エッジゲートウェイ　SDK 利用ガイド

ご注意

(1) 本書は内容について万全を期して作成いたしましたが、万一ご不審な点や誤り、記載漏れなどお気づきのことがありましたらご連絡ください。
(2) 運用した結果の影響について(1)にかかわらず責任を負いかねますのでご了承ください。

- Linux は、Linus Torvalds 氏の米国およびその他の国における登録商標です。
- Debian は Software in the Public Interest, Inc. の登録商標です。
- Oracle VM VirtualBox は Oracle Corporation の登録商標です。
- その他、記載されている商品名、会社名などの固有名詞は各社の商標または登録商標です。
- 本文中および図中では、TM、®マークは表記しておりません。

目次

1　はじめに..... 4
1.1　本書の位置づけ 4
1.2　目的 4
1.3　前提条件..... 4
2　SDK 概要..... 5
2.1　動作環境..... 5
2.2　ディレクトリ構成 5
3　エッジ GW 内蔵アプリケーションの利用方法..... 6
4　920MHz テストツール 9
4.1　920MHz テストツールプログラムの構成および概要..... 9
5　SDK 操作.....17
5.1　SDK の準備 17
5.2　アプリケーション実行のスクリプト作成 18
5.3　FW 書き換え.....20

1 はじめに

1.1 本書の位置づけ

本書では、エッジゲートウェイ(以下、エッジＧＷ)の Software Development Kit(以下、SDK)での操作に関して記載します。

1.2 目的

本書は、SDK を利用した FW(1.3 章参照)、アプリケーションの作成ができることを目的としています。

1.3 前提条件

本書は、エッジ GW を購入し、システム開発を考えているソフトウェア開発者を対象としています。また、Linux OS(Debian)についての基礎知識があることを前提としています。

本資料で記載している「FW」は「NEC でカスタマイズした OS に各種アプリケーションを組み込んだもの」と定義します。

SDK の入手については販売元あるいは以下のメール宛にお問い合わせください。

edge-products@ppf.jp.nec.com

SDK およびアプリケーションは C 言語ベースでの開発を前提としております。

2 SDK 概要

2.1 動作環境

本 SDK を動作するには、下記の環境が必要です。

- Oracle 社 VirtualBox (x86-64 ベース・システム用のクロスプラットフォーム仮想化ソフトウェア)以下、VirtualBox と略す。
- VirtualBox の仮想ハードドライブファイル(vdi)をサポートしている OS を搭載するパソコン
- VirtualBox 動作環境は公式サイト(https://www.virtualbox.org/)を参照ください。

VirtualBox は配布メディア内に格納しています。なお、本 SDK は VirtualBox の Version 6.0.2 にて動作確認済みです。最低限 CPU1GHz、メモリ 2GB、ストレージ 20GB のスペックを持つパソコンが必要です。最新版は上記 URL の製品情報より取得してください。

2.2 ディレクトリ構成

本ＳＤＫのディレクトリ構成と格納ファイルの説明について記載します。

/home/edgegw/sdk/に以下のディレクトリがあります。

No	ディレクトリ	ファイル	説明
1	/free_app/test/	Makefile test.sh test_fw.sh	アプリケーションのサンプルスクリプト
2	/free_app/test/src/	test.c	アプリケーションのサンプルソースコード
3	/free_app/920TestTool/	920_module.sh 920_unit.sh Makefile	920MHz テストツールのサンプルスクリプト
4	/free_app/920TestTool/include/	920_module.h	920MHz テストツールのサンプルインクルードファイル
5	/free_app/920TestTool/src/	920_module.c 920_module.h 920_module_reset.c 920_module_send.c 920_module_set.c 920_module_start.c 920_module_status.c 920_unit_echo.c	920MHz テストツールのサンプルソースファイル

3 エッジ GW 内蔵アプリケーションの利用方法

1. LED 操作に関して

エッジ GW には PG0,PG1 の 2 つの LED があり、下記にて操作することが可能です。

✧ 機能

LED の PG0,PG1 を指定し、点灯、消灯、点滅(1 秒間隔、0.5 秒間隔)を制御する。

✧ コマンド

下記 2 つのコマンド実行方法があります。

✓ /opt/nec/pf/edgetool/bin/LED 第一引数 第二引数

引数	設定値	説明
第一引数	on	LED を点灯
	off	LED を消灯
	blink	LED を点滅(1 秒間隔)
	blink_fast	LED を点滅(0.5 間隔)
第二引数	0	PG0 の LED を制御
	1	PG1 の LED を制御

