

## 長距離通信タイプ Bluegiga WT41 Bluetoothモジュール



### かんたんスタートアップガイド

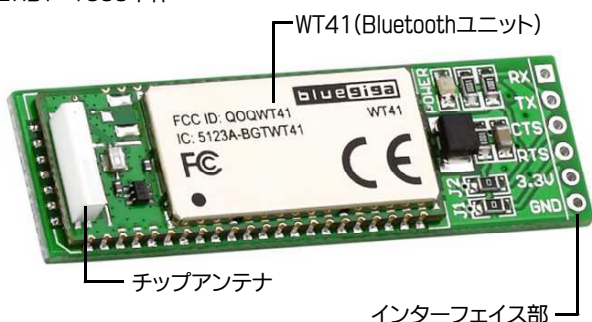
本ガイドは、RBT-1000を簡単に使い始めるための基本的な使い方について解説したスタートアップガイドです。

より詳しい使い方についてはBluegiga社WT41のマニュアルをご覧ください。

(C)2014 マイクロテクニカ

## 製品本体

### ■RBT-1000本体



## 製品の概要

長距離通信タイプBluetoothモジュール(型式:RBT-1000、以下型式で記載)は、外付けアンテナを付けることなく、通信距離1kmの通信(見通し内通信)※1が可能なBluetooth 2.1+EDR、クラス1のBluetoothモジュールです。

標準構成では、本体にiWARPファームウェアが書き込まれており、BluetoothのSPPに対応しています。非同期式シリアル通信(UART)の信号を無線化して、リモートデバイスと通信できます。パソコンと通信を行う場合、本機と接続すると自動的にパソコン側には仮想COMポートが作られ(Bluetoothのスタックソフトウェアによる)、OS上からは旧来のCOMポートと同様にアクセスが可能となります。RBT-1000同士での通信も可能です。

本体を制御するコマンドは簡単なASCIIコード(文字列)を採用していますので、PICマイコン等から簡単に制御できます。

RBT-1000は電波法に基づく技術基準適合証明を取得済みですので、本機を使用すれば面倒な技適をお客様が取得することなくBluetooth通信機能をお客様のアプリケーションに追加できます。また、Bluetooth SIGのライセンスも取得しています。

RBT-1000は主にシリアル通信(UART)を無線化することに特化した構成になっています。

※1:障害物や周囲の電磁環境によって通信距離は変わります。1kmの通信は障害物のない環境で、かつ干渉する電波が近接界にない場合の理想通信距離です。

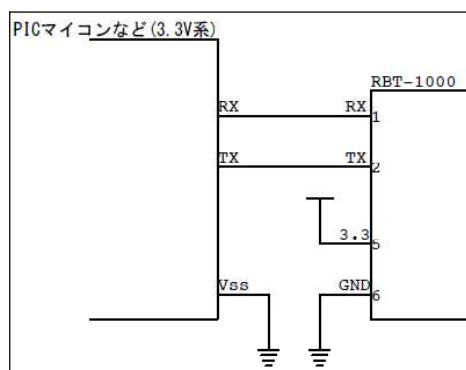
## インターフェイス端子の概要

ピン	端子記号	種類	内容
1	RX	O	マイコン側UART 受信データ
2	TX	I	マイコン側UART 送信データ
3	CTS	O	マイコン側UART 送信可 (J2)
4	RTS	I	マイコン側UART 送信要求 (J1)
5	3.3V	P	電源ピン +3.3V 要安定化
6	GND	P	電源ピン GND

※種類: I→入力 O→出力 P→電源

UART信号線については、シルク印刷で記載の信号内容は、マイコンと接続するピンの名称が印刷されています。例えば1ピンのRXピンは、マイコンと本機を接続する時、この1ピンをマイコンのRXピンと接続するという意味になります。(RBT-1000のデータ受信ピンという意味ではありません。クロスで配線しないようご注意ください。よって端子記号と、一般的なI/O方向が逆になっています。)

下記を参考にマイコンなどと接続してください。



ロジック信号の電圧レベルは、0V-3.3Vです。(Vp=+3.3V)

よって、5V系の電圧を使うマイコンとは直結できませんのでご注意ください。3.3V系のマイコンとは直結できますが、5V系マイコンと接続する場合には、レベルシフタ回路を介して接続してください。

