

IEEE802.15.4通信サンプルソフトウェア

# SimpleMACstd24X・取り扱い説明書

Ver 3.5

**RF****LINK**

# 目次

1. はじめに
2. SimpleMACstdの概要
3. SimpleMACstdの基本動作
4. コンフィグレーション項目一覧
  - 4.1 コンフィグレーション "a" ~ "l" まで
  - 4.2 コンフィグレーション "m" ~ "z" まで
  - 4.3 コンフィグレーションの使用例
5. シンプルモードと対話モード
6. コンフィグレーション項目・詳細
  - 6.1 使用CH番号の設定コマンド ('a' コマンド)
  - 6.2 ホップカウンタの設定コマンド ('f' コマンド)
  - 6.3 PAN-ID、ネットワークアドレスの設定コマンド ('b' 'c' 'e' コマンド)
  - 6.4 中継通信の使用/不使用の設定コマンド ('f' コマンド)
  - 6.5 タイマーハンドラ機能 使用/不使用の設定コマンド ('k' コマンド)
  - 6.6 スリープ機能の設定コマンド ('l' 'm' 'n' コマンド)
  - 6.7 データ転送機能設定コマンド ('x' コマンド)
  - 6.8 電波受信強度データ(RSSI値)付加コマンド ('p' コマンド)
  - 6.9 PAN-ID機能の有効/無効コマンド ('q' コマンド)
  - 6.10 MAC層リトライ回数設定コマンド ('t' コマンド)
  - 6.11 タイマー同期コマンド ('将来実装' コマンド)
  - 6.12 電波送信出力制御コマンド ('u' 'v' コマンド)
  - 6.13 FlashROM書き込み/読み出しコマンド ('w' 'r' コマンド)
  - 6.14 システム開始コマンド ('s' コマンド)
  - 6.15 モニターモードコマンド ('@' コマンド)
  - 6.16 RSSI値受信閾値設定コマンド ('g' コマンド)
  - 6.17 改行コード出力設定コマンド ('h' コマンド)
7. 応答コード一覧
8. フレームフォーマット (全モード共通)
9. UARTフォーマット (垂れ流しモード/フラグ付き垂れ流しモード)
10. UARTフォーマット (フレームモード)
11. サンプルプログラムの基本構成
  - 11.1 SimpleMACstd のファイル階層図
  - 11.2 コンフィグレーション設定例 ~ 単純なシリアルデータの送受信
  - 11.3 コンフィグレーション設定例 ~ 単純なシリアルデータの 1:Nによる構築
  - 11.4 コンフィグレーション設定例 ~ 1:Nによる子機指定通信
  - 11.5 コンフィグレーション設定例 ~ 経路指定通信を含む 1:Nによる子機指定通信
  - 11.6 コンフィグレーション設定例 ~ スリープ機能を有効にした送信設定
  - 11.7 コンフィグレーションした設定で自動スタートをさせる方法
12. 自動ルーティング機能(オプション)
  - 12.1 自動ルーティングの概要
  - 12.2 自動ルーティングの仕組み

# 1.はじめに

本書では、RM-24X シリーズ用に開発した、通信ソフトウェア(SimpleMACstd)についての概要を説明した資料です。

SimpleMACstdは、IEEE802.15.4通信仕様に準拠した、2.4GHz帯用通信ソフトウェアです。

ZigBeeスタックは使用せずに、通信のリアルタイム性や短時間のネットワーク構築を目的として、MAC層のペイロード層を使用した独自のフォーマットで通信を行います。

ユーザー設定により、電波送信強度の調整、通信経路の設定、リトライ回数の設定など、その他多くの設定項目をお客様のアプリケーション仕様に合わせて自由にカスタマイズする事が出来ます。

本通信スタックは、ZigBeePROが標準サポートする「メッシュプロトコル」に変わる「自動ルーティング」を実装しています。(オプション機能)  
これは、メッシュネットワークのデメリットである、「省電力性」、「リアルタイム性」を改善した弊社独自のプロトコルで、メッシュネットワークの一部機能を制限する代わりに、電力消費、リアルタイム通信性を向上させる事を目的とした中継通信機能です。

## 2. SimpleMACstdの概要

SimpleMACstdは、IEEE802.15.4上で動作するプロトコルスタックです。

図1と図2の通り、SimpleMACstdは、ZigBee仕様に準拠したネットワーク層を持たず、直接MAC層上で動作します。ZigBeeスタックが本来行う、アドレス管理、経路選択、無線CH選択、などを直接コントロールする事で、無線データのスループット性能を向上しています。

また、SimpleMACstdでは、外部にホストマイコン、又はPCなどから送受信するデータをUARTでI/Fする事を前提にしていますが、RM-240内蔵のA/Dや、SPIなどからのデータをスタンドアロンで送信する事も可能です。

図1 ZigBee基本モデル

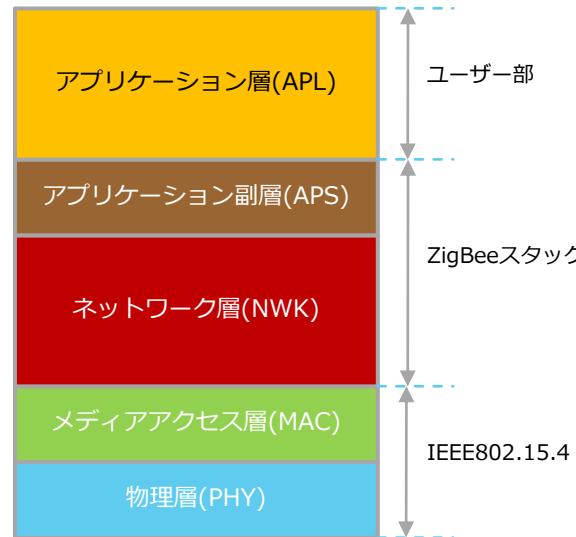
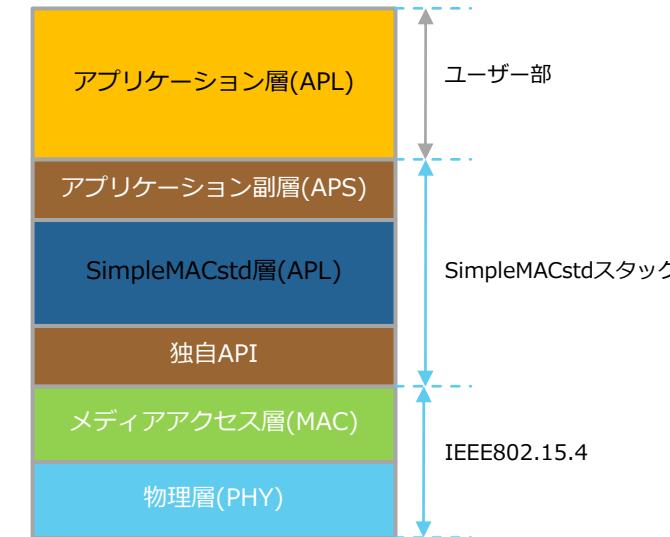


図2 MAC層通信モデル



## 2.SimpleMACstdの概要

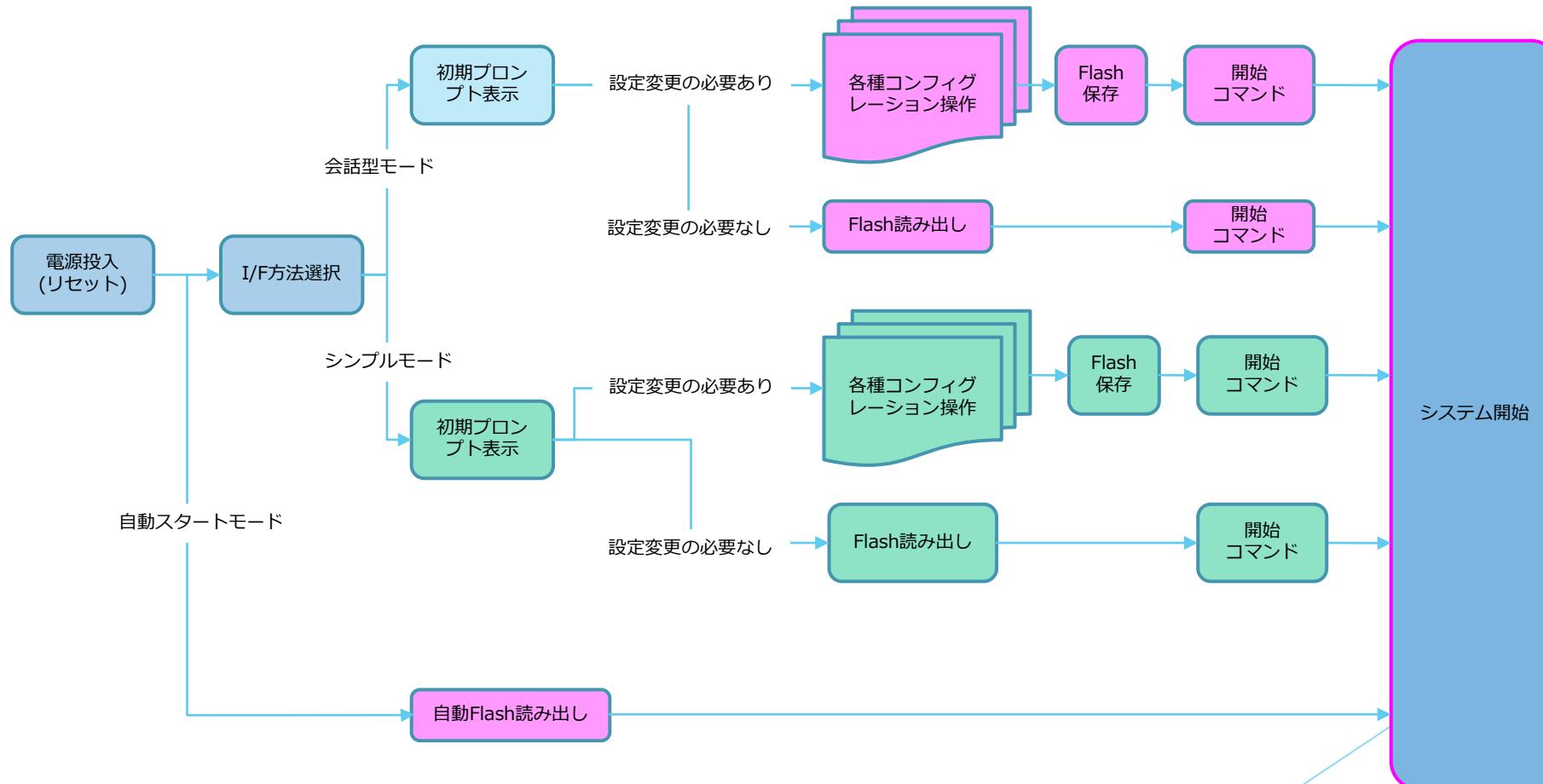
SimpleMACstdの基本的な動作は、RM-240/241 のUARTデバイスを使用して、外部のホストCPU、又はパソコンと送受信して通信する事を前提として設計しています。

