|  |  |
| --- | --- |
| AN226546 | |
| CYW208XX機能および周辺機器ガイド   |  | | --- | | Author: Abhishek Khinvesra, Ranjith Kumar | | [関連部品ファミリ：CYW20819、CYW20820](https://www.cypress.com/datasheet/CYW20819) | | 関連アプリケーションノート：参照  [関連資料](#_Related_Application_Notes) | | |
| ModusToolbox™IDEを使用して増え続けるCYW208XXコード例のリストにアクセスするには、GitHubリポジトリにアクセスしてください。 | |
| このドキュメントでは、ModusToolbox™ソフトウェアを使用してアプリケーションを作成する開発ツールとともにCYW208XXデバイスアーキテクチャを探索することにより、Arm®Cortex®-M4CPUを備えたCYW208XX超低電力デュアルモードBT v5.0ワイヤレスMCUを紹介します。これらのアプリケーションの開発を支援するために、このテクニカルガイドでは、デバイスのビルディングブロックとアプリケーションの開発中に役立つツールの使用に焦点を当てています。このテクニカルガイドは、CYW208XXの機能と周辺機器の概要を説明することを目的としています。 |

目次

[1 はじめに 1](#_Toc38277582)

[2 サイプレスのリソース 2](#_Toc38277583)

[3 ハードウェアEVB 2](#_Toc38277584)

[4 CYW208XX Bluetooth SoC 2](#_Toc38277585)

[5 CYW208XX SoC周辺機器 3](#_Toc38277586)

[5.1 汎用入出力（GPIO） 3](#_Toc38277587)

[5.2 アナログ-デジタル変換器 (ADC) 4](#_Toc38277588)

[5.3 クロック 6](#_Toc38277589)

[5.4 リアルタイムクロック(RTC) 7](#_Toc38277590)

[5.5 ウォッチドッグ タイマ(WDT) 8](#_Toc38277591)

[5.6 アプリケーションタイマー 8](#_Toc38277592)

[5.7 パルス幅変調器（PWM） 9](#_Toc38277593)

[5.8 乱数ジェネレーター（RNG） 9](#_Toc38277594)

[5.9 相互集積回路（I2C） 10](#_Toc38277595)

[5.10 シリアル ペリフェラル インターフェース(SPI) 11](#_Toc38277596)

[5.11 UART 12](#_Toc38277597)

[5.12 キーボードスキャナー（Keyscan） 14](#_Toc38277598)

[6 BLEラジオの機能 15](#_Toc38277599)

[6.1 TX電力制御 15](#_Toc38277600)

[6.2 共存-SECI 15](#_Toc38277601)

[7 ソフトウェア 15](#_Toc38277602)

[7.1 アプリケーションコード 15](#_Toc38277603)

[7.2 低電力機能 16](#_Toc38277604)

[7.3 ファームウェアのアーキテクチャ 16](#_Toc38277605)

[7.4 メモリレイアウト 17](#_Toc38277606)

[7.5 プログラム 18](#_Toc38277607)

[7.6 プログラミング用ツール 21](#_Toc38277608)

[8 まとめ 22](#_Toc38277609)

[9 関連文書: 23](#_Toc38277610)

[改訂履歴 24](#_Toc38277611)

[世界中にわたるセールスおよびデザイン サポート 25](#_Toc38277612)

# はじめに

CYW208XXは、Bluetooth基本レート（BR）、拡張データレート（EDR）、およびBluetooth低エネルギー（BLE）をサポートするBluetooth 5.0準拠のSoCです。CYW208XXは、Bluetoothコア仕様v4.2に基づくすべてのオプションのLE機能と、仕様v5.0に基づくLE 2 Mbps機能をサポートしています。最先端の40 nm CMOS低電力プロセスを使用して製造されたCYW208XXは、最高レベルの統合を採用してすべての重要な外部コンポーネントを排除し、デバイスの設置面積とBluetoothソリューションの実装に関連するコストを最小限に抑えます。

CYW208XXは、96 MHzのArm®Cortex®-M4CPUを統合し、優れた処理能力を提供しながら、より小さなフットプリントを維持します。CYW208XXは、Bluetoothアプリケーション用の160 KBのRAM、16 KBのパッチRAM、8 KBのキャッシュ、256 KBのオンチップセキュアフラッシュ、および低レベルBluetoothスタックとドライバー専用の1 MBのROMスペースを提供します。CYW208XXは、ADC、PWM、I2C、SPI、UART、キーボードスキャナーなどのいくつかのハードウェア周辺ブロックもホストし、他のコンポーネントとのインターフェースを支援します。このガイドは、CYW208XXデバイスの機能を探索し、SoCペリフェラルの使用をガイドするのに役立ちます。

# サイプレスリソース

サイプレスは、www.cypress.comで豊富なデータを提供し、適切なデバイスを選択し、デバイスを設計に迅速かつ効果的に統合するのに役立ちます。以下は、このアプリケーションノートに関連するリソースの簡略リストです。

|  |  |
| --- | --- |
| * [概要：BLEおよびBluetooth](https://www.cypress.com/products/ble-bluetooth) * [製品セレクター：Bluetooth製品セレクター](https://www.cypress.com/products/ble-bluetooth) * CYW20819データシート：CYW20819 Bluetooth SoCの電気的仕様を説明および提供します。 * CYW20820データシート：CYW20820 Bluetooth SoCの電気的仕様を説明および提供します。CYW20820には、10 dBmの送信電力で送信できるパワーアンプが統合されています。 * IDE：ModusToolbox IDE for BT-SDK * アプリケーションノート：基本的なレベルから高度なレベルまで、幅広いトピックをカバーします。 * コード例：GitHubリポジトリで入手可能なBT-SDKコード例。 * BT-SDKドキュメント：ファームウェア開発、デバッグガイド、テストツール、ライブラリガイド、APIドキュメントを含む最新のサイプレスBT-SDKドキュメント。 | * **開発キット：いくつかの例は次のとおりです。** * CYW920819EVB-02評価キットを使用すると、超低消費電力のデュアルモードBluetooth 5.0ワイヤレスMCUデバイスであるCYW20819を使用して、シングルチップBluetoothアプリケーションを評価および開発できます。 * CYW920820EVB-02評価キットを使用すると、超低消費電力のデュアルモードBluetooth 5.0ワイヤレスMCUデバイスであるCYW20820を使用して、シングルチップBluetoothアプリケーションを評価および開発できます。CYW20820には、10 dBmの送信電力で送信できるパワーアンプが統合されています。 * CYBT-213043-MESH EZ-BT™モジュールメッシュ評価キットを使用すると、EZ-BT Bluetooth 5.0認定モジュールCYBT-213043-02を使用して、Bluetooth SIGメッシュ機能を評価および開発できます。 |

# ハードウェアVB

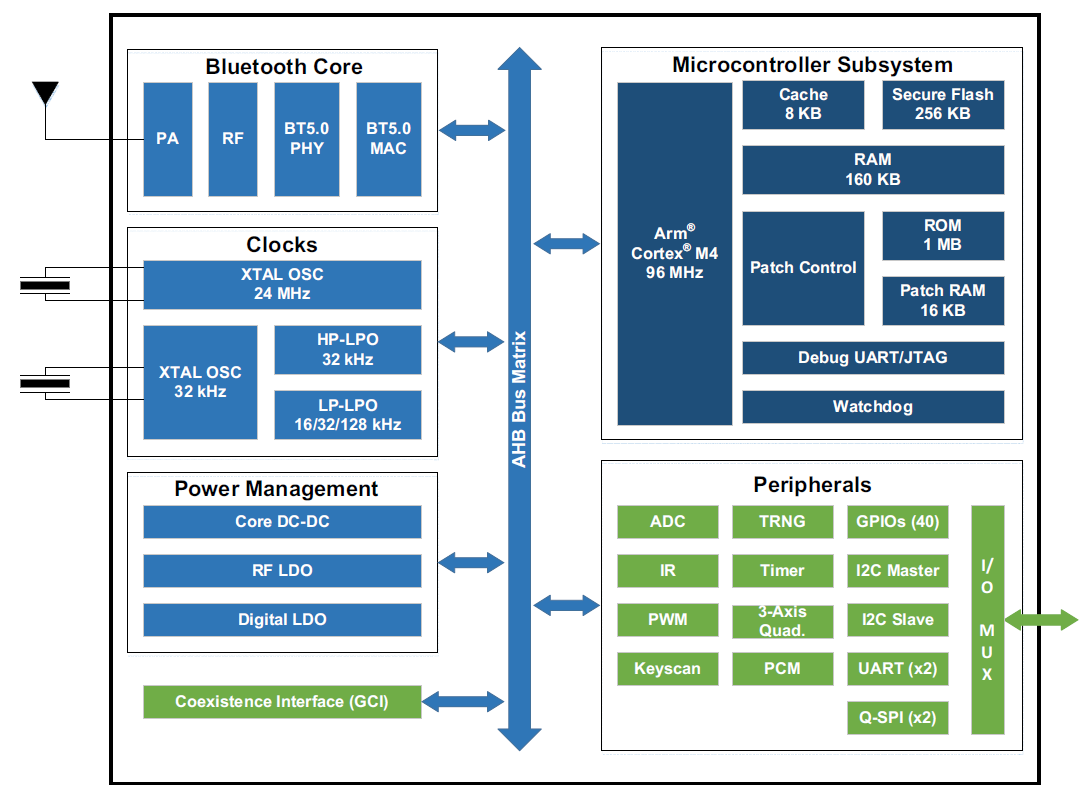
サイプレスは、CYW208XX Bluetooth SoCを評価するための3つの評価キットを提供しています。以下を使用できます。

* サイプレスCYW920819EVB-02評価キットは、超低消費電力のデュアルモードBluetooth 5.0ワイヤレスMCUデバイスであるCYW20819を使用してシングルチップBluetoothアプリケーションを評価および開発します。
* サイプレスCYW920820EVB-02評価キットは、超低消費電力のデュアルモードBluetooth 5.0ワイヤレスMCUデバイスであるCYW20820を使用してシングルチップBluetoothアプリケーションを評価および開発します。CYW20820には、10 dBmの送信電力で送信できるパワーアンプが統合されています。
* CYBT-213043-02 EZ-BT Bluetooth 5.0認定モジュールを使用してBluetooth SIGメッシュ機能を評価するEZ-BTメッシュ評価キット（CYBT-213043-MESH）。