### ■ハードウェアフロー制御の接続

RBT-1000の基板には、J1及びJ2の面実装ジャンパーの実装されたパターンがあります。これらは、RBT-1000のインターフェイス端子3ピン及(CTS)び4ピン(RTS)に、BluetoothモジュールのWT41のCTS及びRTSを接続するかどうかを設定するものです。

工場出荷時の状態では、J1及びJ2は未ショートです。CTS/RTSを使う場合には、パッドを半田でショートしていただく。

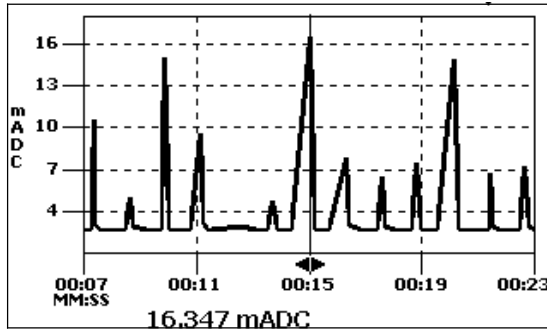
- ・J1: WT41のRTSピンとRBT-1000の4ピンを接続します。
- ・J2: WT41のCTSピンとRBT-1000の3ピンを接続します。

## 電源について

RBT-1000の電源電圧は、+3.3Vです。許容誤差は±5%以内です。電源には安定化された電源をご使用下さい。リップルの重畳している電源は使えません。スイッチング電源やスイッチングレギュレーターを使用している場合、出力電圧の波形に発振波形が乗っていることがあり、それらがパフォーマンスに影響を及ぼすことがあります。

RBT-1000の消費電流は待機時平均で約4mA程度ですが間欠で待機時に約20mA程度の電流が流れます。さらにBluetooth通信を行っている時には、最大で170mA程度(平均60mA程度)の電流を消費しま

す。よって、電源は最低でも300mA以上を流せるように設計されている必要があります。間欠で消費電流が大きくなるので、電圧レギュレータを使用する場合には負荷過渡応答のよいものを選定してください。



上のグラフは待機時の消費電流を実測した際のグラフです。横軸が時間で縦軸が消費電流です。ピーク時で実測16.3mA程度の電流が流れています。その他にも小さいな電流変動があり、主に大きな電流変動は5秒間隔程度で起こっていることが実測により観測できます。

## UART信号について

### ■UART信号の規格

UART信号は、ビット番号が小さい方から送るLSBファーストです。デフォルトのUARTパラメータは次のようになっています。

- ・通信速度: 115200bps
- ・データビット: 8ビット長
- ・ストップビット: 1
- ・パリティ: なし
- ・フロー制御: ハードウェア有効

原則としてこの初期設定値は変更できません。

### ■ロジック信号の電圧レベルについて

Vccが3.3Vの時、UART信号電圧レベルは0V-3.3Vとなりますが、電氣的仕様は下記の通りです。

- ・ViH[ロジック信号Hレベル入力電圧]  
最小値+2.6~最大Vcc
- ・ViL[ロジック信号Lレベル入力電圧]  
最小値Vss~最大0.25V

RBT-1000の電圧レベルは3.3V系のため、5V系のとでは直結できません。(直結した場合、Vccよりも高い電圧がロジック信号線に印加された場合、RBT-1000が破損します)

### ■CTS線及びRTS線について

RBT-1000とUART機器(例えばマイコン等)の基本的なUART接続は4線式のハードウェアフロー制御通信となります。

TXとRX、RTSとCTSをクロスして配線することで、UART機器との通信ができます。UART通信ではTXとRXだけの2本のデータ線だけで通信を行うこともできますが、その場合には、バッファの空き状態に応じた通信制御はできません。2線式で通信を行う場合には、バッファがフルになってしまわないよう、RBT-1000にデータを溜めずに、データを逐次読み出す必要があります。また、RBT-1000のCTSピンとRTSピンを短絡(ショート)させてご使用下さい。