ソースコードは技適に抵触する部分以外は公開していますので、ソフトウェアを独自に改造して頂く事で、RM-240シリーズの内部デバイス(A/D、SPI、I2Cなど)を利用して、お客様によるセンサを接続した無線センサネットワークシステムの開発など行う事が出来ます。

※各センサ用のサンプルコードもご用意しています。

### 3. SimpleMACstdの基本動作

SimpleMACstdは、電源投入後UARTにより対話形式で、コンフィグレーション設定を行います。  
最後に、設定情報を内蔵FlashROMに保存し、スタートコマンドを受け付ける事で、システムスタートとなります。  
所定のポートをON(ショート)した状態で電源を入れる事で、自動的にFlashROMから設定情報を読み出してスタートさせる事が出来ます。



# 4.コンフィグレーション項目一覧

## 4.1 コンフィグレーション "a" ~ "l" まで

コマンド	設定項目	出荷時設定	設定範囲	説明
"a"	使用する無線CH番号の設定	11	11~26	2405MHz~2480MHz
"b"	PANネットワークアドレスの設定	1000	1~65535(0xFFFF)	
"c"	自局ノードアドレス	1	1~65535(0xFFFF)	
"d"	マルチホップカウンタ指定	1	1~255	
"e"	対向局ノードアドレス	0	1~65535(0xFFFF)	
"f"	中継・ホップ通信の使用/不使用の切り替え	0	0:中継無し 1:マルチホップ通信 2:中継通信	
"g"	RSSI受信閾値設定	0	0:閾値設定なし 1(-1dBm)~98(-98dBm)	-1dBm~-98dBmの範囲で設定した値よりも弱い電力で受信したデータを破棄
"h"	改行コードUART出力設定	0	0:出力無し 1:出力	受信データのUART出力時に改行コードの出力有無を設定
"j"	タイマーハンドラ周期の設定	0	1~255 (100ms~25.5s)	
"k"	タイマーハンドラ使用/不使用	0	0:不使用 1:使用	
"l"	スリープモードの指定	0	0:不使用 1:タイマー起動 2:GPIO起動	
"m"	スリープタイマー時間の設定	0	0~65530 (250ms単位)	(例) 4 → 1秒
"n"	スリープモードの使用/不使用の設定	0	0:不使用 1:使用	
"o"	UART 転送速度設定	115,200	4,800~960,800	調歩同期モード
"p"	電波受信強度(RSSI)付加	不使用	0:不使用 / 1:使用(HEX) / 2:使用(ASCII)	フラグ付きフレームモードで有効
"q"	PANネットワークアドレスの有効/無効	有効	0:無効 / 1:有効	無効にするとPANによるフィルタ解除
"r"	FlashROMからのコンフィグレーション読み出し	規定値	-	保存コンフィグレーション値読み出し
"s"	システムスタート			
"t"	リトライ回数指定	3	1~255	

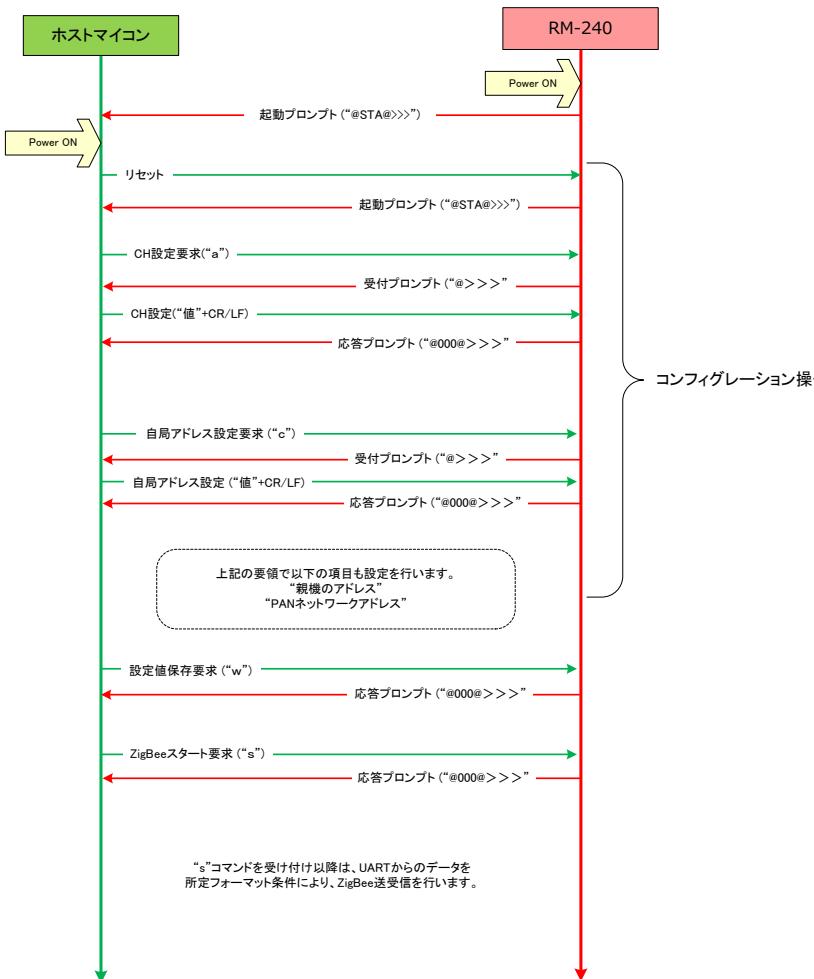
# 4.コンフィグレーション項目一覧

## 4.2 コンフィグレーション "m" ~ "z" まで

コマンド	設定項目	出荷時設定	設定範囲	説明
"u"	電波送信レベルの設定	0	0~46(+3dbm~-43dbm)	
"v"	RFブーストモードの有効/無効	1	0:uコマンドで設定した値+0dbm 1:uコマンドで設定した値+4dbm	Uとvの組み合わせにより、最大+7dBm
"w"	FlashROMへの設定項目の書き込み	-	-	
"x"	データ転送モード	0	0:垂れ流し 1:フラグ+垂れ流し 2:フレーム	データ転送モード
"y"	送信元アドレスのUART出力	1	0:する 1:しない	
"z"	予備		-	
"?"	設定内容の確認			

# 4.コンフィグレーション項目一覧

## 4.3 コンフィグレーションの使用例



左図の例では、ホストマイコンとRM-240間のコンフィグレーション設定プロトコル例を示しています。

左図ではRM-240が先に起動した場合を想定しており、RM-240からホストマイコン向けに、「起動プロンプト(@STA@>>>)」を送信していますが、ホストマイコンがまだ起動していないため、無視されています。

ホストマイコンが起動後、RM-240をリセットする事で、RM-240から再度「起動プロンプト」が送信されてきますので、ここからコンフィグレーション操作を開始します。

別章に記載していますが、RM-240内部のテーブルは、デフォルト値が設定されておりコンフィグレーションでは、デフォルト値に対して変更する部分のみを変更します。

左図の例では、CH番号(無線CH)、自局アドレス、親局アドレス、PANネットワークアドレスの変更を行っています。

コンフィグレーションの最後に、「w」コマンドを発行して、設定した内容をRM-240内蔵のFlashROMに保存します。

コンフィグレーションする必要が無ければ、起動プロンプト後に、「s」コマンドを送信する事で、RM-240は直ぐにスタートします。

また、リセット後の「起動プロンプト」受信後、「r」コマンドを送信すると、前回設定したコンフィグレーション値を復元する事が出来ますので、

ホストマイコンから

「リセット」発行  
"r" コマンド送信  
"s" コマンド送信

の手順で、前回の設定値で直ぐにスタートする事が可能です。

# 5. シンプルモードと対話モード

SimpleMACstdでは、コンフィグレーションのやり取りをUART経由で行います。  
その際、PCなどに接続して人によるオペレーションで設定する場合と、マイコンなどにより半自動的にコンフィグレーションする場合とで、やり取りするメッセージの出し方を分けています。  
RM-240に電源投入後、又はリセット後の一一番最初に、コマンド応答方法について、聞いて来ます。

Command I/F MODE [0:Message Mode 1:Simple Mode] ?=

## ◆Message Mode(メッセージモード)

RM-240からのコマンド、及び応答メッセージは、人によるオペレーションを想定したメッセージが出力されます。

## ◆Simple Mode (シンプルモード)

RM-240からのコマンド、及び応答メッセージは、独自パターンのメッセージで統一しています。  
短いキャラクタコードのやり取りで、コンフィグレーション設定が可能です。

IEEE802.15.4では、16個の周波数チャンネルを使用する事が出来ます。チャンネル番号は、CH11～CH26 の中からコンフィグレーションで指定します。  
※デフォルトは、CH11 に設定しています。

