致していた場合は、文字列 “mypasswd\r” をシリアル回線に送信
なお、3000 ミリ秒が経過してもシリアル回線から文字列を受信していなかった場合、および、受信文字列の末尾が “assword: ” と一致していなかった場合は、計 3 回まで待ち受け処理を実行
- 1000 ミリ秒が経過した際、シリアル回線から文字列を受信していたか否かを検査し、受信文字列の末尾が “\0” と一致していた場合は、正常終了
なお、1000 ミリ秒が経過してもシリアル回線から文字列を受信していなかった場合、および、受信文字列の末尾が “\0” と一致していなかった場合は、計 2 回まで待ち受け処理を実行

5.5 API 関数解説

次項から RX-NET(PPP) 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(PPP) への入力パラメータ
- O ... RX-NET(PPP) 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(PPP) が提供している API 関数の一覧を示します。

表 5-4 API 関数

API 関数名	機能概要
ppp_connect	Point-to-Point プロトコルによる接続の確立
ppp_disconnect	Point-to-Point プロトコルによる接続の遮断
ppp_setPassDB	パスワード・データの登録
ppp_setChapChallengeMsg	Challenge メッセージの登録
ppp_getNameServerIP	IP アドレス (プライマリ・ネーム・サーバ) の獲得
ppp_getNameServerIP2	IP アドレス (セカンダリ・ネーム・サーバ) の獲得

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。

ppp_connect

発行有効範囲：タスク

概要

Point-to-Point プロトコルによる接続の確立

C 言語形式

```
int ppp_connect ( char *username , char *password , u32 *ipaddress , u32 ipmask , int wait , int param ) ;
```

パラメータ

I/O	パラメータ	説明
I	char *username ;	ユーザ名を格納した領域へのポインタ
I	char *password ;	パスワードを格納した領域へのポインタ
I, O	u32 *ipaddress ;	ネットワーク・デバイスに割り当てる IP アドレス (ネットワーク・バイト・オーダ) を格納した領域へのポインタ
I	u32 ipmask ;	ネット・マスク (ネットワーク・バイト・オーダ)
I	int wait ;	待ち時間 (単位 : ミリ秒) 数値 : 0x0 ~ 0x7ffffff
I	int param ;	接続条件 接続形態の種類 PPP_DIAL : モデム接続 認証プロトコルの種類 PPP_LOGIN : LOGIN 認証 PPP_CL_PAP : PAP 認証クライアント PPP_CL_CHAP : CHAP 認証クライアント PPP_SV_PAP : PAP 認証サーバ PPP_SV_CHAP : CHAP 認証サーバ DNS アドレスの通知要求有無 PPP_CL_MS DNS : DNS アドレスの通知要求を行う 圧縮オプションの種類 PPP_CMP_VJ : Van Jacobson TCP ヘッダ圧縮 PPP_CMP_ADDR : Address/Control フィールド圧縮 PPP_CMP_PROT : プロトコル・フィールド圧縮

機能

username , password , ipaddress , ipmask , param で指定された情報をもとに , Point-to-Point プロトコルによる接続を確立します。

ただし , 本 API 関数の発行から wait で指定された待ち時間が経過するまでの間に接続設定処理 (LCP 通信設定処理 , IP CP通信設定処理など) が完了しなかった場合には , Point-to-Point プロトコルによる接続は行わず , 戻り値として EPPP_TIMEDOUT を返しています。

注意 1 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。

注意 2 ipaddress に 0x0 を指定した場合 , RX-NET(PPP) は IP アドレスの割り当て処理は通信相手が行うものとして処理を行います。

- なお、本 API 関数が正常終了した際には、*ipaddress* で指定された領域に “通信相手が割り当てた IP アドレス” が格納されます。
- 注意 3 *param* に PPP_DIAL を指定した場合、ダイヤル名エントリ・テーブル *dial_array* に対する参照処理が実行されます。
 なお、*param* に PPP_DIAL を指定しなかった際には、接続形態は “ヌルモデム接続” となり、ダイヤル名エントリ・テーブル *dial_array* に対象する参照処理は実行されません。
- 注意 4 *param* に PPP_LOGIN を指定した場合、ログイン名エントリ・テーブル *login_array* に対する参照処理が実行されます。
- 注意 5 *param* に PPP_CL_PAP、または、PPP_CL_CHAP を指定した場合、通信相手が PAP 認証要求、または、CHAP 認証要求を行ってきた際には、*username*、および、*password* で指定された値を用いて認証処理が実行されます。
- 注意 6 *param* に PPP_CL_MSDNS を指定した場合、通信相手が DNS アドレスの通知を行ってきた際には、API 関数 *ppp_getNameServerIP*、または、*ppp_getNameServerIP2* を発行することにより該当情報を獲得することができます。
- 注意 7 本 API 関数では、Point-to-Point プロトコルによる接続の確立処理を実行する際、ドライバ関数 *serial_updown* の呼び出しを行っています。
 なお、ドライバ関数 *serial_updown* についての詳細は、「第 6 章 RX-NET(PPP) 依存部 *serial_updown*」を参照してください。