ハンドシェイク通信を行う場合、CTS(送信可能)線及びRTS(送信要求)線で制御を行います。

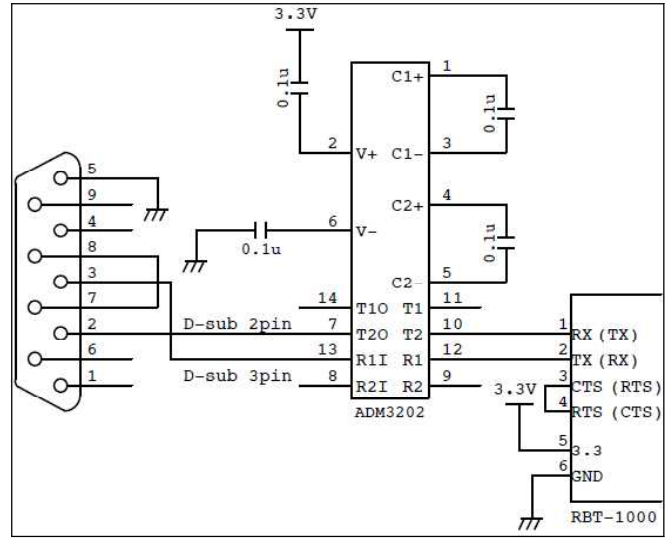
4ピンのRTSとシルク印刷で記載されたピンはUARTホスト機器のR

TSピンと接続し、本機側では入力ピン(CTS信号入力)となっています。

このピンがLレベルの時はRBT-1000はデータを出力します。このピンをHレベルにするとRBT-1000はホスト機器に対して出力するデータがある場合でも出力せずデータをバッファに保持します。ホスト機器は受信の準備が完了したらこのピンを速やかにLレベルにしてデータを受信します。フロー制御を使わない場合には、3-4ピンを短絡してお使い下さい。

### ■RBT-1000をパソコンのRS232Cポートと接続する場合

パソコンのRS232Cポートと接続する場合には、ADM3202などのレベル変換ICを介して接続します。RBT-1000のロジック電圧レベルは0V-3.3Vなので、レベル変換ICの電源電圧は必ず+3.3Vとします。配線例を下図に示します。下図は2線式通信の場合の例です。



## RBT-1000のデータ転送モード

RBT-1000には2つの転送モードがあります。

1つは"コマンドモード"、もう1つは"データモード"です。通常電源を投入直後で、他のBluetooth機器との通信を行っていない時は、コマンドモードになっています。

コマンドモード、主にシリアルコマンドによってRBT-1000の設定を行うモードです。ASCIIコードで各種コマンドを送信します。このコマンドモード実行中は、Bluetoothによって他の機器とのデータ通信は行えません。

Bluetoothによって、別の機器(リモートデバイスと呼びます)とコネクションを確立すると、RBT-1000は自動的に(デフォルトの設定では)データモードに移行します。データモードは1つのリモートデバイスとリンクを確立後、高速にデータ転送を行うデータ転送専用のモードです。RBT-1000に送られたデータはすべて、Bluetooth通信によって接続されたリモートデバイスへと送信されます。またリモートデバイスからBluetoothによる無線通信で送られて来たデータはすべてUARTのTXより出力されます。通信は高速でバースト的に行われるため、バッファの空き状態を常に監視しながら通信を継続する必要があり、このモードを使用する場合には原則としてハードウェアフロー制御通信を行うことが推奨されます。フロー制御を行わず(CTS/RTS線を短絡しての通信)2線式通信で通信をする場合には、データの取りこぼしが発生することがありますので注意が必要です。(低速な通信であれば取りこぼしの頻度は少ないです。)

この2つのモード間の状態遷移は次のようになっています。

### ①コマンドモードからデータモードへの移行

RBT-1000がコマンドモードからデータモードに移行する要因は次のものがあります。

1つは、CALLコマンドにより、リモートデバイスと接続されてコネクティブイベントが発生した時です。CALLコマンドは、Bluetooth IDを指定してリモートデバイスに接続するものです。これによって接続が完了するとコネクティブイベントが発生します。コネクティブイベントが発生するとRBT-1000は自動的にデータモードへ移行してデータ通信ができるようになります。