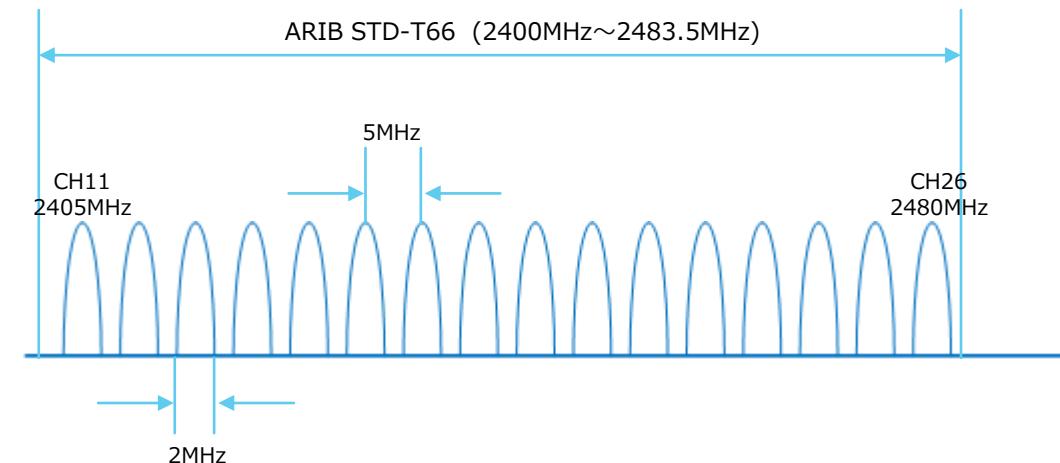
## 6.コンフィグレーション項目・詳細

### 6.1 使用CH番号の設定コマンド (“a”コマンド)

コマンド	説明	デフォルト値
“a” コマンド	自ノードが参加するPANアドレスを設定します。 PANアドレスとは、同一の無線CH内を更に、分けて管理したい時に使用します。 PANアドレスは、0x0001～0xFFFF の範囲で使用する事が出来ます。	0x1234

CH番号	中印周波数
CH11	2405MHz
CH12	2410MHz
CH13	2415MHz
CH14	2420MHz
CH15	2425MHz
CH16	2430MHz
CH17	2435MHz
CH18	2440MHz
CH19	2445MHz
CH20	2450MHz
CH21	2455MHz
CH22	2460MHz
CH23	2465MHz
CH24	2470MHz
CH25	2475MHz
CH26	2480MHz

#### ◆IEEE802.15.4の2.4GHz帯のチャンネルについて



チャンネル設定を行う場合、IEEE802.15.4のチャンネル割り当て内での設定では、隣接するチャンネルと重ならないため、同時に複数チャンネル設定をしても、理論上では干渉は発生しません。

ただし、同じ2.4GHz帯に共存する無線LAN等との競合については、考慮してチャンネル設定を行って下さい。

無線LANでは、13チャンネル分の設定が可能ですが、無線LANの中では隣接チャンネルと重なり合っているため、同時に使用可能な無線LANチャンネルは3つになります。

周辺で使用している無線LANチャンネル番号に対して、出来るだけ離れたチャンネル設定をして下さい。

# 6.コンフィグレーション項目・詳細

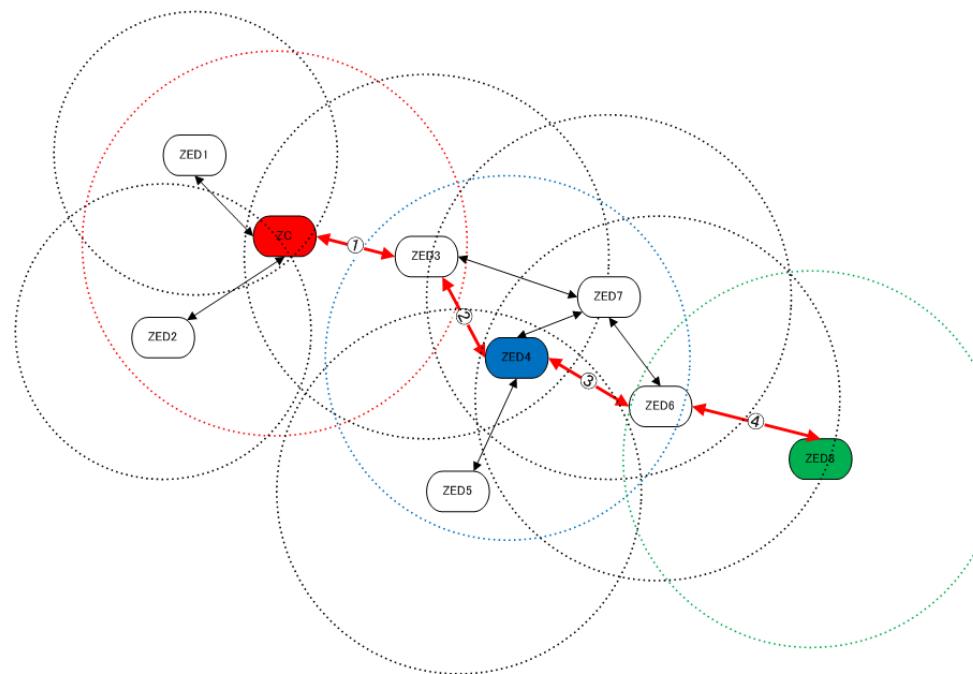
## 6.2 ホップカウンタの設定コマンド（'f' コマンド）

中継通信モード指定(fコマンド)で、ホップ通信を指定した上で、ホップカウンタの設定を行うと、指定回数分、周辺子機を利用してホップ通信を行う事が出来ます。これは、子機がそれぞれ移動可能な状態の時に有効です。

注意点は、実際にデータが目的ノードに届いても周辺に拡散したデータは指定ホップ回数まで子機間を回りますので、通常の通信への電波的な衝突率が高くなり、リトライの発生要因になります。本モードを使用する場合は、1ホップで電波が届く範囲に、子機が比較的少なく、かつ遠方にデータを届けたい場合に使用して下さい。

下記の図では、ZCからZED8にデータを送信した場合のイメージになります。

有効なデータは、①→②→③→④のルートで3ホップで到達しますが、それ以外にも黒矢印線の通信が発生します。



# 6.コンフィグレーション項目・詳細

## 6.3 PAN-ID、ネットワークアドレスの設定コマンド（'b' 'c' 'e' コマンド）

コマンド	説明	デフォルト値
"b" コマンド	<u>自ノードが参加するPANアドレスを設定します。</u> PANアドレスとは、同一の無線CH内を更に、分けて管理したい時に使用します。 PANアドレスは、0x0001～0xFFFF の範囲で使用する事が出来ます。	0x1234
"c" コマンド	<u>自ノードのネットワークアドレスを設定します。</u> 1つのPANアドレス内に、0x0000～0xFFFF の範囲で設定が可能です。	0x0001
"e" コマンド	<u>送信先ノードのアドレスを設定します。</u> (一般的には親機のアドレスを設定しますが、他の子機アドレスを指定も可能です。) 1つのPANアドレス内に、0x0000～0xFFFF の範囲で設定が可能です。	0x00FF

SimpleMACstdでは、ネットワークアドレスの管理はしていないため、同一アドレスの子機が複数いても通信が出来てしましますのでお客様によるアドレス管理が必要になります。

# 6.コンフィグレーション項目・詳細

## 6.4 中継通信の使用/不使用の設定コマンド（'f' コマンド）

コマンド	説明	デフォルト値
"f" コマンド	<p>中継通信を行う場合の、通信方式の選択をします。</p> <p>0 : 中継通信不使用 1ホップで対向機との通信を行います。</p> <p>1 : マルチホップ通信 他の子機を利用して、指定回数分ホップを繰り返します。</p> <p>2 : 経路指定通信 経由する子機アドレスを指定して通信します。</p>	0 : 中継通信不使用

マルチホップ通信の場合、ホップ数を多くする事で通信相手の位置が特定できない場合、障害物が多い場合などに効果が期待できますが電波圏内に他のノードが多い場合は、無駄なトラフィックがかかり、電波障害の原因になります。

経路指定通信は、中継するノード位置が固定化されている場合に、有効です。  
無駄なトラフィックも無く、最低限の通信数でデータを送信する事が可能です。

# 6.コンフィグレーション項目・詳細

## 6.5 タイマーハンドラ機能 使用/不使用の設定コマンド（'k' コマンド）

コマンド	説明	デフォルト値
"k" コマンド	タイマーハンドラ機能の使用/不使用を設定します 0 : タイマーハンドラ不使用 タイマー割り込みハンドラ機能を起動しません 1 : タイマーハンドラ使用 タイマー割り込みを利用して周期的にユーザー関数を起動したい場合に設定します。	0 : 不使用

ハンドラの起動周期は、デフォルトで10秒に設定しています。

設定値を変更する場合は、ソフトウェアの設定テーブルの変更と、再ビルドが必要です。（開発環境が必要になります）

# 6.コンフィグレーション項目・詳細

## 6.6 スリープ機能の設定コマンド ('l(エル)’ ‘m’ ‘n’ コマンド)

コマンド	説明	デフォルト値
“l” コマンド	<p><u>スリープ起床条件の設定</u></p> <p>0 : スリープ不使用      スリープモードを使用しません</p> <p>1 : タイマー起床      kコマンドで設定した時間後に起床します</p> <p>2 : GPIO起床      GPIO入力条件で起床します ※サンプルプログラムの設定例   スリープ条件    GPIO21 (PortC(5))を"Low"(RM-240EVの J4の2番PINをショート)   起床条件      RM-240EV の SW2を押下 (GPIO07 / PoartA(7) Low)</p>	0:スリープ不使用
“m” コマンド	<p><u>スリープタイマー起床時の時間設定</u></p> <p>0~65530 の範囲で設定    1カウント/250ms    (例) 5秒設定の場合は、"20"を設定</p>	0
“n” コマンド	<p><u>スリープ機能の使用の有無</u></p> <p>0 : スリープ不使用</p> <p>1 : スリープ使用</p>	0:スリープ不使用