戻り値

EPPP_NOERR	0x0	正常終了
EPPP_INVALID	0x1300	パラメータの指定が不正です <ul style="list-style-type: none"> - ユーザ名を格納した領域の指定が不正 (NULL ポインタ) です - ユーザ名が制限文字数 (24 文字) を超えています - パスワードを格納した領域の指定が不正 (NULL ポインタ) です - パスワードが制限文字数 (14 文字) を超えています - ネットワーク・デバイスに割り当てる IP アドレスを格納した領域の指定が不正 (NULL ポインタ) です - 待ち時間の指定が不正 (0x0 ~ 0x7ffffff 以外の値) です
EPPP_NOPPP	0x1301	RX-NET(PPP) の初期化処理が失敗しました
EPPP_NETDOWN	0x1302	<i>so_initialize</i> が発行されていません
EPPP_DIALFAIL	0x1303	ダイヤルアップ処理が失敗しました
EPPP_LOGINFAIL	0x1304	LOGIN 文字列処理が失敗しました
EPPP_DEVICEFAIL	0x1305	ネットワーク・インタフェースの起動処理が失敗しました
EPPP_TIMEDOUT	0x1307	ネゴシエーション処理が失敗しました
EPPP_NOMEM	0x1308	ヒープ用メモリ領域が確保できません
EPPP_CONNECT	0x1309	既に本 API 関数が発行されています

ppp_disconnect

発行有効範囲：タスク

概要

Point-to-Point プロトコルによる接続の遮断

C 言語形式

```
int ppp_disconnect ( u32 ipaddress , u32 ipmask );
```

パラメータ

I/O	パラメータ	説明
I	u32 <i>ipaddress</i> ;	IP アドレス (ネットワーク・バイト・オーダ)
I	u32 <i>ipmask</i> ;	ネット・マスク (ネットワーク・バイト・オーダ)

機能

ipaddress , *ipmask* で指定された情報をもとに , Point-to-Point プロトコルによる接続を遮断します。

- 注意 1 IP アドレス *ipaddress* には , API 関数 `ppp_connect` の発行により割り当てられた IP アドレス (ネットワーク・バイト・オーダ) を設定します。
また , ネット・マスク *ipmask* には , API 関数 `ppp_connect` を発行した際に指定したネット・マスク (ネットワーク・バイト・オーダ) を設定します。
- 注意 2 本 API 関数では , Point-to-Point プロトコルによる接続の遮断処理を実行する際 , ドライバ関数 `serial_updown` の呼び出しを行っています。
なお , ドライバ関数 `serial_updown` についての詳細は , 「第 6 章 RX-NET(PPP) 依存部 `serial_updown`」を参照してください。

戻り値

EPPP_NOERR	0x0	正常終了
EPPP_NOPPP	0x1301	RX-NET(PPP) の初期化処理が失敗しました
EPPP_NETDOWN	0x1302	<code>so_initialize</code> が発行されていません
EPPP_DEVICEFAIL	0x1305	ネットワーク・インタフェースの遮断処理が失敗しました
EPPP_NOTCONN	0x130a	既に本 API 関数が発行されています

ppp_setPassDB

発行有効範囲：タスク

概要

パスワード・データの登録

C 言語形式

```
int ppp_setPassDB ( int number , char *username , char *password , u32 ipaddress );
```

パラメータ

I/O	パラメータ	説明
I	int <i>number</i> ;	パスワード・データを登録するエントリ番号 数値 : 0x0 ~ 0x9
I	char <i>*username</i> ;	ユーザ名を格納した領域へのポインタ
I	char <i>*password</i> ;	<i>username</i> で指定されたユーザ名に割り当てるパスワードを格納した領域へのポインタ
I	u32 <i>ipaddress</i> ;	<i>username</i> で指定されたユーザ名に割り当てる IP アドレス (ネットワーク・バイト・オーダ)

機能

username , *password* , *ipaddress* で指定されたパスワード・データを認証クライアントからの接続要求を受け付けた際に参照するパスワード・データ・ベースに登録します。