もう1つはリモートデバイス側(RBT-1000とBluetooth経由で接続する機器、PCなど)からSPPの仮想COMポートなどを開いた場合です。この時にはリングイベントが発生します。リングイベントが発生すると、RBT-1000はリモートデバイスのBluetooth ID、RFCOMMのチャンネル番号などを返した後、自動的にデータモードに移行します。

### ②データモードからコマンドモードへの移行

RBT-1000がデータモードからコマンドモードに移行する要因は次のものがあります。

1つはエスケープシーケンスコマンドをUARTで、UARTホストが送信する方法です。UARTホストはRBT-1000に対してエスケープシーケンスで指定された文字(ASCII)を3回送る(3バイト)ことでRBT-1000をコマンドモードに移行させることができます。工場出荷時の状態では、エスケープシーケンスの文字は"+"(2Bh)に設定されています。この"+"を3回連続します。またエスケープシーケンスの文字を送信する場合には必ず前後に1秒の何もデータを送信しない時間が必要です。

<1秒待ち>+++<1秒待ち>

この1秒待ちの間にエスケープシーケンスの文字以外のデータを送ってしまうと正しくコマンドモードには移行しません。

コマンドモードに正しく移行すると、RBT-1000は"READY."の文字列を返します。(文字列はCRとLFで終端されています。)なお、このエスケープシーケンスの文字は、"SET CONTROL ESCAPE"コマンドで任意のASCIIコードに変更できます。

もう1つの方法は、リモートデバイスがRBT-1000との接続を切断した場合です。この場合、RBT-1000ではNO CARRIERイベントが発生します。このイベントが発生すると、自動的にコマンドモードに移行します。例えばパソコンとRBT-1000が接続されていて、パソコン側からRBT-1000の接続を切断した場合がこれに相当します。

### ③エスケープシーケンスによるコマンドモードからデータモードへの移行(リモートデバイスとの接続が確立されている時)

先の②で、リモートデバイスとの接続が確立している状態で、エスケープシーケンスによって、コマンドモードに移行した場合、リモートデバイスとの接続はそのまま確立されています。単にRBT-1000がコマンドモードになっただけであって、Bluetooth通信によるリモートデバイスとの接続は切断されたわけではありません。この場合には、コマンドモードからデータモードへSELECTコマンドで移行することができます。最初にLISTコマンドで、現在RBT-1000と接続が確立されている接続先リストを表示させます。もしLISTコマンドで"LIST 0"が返るようならば接続を維持しているリモートデバイスが存在しませんので、SELECTコマンドではデータモードへの移行はできません。LISTコマンドで接続を維持しているリモートデバイスが返るようならば、そのリスト番号を引数として指定してSELECTコマンドを発行すると、RBT-1000はコマンドモードからデータモードへ移行します。

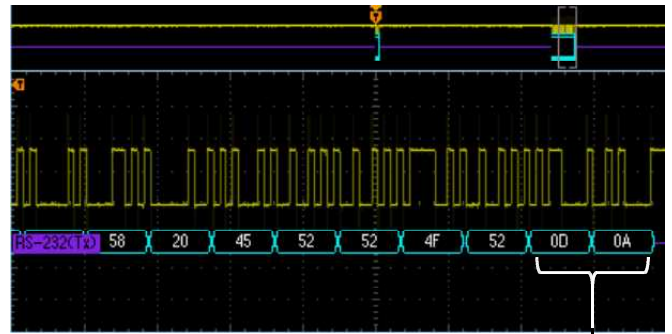
このようにRBT-1000には2つのモードがあり、それぞれのモードを切り替えながら使うことになります。

## コマンドモードでのきまり

### ■コマンドはキャリッジリターンで終端、戻り値にはCRとLFが付加

コマンドモードは、RBT-1000の設定などを行うモードです。UART通信で各種コマンドをRBT-1000に送信します。コマンドはすべて文字列で、コマンドの最後は必ずキャリッジリターン(0Dh)で終端する必要があります。

