

SmartMesh IP Mote Serial API ガイド

目次

1	本書について	5
1.1	関連資料	5
1.2	表記規則	7
1.3	改訂履歴	8
2	はじめに	9
3	モートのシリアル・モード	10
4	プロトコル	11
4.1	データ表現	11
4.1.1	共通データ・タイプ	11
4.1.2	整数表現	12
4.1.3	伝送順序	13
4.2	パケット形式	13
4.2.1	HDLC パッケージのカプセル化	13
4.2.2	HDLC ペイロードの内容	16
4.3	モートとマイクロプロセッサ間の通信	18
4.3.1	アクノリッジ対象リンク	18
4.3.2	前方互換性を備えたクライアントのガイドライン	18
5	コマンド	19
5.1	blink(0x2E)	20
5.2	bindSocket(0x17)	22
5.3	clearNV(0x10)	23
5.4	closeSocket(0x16)	24
5.5	disconnect(0x07)	25
5.6	getParameter(0x02)	26
5.6.1	getParameter<applInfo>	26
5.6.2	getParameter<antGain>	27
5.6.3	getParameter<autoJoin>	28
5.6.4	getParameter<charge>	29
5.6.5	getParameter<entropy>	30
5.6.6	getParameter<euCompliantMode>	31
5.6.7	getParameter<eventMask>	32
5.6.8	getParameter<ipv6Address>	33
5.6.9	getParameter<joinDutyCycle>	34
5.6.10	getParameter<macAddress>	35
5.6.11	getParameter<shortAddress>	36
5.6.12	getParameter<moteInfo>	37
5.6.13	getParameter<moteStatus>	38
5.6.14	getParameter<netInfo>	39
5.6.15	getParameter<networkId>	40
5.6.16	getParameter<OTAPLockout>	41
5.6.17	getParameter<pwrSrcInfo>	42
5.6.18	getParameter<routingMode>	43

5.6.19	getParameter<testRadioRxStats>	44
5.6.20	getParameter<time>	45
5.6.21	getParameter<txPower>	46
5.7	getServiceInfo(0x12)	47
5.8	join(0x06)	48
5.9	lowPowerSleep(0x09)	49
5.10	openSocket(0x15)	50
5.11	requestService(0x11)	51
5.12	reset(0x08)	52
5.13	search(0x24)	53
5.14	sendTo(0x18)	54
5.15	setParameter(0x01)	56
5.15.1	setParameter<advKey>	56
5.15.2	setParameter<antGain>	57
5.15.3	setParameter<autoJoin>	58
5.15.4	setParameter<euCompliantMode>	59
5.15.5	setParameter<eventMask>	60
5.15.6	setParameter <joinDutyCycle>	61
5.15.7	setParameter <joinKey>	62
5.15.8	setParameter<macAddress>	63
5.15.9	setParameter<networkId>	64
5.15.10	setParameter<OTAPLockout>	65
5.15.11	setParameter<pwrSrcInfo>	66
5.15.12	setParameter<routingMode>	67
5.15.13	setParameter<txPower>	68
5.16	socketInfo(0x2B)	69
5.17	stopSearch(0x2F)	70
5.18	testRadioRx(0x0C)	71
5.19	testRadioTxExt(0x28)	72
5.20	zeroize(0x29)	74
6	通知	75
6.1	advReceived	75
6.2	events	76
6.3	receive	77
6.4	timeIndication	78
6.5	txDone	79
7	定義	80
7.1	コマンド	80
7.2	通知	81
7.3	パラメータ	82
7.4	レスポンス・コード	83
7.5	サービスのタイプ	83
7.6	サービスのステート	84
7.7	プロトコルのタイプ	84
7.8	パケット優先順位	84

7.9	モートのステート	84
7.10	イベント	85
7.11	アラーム	85
7.12	無線テスト・タイプ	86
7.13	パケットの伝送ステータス	86

1 本書について

1.1 関連資料

SmartMesh IP ネットワーク向けに以下の資料が提供されています。

Starter Kit のクイック・ガイド

- [SmartMesh IP Easy Start Guide](#) - 基本的なインストール方法とネットワークの動作確認テストについて説明しています。
- [SmartMesh IP Tools Guide](#) - インストールのセクションではシリアル・ドライバのインストール手順について説明しており、Easy Start Guide やその他のチュートリアルで使用されるサンプル・プログラムも含まれています。

ユーザ・ガイド

- [SmartMesh IP User's Guide](#) - ネットワーク概念についての説明と、モートおよびマネージャの API を使用して特定のタスク(データ送信や統計情報の収集など)を実行する方法について説明しています。この資料は、API ガイドを使用するための予備知識を提供します。

デバイスの対話操作インターフェース

- [SmartMesh IP Manager CLI Guide](#) - マネージャとやり取りするために使用します(クライアントの開発中やトラブルシューティングなど)。このガイドは、CLI の接続とそのコマンド・セットについて説明しています。
- [SmartMesh IP Manager API Guide](#) - プログラムを使用してマネージャとやり取りするために使用します。このガイドは、API の接続とそのコマンド・セットについて説明しています。
- [SmartMesh IP Mote CLI Guide](#) - モートとやり取りするために使用します(センサー・アプリケーションの開発中やトラブルシューティングなど)。このガイドは、CLI の接続とそのコマンド・セットについて説明しています。
- [SmartMesh IP Mote API Guide](#) - プログラムを使用してモートとやり取りするために使用します。このガイドは、API の接続とそのコマンド・セットについて説明しています。

ソフトウェア開発ツール

- [SmartMesh IP Tools Guide](#) - [SmartMesh SDK](#) に含まれる各種の評価および開発サポート・ツールについて説明しています。モートおよびマネージャ API の使用とネットワークの視覚化のためのツールを含みます。

アプリケーション・ノート

- [SmartMesh IP Application Notes](#) - SmartMesh IP ネットワーク固有の各種トピックと、SmartMesh ネットワーク全般に当てはまるトピックが含まれています。

新規設計の開始時に役立つ資料

- [LTC5800-IPM SoC](#) またはこれに基づく [modules](#) のデータシート。
- [LTC5800-IPR SoC](#) またはこれに基づく [embedded managers](#) のデータシート。
- モート／マネージャ SoC 用またはモジュール用の [Hardware Integration Guide](#) - 設計に SoC またはモジュールの統合を盛り込むためのベスト・プラクティスを提供しています。
- 組み込みマネージャ用の [Hardware Integration Guide](#) - 設計する際に組み込みマネージャを統合させるためのベスト・プラクティスを提供しています。
- [Board Specific Integration Guide](#)- SoC モートおよびマネージャ用。デフォルトの I/O 設定方法と、「ヒューズ表」を使用した水晶発振器のキャリブレーション情報について説明しています。
- [Hardware Integration Application Notes](#) - SoC 設計チェックリスト、アンテナ選定ガイドなどを含みます。
- [ESP Programmer Guide](#) - DC9010 Programmer Board と、デバイスへのファームウェアのロードに使用する ESP ソフトウェアのガイドです。
- ESP ソフトウェア - モートまたはモジュールにファームウェア・イメージをプログラミングするために使用します。
- Fuse Table ソフトウェア - [Board Specific Configuration Guide](#) で説明されているヒューズ表を作成するために使用します。

その他の役立つ資料

- SmartMesh 技術文書で使用されるワイヤレス・ネットワークの用語については、[SmartMesh IP User's Guide](#) を参照してください。
- [よくある質問の一覧](#)。

1.2 表記規則

本書では、以下の表記規則を使用します。

コンピュータ・タイプ(Computer type)は、URL の指定など、ユーザが入力する情報を示します。

太字は、ボタン、フィールド、メニュー・コマンド、デバイス・ステート、モードを示します。

斜体は、新しい用語や API とそのパラメータを示します。

✔ ヒントは、製品に関して役立つ情報を提供します。

ℹ 情報テキストは、背景や前後関係の理解に役立つ追加情報を提供します。

⚠ 注記は、概念についてより詳しい説明を提供します。

⊖ 警告！ 警告は、データ損失やハードウェアまたはユーザへの物理的な損害を引き起こす可能性のある動作をユーザに知らせます。

code blocks display examples of code

1.3 改訂履歴

リビジョン	日付	概要
1	2012/07/18	初期リリース
2	2012/08/10	radiotest API の更新
3	2013/03/18	多数の軽微な変更
4	2013/10/22	文書タイトルの変更
5	2014/04/04	無線テスト・コマンドの更新および明確化
6	2014/10/28	タイトルにコマンド ID を追加、拡張された testRadio コマンドにステーション ID を追加、通知構造の明確化
7	2015/04/22	setParameter コマンドに advKey を追加、その他の軽微な変更
8	2015/12/03	EN 300 328 準拠の設定の追加、appVer パラメータの追加、その他の軽微な変更
9	2016/11/07	getParameter<txPower>および setParameter<txPower>の説明の明確化、blink、getParameter<entropy>、および stopSearch コマンドの追加

2 はじめに

本書では、外部プロセッサが API シリアル・ポートを介した SmartMesh IP モードとの通信に使用するコマンドについて説明します。API はマシン間通信(センサー・アプリケーションがモードと通信する場合など)での使用を意図したものです。

それに対して、コマンド・ライン・インターフェース(CLI)は、開発中やインタラクティブなトラブルシューティングなどで、モードとやり取りするために使用します。詳細については、[SmartMesh IP Mote CLI Guide](#) を参照してください。

3 モートのシリアル・モード


CLI UART は以下の 1 つの動作モードをサポートします。

- モード 0: 9600 ボー、HDLC エンコードなし、
 - 2 線式インターフェース: UART_RX および UART_TX 信号のみを使用

API UART は、デバイスのヒューズ表で設定される以下の 2 つの動作モードをサポートします。

- モード 2: 9600 または 115.2K ボー、HDLC エンコード
 - 6 線式インターフェース: 全ての UART 信号を使用
- モード 4: 9600 または 115.2K ボー、HDLC エンコード
 - 4 線式インターフェース: TX、RX、UART_TX_CTSn、UART_TX_RTSn 信号を使用

信号タイミングの詳細については、デバイスのデータシートを参照してください。

 LTC5800-IPM のヒューズ表は、通常はボード・レベルの設計プロセスの一部として開発されます。API ポート上ではモード 2 またはモード 4 のどちらもそれぞれのボー・レートで使用できます。LTP5901-IPM などのモジュール式認証を取得している製品やスタータ・キット・モート(DC9003A-B)では、ヒューズ表はモード 4 の 115.2Kbps 用に事前にプログラムされ、変更できません。

4 プロトコル

モートのシリアル API は、厳密に定義された、容易に統合できるアプリケーション・プログラミング・インタフェース (API) を、モートのシリアル・インタフェースを介して OEM に提供します。このインタフェースを介して、OEM はモートとネットワークの豊富な機能にアクセスできます。API には、モートの設定と制御、モートの設定のクエリ、ワイヤレス・メッシュ・ネットワーク経由のデータ送受信、および RF 機能のテストに使用するコマンドが含まれます。

モートとマイクロプロセッサ間の接続は有線シリアル接続です。電圧シグナリング・レベル、シリアル・ハンドシェイクの定義、および信号タイミングの詳細および仕様については、モート製品のデータシートを参照してください。本書で説明するモートのシリアル API は、このインタフェースを介して動作します。