# 6.コンフィグレーション項目・詳細

## 6.7 データ転送機能設定コマンド（'x' コマンド）

コマンド	説明	デフォルト値
“x” コマンド	<p>データの送信モードを設定します。</p> <p>0 : 垂れ流し送信モード      UART入力されたデータを無条件に送信します。</p> <p>1 : フラグ付き垂れ流しモード      UART入力されたデータのヘッダーに、各種フラグが付加されて出力されます。</p> <p>2 : フレームモード      UARTから所定のデータフォーマットを受信する事により送信します。</p>	0 : 垂れ流しモード

垂れ流しモードは、データがUARTから入力され次第、直ちに送信されます。ZigBeeパケットはタイミングによりランダムに分割されます。  
フラグ付き垂れ流しモードに付加されるフラグは、電波強度、送信元アドレス、PAN情報などがヘッダーに付加されます。  
フレームモードは、所定のフォーマットのデータがUART入力された時点で無線送信され、データはZigBeeの1つのパケットで送信されます。

主に、1:1で高速通信を行いたい場合は、垂れ流しモードが適しています。

# 6.コンフィグレーション項目・詳細

## 6.8 電波受信強度データ(RSSI値)付加コマンド（'p' コマンド）

コマンド	説明	デフォルト値
"p" コマンド	<p>電波の受信強度(RSSI値)をデータのヘッダーに付加して出力します。</p> <p>0 : RSSI値なし フラグ付き垂れ流しモード、フレームモード時に付加されるヘッダーに、RSSI値を出力しません。</p> <p>1 : RSSI値付加(HEX) フラグ付き垂れ流しモード、フレームモード時に付加されるヘッダーに、RSSI値をHEX出力します。</p> <p>2 : RSSI値付加(ASCII) フラグ付き垂れ流しモード、フレームモード時に付加されるヘッダーに、RSSI値をASCII出力します。</p>	0 : RSSI値なし

RSSI値は、0x07(+7dbm)～0x9E(-98dbm)の範囲で出力されます。

# 6.コンフィグレーション項目・詳細

## 6.9 PAN-ID機能の有効/無効コマンド（'q' コマンド）

コマンド	説明	デフォルト値
"q" コマンド	PANアドレスによるフィルタリング機能の有効/無効の設定を行います。  0 : フィルタ無効 PANアドレスによるフィルタを無効にします。 1 : フィルタ有効 PANアドレスによるフィルタを有効にします。	1 : フィルタ有効

PANアドレスのフィルタを無効にすると、同一CHの全てのノードへのデータの送信が可能になります。

PANが有効設定されているグループに、PAN無効設定した機器を持ち込んでデータを送信すると、送信先アドレスで指定したノードにデータを送信する事が出来ます。

※受信については、本コマンドではフィルタ解除ができません。

PANの設定にかかわらず、全てのデータをモニタする場合は、 "@"コマンドによるモニターモードで確認します。

# 6.コンフィグレーション項目・詳細

## 6.10 MAC層リトライ回数設定コマンド ('t' コマンド)

コマンド	説明	デフォルト値
"t" コマンド	リトライ回数の設定 1~255 : MAC層のリトライ回数を設定します。	1

# 6.コンフィグレーション項目・詳細

## 6.11 タイマー同期コマンド（‘将来実装’コマンド）

コマンド	説明	デフォルト値
将来実装	親機のタイマー軸に、ノード側のタイマー同期を合わせる機能です。 0 : 同期無効 タイマー同期を行いません。 1 : 同期有効 タイマー同期を行います。	0 : 同期無効

親子間のタイマー同期を合わせる為の、同期コマンドを発行するかしないかを設定する機能です。  
この機能により、同一CH,PAN内において、全ての子機は親機のタイマー軸に同期させる事が出来ます。

同期有効に設定した場合、システム立ち上げ時は同期完了するまで、外部からのデータ受け付けは行いません。  
また、同期する周期は1時間に1回発行されます。

# 6.コンフィグレーション項目・詳細

## 6.12 電波送信出力制御コマンド（'u','v' コマンド）

コマンド	説明	デフォルト値
"u" コマンド	<p>電波の送信強度の設定を行います。</p> <p>設定範囲は、+3dbm ~ -43dbm の範囲を、0~46の値で設定します。</p> <p>0 : +3dbm 1 : +2dbm 2 : +1dbm 3 : ±0dbm 4 : -1dbm 5 : -2dbm ↓ 46: -43dbm</p>	+3dbm
"v" コマンド	<p>パワーモードの有効/無効の設定を行います。</p> <p>0 : パワー送信モード無効 uコマンドで設定した値のまま送信されます。</p> <p>1 : パワー送信モード有効 uコマンドで設定した値に対して、+4dbmが加算されます。</p>	有効

# 6.コンフィグレーション項目・詳細

## 6.13 FlashROM書き込み/読み出しコマンド（'w' 'r' コマンド）

コマンド	説明	デフォルト値
"w" コマンド	<p><u>コンフィグレーション内容をFlashROMに保存します</u></p> <p>出荷時のコンフィグレーション値は、デフォルト値が設定されています。 お客様で変更された内容は、"w"コマンドによりRM-240内蔵のFlashROMに保存する事でリセット後に、"r"コマンドで読み出す事で前回設定値を復元する事が出来ます。</p>	-
"r" コマンド	<p><u>前回コンフィグレーションした内容をFlashROMから読み出します</u></p> <p>"w"コマンドで保存された内容をRM-240内蔵のFlashROMから読み出して、再設定します。</p>	-

# 6.コンフィグレーション項目・詳細

## 6.14 システム開始コマンド（'s' コマンド）

コマンド	説明	デフォルト値
“s” コマンド	コンフィグレーションした内容に基づいて、システムをスタートさせます。	-

# 6.コンフィグレーション項目・詳細

## 6.15 モニターモードコマンド（‘@’ コマンド）※将来実装機能

コマンド	説明	デフォルト値
“@” コマンド	同一CH内のすべてのデータをPAN設定に関係なく、受信します。 受信したデータはUARTにそのまま出力します。	0

# 6.コンフィグレーション項目・詳細

## 6.16 RSSI値受信閾値設定コマンド (“g” コマンド)

コマンド	説明	デフォルト値
“g” コマンド	<p>無線データ受信時のRSSI値(受信電力値)の受信閾値の設定を行います。 ここで設定した値よりも受信電力が弱い場合は、受信データは破棄されます。</p> <p>0 : 閾値無効 すべてのデータを受信します。 1~98 : 左記の設定値をマイナス値として処理します。 ※1の場合は、-1dBm と処理されます。 50(-50dBm)と設定した場合は、受信データの電力値が-50dBmよりも弱い場合は破棄します。</p>	0

# 6.コンフィグレーション項目・詳細

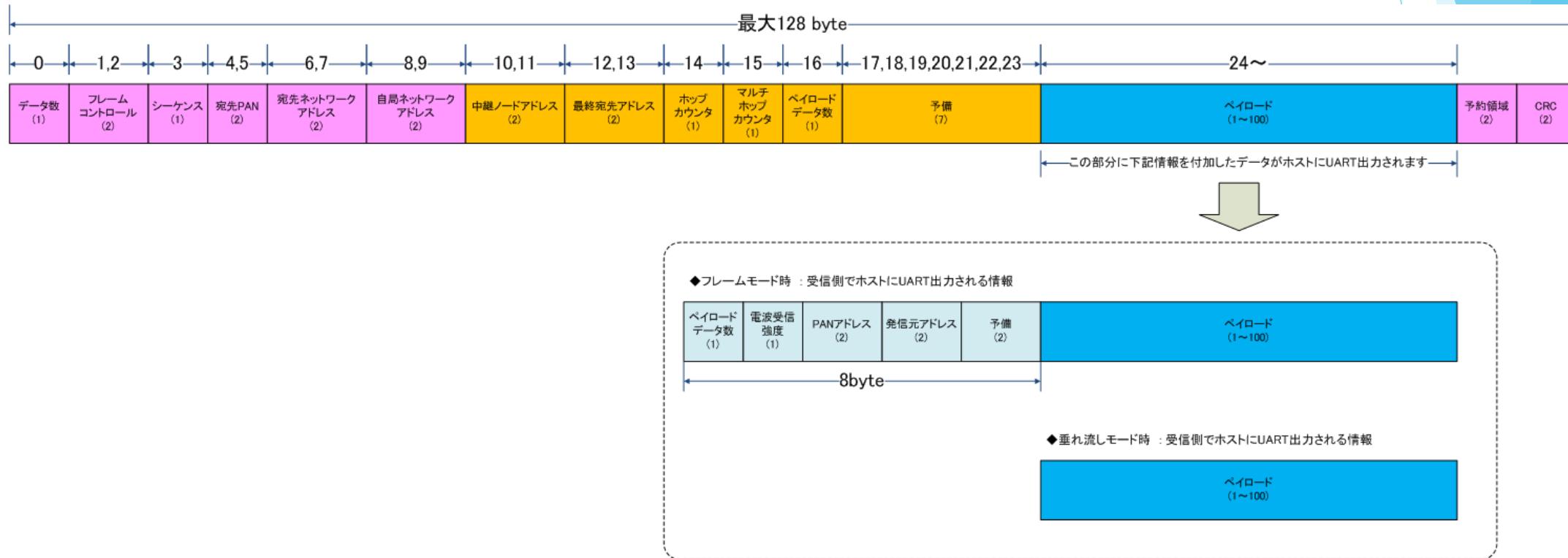
## 6.17 改行コード出力設定コマンド ('h' コマンド)

コマンド	説明	デフォルト値
"h" コマンド	無線受信したデータをUART出力する時に、改行コード(CR+LF)を付加するか否かの設定を行います。 0 : 出力なし 1 : 出力します	0