なお , *number* には , パスワード・データ・ベースのエントリ番号を指定します。

戻り値

EPPP_NOERR	0x0	正常終了
EPPP_INVALID	0x1300	パラメータの指定が不正です <ul style="list-style-type: none"> - パスワード・データを登録するエントリ番号の指定が不正 (0x0 ~ 0x9 以外の値) です - ユーザ名を格納した領域の指定が不正 (NULL ポインタ) です - ユーザ名が制限文字数 (24 文字) を超えています - パスワードを格納した領域の指定が不正 (NULL ポインタ) です - パスワードが制限文字数 (14 文字) を超えています
EPPP_NETDOWN	0x1302	so_initialize が発行されていません

ppp_setChapChallengeMsg

発行有効範囲：タスク

概要

Challenge メッセージの登録

C 言語形式

```
int ppp_setChapChallengeMsg ( char *buf , int buf_len , char *name ) ;
```

パラメータ

I/O	パラメータ	説明
I	char *buf ;	Challenge メッセージを格納した領域へのポインタ
I	int buf_len ;	buf で指定された領域のサイズ (単位 : バイト) 数値 : 0x1 ~ 0x20
I	char *name ;	CHAP 認証サーバのマシン名を格納した領域へのポインタ

機能

buf , buf_len , name で指定された情報をもとに , CHAP 認証クライアントからの接続要求を受け付けた際に CHAP 認証クライアントに対して送信する Challenge メッセージ , および , CHAP 認証サーバのマシン名を登録します。

戻り値

EPPP_NOERR	0x0	正常終了
EPPP_INVALID	0x1300	パラメータの指定が不正です <ul style="list-style-type: none"> - Challenge メッセージを格納した領域の指定が不正 (NULL ポインタ) です - buf で指定された領域のサイズの指定が不正 (0x1 ~ 0x20 以外の値) です - CHAP 認証サーバのマシン名を格納した領域の指定が不正 (NULL ポインタ) です - CHAP 認証サーバのマシン名が制限文字数 (24 文字) を超えています
EPPP_NETDOWN	0x1302	so_initialize が発行されていません

ppp_getNameServerIP

発行有効範囲：タスク

概要

IP アドレス (プライマリ・ネーム・サーバ) の獲得

C 言語形式

```
int ppp_getNameServerIP ( u32 *ipaddress );
```

パラメータ

I/O	パラメータ	説明
O	u32 *ipaddress;	IP アドレス (ネットワーク・バイト・オーダ) を格納する領域へのポインタ

機能

プライマリ・ネーム・サーバの IP アドレスを *ipaddress* で指定された領域に格納します。

注意 本 API 関数の発行は、API 関数 `ppp_connect` の処理完了後に行う必要があります。ただし、API 関数 `ppp_connect` を発行した際に接続条件として DNS アドレスの通知要求を行う `PPP_CL_MSDNS` が指定されていなかった場合、および、通信相手が DNS アドレスの通知を行わなかった場合には、IP アドレスの獲得処理は行わず、戻り値として `EPPP_NOMSDNS` が返されます。

戻り値

<code>EPPP_NOERR</code>	0x0	正常終了
<code>EPPP_INVALID</code>	0x1300	パラメータの指定が不正です
<code>EPPP_NOPPP</code>	0x1301	<code>ppp_connect</code> が発行されていません
<code>EPPP_NETDOWN</code>	0x1302	<code>so_initialize</code> が発行されていません
<code>EPPP_NOTCONN</code>	0x130a	既に <code>ppp_disconnect</code> が発行されています
<code>EPPP_NOMSDNS</code>	0x130b	DNS アドレスの獲得処理が失敗しました

ppp_getNameServerIP2

発行有効範囲：タスク

概要

IP アドレス (セカンダリ・ネーム・サーバ) の獲得

C 言語形式

```
int ppp_getNameServerIP2 ( u32 *ipaddress );
```

パラメータ

I/O	パラメータ	説明
O	u32 *ipaddress;	IP アドレス (ネットワーク・バイト・オーダ) を格納する領域へのポインタ

機能

セカンダリ・ネーム・サーバの IP アドレスを *ipaddress* で指定された領域に格納します。

注意 本 API 関数の発行は、API 関数 `ppp_connect` の処理完了後に行う必要があります。ただし、API 関数 `ppp_connect` を発行した際に接続条件として DNS アドレスの通知要求を行う `PPP_CL_MSDNS` が指定されていない場合、および、通信相手が DNS アドレスの通知を行わなかった場合には、IP アドレスの獲得処理は行わず、戻り値として `EPPP_NOMSDNS` が返されます。