# CYW208XX Bluetooth SoC

図1に、CYW208XXデバイスのブロック図を示します。CYW208XX SoC機能の詳細については、CYW20819データシートとCYW20820データシートを参照してください。

図. CYW208XX Bluetooth SoCのブロック図



# CYW208XX SoC周辺機器

サイプレスは、BT SDKにハードウェアアブストラクションレイヤー（HAL）ドライバーを提供し、CYW208XXのADCやI2CなどのいくつかのSoC周辺機器を利用します。BT SDKのHALドライバーはブロッキング関数です。BT SDKのAPI実装の詳細については、wiced\_btsdkのREADME.mdファイルのAPIドキュメントリンクをクリックするか、Githubドキュメントにアクセスしてください。一部のSoCペリフェラルには、ハードウェアブロックの機能を示す専用のコード例があり、BT SDKで使用できます。サイプレスのGitHubリポジトリでより多くのコード例を入手できます。

## 汎用入出力（GPIO）

GPIOは、ユーザーによる実行時の構成に応じて入力または出力として機能できるデジタル信号ピンです。BT SDKは、wiced\_hal\_gpio.hでGPIOドライバーを提供します。CYW208XXは、LHL GPIO（最大40）とArm GPIO（最大10）の2種類のGPIOをサポートしています。デバイスで使用可能なLHL GPIOとArm GPIOの数は、パッケージによって異なります。

* CYW20819A1KFB1G（112ボールFBGA）デバイスパッケージには、40 LHL GPIOと6 Arm GPIOが含まれています。CYW20819A1KFBG（62ボールFBGA）デバイスパッケージには、22のLHL GPIOと2つのArm GPIO（DEV\_WAKEとHOST\_WAKE）があります。LHLとArm GPIOの両方がwiced\_hal\_gpio.hのwiced\_bt\_gpio\_numbers\_t（）リストに列挙されています。
* CYW20820A1KFB1G（112ボールFBGA）デバイスパッケージには、40 LHL GPIOと6 Arm GPIOが含まれています。CYW20820A1KFBG（62ボールFBGA）デバイスパッケージには、22個のLHL GPIOと2個のArm GPIO（DEV\_WAKEおよびHOST\_WAKE）があります。LHLとArm GPIOの両方がwiced\_hal\_gpio.hのwiced\_bt\_gpio\_numbers\_t（）リストに列挙されています。

LHL GPIOは、いくつかの点でArm GPIOと異なります。

* LHL GPIOは低電力モードで動作し、多重化可能な周辺機器をサポートできます。I2C、SPI、PUART、ACLKなどのペリフェラルは、LHL GPIOにその機能を引き出すことができ、スーパーマルチプレクサと呼ばれています。Super-Muxableのペリフェラルのリストについては、データシートを参照してください。LHL GPIOは、ほとんどのユーザーアプリケーションで通常使用されるものです。
* Arm GPIOはアクティブモードでのみ動作でき、マルチプレクサ可能な周辺機器をサポートしていません。DEV\_WAKE、HOST\_WAKE、および4つのBT\_GPIOは、CYW20819A1KFB1Gパッケージで利用可能なArm GPIOですが、DEV\_WAKEおよびHOST\_WAKEは、CYW20819A1KFBGパッケージで利用可能なArm GPIOです。DEV\_WAKEを使用して、ホストからの信号でBluetoothデバイスをウェイクさせることができます。HOST\_WAKE信号を使用して、Bluetoothデバイスからの信号でホストデバイスをウェイクアップできます。BT\_GPIOは、GCI（グローバル共存インターフェイス）ピンとして構成できます。

CYW208XXは、BT SDKで以下のGPIO機能をサポートしています。

* 入出力選択
* プルアップ/プルダウン選択
* ドライブ強度の選択
* ヒステリシス制御
* レベル/エッジトリガー割り込み選択

以下は、最も一般的に使用されるGPIO APIとその説明の一部です。

* wiced\_hal\_gpio\_configure\_pin（）APIは、特定のGPIOでGPIO機能を構成するために使用されます。
* wiced\_hal\_gpio\_register\_pin\_for\_interrupt（）を使用してGPIOの割り込みを登録し、エッジまたはレベルの変更に基づいてコールバック関数（つまり、割り込みサービスルーチンまたは割り込みハンドラー）をトリガーします。
* GPIO割り込みコールバック関数で割り込みをクリアするには、wiced\_hal\_gpio\_clear\_pin\_interrupt\_status（）を使用する必要があります。
* wiced\_hal\_gpio\_select\_function（）APIを使用して、機能を特定のピンにマップできます（使用方法の例として、次のコードスニペットを参照してください）。この関数の最初のパラメーターはGPIOピン（wiced\_bt\_gpio\_numbers\_t）を参照し、2番目のパラメーターはGPIO機能（wiced\_bt\_gpio\_function\_t）を参照します。

wiced\_hal\_gpio\_select\_function (*WICED\_P03*, *WICED\_PWM0*);

これらのGPIO APIの詳細については、**Components** > **Hardware Drivers** > **GPIO** をクリックし、WICED APIリファレンスガイドのGPIO APIリファレンスに移動してください。

SDKはすべてのI / Oの構成を提供していますが、アプリケーションでのピンの選択は、デバイスに関する以下の制約に基づいて行う必要があります。

* 特定の機能（ADC入力など）は、特定のLHL I / Oでのみサポートされており、他のピンにルーティングすることはできません。詳細については、データシートの「ピン割り当てとGPIO」セクションを参照してください。
* P26、P27、P28、およびP29などのGPIOは、他のピンと比較してより高い電流を駆動できます。詳細については、データシートを参照してください。

## A / Dコンバータ（ADC）

CYW208XXのADCブロックは、オーディオ測定用に12ビット、DC測定用に10ビットのシングルエンドスイッチトキャパシタシグマデルタADCです。ADCには、28個のGPIO入力を含む32個のDC入力チャンネルがあります。 ADCはSuper-Muxing機能をサポートしていません。 説明されているADC APIの詳細については、**Components** > **Hardware Drivers** > **Analog-to-Digital Converter (ADC)** をクリックし、WICED APIリファレンスガイドのADC APIリファレンスに移動してください。。

* 次の15ピンは、CYW20819A1KFBG / CYW20820A1KFBG（62ボールFBGA）でADC入力としてプログラムできます。

P0, P1, P8-P15, P17, P28, P29, P32, P37

* 以下の28ピンは、CYW20819A1KFB1G / CYW20820A1KFB1G（112ボールFBGA）のADC入力としてプログラムできます。

P0, P1, P8-P19, P21-P23, P28-P38

BT SDKは、wiced\_hal\_adc.hのVDDIO（入力/出力電圧）、VDD\_CORE（コア電圧）、ADC\_BGREF（バンドギャップリファレンス）、およびADC\_REFGND（リファレンスグランド）の電圧/未加工サンプルを読み取るためのAPIを提供します。詳細については、データシートのADC電気的特性を参照してください。

CYW208XXは、DC測定モードとAudio1モードの2つの動作モードをサポートしています。Audio1アプリケーションの場合、ADCはプログラマブルゲインアンプ（PGA）を使用してゲインを制御します。

.

### DC測定モード

DC測定モードは、ADCグランドを基準とした入力ピンでのシングルエンド電圧測定を指します。ADCのブロック図から、28個のGPIOチャネル、VDDIO、VDD\_CORE、ADC\_BGREF、およびADC\_REFGNDが多重化され、スイッチトキャパシタシグマデルタADCに供給されます。

wiced\_hal\_adc\_init（）関数は、DC測定モードでのADC動作に必要なレジスターを初期化します。ADCが特定の低電力モードでパワーダウンされている場合は、ADCを再初期化する必要があります。

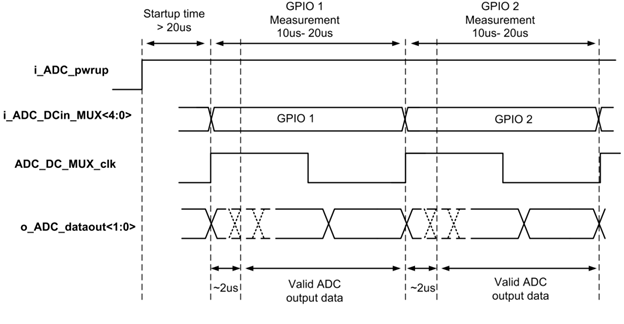
#### ADC出力データとクロック

ADCクロック入力は24 MHzで、ADCハードウェアでは内部で12 MHzに分割されます。ADCの静的測定では、標準的な変換レートは100 kHzであり、最小変換レートは50 kHzです。ADCの初期化後にDC電圧を読み取る実際の時間は約150マイクロ秒です。同様に、ADCからDC生サンプルを読み取る実際の時間は約50マイクロ秒です。変換レートの低下は、生のサンプル読み取り関数がサンプルの後処理を行わないためです。これに対して、電圧を読み取る場合、ファームウェアは、サンプリングされた値の平均フィルタリング（ローパスFIR）を移動して精度を高めます。この変換率は、おおよそ16 kspsから20 kspsの範囲に変換されます。

#### DC測定モードでのADC入力MUXクロッキング

ADCは、DC入力チャネルの切り替え後、セトリングに最大2マイクロ秒を必要とします。ADCの電源投入後、ADC REFが安定するまでに約20マイクロ秒かかります。DC測定モードの集録時間は10 µsです。これは、API関数によって内部的に処理されます。ADC\_DC\_MUX\_clkは、DC入力チャネルのスイッチングと測定を同期させるためのデジタル側のタイミングクロックです。図2に、MUXのクロッキングとタイミングを示します。

図2. ADCのMUXクロッキングとタイミング図



#### ADCサンプリングとゲイン計算

wiced\_hal\_adc\_read\_voltage（）API関数を使用すると、プログラムはADCからアナログ電圧を読み取ることができます。電圧を読み取る前にADCを初期化する必要があります。そうしないと、関数は「0」を返します。ADCレジスタへの読み取り関数呼び出しはすべてブロックされます。DC測定モードでのADC入力とADC出力ゲインの関係は、電源電圧（ADC\_AVDDBAT、MIC\_AVDD、VDDC）により変化する可能性があります。電源電圧が変化した場合、より正確なADCサンプルを取得するには、次のAPI関数を呼び出す必要があります。これらの関数は、呼び出されると、内部バンドギャップと基準電圧を再調整します。

