参 考 資 料

SimpliciTI: 簡易モジュール型RF（無線）ネットワークの仕様

Version 1.05
October 2, 2007
Literature No. : JAJA169

TEXAS INSTRUMENTS

内　容

説明図

説明表

1. はじめに

このドキュメントでは、簡易型RF（無線）ネットワーク・ソリューションの仕様を紹介します。このソリューションの目的は、低消費電力、低コスト、低データ・レートのネットワークが目的でワイヤレス・ソリューションを導入する顧客を支援して、ワイヤレス・ネットワーク・サポートの詳しい知識がなくても短時間で製品化を実現できるようにすることです。

このドキュメントでは、小規模かつ低データ・レートのワイヤレス・ネットワーク・ソリューション向けに、簡易型のアプローチの仕様を定めています。

2. 略語と用語

- アクセス・ポイント（AP：アクセス・ポイント）：SimpliciTIデバイス3種類のうちのひとつ。
- アプリケーション・プログラマブル・インターフェイス（API：Application Programming Interface）
- クリア・チャネル・アセスメント（CCA：Clear Channel Assessment）：LBT（Listen-Before-Talk）規律（discipline）の一部。
- 顧客（Customer）：このソリューションの対象となる顧客は、自社製品にSimpliciTIを使用する企業です。
- エンド・デバイス（ED：エンド・デバイス）：SimpliciTIデバイス3種類のうちのひとつ。
- エンド・ユーザー（End User）：顧客の顧客。
- LBT（Listen-Before-Talk）：ネットワークでの無線周波数スペクトルの使用を調停する手段。
- 論理リンク制御（LLC：Logical Link Control）
- メディア・アクセス制御（MAC：Medium Access Control）
- マイクロコントローラ・ユニット（MCU：MicroController Unit）
- ランダム・アクセス・メモリ（RAM：Random Access Memory）
- レンジ・エクステンダ（RE：レンジ・エクステンダ）：SimpliciTIデバイス3種類のうちのひとつ。
- システム・オン・チップ（SoC：System-On-Chip）
- ユーザー・インターフェイス（UI：User Interface）

3. アプリケーション例

次に示すいくつかのアプリケーション例は、SimpliciTIを利用したデバイスに関して市場でどのような需要が見込まれるかということを示しています。

3.1 警報・防犯用アプリケーション

この用途では、ガラス破壊センサ、人感センサ、ドアロック、CO2センサ、光センサ等のデバイスの需要が見込まれます。

3.2 煙感知用アプリケーション

警報・防犯用アプリケーション同様、煙感知用アプリケーションにもひとつの分野として成り立つだけの需要があります。このアプリケーションには連動型の煙感知器も含めることができます。連動型の煙感知器では、どれかひとつのデバイスが作動すると相互接続された全デバイスが作動するようになっているため、警報を効率よく広めることができます。

3.3 自動検針システム（AMR：Automatic Meter Reading）

AMR環境の中には、SimpliciTIの実装に適したものもあります。ガスメーターや水道メーターのように、電気を動力源としないメーターは特に適していると思われます。

3.4 ホーム・オートメーション（HA：Home automation）

このアプリケーションとしては、車庫シャッターの制御、家電製品の制御、居住環境関連デバイスの制御等が考えられます。

4. 要件

4.1 低消費電力

SimpliciTIでは、主電源を取らない低消費電力デバイスをサポートすることが意図されています。

その代表はバッテリ駆動型の低データ・レート・デバイスであり、シンプルで限定的なトポロジーに従って配置されています。

R1. MCUの電源管理は、ネイティブのMCUリソースを利用してアプリケーション・レベルで実現されます。

R2. 無線の電源管理は、関数呼び出しを介してアプリケーションで使用可能になります。（R7を参照）

4.2 低コスト

サポート対象の構成は、「無線MCUソリューション」と「(MCUと無線を統合する)SoCソリューション」の2つです。

R3. コストを低く保つために、対象MCUであるMSP430コアでは次のどちらかの組み合わせが選択されます。

- フラッシュメモリ8K + RAM 512バイト
- フラッシュメモリ4K + RAM 256バイト

どちらを選択するかは、デバイスの種類によって異なります。（R10を参照）

R4. 対象SoC（8051コア）では、フラッシュメモリ16K + RAM 1Kよりも小さい容量の組み合わせが選択されます。

4.3 簡易型の実装と展開

SimpliciTIベースのネットワークは、開発しやすく展開しやすいものでなければなりません。簡易型の開発環境には、開発者が複雑な構成を行わなくても無線通信の細かい手順を実行できるような、簡易型のAPIが必要です。

また簡易型の無線環境では、現場のエンドユーザーがソリューションを実現しやすくするためのアプリケーションを、顧客が容易に開発できる必要もあります。

R5. 無線およびネットワークの構成はカプセル化してアプリケーション層から見えないようにした上で、ビルドタイム構成で駆動します。場合によっては、SmartRF Studio等のツールで補助します。

R6. この環境でのメッセージング用APIセットは、open/read/write/close バラエティに属します。

R7. ioctl() のようなメソッドを使用して、アクセス･ランタイムを構成することも可能です。

R8. アプリケーション･ピア同士をリンクすることは、link()/listenLink() バラエティに属します。

4.4 対象となる無線通信の構成

R9. プロジェクトの対象となる無線通信のコードのリリース順は、次の通りです。

1. CC1100/CC2500（MSP430を使用した、1GHz未満/2.4 GHz無線通信）
2. CC1110/CC2510（8051コアSoC付き、1GHz未満/2.4GHz 無線通信）
3. CC1111/CC2511（8051コア、USB I/F付きSoC)
4. CC2430/CC2420（8051コアSoC付き/8051コアなしの、DSSS無線通信）

4.5 限定ネットワーク･デバイスの種類

ここで説明されるのは「論理デバイス」です。固有の物理デバイスに割り当てることが可能な場合もありますが、そうでない場合もあります。単体の物理デバイスが複数の論理デバイスとして機能することも可能です。

R10. 定義されるデバイスの種類は、エンド･デバイス（必須）、アクセス･ポイント（オプション）、レンジ･エクステンダ（オプション）の3つのみです。

4.5.1 アクセス･ポイント

R11. アクセス･ポイント（AP）は常時接続であり、場合によりバッテリーのバックアップを併用しながら、主電源を取っていることが前提となります。1つのネットワークに許可されるAPは1つだけです。

R12. APには次のような機能があります。ただし、すべての機能がどのネットワークにも当てはまるわけではありません。

1. ネットワークアドレスの管理
2. スリープ中のRxデバイスに代わって行う、蓄積転送型（ストア･アンド･フォワード型）メッセージ

R13. APには、センサ/アクチュエータ（エンド･デバイス）の機能を組み込むことができます。

R14. APには、レンジ･エクステンダの機能を組み込むことができます。

4.5.2 アクセス･ポイント

R15. レンジ･エクステンダ（RE）は常時接続であり、場合によりバッテリーのバックアップを併用しながら、主電源を取っていることが前提となります。このデバイスでは、受信した固有フレームをひとつひとつ再送信します。

R16. レンジ･エクステンダには、センサ/アクチュエータ（エンド･デバイス）の機能を組み込むことができます。

4.5.3 エンド･デバイス

R18. エンド･デバイスは常時接続にすることが可能な場合もありますが、そうでない場合もあります。

R19. エンド･デバイスは RxTx（送受信）デバイスにすることも、Tx（送信）専用デバイスにすることもできます。

R20. 外部的に構成されていない場合は、Tx専用デバイスは次のような働きをします。

1. プリセットされた単一周波数/単一チャネル、またはプリセットされた周波数シーケンス/チャネル･シーケンスの送信を行います。
2. プリセットされた暗号化鍵を使用して、メッセージの暗号化/復号化を行います。

Tx専用デバイスにはスイッチやボタン等の、簡易ネットワーク構成用UIオポチュニティを付けることが可能です。

4.6 トポロジー

R21. 論理メッセージング構成は、次の2種類とします。

1. スター型
2. 直接ピア･ツー･ピア型
   （デバイス･ツー･デバイス型）

4.6.1 スター型

スリープ中のRxデバイスに代わってAPが蓄積転送をサポートしている期間中は、APがスター型ネットワーク･トポロジーでのハブの役割を果たします。すべてのRxデバイスが常時接続である場合、APでは蓄積転送をサポートしませんが、ネットワーク管理とレンジ･エクステンダ 機能はサポート可能です。

4.6.2 ピア･ツー･ピア

常時接続のRx（送信）デバイスでは、場合によりレンジ･エクステンダを介して、発信元デバイスから直接フレームを受信します。このような関係はピア･ツー･ピアとみなされます。ネットワーク上にAPが存在しないこともありますが、その場合はトポロジー全体がピア･ツー･ピアになります。たとえメッセージがすべてブロードキャストで送信されたとしても、アプリケーション側からはどのメッセージもある1つのピアから来るように見えます。

4.7 ルーティング（経路指定）

R22. 明示的なルーティングはサポートされません。

R23. 常時接続ではないRxデバイスでは、場合によりレンジ･エクステンダを介してAPに対するポーリングを行うことでデータを受信できます。MCUと無線の組み合わせで実現するさまざまな低消費電力モードを利用して、割り込み駆動スキームをサポートすることも可能です。

R24. 常時接続のRxデバイスでは、場合によりレンジ･エクステンダを介して、発信元から直接データを受信します。

4.8 セキュリティ

4.8.1 暗号化

R25. ハードウェア暗号化がサポートされる場合は、顧客側でハードウェア暗号化が使用可能になっているものとします。ハードウェア暗号化が使用可能でない場合は、デフォルトのソフトウェア暗号化ソリューションが顧客側で使用可能になっているものとします。

R26. 暗号化鍵の配布は、APがオプションとして提供するネットワーク管理サポートの一部とされます。APが存在しない場合や、ネットワーク･メンバのどれかがTx専用デバイスである場合は、デフォルトのビルドタイム･スキームか、場合によりエンド･ユーザが決めた外部設定を使用して鍵が設定されます。