✓ /opt/nec/pf/edgetool/bin/LED –r リソースファイルのパス ステータス名

(以降、リソースファイル読込みモード)

リソースファイルに LED 制御を status で区別して定義し、それを指定することができます。

リソースファイル作成は「ステータス名,第一引数 第二引数」を一行単位で記載します。

例)

```
status1,blink 1
status2,on 1
status3,off 0
```

✧ 戻り値

戻り値	説明
0	成功
1	バイナリ内の関数処理エラー
2	パラメータ不足・過剰エラー(リソースファイル読込みモードの有無にかかわらず)
3	パラメータ不足・過剰エラー(リソースファイル読込みモードでない場合)

4	パラメータの形式エラー(リソースファイル読込みモードの場合)
5	入力コマンドのフォーマットエラー
6	リソースファイルのフォーマットエラー
7	LED 制御失敗
8	LED 制御プロセスへの接続エラー
9	LED 制御プロセスへメッセージ送信エラー
10	LED 制御プロセスからのメッセージ受信エラー

2.　BEEP 操作に関して

エッジ GW には BEEP が内蔵されており、下記にて操作することが可能です。

✧　機能

BEEP の開始/停止を行う。

✧　コマンド

下記 2 つのコマンド実行例があります。

[BEEP 動作を開始する場合]

/opt/nec/pf/edgetool/bin/BEEP start [1/2/3/4/5]

第 2 引数の数字を変えることで、動作パターンを変えることが出来ます。

第 2 引数の数字が大きくなるに従い、ビープ音と無音状態の時間が短くなります。

例:

第 2 引数が 1 の場合、約 1 秒のビープ音と約 0.5 秒の無音状態を繰り返します。

第 2 引数が 5 の場合、約 0.25 秒のビープ音と約 0.1 秒の無音状態を繰り返します。

- [BEEP 動作を停止する場合]

  /opt/nec/pf/edgetool/bin/BEEP stop

✧ 戻り値

戻り値	説明
0	成功
1	状態エラー(すでに BEEP が動作している、停止している)
4	引数エラー
上記値以外	その他のエラー

3. プロセス監視ツールに関して

エッジ GW には、異常状態検知のために、プロセス監視ツールが内蔵、動作しています。

✧ 機能

監視リストファイル(watchctl.xml)の内容に従い、プロセスの監視を行う。

監視リストファイル(watchctl.xml)の書式は以下のとおりです。

```
<?xml version="1.0" encoding="utf-8" ?>
<systemControlInfo>
  <resettime>180</resettime>
  <looptime>50</looptime>
  <maxretry>3</maxretry>
  <watchdog>/dev/watchdog1</watchdog>
  <watchlist>
    <process>cron</process>
    <process>sshd</process>
    <process>node</process>
  </watchlist>
</systemControlInfo>
```

項目	説明
resettime	指定した時間内に監視対象プロセスが見つからない、OS が再起動する
looptime	プロセスをチェックする間隔
maxretry	プロセスチェック時に存在しないプロセスへのリトライ回数
watchlist-> process	監視を行うプロセス名

4 920MHz テストツール

4.1 920MHz テストツールプログラムの構成および概要

エッジ GW に 920MHz 無線内蔵オプションのテスト用のプログラムです。SDK に付属するサンプルアプリケーションの 920MHz テストツールプログラムについて構成および概要を説明します。下記を参考にし、アプリケーションの環境にあわせて適宜カスタマイズしてください。なお、本資料の 5.2 章を参照し、ビルドを実行してください。