* 電源電圧が1.8 Vに設定されている場合は、wiced\_hal\_adc\_set\_input\_range（ADC\_RANGE\_0\_1P8V）を使用します。
* 電源電圧が3.3 Vに設定されている場合は、wiced\_hal\_adc\_set\_input\_range（ADC\_RANGE\_0\_3P6V）を使用します。

## クロック

CYW208XXには、複数の高速および低速周辺機器に接続された複数のクロック分周器とPLLがあります。以下の図は、CYW208XXの簡略化されたクロックツリーを示しています。

CYW208XXのすべての高周波クロックの基本的なソースは、24 MHzの外部クリスタル（xtal）です。これは、周波数を96 MHzに変換するADPLLに供給されます。このクロックは、BTコアやCPUなどのCYW208XXデバイスのほとんどの周辺機器に供給されます。PWMなどの一部の周辺機器は、24 MHzクロックを直接取得することもできます。これらのペリフェラルには、必要に応じて入力クロックを分周できる内部分周器があります。

低周波数クロックは32 kHzで、複数のソースから生成できます。詳細については、低周波数クロックソースを参照してください。

### 高周波クロックソース

高周波クロックは、内部PLLを使用して周波数をアップスケールする外部24 MHz水晶発振器から生成されます。これらの周波数はハードウェアによって自動的に決定され、ユーザーが構成することはできません。

1. HCLK：このクロックは、リソース処理ユニット（RPU）に電力を供給します。RPUには、CM4 CPUとDMAコントローラーが含まれています。HCLKの周波数の範囲は1 MHz〜96 MHzです。
2. BTコアクロック：このクロックは、Bluetooth RFハードウェアに電力を供給します。BTコアのクロック周波数の範囲は1 MHz〜48 MHzです。
3. PTUクロック：このクロックは、SPI、I2C、UART、PCM、および乱数ジェネレーターを含むペリフェラルトランスポートユニット（PTU）に電力を供給します。PTUクロック周波数は、24 MHzまたは48 MHzのいずれかです。周辺機器は、特定の目的のために内部でクロック周波数を分割します。たとえば、I2Cマスタークロック（SCL）の範囲は100 kHz〜1 MHzです。
4. タイマークロック：アプリケーションタイマー（デュアル入力32ビットタイマー）は1 MHzでクロックされます。

### 低周波数クロックソース

32 kHzの低周波数クロック（次の図ではlhl\_lpo\_32kHz）は、複数のソースから取得できます。LP-LPOおよびHP-LPOと呼ばれる2つの内部低電力発振器（LPO）と外部水晶（OSC32K）があります。ファームウェアは、精度と電力要件に応じて、使用可能なLPO間で使用するクロックソースを決定します。精度が高く、消費電流が最小であるため、推奨されるソースは外部LPO（OSC32K）です。内部LP-LPOの消費電流と精度は低く、HP-LPOの精度と消費電流は高くなります。ファームウェアは、外部LPOのエラーが250 PPM未満で、ジッターがほとんどないかまったくないと想定しています。

図3。CYW208XXのクロック図



## リアルタイムクロック(RTC)

CYW208XXは、32 kHzクリスタル（XTAL32K）LPOで実行される48ビットRTCタイマーをサポートします。RTCは、外部または内部LPOからのクロック入力をサポートします。外部LPOがCYW208XXに接続されていない場合、ファームウェアはRTCの内部LPOからクロック入力を取得します。RTCタイマーは構造wiced\_rtc\_time\_tで表され、メンバーは秒、分、時、日、月、年を表します。デフォルトでは、日付と時刻は2010年1月1日に設定され、時刻は00:00:00で、HH：MM：SSを示します。

BT SDKは、以下のAPI関数を提供します。

* 現在の時刻を設定します（wiced\_set\_rtc\_time（））。
* 現在の時刻を取得します（wiced\_rtc\_get\_time（））。
* 現在の時刻の値を文字列に変換します（wiced\_rtc\_ctime（））。

上記のRTC APIの詳細については、**Components** > **Hardware Drivers** > **RTCDriver** をクリックし、BT-SDKドキュメントページのRTCDriver APIリファレンスに移動してください

## ウォッチドッグタイマー（WDT）

CYW208XXのウォッチドッグタイマーモジュールは、誤動作の検出と回復に使用される32ビットダウンカウンターに基づいています。通常の動作中、デバイスは、カウント値がゼロに達する前にウォッチドッグタイマーを定期的にリセットして、経過やタイムアウトを防ぎます。ハードウェア障害またはプログラムエラーが原因で、デバイスがウォッチドッグをリセットできなかった場合、タイマーが経過し、タイムアウト時にシステムリセットが生成されます。ウォッチドッグタイマーのカウンターをリセットするプロセスは、「犬をかわいがる」と呼ばれます。デフォルトのウォッチドッグタイムアウトは2秒に設定され、ウォッチドッグペッティングはアイドルスレッドで行われます。本番アプリケーションはウォッチドッグを満足させる必要はありません。CPUを2秒より長く保持しようとするアプリケーションは、ウォッチドッグをトリガーしてシステムを再起動します。ウォッチドッグタイマーを無効にするために、wiced\_hal\_wdog\_disable（）APIがwiced\_hal\_wdog.hで利用可能です。

BT SDKは、wiced\_hal\_wdog.hを使用したウォッチドッグタイマーを介して、制限されたデバッグ機能を提供します。ウォッチドッグタイマーが期限切れになると、システムはコアダンプ後にリセットされます。コアダンプには、デバイスのファームウェアまたはハードウェアのバージョン、警告フラグ、メモリ情報、CPU / HWレジスタ、SRAM、およびパッチRAMイメージの情報が含まれています。コアダンプは、BT-HCIベンダー固有のイベントとしてHCI-UARTを介して送信されます。

## アプリケーションタイマー

CYW208XXは、1つの汎用32ビットデュアル入力タイマーを提供します。ファームウェアは、2 x 32ビットタイマーを1 x 64ビットタイマーとして使用します。BT SDKは、wiced\_timer.hのタイマー機能を使用するためのAPIを提供します。最初に、アプリケーションタイマーを特定の秒またはミリ秒のタイマー値に初期化してから、タイマーを開始し、タイムアウトに達すると、コールバック関数が実行されます。このコールバック関数は、LEDの点滅、シリアルデータ転送、RTC値の読み取り、ADCサンプルの読み取りなどのアプリケーションコードを保持できます。タイマーは2つの動作モードをサポートしています。

* 定期割り込みモード
* シングルショットモード

BT SDKには、4つのタイマータイプがあります。

* WICED\_SECONDS\_TIMER
* WICED\_MILLI\_SECONDS\_TIMER
* WICED\_SECONDS\_PERIODIC\_TIMER
* WICED\_MILLI\_SECONDS\_PERIODIC\_TIMER

2つの定期的なタイマーは、タイマーカウントがゼロ（ダウンカウンター）に達するたびにタイマーコールバック関数をトリガーしますが、他の2つのタイマーはシングルショットです。タイマーコールバックは1回だけ実行され、タイマーが期限切れになるたびに終了します。タイマーAPIの使用を開始するのに役立ついくつかのAPIは次のとおりです。

* wiced\_init\_timer（）関数はタイマーを初期化します。タイマーが切れたときに呼び出されるコールバック関数を指定できます。この関数では、使用するタイマーのタイプを指定することもできます。この関数はタイマーを開始しません。
* wiced\_start\_timer（）関数は、カウントダウンを開始し、カウント値を指定するために使用されます。wiced\_stop\_timer（）関数を使用して、タイマーを停止できます。
* wiced\_is\_timer\_in\_use（）関数を使用すると、ブール値を返すことにより、指定したタイマーが現在使用中かどうかを確認できます。

これらのアプリケーションタイマーAPIの詳細については、Components > Hardware Drivers > Timer Management Servicesをクリックし、WICED APIリファレンスガイドのタイマーAPIリファレンスに移動してください。

アプリケーションタイマーを使用するサンプルコードスニペットを次に示します。

wiced\_timer\_t my\_timer\_handle; /\* Typically defined as a global \*/

.

.

.

/\* Typically, inside the BTM\_ENABLED\_EVT \*/

wiced\_init\_timer (&my\_timer\_handle,

myTimer,

0,

*WICED\_MILLI\_SECONDS\_PERIODIC\_TIMER*);

wiced\_start\_timer (&my\_timer\_handle, 100);

.

.

.

/\* The timer function \*/

**void myTimer** (uint32\_t arg)

{

/\* Put timer code here \*/

}

## パルス幅変調器（PWM）

CYW208XXには6つの16ビットハードウェアPWMチャネルがあり、Super-Muxable機能をサポートしています。BT SDKのwiced\_hal\_pwm.hファイルは、PWM固有のドライバーを定義します。LHL\_CLKまたはPMU\_CLKは、各PWMチャネルのクロックソースとして使用できます。クロックソースがLHL\_CLKの場合、クロック周波数は32 kHzになります。PMU\_CLKでは、最初に補助クロックを設定する必要があります。補助クロックを構成する場合、ACLK0はPWMで使用できません。したがって、ACLK1は、24 MHzまたは1 M​​Hz（内部クロック分周を使用）で実行される唯一の利用可能なPMU\_CLKです。SuperMUXを使用して、各PWMチャネルをGPIOピンにルーティングできます。wiced\_hal\_gpio\_select\_function（）は、LHL GPIOを指定されたPWMチャネル（WICED\_PWM< x>）にマッピングするために使用されます。ここで、「x」はPWMチャネルを表します。PWM構成パラメーターのランタイム変更中にグリッチが発生する可能性があります。このAPIについては、先に汎用入出力（GPIO）で説明しました。PWM APIの詳細については、**Components** > **Hardware Drivers** > **Pulse Width Modulation (PWM)** をクリックし、BT-SDKドキュメントページのPWM APIリファレンスに移動してください。

* wiced\_hal\_aclk.hのwiced\_hal\_aclk\_enable（）関数は、補助クロックを有効にし、PWMへのルーティングに1 MHzまたは24 MHzを選択する機能を許可します。