R27. 暗号化ソリューションでは、対称鍵を使用するものとします。

4.8.2 不正デバイス

R28. 悪意に基づくものかどうかにかかわらず、不正に設置されたデバイスが存在する場合には、それが原因でネットワークが混乱することのないようにする必要があります。フレーム･リプレイ等の形をとる干渉妨害（interference）を防ぐための対策が必要になります。

4.9 メディア･アクセス

R29. デバイスによる送信のためのアクセスは「listen-before-talk」手順で管理されます。この手順は、ETSI（欧州通信規格協会）の仕様に従います。

4.10 メディア･アクセス

周波数アジリティの目的は、ノイズが多い等の原因で特定の周波数が無効の場合に、周波数を変更する手段を提供して堅牢性（ロバストネス）を確保することです。

周波数アジリティは、チャネル移行という形で実現されます。ただし、FCC(連邦通信委員会)Part15の要件に定めた、スペクトル全体へのエネルギー拡散のためにチャネル移行を用いることは意図されていません。周波数が(パケット間でのように)非常な短時間で変化することは意図されていません。

R30. ネットワークでは移行をサポートして、既存の周波数で見られる無線干渉（radio interference）が大きすぎる場合に周波数が変更できるようにします。

4.11 無線パラメータ

R31. 各ネットワークでは、最大250kbpsまでのパラメータをサポートします。※2008年4月現在

R32. サポートされる周波数は、315、433、868、915（1GHz未満）、2.4GHzです。

4.12 送信ループのないレンジ拡張

R33. フレームのコンテンツには、ループを防止するためのヒントとなる情報を含むものとします。

R34. レンジ拡張は、メッセージ発信元～メッセージ宛先間で4ホップに制限されます。

R35. レンジ･エクステンダの数は、ネットワーク1つにつき4個までとします。

R36. レンジ･エクステンダが、他のレンジ･エクステンダから受信したフレームを再送信することはありません。

4.13 バッテリ専用 ネットワーク

R37. バッテリ専用ネットワークはサポート対象となります。つまり、このようなネットワークにはレンジ･エクステンダやAPが存在しません。

4.14 インターオペラビリティ（相互互換性）

R38. SimpliciTIにはそれぞれ異なる機能セットを伴うバージョンがいくつかありますが、各バージョン間では必ずしも相互互換性が必要なわけではありません。リリースされるバージョンによってはサイド･エフェクト（副次的な効果）として相互互換性がサポートされることもありますが、必須要件ではないため保証されることもありません。

5. SimpliciTIの概要

SimpliciTIの目的は、顧客の要望がシンプルな手段を用いた無線メッセージングである場合に、簡易ネットワーク・サポート環境で顧客が行うエンド・ユーザー向け無線デバイス開発をサポートすることです。

プロトコルは、アプリケーション・ピア・ツー・ピアのメッセージング等での使用を目的としています。多くの場合、ピア同士は明示的に相互リンクされています。ただしSimpliciTIでは、デバイス・ペア間の明示的なリンキングが必要でもなく望まれもしないケースも想定されています。たとえば煙感知器で、複数のセンサ/警報器を使用して警報をネットワーク全体に広めることが意図されるようなケースがそうです。

このケースでは、警報器がどれかひとつ作動すると、その警報器によって近くにあるすべての警報器にブロードキャスト・メッセージが送信され、アラーム信号が警報器のネットワーク全体に広められるようにします。この場合は明示的なリンクが発生しないため、顧客側で注意してデフォルトのアクセス・トークンを選択し、ネットワーク以外の発信元からの干渉が起こらないようにする必要があります。

顧客側の開発の立場から見ると、APIを使用すればネットワークを初期化し、デバイス同士をリンクし、メッセージを送信する手段が得られることになります。

5.1 モジュール・コンポーネント

基本スタック自体がサポートするのは簡易型無線メッセージングのみであり、他の機能は含みません。基本スタックには、図1に示すようなモジュール（機能や関数）を様々に組み合わせて追加することができます。

以下、各コンポーネントを簡単に説明します。

5.1.1 基本スタック

5.1.2 暗号化

暗号化の方法として、現時点ではハードウェアかソフトウェアのどちらかを選択できます。CC1100/CC2500の無線通信ではハードウェア暗号化をネイティブでサポートしないために、CC1100/CC2500の無線通信を使用しているプラットフォームでの暗号化はソフトウェア暗号化になります。※

暗号化が有効（enabled）の場合は、アドレス・フィールドと暗号化コンテクスト・フィールドを除くすべてのフィールドが暗号化されます。つまり、あるフレームをリピートすべきかどうかを判断するためにはレンジ・エクステンダでそのフレームを復号化する必要がありますが、これにより、不正デバイスが偽のフレームでネットワークを混乱させてレンジ・エクステンダを不正に利用することを防止できます。

5.1.3 周波数アジリティ

既存の周波数に雑音が多い等の障害がある場合は、この機能によりネットワークが別の周波数に移行することが可能になります。この機能は、ビルドタイムにデータを追加される周波数テーブルで駆動されます。

パケットを受信できるデバイスでは、フレームを送信および再送信した後でも確認応答が受信されない場合は、使用している周波数が正しくないということを検知できます。すると送信側では、確認応答が受信されるまで、周波数テーブルの値を次々に試していきます。このようなことは、ネットワークに参加しようと試みている新しいデバイスと、

※ CC1110、CC1111、CC2510、CC2511はハードウェアにて暗号化をサポートしています。

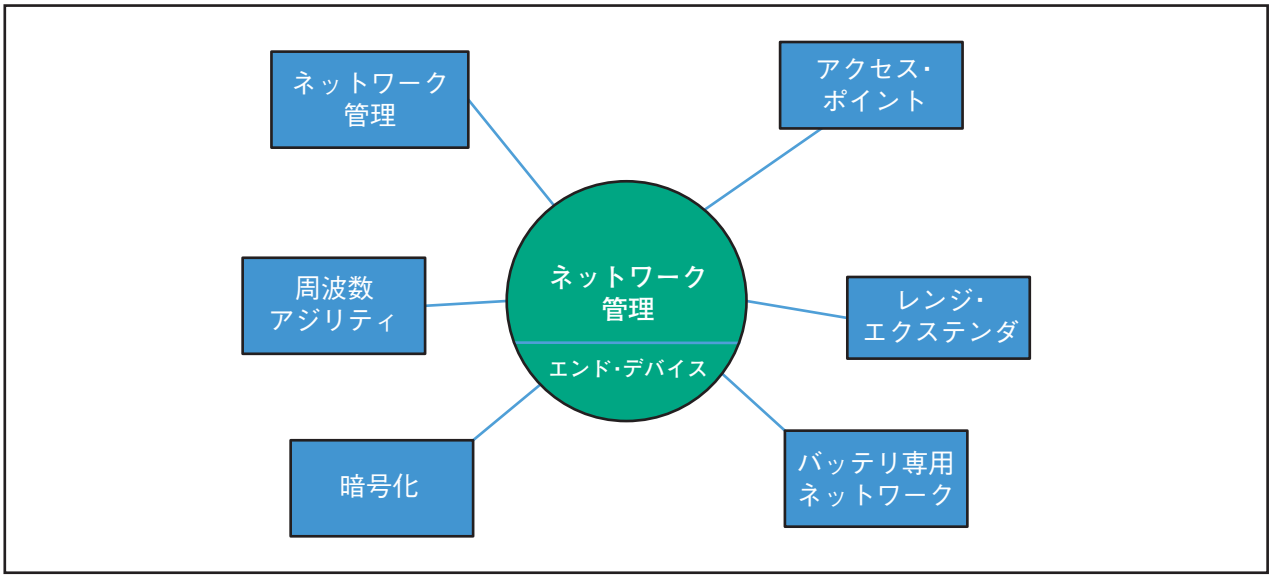

図 1. SimpliciTI のモジュール・コンポーネント

周波数移行があった後でスリープ状態から復帰したデバイスで起こります。周波数の変更は、能動的に行うこともできます。

APのあるネットワークでは、APがブロードキャスト・メッセージで変更をアナウンスします。APのないネットワークではスイッチやボタンといった外部ハードウェアのアナウンスにより、各デバイスに変更を伝えることが可能です。

Tx専用デバイスでは常に、テーブルに記載のあるすべての周波数を使用して送信と再送信を行う必要があります。

周波数の変更が必要であると検知することは、本仕様の範囲外になります。

5.1.4 ネットワーク管理

これは、APの担当領域です。スリープ状態デバイスの蓄積転送機能、暗号化鍵の管理、周波数アジリティ管理といった機能からなります。

5.1.5 アクセス・ポイント

アクセス・ポイントでは、エンド・デバイスをサポートできます。

アクセス・ポイントではフレームをリピートし、フレームをアクセス・ポイント（送信側タイプ）に設定します。（セクション6.4.4参照）。

フレームをリプレイする時は、ホップ数が減らされます。

アクセス・ポイントにより、ネットワーク管理機能が実現されます。

5.1.6 レンジ・エクステンダ

ホップ数によって実現される輻輳防止制限に従って、レンジ・エクステンダでは受信したすべてのパケットをリプレイします。ただし、フレームの宛先がRE自体である場合（例：Ping）や、受信したフレームがレンジ・エクステンダ（送信側タイプ）に設定されている場合（セクション6.4.4参照）は別です。APでもフレームをリプレイしますが、これらのフレームはアクセス・ポイント（送信側タイプ）に設定されています。そのため、レンジ・エクステンダではAPから来るフレームをリプレイすることが許可されています。フレームをリプレイする時は、ホップ数が減らされます。

5.1.7 バッテリ専用ネットワーク

APは常時電源接続されている必要があるため、バッテリ専用ネットワークにはAPが存在しません。バッテリ専用ネットワークでは、すべてのデバイスがスリープ状態デバイスであると仮定されています。蓄積転送機能がないため、フレームの受信は送信側による再送信に依存します。動作中の受信側デバイスの周波数、デバイスがリスン状態である時間の長さ、そして送信側デバイスが再送信に使用する周波数の間で適切なバランスを取る必要があります。