戻り値

<code>EPPP_NOERR</code>	0x0	正常終了
<code>EPPP_INVALID</code>	0x1300	パラメータの指定が不正です
<code>EPPP_NOPPP</code>	0x1301	<code>ppp_connect</code> が発行されていません
<code>EPPP_NETDOWN</code>	0x1302	<code>so_initialize</code> が発行されていません
<code>EPPP_NOTCONN</code>	0x130a	既に <code>ppp_disconnect</code> が発行されています
<code>EPPP_NOMSDNS</code>	0x130b	DNS アドレスの獲得処理が失敗しました

第 6 章 RX-NET(PPP) 依存部

本章では、RX-NET(PPP) が提供している RX-NET(PPP) 依存部について解説しています。

6.1 概要

RX-NET(PPP) では、RX-NET(PPP) が提供する機能を実現する際に必要となる基本情報、および、ドライバ関数を RX-NET(PPP) 依存部 (ユーザ・OWN・コーディング部) として切り出し、サンプル・ソース・ファイルを提供しています。

そこで、システム構築時には、これら基本情報、および、ドライバ関数をユーザの実行環境 / アプリケーション・システムにあわせてカスタマイズ化する必要があります。

以下に、RX-NET(PPP) が RX-NET(PPP) 依存部として切り出しているドライバ関数 (5 種類) を示します。

```
serial_updown          serial_read_buf      serial_write          serial_task
serial_intr
```

なお、ドライバ関数の呼び出しは、Point-to-Point プロトコルによる接続の確立処理 / 遮断処理、RX-NET(PPP) の送信処理 / 受信処理などを実行した際、RX-NET(PPP) により行われます。

したがって、ドライバ関数への入力パラメータに対する設定処理は RX-NET(PPP) が行っています。

6.2 基本情報

RX-NET(PPP) が提供する機能を実現する際に必要となる基本情報 (デバイス・ドライバ・エントリ・テーブル情報など) について以下に示します。

6.2.1 デバイス・ドライバ・エントリ・テーブル情報

RX-NET(PPP) では、RX-NET(PPP) が提供しているネットワーク接続機能を実現する際に必要となる内部関数の登録処理を RX-NET(PPP) 依存部として切り出しています。

このため、ユーザは、規定された変数名を用いてデバイス・ドライバ・エントリ・テーブル情報を定義する必要があります。

```
/* デバイス・ドライバ・エントリ・テーブル情報の定義 */
netdev          ndevsw [ ] = {
/*
    ネットワーク・デバイス名,
    システム予約領域 (0x0 固定),
    システム予約領域 (0x0 固定),
    初期化関数用第 1 パラメータ,
    初期化関数用第 3 パラメータ,
    システム予約領域,
    初期化関数名,
    システム予約領域,
    システム予約領域
    システム予約領域 (0x0 固定),
    システム予約領域 (0x0 固定),
    初期化関数用第 2 パラメータ,
    初期化関数用第 4 パラメータ,
    システム予約領域 ( { 0x0 } 固定 ),
    開始 / 終了関数名,
    送信関数名,

*/
    /******
    第 1 要素
    *****/
    {
        "im",
        0x0,
        0x0,
        0x0,
        0x0,
        { 0x0 },
        0x0,
        0x0,
        0x0,
        0x0,
        { 0x0 },
    }
}
```

```

        noent_init ,
        im_send ,
        noent_ioctl
    },
    /*****
    第 2 要素
    *****/
    {
        "ppp0" ,
        0x0 ,
        0x0 ,
        PPP0_P0 ,
        PPP0_P2 ,
        { AF_PPP } ,
        ppp_init ,
        ppp_scomm ,
        noent_ioctl
    }
};

```

注意 1 第 1 要素については ,RX-NET(TCP/IP) が提供しているネットワーク通信機能を実現する際に必要となる内部処理関数の登録処理が ,第 2 要素については ,RX-NET(PPP) が提供しているネットワーク接続機能を実現する際に必要となる内部処理関数の登録処理が記述されています。このため ,第 1 要素 ,および ,第 2 要素の内容を変更することは禁止されています。

注意 2 RX-NET(PPP) では ,本デバイス・ドライバ・エントリ・テーブル情報のサンプル・ソース・ファイルを提供しています。

【 CA850 対応版 ,CCV850E 対応版の場合 】
 nectools32\src\netconf\<conf_name>\src\rxnetconf.c

6.3 データ・マクロ

RX-NET(PPP) が RX-NET(PPP) 依存部として切り出しているドライバ関数を発行する際に使用する各種データ・マクロ (データ・タイプなど) について以下に示します。