注：BT SDK 1.3では、CYW20820は1 MHzのACLKをサポートしていません。

* wiced\_hal\_pwm\_start（）関数は、PWMを構成、有効化、開始し、事前構成されたGPIOピンにルーティングします。PWMで使用するために割り当てる前に、目的のGPIOピンを出力として構成する必要があります。この関数は、5つのパラメーター（つまり、チャネル、クロックソース、トグルカウント、初期カウント、および反転信号）を受け取ります。
* チャネルは、PWMチャネル0〜5を指します。
* クロックソースは、LHL\_CLKまたはACLK1（PMU\_CLK）のいずれかです。
* トグルカウントは、信号を切り替える前に待機するティックの数を指します。
* 初期カウントは、レジスタの初期値です。
* 反転信号は1または0のいずれかです。反転信号が1の場合、PWM出力は論理HIGHで開始し、反転信号が0の場合、PWM出力は論理LOWで開始します。
* wiced\_hal\_pwm\_change\_values（）関数は、PWMハードウェアブロックが既に起動された後、特定のPWMチャネルのトグルカウントや初期カウントなどのPWM設定を変更します。
* wiced\_hal\_pwm\_get\_params（）は、PWMパラメータを計算するユーティリティ関数です。この関数は、入力クロック周波数、デューティサイクル、および必要な初期出力カウントとトグルカウントを決定するために必要なPWM出力周波数の3つのパラメーターを受け取ります。初期カウントとトグルカウントは、次の式を使用して関数によって決定されます。
* Initial count = 0xFFFF - (Input clock frequency/PWM frequency output)
* Toggle count = 0xFFFF - ((duty cycle in percentage) x (Input clock frequency/PWM frequency output)/ 100)

## 乱数ジェネレーター（RNG）

BT SDKは、RNGハードウェアモジュールを使用するためのAPIを提供します。単一の32ビット乱数を生成するか、特定の配列に32ビット乱数を入力する関数を提供します。これらの機能は、認証などのセキュリティ関連のアプリケーションに役立ちます。これらの関数は、「シード」と呼ばれる入力から乱数を生成します。この入力は、生成ハードウェアブロックの温度が高い場合はその温度から、BTクロックが低い場合はそのクロックから取得されます。wiced\_hal\_rand.hファイルは、32ビットのランダムな整数を生成するwiced\_hal\_rand\_gen\_num（）や、ランダムに生成された32ビットの整数で特定の配列を埋めるwiced\_hal\_rand\_gen\_num\_array（）などの関数を定義します。 RNGハードウェアブロックは、ウォームアップ時間を含む乱数（単一の数値または配列）を生成するために1ミリ秒かかります。 RNG APIの詳細については、**Components** > **Hardware Drivers** > **Random Number Generator (RNG)** をクリックし、BT-SDKドキュメントページのPWM APIリファレンスに移動してください。

## 相互集積回路（I2C）

CYW208XXは、I2Cスレーブデバイスとシリアル通信するための1つのI2C互換マスターインターフェイスを提供します。I2CブロックはSuper-Muxable機能をサポートします。

I2Cモジュールの機能には、次のサポートが含まれます。

* 7ビットアドレス指定モード
* Clock stretching as a Master
* SCLクロック周波数：100 kHz（標準モード）、400 kHz（高速モード）、800 kHz（独自仕様）、および1 MHz（高速モード+）
* マルチスレーブ操作

### I2Cクロック

I2Cクロック（SCL）は、I2Cバス上のマスターによって提供されます。I2Cモジュールがマスターモードの場合、シリアルクロックジェネレーターは24 MHzのトランスポートクロックからSCLクロックを生成します。wiced\_hal\_i2c\_set\_speed（）関数を使用すると、SCLのクロックレートを概要にリストされている速度のいずれかに設定できます。I2Cブロックには、トランザクションバッファー用に64バイトのRXおよびTX FIFOがあるため、各I2Cトランザクションペイロードは最大64バイトにすることができます。

必要なSCLクロック周波数を設定するために、次の列挙定数がwiced\_hal\_i2c.hで提供されています。

* I2CM\_SPEED\_100KHZ
* I2CM\_SPEED\_400KHZ
* I2CM\_SPEED\_800KHZ
* I2CM\_SPEED\_1000KHZ

たとえば、I2CM\_SPEED\_100KHzの列挙値は、除算パラメーターである240に設定されます。つまり、24,000,000を240で除算すると、100 kHzになります。マルチマスターを使用しようとすると、2つ以上のマスターからのクロックが同期されないため、未定義の動作が発生します。また、独自仕様の800 kHzモードは、すべてのI2Cスレーブで動作するとは限りません。wiced\_hal\_i2c\_get\_speed（）関数を使用すると、I2Cブロックの現在のクロック周波数を取得できます。

### I2C初期化

I2Cモジュールレジスタは、wiced\_hal\_i2c.hで定義されているwiced\_hal\_i2c\_init（）を呼び出すことによって初期化されます。この関数呼び出しは、I2Cドライバーとそのプライベート値を初期化します。一部のモジュールは節電のためにブロックをオフにするため、I2Cを使用する前にこの関数を呼び出す必要があります。この関数呼び出しにより、ハードウェアブロックへのクロック信号がデフォルトのSCLクロック周波数100 kHzで確実にオンになります。したがって、100 kHz以外の速度を使用する場合は、wiced\_hal\_i2c\_init（）関数の後にwiced\_hal\_i2c\_set\_speed（）関数を呼び出す必要があります。

### I2Cオペレーション

I2Cマスターモジュールは、マスター書き込み、マスター読み取り、結合書き込みとそれに続く読み取りの3つの操作をサポートしています。これらの操作については、次のセクションで説明します。

#### I2Cマスター書き込み

wiced\_hal\_i2c\_write（）関数は、特定のスレーブアドレスにデータを書き込みます。任意の長さのデータをスレーブに書き込むことができますが、ハードウェアの能力を超えるアトミックトランザクションは不可能であり、データは複数のトランザクションに分割されます。これはブロッキングコールです。I2Cスレーブデバイスが成功したI2Cマスター書き込み操作を確認すると、関数はI2CM\_SUCCESSを返し、I2CマスターがI2Cスレーブから確認を受信しない場合、関数はI2CM\_OP\_FAILEDを返します。

#### I2Cマスター読み取り

wiced\_hal\_i2c\_read（）関数は、特定のスレーブアドレスレジスタから指定されたバッファにデータを読み取ります。これはブロッキング呼び出しです。このI2C関数は、I2CスレーブデバイスからのNACK（確認応答なし）がある場合、またはI2Cスレーブが0バイトを返す場合、I2CM\_OP\_FAILEDを返します。詳細については、BT-SDKのドキュメントを参照してください。

#### I2Cマスターの書き込みと読み取りの組み合わせ

wiced\_hal\_i2c\_combined\_read（）関数は、書き込みトランザクションを実行し、続いて読み取りトランザクションが実行され、トランザクション間に開始条件が繰り返されます。最初のトランザクションでは、データはスレーブアドレスに書き込まれ、繰り返し開始した後、2番目のトランザクションでスレーブからデータが読み取られます。この操作は通常、スレーブデバイスのレジスタを読み取るために使用されます。最初の書き込みトランザクションでは、読み取る必要があるスレーブのレジスタアドレスを指定します。このI2C関数は、I2CスレーブデバイスからのNACK（確認応答なし）がある場合、またはI2Cスレーブが0バイトを返す場合、I2CM\_OP\_FAILEDを返します。

## シリアル ペリフェラル インターフェース(SPI)

CYW208XXには、他のSPIマスターおよびスレーブデバイスとの通信に使用される2つのSPIブロックが含まれています。CYW208XX SPIインターフェースは、Super-Muxable機能をサポートしています。このブロックは、温度センサーやモーションセンサーなどのSPIベースのセンサーとの通信に使用できます。SPIブロックは次の機能をサポートします。

* 3線式（マスター）および4線式（マスターおよびスレーブ）SPIインターフェース
* マスターモードとスレーブモード
* 構成可能なSCK極性と位相
* 設定可能なLSBまたはMSBファースト転送
* SPI 1用の1024バイトの送信バッファーと1024バイトの受信バッファー（HCI UARTと共有）
* SPI 22用の64バイトの送信バッファーと64バイトの受信バッファー（PUARTと共有）

CYW208XXは、2つのSPIハードウェアブロックから選択する機能を提供します。spi\_interface\_tでは、SPI1とSPI2のどちらかを選択できます。各SPIユーティリティ関数には、2つのSPIブロックから選択する引数があります。

SPIブロックは、汎用マスターまたは汎用スレーブとして使用できます。SPIインターフェースには、どのLHL GPIOも使用できます。BT SDKは、SPIドライバーにアクセスするためのパラメーターと関数のリストを提供します。 SPI APIの詳細については、**Components** > **Hardware Drivers** > **PeripheralSpiDriver** をクリックし、BT-SDKドキュメントページのSPI APIリファレンスに移動してください。

SPIブロックは、wiced\_hal\_pspi\_init（）関数を使用して、必要な構成で初期化できます。汎用マスターモードでは、CSをさまざまな周辺機器にアサートするのはファームウェア次第です。マスターモードでは、CSピンとINTピンをGPIOとして使用できます。SPIクロックとピンは、SPIブロックにアクティビティがない場合、アプリケーションの実行中に変更できます。以下に説明するように、3種類のマスタートランザクションと2種類のスレーブトランザクションがあります。

### SPIマスターTXのみの動作

データのみを送信する必要がある場合、SPIブロックはTX専用モードです。このモードでは：

* TxFIFOを有効にする必要があります
* RxFIFOを無効にする必要があります

この構成では、SPIブロックはTxFIFOに配置されたすべてのデータを送信します。wiced\_hal\_pspi\_tx\_data（）関数を使用して、データをTxFIFOに配置し、データをマスターとして送信します。

TxFIFOが空になると、wiced\_hal\_pspi\_tx\_data（）関数を使用してTxFIFOにさらにデータが配置されるまで、バイト境界でSPI\_CLKが一時停止します。RxFIFOでデータが受信されません。

### SPIマスターRXのみの動作

データの受信のみが必要な場合、SPIブロックはRX-Onlyモードです。このモードでは：

* TxFIFOを無効にする必要があります
* RxFIFOを有効にする必要があります

この構成では、SPIブロックはスレーブから指定されたバイトのデータを取得します。wiced\_hal\_pspi\_rx\_data（）関数は、マスターとしてデータを受信するために使用されます。

RxFIFOがいっぱいになると、データストリームが一時停止します。RxFIFOにスペースがあると、データストリームが再開されます。このモードではデータは送信されません。