4.1 データ表現

4.1.1 共通データ・タイプ

この API ガイドでは、データを表すために以下のデータ・タイプを使用します。

タイプ	長さ(バイト)	注釈
INT8U	1	符号なしバイト。
INT16U	2	短い符号なし整数。
INT32U	4	長い符号なし整数。
INT8S	1	符号付きバイトまたは文字。
INT16S	2	短い符号付き整数。
INT32S	4	長い符号付き整数。
INT8U[n]	N	固定サイズ・アレイ。固定サイズ・アレイは常に[n]個の要素を含みます。有効な値が[n]個より少ない場合、残りがデフォルト値で埋められます。
INT8U[]	可変	可変長アレイ。可変長アレイのサイズは、パケットの長さによって決まります。可変長アレイは、常に、パケット構造の最後のフィールドになります。
IPV6_ADDR	16	INT8U[16]バイト・アレイとして表される IPV6 アドレス。
ASN	5	絶対スロット番号 (ASN) は、ネットワークが起動してからのタイム・スロット数で、5 バイトの整数で表されます。
UTC_TIME	8	UTC 時刻は、1970 年 1 月 1 日午前 0 時 (UTC) 以降の秒数およびマイクロ秒数です。シリアル化された形式は以下のとおりです。 <ul style="list-style-type: none"> INT32U - seconds - 1970 年 1 月 1 日午前 0 時以降の秒数 INT32U - microseconds - 現在の秒が始まってからのマイクロ秒数

UTC_TIME_L	12	Long UTC 時刻は、1970 年 1 月 1 日午前 0 時(UTC)以降の秒数およびマイクロ秒数です。シリアライズされた形式は以下のとおりです。 <ul style="list-style-type: none"> ● INT64 - seconds - 1970 年 1 月 1 日午前 0 時以降の秒数 ● INT32 - microseconds - 現在の秒が始まってからのマイクロ秒数
MAC_ADDR	8	INT8U[8]バイト・アレイとして表される EUI-64 識別子または MAC アドレス。
SEC_KEY	16	INT8U[16]バイト・アレイとして表されるセキュリティ・キー。
BOOL	1	True(= 1)、False(= 0)。ブール・フィールドは 1 バイトを占めます。
APP_VER	5	アプリケーションのバージョン。シリアライズされた形式は以下のとおりです。 <ul style="list-style-type: none"> ● INT8U - major - メジャー・バージョン ● INT8U - minor - マイナー・バージョン ● INT8U - patch - パッチ・バージョン ● INT16U - build - ビルド・バージョン

4.1.2 整数表現

マルチバイトの数値フィールドは、最上位ビット(MSB)から始まる順序のオクテット文字列として表されます。全てのオクテットは、最上位ビット(MSB)で始まるバイナリ文字列として表現されます。符号付き整数は、2 の補数形式で表されます。

INT8S、INT8U

ビット 7	...	ビット 0
MSB		LSB

INT16S、INT16U

ビット 15~ビット 8	ビット 7~ビット 0
--------------	-------------

INT32S、INT32U

ビット 31~ビット 24	ビット 23~ビット 16	ビット 15~ビット 8	ビット 7~ビット 0
---------------	---------------	--------------	-------------

ASN

ビット 39~ビット 32	ビット 31~ビット 24	ビット 23~ビット 16	ビット 15~ビット 8	ビット 7~ビット 0
---------------	---------------	---------------	--------------	-------------

4.1.3 伝送順序


本書に示す構造はすべて、伝送される順序で左から右へ(表形式の場合は上から下へ)と示します。

4.2 パケット形式

4.2.1 HDLC パッケージのカプセル化

HDLC プロトコルは、モートとシリアル・マイクロプロセッサの間の全ての API 通信に使用されます。全てのパケットは、RFC1662 に規定された HDLC フレーミングでカプセル化されます。ただしこの API は、RFC1662 の全ての項目に準拠しているわけではありません。フレーミング(すなわち、開始フラグと停止フラグ)、ペイロード内のフラグのエスケープ処理、および FCS のみが、RFC1662 から使用されます。パケットは 0x7E フラグで開始および終了し、16 ビットの CRC-CCITT フレーム・チェック・シーケンス(FCS)を含みます。パケットには、RFC1662 で言及している HDLC 制御およびアドレス・フィールドは含まれないことに注意してください。

開始フラグ (バイト 0)	HDLC ペイロード (バイト 1-n)	FCS (バイト n+1, n+2)	終了フラグ (バイト n+3)
0x7E	HDLC エスケープ処理された API ペイロード	(2 バイト)	0x7E

 一部の製品では、高ビット・レートで正常に動作するには追加の 0x7E 開始デリミタが必要です。詳細については、製品のデータシートを参照してください。

バイト・スタッフィングを使用して、ペイロードまたは FCS フィールドに含まれる可能性のある、フラグ・シーケンス(0x7E)バイトと制御エスケープ(0x7D)バイトをエスケープ処理できます。非同期制御文字マップ(ACCM)メカニズムは使用しないため、他の全てのバイト値はエスケープなしで送信可能です。FCS 演算の実行後、送信側は、開始フラグ・シーケンスと終了フラグ・シーケンスの間のフレーム全体を調べます。それぞれの 0x7E と 0x7D(開始フラグと終了フラグを除く)は、制御エスケープ(0x7D)と、それに後く元のバイトと 0x20 の XOR の結果からなる、2 バイト・シーケンスで置き換えられます。すなわち、次のようになります。

- 0x7D -> 0x7D 0x5D
- 0x7E -> 0x7D 0x5E

HDLC のエンコーディングの例

以下のペイロードを送信する必要があります(コマンド、長さ、フラグ、データ)

ペイロード
03 07 02 00 00 00 00 03 00 7D

以下の FCS を計算して付加します。

ペイロード	FCS
03 07 02 00 00 00 00 03 00 7D	9A B2

バイト・スタッフィングを実行します。この場合、1つの 0x7D をエスケープ処理する必要があります。

ペイロードと FCS(スタッフィング済み)
03 07 02 00 00 00 00 03 00 7D 5D 9A B2

最後に開始フラグと終了フラグを追加します。パケットを伝送する準備ができました。

フラグ	ペイロードと FCS(スタッフィング済み)	フラグ
7E	03 07 02 00 00 00 00 03 00 7D 5D 9A B2	7E

HDLC のデコードの例

以下のパケットを受信したとします。

フラグ	ペイロードと FCS(スタッフィング済み)	フラグ
7E	04 03 01 00 03 00 7D 5E A2 91	7E

開始フラグと終了フラグを削除します。

ペイロードと FCS(スタッフィング済み)
04 03 01 00 03 00 7D 5E A2 91

バイト・アンスタッフィングを実行します。この場合、1シーケンスの 0x7D 0x5E が 0x7E に置き換えられます。これで最後の 2 バイトを FCS として処理できます。

ペイロード	FCS
04 03 01 00 03 00 7E	A2 91

最後に、ペイロードの FCS と受信した最後の 2 バイトが一致することを確認します。一致していれば、ペイロードは有効とされ、処理されます。

ペイロード
04 03 01 00 03 00 7E

4.2.2 HDLC ペイロードの内容

シリアル・インターフェース上に送信される全てのパケットは、共通の API ヘッダとそれに続く API ペイロードを持ちます。ヘッダ内のフラグは、ペイロードを要求または応答として識別します。

開始	API ヘッダ	API ペイロード	FCS	終了
0x7E	コマンド 長さ フラグ	レスポンス・コード(応答のみ) メッセージ・ペイロード	(2 バイト)	0x7E

HDLC ペイロードの形式

API ヘッダ

API ヘッダには以下のフィールドが含まれます。

フィールド	タイプ	
Command Id	INT8U	コマンド識別子(コマンドを参照)
Len	INT8U	API ペイロードの長さ(このヘッダを除く)
フラグ	INT8U	パケット・フラグ

Flags フィールド

Flags は以下のフィールドを含む INT8U フィールドです。

ビット	説明
0 (LSB)	0=要求、1=応答
1	パケット ID
2	予約ビット、0 にセットされます。
3	Sync
4	予約ビット、0 にセットされます。
5	予約ビット、0 にセットされます。
6	予約ビット、0 にセットされます。
7 (MSB)	予約ビット、0 にセットされます。

要求/応答フラグ

要求/応答フラグは、パケット内のペイロードが要求ペイロードか応答ペイロードかを識別します。各コマンドおよび通知のペイロードは固有であり、後述のセクションで定義されます。

パケット ID

パケット ID は、パケットを確実に正しい順序で処理するために使用される、1 ビットのシーケンス番号です。

送信者は、新しいパケットが送信される場合は **Packet id** フィールドをトグルし、再伝送に備えてそのフィールドを変更しないままにしておく必要があります。受信者は、受信した **Packet id** を追跡する必要があります。新しいパケットを受信した場合、そのパケットは処理され、応答には **Packet id** のコピーが含まれます。重複する **Packet id** を受信した場合は、応答のキャッシュされたコピーが送信されます。

i モートはパケット ID の繰り返しを重複パケットとして扱います。モートは(コマンド自体は同じパケット ID を持つ前のコマンドとは異なる場合でも)受信したコマンドを除外し、前のコマンドに対するキャッシュされた応答を返します。

Sync フラグ

Sync ビットを使用して、シリアル・リンク上のシーケンス番号をリセットします。

モートからの最初の要求パケットは、Sync フラグがセットされています。これにより、シリアル・マイクロプロセッサは、デバイスが起動し、通信を確立していることを認識します。同様に、シリアル・マイクロプロセッサからの最初の要求は、Sync フラグを含んでいる必要があります。

API ペイロード

パケットの API ペイロード部分には、**コマンド**および**通知**セクションに示した要求ペイロードまたは応答ペイロードが含まれます。

⚠ 応答の場合、最初のペイロード・バイトは常にレスポンス・コードになり、これは長さには含まれません。

最大パケット・サイズ

要求パケットと応答パケットの最大 HDLC ペイロード・サイズは 128 バイトです。これには開始／停止デリミタやフレーム・チェックサムは含まれません。オクテット・スタッフィング・エスケープ・シーケンスはこの制限に含まれません。

4.3 モートとマイクロプロセッサ間の通信

4.3.1 アクノリッジ対象リンク

シリアル・リンクを介して送信される全てのパケットは、アクノリッジされる必要があります。ACK を受信するまでは、次のシリアル・パケットを送信できません。これはモートとマイクロプロセッサのどちらから開始されるパケットにも適用されます。受信側が新しいパケットと再伝送されたパケットを識別できるように、送信者は、新しいパケットが送信された場合や RC_NO_RESOURCES コードを持つ応答を受信した場合、**Packet id** フィールドをトグルする必要があります。送信者が応答を受信しない場合は、Packet id フィールドは変更されないままになる必要があります。

パケットの再伝送の場合も、モートに同じ規則が適用されます。

4.3.2 前方互換性を備えたクライアントのガイドライン

シリアル API プロトコルは、モート・ソフトウェアのリリースが新しくなってもクライアントの互換性を維持できるように設計されています。将来的なモート・ソフトウェアの改定によって以下の変更が生じる可能性を考慮に入れる必要があります。

- 新しいフィールドを追加するために、ペイロードが拡張される場合があります。新規フィールドは、最後または予約バイトの位置に追加されます。
- 既存のフィールドが廃止される場合がありますが、フィールド自体は削除されません。
- 新しいコマンドおよび通知が追加される場合があります。
- 新しいアラームおよびイベントが追加される場合があります。
- 新しいレスポンス・コードが追加される場合があります。