注意事項

920MHz 親機および子機の設定につきましては、OKI 社が作成した以下のマニュアルを参照していただき、ソフトウェアをご使用ください。

https://www.oki.com/jp/920M/
取り扱い説明書・ソフトウェア

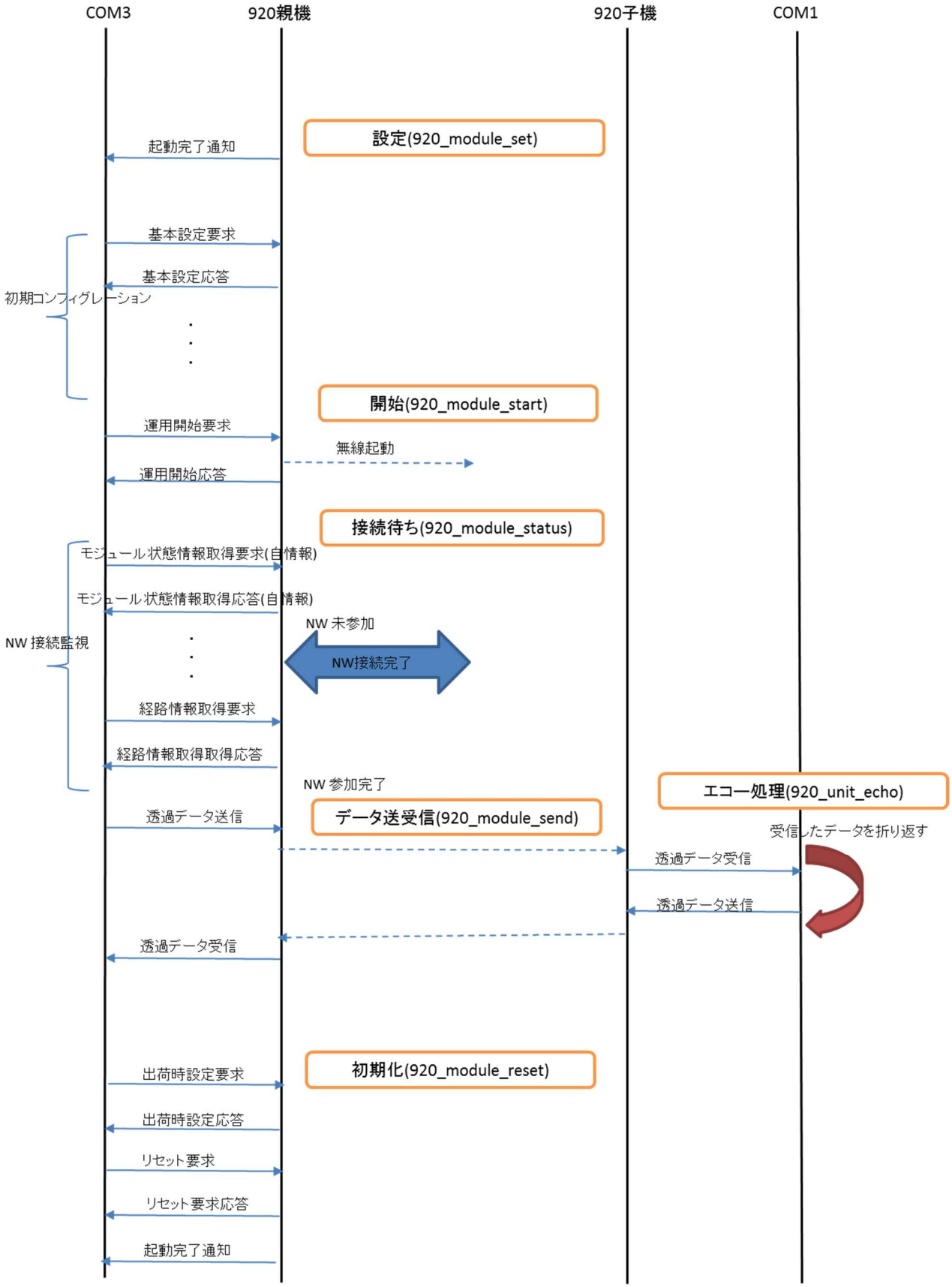
・920 親機：「920Hz 帯無線通信モジュール　透過モデル 取扱説明書」

・920 子機：「920Hz 帯無線ユニット　透過モデル 取扱説明書」

1. 920MHz テストツールプログラムの構成

項番	ファイル名	概要	備考
1	Makefile	実行ファイルを作成する Makefile	
2	920_module.h	定数等を定義するヘッダファイル	
920 親機用			
3	920_module_set.c	920 親機に対してパラメータを設定する	
4	920_module_start.c	920 親機に接続開始を送信する	
5	920_module_status.c	920 親機の状態を確認する	
6	920_module_send.c	920 親機から子機へデータを送信する	
7	920_module_reset.c	920 親機の設定および状態をリセットする	
8	920_module.c	920 親機の電文送受信関数群	
9	920_module.sh	920 テストツールの実行スクリプト	
920 子機用			
10	920_unit_echo.c	920 子機で受信したデータをエコーする	
11	920_unit.sh	920 子機用のエコーするツールの実行スクリプト	

2.　920MHz テストツールサンプルの処理概要



3. 920 設定処理の流れ(920_module_set.c)