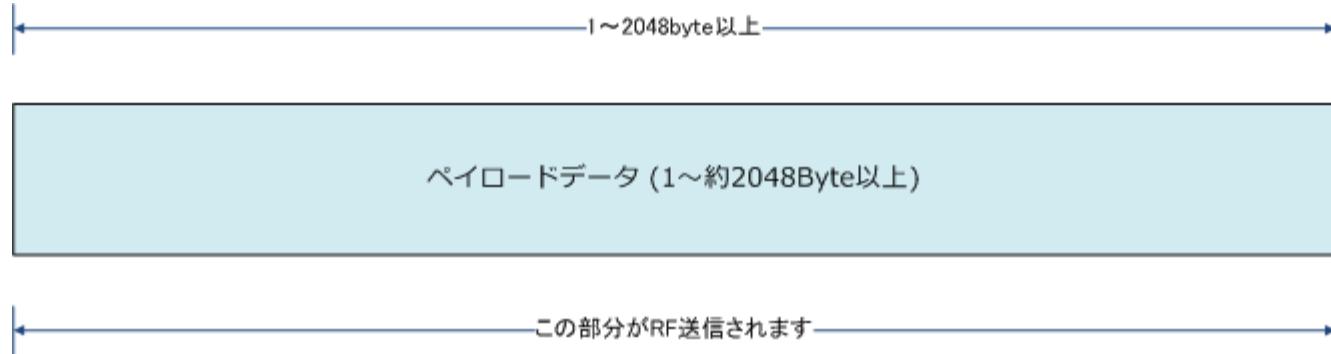
## 7.応答コード一覧

No	応答内容	対話モード	シンプルモード
01	起動通知	なし	"@STA"
02	プロンプト	Please input!! >	"@>>>"
03	正常応答	Configration Succeeded!!	"@000"
04	設定範囲エラー	Configration Error!! [xxxxxxxxxx]	"@001"
05	未定義エラー	-	"@002"
06	ACK未応答	@003	"@003"
07	FlashROM消去エラー	E_FERER	"@004"
08	FlashROM読み込みエラー	E_FRDER	"@005"
09	FlashROM書き込みエラー	E_FWTER	"@006"
10	UART受信FIFOフル	E_UARTF	"@007"
11	予備		"@008"
12	予備		"@009"
13	予備		"@00A"
14	予備		"@00B"
15	予備		"@00C"
16	予備		"@00D"
17	システム異常	E_SYSHLT	"@00E"

## 8. フレームフォーマット(全モード共通)

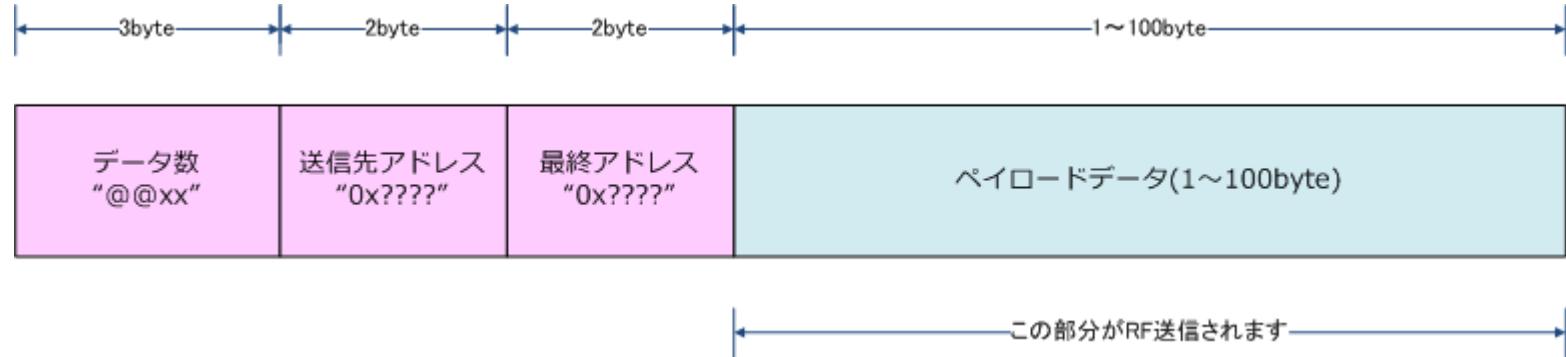


## 9.UARTフォーマット(垂れ流しモード/フラグ付き垂れ流しモード)



No	フィールド項目	説明	備考
01	ペイロードデータ	UARTから受信したままのデータを送信します。	RM-240でUART割り込みで取り込んだタイミングで順次RF送信しますので、ペイロードサイズは可変長になります。 最大送信数は定義していませんが、RM-240内部の一次記憶用バッファサイズとして、2048byte確保しています。

# 10.UARTフォーマット(フレームモード)

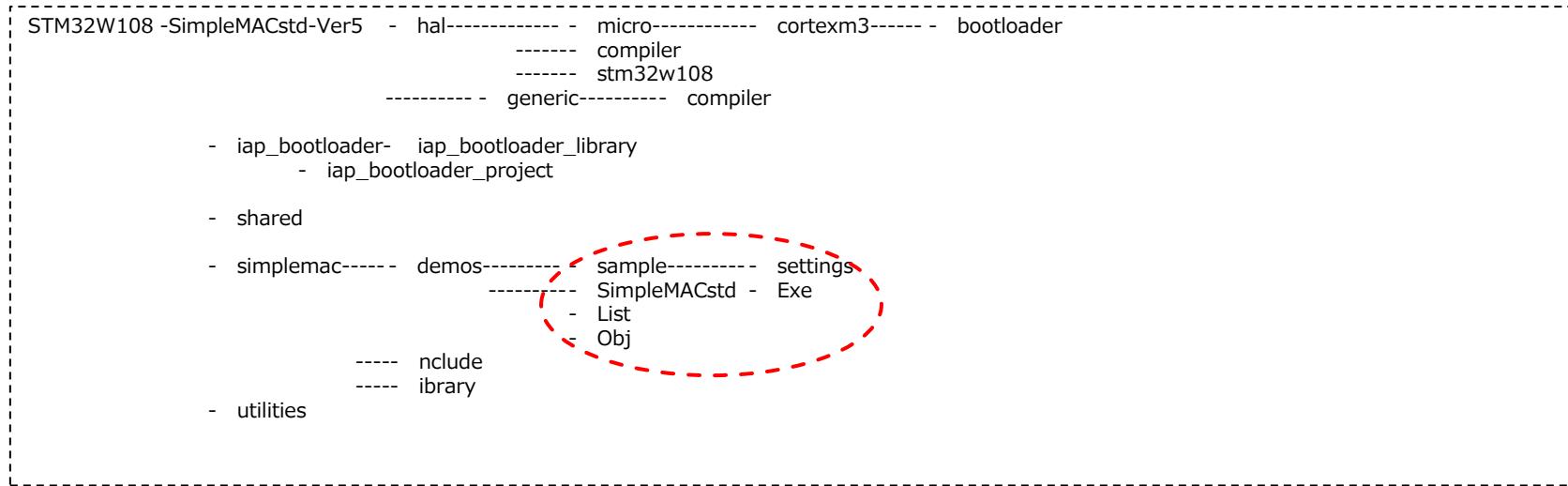


No	フィールド項目	説明	備考
01	データ数	ペイロードのデータ数 (MAX 100)	
02	送信先アドレス	送信先のネットワークアドレス ※コンフィグレーション設定値より優先します	指定なしの時は、0x0000 (コンフィグレーション値を使用する場合)
03	最終アドレス	送信先アドレスを経由して最終的に通信したい機器のアドレス	1ホップ(送信先アドレスまで)で良い場合は 最終アドレスを 0xFFFFにして下さい。 0xFFFF以外の数値が入っていると、そのア ドレス向けに、送信先アドレスからホップ通 信が実行されます。
04	ペイロードデータ	ペイロードのデータ数をバイナリで指定します (例)90byteのデータを送信する場合 "0x40,0x40,0x5a" となります。(0x40は、"@")	

# 11.サンプルプログラムの基本構成

## 11.1 SimpleMACstd のファイル階層図

### ◆SimpleMACstd-フォルダ階層図



◆上図の赤破線表示の箇所に、プロジェクトファイル、及びSimpleMACstdのmainプログラムがあります。

- ①IAR社EWARM用プロジェクトファイル ..... sample.eww
- ②SimpleMACstdのmainプログラムファイル ..... simpleMACstd.c
- ③ビルト実行後生成される実行モジュール ..... /exe/SimpleMACstd.s37
- ④mapファイル ..... /List/SimpleMACstd.map

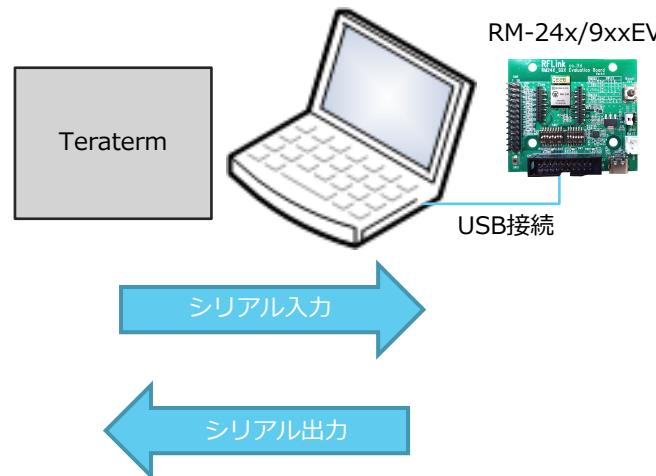
# 11.サンプルプログラムの基本構成

## 11.2 コンフィグレーション設定例 ~ 単純なシリアルデータの送受信

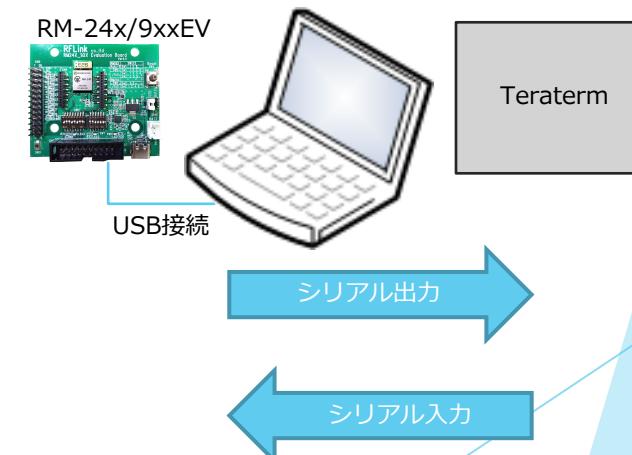
1:1通信で、PC(又はマイコン)～PC(又はマイコン) 間で単純な通信を行う場合の設定例