互換性を維持するには、クライアントが以下のルールに従う必要があります。

- 受信した応答ペイロードが想定より長い場合、余分なバイトは無視し、既知のバイトのみを処理します。
- 認識されない通知タイプを持つパケットを受信した場合、RC_OK を使用してアクノリッジします。
- 認識されないアラームまたはイベントを受信した場合、RC_OK を使用してアクノリッジします。
- 予約フィールドの値は決して使用せず、無視します。
- 要求ペイロードで未使用または予約としてマークされているフィールドでは、特別な記載のない限り値を 0 にセットします。
- 認識されないレスポンス・コードを受信した場合、一般的なエラー・レスポンス・コードとして扱います。

上記以外の互換性のない方法でプロトコルが変更された場合、[getParameter<motelInfo>](#) で報告されるプロトコル・バージョンが変更されます。

5 コマンド

以下のセクションでは、モートがサポートしているコマンドの API ペイロードについて説明します。API ヘッダ・バイトについては記載していないことに注意してください。

API ヘッダ

API ペイロード

コマンドの内容

5.1 blink(0x2E)

説明

ネットワークに参加することなく、ネットワークにパケットを送信します。モートはネットワークを検索し、パケットを送信します。オプションにより、検索プロセス中に検出された(最大 4 つまでの)近隣モートのリストも送信されます。検出された近隣モートのリストを含まない場合、ペイロードの最大サイズは 73B です。検出された近隣モートを含む場合、最大サイズは 58B です。

blink コマンドを受信すると、モートは *blink* ステートに移行し、アドバタイズメントの検索を開始します。アドバタイズメントを受信すると、モートは同期し、更に近隣モートを検出しようとして短時間リスニングを続けます。この短いタイムアウトの後、モートは検出された近隣モートのうち 1 つに直ちにデータ・パケットを送信します。*blink* コマンドを繰り返して呼び出して連続するパケットを送信した場合、検出された近隣モートのリストを要求されない限り、モートは再検索を実行しません。

モートがパケットの送信に成功すると、*status* が 0 にセットされた *txDone* 通知が送信されます。モートがパケットを送信できない場合(例えば MAC 層 ACK を受信しないまま 60 秒が経過した場合)、*status* が 1 にセットされた *txDone* 通知が送信されます。

i ブリンク・パケットの場合、モートは一度に 1 つのパケットしか受け付けません。複数のパケットを送信するには、アプリケーションは *txDone* 通知を待たなければなりません。MAC 層 ACK を受信しないまま 60 秒が経過すると、モートは低消費電力スリープに戻ります。したがって、モートがスリープ状態に戻るのを防ぐには、アプリケーションは 60 秒のタイムアウトが経過する前にパケットを送信する必要があります。マネージャ側のブリンク通知の詳細については、SmartMesh Embedded IP Manager API Guide を参照してください。

i リセットされるか、または *join* コマンドが発行されるまで、モートは *blink* ステートにとどまります。これにより、少なくとも 1 つのブリンク・パケットの送信後は、いつでも *getTime* コマンドを呼び出すことができます。なお、最後のパケットの送信以降、クロックは自走し、ドリフトします。ただし、例えば 1 日に 1 回パケットを送信する場合など、多くのアプリケーションでは比較的正確なクロック・ソースとして使用できます。

要求

パラメータ	タイプ	一覧	説明
<i>flIncludeDscvNbrs</i>	INT8U	なし	検出された近隣モートのリストをブリンク・パケットに含めるかどうかを指定します。検出された近隣モートを含めるには、1 にセットします。それ以外の場合は、0 にセットします。
<i>payload</i>	INT8U[]	なし	パケットのペイロード

応答

パラメータ	タイプ	一覧	説明
<i>rc</i>	INT8U	レスポンス・コード	レスポンス・コード

レスポンス・コード

コード	説明
RC_OK	パケットを受け付けた
RC_INVALID_STATE	次のブリンク・パケットはキュー内で送信待ち
RC_INVALID_LENGTH	ペイロードが長すぎる

5.2 bindSocket(0x17)

説明

開かれているソケットをポートにバインドします。ソケットの作成時には、プロトコル・ファミリーのみが指定され、ポートは割り当てられません。ソケットが他のホストからの接続を受け付けるには、この関連付けが実行されている必要があります。

要求

パラメータ	タイプ	一覧	説明
socketId	INT8U	なし	バインドするソケット ID
ポート	INT16U	なし	ソケットをバインドするポート

応答

パラメータ	タイプ	一覧	説明
rc	INT8U	レスポンス・コード	レスポンス・コード

レスポンス・コード

コード	説明
RC_OK	操作が正しく完了した
RC_NOT_FOUND	無効なソケット ID

5.3 clearNV(0x10)

説明

clearNV コマンドは、モートの不揮発性メモリ(NV)を工場出荷時の状態にリセットします。デフォルト値の詳細についてはユーザ・ガイドを参照してください。多くのパラメータは起動時にのみモートによって読み出されるため、このコマンドに続いてモートのリセットを実行する必要があります。

要求

パラメータ	タイプ	一覧	説明
-------	-----	----	----

応答

パラメータ	タイプ	一覧	説明
rc	INT8U	レスポンス・コード	リターン・コード

レスポンス・コード

コード	説明
RC_OK	コマンドが正しく完了した
RC_WRITE_FAIL	フラッシュの操作に失敗した

5.4 closeSocket(0x16)

説明

開かれているソケットを閉じます。

要求

パラメータ	タイプ	一覧	説明
socketId	INT8U	なし	ソケット ID

応答

パラメータ	タイプ	一覧	説明
rc	INT8U	レスポンス・コード	レスポンス・コード

レスポンス・コード

コード	説明
RC_OK	ソケットを閉じるのに成功した
RC_NOT_FOUND	ソケット ID が見つからない

5.5 disconnect(0x07)

説明

disconnect コマンドは、モートがネットワークからの切断を開始するように要求します。切断の完了後、モートは切断イベントを生成し、リセットに進みます。モートがネットワーク内にない場合、切断イベントは直ちに生成されます。モートが現在ネットワークに参加していない場合、このコマンドはエラーを返します。

要求

パラメータ	タイプ	一覧	説明
-------	-----	----	----

応答

パラメータ	タイプ	一覧	説明
rc	INT8U	レスポンス・コード	レスポンス・コード

レスポンス・コード

コード	説明
RC_OK	コマンドが受け付けられた

5.6 getParameter(0x02)

5.6.1 getParameter<appInfo>

説明

(ネットワーク・スタックに対して)アプリケーションのバージョン情報を取得します。

要求

パラメータ	タイプ	一覧	説明
paramId	INT8U	パラメータ	パラメータ ID (appInfo)

応答

パラメータ	タイプ	一覧	説明
rc	INT8U	なし	レスポンス・コード
paramId	INT8U	パラメータ	パラメータ ID (appInfo)
vendorId	INT16U	なし	ベンダ ID
appId	INT8U	なし	アプリケーション ID
appVer	APP_VER	なし	アプリケーションのバージョン

レスポンス・コード

コード	説明
RC_OK	コマンドが正しく完了した

5.6.2 getParameter<antGain>

説明

`getParameter<antGain>` コマンドは、(放射電力を正確に計算するために)システムが使用するアンテナ利得を取得します。デフォルトでは 2dBi になります。

注記: このパラメータは、1.4.x 以上のモート・ソフトウェアを実行しているデバイスで利用可能です。

要求

パラメータ	タイプ	一覧	説明
paramId	INT8U	パラメータ	パラメータ ID (<code>antGain</code>)

応答

パラメータ	タイプ	一覧	説明
rc	INT8U	レスポンス・コード	レスポンス・コード
paramId	INT8U	パラメータ	パラメータ ID (<code>antGain</code>)
antGain	INT8S	なし	アンテナ利得 (dBi)

レスポンス・コード

コード	説明
RC_OK	コマンドが正しく完了した

5.6.3 getParameter<autoJoin>

説明

このコマンドを利用して、マイクロプロセッサは現在の *autoJoin* の設定を取得できます。

要求

パラメータ	タイプ	一覧	説明
paramId	INT8U	パラメータ	パラメータ ID (<i>autoJoin</i>)

応答

パラメータ	タイプ	一覧	説明
rc	INT8U	レスポンス・コード	レスポンス・コード
paramId	INT8U	パラメータ	パラメータ ID (<i>autoJoin</i>)
autoJoin	BOOL		0=自動参加しない、1=自動参加

レスポンス・コード

コード	説明
RC_OK	コマンドが正しく完了した

5.6.4 getParameter<charge>

説明

getParameter<charge>コマンドは、前回のリセット以降のモートの消費電荷量を取得します。

要求

パラメータ	タイプ	一覧	説明
paramId	INT8U	パラメータ	パラメータ ID (<i>charge</i>)

応答

パラメータ	タイプ	一覧	説明
rc	INT8U	レスポンス・コード	レスポンス・コード
paramId	INT8U	パラメータ	パラメータ ID (<i>charge</i>)
qTotal	INT32U	なし	前回のリセット以降の電荷量(ミリクーロン)
upTime	INT32U	なし	リセット以降の時間(秒)
tempInt	INT8S	なし	温度(整数部分、°C)
tempFrac	INT8U	なし	温度(小数部分、1/255°C)

レスポンス・コード

コード	説明
RC_OK	コマンドが正しく完了した

5.6.5 getParameter<entropy>

説明

getParameter<entropy>コマンドを使用して、ランダム・データの 16 バイト・ブロックを取得できます。このデータは、無線フロントエンドを無効にした LTC5800 受信信号チェーン内の熱ノイズから得られます。したがって、このコマンドは、モートがアイドル状態のときにのみ呼び出すことができます。このコマンドは暗号化処理に適していますが、この制限のため、DRBG のシードとして使用することを推奨します。

このパラメータは、モート・ソフトウェア 1.4.x およびそれ以降を実行しているデバイスで利用可能です。

要求

パラメータ	タイプ	一覧	説明
paramId	INT8U	パラメータ	パラメータ ID(entropy)

応答

パラメータ	タイプ	一覧	説明
rc	INT8U	レスポンス・コード	レスポンス・コード
paramId	INT8U	パラメータ	パラメータ ID(entropy)
entropy	INT8U[16]	なし	エントロピー

レスポンス・コード

コード	説明
RC_OK	コマンドが正しく完了した
RC_INVALID_STATE	モートが無効な状態のためエントロピーを取得できない

5.6.6 getParameter<euCompliantMode>

説明

getParameter<euCompliantMode>コマンドを使用して、デバイスが使用する EN 300 328 準拠モードを取得できます。有効にした場合、モードは平均消費電力を制限値以内に抑えるため、一部の伝送の機会をスキップします。放射電力が+10dBm 未満のモードには、EN 300 328 の条件を満たすデューティ・サイクルは必要ありません。

注記: このパラメータは、1.4.x 以上のモード・ソフトウェアを実行しているデバイスで利用可能です。

要求

パラメータ	タイプ	一覧	説明
paramId	INT8U	パラメータ	パラメータ ID (<i>euCompliantMode</i>)

応答

パラメータ	タイプ	一覧	説明
rc	INT8U	レスポンス・コード	レスポンス・コード
paramId	INT8U	パラメータ	パラメータ ID (<i>euCompliantMode</i>)
compMode	INT8U	なし	EN 300 328 準拠モード 0=オフ、1=オン

レスポンス・コード

コード	説明
RC_OK	コマンドが正しく完了した

5.6.7 getParameter<eventMask>

説明

`getParameter<eventMask>`を利用して、マイクロプロセッサは現在サブスクライブされているイベント・タイプを読み出すことができます。

要求

パラメータ	タイプ	一覧	説明
paramId	INT8U	パラメータ	パラメータ ID (<code>eventMask</code>)