### SPIマスター全二重動作

データを同時に送受信する必要がある場合、SPIブロックは全二重モードです。このモードでは：

* TxFIFOが有効になっている
* RxFIFOが有効になっている

この構成では、次の場合、SPIブロックは両方のFIFOとの間でデータを転送します。

* RxFIFOのスペース
* TxFIFOのデータ

これらの条件のいずれかが偽になると、両方の条件が再び真になるまでSPI\_CLKが一時停止します。wiced\_hal\_pspi\_exchange\_data（）関数を使用して、データを同時に送受信できます。

### SPIスレーブTXのみの動作

汎用スレーブとして、次の場合にデータがホストに転送されます。

* TxFIFOにデータがあります
* TxFIFOが有効になっている
* ホストがSPI\_CLKを切り替え、SPI\_CSNがアサートされます

条件（1）が偽で、条件（2）および（3）が真の場合、データアンダーフロー条件が発生します。wiced\_hal\_pspi\_slave\_tx\_data（）関数を使用して、データをTxFIFOに配置できます。

### SPIスレーブRXのみの動作

一般的なスレーブとして、次の場合にデータがホストから受信されます。

* RxFIFOにスペースがあります
* RxFIFOが有効になっている
* ホストがSPI\_CLKを切り替え、SPI\_CSNがアサートされます

条件（1）が偽で、条件（2）および（3）が真の場合、データオーバーフロー条件が発生します。wiced\_hal\_pspi\_slave\_rx\_data（）関数を使用して、データを受信できます。

### SPIスレーブ全二重動作

データを同時に送受信する必要がある場合、SPIブロックは全二重モードです。このモードでは：

* TxFIFOが有効になっている
* RxFIFOが有効になっている

この構成では、次の場合、SPIブロックは両方のFIFOとの間でデータを転送します。

* RxFIFOのスペース
* TxFIFOのデータ

データを同時に送受信するには、TxバッファーとRxバッファーを有効にし、wiced\_hal\_pspi\_slave\_tx\_data（）APIを使用してデータをバッファーに配置し、マスターがデータを転送した後にwiced\_hal\_pspi\_slave\_rx\_data（）を使用してRxバッファーからデータを読み取ります。

## UART

このデバイスには、2つのUARTブロックがあります。周辺UART（PUART）とHCI UART（BT UART）

### Peripheral UART (PUART)

CYW208XXのPUARTは、デバッグ印刷機能に加えて、汎用Tx / Rx UARTブロックとして使用できます。CYW208XXは、ペリフェラルUARTを任意のLHL GPIOにマップできます。PUARTは24 MHzでクロックされ、ボーレートとパリティを設定できます。PUART RXとPUART TXの両方に256バイトのFIFOがあります。[[1]](#footnote-2)

PUARTには次の機能があります。

* 最大3 Mbpsの8ビット転送
* 最大2.5 Mbpsのストップビットを含む9ビット転送
* 構成可能なフロー制御
* 受信操作時の割り込み機能

#### PUART初期化

wiced\_hal\_puart.hで定義されているwiced\_hal\_puart\_init（）関数は、PUARTブロックを初期化するために使用されます。

wiced\_hal\_puart\_select\_uart\_pads（）関数を使用して、PUARTハードウェアが使用するTX / RXおよびオプションのCTS / RTSピンを選択できます。

#### PUARTボーレート

wiced\_hal\_puart\_set\_baudrate（）関数は、ボーレートを設定するために使用されます。デフォルトのPUARTボーレートは115200です。標準的なボーレートには115200、921600、1500000、3000000 bpsが含まれますが、中間速度も利用できます。ボーレートの詳細については、データシートを参照してください。

#### PUART送信

wiced\_hal\_puart\_enable\_tx（）およびwiced\_hal\_puart\_disable\_tx（）関数は、それぞれPUARTの送信機能を有効または無効にします。PUARTを介してデータを送信するには、送信を有効にする必要があります。

wiced\_hal\_puart\_print（）関数は、TXラインを介して文字列を送信するために使用されます。

wiced\_hal\_puart\_write（）関数は、TXラインを介して1バイトを送信するために使用されます。

#### PUART受信

wiced\_hal\_puart\_enable\_rx（）関数は、受信機能を有効にするために使用されます。

wiced\_hal\_puart\_rx\_fifo\_not\_empty（）関数は、RX FIFOに使用可能なデータがあるかどうかを確認するために使用されます。

wiced\_hal\_puart\_register\_interrupt（）関数は、データの受信時に割り込みコールバックを提供します。

データを受信するには、割り込みコールバック関数を登録し、ウォーターマークを設定して、割り込みがトリガーされる前に受信するバイト数を決定し、Rxを有効にします。PUARTでデータを受信するためのコードスニペットを以下に示します。

wiced\_hal\_puart\_register\_interrupt(rx\_interrupt\_callback);

/\* Set watermark level to 1 to receive interrupt up on receiving each byte \*/

wiced\_hal\_puart\_set\_watermark\_level (1);

wiced\_hal\_puart\_enable\_rx ();

Rx処理は、割り込みコールバック関数内で行われます。追加の文字を受信するには、コールバック関数内でPUART割り込みをクリアする必要があります。

**void rx\_interrupt\_callback** (**void**\* unused)

{

**uint8\_t** readbyte;

/\* Read one byte from the buffer and then clear the interrupt \*/

wiced\_hal\_puart\_read (&readbyte);

wiced\_hal\_puart\_reset\_puart\_interrupt ();

/\* Add your processing here \*/

}

#### PUARTフロー制御

wiced\_hal\_puart\_flow\_on（）およびwiced\_hal\_puart\_flow\_off（）関数は、それぞれPUARTのフロー制御を有効または無効にします。

### ホストコントローラーインターフェイスUART

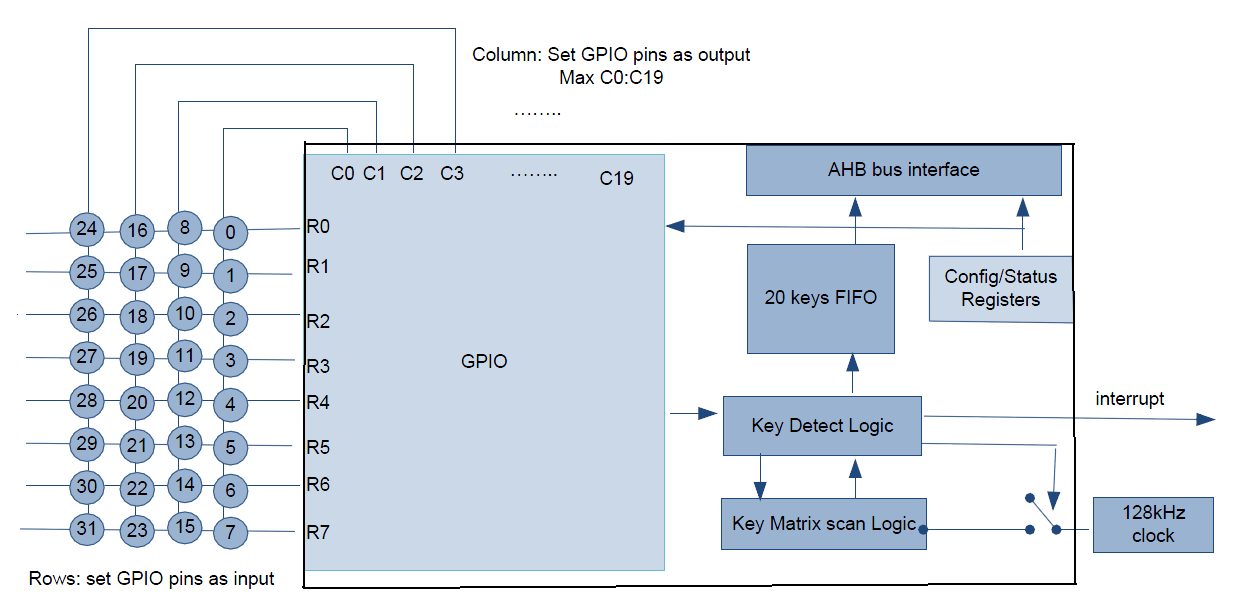
このUARTは、コントローラーモードのHCIトランスポートに使用され、デバイスのプログラミングにも使用されます。HCI UART信号は特定のI / Oパッドに固定されています。DMAは、受信するデータの長さが約64バイトを超える場合にHCI UARTパケットに使用されます。

HCI UARTインターフェイスを介したプログラミングシーケンスを理解するには、GitHub btsdk-docsリポジトリのWICED-HCI-Control-Protocol.pdfを参照してください。

## キーボードスキャナー（Keyscan）

キーボードスキャナーは、GPIOからのキープレスおよびリリースイベントを処理できるマトリックススキャナーハードウェア周辺機器です。キーボードスキャナーは、ハードウェアFIFOの上にファームウェアキューとして実装されます。キーボードスキャナーのアーキテクチャについては、図4を参照してください。

図4。CYW208XX用キーボードスキャナー



キーボードスキャナーハードウェアブロックの仕様は次のとおりです。

* 構成できる最大行数（キースキャン入力（KSI））は8です。
* 構成できる列（キースキャン出力（KSO））の最大数は20です。
* キーボードスキャナーへのすべての列ピンはスーパーマルチプレクサーです。
* キーボードスキャナーブロックは、128 KHzで160キー全体をサンプリングします。
* ハードウェアデバウンスとゴーストキー検出が利用可能です。

CYW208XXのキーボードスキャナーは、ホストマイクロコントローラーが介入する必要なく、キーを自律的にサンプリングしてバッファーレジスターに格納するように設計されています。キーアップ/ダウンイベントは、APIドライバーからのバイトストリームです。ROMファームウェアは、キーボードスキャナーハードウェアのステートマシン処理を処理します。ステートマシンはアイドル状態で始まり、いずれかのキーが押されるとスキャン状態になります。スキャン状態の間、行カウンターは0からプログラム可能な行数-1およびプログラム可能な列数-1までカウントします。特定のパッケージで利用可能な行と列の最大数については、CYW208xxデータシートを参照してください。

キーボードスキャナーの構成と利用に必要なAPIは、wiced\_hal\_keyscan.hにあります。キーボードスキャナーAPIは、次の機能のための関数を提供します。

* キーボードスキャナーの行と列の数を設定します
* 新しいキープレスイベントと保留中のキープレスイベントをチェックします
* ゴーストキーの検出を有効/無効にします
* キーボードスキャナーハードウェアブロックをリセットします
* デバイスのウェイクアップソースを提供します

関数のプロトタイプとパラメーターの詳細については、APIリファレンスガイドをご覧ください。アプリケーション開発者は、ModusToolboxのデバイスコンフィギュレーターツールを使用して、キーボードスキャナーハードウェアを構成できます。