2台のPC間で、それぞれのキーボードから入力した値を、リアルタイムに相互通信します。  
下記コンフィグレーション値以外はデフォルト設定を使用します。(CH11/垂れ流しモード)

◆コンフィグレーション(例)  
"c" : 自ノードアドレス 0x01  
"e" : 相手先ノードアドレス 0x09



◆コンフィグレーション(例)  
"c" : 自ノードアドレス 0x09  
"e" : 相手先ノードアドレス 0x01



# 11.サンプルプログラムの基本構成

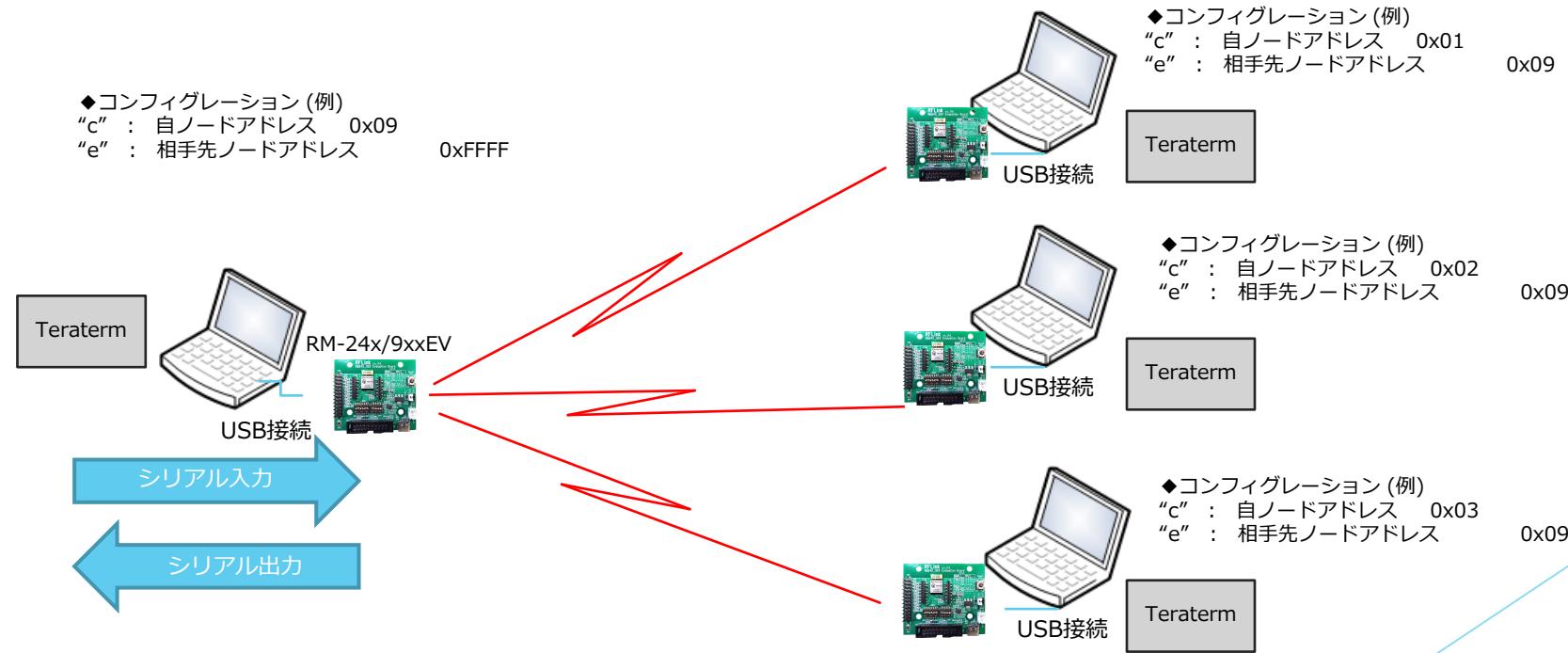
## 11.3 コンフィグレーション設定例 ~ 単純なシリアルデータの 1:Nによる構築

1:N通信で、PC(又はマイコン)～PC(又はマイコン) 間で単純な通信を行う場合の設定例

親機 1：子機3 の構成で、それぞれのキーボードから入力した値を、リアルタイムに相互通信します。

親機から子機に対しては、ブロードキャストアドレスで送信します。

下記コンフィグレーション値以外はデフォルト設定を使用します。(CH11/垂れ流しモード)



# 11.サンプルプログラムの基本構成

## 11.4 コンフィグレーション設定例 ～ 1:Nによる子機指定通信

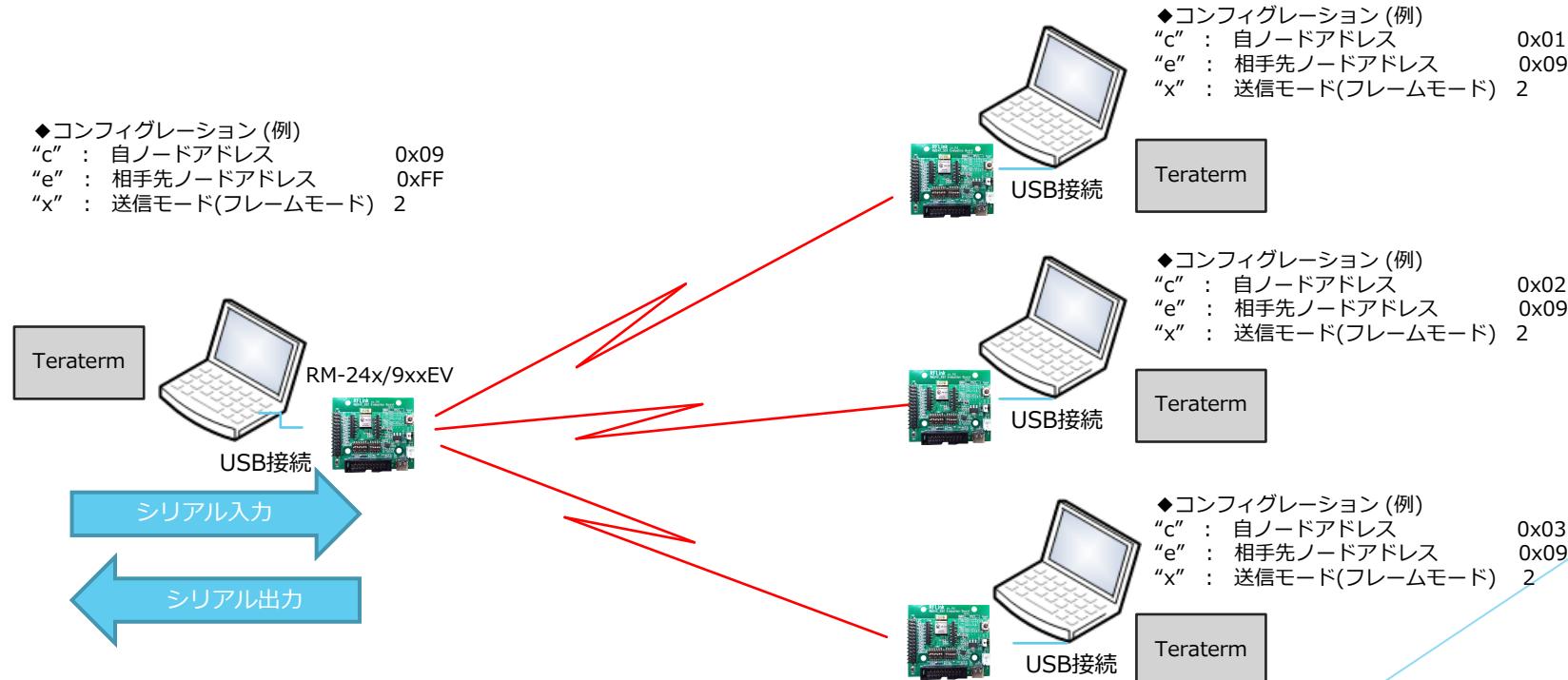
### 1:N通信で、PC(又はマイコン)～PC(又はマイコン) 間で指定フォーマットによる通信を行う場合の設定例

親機 1：子機3 の構成で、所定フォーマットのデータをリアルタイムに相互通信します。

親機から子機に対しては、子機アドレスを指定して送信します。

下記コンフィグレーション値以外はデフォルト設定を使用します。(CH11)