応答

パラメータ	タイプ	一覧	説明
rc	INT8U	レスポンス・コード	レスポンス・コード
paramId	INT8U	パラメータ	パラメータ ID (<code>eventMask</code>)
eventMask	INT32U	イベント	イベント・マスク 1=サブスクライブされている、0=サブスクライブされていない

レスポンス・コード

コード	説明
RC_OK	コマンドが正しく完了した

5.6.8 getParameter<ipv6Address>

説明

このコマンドを利用して、マイクロプロセッサはモートに割り当てられた IPV6 アドレスを読み出すことができます。モートがまだアドレスを割り当てられていない場合は、全て 0 を返します。

要求

パラメータ	タイプ	一覧	説明
paramId	INT8U	パラメータ	パラメータ ID (<i>ipv6Address</i>)

応答

パラメータ	タイプ	一覧	説明
rc	INT8U	レスポンス・コード	レスポンス・コード
paramId	INT8U	パラメータ	パラメータ ID (<i>ipv6Address</i>)
ipv6Address	IPV6_ADDR	なし	IPV6 アドレス

レスポンス・コード

コード	説明
RC_OK	コマンドが正しく完了した

5.6.9 getParameter<joinDutyCycle>

説明

このコマンドを利用して、*joinDutyCycle* パラメータの現在値を取得できます。

要求

パラメータ	タイプ	一覧	説明
paramId	INT8U	パラメータ	パラメータ ID (<i>joinDutyCycle</i>)

応答

パラメータ	タイプ	一覧	説明
rc	INT8U		レスポンス・コード
paramId	INT8U	パラメータ	パラメータ ID (<i>joinDutyCycle</i>)
joinDutyCycle	INT8U		デューティ・サイクル(0~255)、 (0=0.2%、255=99.8%)

レスポンス・コード

コード	説明
RC_OK	コマンドが正しく完了した

5.6.10 getParameter<macAddress>

説明

このコマンドは、デバイスの MAC アドレスを返します。デフォルトでは、返される MAC アドレスはモート・メーカーによって割り当てられるデバイスの EUI64 アドレスですが、このアドレスは `setParameter<macAddress>` コマンドを使用して上書きできます。

要求

パラメータ	タイプ	一覧	説明
paramId	INT8U	パラメータ	パラメータ ID (<i>macAddress</i>)

応答

パラメータ	タイプ	一覧	説明
rc	INT8U	なし	レスポンス・コード
paramId	INT8U	パラメータ	パラメータ ID (<i>macAddress</i>)
macAddress	MAC_ADDR	なし	デバイスの MAC アドレス

レスポンス・コード

コード	説明
RC_OK	コマンドが正しく完了した

5.6.11 getParameter<shortAddress>

説明

このコマンドは、モートのモート ID を取得します。モートがネットワーク内にはない場合は、0 が返されます。

要求

パラメータ	タイプ	一覧	説明
paramId	INT8U	パラメータ	パラメータ ID (<i>moteld</i>)

応答

パラメータ	タイプ	一覧	説明
rc	INT8U	レスポンス・コード	レスポンス・コード
paramId	INT8U	パラメータ	パラメータ ID (<i>moteld</i>)
moteld	INT16U		モート ID

レスポンス・コード

コード	説明
RC_OK	コマンドが正しく完了した

5.6.12 getParameter<moteInfo>

説明

`getParameter<moteInfo>` コマンドは、モートのハードウェアとネットワーク・スタック・ソフトウェアに関する静的な情報を返します。

要求

パラメータ	タイプ	一覧	説明
paramId	INT8U	パラメータ	パラメータ ID (<i>moteInfo</i>)

応答

パラメータ	タイプ	一覧	説明
rc	INT8U	レスポンス・コード	レスポンス・コード
paramId	INT8U	パラメータ	パラメータ ID (<i>moteInfo</i>)
apiVersion	INT8U		API プロトコルのバージョン
serialNumber	INT8U[8]		デバイスのシリアル番号
hwModel	INT8U		ハードウェア・モデル
hwRev	INT8U		ハードウェア・リビジョン
swVerMajor	INT8U		ネットワーク・スタック・ソフトウェアのバージョン: メジャー
swVerMinor	INT8U		ネットワーク・スタック・ソフトウェアのバージョン: マイナー
swVerPatch	INT8U		ネットワーク・スタック・ソフトウェアのバージョン: パッチ
swVerBuild	INT16U		ネットワーク・スタック・ソフトウェアのバージョン: ビルド
bootSwVer	INT8U		ブートローダ・ソフトウェアのバージョン:

レスポンス・コード

コード	説明
RC_OK	コマンドが正しく完了した

5.6.13 getParameter<moteStatus>

説明

`getParameter<moteStatus>` コマンドを使用して、現在のモートの状態やその他の動的情報を取得できます。

要求

パラメータ	タイプ	一覧	説明
paramId	INT8U	パラメータ	パラメータ ID (<i>moteStatus</i>)

応答

パラメータ	タイプ	一覧	説明
rc	INT8U	レスポンス・コード	レスポンス・コード
paramId	INT8U	パラメータ	パラメータ ID (<i>moteStatus</i>)
state	INT8U	モートのステート	モートのステート
reserved_1	INT8U[3]		このフィールドは無視される
numParents	INT8U		親の数
alarms	INT32U	アラーム	現在のアラーム (ビットマップ)
reserved_2	INT8U		このフィールドは無視される

レスポンス・コード

コード	説明
RC_OK	コマンドが正しく完了した

5.6.14 getParameter<netInfo>

説明

`getParameter<networkInfo>`コマンドを使用して、モートのネットワーク関連パラメータを取得できます。

要求

パラメータ	タイプ	一覧	説明
paramId	INT8U	パラメータ	パラメータ ID (<i>netInfo</i>)

応答

パラメータ	タイプ	一覧	説明
rc	INT8U	レスポンス・コード	レスポンス・コード
paramId	INT8U	パラメータ	パラメータ ID (<i>netInfo</i>)
macAddress	MAC_ADDR	なし	デバイスの MAC アドレス
moteld	INT16U	なし	モート ID
networkId	INT16U	なし	ネットワーク ID
slotSize	INT16U	なし	スロット・サイズ (マイクロ秒)

レスポンス・コード

コード	説明
RC_OK	コマンドが正しく完了した

5.6.15 getParameter<networkId>

説明

このコマンドは、モートの永続的なストレージに格納されるネットワーク ID を返します。

要求

パラメータ	タイプ	一覧	説明
paramId	INT8U	パラメータ	パラメータ ID (<i>networkId</i>)

応答

パラメータ	タイプ	一覧	説明
rc	INT8U		レスポンス・コード
paramId	INT8U	パラメータ	パラメータ ID (<i>networkId</i>)
networkId	INT16U		ネットワーク識別子

レスポンス・コード

コード	説明
RC_OK	コマンドが正しく完了した

5.6.16 getParameter<OTAPLockout>

説明

このコマンドは、OTAP ロックアウトの現在の状態(すなわち、このモード上で無線通信経由でのソフトウェア・アップグレードが許可されているかどうか)を読み出します。

要求

パラメータ	タイプ	一覧	説明
paramId	INT8U	パラメータ	パラメータ ID (OTAPLockout)

応答

パラメータ	タイプ	一覧	説明
rc	INT8U	レスポンス・コード	レスポンス・コード
paramId	INT8U	パラメータ	パラメータ ID (OTAPLockout)
mode	BOOL		0=OTAP が許可されている、 1=OTAP が許可されていない

レスポンス・コード

コード	説明
RC_OK	コマンドが正しく完了した

5.6.17 getParameter<pwrSrcInfo>

説明

このコマンドを利用して、マイクロプロセッサはモートの電源設定を読み出すことができます。

要求

パラメータ	タイプ	一覧	説明
paramId	INT8U	パラメータ	パラメータ ID (<i>powerSrcInfo</i>)

応答

パラメータ	タイプ	一覧	説明
rc	INT8U	レスポンス・コード	レスポンス・コード
paramId	INT8U	パラメータ	パラメータ ID (<i>powerSrcInfo</i>)
maxStCurrent	INT16U	なし	定常状態の最大電流 (μA)、0xFFFF=無制限
minLifetime	INT8U	なし	最小寿命 (月)、0=無制限
currentLimit_0	INT16U	なし	電流制限 (μA)、0=制限エントリが空白
dischargePeriod_0	INT16U	なし	放電時間 (秒)
rechargePeriod_0	INT16U	なし	再充電時間 (秒)
currentLimit_1	INT16U	なし	電流制限 (μA)、0=制限エントリが空白
dischargePeriod_1	INT16U	なし	放電時間 (秒)
rechargePeriod_1	INT16U	なし	再充電時間 (秒)
currentLimit_2	INT16U	なし	電流制限 (μA)、0=制限エントリが空白
dischargePeriod_2	INT16U	なし	放電時間 (秒)
rechargePeriod_2	INT16U	なし	再充電時間 (秒)

レスポンス・コード

コード	説明
RC_OK	コマンドが正しく完了した

5.6.18 getParameter<routingMode>

説明

このコマンドを利用して、マイクロプロセッサはモートの現在のルーティング・モードを取得できます。

要求

パラメータ	タイプ	一覧	説明
paramId	INT8U	パラメータ	パラメータ ID (<i>routingMode</i>)

応答

パラメータ	タイプ	一覧	説明
rc	INT8U	レスポンス・コード	レスポンス・コード
paramId	INT8U	パラメータ	パラメータ ID (<i>routingMode</i>)
routingMode	BOOL		ルーティング・モード (0=ルーティング有効、1=ルーティング無効)

レスポンス・コード

コード	説明
RC_OK	コマンドが正しく完了した

5.6.19 getParameter<testRadioRxStats>

説明

`getParameter<testRadioRxStats>` コマンドは、`testRadioRx` コマンドを使用して実行された最新の無線テストの統計情報を取得します。この統計情報は、テスト中に受信した正常パケットと不良パケット(CRC エラー)の数を示します。

要求

パラメータ	タイプ	一覧	説明
paramId	INT8U	パラメータ	パラメータ ID (<code>testRadioRxStats</code>)

応答

パラメータ	タイプ	一覧	説明
rc	INT8U	レスポンス・コード	レスポンス・コード
paramId	INT8U	パラメータ	パラメータ ID (<code>testRadioRxStats</code>)
rxOk	INT16U		受信に成功したパケットの数
rxFailed	INT16U		受信したエラーありのパケットの数(CRC エラー)

レスポンス・コード

コード	説明
RC_OK	コマンドが正しく完了した

5.6.20 getParameter<time>

説明

`getParameter<time>` コマンドを使用して、モート上の現在の時間を要求できます。モートはコマンドを処理した瞬間の時間を報告するため、情報には可変の遅延が含まれます。より正確な時間情報が必要な場合は、TIMEn ピン ([timeIndication](#)) を使用してください。

要求

パラメータ	タイプ	一覧	説明
paramId	INT8U	パラメータ	パラメータ ID (<i>time</i>)

応答

パラメータ	タイプ	一覧	説明
rc	INT8U	レスポンス・コード	レスポンス・コード
paramId	INT8U	パラメータ	パラメータ ID (<i>time</i>)
upTime	INT32U	なし	前回のリセットからの時間 (秒)
utcTime	UTC_TIME_L	なし	long UTC 時刻構文
asn	ASN	なし	ネットワーク (スロット) の開始以降の時間
asnOffset	INT16U	なし	このスロットの開始からの時間 (ミリ秒)