6.3.1 データ・タイプ

表 6-1 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 nectools32\inc850\rxnet.h から呼び出されるヘッダ・ファイル nectools32\inc850\rxnet\ccdep.h ,および ,標準ヘッダ・ファイル nectools32\inc850\stdrx85p.h から呼び出されるヘッダ・ファイル nectools32\inc850\types.h , types.inc で行われています。

表 6-1 データ・タイプ

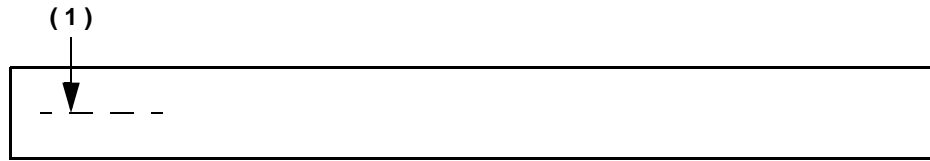
マクロ	型	意味
u8	unsigned char	符号無し 8 ビット整数
u16	unsigned short	符号無し 16 ビット整数
u32	unsigned int	符号無し 32 ビット整数
INT	int	符号付き 32 ビット整数
ID	short	管理オブジェクトの ID 番号

注意 標準ヘッダ・ファイル nectools32\inc850\rxnet.h ,および ,ヘッダ・ファイル nectools32\inc850\rxnet\ccdep.h は ,RX-NET(TCP/IP) が提供しています。

また、標準ヘッダ・ファイル `nctools32\inc850\stdrx85p.h`、および、ヘッダ・ファイル `nctools32\inc850\types.h`、`types.inc` は、RX850 Pro が提供しています。

6.4 ドライバ関数解説

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



(2) —▶ 概要

(3) —▶ C 言語形式

(4) —▶ パラメータ

I/O	パラメータ	説 明

(5) —▶ 機能

(6) —▶ 戻り値

(1) 名称

ドライバ関数の名称を示しています。

(2) 概要

ドライバ関数の機能概要を示しています。

(3) C 言語形式

ドライバ関数を RX-NET(PPP) から呼び出す際の記述形式を示しています。

(4) パラメータ

ドライバ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 ... ドライバ関数への入力パラメータ
- O ... ドライバ関数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

(5) 機能

ドライバ関数の機能詳細を示しています。

(6) 戻り値

ドライバ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。

6.4.1 外部インタフェース仕様

表 6-2 に、RX-NET(PPP) が提供しているドライバ関数の一覧を示します。

表 6-2 ドライバ関数

ドライバ関数名	機能概要
serial_updown	ppp_connect , または , ppp_disconnect を発行した際に呼び出される開始 / 終了関数
serial_write	RX-NET(PPP) が送信処理を実行した際に呼び出される送信関数
serial_read_buf	RX-NET(PPP) が受信処理を実行した際に呼び出される受信関数
serial_task	ネットワーク割り込みハンドラから呼び出されるネットワーク・タスク
serial_intr	受信割り込み, 送信完了割り込み, エラー割り込みが発生した際に呼び出されるネットワーク割り込みハンドラ

次頁以降に、各種ドライバ関数の外部インタフェース仕様詳細を示します。

serial_updown

概要

ppp_connect , または , ppp_disconnect を発行した際に呼び出される開始 / 終了関数

C 言語形式

```
int serial_updown ( netdev *ndp , u16 flags );
```

パラメータ

I/O	パラメータ	説明
I	netdev *ndp ;	システム予約領域
I	u16 flags ;	呼び出し元判定フラグ 0x0 : ppp_connect からの呼び出し 0x1 : ppp_disconnect からの呼び出し

機能

RX-NET(PPP) が提供している API 関数 ppp_connect , または , ppp_disconnect を発行した際に呼び出される開始 / 終了関数です。

以下に , 本ドライバ関数で実行すべき処理を示します。

【 ppp_connect から呼び出された場合 】

- ネットワーク・タスク状態フラグ rtsk_cre に 0x0 (未生成) を設定
- DCD 状態フラグ RXNET_DCD_SEM に 0x0 (未接続) を設定
- 送信バッファの初期化
- 受信バッファの初期化
- マスカブル割り込みの受け付け禁止
- ネットワーク割り込みハンドラ serial_intr の登録
- ネットワーク・コントローラの初期化 (動作開始)
- UART 状態フラグ serial_isup に 0x1 (UP) を設定
- ネットワーク・タスク serial_task の生成
- ネットワーク・タスク状態フラグ rtsk_cre に 0x1 (生成済み) を設定
- ネットワーク・タスク serial_task の起動
- マスカブル割り込みの受け付け許可
- 該当処理が正常終了したか否かを本ドライバ関数の戻り値として設定