RBT-1000からの戻り値にはデータの最後にキャリッジリターン(0Dh)とラインフィード(0Ah)の2バイトが付加されます。



CRとLFが付加されている

### ■コマンド、引数は必ず半角英数大文字

RBT-1000のすべてのコマンドは大文字です。小文字でも認識はしますが、間違いが発生しやすいためコマンドは大文字で統一しておくことを強くお奨めします。

### ■マニュアルではコマンドは"でくくり、戻り値は斜体で記載

本マニュアルではコマンドは"でくって記載します。また、各コマンドは必ずキャリッジリターンで終端されますので、特別な事項がある場合以外は、キャリッジリターンは省略します。また半角スペース(20h)が入る場合も特にわかりにくい場合を除いては数値は記述しません。

なお、特殊な場合やわかりにくい部分などで文字列以外のコマンドを示したい場合には、< >を付けて記載します。例えば半角スペース(20h)は、<sp>と記載します。

RBT-1000の戻り値は"でくった斜体文字で記載します。

### ■RBT-1000の初期設定ではUART信号はエコーバックする

RBT-1000は、コマンドに対する戻り値もすべて文字列で返ります。またRBT-1000はデフォルト設定で送信した文字列のエコーバックが有効になっています。よってUARTホストからRBT-1000に対して送信した文字列は、すべてそのまま返ってきます。キャリッジリターンを送信した直後にそのコマンドが実行され、戻り値がある場合には戻り値が返ります。エコーバックを無効にするには"SET CONTROL ECHO 0"を送信することで設定できます。

### ■コマンドや引数が正しくない場合には"SYNTAX ERROR"が返る

コマンドや引数の内容に誤りがある場合には、多くのコマンドで"SYNTAX ERROR"が返ります。これが返った場合には、そのコマンドは実行されていませんので再度コマンド文字列に誤りが無いか、引数に誤りがないかを確認して再実行する必要があります。

## とりあえずBluetoothで通信してみる

RBT-1000を使う上では様々な設定が必要になります。もっとも一般的なものはUARTの通信速度やパリティビットなどです。そのほか、Bluetoothでは、特有の設定などもあり、その使い方を詳しく理解するにはある程度の専門的知識が必要です。このRBT-1000に使用されているWT41は大変多機能なBluetooth通信モジュールのため、設定項目や設定に関わるコマンドもたくさんあります。しかし実際によく使うコマンドはいくつかに限られるため、まずは最も簡単な方法でBluetooth通信を行って、動作を体験することになります。複雑な設定部分はすべて初期設定の状態ですべて使ってみます。

パソコンと、RBT-1000との間で1対1の通信を行います。コマンドモードでコマンドを設定後、Bluetoothリンクを確立してデータモードで通信を行います。本項で解説する方法で機能として十分であれば、このまま、この方法で使用しつづけても問題ありません。

### ①RBT-1000とUART制御機器を接続する

最初のRBT-1000のUART側端子に、UART経由で制御しデータ通信を行いたいマイコン等の機器を接続します。インターフェイスの電気的仕様については本書2ページに記載されていますので、必ずこちらをお読みの上、仕様を逸脱しないよう設計をお願い致します。最初は、UART-RS232C変換回路などを使ってパソコンと接続してUARTコマンド送受信をすると仕組みがわかりやすいです。

### ②電源の給電

RBT-1000の電源は+3.3Vです。余裕をみて200mA以上は流せる電源をご用意下さい。なお電源投入する場合には、必ずRBT-1000と、UART経由で接続しているマイコンなどの機器の電源を同時に入れてください。UART機器側の電源がRBT-1000より先に投入されるとUART信号線にロジック信号のHレベルの電位(3.3V)が印加されることになり、RBT-1000側がラッチアップして故障しますので、ご注意ください。

### ③UART通信を確認してみよう

RBT-1000が正しく動作しているか、UART通信ができていかなど動作確認をするのに便利なコマンドが"AT"コマンドです。このコマンドをUART通信でRBT-1000が受信すると、正しく動作している場合には"OK"を返します。

```
AT<cr>
AT<cr><LF>
OK<cr><LF>
```

最初は送信したコマンドに対するエコーバックです。

この"OK"を受信することでRBT-1000との通信が正しく確立していること及びRBT-1000が正しく動作していることの確認ができます。

### ④Bluetooth搭載Windows7パソコンと接続(ペアリング)

RBT-1000の準備ができたら早速パソコンと無線通信をするために、接続を開始しましょう。RBT-1000の周囲にBluetooth搭載パソコンを設置してください。Bluetooth通信機能が内蔵されていないパソコンの場合には、別途Bluetooth dongleを取り付けてBluetooth機能を付加させます。(当方でもオプション品として販売しています。)