5.2 アーキテクチャの概要

図2に示すように、SimpliciTIソフトウェアでは概念的に3つの層をサポートします。アプリケーション層は、顧客が開発する必要のある唯一の部分です。ネットワークの初期化と構成に使用されるAPIシンボルの簡易セットと、無線でのread/writeメッセージによって通信がサポートされます。

アーキテクチャは、OSI参照モデルに厳密に従ったものではありません。

a. 正式な物理層（PHY）やデータリンク層（MAC/LLC）はありません。データはすでにフレーム化された状態で無線から直接受信されるため、物理層やデータリンク層の機能は無線によって実行されます。

b. セキュリティのサポートはここではネットワーク層のピアとして実装されているため、OSIモデルでこの機能を正式に実装しているプレゼンテーション層はありません。

c. 信頼性のある送信（トランスポート）が必須となる場合は、アプリケーションによって実装される必要があります。したがって、トランスポート層もありません。

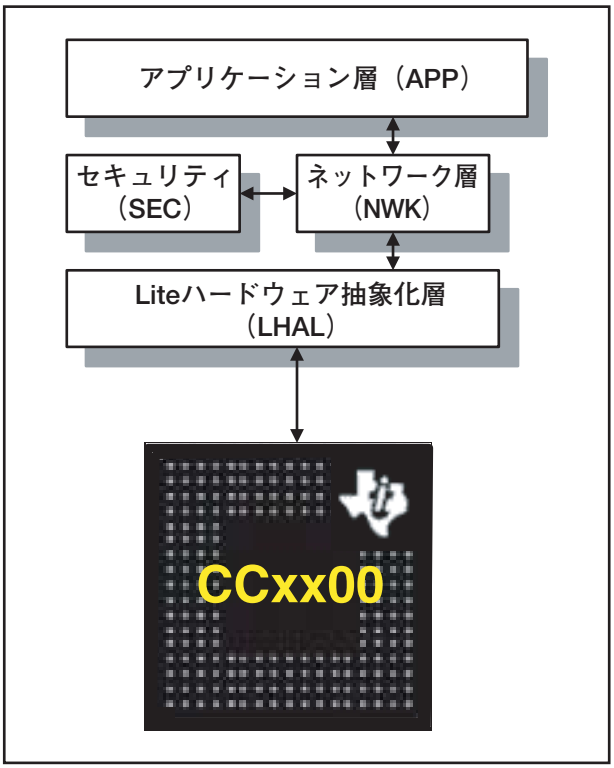

図 2. SimpliciTI のアーキテクチャ

5.2.1 アプリケーション層

顧客は、センサ/アクチュエータによる環境との相互作用の実装としてアプリケーションを開発します。SimpliciTIのAPIを使用すれば、 アプリケーションは、もう一方のデバイスにあるアプリケーション·ピアとの間でメッセージを送信/受信できます。

5.2.2 ネットワーク層

ネットワーク層では、標準的なOSI基本参照モデルと同等の機能を果たします。機能を折りたたんでアプリケーションから見えないようにしているためです。

5.2.2.1 カプセル化されたネットワーク·パラメータ

ネットワークの設定は、ビルドタイム構成スキームによって駆動されます。(TBD) このツールの機能は、SmartRF Studioサポートに似たものになる予定です。これらのスキームには、静的オブジェクト、ヘッダ·ファイル、定数を生成するためのコード生成ツールも含めることができます。これらのうちいくつかのランタイム調整は、ioctl()- のようなインターフェイス経由でAPPからアクセス可能です。

ネットワーク パラメータには次のようなものがあります。

1. 基本周波数と周波数スペーシング
2. サポートされる周波数の数 (周波数アジリティ·テーブル用)
3. 変調方法、データ·レート、その他の一般的な無線パラメータ
4. デフォルト及び生成されたネットワーク 暗号化鍵
5. 保持する蓄積転送メッセージの数
6. クリア·チャネル·アセスメント (CCA：Clear Channel Assessment) のパラメータ
7. ネットワーク ID
8. デバイスのアドレス
9. Tx専用 デバイス上のリピート率
10. 参加トークンとリンク·トークン

5.2.3 セキュリティ

セキュリティは、暗号化/復号化の両方、またフレーム·リプレイ攻撃に対処するためのnonceフィールド等のスキームによってサポートされます。

暗号化と復号化は、ハードウェアによるサポートかソフトウェアによるサポートのどちらかになります。ハードウェアがネイティブでサポートされていない場合は、ソフトウェアによるサポートが提供されます。フットプリントが小さく、パフォーマンス占有を最小限にした暗号化スキームの一例が、XTEA (Extended Tiny Encryption Algorithm) [1]です。

[1] XTEAは64ビットのブロック暗号であるため、フレームにパディングが必要になる可能性があります。パディングにより、オーバーヘッドが少し加わります。

5.2.4 Liteハードウェア抽象化層

Liteハードウェア抽象化層 (LHAL) がサポートするのは、ソフトウェアのネットワーキング機能と無線構成機能のみです。これにより、ネットワーキング機能がMCUのリソースにアクセスできるようになります。HAL全体でアプリケーションをサポートすることが意図されているわけではありません。

LHALがネットワーク層をサポートして抽象化するサービスは、以下の通りです。

1. タイマ·リソース
2. 無線へのSPIインターフェイス
3. 無線からの非同期通知をサポートするための、GPIO コネクション経由の割り込み管理

5.3 トポロジー

下の図は、SimpliciTIがサポートするトポロジーの例です。次に挙げる各エンド·デバイス1 (ED1) では、エンド·デバイス2 (ED2) にデータを送信しています。下の凡例で、図中の様々な形状の線が示すものを示しています。

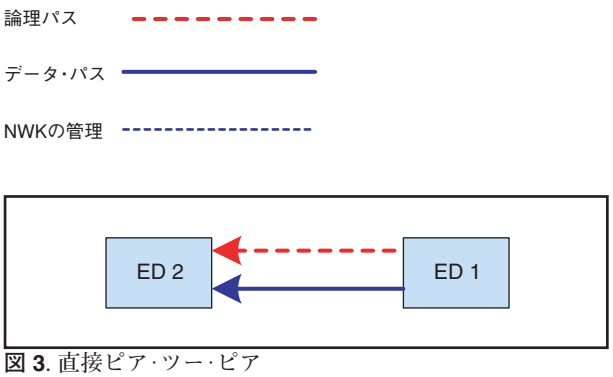

図 3. 直接ピア·ツー·ピア

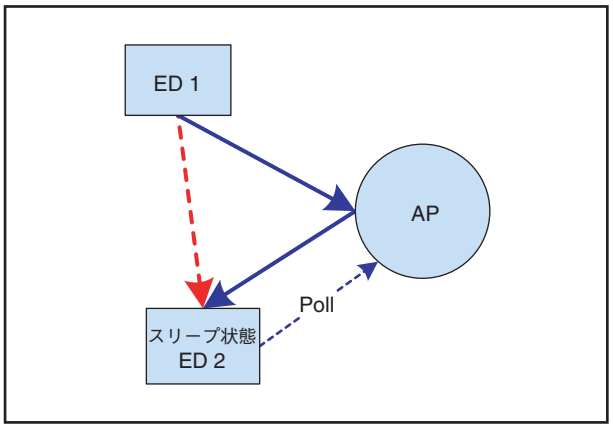

図 4. レンジ·エクステンダ経由の直接ピア·ツー·ピア

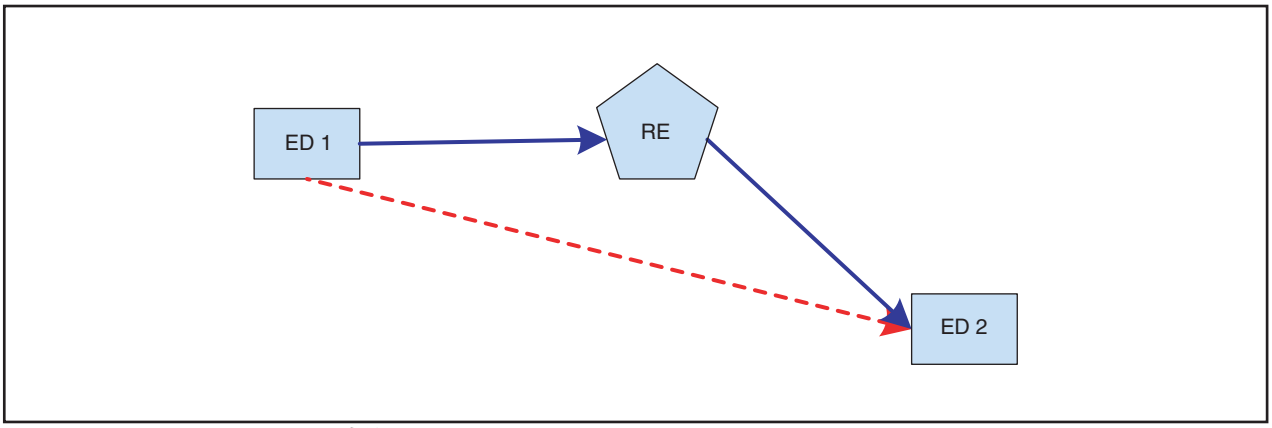

図 5. レンジ·エクステンダ経由の直接ピア·ツー·ピア

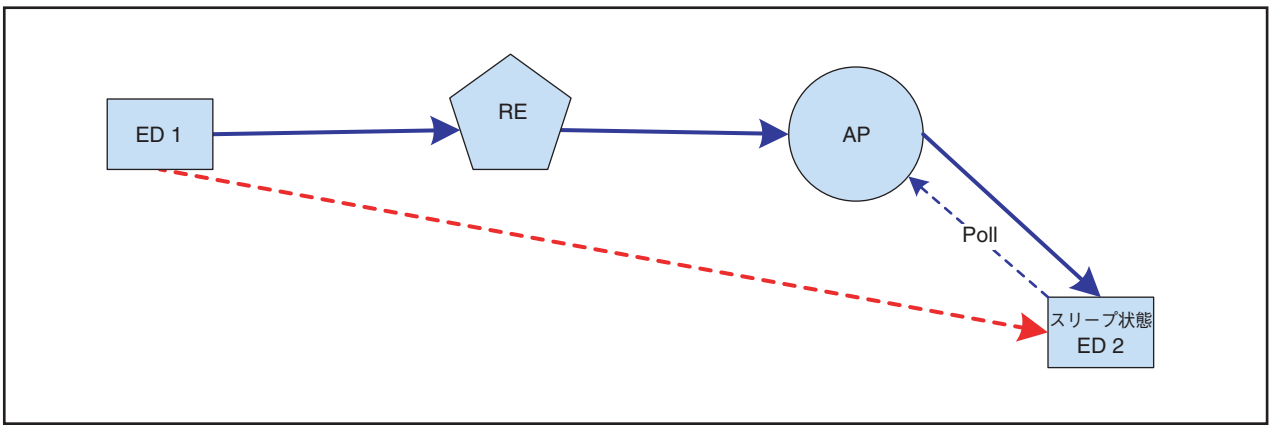

図 6. レンジ·エクステンダとアクセス·ポイント経由の蓄積転送ピア·ツー·ピア

5.4 ネットワークの規律 (discipline)