指定フォーマットについては、「12章 UARTフォーマット」を参照下さい



# 11.サンプルプログラムの基本構成

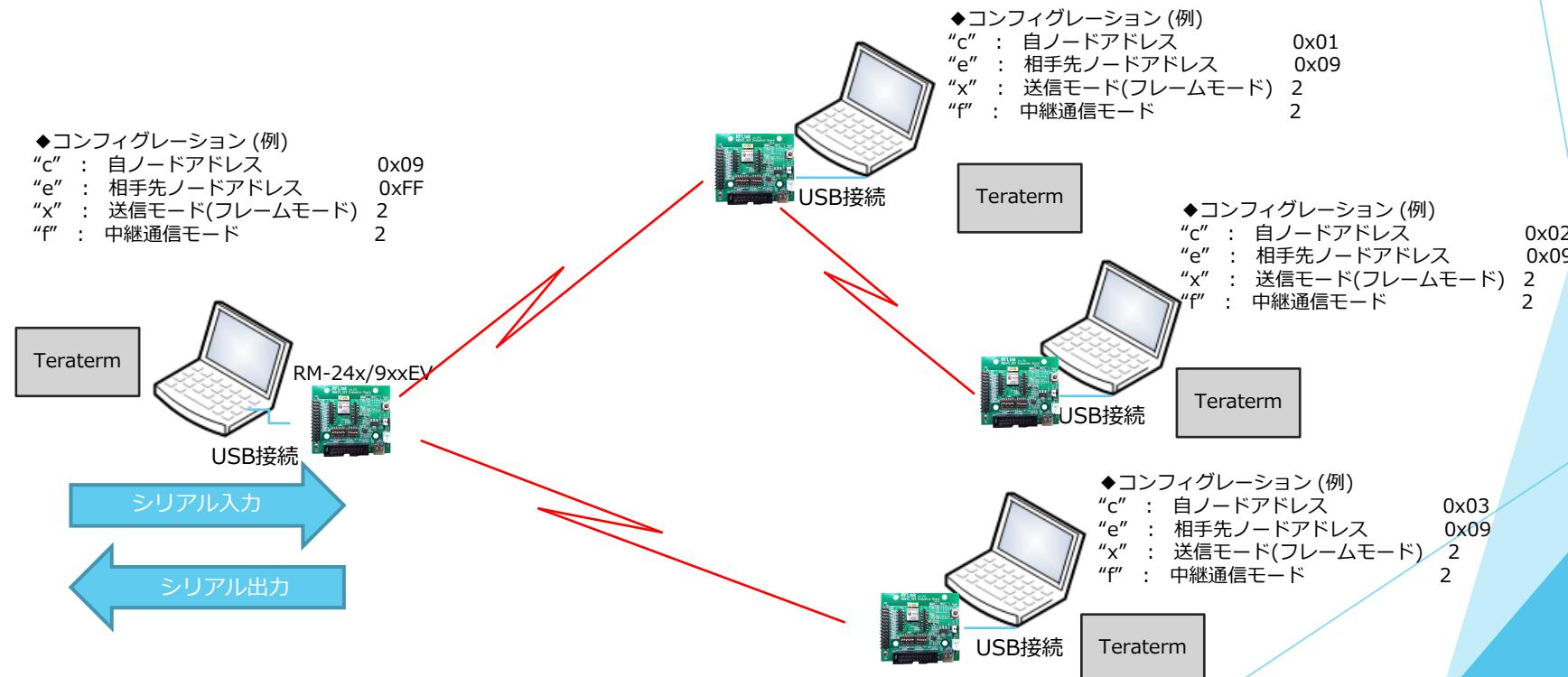
## 11.5 コンフィグレーション設定例 ~ 経路指定通信を含む 1:Nによる子機指定通信

1:N通信で、PC(又はマイコン)～PC(又はマイコン)間で、経路指定を通信を行う場合の設定例

基本構成は、前項の11.4と同じです。

“f”コマンドで、経路通信モードを指定します。12章の指定フォーマットに従い、経由する子機のアドレスを指定してUARTでRM-240に送信します。

“f”コマンドで、経路通信モードを指定していても、指定フォーマットのアドレス指定を0xFFFFにする事で、動的に経路通信なしで送信する事も可能です。



# 11.サンプルプログラムの基本構成

## 11.6 コンフィグレーション設定例 ~ スリープ機能を有効にした送信設定

### 1:1通信、又は1:N通信で、子機がデータ送信後にスリープし、10秒後に起床してA/Dデータを送信する設定例

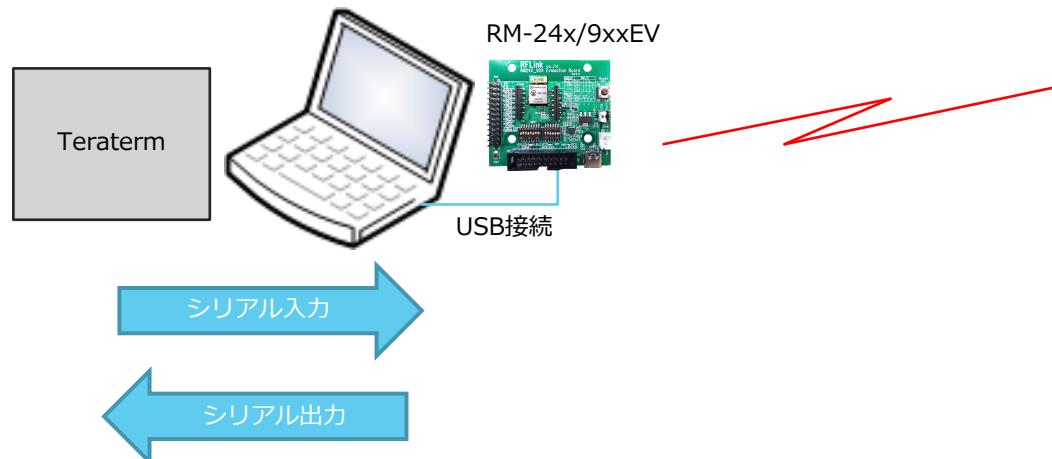
子機はRM-240EV単体で動作、親機はPCとして接続します。(下図は1:1ですが、1:Nでも可能です)

スリープ時間は、10秒の設定例です。

親機はスリープしません。

子機は、A/D値を取りこんで、親機に送信後MAC層のACK受信で10秒間スリープします。

◆コンフィグレーション(例)  
"c" : 自ノードアドレス 0x01  
"e" : 相手先ノードアドレス 0x09



◆コンフィグレーション(例)  
"c" : 自ノードアドレス 0x09  
"e" : 相手先ノードアドレス 0x01  
"l" : スリープモード使用有無 1  
"j" : スリープ起床条件設定 2  
"k" : スリープ時間設定 40



# 11.サンプルプログラムの基本構成

## 11.7 コンフィグレーションした設定値で自動スタートをさせる方法

RFLINKの開発ボード(Ver4.0以上)を使用する場合、DipSw設定により、FlashROMから事前設定したネットワーク情報を読み出して自動スタートさせる事が出来ます。  
(図の黄色破線部)

※DipSWの具体的な説明は、開発環境構築ガイドを参照下さい。

- 【手順1】 自動起動設定をOFF
- 【手順2】 PCと開発ボードをUSB接続して teraterm等のシリアルソフトを準備
- 【手順3】 RM-240を起動して、teraterm画面に設定メニューを表示させる
- 【手順4】 任意の設定を行い、wコマンドにより設定情報をFlashROMに保存
- 【手順5】 電源をOFF
- 【手順6】 自動起動設定をON
- 【手順7】 電源投入

これで、手順4で設定した内容で自動スタートできます。



# 12.自動ルーティング(オプション)

## 12.1 自動ルーティングの概要

SimpleMACStdでは、オプション機能として「自動ルーティング」をご用意しています。

以下は、ZigBeePROのメッシュ通信、自動ルーティング、クラスタツリー、との違いをした表です。

No	項目	メッシュプロトコル (ZigBeePRO)	自動ルーティング (SimpleMACStd)	クラスタツリー (ZigBee/SimpleMACStd)
1	リアルタイム性(ルーティング能力)	○	○	○
2	信頼性(対障害物)	○	○	×
3	省電力性	△	○	○
4	大規模ネットワーク(300ノード以上)	○	△	○
5	アプリケーションの開発性の自由度	△	○	○
6	コスト性	△	○	○

### ◆自動ルーティングの利点

自動ルーティングは、メッシュプロトコルの様に、常に周辺ノードとの経路検索や、新ノードサーチを行わないため、ネットワークのトラフィックが非常に軽いスタックです。

このため、他ノードからのデータを中継する場合のエア上の衝突率が低く、かつ内部のルーティング時間も2ms以下と非常に短時間でルーティングを行う事が出来ます。

また、メッシュプロトコルが、約100KB前後の内部メモリを必要とするのに対して、SimpleMACStd+自動ルーティングの場合、約半分の容量のため低成本のチップを採用しています。

更に、自動ルーティングでは、専用のルーターが必要なく、全子機がエンドデバイス(ZED)の機能を持ちつつ、中継機能も持ち合わせています。

メッシュプロトコルの場合、停電後のネットワーク復旧に時間を要するのに対して(ノード数が30ノード以上の場合、3~5分以上かかる場合もある)、自動ルーティングの場合は、瞬時に復旧が可能です。

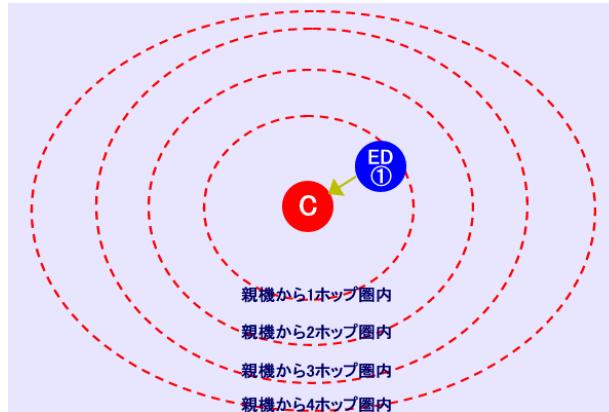
### ◆自動ルーティングのデメリット

ホッピング数を4段までに制限しているため、4段を超えた場所にあるZC(親機)まで送信する事が出来ません。  
このため、1:Nの大規模ネットワークには適していません。

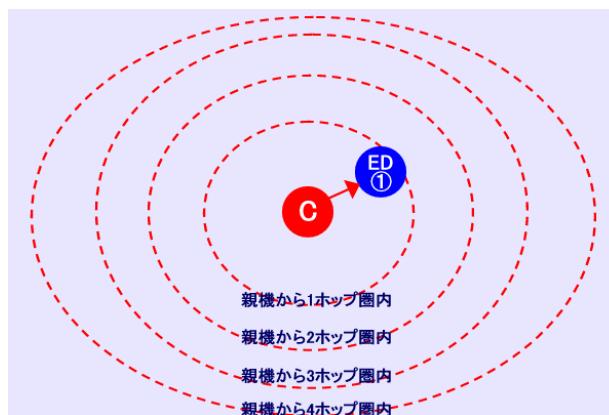
# 12.自動ルーティング(オプション)