【 ppp_disconnect から呼び出された場合 】

- マスカブル割り込みの受け付け禁止
- UART 状態フラグ serial_isup に 0x0 (DOWN) を設定
- ネットワーク割り込みハンドラ serial_intr の登録解除
- ネットワーク・タスク serial_task に対する起床要求の発行
- マスカブル割り込みの受け付け許可
- 該当処理が正常終了したか否かを本ドライバ関数の戻り値として設定

注意 RX-NET(PPP) では、本ドライバ関数のサンプル・ソース・ファイルを提供しています。

【CA850 対応版，CCV850E 対応版の場合】

nectools32\src\rxnet\drivers*driver_name*\devnameapi.c

戻り値

正常終了	0x0
異常終了	0x0 以外

serial_write

概要

RX-NET(PPP) が送信処理を実行した際に呼び出される送信関数

C 言語形式

```
void serial_write ( char *buf , int buf_len , netdev *ndp );
```

パラメータ

I/O	パラメータ	説明
I	char * <i>buf</i> ;	送信するパケットを格納した領域へのポインタ
I	int <i>buf_len</i> ;	<i>buf</i> で指定された領域のサイズ (単位 : バイト)
I	netdev * <i>ndp</i> ;	システム予約領域

機能

RX-NET(PPP) が送信処理を実行した際に呼び出される送信関数です。
以下に、本ドライバ関数で実行すべき処理を示します。

- 排他制御処理の開始宣言
- マスカブル割り込みの受け付け禁止
- *buf* で指定された領域に格納されているパケットを送信バッファに格納
- マスカブル割り込みの受け付け許可
- 排他制御処理の終了宣言

注意 RX-NET(PPP) では、本ドライバ関数のサンプル・ソース・ファイルを提供しています。

【CA850 対応版, CCV850E 対応版の場合】

nectools32\src\rxnet\drivers*driver_name*\devnameapi.c

戻り値

なし

serial_read_buf

概要

RX-NET(PPP) が受信処理を実行した際に呼び出される受信関数

C 言語形式

```
int serial_read_buf ( char *buf , int buf_len );
```

パラメータ

I/O	パラメータ	説明
O	char *buf;	受信したパケットを格納する領域へのポインタ
I	int buf_len;	buf で指定された領域のサイズ (単位 : バイト)

機能

RX-NET(PPP) が受信処理を実行した際に呼び出される受信関数です。
以下に、本ドライバ関数で実行すべき処理を示します。

- 排他制御処理の開始宣言
- マスカブル割り込みの受け付け禁止
- 受信バッファに格納されているパケットを buf で指定された領域に格納
- マスカブル割り込みの受け付け許可
- 排他制御処理の終了宣言
- buf で指定された領域に格納したパケットのサイズ (単位 : バイト) を本ドライバ関数の戻り値として設定

注意 RX-NET(PPP) では、本ドライバ関数のサンプル・ソース・ファイルを提供しています。

【CA850 対応版，CCV850E 対応版の場合】
nectools32\src\rxnet\drivers\

戻り値

正常終了 buf で指定された領域に格納したパケットのサイズ (単位 : バイト)

serial_task

概要

ネットワーク割り込みハンドラから呼び出されるネットワーク・タスク

C 言語形式

```
void serial_task ( INT stacd );
```

パラメータ

I/O	パラメータ	説明
I	INT <i>stacd</i> ;	システム予約領域

機能

ネットワーク割り込みハンドラ `serial_intr` から呼び出されるネットワーク・タスクです。
以下に、本ドライバ関数で実行すべき処理 (3 種類) を示します。

【 起動待ちループ処理 】

- RX-NET(PPP) の初期化処理, DIAL 処理, LOGIN 処理などが完了したか否かを定期的にポーリング

【 送受信ループ処理 】

- ネットワーク・タスク (自タスク) を wait 状態 (起床待ち状態) へと遷移
- 排他制御処理の開始宣言
- マスカブル割り込みの受け付け禁止
- 受信バッファからパケットを獲得
- 獲得したパケットを RX-NET(PPP) に通知
- マスカブル割り込みの受け付け許可
- 排他制御処理の終了宣言

【 終了処理 】

- 排他制御処理の開始宣言
- UART レベルで接続が遮断した旨を RX-NET(PPP) に通知
- ドライバ関数 `serial_updown` の呼び出し
- 排他制御処理の終了宣言