次の手順でRBT-1000とパソコンとを無線接続します。Bluetoothにおいて、リモートデバイスと接続を行う操作のことをペアリングと言います。ペアリングにはデバイスによってはパスキーと呼ばれるパスワードを入力する必要があるものもありますが、RBT-1000は工場出荷時の状態ではパスキーは設定されていません。後からコマンドでパスキーを設定することができます。

## ■Windows7/8などのパソコンと接続する

以下の手順はWindows7において、一般的なBluetooth Stack for Windows by Toshibaがインストールされている環境での接続例を紹介しています。OSやBluetoothのスタックソフトウェアによって画面や設定手順が異なる場合があります。Bluetoothでの接続設定は一般的なヘッドセットなどのBluetooth周辺機器と同じです。

- 1 パソコンのBluetooth通信機能を有効にします。
- 2 Bluetooth通信機能が有効になっている場合、タスクバーの通知領域にBluetoothのアイコンが表示されています。



Bluetoothアイコンをクリックすると、設定画面が表示されます。

- 3 新規にRBT-1000を検索してパソコンに登録しましょう。パソコン環境によって操作画面が異なりますが、Bluetoothで周辺の通信可能デバイスを検索してください。



Windows8の場合には、"PC設定"から"デバイス"の項目を選択して、"デバイスの追加"ボタンを押します。

※"PC設定"はチャームバーを表示し、"設定"をクリックし、チャームバー一番下の"PC 設定の変更"をクリックします。



- 4 本製品は"WT41-A"というデバイス名で検出されます。



デバイス名は任意の文字列に後からコマンドで変更することができます。ここでは、検出された"WT41-A"を選択してください。Windows8などでパスキーが求められた場合には、何も入力せずに続行してください。

5 RBT-1000はSPPIに対応しているため、パソコンはRBT-1000と接続すると、仮想COMポートを作ります。

仮想COMポートとは、実際にはパソコン上に物理的には存在しないRS232Cポートを作り、そのポートにRS232Cと同じ方法でアクセスすることで、パソコンとRBT-1000が通信できるようになる便利な機能です。

ウィザードがデバイスの認識を完了すると、割り当てられたCOMポート番号がウィザードに表示されます。このポートに対してアクセスできるようになりますので、必ず控えておいて下さい。



必ず控えておいて下さい。

上図の例では、"COM40"が割り当てられたポート番号です。

これでパソコンとRBT-1000は上記のCOMポートを介して通信を行うことができるようになりました。続いてRBT-1000の最も基本的な設定と基本的な使い方を紹介します。

**i Windows標準のプロトコルスタックが使いにくい場合**

Windows7や8にはBluetoothのプロファイルとしてSPPが実装されていますが、やや扱いにくい場合があります。特にWindows8の場合には設定やデバイスの検索など操作がしにくいことがあります。その場合には、サードパーティー製のプロトコルスタックソフトウェアを使うと操作性が向上します。

当方ではBluetooth dongleの付属ソフトウェアとして広く普及している"Bluetooth Stack for Windows by Toshiba"の使用をお奨めします。

◎PICマイコンとRBT-1000を接続して設定を行う

細かい設定は色々ありますが、ここではまず使い方を一通り体験するためにデフォルト設定で通信を試みましょう。Bluetoothデバイスとのリンク確立前はコマンドモードになっています。

UART側の通信設定は、デフォルトで次のようになっています。

- ・通信速度: 115.2kbps
- ・データ長: 8ビット長 (LSB先頭)
- ・ストップビット: 1
- ・パリティ: なし

接続方法については、本書の1ページ、2ページをご参照ください。

◎データ通信を開始しましょう

デフォルト設定はRBT-1000に電源を投入すると、最初にバナー文字列が返ります。まずはこの文字列を受信します。

```
WRAP THOR AI (4.0.0 build 317)<cr><lf>
Copyright (c) 2003-2010 Bluegiga Technologies
Inc.<cr><lf>
READY.<cr><lf>
```