初期処理
　パラメータチェック
　ログファイルオープン
　シリアルオープン
　シグナル設定
　シリアル設定退避
　シリアル設定
電文受信処理
　起動完了通知を受信
　　親機の電源 ON 後、親機が起動したらこの起動完了通知が送信されるのでこの起動完了通知の受信後設定処理を開始すること
電文送信処理
　基本設定要求を送信
　　親機がもつ基本設定の情報を取得して、設定を変更する。
　　このサンプルでは、基本設定情報は固定情報としてテーブル cmd_01w[]で持ち、親機と子機を１：１で接続する設定としている。(固定で情報を持っているため、基本設定の情報取得処理はしていない)
　　設定を変更する場合、PAN-ID(グループ番号)、チャネル番号、ネットワーク名、暗号鍵、監視単位時間は、同一 NW に参加する 920 子機と同じ値を設定する必要がある。
　　※OKI 社の「無線通信モジュール／ホスト CPU 間　API 仕様書」⇒5 要求コマンド詳細(通常フォーマット、通常モード)⇒5.1 基本設定/取得要求【msgid:0x01】を参照。
電文受信処理
　基本設定要求応答を受信
電文送信処理
　局番解析用データ設定要求を送信
　　このサンプルでは、固定情報としてテーブル cmd_06w[]で持っている。
電文受信処理
　局番解析用データ設定要求応答を受信
電文送信処理
　暗号化鍵登録要求を送信
　　このサンプルでは、固定情報としてテーブル cmd_10w[]で持っている。
　　暗号化鍵を変更する場合は、このテーブル cmd_10w[]を変更する。また、この値は 920 子機と同じ値であること。
電文受信処理

暗号化鍵登録要求応答を受信
電文送信処理
プレフィックス設定要求を送信
このサンプルでは、固定情報としてテーブル cmd_85w[]で持っている。
電文受信処理
プレフィックス設定要求応答を受信
電文送信処理
基本設定取得要求を送信
この処理は基本設定情報の確認用に使用している。
電文受信処理
基本設定取得要求応答を受信
終了処理
シリアル設定復帰
シリアルクローズ
ログファイルクローズ

4. 920 親機から 920 子機への接続(920_module_start.c)

初期処理
パラメータチェック
ログファイルオープン
シリアルオープン
シグナル設定
シリアル設定退避
シリアル設定
電文送信処理
運用開始要求を送信
電文受信処理
運用開始要求応答を受信
終了処理
シリアル設定復帰
シリアルクローズ
ログファイルクローズ

5. 920 親機の接続状態の確認(920_module_status.c)

初期処理
パラメータチェック

```
　ログファイルオープン
　シリアルオープン
　シグナル設定
　シリアル設定退避
　シリアル設定
電文送信処理
　モジュール状態情報取得要求を送信
電文受信処理
　モジュール状態情報取得要求応答を受信
モジュール状態チェック
電文送信処理
　経路情報取得要求を送信
電文受信処理
　経路情報取得要求応答を受信
設定経路チェック
終了処理
　シリアル設定復帰
　シリアルクローズ
　ログファイルクローズ
```

6.　920 親機から 920 子機へデータを送信し、920 子機からのデータを受信(920_module_send.c)

```
初期処理
　パラメータチェック
　ログファイルオープン
　シリアルオープン
　シグナル設定
　シリアル設定退避
　シリアル設定
電文送信処理
　ランダムデータ作成
　データ送信
電文受信処理
　データ受信
データチェック
終了処理
　シリアル設定復帰
```

```
  シリアルクローズ
  ログファイルクローズ
```

7. リセット(920_module_reset.c)

```
初期処理
  パラメータチェック
  ログファイルオープン
  シリアルオープン
  シグナル設定
  シリアル設定退避
  シリアル設定
電文送信処理
  出荷時設定要求を送信
電文受信処理
  出荷時設定要求応答を受信
電文送信処理
  リセット要求を送信
電文受信処理(応答待ち)
  リセット要求応答を受信
電文受信処理(初期化の通知待ち)
  起動完了通知を受信
終了処理
  シリアル設定復帰
  シリアルクローズ
  ログファイルクローズ
```

8. エコー(920_unit_echo.c)

```
初期処理
  パラメータチェック
  シリアルオープン
  シグナル設定
  シリアル設定退避
  シリアル設定
LOOP
  電文受信処理
    電文を受信するまで待つ
  電文送信処理
```

受信したデータをそのまま送信データにする 終了処理 シリアル設定復帰 シリアルクローズ

9. 920 親機電文送受信関数群(920_module.c)