# BLEラジオの機能

このセクションでは、CYW208XXで利用可能ないくつかのBLE無線機能について説明します。CYW208XXデバイスは、高度に統合された低電力の2.4 GHz無線トランシーバを提供し、複数の変調およびパケットフォーマットをサポートします。CYW20820は、10 dBmの送信電力で送信できる統合パワーアンプを追加します。MCUサブシステムは、周辺機器と無線の間のインターフェースを提供し、コマンドの発行、ステータスの読み取り、無線イベントの自動化とシーケンス化を可能にします。Arm Cortex-M4プロセッサは、アナログRFおよびベースバンド回路との間のデータを処理し、情報ビットを所定のパケット構造にアセンブルします。

## TX電力制御

CYW20819の最大出力は、クラス2動作で+ 4dbmです。+ 4dBm〜-16dBmのTXパワーをサポートしています。解像度は4dBmです。つまり、設定可能なTX電力レベルは{+4 dBm、0 dBm、-4 dBm、-8 dBm、-12 dBm、-16 dBm}です。

CYW20820には、クラス2.5で動作する最大電力出力が+ 10dbmの内部パワーアンプがあります。+ 4dBmから-24dBmまでのTX電力をサポートします。解像度は4dBmです。つまり、設定可能なTX電力レベルは{+4 dBm、0dBm、-4 dBm、-8 dBm、-12 dBm、-16 dBm、-20 dBm、-24 dBm}です。

BT SDKは、TX電力を制御するために次の機能を提供します。

* wiced\_bt\_set\_tx\_power（）：wiced\_bt\_dev.hで定義されているこの関数を使用して、特定の接続のデータチャネルのTX電力を設定できます。ピアデバイスのBD（Bluetoothデバイス）アドレスを入力パラメーターとして受け取ります。このAPIを使用して、BD\_ADDRを0として渡すことにより、広告チャネルのTX電力を設定することもできます。

## 共存-SECI

シリアル拡張共存インターフェース（SECI）は、サイプレス対サイプレス共存ソリューションにおける独自のサイプレスインターフェースです。WICED\_GCI\_SECI\_INおよびWICED\_GCI\_SECI\_OUT機能は、CYW208XXで利用可能なGPIOのいずれかにルーティングする必要があります。wiced\_bt\_dev.hで定義されているwiced\_bt\_coex\_enable（）およびwiced\_bt\_coex\_disable（）関数を使用すると、共存インターフェースを有効または無効にできます。共存インターフェースとそれらをCYW208XXで使用する方法については、AN214852を参照してください。

# ソフトウェア

このセクションでは、CYW208XX Bluetoothデバイスでのアプリケーションコードのビルドおよびダウンロードプロセスについて説明します。

## アプリケーションコード

すべてのothワイヤレスシステムは同じ基本的な方法で動作します。プログラマは、ネットワークスタックでBluetooth APIを呼び出すアプリケーションファームウェアを記述します。次に、スタックは無線ハードウェアと通信し、無線ハードウェアはデータを送受信します。無線ハードウェアでイベントが発生すると、スタックはイベントを作成することにより（たとえば、Bluetooth接続の反対側からメッセージを受信したとき）、アプリケーションファームウェアでアクションを開始します。アプリケーションコードは、スタックによってプッシュされたこれらのイベントを処理し、各イベントに必要な後処理を行います。ROMには、ブートシーケンス、デバイスドライバー機能、低レベルのプロトコルスタックコンポーネント、およびブートローダーがあらかじめプログラムされています。CYW208XXのROMファームウェアは、Bluetoothスタックアクティビティを処理し、無線イベントを処理します。ROMファームウェアは、Bluetoothイベントの処理において有限状態マシン（FSM）のように機能します。アプリケーション開発者は、低レベルのBluetooth機能を無視できます。

アプリケーションの最小Cファイルは次のようになります。

#include "wiced.h"

#include "wiced\_platform.h"

#include "sparcommon.h"

#include "wiced\_bt\_dev.h"

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Function Prototypes \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

wiced\_result\_t bt\_cback (wiced\_bt\_management\_evt\_t event, wiced\_bt\_management\_evt\_data\_t \*p\_event\_data);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Functions \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* Main application. This just starts the BT stack and provides the callback function.

\* The actual application initialization will happen when stack reports that BT device is

\* ready.

\*/

APPLICATION\_START ()

{

/\* Add initialization required before starting the BT stack here \*/

wiced\_bt\_stack\_init (bt\_cback, NULL, NULL); /\* Register BT stack callback \*/

}

/\* Callback function for Bluetooth events \*/

wiced\_result\_t bt\_cback (wiced\_bt\_management\_evt\_t event, wiced\_bt\_management\_evt\_data\_t \*p\_event\_data)

{

wiced\_result\_t result = WICED\_SUCCESS;

switch (event)

{

/\* Bluetooth stack enabled \*/

case BTM\_ENABLED\_EVT:

/\* Initialize and start your application here once the BT stack is running \*/

break;

default:

break;

}

return result;

}

RTOSの目的は、リソース要件が重複している複数の非同期応答時間クリティカルタスクを持つ組み込みファームウェアの記述の複雑さを軽減することです。RTOSは、アクティブ、非アクティブ、スケジュールなどのさまざまな状態にあるスレッドと呼ばれるタスクのリストを維持し、これらのタスクを優先度に基づいて実行します。SoCペリフェラルを利用するマルチスレッドアプリケーションは、共有メモリ、相互排他、ホールドアンドウェイト、循環ウェイトなどのリソースの競合がある場合に、アプリケーションのスレッド同期を処理する必要があります。

ROMファームウェアのRTOSは、Bluetooth機能を処理するための起動直後に複数のスレッドを作成し、アプリケーションスレッドに制御を与えます。現在、ModusToolboxは複数のRTOSをサポートしています。デバイスROMには、Express LogicによるThreadXが組み込まれており、ライセンスが含まれているため、CYW208XXを使用したアプリケーションの開発に最適です。複数のRTOSの使用を簡素化するために、BT SDKには、基本的なRTOS機能への統一されたインターフェイスを提供する組み込みの抽象化レイヤーがあります。**Components** > **RTOS** のWICED RTOS APIのドキュメントは、BT-SDKドキュメントページにあります。

## 低電力機能

AN225270は、CYW208XXを使用した低電力アプリケーションの設計と開発に関するガイドラインと、電力効率の良い設計を検討するためのパラメーターを提供します。

## ファームウェアのアーキテクチャ

CYW208XXデバイスは、コントローラーモードまたは組み込みモードで使用できます。

コントローラモードでは、CYW208XXデバイスはBluetoothコントローラスタックを実行し、Bluetoothホストスタックは外部ホストMCUで実行されます。CYW208XXデバイスは、ホストコントローラーインターフェイス（HCI）を使用してホストコントローラーと通信します。

組み込みモードでは、Bluetoothホストスタックとコントローラースタックの両方がCYW208XXデバイスで実行されます。コントローラスタックのすべてのコンポーネントとホストスタックのほとんどのコンポーネントは、デバイスROMにあります。APIを呼び出してROMコードにアクセスできるユーザーアプリケーションは、フラッシュにプログラムされています。フラッシュにコードがプログラムされておらず、デバイスの電源が入っている場合、HCIによって制御されるBluetoothコントローラーのように動作します。

CYW208XXデバイスは、ROMコードに「パッチ」を使用して動作を変更し、バグを修正します。これらのパッチは、すべてのModusToolboxビルドに含まれるコアパッチファイル（このドキュメントの残りの部分では一般パッチと呼ばれます）に組み込むことも、オプションのパッチライブラリ（このドキュメントの残りの部分ではパッチライブラリと呼ばれる）に組み込むこともできます）オーディオまたはメッシュネットワークアプリケーションのみに関連するパッチなど、アプリケーション固有のものです。このアプローチでは、特定のアプリケーションに必要なパッチライブラリのみを使用して、RAMの使用を最適化します。

サイプレスBTデバイスは、アプリケーションコードの無線（OTA）更新をサポートしています。OTAの詳細については、GitHub btsdk-docsリポジトリのWICED-Firmware-Upgrade-Library.pdfおよびWICED-Secure-Over-the-Air-Firmware-Upgrade.pdfを参照してください。

デバイスのプログラミングは、パッチとアプリケーションコードをフラッシュにコピーするだけではありません。また、コマンドを使用して構造化データを構成し、CPUに関数呼び出しを行わせたり、ハードウェアレジスタを構成したりします。ファームウェアをダウンロードするデフォルトの方法は、HCI UARTを介したブートロードです。デバイスROMには、外部デバイス（PCまたはMCU）からHCIコマンドを介してデータを受け取り、フラッシュに格納できるコードが含まれています。アプリケーションコードがダウンロードされると、ホストMCU（またはPC）は最初にミニドライバーと呼ばれるコードをSRAMに転送します。次に、ミニドライバーはHCIコマンドを介して追加のデータを受け入れ、それをオンチップフラッシュ（OCF）に保存します。ミニドライバーの詳細については、このドキュメントで後ほど説明します。

図5. CYW208XXファームウェアのダウンロードフロー



## メモリレイアウト

はじめにで説明したように、CYW208xx SoCはさまざまな物理メモリで構成されています。このセクションでは、CYW208xx SoCのさまざまなメモリについて詳しく説明します。ROMとフラッシュは不揮発性の物理メモリですが、SRAMはCYW208xxの揮発性メモリの大部分を提供します。ROMコードを変更するためのパッチRAMの使用については、ファームウェアアーキテクチャで説明しました。

Bluetoothデバイスの不揮発性メモリ（オンチップフラッシュまたは外部シリアルフラッシュ）は、図7に示すように5つの論理セクションに分かれています。表1に、メモリレイアウトの各セクションの詳細を示します。Bluetoothスタックのアプリケーションバッファープール要件の詳細については、AN216403-ModusToolboxアプリケーションバッファーを参照してください。

図6.不揮発性メモリのメモリ構成



表1.不揮発性メモリ構成、オンチップフラッシュ

| セクション名 | オフセット | 長さ | 変更内容 |
| --- | --- | --- | --- |
| Static Section (SS) | 0x500000 | 0x2000 | チップファームウェアによって内部的に使用される静的セクション。これには通常、出荷時のプログラミング後に変更されない構成レコードが含まれています。 |
| Volatile Section (VS) | 0x500400 | 0x1000 | アプリケーションまたはスタックがデータを外部またはオンチップフラッシュメモリに格納するために使用する最初の揮発性セクション |
| Data Section (DS1) | 0x501400 | 0x1F600 | 最初のパーティション。アプリケーションコードを保存するために使用されます |
| Data Section (DS2) | 0x520A00 | 0x1F600 | 2番目のパーティション。アプリケーションコードを保存するために使用されます |

これらの値は目安であり、実際の値は次に説明するBTPファイルにあります。

表1に記載されている論理セクションは、メモリの消去セクター境界に揃えられています。これらの詳細は、\< mtw\_path> \ wiced\_btsdk \ dev-kit \ baselib \< device> \ platforms \ 208XX\_OCF.btpにある\* .btpファイルにあります。メモリレイアウトの詳細については、GitHub btsdk-docsリポジトリのWICED-Firmware-Upgrade-Library.pdfファイルを参照してください。

.