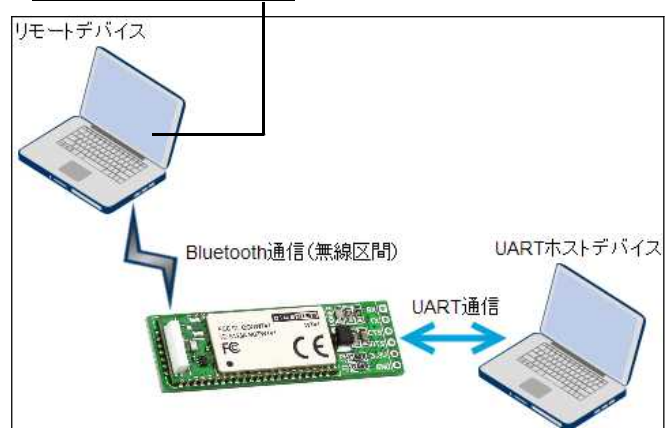
バージョン値、ビルド値などはファームウェアのバージョンによって変わることがあります。最後の行の"READY.<CR><LF>"の文字列を受信したら各種操作が可能となります。なお、このバナー文字列を電源投入時に返さないように設定場合は、"SET CONTROL ECHO"コマンドで行えます。

動作確認のため"AT"コマンドを送信します。"OK"の文字列が返ることを確認します。なお、コマンドはASCIIコードで送信します。"AT"コマンドの場合には、41h, 54h, 0Dh の3バイトを送信することになります。下図はRBT-1000の戻り値をオシロスコープで観察し、シリアル通信コマンドを16進数で解析して表示させた図です。



最初の"AT"<CR><LF>のあと、"OK"の文字列が返るまでには実測で7.05ミリ秒経過していることがわかります。

リモートデバイスのパソコン側からRBT-1000に接続しましょう。



リモートデバイスのパソコンの仮想COMポートをオープンします。なお、この際のリモートデバイス側の仮想COMポートのボーレートは問いません。1200bps~3Mbpsまで(仕様値)で通信が可能です。実際にはボーレートは9.6kbps~115.2kbps程度の範囲で通信することが推奨されます。特にハードウェアフロー制御を行わない場合にはデータの取りこぼしがないよう低速で通信をします。

仮想COMポートを開くとRBT-1000は次の文字列をUART経由で返し、自動的にデータモードに移行します。

```
RING 0 xx:xx:xx:xx:xx:xx 1 RFCOMM
```

これは、RBT-1000でRINGイベントが発生したことを意味します。RINGイベントの書式は次の通りです。

RING {リンクID} {address} {channel} {profile}

リンクIDは、接続のIDで、他のコマンドでこの接続について操作をするとき、このID番号を指定することになります。

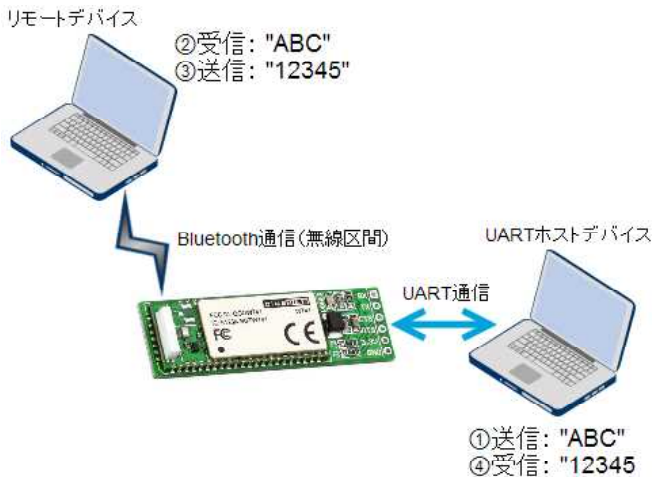
アドレスはリモートデバイスのBluetooth IDです。

チャンネルは、SPPで接続した時のRFCOMMのローカルチャンネルです。RFCOMMのチャンネル番号です。

最後のプロファイルは、この接続がBluetoothのどのプロファイルで接続したのかを示します。RBT-1000でシリアル通信する場合には、RFCOMMとなります。