レスポンス・コード

コード	説明
RC_OK	コマンドが正しく完了した

5.6.21 getParameter<txPower>

説明

このコマンドは、アンテナ利得を除く無線出力電力 (dBm) を取得します。この値は無線ドライバによって使用される実際の出力電力に対応しており、setParameter<txPower>を使って入力された値 (ハードウェアが入力値をサポートしない場合は最も近い値に設定される) とは一致しないことがあります。

要求

パラメータ	タイプ	一覧	説明
paramId	INT8U	パラメータ	パラメータ ID (<i>txPower</i>)

応答

パラメータ	タイプ	一覧	説明
rc	INT8U	なし	レスポンス・コード
paramId	INT8U	パラメータ	パラメータ ID (<i>txPower</i>)
txPower	INT8S	なし	送信出力 (dBm)

レスポンス・コード

コード	説明
RC_OK	コマンドが正しく完了した

5.7 getServiceInfo (0x12)

説明

getServiceInfo コマンドは、モートに現在割り当てられているサービスに関する情報を返します。

要求

パラメータ	タイプ	一覧	説明
destAddr	INT16U	なし	サービスのメッシュ内の宛先のアドレス。現在サポートしている値はマネージャのアドレス(0xFFFE)のみです。
type	INT8U	サービスのタイプ	サービスのタイプ(帯域幅または遅延)

応答

パラメータ	タイプ	一覧	説明
rc	INT8U	レスポンス・コード	レスポンス・コード
destAddr	INT16U	なし	メッシュ内のサービスの宛先の短縮アドレス
type	INT8U	サービスのタイプ	サービスのタイプ
state	INT8U	サービスのステート	サービスのステート
value	INT32U	なし	パケット間の間隔(ms)

レスポンス・コード

コード	説明
RC_OK	コマンドが正しく完了した

5.8 join(0x06)

説明

join コマンドは、モートがネットワークの検索を開始し、参加を試みるように要求します。このコマンドが有効であるためには、モートがアイドル・ステートまたはプロミスキャス・リッスン・ステート([search](#) を参照)になっている必要があります。参加時間は最大電流の設定の影響を受けることに注意してください。

要求

パラメータ	タイプ	一覧	説明
-------	-----	----	----

応答

パラメータ	タイプ	一覧	説明
rc	INT8U	レスポンス・コード	レスポンス・コード

レスポンス・コード

コード	説明
RC_OK	コマンドが受け付けられた
RC_INVALID_STATE	モートが無効な状態のため参加操作を開始できない
RC_INCOMPLETE_JOIN_INFO	設定が不完全なため参加を開始できない

5.9 lowPowerSleep(0x09)

説明

lowPowerSleep コマンドは、全てのペリフェラルをシャットダウンし、モートをディープ・スリープ・モードに移行させます。このコマンドは、モートが応答を送信した後に実行されます。コマンドの実行後 2 秒以内に、モートはディープ・スリープに移行します。このコマンドはいつでも発行できます。モートは進行中のネットワーク操作を全て中断します。ネットワークからモートを正常に切断するには、*lowPowerSleep* コマンドを使用する前に *disconnect* コマンドを実行します。モートをディープ・スリープ・モードから復帰させるには、ハードウェア・リセットを実行する必要があります。

要求

パラメータ	タイプ	一覧	説明
-------	-----	----	----

応答

パラメータ	タイプ	一覧	説明
rc	INT8U	レスポンス・コード	レスポンス・コード

レスポンス・コード

コード	説明
RC_OK	コマンドが受け付けられた

5.10 openSocket(0x15)

説明

`openSocket` コマンドは、IP 通信用のエンドポイントを作成し、そのソケットの ID を返します。

要求

パラメータ	タイプ	一覧	説明
protocol	INT8U	プロトコルのタイプ	このソケットのプロトコル

応答

パラメータ	タイプ	一覧	説明
rc	INT8U	レスポンス・コード	レスポンス・コード
socketId	INT8U		ソケット ID

レスポンス・コード

コード	説明
RC_OK	コマンドが正しく完了した
RC_NO_RESOURCES	リソースが不足しているために新しいソケットを作成できない

5.11 requestService (0x11)

説明

`requestService` コマンドを使用して、メッシュ内の宛先デバイスに対する新しいサービス・レベルまたは変更されたサービス・レベルを要求できます。このコマンドは、既存の接続(セッション)があるデバイスに対するサービスの更新にのみ使用できます。

帯域幅の割り当てが変更された場合、アプリケーションは `serviceChanged` イベントを受信し、このイベントをトリガとして新しいサービス割り当てを読み出します。

要求

パラメータ	タイプ	一覧	説明
destAddr	INT16U		サービスのメッシュ内の宛先のアドレス。現在サポートしている値はマネージャのアドレス(0xFFFE)のみです。
serviceType	INT8U	サービスのタイプ	サービスのタイプ
value	INT32U		パケット間の間隔(ms)

応答

パラメータ	タイプ	一覧	説明
rc	INT8U	レスポンス・コード	レスポンス・コード

レスポンス・コード

コード	説明
RC_OK	サービス要求を受け付けた

5.12 reset(0x08)

説明

`reset` コマンドは、デバイスのソフトリセットを開始します。デバイスは、このコマンドに対する応答の送信後すぐにリセット・シーケンスを開始します。モートをリセットすると、下位モードに直接悪影響を与えます。ネットワークからモートを正常に切断するには、`disconnect` コマンドを使用します。

要求

パラメータ	タイプ	一覧	説明
-------	-----	----	----

応答

パラメータ	タイプ	一覧	説明
rc	INT8U	レスポンス・コード	レスポンス・コード

レスポンス・コード

コード	説明
RC_OK	コマンドが受け付けられた

5.13 search(0x24)

説明

`search` コマンドは、モートがアドバタイズメントのリッスンを開始し、参加を試みることなく任意のネットワークからリッスンした内容を報告するように要求します。これは**プロミスキャス・リッスン・ステート**と呼ばれます。このコマンドが有効であるためには、モートが**アイドル・ステート**になっている必要があります。検索ステートを終了するには、`join` コマンドまたは `reset` コマンドを発行します。

要求

パラメータ	タイプ	一覧	説明
-------	-----	----	----

応答

パラメータ	タイプ	一覧	説明
rc	INT8U	レスポンス・コード	レスポンス・コード

レスポンス・コード

コード	説明
RC_OK	コマンドが受け付けられた
RC_INVALID_STATE	モートが無効な状態のため検索操作を開始できない

5.14 sendTo(0x18)

説明

ネットワークにパケットを送信します。このコマンドが RC_OK を返した場合、モートはパケットを受け付けて、伝送するためにキューに入れました。パケットが送信されると、このコマンドで渡されるパケット ID が 0xffff とは異なる場合にのみ、*txDone* 通知が発行されます。パケット ID は任意の値に設定できます。この通知には送信されたばかりのパケットのパケット ID が含まれるため、通知と特定のパケットを関連付けることができます。ペイロードを最大化するには、宛先ポートを 0xF0B8~F0BF (61624~61631) の範囲内にする必要があります。

要求

パラメータ	タイプ	一覧	説明
socketId	INT8U	なし	ソケット ID
destIP	IPV6_ADDR	なし	宛先 IPV6 アドレス (LBR を介して) インターネット・ホストに送信しないアプリケーションでは、マネージャの既知のアドレス (FF02::2) が使用されます。
destPort	INT16U	なし	宛先ポート
serviceType	INT8U	サービスのタイプ	サービスのタイプ
priority	INT8U	パケット優先順位	パケットの優先順位。優先順位の低いパッケージは、優先順位の高いパケットの後でキューに入れます。データ・トラフィックには中位の優先順位を使用することを推奨します。
packetId	INT16U	なし	txDone 通知用のユーザ定義のパケット ID。0xFFFF=通知を生成しない
payload	INT8U[]	なし	パケットのペイロード

応答

パラメータ	タイプ	一覧	説明
rc	INT8U	レスポンス・コード	レスポンス・コード

レスポンス・コード

コード	説明
RC_OK	パケットは伝送のためにキューに入れられた
RC_NO_RESOURCES	キューの容量が不足しているためパケットを受け付けられない

ペイロード・サイズの制限

送信元／宛先ポート	マネージャへ	その他のアドレスへ	
両方とも F0Bx	90	74	バイト
一方(または両方)が F0xx	88	72	バイト
その他	87	71	バイト

5.15 setParameter(0x01)

`setParameter` コマンドを使用して、モート上のパラメータを変更できます。各 `setParameter` コマンドのペイロードは、変更されるパラメータを指定するパラメータ ID フィールドから始まります。このコマンドで変更される全てのパラメータは、リセットおよび電源入れ直しの前後で不揮発性メモリに永続的に維持されます。

5.15.1 setParameter<advKey>

説明

アドバタイズメントの MIC 鍵を設定します。この鍵はアドバタイズメントの認証に使用されるものであり、ベンダ/インストレーションごとに設定され、無許可のデバイスがアドバタイズメントに応答することを防ぎます。この鍵を変更した場合、モートがネットワークに参加するには、この鍵が(マネージャの CLI 上で `mset` を使用して)対応する AP 上で設定された鍵と一致する必要があります。この鍵をデフォルト値にリセットするには、`clearNV` コマンドを使用します。

要求

パラメータ	タイプ	一覧	説明
paramId	INT8U	パラメータ	パラメータ ID (advKey)
advKey	SEC_KEY	なし	16 バイトのアドバタイズメントの MIC 鍵

応答

パラメータ	タイプ	一覧	説明
rc	INT8U	レスポンス・コード	レスポンス・コード
paramId	INT8U	パラメータ	パラメータ ID (advKey)

レスポンス・コード

コード	説明
RC_OK	コマンドが正しく完了した
RC_WRITE_FAIL	永続的なストレージを更新できなかった

5.15.2 setParameter<antGain>

説明

setParameter<antGain>コマンドは、(放射電力を正確に計算するために)システムのアンテナ利得を設定します。設定しない場合、デフォルトにより 2dBi に設定されます。

注記:このパラメータは、1.4.x 以上のモート・ソフトウェアを実行しているデバイスで利用可能です。

要求

パラメータ	タイプ	一覧	説明
paramId	INT8U	パラメータ	パラメータ ID (<i>antGain</i>)
antGain	INT8S	なし	アンテナ利得 (dBi)

応答

パラメータ	タイプ	一覧	説明
rc	INT8U	レスポンス・コード	レスポンス・コード
paramId	INT8U	パラメータ	パラメータ ID (<i>antGain</i>)

レスポンス・コード

コード	説明
RC_OK	コマンドが正しく完了した
RC_WRITE_FAIL	永続的なストレージを更新できなかった

5.15.3 setParameter<autoJoin>

説明

このコマンドを利用して、マイクロプロセッサはモートのネットワーキング・スタックごとに自動参加と手動参加を切り替えることができます。手動モードでは、参加を開始するにはアプリケーションからの明示的な `join` コマンドが必要です。この設定は永続的であり、モートのリセット後に有効になります。

アプリケーションも参加するように設定されている場合は、自動参加モードに設定してはなりません（例えば、「自動参加」モードと「マスタ」モードを組み合わせると、モートはネットワークに参加しません）。

要求

パラメータ	タイプ	一覧	説明
paramId	INT8U	パラメータ	パラメータ ID (autoJoin)
mode	BOOL	なし	0=手動参加、1=自動参加