* + 1. **\* .btpファイルの重要性**

\* .btpファイルには、ミニドライバーの場所、ボーレート、メモリセクションアドレス、その他の詳細など、ダウンロードツールで使用される情報が含まれています。また、デバイスのプログラミング中に変更できるBD ADDRESSなどの情報も含まれています。パラメータの一部は次のとおりです。

* DLConfigCrystalFreqMHz X 10000 –デバイスに接続されている水晶の周波数
* DLConfigSerialControlBaudRate –プログラミングに使用されるUARTボーレート
* ConfigDSLocation – DS1ロケーションオフセット
* ConfigDS2Location – DS2ロケーションオフセット
* DLConfigSSLocation – SSロケーションオフセット
* DLConfigBD\_ADDRBase – BDアドレス

## プログラム

前に説明したように、CYW208XXデバイスは通常、コントローラーモードまたは組み込みモードの2つの構成で使用できます。

コントローラーモードでは、ホストは通常​​、BluetoothデバイスのSRAMまたはOCFに一般的なパッチなどのコードをダウンロードします。コードがSRAMにダウンロードされた場合、リセット時にSRAMの内容が失われるため、起動ごとにコードを実行する必要があります。

組み込みモードでは、デバイスの組み込みスタックが一部のホスト機能に使用され、アプリケーションコードをデバイスのOCFにダウンロードする必要があります。

### ファイルフォーマット

アプリケーションコードは通常、*.hex*または*.hcd*形式で保持されます。通常、\*。hcdファイルはSRAMのダウンロードに使用され、\*。hexファイルはフラッシュのダウンロードに使用されます。中間ステップとして、ソースコードといくつかの構成値が.cgs（構成ソース）形式に変換され、次に.hexまたは.hcd形式に変換されます。.hcdファイルは、ファイルサイズが小さく、ペイロード形式がHCIコマンドに似ているため、外部MCUがコントローラーモードでBluetoothデバイスをプログラムする必要がある場合に役立ちます。Intel .hexファイルは、フラッシュへのプログラミング時に使用されます。次の図は、ファイル変換プロセスを示しています。

デバイスにダウンロードするには、これらのファイルを解析してHCIコマンドに変換する必要があります。ModusToolboxでは、これはChipLoadツールによって行われます。



#### \* .hcd形式の詳細

hcdファイルは、HCIコマンドとして直接解析および解釈できるバイナリファイルです。hcdファイルには、ホストが追加する必要のあるパケットタイプを除くすべてのコマンドがバイナリ形式で含まれています。hcdファイルの形式は次のとおりです。

* 最初の2バイトはコマンドIDです。たとえば、HCI\_WRITEコマンドは0x4C、0xFCで表されます。
* 次のバイトはコマンドペイロードの長さです。たとえば、0x06は、コマンドを完了するためにさらに6バイトが続くことを示します。
* コマンドのペイロード。

ファイルをHCIコマンドに変換するには、HCIパケットタイプのみを追加する必要があります。たとえば、ホストからコマンドを送信する場合、0x01はHCIコマンドの前に配置して、イベントではなくコマンドであることを示す必要があります。

次のWRITE\_RAMコマンドは例です。

01 4C FC nn xx xx xx xx yy yy yy…

このWRITE\_RAMコマンドでは：

* nnは4 + Nで、4つのアドレスバイトとNペイロードバイトを表します。
* xx xx xx xxは、4バイトの絶対RAMアドレスです。
* yy yy yy…は、アドレス指定されたRAMロケーションにロードされるNペイロードバイトです。

各WRITE\_RAMコマンドに対する次の応答は、200ミリ秒以内に予想されます。

04 0E 04 01 4C FC 00

#### \* .hex形式の詳細

.hexファイルは、Intel I32HEXの規則に従います。形式は、ASCIIキャリッジリターンとラインフィード（0x0D、0x0A）で区切られたレコードで構成されます。.hexファイルの形式は次のとおりです。

* 最初の文字は開始コード '：'（ASCII 0x3A）です
* 次のバイトはペイロードの長さを示します。たとえば、「FF」は255を示します。
* 次の4バイトは16アドレスビットを示します。たとえば、「1000」はアドレス0x1000を表します。アドレスのタイプは、コマンドのタイプによって異なります。
* 次のバイトはコマンドのタイプを示します。
* 「00」はデータ用で、アドレスフィールドは宛先アドレスの下位16ビットを表します。
* 「01」はファイルの終わりを示します。ペイロードは0で、アドレスフィールドは「0000」に設定されています。
* '04'は拡張アドレスを示します。アドレスフィールドはアドレスの下位16ビットを示し、ペイロードはアドレスの上位16ビットを表します。
* 32ビットのアドレスペイロードの場合は「05」。アドレスフィールドは「0000」に設定され、ペイロード長は「04」に設定され、ペイロードは32ビットアドレスとして解釈されます。これは、LAUNCH\_RAM（後述）の宛先を示すためによく使用されます。

#### \* .cgs形式の詳細

.cgsファイルには、構成パラメーター、パッチコード、およびアプリケーションコードが含まれています。アプリケーションのソースコードから生成された.elfファイルは.cgs形式に変換され、\< mtw\_path> \ wiced\_btsdk \ dev-kit \ baselib \ 208XXA1 \ internal \< device> \にあるパッチ.cgsファイルに追加されます。 patch \ patch.cgs。

.cgsファイル内の構成パラメーターの定義は、\< mtw\_path> \ wiced\_btsdk \ dev-kit \ baselib \ 208XXA1 \ internal \< device> \ configdef< device> .hdfにあります。

注：< device> -ディレクトリパス内のデバイスは、20819A1または20820A1を表します。

水晶周波数、UART構成、BD\_ADDR、ローカル名、RSSI構成などのパラメーターは、パッチ.cgsファイルを使用して変更できます。

### ビルド中のファイル生成

ファイル変換プロセスは次のとおりです。

1. ソースファイル（.c、.h）はコンパイルされ、GNUツールを使用して.o（オブジェクト）ファイルに変換されます。
2. ライブラリのソースは、ライブラリの.aファイルにリンクされています。
3. これらは、事前にビルドされたライブラリファイルとともに、GNUツールによって.elf（実行可能でリンク可能な形式）ファイルにリンクされます。
4. .elfファイルは、\< mtw\_path> \ wiced\_btsdk \ dev-kit \ baselib \< device> \ make \ scripts \ wiced-gen-cgs.plスクリプトを使用して.cgs（構成ソース）ファイルに変換されます。ROMパッチコードとプラットフォーム.cgsファイルも追加されます。
5. 次に、\< mtw\_path> \ wiced\_btsdk \ dev-kit \ btsdk-tools \< OS> \ CGS \ cgs.exeツールを使用して、.cgsファイルをIntel hex形式に変換します。cgs.exeアプリケーションは、前のステップで作成された.cgsファイルや.btpファイルなどの入力を受け取ります。cgs.exeアプリケーションは、BD ADDRESSなどのランタイム引数を取り、*.btp*ファイルのBD ADDRESSを上書きできます。これは、異なるアドレスで複数のデバイスをプログラミングするときに役立ちます。
6. hexファイルは、ツール\< mtw\_path> \ wiced\_btsdk \ dev-kit \ btsdk-tools \< OS> \ IntelHexToBin \ IntelHexToHCD.exeを使用してhcdファイルに変換されます。

### 回復プロセス

場合によっては、CYW208XXデバイスが不明な状態になるか、HCIボーレートがデフォルトから変更された可能性があります。このような場合、HCI UARTを使用してデバイスをプログラムすることはできません。デバイスを正しくプログラムするには、デバイスをリカバリモードにする必要があります。これは、デバイスのリセット中にHCI UART CTSラインをアサートすることによって行われます。

Bluetoothデバイスはリセットされると、電源投入時にHCI UART CTSピン（リカバリピン）のステータスをチェックします。リセット中にリカバリピンがアサート（LOW）されると、デバイスはいわゆるリカバリモードに入ります。このモードは、HCI\_RESETコマンドのビットパターンについてRXラインをチェックすることにより、UARTボーレートの検出を試みます。検出されると、HCI\_RESET応答は同じボーレートで与えられます。

このモードでは、ほとんどのHCIコマンドは応答しません。このモードでデバイスにダウンロードするには、HCI\_DOWNLOAD\_MINIDRIVERに対する「応答なし」を無視して、ダウンロード手順を続行します。リセット後にCTSピンがHighの場合、デバイスはOCFをチェックし、保存されている構成を適用します。通常、デフォルトのボーレートですべてのHCIコマンドを受け入れる準備ができたモードで終了します。設定が利用できない場合、デバイスは自動ボーモードに入ります。アプリケーションコードはSRAMに直接ロードして実行することも、シリアルフラッシュに実行して、次回の起動時に実行することもできます。

### ミニドライバー

ミニドライバーは、ROMコードから独立した別のFWです。アプリケーションコードをオンチップフラッシュにダウンロードするために使用されます。ミニドライバーには、WRITE\_RAMやREAD\_RAMなどの特定のHCIコマンドを解釈し、それらのコマンドを実行するためのコードが含まれています。オンチップフラッシュにコードをダウンロードするには、最初にSRAMからミニドライバーを作成して実行する必要があります。これにより、HCIコマンドがさらに解釈され、コードがフラッシュの正しい場所に書き込まれます。コードのダウンロードが完了するとすぐに、ミニドライバーは不要になったため破棄されます。ミニドライバーは、インターフェイスがUARTからSPIなどに変更された場合に異なります。デフォルトのミニドライバーは\< mtw\_path> \ wiced\_btsdk \ dev-kit \ baselib \< device> \ platforms \ minidriver-< device> -uart-patchram.hexにあります。