項番	関数名	概要	引き数	戻り値
1	ReceiveCommand	コマンド受信処理	dt：920 親機用受信構造体ポインタ	0： 正常終了 1： 異常終了(タイムアウト)
2	SendCommand	コマンド送信処理	dt：920 親機用送信構造体ポインタ	0： 正常終了 1： 異常終了(write 処理でエラー)
3	ReceiveData	透過データ用受信処理	dt：920 親機用受信構造体ポインタ	0： 正常終了 1： 異常終了(タイムアウト) 2： 受信データなし
4	SendData	透過データ送信処理(4 バイトのデータのみ設定する)	dt　　：920 親機用送信構造体ポインタ rand_dt： 送信する 4 バイトデータのポインタ out_tmp： 送信したデータを保存するバッファポインタ	0： 正常終了 1： 異常終了(write 処理でエラー)

5 SDK 操作

SDK の各機能の操作について記載します。

5.1 SDK の準備

下記の手順をおこない、SDK を準備します。

1. SDK の VM 起動

   SDK がインストールされている VirtualBox 用 vdi ファイル（EdgeGW_SDK_”日付”.vdi）を使用して、VM を起動します。

2. SDK ログイン

   ログイン ID/パスワードを入力し、ログインします。

3. ターミナル起動

   メニューボタンを押下して、ターミナルを指定して起動します。

4. SDK のディレクトリに移動します

```
$cd /home/edgegw/sdk
```

注意事項

ログイン ID/パスワードに入力する値は、SDK を格納しているメディア内に付属しています。

5.2　アプリケーション実行のスクリプト作成

アプリケーションのビルドがおこなえます。下記手順は SDK に付属しているサンプルのアプリケーション(“SLEEP_TEST”メッセージを 5 秒置きに標準出力)をもとに説明します。環境にあわせて適宜カスタマイズしてください。

1. サンプル用アプリケーションをビルドする用のディレクトリに移動

```
$cd /home/edgegw/sdk/free_app
```

2. サンプル用アプリケーションのビルド実行

```
$cd test
$make
```

3. 下記の実行ファイルが作成される

   /home/edgegw/sdk/free_app/test/test

4. アプリケーションをサービス起動する場合は、以下の記載を参考としてください。

```
# /etc/systemd/system/edge-ap1.service ←(※1)
[Unit]
Description=Edge AP1 Service  ←(※2)
After=edge-pfb.target  ←(※3)
Before=edge-ap.target ←(※4)
[Service]
Type=oneshot ←(※5)
RemainAfterExit=yes ←(※6)
ExecStart=/bin/sleep 0.2 ←(※7)
[Install]
WantedBy=edgesw.target ←(※8)
```

- ✓ ※1:ファイル名

  /etc/systemd/system/に任意の名前の設定ファイルを格納してください。
- ✓ ※2: Description

  スクリプトの説明を記述します。
- ✓ ※3:after 設定

  「/etc/systemd/system/edge-pfb.target」の後で自サービスを開始します。

記載は変更しないでください。

- ✓ ※4:brfreShort-Description
  「/etc/systemd/system/edge-pfb.target」よりも前に自サービスを開始します。
  記載は変更しないでください。
- ✓ ※5: Type
  「ExecStart」で指定したコマンドでサービスのプロセスを起動した際に、起動完了をどのように判定するかを指定します。アプリの用途に応じて記載してください。
- ✓ ※6: RemainAfterExit
  コマンドが完了したあともサービスとしては起動状態として認識させるかを指定します。アプリの用途に応じて記載してください。
- ✓ ※7: ExecStart
  起動時に実行するコマンドまたはスクリプトを指定します。アプリの用途に応じて記載してください。
- ✓ ※8: WantedBy
  「edgesw.target」の起動モード（ランレベル）でのみ有効。記載は変更しないでください。

5. syslog に関して
   - ✧ アプリケーションで使用できる syslog のファシリティは下記のみです。
     なお、local1～local4 はエッジ GW の基盤用のため、使用しないでください。
     - ✓ user
     - ✓ local5～local7
   - ✧ syslog は/var/log/に格納されます。ただし、/var/log/は RAM ディスクのため、エッジ GW を再起動すると/var/log/配下のファイルは消えます。ログファイルは/var/oldlogs/に logrotate されます。

5.3　FW 書き換え

FW 書き換え方法については販売元あるいは以下のメール宛にお問い合わせください。

edge-products@ppf.jp.nec.com

エッジゲートウェイ　SDK 利用ガイド

2020 年 3 月

日本電気株式会社

東京都港区芝五丁目 7 番 1 号

E-mail:edge-products@ppf.jp.nec.com

© NEC Corporation 2020
日本電気株式会社の許可なく複製・改変などを行うことはできません。
本書の内容の一部または全部を無断転載することは禁止されています。
本書の内容に関しては将来予告なしに変更することがあります。