応答

パラメータ	タイプ	一覧	説明
rc	INT8U	レスポンス・コード	レスポンス・コード
paramId	INT8U	パラメータ	パラメータ ID (autoJoin)

レスポンス・コード

コード	説明
RC_OK	コマンドが正しく完了した
RC_WRITE_FAIL	永続的なストレージを更新できなかった
RC_INVALID_VALUE	無効なモード値が指定された

5.15.4 setParameter<euCompliantMode>

説明

setParameter<euCompliantMode>コマンドを使用して、デバイスが使用する EN 300 328 準拠モードを設定できます。有効にした場合、モートは平均消費電力を制限値以内に抑えるため、一部の伝送の機会をスキップします。放射電力が+10dBm 未満のモートには、EN 300 328 の条件を満たすデューティ・サイクルは必要ありません。注記:このパラメータは、1.4.x 以上のモート・ソフトウェアを実行しているデバイスで利用可能です。

要求

パラメータ	タイプ	一覧	説明
paramId	INT8U	パラメータ	パラメータ ID (euCompliantMode)
euCompliantMode	INT8U	なし	EN 300 328 準拠モード 0=オフ、1=オン

応答

パラメータ	タイプ	一覧	説明
rc	INT8U	レスポンス・コード	レスポンス・コード
paramId	INT8U	パラメータ	パラメータ ID (euCompMode)

レスポンス・コード

コード	説明
RC_OK	コマンドが正しく完了した
RC_WRITE_FAIL	永続的なストレージを更新できなかった

5.15.5 setParameter<eventMask>

説明

setParameter<eventMask>コマンドを利用して、マイクロプロセッサはイベント通知を選択的にサブスクライブできます。モートのリセット時の eventMask のデフォルト値は全て 1 で、全てのイベントが有効になります。この設定は永続的ではありません。

⊖ モート・ソフトウェアの将来のリビジョンでは、新しいイベント・タイプが追加される可能性があります。クライアント・コードは既知のイベントのみをサブスクライブし、未知のイベントを全て無視することを推奨します。

要求

パラメータ	タイプ	一覧	説明
paramId	INT8U	パラメータ	パラメータ ID (eventMask)
eventMask	INT32U	イベント	イベント・マスク、0=サブスクライブしない、1=サブスクライブする

応答

パラメータ	タイプ	一覧	説明
rc	INT8U	レスポンス・コード	レスポンス・コード
paramId	INT8U	パラメータ	パラメータ ID (eventMask)

レスポンス・コード

コード	説明
RC_OK	コマンドが正しく完了した

5.15.6 setParameter <joinDutyCycle>

説明

`setParameter<joinDutyCycle>` コマンドを利用して、マイクロプロセッサは、モートがネットワークを検索している期間の、アクティブなリスニング時間とドーズ時間（低消費電力の無線状態）の比率を制御できます。消費電力が高くなることを承知のうえで、高速な参加時間を求める場合、`setParameter<joinDutyCycle>` コマンドを使用して参加デューティ・サイクルを最大 100%に上げることができます。この設定は永続的であり、デバイスがネットワークを検索すると直ちに有効になります。

要求

パラメータ	タイプ	一覧	説明
paramId	INT8U	パラメータ	パラメータ ID (<code>joinDutyCycle</code>)
dutyCycle	INT8U	なし	デューティ・サイクル (0~255)、 (0=0.2%、255=99.8%)

応答

パラメータ	タイプ	一覧	説明
rc	INT8U	レスポンス・コード	レスポンス・コード
paramId	INT8U	パラメータ	パラメータ ID (<code>joinDutyCycle</code>)


レスポンス・コード

コード	説明
RC_OK	コマンドが正しく完了した

5.15.7 setParameter <joinKey>

説明

setParameter<joinKey>コマンドを使用して、モートの永続的なストレージ内の参加鍵を設定できます。モートは参加鍵を使用してネットワークとのセキュアな接続を確立します。参加鍵は次の参加時に使用されます。

 セキュリティ上の理由で、joinKey パラメータの読出しは禁止されています。

要求

パラメータ	タイプ	一覧	説明
paramId	INT8U	パラメータ	パラメータ ID (joinKey)
joinKey	SEC_KEY	なし	参加鍵

応答

パラメータ	タイプ	一覧	説明
rc	INT8U	レスポンス・コード	レスポンス・コード
paramId	INT8U	パラメータ	パラメータ ID (joinKey)

レスポンス・コード

コード	説明
RC_OK	コマンドが正しく完了した
RC_WRITE_FAIL	鍵をストレージに書き込めなかった

5.15.8 setParameter<macAddress>

説明

このコマンドを利用して、メーカーが割り当てたモートの MAC アドレスを上書きできます。新しい値はモートのリセット後に有効になります。

要求

パラメータ	タイプ	一覧	説明
paramId	INT8U	パラメータ	パラメータ ID (<i>macAddress</i>)
macAddress	MAC_ADDR	なし	使用する新しい MAC アドレス

応答

パラメータ	タイプ	一覧	説明
rc	INT8U	レスポンス・コード	レスポンス・コード
paramId	INT8U	パラメータ	パラメータ ID (<i>macAddress</i>)

レスポンス・コード


コード	説明
RC_OK	コマンドが正しく完了した
RC_WRITE_FAIL	永続的なストレージを更新できなかった

5.15.9 setParameter<networkId>

説明

このコマンドを使用して、モートのネットワーク ID を設定できます。この設定は永続的であり、次の参加試行時に使用されます。

バージョン 1.4.x から、0xFFFF のネットワーク ID を使用して、モートが最初に受信したネットワークに参加するように指示できるようになりました。

 0xFFFF が有効なネットワーク ID として無線通信経由で使用されることはありません。マネージャのネットワーク ID を 0xFFFF に設定してはなりません。

要求

パラメータ	タイプ	一覧	説明
paramId	INT8U	パラメータ	パラメータ ID (<i>networkId</i>)
networkId	INT16U	なし	ネットワーク ID

応答

パラメータ	タイプ	一覧	説明
rc	INT8U	レスポンス・コード	レスポンス・コード
paramId	INT8U	パラメータ	パラメータ ID (<i>networkId</i>)

レスポンス・コード

コード	説明
RC_OK	コマンドが正しく完了した
RC_WRITE_FAIL	パラメータを格納できなかった

5.15.10 setParameter<OTAPLockout>

説明

このコマンドを利用して、マイクロプロセッサはモートの無線プログラミング(OTAP)が許可されるかどうかを制御できます。この設定は永続的であり、直ちに有効になります。

要求

パラメータ	タイプ	一覧	説明
paramId	INT8U	パラメータ	パラメータ ID (OTAPLockout)
mode	BOOL		0=OTAP が許可されている、 1=OTAP が許可されていない

応答

パラメータ	タイプ	一覧	説明
rc	INT8U	レスポンス・コード	レスポンス・コード
paramId	INT8U	パラメータ	パラメータ ID (OTAPLockout)

レスポンス・コード

コード	説明
RC_OK	コマンドが正しく完了した
RC_INVALID_VALUE	モード・パラメータの無効な値
RC_WRITE_FAIL	永続的なストレージを更新できなかった

5.15.11 setParameter<pwrSrcInfo>

説明

このコマンドを利用して、マイクロプロセッサはデバイス上の電源情報を設定できます。この設定は永続的であり、ネットワーク参加時に使用されます。

要求

パラメータ	タイプ	一覧	説明
paramId	INT8U	パラメータ	パラメータ ID (powerSrcInfo)
maxStCurrent	INT16U		定常状態の最大電流 (μA)、0xFFFF=無制限
minLifetime	INT8U		最小寿命 (月)、0=無制限
currentLimit_0	INT16U		電流制限 (μA)、0=制限エントリが空白
dischargePeriod_0	INT16U		放電時間 (秒)
rechargePeriod_0	INT16U		再充電時間 (秒)
currentLimit_1	INT16U		電流制限 (μA)、0=制限エントリが空白
dischargePeriod_1	INT16U		放電時間 (秒)
rechargePeriod_1	INT16U		再充電時間 (秒)
currentLimit_2	INT16U		電流制限 (μA)、0=制限エントリが空白
dischargePeriod_2	INT16U		放電時間 (秒)
rechargePeriod_2	INT16U		再充電時間 (秒)

応答

パラメータ	タイプ	一覧	説明
rc	INT8U	レスポンス・コード	レスポンス・コード
paramId	INT8U	パラメータ	パラメータ ID (powerSrcInfo)

レスポンス・コード

コード	説明
RC_OK	コマンドが正しく完了した

5.15.12 setParameter<routingMode>

説明

このコマンドを利用して、マイクロプロセッサは、モートがネットワークへの参加後にルータになるかどうかを制御できます。無効にした場合、マネージャはモートをリーフ・ノードのままにします。

要求

パラメータ	タイプ	一覧	説明
paramId	INT8U	パラメータ	パラメータ ID (routingMode)
mode	BOOL		0=ルーティング、1=ルーティングしない

応答

パラメータ	タイプ	一覧	説明
rc	INT8U	レスポンス・コード	レスポンス・コード
paramId	INT8U	パラメータ	パラメータ ID (routingMode)

レスポンス・コード

コード	説明
RC_OK	コマンドが正しく完了した
RC_INVALID_VALUE	モード・パラメータの無効な値

5.15.13 setParameter<txPower>

要求

パラメータ	タイプ	一覧	説明
paramId	INT8U	パラメータ	パラメータ ID (<i>txPower</i>)
txPower	INT8S	なし	送信出力 (dBm)

応答

パラメータ	タイプ	一覧	説明
rc	INT8U	レスポンス・コード	レスポンス・コード
paramId	INT8U	パラメータ	パラメータ ID (<i>txPower</i>)

レスポンス・コード

コード	説明
RC_OK	コマンドが正しく完了した
RC_INVALID_VALUE	要求した電力はサポートしていない

5.16 socketInfo (0x2B)

説明

ソケットに関する情報を取得します (1.4.0 以上の IP モードで使用可能)。

要求

パラメータ	タイプ	一覧	説明
index	INT8U	なし	要求したソケットのインデックス。インデックスは 0 から始まります。

応答

パラメータ	タイプ	一覧	説明
rc	INT8U	レスポンス・コード	レスポンス・コード
index	INT8U	なし	要求したインデックス
socketId	INT8U	なし	要求したインデックスのソケットの ID
protocol	INT8U	なし	常に 0 (UDP)
bindState	INT8U	なし	0 = バインドされていない、1 = バインドされている
ポート	INT16U	なし	このソケットにバインドされているポート。 バインドされていない場合は 0

レスポンス・コード

コード	説明
RC_OK	操作が正しく完了した
RC_NOT_FOUND	指定したインデックスのソケットが見つからない

5.17 stopSearch(0x2F)

説明

`stopSearch` コマンドは、`join` または `search` コマンドによって開始された検索を停止します。このコマンドが有効であるためには、モートが**プロミスキャス・リッスン・ステート**または**検索ステート**になっている必要があります。有効な状態でこのコマンドを受信すると、モートは**アイドル・ステート**に戻ります。1.4 以上のモートで使用可能です。

要求

パラメータ	タイプ	一覧	説明
-------	-----	----	----

応答

パラメータ	タイプ	一覧	説明
rc	INT8U	レスポンス・コード	レスポンス・コード

レスポンス・コード

コード	説明
RC_OK	コマンドが受け付けられた
RC_INVALID_STATE	モートが無効な状態のため検索操作を停止できない

5.18 testRadioRx(0x0C)