注意 1 本ドライバ関数の送受信ループ処理は、API 関数 `ppp_disconnect` が発行された際、または、DCD 状態フラグ `RXNET_DCD_SEM` に `0x0` (未接続) が設定されていることを検出した際に終了します。

注意 2 RX-NET(PPP) では、本ドライバ関数のサンプル・ソース・ファイルを提供しています。

【 CA850 対応版, CCV850E 対応版の場合 】

`nectools32\src\rxnet\drivers\<driver_name>\devnameapi.c`

戻り値

なし

serial_intr

概要

受信割り込み，送信完了割り込み，エラー割り込みが発生した際に呼び出されるネットワーク割り込みハンドラ

C 言語形式

```
ID    serial_intr ( void );
```

パラメータ

なし

機能

受信割り込み，送信完了割り込み，エラー割り込みが発生した際に呼び出されるネットワーク割り込みハンドラです。以下に，本ドライバ関数で実行すべき処理を示します。

- 割り込み要因 (受信割り込み，送信完了割り込み，エラー割り込み) の種別判定
- 割り込み要因の種別に対応した処理
 - 【 受信割り込み発生時に呼び出された場合 】
 - ネットワーク・コントローラからパケットを獲得
 - 獲得したパケットを受信バッファに格納
 - 【 送信完了割り込み発生時に呼び出された場合 】
 - 送信バッファからパケットを獲得
 - 獲得したパケットをネットワーク・コントローラに通知
 - 【 エラー割り込み発生時に呼び出された場合 】
 - エラーの発生要因に対応した処理
- モデム制御信号 DCD の確認
 - 【 モデム制御信号 DCD が 0x1 → 0x0 に変化した場合 】
 - UART 状態フラグ serial_isup に 0x0 (DOWN) を設定
 - DCD 状態フラグ RXNET_DCD_SEM に 0x0 (未接続) を設定
 - 【 モデム制御信号 DCD が 0x0 → 0x1 に変化した場合 】
 - DCD 状態フラグ RXNET_DCD_SEM に 0x1 (接続) を設定
- プロセッサの割り込み終了処理
- ネットワーク・タスクの ID 番号を本ドライバ関数の戻り値として設定

注意 RX-NET(PPP) では，本ドライバ関数のサンプル・ソース・ファイルを提供しています。

【 CA850 対応版，CCV850E 対応版の場合 】
nectools32\src\rxnet\drivers*<driver_name>*\devnameisrsub.c

戻り値

正常終了

ネットワーク・タスクの ID 番号

索引

A

API 関数	34, 41, 48
ppp_connect	34, 41, 48, 49
ppp_disconnect	36, 41, 48, 51
ppp_getNameServerIP	39, 41, 48, 54
ppp_getNameServerIP2	40, 41, 48, 55
ppp_setChapChallengeMsg	38, 41, 48, 53
ppp_setPassDB	37, 41, 48, 52
外部インタフェース仕様	48
データ構造体	44
データ・マクロ	42
呼び出し方法	41

C

cf850pro	23
CF 定義ファイル	23
SCT 情報	23
SIT 情報	23
情報ファイル	23
CHAP 認証	29

F

fnsconfig.h	41
-------------------	----

I

IPCP 通信設定要求	32
IPCP 通信設定要求応答	33

L

LCP 通信設定要求	32
LCP 通信設定要求応答	32
libppp.a	18, 20
LOGIN 認証	29

M

Makefile	18, 19
----------------	--------

N

netconf.bld	20
-------------------	----

P

PAP 認証	29
ppp_connect	34, 41, 48, 49

ppp_disconnect	36, 41, 48, 51
ppp_getNameServerIP	39, 41, 48, 54
ppp_getNameServerIP2	40, 41, 48, 55
ppp_setChapChallengeMsg	38, 41, 48, 53
ppp_setPassDB	37, 41, 48, 52
PPP ライブラリ	18, 20
libppp.a	18, 20

R

README.PPP	19, 20, 21
RFC	14
RX850 Pro 依存部	24
エントリ処理	24
初期化ハンドラ	24
ハードウェア初期化部	24
ブート処理	24
RX850 Pro 用標準ヘッダ・ファイル	41
stdrx85p.h	41
rxnetcfg_<driver_name>.o	18, 20
rxnet.h	41
RX-NET(PPP)	13
API 関数	41
RX-NET(PPP) 依存部	56
位置付け	13
インストレーション	16
階層的 position	14
開発環境	15
システム構築	22
実行環境	15
特徴	14
ネットワーク接続機能	29
rxnet_ppp.h	18, 20, 41
RX-NET(PPP) 依存部	56
基本情報	25, 56
ドライバ関数	25
RX-NET(PPP) 用標準ヘッダ・ファイル	18, 20, 41
rxnet_ppp.h	18, 20, 41
RX-NET(TCP/IP) 依存部	25
RX-NET(TCP/IP) 用標準ヘッダ・ファイル	41
rxnet.h	41