## 12.2 自動ルーティングの仕組み

- ① 自動ルーティングは、各子機(ZED)が電源投入後に、親機(ZC)に対する通信経路の確立を試みます。



- ② 接続要求に対して、親機から接続完了応答があった場合は、ネットワーク参加成功です。



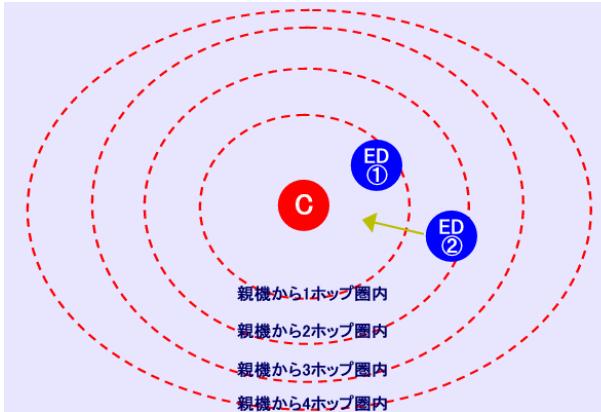
### 【記号】

- C** コーディネータ(親機)
- ED (N)** エンドデバイス(子機)
- ビーコン範囲**
- 親機ダイレクト接続要求**
- 接続完了応答**
- ビーコン応答**
- 通信データ**

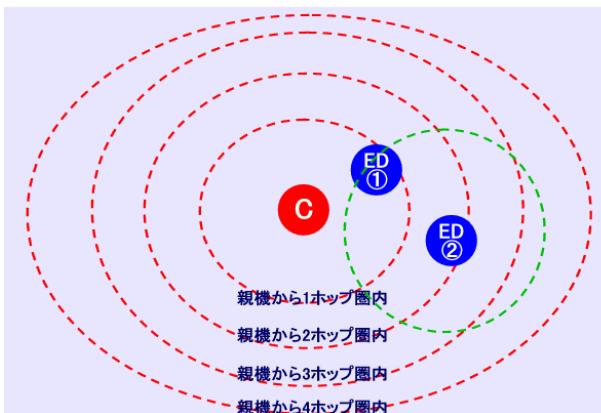
# 12.自動ルーティング(オプション)

## 12.2 自動ルーティングの仕組み

- ③ 親機に1ホップで電波の届かない位置にED②が立ち上がった場合、①と同様に最初に親機への接続を試みます。



- ④ 親機からの応答が来ないため、周辺検索を行いビーコンを送信します。



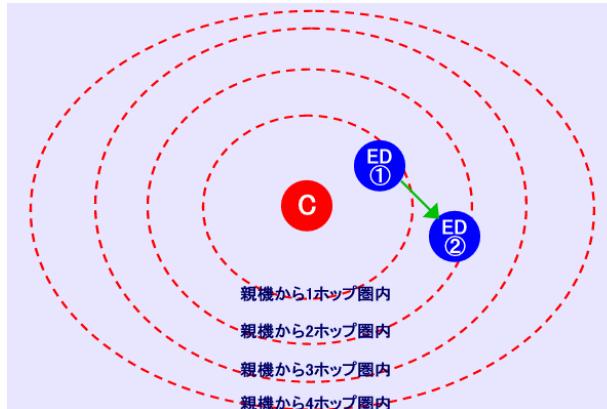
### 【記号】

- C コーディネータ(親機)
- ED(N) エンドデバイス(子機)
- ビーコン範囲
- 親機ダイレクト接続要求
- 接続完了応答
- ビーコン応答
- 通信データ

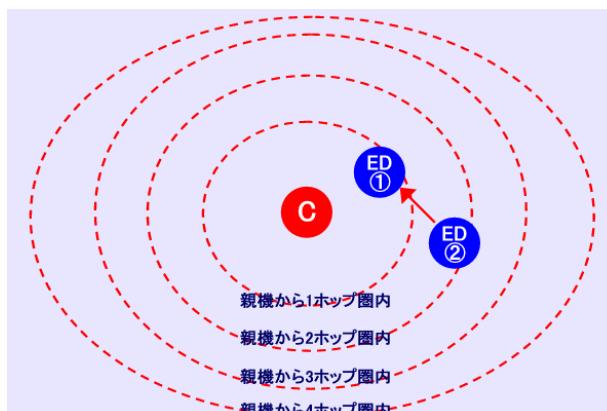
# 12.自動ルーティング(オプション)

## 12.2 自動ルーティングの仕組み

⑤ 周辺検索のビーコンに対して、親機への接続を完了しているED①がビーコン応答を返信します。



⑥ ED②は、ED①に対して接続完了応答を送信し、ED②のネットワーク参加が完了します。



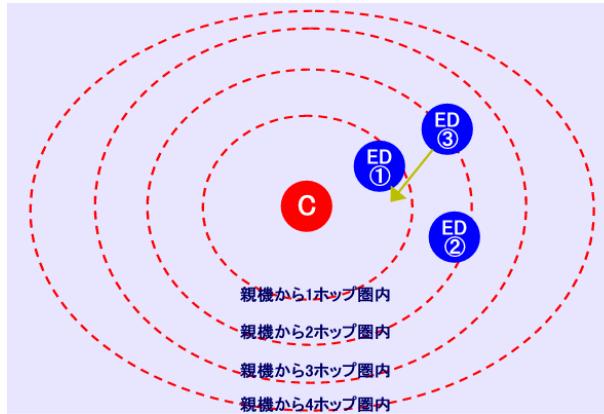
### 【記号】

- C コーディネータ(親機)
- ED(N) エンドデバイス(子機)
- ビーコン範囲
- 親機ダイレクト接続要求
- 接続完了応答
- ビーコン応答
- 通信データ

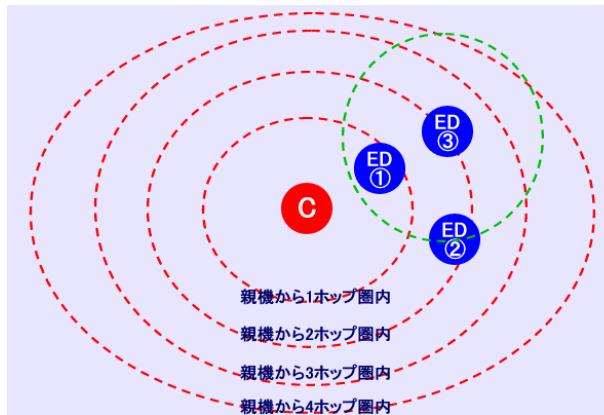
# 12.自動ルーティング(オプション)

## 12.2 自動ルーティングの仕組み

- ⑦ ED①、ED②がネットワークに参加している状態に、さらにED③が立ち上がり、親機への接続要求を試みます。



- ⑧ ED③が親機への直接接続できなかったことにより、周辺検索ビーコンを送信します。



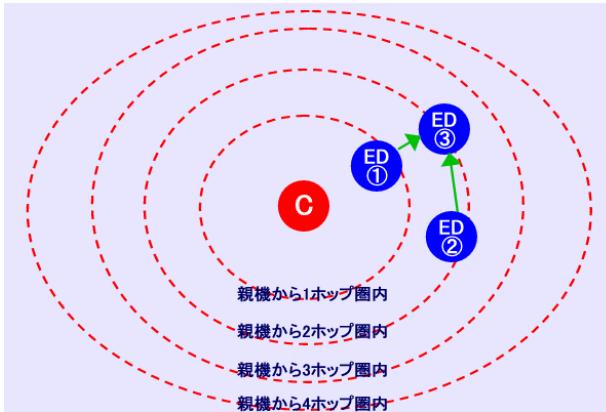
### 【記号】

- C** コーディネータ(親機)
- ED(N)** エンドデバイス(子機)
- ビーコン範囲** Beacon Range
- 親機ダイレクト接続要求** Direct Connection Request to Coordinator
- 接続完了応答** Connection Completion Response
- ビーコン応答** Beacon Response
- 通信データ** Communication Data

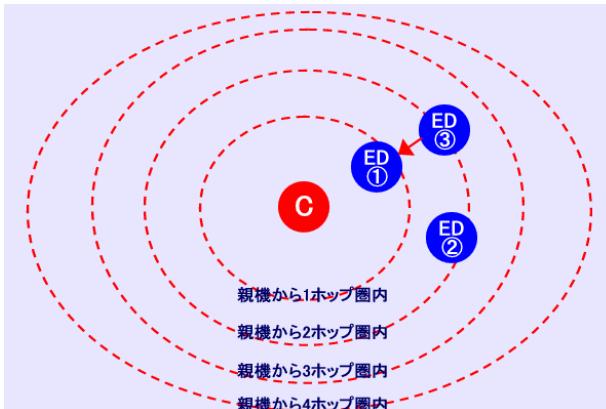
# 12.自動ルーティング(オプション)

## 12.2 自動ルーティングの仕組み

- ⑨ ED③のビーコンに対して、ED①とED②がビーコン応答を送信します。



- ⑩ ED③は、ED①とED②からの応答情報から親機への接続段数、自ノードとの電波強度の値に関する情報を取得し、その情報に基づいて、ED①を選択します。



### 【記号】

- C コーディネータ(親機)
- ED(N) エンドデバイス(子機)
- ビーコン範囲
- ← 親機ダイレクト接続要求
- ← 接続完了応答
- ビーコン応答
- ← 通信データ

# SimpleMACstd 取り扱い説明書

## ◆Release version

Rev 1.0	2011-04-20
Rev 1.1	2011-05-30
Rev 1.2	2011-07-19
Rev 1.3	2011-08-11
Rev 2.0	2014-04-01
Rev 3.0	2016-08-20
Rev 3.3	2018-02-01
Rev 3.4	2019-03-10
Rev 3.5	2020-04-01