説明

`testRadioRx` コマンドは、それまでに収集した全ての統計情報をクリアし、指定されたチャンネルおよび持続時間の無線受信テストを開始します。テスト中、モートで受信したパケット数に関する統計情報は維持されます(エラーあり/なしの両方)。テスト結果を取得するには、`getParameter<testRadioRxStats>` コマンドを使用します。`testRadioRx` コマンドを発行できるのは、**アイドル・モード**のときだけです。無線テストの完了から参加までの間に、モートは(ハードウェア・リセットまたはソフトウェア・リセットで)リセットされる必要があります。

i ステーション ID は 1.4 以上の IP モードおよび 1.1.2 以上の WirelessHART モードで利用可能です。ステーション ID は選択できる値であり、同じ無線空間で複数のテストが実行されている場合に、トラフィックを分離するために使用します。送信側で使用しているステーション ID と同じ値を設定してください。

⚠ チャンネル番号は 0~15 で、IEEE 2.4GHz チャンネルの 11~26 に相当します。

要求

パラメータ	タイプ	一覧	説明
channelMask	INT16U	なし	テスト用に有効化されたチャンネル(0~15)のマスク。ビット 0 はチャンネル 0 に対応します。1 チャンネルのみ有効化されます。
time	INT16U	なし	テストの持続時間(秒)
stationId	INT8U	なし	このモートの一意的ステーション ID(1~255)。送信側のステーション ID と一致しなければなりません。送信側のステーション ID を無視する場合、値に 0 を使用します。

応答

パラメータ	タイプ	一覧	説明
rc	INT8U	レスポンス・コード	レスポンス・コード

レスポンス・コード

コード	説明
RC_OK	コマンドが受け付けられた
RC_INVALID_VALUE	モートのステートが アイドル ではない
RC_BUSY	他のテスト操作が進行中

5.19 testRadioTxExt(0x28)

説明

`testRadioTxExt` コマンドを実行すると、マイクロプロセッサが無線伝送テストを開始できるようになります。このコマンドは、モードがネットワークに参加する前にのみ発行できます。以下の 4 種類の伝送テストがサポートされています。

- パケット伝送
- 連続変調
- 連続波(非変調信号)
- クリア・チャンネル・アセスメント(CCA)を有効にしたパケット伝送(1.4.0 以上の IP モードと 1.1.2 以上の WirelessHART モードで使用可能)

パケット伝送テストでは、モードで `repeatCnt` 個のパケット・シーケンスが生成されます。各シーケンスは最大 10 個のパケットで構成され、サイズと遅延を設定できます。それぞれのパケットは、PHY プリアンブル(5 バイト)、PHY 長フィールド(1 バイト)、データ・ペイロード(最大 125 バイト)、802.15.4 CRC(2 バイト)の順で構成されます。ペイロードのバイト 0 には、送信側のステーション ID が含まれます。バイト 1 および 2 にはパケット番号(ビッグエンディアン形式)が含まれ、この番号は伝送パケットごとにインクリメントされます。バイト 3~N にはカウンタ(0~N-3)が含まれ、このカウンタはペイロード内のバイトごとにインクリメントされます。伝送は、`chanMask` で定義され、疑似ランダムな順序で選択された一連のチャンネルで実行されます。

連続変調テストでは、指定されたチャンネルを中心に、連続した疑似ランダム変調信号がモードで生成されます。モードをリセットすると、テストが停止されます。

連続波テストでは、指定されたチャンネルを中心に、非変調トーンがモードで生成されます。モードをリセットすると、テスト・トーンが停止されます。

CCA を有効にしたパケット伝送テストでは、パケット伝送テストとまったく同じようにデバイスが設定されます。ただし、デバイスは各伝送の前にクリア・チャンネル・アセスメントを実行し、チャンネルがビジーの場合はパケットをアポートします。

`testRadioTxExt` コマンドは、モードがネットワークに参加する前に**アイドル・モード**になっているときにのみ発行できます。無線テストの完了から参加までの間に、モードは(ハードウェア・リセットまたはソフトウェア・リセットで)リセットされる必要があります。

i ステーション ID は 1.4 以上の IP モードおよび 1.1.2 以上の WirelessHART モードで利用可能です。ステーション ID は選択可能な値であり、パケット・テストでこの値を使用することで、同じ無線空間で複数のテストが実行される可能性があるケースでも、受信側がこのデバイスからのパケットを識別できます。このフィールドは、CM または CW テストでは使用しません。[testRadioRX](#) (SmartMesh IP) または [testRadioRxExt](#) (SmartMesh WirelessHART) を参照してください。

⚠ チャンネル番号は 0~15 で、IEEE 2.4GHz チャンネルの 11~26 に相当します。

要求

パラメータ	タイプ	一覧	説明
testType	INT8U	無線テスト・タイプ	伝送テストの種類
chanMask	INT16U	なし	テスト用に有効化されたチャンネル(0~15)のマスク。ビット0がチャンネル0に相当します。連続波および連続変調テストでは、1つのチャンネルのみを有効化してください。
repeatCnt	INT16U	なし	パケット・シーケンスを繰り返す回数(0 = 中断なし)。パケット伝送テストのみに適用されます。
txPower	INT8S	なし	送信出力 (dBm)。有効な値は、0 および 8 です。
seqSize	INT8U	なし	各シーケンスに含まれるパケットの数。このパラメータはパケット・テストのみで使用されます。
sequenceDef	seqDef[]	なし	seqSize シーケンス定義(最大 10)の配列には、各パケットの長さ とパケット伝送後の遅延を指定します。このパラメータはパケット・ テストのみで使用されます。 各シーケンス定義の形式は以下のとおりです。 INT8U pkLen; /* パケットの長さ(2~125 バイト) */ INT16U delay; /* パケット伝送後の遅延(マイクロ秒) */
stationId	INT8U	なし	パケットに含まれる一意の(1~255)識別子このフィールドが省略されている場合は、パケット内のステーション ID に 0 が使用されます。

応答

パラメータ	タイプ	一覧	説明
rc	INT8U	レスポンス・コード	レスポンス・コード


レスポンス・コード

コード	説明
RC_OK	コマンドが受け付けられた
RC_INVALID_STATE	モードが正しい状態でないためにコマンドを受け付けられない
RC_BUSY	モードは他の無線テスト操作を実行している

5.20 zeroize (0x29)

説明

ゼロ化 (zeroize) コマンドは、参加鍵などの設定パラメータの格納に使用されているフラッシュ領域を消去します。このコマンドは、FIPS-140 規格のゼロ化条件を満たすことを意図したものです。コマンドの実行後、モートをリセットする必要があります。1.4.x 以上のモードで使用可能です。

 ゼロ化コマンドを発行すると、モートは機能停止します。モートを使用可能にするには、SPI または JTAG を介して再プログラムする必要があります。

要求

パラメータ	タイプ	一覧	説明
password	INT32U	なし	誤って消去することを防ぐためのパスワード。パスワードは 57005 (0xDEAD) です。

応答

パラメータ	タイプ	一覧	説明
rc	INT8U	レスポンス・コード	レスポンス・コード

レスポンス・コード

コード	説明
RC_OK	コマンドが正しく完了した
RC_ERASE_FAIL	予想外の内部エラーのためにフラッシュを消去できなかった

6 通知

以下のセクションでは、モートがサポートしている通知の API ペイロードについて説明します。API ヘッダ・バイトについては記載していないことに注意してください。

API ヘッダ

API ヘッダ

通知の内容

6.1 advReceived

説明

advReceived 通知は、モートがプロミスキュアス・リッスン・ステート([モートのステートの表を参照](#))のときにアドバタイズメントを受信すると、モートによって生成されます。

ペイロード

パラメータ	タイプ	一覧	説明
netId	INT16U	なし	受信したアドバタイズメントに含まれるネットワーク ID
moteld	INT16U	なし	受信したアドバタイズメントに含まれるソース・アドレス(すなわち、送信者の ID)
rssI	INT8S	なし	受信したアドバタイズメントの受信信号強度表示 (RSSI)
joinPri	INT8U	なし	受信したアドバタイズメントに含まれる参加優先順位 (ホップ深度)

6.2 events

説明

モートは、*events* 通知を送信して、前回の *events* 通知以降に発生した新しいイベントをアプリケーションに通知します。通知には、現在のモート・ステートと保留中のアラームに関する最新情報も含まれます。

ペイロード

パラメータ	タイプ	一覧	説明
events	INT32U	イベント	最近のイベントのビットマップ
state	INT8U	モートのステート	現在のモート・ステート
alarmsList	INT32U	アラーム	現在のアラームのビットマップ

6.3 receive

説明

パケットを受信したことをアプリケーションに通知します。

ペイロード


パラメータ	タイプ	一覧	説明
socketId	INT8U	なし	パケットを受信したソケット ID
srcAddr	IPV6_ADDR	なし	パケットのソース・アドレス
srcPort	INT16U	なし	パケットのソース・ポート
payload	INT8U[]	なし	パケットのペイロード

6.4 timeIndication

説明

timeIndication 通知は、モートへの時間割り込みをサポートするモート製品に適用されます。タイム・パケットには、マネージャを基準とするネットワーク時間と現在の UTC 時刻が含まれます。

LTC5800-IPM ベースの製品では、TIME_n ピンをローに駆動すると(アサート)、プロセッサが起動します。このピンは少なくとも t_{strobe} マイクロ秒の間アサートされる必要があります。このピンをディアサートすると時間がラッチされ、 t_{response} ミリ秒以内に *timeIndication* が生成されます。TIME ピンの詳細については、[LTC5800-IPM Datasheet](#) を参照してください。

 TIME_n ピンがアサートされている間、プロセッサは起動したままになり、電流が流れたままになります。過電流を削減するために、 t_{strobe} 後に TIME_n ピンがアサートされる時間を最小限に抑えるように注意してください。

ペイロード

パラメータ	タイプ	一覧	説明
uptime	INT32U	なし	前回のリセットからの時間(秒)
utcTime	UTC_TIME_L	なし	TIME _n トリガ時の UTC 時刻
asn	ASN	なし	TIME _n トリガ時の時間(スロットの整数値)
asnOffset	INT16U	なし	このスロットの開始からの時間(マイクロ秒)
asnSubOffset	INT16U	なし	報告されるマイクロ秒の開始からの時間(ナノ秒) LTC5800-IPM では、このフィールドの細粒度は 250 ナノ秒です。 このフィールドは、バージョン 1.4.0 以上のモートで使用できません。

6.5 txDone

説明

txDone 通知は、モートがパケットの送信を完了したことをアプリケーションに通知します。この通知は、有効な (0x0000~0xFFFFE) パケット ID を指定して *sendTo* コマンドを呼び出した場合にのみ生成されます。

ペイロード

パラメータ	タイプ	一覧	説明
packetId	INT16U	なし	<i>sendTo</i> 呼び出しで指定されたパケット ID
status	INT8U	送信ステータス	パケットの送信ステータス

7 定義

7.1 コマンド


以下の 2 つの表に、API ヘッダ内のコマンド ID フィールドに対応するモートの API コマンドおよび通知のまとめを示します。モートが正常に動作するには、全てのパケットがアクノリッジされる必要があります。認識されないコマンドまたは通知は、RC_UNKNOWN_CMD レスポンス・コードでアクノリッジされる必要があります。