すべてのフレームは、NWK層のトップにあるアプリケーションで送受信されます。ユーザー·アプリケーションとNWKアプリケーションの両方があります。ユーザー·アプリケーションのフレームは、ピア·ツー·ピアのユーザー·アプリケーション·メッセージです。宛先アドレスやフレーム長などの固定的なネットワーク情報のオーバーヘッド以外では、パケットのペイロードの中身はアプリケーション·データです。

ネットワーク管理は、NWKアプリケーションによって行われます。NWKアプリケーションのフレームも、ピア·ツー·ピアです。ネットワーク管理には、デバイスのリンキング、無線周波数管理、セキュリティ鍵管理、ネットワーク·メンバーシップ管理などが含まれます。詳細については、セクション 6.4.3を参照してください。

フレーム応答は、(ユーザー·アプリケーションかNWKアプリケーションかには関係なく)アプリケーションの担当になります。応答は、アプリケーション·ピア·ツー·ピア規律(discipline)の一部です。したがって、たとえばあるPingフレームへの応答は、応答としてエコーされたPingフレームになります。

ユーザー·アプリケーションはコネクション·ベースです。コネクションは双方向として確立されます(ただしセクション7.2.2.1も参照)。コネクションは、デバイス同士がリンクされたときに確立されます。

NWKアプリケーションはコネクションレス型です。これらの中には応答を実装しているアプリケーションもあれば、実装していないものもあります。詳細については、セクション7を参照してください。

5.5 ネットワーク構成

次に示すのは、初期化呼び出しが行われた場合にAPで行われる手順です(セクション6.4.3にも関連記述があります)。

- テーブルに記載されている全周波数上で1つのFreqフレームをブロードキャストして、ネットワークの周波数をデフォルト周波数に設定します。
- ネットワーク 周波数上で1つのSecurityフレームをブロードキャストして、暗号化鍵を設定します。
- 新しく到着したデバイスによって送信されたJoinフレームをサポートします。
- フレームの受信は可能でも常時接続ではない参加デバイス用に、蓄積転送をサポートします。

5.5.1 アドレス·名前空間の使用

アドレス·名前空間は、0x00と0xFFで終わるもの以外はすべて4バイト単位の値で構成されます。

これらは、ブロードキャスト·アドレスとして予約されます。ブロードキャストが送信または受信された場合には、MSBから3ビットは無視できます。

5.6 電源管理

SimpliciTIの簡易型RF（無線）ネットワークでは、低消費電力デバイスをサポートすることが意図されています。デュアル·チップのソリューションでは、無線段の電源とMCU段の電源を分けて管理できます。

デュアル·チップ·ソリューションでは、アプリケーションがMCUの電源管理を担当します。

無線電源管理は、APIで提供されるioctl()のようなインターフェイスを使用してMCU電源管理から導出できます。（セクション8.6を参照）.

5.7 Tx専用 デバイス

送信専用デバイスの動作は、次のようなものです。

- デフォルトのネットワーク鍵を使用します。ただし、たとえば外部スイッチのように、スタックが鍵を検知して実装できるような手段を顧客が提供する場合は、他の鍵を使うこともあります。
- 再送信を行うことにより、パケットの伝播を確実にします。応答の受信ができないためです。
- Tx専用デバイス上で周波数の移行を処理する方法は2通りあります。
  - テーブルに記載された全周波数上の各フレームを送信します。応答の受信ができないためです。
  - スイッチやボタンのような外部介入によって周波数を変更します。
- Tx専用デバイスが他のデバイスにリンクする場合は、リンク·トークンが有効である必要があります。
- 他のネットワーク·インフラストラクチャではTx専用デバイスから来たフレームに対する応答を行いません。

6. フレーム構成

フレームは、次の3つの論理部から構成されます。

- PHY/MAC層で処理される部分
- NWK層に実装されるネットワーク管理をサポートする部分
- APP層でサポートされるアプリケーションのペイロードを表す部分

6.1 PHY/MACフレーム部

この部分には、ハードウェアによって処理されるフレームの部分が含まれます。CC1110、CC1111、CC2510、CC2511の無線通信では、これはプリアンブル·ビットと同期(sync)ビットから成ります。

6.2 NWKフレーム部

フレームのこの部分は、NWK層のファームウェアによって処理されます。この部分のフィールドはネットワーク制御用であり、フレーム·タイプ、 暗号化ステータス、ホップ数、シーケンス数等のパラメータを指定します。

6.3 APPフレーム部

これは、ユーザー·アプリケーションに向けた受信API呼び出しの結果としてアプリケーションに送信される、カプセル化されたアプリケーション·データの ペイロードです。NWKアプリケーションの場合、フレームのアプリケーション部は受信スレッドの一部として処理されます。

6.4 フレームの詳細

図7に、一般的なフレームのレイアウトを示します。詳細な説明が図の後に続きます。

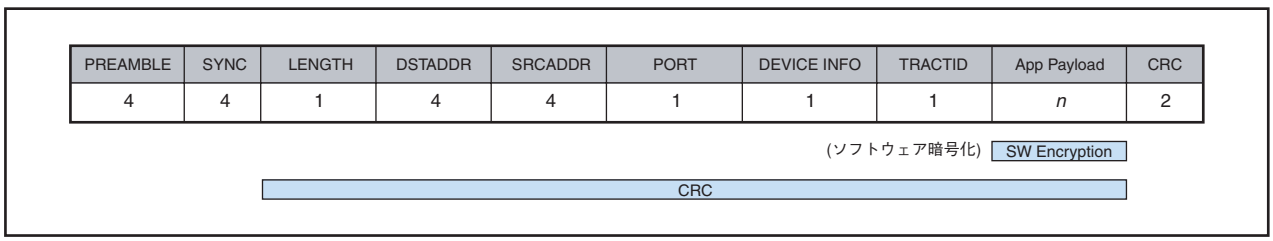

図 7. SimpliciTI のフレーム構造

フィールド	定義	コメント
PREAMBLE	無線同期	無線ハードウェアによって挿入
SYNC	無線同期	無線ハードウェアによって挿入
LENGTH	残存パケットのパケット長 （バイト単位）	Txでファームウェアによって挿入。 Rxで部分的にフィルタリング可能。
DSTADDR	宛先アドレス （Destination Address）	ファームウェアによって挿入。LSBでフィルタリング可能。0x00と0xFFのLSB値は、ブロードキャストとして予約する。
SRCADDR	発信元アドレス（Source Address）	ファームウェアによって挿入
PORT	暗号化コンテクスト（7-6） アプリケーションのポート番号 （ビット 5-0）	ファームウェアによって挿入。ポートの名前空間では0x20～0x3Fを顧客アプリケーション用、0～1FをNWK管理用に予約する。
DEVICE INFO	送信側/受信側と、 プラットフォームの機能	ファームウェアによって挿入。詳細は後述。
TRACTID	トランザクションID	ファームウェアによって挿入。規律（discipline）はコンテクストに依存する。nonceとして使用することも可能。連続的（シーケンシャル）である必要はない。
APP PAYLOAD	アプリケーション・データ	0 ≤ n ≤ 52（APPEND_STATUS無線オプションを使用の場合は50） 暗号化が使用可能な場合は、0 ≤ n ≤ 48
CRC	巡回冗長検査バイト （Cyclic Redundancy Check bytes）	APPEND_STATUS無線オプション使用の場合、CRCはハードウェアによって計算されてTxで挿入され、またハードウェアによって計算されてRx上で削除（strip）される。検査が失敗した場合は、フレームは破棄される。

表 1. SimpliciTI フレーム・フィールドの要約

以下は、詳細説明です。

6.4.1 PREAMBLE/SYNC/LENGTH

プリアンブル・フィールドと同期（sync）フィールドは、無線ハードウェアによって挿入されます。プリアンブルのバイト数は構成可能であり、フィールド試行に基づいて4に設定されます。

同期（sync）ワードの値も構成可能です。最大受信機能用に構成されているため、デフォルト値が使用されます。

無線は、可変長フレーム[2]の送信と受信用に構成されます。そのように構成した場合、最後の同期（sync）バイトに続くバイトがフレームの長さ（長さのバイト自体は含みません）になるものとされます。

[2] フレームは≤64バイトである必要があります。したがって、送信FIFOまたは受信FIFOよりも大きいフレームはありません。

6.4.2 DSTADDR/SRCADDR

宛先アドレスと発信元アドレスは、それぞれリトル・エンディアン形式に従います。宛先アドレスのLSBは、Proprietary無線上でフィルタリング可能な位置にあります。このバイトにある、0x00と0xFFの付いたデバイス・アドレスは、これらの無線によってブロードキャスト・アドレスとしてフィルタリングされるため、アドレス・名前空間からは除外されます。LSBが示す通りにブロードキャストを受信する場合は、残りのアドレス・ビットを無視できます。