S

sample.bld	20
SCT 情報	23
タスク管理機能情報	23
タスク付属同期機能情報	23
割り込み処理管理機能情報	23
serial_intr	25, 56, 61, 67
serial_read_buf	25, 56, 61, 65
serial_task	25, 56, 61, 66
serial_updown	25, 56, 61, 62
serial_write	25, 56, 61, 64

SIT 情報	23
システム最大値情報	23
システム情報	23
stdrx85p.h	41

あ

アーカイブ・オブジェクト 27

い

インストール 16
UNIX ベース 17
Windows ベース 16

え

エントリ処理 24

お

オブジェクト・ファイル 26

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 48, 61
拡張 SVC ハンドラ 26
拡張 SVC ハンドラ用インタフェース・ルーチン 26
間接起動割り込みハンドラ 26

き

基本情報 25, 56
デバイス・ドライバ・エントリ・テーブル情報 56

こ

コンフィギュレータ 23
cf850pro 23

し

システム構築 22
CF 定義ファイル 23
RX850 Pro 依存部 24
RX-NET(TCP/IP) 依存部 25
アーカイブ・オブジェクト 27
オブジェクト・ファイル 26
処理プログラム 26
リンク・ディレクティブ・ファイル 27
ロード・モジュール 27
システム・コール・テーブル 23
システム最大値情報 23
システム情報 23
システム情報テーブル 23
システム情報ヘッダ・ファイル 23

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 26
情報ファイル 23
システム・コール・テーブル 23
システム情報テーブル 23
システム情報ヘッダ・ファイル 23
初期化ハンドラ 24
処理プログラム 26
拡張 SVC ハンドラ 26
拡張 SVC ハンドラ用インタフェース・ルーチン 26
間接起動割り込みハンドラ 26
周期起動ハンドラ 26
タスク 26
直接起動割り込みハンドラ 26

せ

静的設定情報ヘッダ・ファイル 41
fnsconfig.h 41
接続形態 30
ヌルモデム接続 30
モデム接続 31
接続条件 42

た

タスク 26
タスク管理機能情報 23
タスク付属同期機能情報 23

ち

直接起動割り込みハンドラ 26

つ

通信オプション 32
IPCP 通信設定要求 32
IPCP 通信設定要求応答 33
LCP 通信設定要求 32
LCP 通信設定要求応答 32

て

ディレクトリ構成 18
CA850 対応版 18
CCV850E 対応版 20
データ構造体 44
モデム制御シーケンス情報 44
データ・タイプ 42, 57
データ・マクロ 42, 57
接続条件 42
データ・タイプ 42, 57
戻り値 43

テキスト・ファイル	19, 20
README.PPP	19, 20, 21
デバイス・ドライバ・エントリ・テーブル情報	56
ndevsw	56
デバイス・ドライバ・オブジェクト	18, 20
rxnetcfg_<driver_name>.o	18, 20

と

ドライバ関数	25, 56
serial_intr	25, 56, 61, 67
serial_read_buf	25, 56, 61, 65
serial_task	25, 56, 61, 66
serial_updown	25, 56, 61, 62
serial_write	25, 56, 61, 64
外部インタフェース仕様	61
データ・マクロ	57

に

認証プロトコル	29
CHAP 認証	29
LOGIN 認証	29
PAP 認証	29

ぬ

ヌルモデム接続	30
---------------	----

ね

ネットワーク接続機能	29
API 関数	34
接続形態	30
通信オプション	32
認証プロトコル	29

は

ハードウェア初期化部	24
------------------	----

ひ

ビルド・ファイル	20
netconf.bld	20
sample.bld	20

ふ

ブート処理	24
-------------	----

め

メイク・ファイル	18, 19
Makefile	18, 19

も

モデム制御シーケンス情報	44
モデム接続	31
戻り値	43

り

リンク・ディレクティブ・ファイル	27
------------------------	----

ろ

ロード・モジュール	27
-----------------	----

わ

割り込み処理管理機能情報	23
--------------------	----

— 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

【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 PPP Ver.1.30 ユーザーズ・マニュアル
(U15303JJ3V0UM00 (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してください。

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