名前	値	説明
setParameter	0x01	パラメータを設定します
getParameter	0x02	パラメータを取得します
join	0x06	ネットワークに参加します
disconnect	0x07	ネットワークから切断されます
reset	0x08	モートをリセットします
lowPowerSleep	0x09	低消費電力スリープ
testRadioRx	0x0C	無線受信テストを開始します
clearNV	0x10	不揮発性設定ストレージをクリアします
requestService	0x11	サービスを要求します
getServiceInfo	0x12	サービス情報を取得します
openSocket	0x15	UDP ソケットを開きます
closeSocket	0x16	UDP ソケットを閉じます
bindSocket	0x17	UDP ソケットをバインドします
sendTo	0x18	パケットを送信します
search	0x24	プロミスクラス・リッスン・モードに移行します
testRadioTxExt	0x28	無線送信テストを開始します
zeroize	0x29	モートの全ての設定を消去します
socketInfo	0x2B	ソケットのバインド・ステータスを読み出します
blinkPayload	0x2E	ブリンク・ペイロードを書き込みます
stopSearch	0x2F	検索を停止します (IP Stack 1.4 で追加)

7.2 通知

モートは以下の通知タイプを生成できます。

名前	値	説明
timeIndication	0x0D	時間情報
events	0x0F	イベント
receive	0x19	受信パケット
txDone	0x25	伝送完了
advReceived	0x26	受信したアドバタイズメント(プロミスキャス・リッスン・モード)

 モート API では、各通知に固有のコマンド・タイプがあります。イベント通知の最初のフィールドで、イベントの種類を識別します。マネージャ API には通知コマンド・タイプが 1 つ(0x14)しかなく、最初の通知フィールドで通知の種類を識別します。

7.3 パラメータ

以下のパラメータを `setParameter` および `getParameter` コマンドで使用できます。

名前	値	永続的	説明
macAddress	0x01	Y	モートの MAC アドレス
joinKey	0x02	Y	セキュリティ参加鍵
networkId	0x03	Y	ネットワーク識別子
txPower	0x04	Y	送信出力
joinDutyCycle	0x06	Y	参加デューティ・サイクル
eventMask	0x0B		イベント・マスク
moteInfo	0x0C		モート情報
netInfo	0x0D		ネットワーク情報
moteStatus	0x0E		モートのステータス
time	0x0F		時間情報
charge	0x10		電荷量
testRadioRxStats	0x11		無線テストから得られた統計情報
OTAPLockout	0x15	Y	OTAP ロックアウト
moteId	0x17		モート ID (別名: モートの短縮アドレス)
ipv6Address	0x18		モートの IPV6 アドレス
routingMode	0x1D	Y	ルーティング・モード
applInfo	0x1E	Y	アプリケーション情報
powerSrcInfo	0x1F	Y	電源情報
advKey	0x22	Y	MIC 鍵のアドバタイジング
autoJoin	0x24	Y	自動参加モード
antGain	0x29	Y	アンテナ利得 (デフォルトでは 2dBi)
euCompliantMode	0x2A	Y	EN 300 328 準拠
sizeInfoExt	0x2B		拡張されたサイズ情報
entropy	0x2C		エントロピー (IP Stack 1.4 で追加)

7.4 レスポンス・コード

API では以下のレスポンス・コードが有効です。レスポンス・コードの原因については、個々のコマンドの説明を参照してください。

名前	値	説明
RC_OK	0x00	コマンドが正しく完了した
RC_ERROR	0x01	処理エラー
RC_BUSY	0x03	デバイスが現在利用できないために操作を実行できない
RC_INVALID_LEN	0x04	無効な長さ
RC_INVALID_STATE	0x05	無効なステート
RC_UNSUPPORTED	0x06	サポートされないコマンドまたは操作
RC_UNKNOWN_PARAM	0x07	未知のパラメータ
RC_UNKNOWN_CMD	0x08	未知のコマンド
RC_WRITE_FAIL	0x09	永続的なストレージへの書込みに失敗した
RC_READ_FAIL	0x0A	永続的なストレージからの読出しに失敗した
RC_LOW_VOLTAGE	0x0B	低電圧が検出された
RC_NO_RESOURCES	0x0C	リソース不足のため(バッファがないなど)、コマンドの処理に失敗した
RC_INCOMPLETE_JOIN_INFO	0x0D	設定が不完全なため参加を開始できない
RC_NOT_FOUND	0x0E	リソースが見つからない
RC_INVALID_VALUE	0x0F	無効な値が指定された
RC_ACCESS_DENIED	0x10	リソースまたはコマンドへのアクセスが拒否された
RC_ERASE_FAIL	0x12	消去操作に失敗した

7.5 サービスのタイプ

以下の表に、サービスのタイプを示します。

名前	値	説明
bandwidth	0x00	帯域幅タイプのサービス

7.6 サービスのステート

以下の表に、サービスのステートを示します。

名前	値	説明
completed	0x00	サービス要求が完了した(アイドル)
pending	0x01	サービス要求が保留中

7.7 プロトコルのタイプ

以下のプロトコルをパケットの送受信に使用できます。

名前	値	説明
udp	0x00	ユーザ・データグラム・プロトコル(UDP)

7.8 パケット優先順位

以下の表に、パケットの送信に使用される優先順位を示します。

名前	値	説明
low	0x00	低い優先順位
medium	0x01	中位の優先順位
high	0x02	高い優先順位

7.9 モートのステート

以下の表に、モートのステートを示します。

名前	値	説明
init	0x00	初期化中
idle	0x01	アイドル、設定または参加の準備完了
searching	0x02	ネットワークを検索中
negotiating	0x03	参加要求を送信した
connected	0x04	マネージャから少なくとも1つのパケットを受信した
operational	0x05	マネージャによって設定され、データを送信する準備完了
disconnected	0x06	ネットワークから切断された
radiotest	0x07	モートは無線テストモードになっている

promiscuous listen	0x08	モートはプロミスキヤス・リッスン・モードになっており、 <i>search</i> コマンドを受信した
blink	0x09	モートはブリンク・コマンドを受信し、MAC 層の ACK を最大 60 秒間待機(タイムアウト時はローパワー・スリープに遷移)

7.10 イベント

注記: 1 つのイベント通知で複数のイベントが報告されることがあります。

名前	値	説明
boot	0x0001	モートがブートした
alarmChange	0x0002	アラームが開かれるか、または閉じられた
timeChange	0x0004	モート上の UTC 時刻のマップが変更された
joinFail	0x0008	参加操作に失敗した
disconnected	0x0010	モートはネットワークから切断された
operational	0x0020	モートはネットワークに接続され、データを送信できる
svcChange	0x0080	サービスの割り当てが変更された
joinStarted	0x0100	モートはネットワークへの参加を開始した

7.11 アラーム

モートは以下のアラームを宣言できます。

名前	値	説明
nvError	0x01	永続的設定ストレージ(NV)内のエラーを検出した
otpError	0x04	フラッシュ内のキャリブレーションまたは bsp データのエラーを検出した
notReady	0x08	モートは(通常は輻輳が原因で)ネットワークにデータ・パケットを転送できない

7.12 無線テスト・タイプ

`testRadioTx` コマンドを使用して、以下のテストを実行できます。


名前	値	説明
packet	0x00	パケット伝送
cm	0x01	連続変調
cw	0x02	連続波
pkcca	0x03	クリア・チャンネル・アセスメントを使用したパケット・テスト

7.13 パケットの伝送ステータス

パケットの伝送ステータスは、`txDone` 通知の一部として返されます。

名前	値	説明
ok	0x00	パケットはネットワークに送信された
fail	0x01	パケットは廃棄された

商標

Eterna、Mote-on-Chip、SmartMesh IP は、Dust Networks, Inc の商標です。Dust Networks ロゴ、Dust、Dust Networks、SmartMesh は、Dust Networks, Inc の登録商標です。LT、LTC、LTM、 は、アナログ・デバイセズの登録商標です。第三者のブランド名および製品名は各社の商標であり、情報提供のみを目的として使用されています。

著作権

本書は、米国著作権法、国際著作権法、その他の知的財産法および産業財産法によって保護されています。本書はアナログ・デバイセズおよびその実施許諾者によって専有されており、制限付きライセンスに従って配布されます。アナログ・デバイセズの書面による事前の認可なく、本書の全部または一部を使用、複製、変更、逆アSEMBル、逆コンパイル、リバース・エンジニアリング、配布、再配布することは、その形式、手段にかかわらず禁じられています。

制限付き権利: 米国政府による使用、複製、開示は、FAR 52.227-14(g) (2) (6/87) および FAR 52.227-19(6/87)、または DFAR 252.227-7015(b) (6/95) および DFAR 227.7202-3(a) ならびにこれに準ずる法律および規制と後継の法律および規制に規定された制限の対象となります。

免責事項

本書は現状のまま提供され、明示、暗示を問わず一切の保証を行わないものとします。かかる保証には、特定目的に対する商品性または適合性の黙示的保証が含まれますが、これに限定されません。

本書には技術的に不正確な記述またはその他の誤りが含まれる場合があります。訂正と改善は、新しいバージョンの文書に取り入れられる可能性があります。

アナログ・デバイセズは、製品やサービスの適用または使用により発生する責任を負いかねます。また、間接的あるいは偶発的損害を含むがそれに限定されない、いかなる責任も負わないものとします。

アナログ・デバイセズの製品は、誤動作がユーザの深刻な人身傷害につながると合理的に予想できる生命維持装置、デバイス、またはその他のシステムでの使用、またはその機能不全により生命維持装置またはシステムの故障あるいはその安全性や有効性に影響すると合理的に予想できる生命維持装置またはシステムの重要な部品としての使用を目的として設計されていません。このような用途での使用を目的としてこれらの製品を使用または販売しているアナログ・デバイセズの顧客は、顧客自身の責任でそれを行い、このような意図しないまたは不正な使用に関連する人身傷害または死亡に直接または間接的に起因するすべての主張、費用、損害、支出、および妥当な額の弁護士費用、また、かかるクレームでアナログ・デバイセズに該当製品の設計または製造に関わる過失があったと主張される場合でも、これを完全に補償し、アナログ・デバイセズとその役員、従業員、子会社、関連会社、および販売代理店に何ら損害を与えないことに同意するものとします。

アナログ・デバイセズは、いつでも製品またはサービスに対する修正、変更、拡張、改良、その他の変更を行う権利を保有し、製品またはサービスを予告なく中止する権利を有します。顧客は、発注の前に最新の関連情報を入手し、その情報が最新で完全であることを確認する必要があります。すべての製品は、注文承諾時または販売時に提供される、販売に関する Dust Network の契約条件に従い販売されます。

アナログ・デバイセズは、アナログ・デバイセズの製品またはサービスが使用される組み合わせ、マシン、またはプロセスに関連するアナログ・デバイセズの特許、著作権、マスクワーク、その他のアナログ・デバイセズの知的所有権に従って、明示か黙示かにかかわらず、ライセンスが付与されることを保証または主張するものではありません。第

三者の製品またはサービスに関してアナログ・デバイセズが公開した情報は、その製品またはサービスを使用するためのアナログ・デバイセズからのライセンス提供、あるいはその保証または推奨を意味するものではありません。このような情報を使用する場合、第三者の特許または他の知的所有権に従って第三者からのライセンスが必要になるか、またはアナログ・デバイセズの特許または他の知的所有権に従ってアナログ・デバイセズからのライセンスが必要になります。

Dust Networks, Inc は、アナログ・デバイセズの完全所有子会社です。

© Analog Devices, Inc. 2012-2016 All Rights Reserved.