6.4.3 セキュリティ・コンテクスト/PORT

ビット7-6では、下の表に定義されたセキュリティ・コンテクストを保持します。

ポートは、フレームを処理している対象アプリケーションを指定する概念的な抽象化（conceptual abstractions）です。ポート番号0x00～0x1Fは予約されるか、特定のサービスを使用して「ウェルノン・ポート（よく知られているポート）」として割り当てられます。

ビット	説明	コメント
7-6	00：暗号化なし 01：ソフトウェア暗号化 10：ハードウェア暗号化 11：予約	デフォルトは「暗号化なし」です。
5-0	ポート番号	フレームの宛先となるアプリケーションを示す抽象化です。

表 2. セキュリティ・コンテクストとポート番号

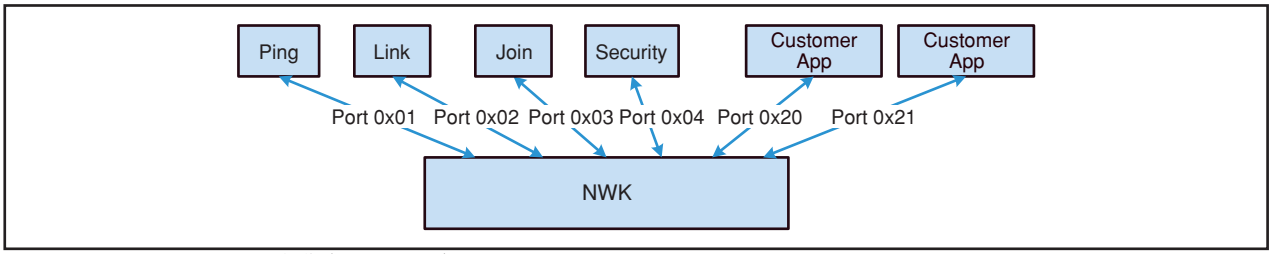

図 8. SimpliciTI のポート抽象化（abstraction）

ポート番号は、ネットワーク管理に使用されることを意図したものです。

ポート番号0x20～0x3Fは、ユーザー側のハンドラ（処理プログラム）に対して割り当てられています。ポート0x3Fは、送信側のデバイスが明示的に受信側デバイスにリンクされていない場合に[3]使用されるブロードキャスト・ポートとして予約されます。ポート0x3Eは、Tx専用デバイスからの送信の対象となるポートとして予約されます。リンク-リスン（link-listen）が成功した場合、ポートは顧客のアプリケーションに提供されたハンドルとみなされます。これにより、アプリケーションではメッセージが送受信される際にこのハンドルを使用します。

ポート抽象化（port abstraction）の概念的な表現（conceptual representation）を、図8に示します。ただし、すべてのウェルノン・ポートが記載されているわけではありません。

6.4.4 DEVICE INFO

デバイス情報バイトでは、フレームを発行しているデバイスの情報を提供します。

デバイス情報ビットは表3のように定義されます。

6.4.5 TRACTID

トランザクションIDは、通信にロバストネス（堅牢性）を付加するためにNWKによって使用されます。このフィールドは、たとえば未解決メッセージへの応答の整合を取ったり、重複フレームの識別を補助したりするために使用できます。各送信側では、それぞれ独自のトランザクションID 規律（discipline）を保持しています。

7. NWKアプリケーション

ウェルノン・ポート上にあるネットワーク層のアプリケーションは、ネットワークの管理に使用されるピア・ツー・ピア・オブジェクトです。必須のアプリケーションもありますが、ある決まった状況下で必須になるという理由で、条件付きのアプリケーションとされるものもあります。リストを下に記載します。条件付きアプリケーションのサポートが必要となる条件に、コメントがついています。

アプリケーション	ポート	デバイスのサポート
Ping	0x01	Tx専用 デバイス以外では必須
Link（リンク）	0x02	必須
Join（参加）	0x03	ネットワークがAPをサポートしている場合のみ必要
Security（セキュリティ）	0x04	ネットワークがAPをサポートしている場合のみ必要
Freq（周波数）	0x05	ネットワークがAPをサポートしている場合のみ必要
Mgmt（管理）	0x06	汎用NWK管理アプリケーション。ポーリングのポートとして使用。

表 4. NWKのアプリケーション

[3]顧客の必要に応じて、明示的なリンキングを、ランタイム同様にビルドタイムに行うこともできます。

ビット	説明	コメント
7-6	00：常時リスン 01：スリープ/ポーリング 10：スリープ/リスン 11：リスンなし（Tx専用）	受信側（Rx）タイプ スリープ状態デバイスは、ポーリングまたはリスンを行う可能性があります。ポーリング中のスリープ状態デバイスは、アクセス・ポイントに対して蓄積転送機能を要求します。リスン中のスリープ状態デバイスは、場合により任意の送信側によるデバイスへの自動再送信を必要とする、WOR-のようなソリューションを使用します。受信側タイプに相当するのは、フレーム内に発信元アドレスが指定されているデバイスです。
5-4	00：エンド・デバイス 01：レンジ・エクステンダ 10：アクセス・ポイント 11：予約	送信側（Tx）タイプ レンジ・エクステンダ（RE）にとって最も重要なのは、あるREが他のREから来たフレームを転送しないようにすることです。これにより、ブロードキャストの混乱を軽減することができます。送信側タイプに該当するのはフレームの送信を行うデバイスであり、フレーム内に発信元アドレスが指定されているデバイスとは区別する必要があります。
3	予約	
2-0	ホップ数	各送信側デバイスによって0になるまで減らされ、その後破棄されます。

表 3. DEVICE INFO ビットの値

NWKアプリケーションのペイロード内の最初のバイトは、情報バイトです。このバイトはアプリケーションごとに特有の情報を搬送します。例えば長さの情報や、フレームの受信に対して確認応答を行うかどうかといったオプションの情報などです。さらに、各アプリケーションは独自のペイロード規律(discipline)を備えています。ブロック暗号による暗号化が使用されている場合には、フレーム長からペイロード長を推定することができないため、長さの情報は事実上必須になります。

ペイロードについては、この後に説明を記載します。

7.1 Ping(ポート 0x01)

7.1.1 セマンティクス(動作)

このアプリケーションの目的は、特定のデバイスの存在を検知することです。

このアプリケーションは、あるデバイスから別のデバイスへメッセージを送信します。 受信側デバイスでは、ペイロードの内容を発信元にそのまま返します。発信元が応答を待機するという意味で、このシーケンスはブロッキングです。タイムアウト·パラメータが適用されます。

Pingは、ユニキャストで送信されます。

7.1.2 ペイロード(クライアントおよびサーバー)

フレームのLENGTHフィールド:12+n

アドレス:ユニキャスト

アプリケーション情報(長さ)	データ
0 ≤ n ≤ 51	nバイト

図 9. Ping のペイロード

同じペイロードが、送信と受信両方で使用されます。

7.2 Link(ポート 0x02)

7.2.1 セマンティクス(動作)

このアプリケーションの目的は、2つのデバイスのリンキングをサポートすることです。

フレームの発信元デバイス(クライアント)は、宛先デバイス(サーバー)にリンクされます。一度リンクが確立されると、クライアントではリンクされたデバイスへメッセージを送信することが可能になります。

クライアントのリンク·メッセージはブロードキャストで送信されます。サーバーの応答はユニキャストでクライアントに返信されます。サーバーでは、クライアントのブロードキャスト·メッセージを利用してクライアントのアドレスを記憶します。クライアントでは、ユニキャストの応答を利用してサーバーのアドレスを記憶します。

リンクを確立するためには、完全な交換(exchange)が必要となります。この機能が完了すると、2つのデバイス間での双方向コネクションが可能になります。

クライアント側から見ると、クライアントが応答を待機するという意味で、このシーケンスはブロッキングです。タイムアウト·パラメータが適用されます。サーバー側から見ると、リスン呼び出し(listen call)もブロッキングとなります。

Tx専用デバイスは上記とは異なる方法で処理されます。Tx専用 デバイスの場合、サーバーの応答は必要ありません。また、固定サーバー·アドレスが実装されていない限り、クライアントのメッセージは常にブロードキャストで送信されます。

この簡易プロトコルでは、リスン(listen)あたり1つのリンクのみが許可されます。リンクは先着順となります。

おそらく、クライアントとサーバーはユーザーによる何らかの介入(ボタン押下など)に基づくLink交換に関与することになります。 ただし、これは必須ではありません。

複数のLinkセッションを行うと、複数のデバイスをリンクさせることができます。各クライアントには、一意のアクセス·トークンがひとつずつ提供されます。また、別々のLinkセッションを促す(instigate)ことにより、1つのクライアントが複数のサーバーに再度データを送信することも可能です。

7.2.2 ペイロード

7.2.2.1 クライアント側

フレームのLENGTHフィールド:19

アドレス:ブロードキャスト

アプリケーション情報	リンク·トークン	ローカル·ポート	リンク番号	Rxタイプ
1	4	1	1	1

図 10. クライアント側のLinkペイロード

ペイロードは、4バイト単位のリンク·トークン、ローカル·ポート番号、リンク番号、クライアントのRxタイプ(セクション6.4.4参照)から構成されます。トークンは、リンク·コンテクストを一意に特定するために使用され、(悪意のあるなしにかかわらず)不正なデバイスによるネットワークへのリンキングを防止するために使用される必要があります。顧客側では、任意の方法でこれらのトークンを一意に設定できます。トークンを一意に設定することは、あるデバイスが正しくないネットワーク上の別のデバイスにリンクしないようにするための手段です。トークンは、ネットワーク全体のパラメータです。セクション10.1.1を参照してください。