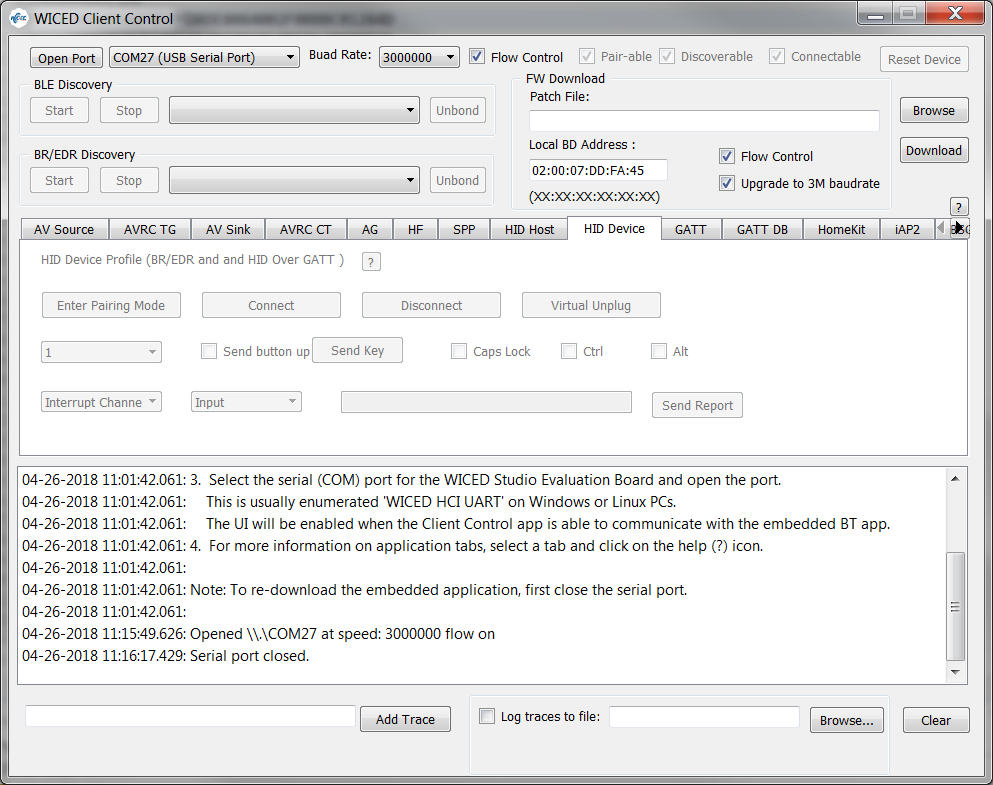
## プログラミング用ツール

ダウンロードプロセスとHCIコマンドフローの詳細を理解するには、GitHub btsdk-docsリポジトリのWICED-HCI-Control-Protocol.pdfを参照してください。

* + 1. **ClientControl**

ClientControlは、PCでBluetoothホストとして機能するWICEDアプリケーションです。HCI UARTを使用してBluetoothデバイスに接続し、WICED HCIプロトコルを介して通信します。このアプリケーションを使用して、アドバタイズの開始/停止、接続要求の送信、HIDプロファイルのテストなどのコマンドを送信し、ファームウェアをデバイスのSRAMにダウンロードすることで、CYW208XXデバイスをテストできます。このツールはデバイスのSRAMにダウンロードするため、ミニドライバーは必要ありません。

図. ClientControlウィンドウ



Windows、macOS、およびLinux用のClientControlアプリケーションを起動するには、ModusToolboxでBT-SDKアプリケーションを選択し、[**Quick Panel** > **Tools** > **ClientControl**]をクリックします。

### チップロード

ChipLoadは、Bluetoothデバイスにファームウェアをダウンロードするために使用されるコマンドラインユーティリティです。.hexファイルや.btpファイルなどの入力を使用して、ダウンロードデータとダウンロード構成をそれぞれ取得します。\< mtw\_path> \ wiced\_btsdk \ dev-kit \ btsdk-tools \< OS> \ ChipLoad \にあります。

ChipLoadツールはIntel hex形式をサポートしていることに注意してください。

### DetectAndId

DetectAndIdツールは、検出とIDの2つのタスクを実行します。

最初に、このツールはサイプレスBluetoothデバイスが存在するシリアルポートを検出しようとします。これは、さまざまなボーレートで利用可能なポートにHCI\_RESETコマンドを送信し、適切な応答を待つことによって行われます。シリアルポートがこのツールへの入力として既に指定されている場合、ポート検出部分はスキップされます。

2番目のタスクは、接続されたデバイスのIDです。これは、チップIDを読み取るためにWICED HCIコマンドを送信することによって行われます。受信したIDは、このパス< mtw\_path> \ wiced\_btsdk \ dev-kit \ baselib \< device> \ platforms \ CYW208XXA1\_IDFILE.txtのチップ固有IDファイルに存在する予想チップIDと比較されます。これにより、適切なデバイスのみがプログラムされます。これは、他のツールがアプリケーションコードをダウンロードするために使用できます。

### サイプレスプログラマ

ヒノキ プログラマーは、CYW208xxベースの評価ボードのプログラミングに使用できるGUIアプリケーションです。CYW208xxのフラッシュをプログラム、消去、および検証するオプションを提供します。サイプレスプログラマは、Bluetoothアプリケーションのビルドプロセスによって生成される\* .hexファイルを想定しています。HCI UARTポートを選択すると、ボードをプログラムできます。詳細については、サイプレスプログラマリファレンスガイドとリリースノートを参照してください。

# まとめ

CYW208XXは、モノのインターネット（IoT）デバイス用のワイヤレス入力、ウェアラブル、医療、ビーコンアプリケーションのアプリケーションに最適なソリューションです。コスト重視のデバイスの設計を容易にするために、Arm Cortex-M4プロセッサは、緊密に結合されたシステムコンポーネントを実装し、Bluetoothパケットの処理と周辺システムの機能を大幅に改善しながら、プロセッサ領域を削減します。同じシャーシを維持しながら、複数の市場に対応できる独自の差別化機能を提供します。このガイドは、CYW208XXデバイスの機能と周辺機器を探索するプログラマーに役立ちます。このガイドでは、さまざまなCYW208XX SoCペリフェラルの使用方法と、アプリケーションの開発および製品の設計時にハードウェアレベルの制約を克服する方法について説明します。

# 関連文書:

|  |  |
| --- | --- |
| アプリケーション注意事項 | |
| [AN225684 – CYW208XXの概要](https://www.cypress.com/an225684) | CYW208XX Bluetooth SoC、およびModusToolbox IDEのデバイスを使用して最初のBLEアプリケーションを構築する方法について説明します |
| [AN225948-CYW20819 / 20820ハードウェア設計ガイドライン](https://www.cypress.com/documentation/application-notes/an225948-cyw2081920820-hardware-design-guideline) | このドキュメントには、ボードスタックアップとインピーダンス制御要件を含む基本的なレイアウトガイドライン、重要なコンポーネントのコンポーネント配置と推奨部品、ルーティングガイドライン、およびさまざまな電源トレースの推奨トレース幅が含まれています。 |
| [AN225270-CYW208XX低電力ガイドライン](https://www.cypress.com/an225270) | このドキュメントは、CYW208XXを使用した低電力アプリケーションの設計と開発に関するガイドラインと、電力効率の良い設計を検討するためのパラメーターを提供します。 |
| [AN216403 –アプリケーションバッファープール](https://cypresssemiconductorco.github.io/btsdk-docs/BT-SDK/WICED-Application-Buffer-Pools.pdf) | このドキュメントでは、アプリケーションで使用されるバッファーと、WICED Bluetooth（BT）スタックの上位層スタックについて説明します。 |
| [AN214852-サイプレスからサイプレスへのソリューションとサイプレスからサードパーティのチップ間の協調的な共存インターフェース](https://www.cypress.com/documentation/application-notes/an214852-collaborative-coexistence-interface-between-cypress-cypress) | このアプリケーションノートでは、CYW43XXのコラボレーティブ共存ハードウェアメカニズムとアルゴリズム、およびサイプレスからサイプレスへのソリューションまたはサイプレス間のCOEX配線をサードパーティのチップに接続する方法について説明します。 |
| サンプル コード | |
| ModusToolbox IDEを使用したコード例の包括的なコレクションについては、サイプレスGitHubリポジトリにアクセスしてください | |
| デバイスと評価キットのドキュメント | |
| [CYW20819デバイスデータシート](https://www.cypress.com/datasheet/CYW20819) | [CYW920819EVB-02評価キット](http://www.cypress.com/CYW920819EVB-02) |
| [CYW20820デバイスデータシート](https://www.cypress.com/datasheet/CYW20820) | [CYW920820EVB-02評価キット](http://www.cypress.com/CYW920820EVB-02) |
| [CYBT-213043-02: EZ-BT™ MODULE](https://www.cypress.com/documentation/datasheets/cybt-213043-02-ez-bt-module) | [CYBT-213043-MESH EZ-BT™モジュールメッシュ評価キット](https://www.cypress.com/documentation/development-kitsboards/cybt-213043-mesh-ez-bt-module-mesh-evaluation-kit) |
| ツールのドキュメント | |
| [ModusToolbox IDE](http://www.cypress.com/modustoolbox) | IoTデザイナー向けのサイプレスIDE |
| [サイプレスプログラマのドキュメント](https://www.cypress.com/products/psoc-programming-solutions) | このページでは、サイプレスプログラマGUIユーザーガイドとサイプレスプログラマリリースノートを提供します。 |

著者と貢献者について

|  |  |
| --- | --- |
| 名前: | Abhishek Khinvesra, Ranjith Kumar |
| 役職: | スタッフアプリケーションエンジニア; シニアアプリケーションエンジニア |
| 経歴: | Abhishek Khinvesraは、VIT大学で電子機器および計装の技術学士号を取得しています。Ranjith Kumarは、VIT大学で組み込みシステムの技術の修士号を取得しています。 |

改訂履歴

＜＜★Document Number: 002-26546

| 版 | Engineering Change Notification (技術変更届) | 発行日 | 変更内容 |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |

1. BT SDKのドライバーの可用性に応じて。BT SDK v1.3以降、SDKはパリティ選択をサポートしていません。 [↑](#footnote-ref-2)