ローカル·ポート番号は、リンクしているデバイスに返信するためにリモート·デバイスが使用する必要のあるポート番号です。リンクしているデバイスがTx専用であるか、このコネクション上でどのメッセージも受信する必要がない場合には、値0が供給されます。この場合、コネクションは単一指向性(一方通行)となります。

リンク番号は、重複をフィルタリングで除去するためにリンクのリスニング･アプリケーションによって使用されます。クライアントでは、ひとつのリンク番号を二度使うことはできません。

Rxタイプはリスナーによって、クライアントにフレームを送信する際のホップ数設定のヒントとなる情報として使用されます。Rxタイプが01b(スリープ/ポーリング)の場合、クライアントへの送信ではアクセス･ポイントのみに到達するのには十分な大きさのホップ数を使用できます。そうでない場合は、受信されたフレームから推定できるリスナーまでの到達のためにクライアントへの送信が要するホップの数を、ホップ数として設定できます。

APのあるトポロジーでは、参加しているデバイスに対してAPがリンク･トークンを供給します。APのないトポロジーやTx専用 デバイスでは、デフォルトのリンク･トークンが使用されます。

7.2.2.2 サーバー側

フレームのLENGTHフィールド：14
アドレス：ユニキャスト

アプリケーション情報	アクセス･トークン	Rxタイプ
1	1(00xxxxxxb)	1

図 11. サーバー側のLink応答ペイロード

応答ペイロードには、サーバー上で現在リンク中の(now-linked)アプリケーションにアクセスするために、クライアント側がすべてのアプリケーション･メッセージで使用するアクセス･トークンが含まれています。このトークンはそれぞれの側のNWK層によって保守されます。また、このトークンにはサーバー側のRxタイプも含まれています。

トークンは、クライアント側のリンクされたアプリケーションがサーバー上のピアにメッセージを送信する時に、フレームにあるPORTフィールドにデータを追加するために使用されます。接続関係(association)は、NWK層で維持されます。

Rxタイプはクライアント側と同じ目的で、つまりサーバーに送信されるフレームでのホップ数を設定するためのヒントとなる情報として使用されます。

Tx専用 デバイスは、(DEVICE INFO のビットを使用している)サーバーによってTx専用 デバイスとして認識されるため、応答を送信することはありません。Tx専用 デバイスでは、有効なリンク･トークンを送信する必要があります。ただし、応答を受信することはできないため、アプリケーション･ピア･ツー･ピアのメッセージを送信する際にはデフォルトのアクセス･トークン(0x3E)を使用します。

7.3 Join (ポート 0x03)

7.3.1 セマンティクス (動作)

このアプリケーションでは、 APのあるトポロジーのネットワークへの参加(エントリ)を保護します。

APのあるネットワークの場合、各デバイスはネットワークに参加しなければなりません。参加すると、デバイスには次のものがオプションとして供給されます。

- 暗号化コンテクスト
- リンク･トークン
- 蓄積転送のサポート(スリープ状態のRxデバイスの場合)

最初の3つのアイテムを供給することにより、不正デバイスの侵入に対してのネットワークの堅牢性(ロバストネス)が強化されます。

参加の試みは、APがまだ起動(up)していない場合にはタイムアウトする必要があります。参加の試みを継続するかどうかは、顧客のアプリケーションで判断します。APの存在はビルドタイム･パラメータであるため、どのような挙動をすべきかはすべてのデバイスに認識されています。

7.3.1.1 クライアント側

フレームのLENGTHフィールド：17
アドレス：ブロードキャスト

アプリケーション情報	参加トークン	コネクション数
1	4	1

図 12. JOIN クライアント側のペイロード

トークンは、ネットワーク上でデバイスが有効になっているかどうかを検証するために使用されます。

7.3.1.2 サーバー側

フレームのLENGTHフィールド：17 + n
アドレス：ユニキャスト

アプリケーション情報	リンク･トークン	機能/長さ	鍵
1	4	1	n

図 13. Joinサーバー側のペイロード

サーバーは、クライアントがネットワーク用のリンク･トークンとして使用する必要のある値を返します。

暗号化コンテクストも返されます。次のバイトは、セキュリティのアプリケーションでの形式(フォーマット)と同じ形式を使用します。

7.4 Security（ポート 0x04）

7.4.1 セマンティクス（動作）

このアプリケーションは、暗号化鍵や暗号化コンテクスト等のセキュリティ情報を変更するために使用されます。このアプリケーションのために、APでは交換（exchange）を初期化して、アプリケーションがAPのあるネットワークのみ、さらにTx専用以外のエンド･デバイスとレンジ･エクステンダのみで見えるようにします。

APにこれらの交換を初期化させることは、APに管理されるスリープ状態Rxデバイスにとっても役立ちます。スリープ状態Rxデバイスが同期から外れることがなくなるためです。APでは複数の暗号化コンテクストを保持しており、スリープ状態デバイスが復帰してポーリングを行う時に警告を出すことができます。

APではこのアプリケーションを使用して暗号化コンテクストを実行し、他の全てのデバイスに新しい鍵を通達します。

これらのフレームは常に、デフォルト暗号化鍵で暗号化してから送信されます。

7.4.1.1 イニシエータ（AP）側

フレームのLENGTHフィールド：12 + n

アドレス：ブロードキャスト

アプリケーション情報 機能/長さ	鍵
1	n

図 14. Securityイニシエータ（AP）のペイロード

7.5 Freq（ポート 0x05）

7.5.1 セマンティクス（動作）

ネットワークにAPがある場合、このアプリケーションでは周波数アジリティをサポートします。APでは、周波数変更コンテクストをアナウンスしているFreqポートへブロードキャスト･フレームを送信することで、周波数の変更を初期化します。

ネットワークの周波数が変更されると、スリープ状態デバイスは復帰した時点で、周波数テーブルの値を次々に試してAPを探します。そうする必要のあることをスリープ状態デバイスが認識できるのは、ポーリングを行ってもACKが帰ってこないためです。

7.5.1.1 イニシエータ（AP）側

フレームのLENGTHフィールド：13

アドレス：ブロードキャスト

アプリケーション情報	周波数インデックス
1	1

図 15. FREQイニシエータのペイロード

ペイロードは、新しい無線周波数の周波数テーブル･インデックスからなります。

フィールド	定義	説明
FUNC/LEN	セキュリティ･コンテクスト（ビット7-5）/ キー･フィールド長（ビット4-0）	FUNC x00b：暗号化オフ x01b：ソフトウェア 暗号化オン x10b：ハードウェア 暗号化オン 0xxb：新しい鍵の添付なし 1xxb：新しい鍵の添付あり LEN：FUNCビットが1xxbの場合は、添付された鍵の長さ（バイト単位）
KEY	暗号化/復号化鍵	FUNCビットが0xxbの場合、フィールドはヌル

表 5. Securityイニシエータのペイロードの詳細

7.6 Mgmt（ポート 0x06）

7.6.1 セマンティクス（動作）

これは、デバイスを管理するために使用する一般管理ポートです。帯域外デバイスにアクセスするために使用され、また他のポートのステート･マシンやトランザクションID等をリセットするために使用することも可能です。さらに、スリープ状態デバイスがAPに対してポーリングを行うためにも使用されます。

これは、スリープ状態/ポーリング状態のエンド･ デバイスが、フレームを探すためにAPに対してポーリングを行う目的で使用されるポートです。

これは、緊急用の meet-meポートとしても使用できます。このポートには、メッセージをクリアテキストでいつでも送信できます。

7.6.1.1 エンド･デバイスのポーリング

エンド･デバイスはネットワークに参加した時点で、メッセージを探すためにアクセス･ポイントにポーリングを行う意図があることをアナウンスします。このコンテクストはDEVICE INFOバイトで搬送されます（6.4.4参照）。ポーリング･クエリがMgmtポートで送信されます。サーバー側の応答は、クエリの対象となるポートで送信されます。注意する必要があるのは、このクエリが受信呼び出しのサイド･エフェクトとして送信されるものであり、SimpliciTI APIの一部ではないということです。

7.6.1.1.1 クライアント側

フレームのLENGTHフィールド：13

アドレス：アクセス･ポイント

アプリケーション情報	ポート
0x01	1

図 16. エンド･デバイスのポーリング

アクセス･ポイントのクエリは特定のポート用に作成されます。ポートは、受信呼び出しで供給するリンクIDから導出されます。

7.6.1.1.2 サーバー側

サーバー（アクセス･ポイント）からの応答はMgmtポートにはありません。サーバーは、クエリの対象となるポートで応答を行います。

クライアントを待機中のフレームがクエリ対象のポートにある場合には、サーバーはフレームを送信します。ポート上に待機中のフレームがない場合、サーバーはペイロードのないフレームを構築して、クライアント上のそのポートに送信します。

8. API

APIの目的は、アプリケーション側の手間をほとんどかけずに機能的で信頼性のあるネットワークを形成できるような方法で、ネットワーク機能をカプセル化することです。このアプローチの主な影響は、結果としてできるネットワークでは簡易性のために柔軟性が犠牲になることです。

APIによって、次の機能がサポートされます。

- 初期化
- リンキング
- アプリケーション･ピア･ツー･ピアのメッセージング
- デバイス管理

NWK層では、（すべてではないにしても）ほとんどの交換が応答を持っています。APP層では、応答ステータスがAPPのピア･ツー･ピアで制御されます。

8.1 smplStatus_t SMPL_Init (uint8 (*pCB)(linkID))

この呼び出しを使用すると、すべてのNWKの初期化が行われます。初期化のサイド･エフェクトのひとつは、ネットワークへの参加の試みです。アクセス･ポイントのサポートするネットワークの場合は、Joinリクエスト（サイレント発行）がアクセス･ポイントと引き換えにネットワーク･パラメータを獲得します。

またこの呼び出しは、セクション10で定義されるシステム･パラメータに基づいて、無線通信の設定やすべてのNWKコンストラクトの初期化を行います。デバイスがAPの場合は、暗号化鍵やリンク･トークンの生成といった他の処理も行われます。

引き数は、リンクID引き数を取ってuint8を返す関数へのポインタです。この引数により、アプリケーションがコールバック関数ポインタを、受信されたフレームのロジックを処理するフレームに提供することが可能になります。次に示す条件が満たされれば、受信されたフレームのリンクIDとともにコールバックが引き数として呼び出されます。

1. 供給された関数ポインタがヌルではない。
2. 受信されたフレームの宛先ポートが、ユーザー側のポートである。
3. フレームの宛先アドレスが、デバイスの宛先アドレスである。
4. コネクションが有効である（つまり、ポートがコネクション･テーブルに記載されている）。

関数がゼロ以外の値を返すと、フレームの処理が完了したとみなされ、フレームのリソースが解放されて再利用できるようになります。それ以外の場合、フレームは入力フレーム･キューに残り、アプリケーションの処理を待機します。

アプリケーションではどちらの場合にも、コールバック引き数に含まれるリンクIDをリンクID引き数としてSMPL_Receive()への呼び出しを実行し、受信フレームを検索します。このような条件でのこの呼び出しは、必ず成功するようになっています。

デバイスがアクセス·ポイントであり、そのAPがデータ·ハブとして設計されていれば、参加しているデバイスがエンド·デバイス·オブジェクトをサポートしている場合にもコールバックが呼び出されます。リンクIDでは0x00という値を持ちます。この場合にはNWK層で受信フレームが処理されるため、アプリケーションで受信フレームを直接検索することはできません。コールバックは、参加デバイスから次のリンク·フレームを受信するためにSMPL_LinkListen()を実行する必要があることを、APへ伝えるための信号として使用されます。

コールバックの各条件が満たされれば、ハンドラ(処理プログラム)が受信ISRコンテクスト内で呼び出されることに注意してください。設計者は、このスレッドでの処理が多くなりすぎないように注意する必要があります。

8.2 smplStatus_t SMPL _Link (linkID_t *linkID)

別のリスン状態デバイス(サーバー)クライアントとしてリンクします。

リスン状態のデバイスは、応答を返します。一度応答が受信されると、リンクIDとともに呼び出しが返ります。このリンクIDは、リンクの設定されたデバイスで送受信される、後続のすべてのメッセージングで使用される必要があります。これはブロッキング呼び出しではなく、リンクが成功したかどうかを示すステータスを返します。成功しなかった場合は、アプリケーションから再試行することが可能です。

8.3 smplStatus_t SMPL _LinkListen (linkID_t *linkID)

これはリンク呼び出しに伴うものであり、クライアントのリンク·メッセージを待機します。

この呼び出しの結果、最初の有効なリンク·メッセージのみが受信されます。1つの未解決リスンのみが、任意の時間で有効になります。リスンが実行される前に到着したリンク·メッセージは破棄されます。

リスン中に、重複したリンク·フレームを受信しないようにするための方策はあります。この方策は、再送信されたフレームや送信の遅れたフレームを検知する仕組みになっています。ただし、リセットされたデバイスが毎回同じリンク·フレームを生成する場合には、リセットされたデバイスから来るリンク·フレームも検知します。例えば、毎回ランダムに新規デバイス·アドレスを生成するデバイスは、重複したリンキング·デバイスとしては検出されず、リソースを消費します。

これはブロッキング·コールであり、有効なリンク·フレームが受信されるまでは返りません。

8.4 smplStatus_t SMPL _Send (linkID_t lid, uint8 *msg, uint8 len)

この呼び出しは、長さlenのメッセージmsgを、リンクID lidとともにデバイスへ送信します。複数の送信呼び出しが同時に未解決になる可能性があります。lidは、SMPL_Link()呼び出しによって返されたのと同じlidです。

この呼び出しは、メッセージが送信されたあとで返ります。これは、MCUと無線の送信コンテクストと電源コンテクストに関して規律(discipline)が保たれるようにするためです。

メッセージ·サイズは最大52バイトに制限されます。セグメンテーションとリアセンブリが必要な場合は、APPの担当となります。

8.5 smplStatus_t SMPL _Receive (linkID_t lid, uint8 *msg, uint8 *len)

この呼び出しは、リンクIDlidからのメッセージをチェックします。メッセージがある場合は、メッセージとその長さを返します。ポーリング·モードでのみ動作するため、非ブロッキングとなります。呼び出し側(caller)は、メッセージをコピーするためのバッファのアドレスを供給する必要があります。

所要の受信メッセージのリンクIDは、クライアントの呼び出しによって供給されます。これにより、複数リンクのサポートが可能になります。

注:次のメッセージをそのlidに関係なく取得するために、受信オペレーションをもうひとつ追加することを考えています。

8.6 void SMPL_Ioctl (ioctlObject_t object, ioctlAction_t action, void *val)

これは、APPがランタイムでNWKを構成できるようにする手段です。現在使用可能な値については、セクション10.2で説明しています。

8.7 疑似コードの例

次に示す疑似コードの抜粋は、デバイスがどのようにしてAPのあるSimpliciTIネットワークに参加し、2つの他のデバイスにリンクして、様々なメッセージを送信するかについての、APIのシーケンスを示しています。

このデバイスがリンクしている時は、対象デバイスではSMPL_LinkListen()を最初に実行済みである必要があります。これが、リンクされたペアを認識する方法です。2つのリンクが同じ物理デバイス上にある必要はありません。

アプリケーションが応答を実装している場合は、各SMPL_Send()ステートメントの後にSMPL_Receive(リンクIDノ)が続くことになります。それが意図されている場合には、受信のアクティビティをタイムアウト規律(discipline)で調節して、メッセージを再送信する必要があるかどうかを知る必要があるかもしれません。ただしこの例では、行方不明のメッセージが重要なものでないかぎりは、単なる応答の返されないメッセージになる可能性の方が高いと思われます。

```
void main()
{
  linkID_t linkIDLow, linkIDHigh;
  uint32   temp;

  // Initialize the board's HW
  BSP_InitBoard();

  // Initialize SimpliciTI. The initialization will cause the
  // device to Join as a side effect. The Join will return
  // security key and a token to be used in linking
  // sequences for this network. This is all hidden from
  // the application.

  SMPL_Init(0);  // no callback supplied

  // Establish links to two different (logical) devices.
  // One will get a message if the sampled temperature is
  // to low. The other gets a message if it is too high.
  SMPL _Link(&linkIDLow);
  SMPL _Link(&linkIDHigh);
  while (TRUE)
  {
    // put board to sleep until timer wakes it up
    // to read the temperature sensor
    MCU_Sleep();
    HW_ReadTempSensor(&temp);
    if (temp > TOO_HIGH)
    {
      SMPL _Send(linkIDHigh, "Hot!", 4);
    }
    if (temp < TOO_LOW)
    {
      SMPL _Send(linkIDLow, "Cold!", 5);
    }
  }
}
```

9. シーケンス・ダイアグラムとステート・マシン

10. 顧客が構成可能なオブジェクト

10.1 ビルドタイム

10.1.1 無線以外の項目

次に挙げるのは、ビルドタイム構成可能項目です。各項目にはデフォルト値があるため、実際にはどれも顧客による修正の必要はありません。リストされているのは、現在サポートされている項目のみです。値を定義しているマクロは、3 つのデバイス・タイプそれぞれに対応しているファイルである、smpl_config.datとsmpl_nwk_config.datに入っています。

項目	デフォルト値	説明
MAX_HOPS	3	フレームが削除される前に、フレームが再送信される回数の最大値。各REとAPではホップ数を減らし、フレームを再送信する。
MAX_HOPS_FROM_AP	1	ひとつのEDがAPから離れていることのできる距離の最大値。このホップ数を使うと、MAX_HOPSを使用している場合にEDのポーリングの結果生じる可能性のあるブロードキャストの混乱を大幅に減らすことが可能。 APネットワーク専用。
NUM_CONNECTIONS	4	SMPL_Link()呼び出しとSMPL_LinkListen()呼び出しの両方の結果としてサポートされるリンクの数。デバイスがEDオブジェクト（APやRE）をサポートしていない場合は、0にする必要がある。
MAX_APP_PAYLOAD	10	アプリケーションのペイロード内のバイト数の最大値。
SIZE_INRAME_Q	2	Rxフレーム・キューに保持されるフレームの数。Tx専用デバイスや、フレームをまったく受信しないデバイス用には0を設定可能。
SIZE_OUTFRAME_Q	2	Txフレーム・キューに保持されるフレームの数。NWKアプリケーションによっては、応答との整合を取るためにTxフレームを置いておくものもある。
DEFAULT_JOIN_TOKEN	0x01020304	ネットワークに参加するには、すべてのデバイスでこの値の整合を取る必要がある。Joinメッセージに載せて送信され、受信アクセス・ポイントで整合が取られる。
DEFAULT_LINK_TOKEN	0x05060708	あるネットワーク・デバイスに対するリンク・アクセスを獲得するには、すべてのデバイスでこの値の整合を取る必要がある。Linkメッセージに載せて送信され、受信側デバイスで整合が取られる。
THIS_DEVICE_ADDRESS	0x12345678	各デバイスのアドレスは一意である必要がある。アドレス割り当てにより、Proprietaryクラスの無線通信のブロードキャスト・アドレスである0xnnnnnn00と0xnnnnnnFFをロックアウトする必要がある。
アクセス・ポイント デバイス		
ACCESS_POINT	定義済み	
NUM_STORE_AND_FWD_CLIENT	10	このアクセス・ポイントがサポートするポーリング エンド・デバイスの数。
AP_IS_DATA_HUB	未定義	このマクロを定義すると、デバイスが参加する度に、コールバックを介してAPに自動的に通知される。この通知を受け取った時点で、APではリンク・メッセージをリスンするアプリケーションを実行している必要がある。参加しているEDは、Join応答を受信した直後にリンクする必要がある。
レンジ・エクステンダ・デバイス		
RANGE_EXTENDER	定義済み	
エンド・デバイス		
RX_LISTENS RX_POLLS RX_ALWAYS RX_NEVER	RX_ALWAYS	このうちどれかひとつは必ず定義する必要がある。この情報は、デバイスによって送信される各フレームに組み込まれる。

表 6. ビルドタイム・顧客構成可能・無線以外のパラメータ

10.1.2 無線構成

必要に応じて読み取ったり修正したりできる無線パラメータは数多くあります。SimpliciTI環境は、これらのパラメータ用のデフォルト設定を付けて出荷される予定です。

基本的に、これらはSmartRFによって設定されるパラメータであり、ここで個別にリストアップすることはしていません。デフォルトのSmartRF値を持たないものもあれば、NWKかAPPのスタートアップ·コードで、SMPL_Ioctl()を直接使用して、またはスタートアップ交換中に間接的に、後から修正されるものもあります。

少数の例外を除いて、SimpliciTIのAPIを通して顧客がこれらのパラメータに直接アクセスすることはできません。顧客がこれらのパラメータのどれかを修正する必要が生じた場合にはコードに直接アクセスできますが、APIのサポートは受けられません。

10.2 ランタイム

NWKオブジェクトのランタイム構成は、SMPL_Ioctl（オブジェクト、リクエスト、値）APIを使用して完成されます。次の表は、ランタイム·オブジェクトを要約したものです。

オブジェク	説明	コメント
IOCTL_OBJ_FREQ	無線周波数の取得/設定	周波数アジリティ。APPまたはNWKによって使用可能。
IOCTL_OBJ_CRYPTKEY	暗号化鍵の設定	顧客側で、ユーザーのためにデフォルト以外の鍵を設定する外部手段を提供する可能性がある。 有効にするにはリセットが必要。
IOCTL_OBJ_RAW_IO	フレームを直接送受信するための、フレーム·ヘッダに対するアプリケーション層のアクセス権	このオブジェクトは、コネクション·テーブルを介さずに、対象デバイスのネットワーク アドレスを直接供給するような他のデバイスにpingを打つためなどに使用される。
IOCTL_OBJ_RADIO	何らかの無線制御に対するアプリケーション層のアクセス権	直接無線通信を行うための限定アクセス権。たとえば、無線をスリープ状態にすることや復帰させること、信号の強さの情報を取得することなど。
IOCTL_OBJ_AP_JOIN	アクセス·ポイントの参加許可コンテクスト	アクセス·ポイントがデバイスを参加させるかさせないかを制御するためのインターフェイス。
OCTL_OBJ_ADDR	デバイス·アドレスの取得/設定	ランタイムにデバイス·アドレスを生成する許可を与える。Set関数（Set function）は、SMPL_Init()呼び出しの前に行う必要がある。

表 7. 顧客が構成可能なランタイム·オブジェクト

ご注意

日本テキサス・インスツルメンツ株式会社（以下TIJといいます）及びTexas Instruments Incorporated（TIJの親会社、以下TIJないしTexas Instruments Incorporatedを総称してTIといいます）は、その製品及びサービスを任意に修正し、改善、改良、その他の変更をし、もしくは製品の製造中止またはサービスの提供を中止する権利を留保します。従いまして、お客様は、発注される前に、関連する最新の情報を取得して頂き、その情報が現在有効かつ完全なものであるかどうかご確認下さい。全ての製品は、お客様とTIJとの間に取引契約が締結されている場合は、当該契約条件に基づき、また当該取引契約が締結されていない場合は、ご注文の受諾の際に提示されるTIJの標準販売契約約款に従って販売されます。

TIは、そのハードウェア製品が、TIの標準保証条件に従い販売時の仕様に対応した性能を有していること、またはお客様とTIJとの間で合意された保証条件に従い合意された仕様に対応した性能を有していることを保証します。検査およびその他の品質管理技法は、TIが当該保証を支援するのに必要とみなす範囲で行なわれております。各デバイスの全てのパラメーターに関する固有の検査は、政府がそれ等の実行を義務づけている場合を除き、必ずしも行なわれておりません。

TIは、製品のアプリケーションに関する支援もしくはお客様の製品の設計について責任を負うことはありません。TI製部品を使用しているお客様の製品及びそのアプリケーションについての責任はお客様にあります。TI製部品を使用したお客様の製品及びアプリケーションについて想定されうる危険を最小のものとするため、適切な設計上および操作上の安全対策は、必ずお客様にてお取り下さい。

TIは、TIの製品もしくはサービスが使用されている組み合せ、機械装置、もしくは方法に関連しているTIの特許権、著作権、回路配置利用権、その他のTIの知的財産権に基づいて何らかのライセンスを許諾するということは明示的にも黙示的にも保証も表明もしておりません。TIが第三者の製品もしくはサービスについて情報を提供することは、TIが当該製品もしくはサービスを使用することについてライセンスを与えるとか、保証もしくは是認するということを意味しません。そのような情報を使用するには第三者の特許その他の知的財産権に基づき当該第三者からライセンスを得なければならない場合もあり、またTIの特許その他の知的財産権に基づきTI からライセンスを得て頂かなければならない場合もあります。

TIのデータ・ブックもしくはデータ・シートの中にある情報を複製することは、その情報に一切の変更を加えること無く、かつその情報と結び付られた全ての保証、条件、制限及び通知と共に複製がなされる限りにおいて許されるものとします。当該情報に変更を加えて複製することは不公正で誤認を生じさせる行為です。TIは、そのような変更された情報や複製については何の義務も責任も負いません。

TIの製品もしくはサービスについてTIにより示された数値、特性、条件その他のパラメーターと異なる、あるいは、それを超えてなされた説明で当該TI製品もしくはサービスを再販売することは、当該TI製品もしくはサービスに対する全ての明示的保証、及び何らかの黙示的保証を無効にし、かつ不公正で誤認を生じさせる行為です。TIは、そのような説明については何の義務も責任もありません。

TIは、TIの製品が、安全でないことが致命的となる用途ないしアプリケーション（例えば、生命維持装置のように、TI製品に不良があった場合に、その不良により相当な確率で死傷等の重篤な事故が発生するようなもの）に使用されることを認めておりません。但し、お客様とTIの双方の権限有る役員が書面でそのような使用について明確に合意した場合は除きます。たとえTIがアプリケーションに関連した情報やサポートを提供したとしても、お客様は、そのようなアプリケーションの安全面及び規制面から見た諸問題を解決するために必要とされる専門的知識及び技術を持ち、かつ、お客様の製品について、またTI製品をそのような安全でないことが致命的となる用途に使用することについて、お客様が全ての法的責任、規制を遵守する責任、及び安全に関する要求事項を満足させる責任を負っていることを認め、かつそのことに同意します。さらに、もし万一、TIの製品がそのような安全でないことが致命的となる用途に使用されたことによって損害が発生し、TIないしその代表者がその損害を賠償した場合は、お客様がTIないしその代表者にその全額の補償をするものとします。

TI製品は、軍事的用途もしくは宇宙航空アプリケーションないし軍事的環境、航空宇宙環境にて使用されるようには設計もされていませんし、使用されることを意図されておりません。但し、当該TI製品が、軍需対応グレード品、若しくは「強化プラスティック」製品としてTIが特別に指定した製品である場合は除きます。TIが軍需対応グレード品として指定した製品のみが軍需品の仕様書に合致いたします。お客様は、TIが軍需対応グレード品として指定していない製品を、軍事的用途もしくは軍事的環境下で使用することは、もっぱらお客様の危険負担においてなされるということ、及び、お客様がもっぱら責任をもって、そのような使用に関して必要とされる全ての法的要求事項及び規制上の要求事項を満足させなければならないことを認め、かつ同意します。

TI製品は、自動車用アプリケーションないし自動車の環境において使用されるようには設計されていませんし、また使用されることを意図されておりません。但し、TIがISO/TS 16949の要求事項を満たしていると特別に指定したTI製品は除きます。お客様は、お客様が当該TI指定品以外のTI製品を自動車用アプリケーションに使用しても、TIは当該要求事項を満たしていなかったことについて、いかなる責任も負わないことを認め、かつ同意します。

弊社半導体製品の取り扱い・保管について

半導体製品は、取り扱い、保管・輸送環境、基板実装条件によっては、お客様での実装前後に破壊/劣化、または故障を起こすことがあります。

弊社半導体製品のお取り扱い、ご使用にあたっては下記の点を遵守して下さい。

1. 静電気
   - 素手で半導体製品単体を触らないこと。どうしても触る必要がある場合は、リストストラップ等で人体からアースをとり、導電性手袋等をして取り扱うこと。
   - 弊社出荷梱包単位（外装から取り出された内装及び個装）又は製品単品で取り扱いを行う場合は、接地された導電性のテーブル上で（導電性マットにアースをとったもの等）、アースをした作業者が行うこと。また、コンテナ等も、導電性のものを使うこと。
   - マウンタやはんだ付け設備等、半導体の実装に関わる全ての装置類は、静電気の帯電を防止する措置を施すこと。
   - 前記のリストストラップ・導電性手袋・テーブル表面及び実装装置類の接地等の静電気帯電防止措置は、常に管理されその機能が確認されていること。
2. 温・湿度環境
   - 温度：0～40℃、相対湿度：40～85%で保管・輸送及び取り扱いを行うこと。（但し、結露しないこと。）
   - 直射日光があたる状態で保管・輸送しないこと。
3. 防湿梱包
   - 防湿梱包品は、開封後は個別推奨保管環境及び期間に従い基板実装すること。
4. 機械的衝撃
   - 梱包品（外装、内装、個装）及び製品単品を落下させたり、衝撃を与えないこと。
5. 熱衝撃
   - はんだ付け時は、最低限260℃以上の高温状態に、10秒以上さらさないこと。（個別推奨条件がある時はそれに従うこと。）
6. 汚染
   - はんだ付け性を損なう、又はアルミ配線腐食の原因となるような汚染物質（硫黄、塩素等ハロゲン）のある環境で保管・輸送しないこと。
   - はんだ付け後は十分にフラックスの洗浄を行うこと。（不純物含有率が一定以下に保証された無洗浄タイプのフラックスは除く。）

以上