このRINGイベントをUART通信で受信した直後からデータモードになります。このRINGイベントの文字列を受信することでUART機器側はデータモードに移行したことを把握できます。

ここから先は、データモードですのでリモートデバイスとの間でUART通信のやりとりが可能になります。例えばUARTホスト側で"ABC"という文字列を送信すれば、リモートデバイスで文字列"ABC"を受信できます。その逆もちろん可能です。



上図は通信の模式図です。①→④の順で送受信した場合です。

### ⑦データ通信を終わらしましょう

一度確立したデータモードは、次のいずれかの操作が行われるまで継続します。次の操作でデータモードは終了します。

- ・リモートデバイスが仮想COMポートを閉じて接続を切断した場合
- ・UARTホストからエスケープシーケンス文字を送信した場合

リモートデバイスで仮想COMポートを閉じると、データモードでの接続は終了し、RBT-1000は自動的にコマンドモードに移行します。このデータモードが終了するとRBT-1000はUART経由で次の文字列を返します。

```
NO CARRIER 0 ERROR 0
```

これは、NO CARRIERイベントが発生したことを意味します。このイベントの発生はデータモードが終了し、RBT-1000がコマンドモードに

移行したことを意味します。NO CARRIERイベントの書式は次の通りです。

NO CARRIER {リンクID} ERROR {error\_code} {message}

リンクIDは終了したデータモードの接続IDです。error\_codeはエラーが無い場合には数字の"0"となります。messageはエラーについてのメッセージで、エラーが無い場合には省略されます。

UARTホスト機器は、このNO CARRIERイベントの発生を受信することでRBT-1000がデータモードを終了したことを把握できます。

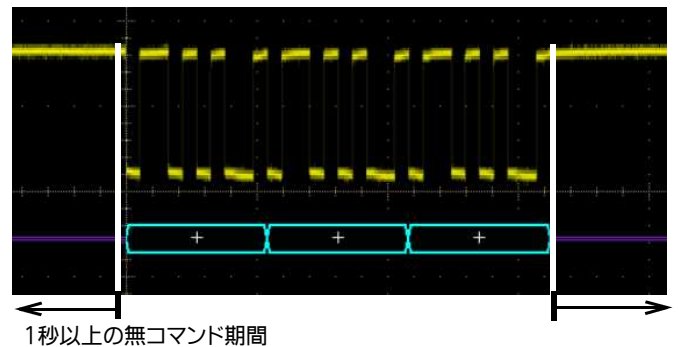
上記はリモートデバイス側から操作した終了した場合ですが、UARTホスト機器がデータモードを意図的に終了させることもできます。但し注意が必要なのは、データモードではUART通信で受信したすべてのデータがBluetooth経由でリモートデバイスへ送信されますので、コマンドを送れないことです。そこで、RBT-1000ではあらかじめ指定したエスケープシーケンスの文字を3回受信した時にデータモードを終了するように設計されています。

エスケープシーケンスの文字は、"SET CONTROL ESCAPE"コマンドで設定できます。デフォルト設定では、"+"(2Bh)に指定されています。またデータの中には、たまたまエスケープシーケンスの文字が3つ続いたデータを送信するケースがあるかもしれません。そこで、このエスケープシーケンスの文字3つ(3バイト分)の前後に1秒の無通信時間があった場合のみRBT-1000は、このデータをエスケープシーケンスの文字として認識します。

<1秒の無通信時間>+++<1秒の無通信時間>

1秒以内に別のデータがUART経由で到来した場合、RBT-1000はこの"+++"文字列をエスケープシーケンスとは見なせず、データモードを継続します。

下図はエスケープシーケンスの文字列をUARTホスト機器からRBT-1000に送信した時のオシロスコープの波形です。



コマンドモードに正しく移行するとRBT-1000はREADYイベントが発生します。"READY."という文字列がRBT-1000から返ります。この文字列を受信できれば、RBT-1000がコマンドモードになったことを確認できます。

注意が必要なのは、パソコン等のシリアル通信ソフトで送信する場合、意図せずデータの後ろ側にCRやLFが付加されてしまっていることがあります。また、場合によっては、データとしてUARTブレイクとして認識されるようなレベル期間が入る場合もあります。こういった2Bh以外のデータが付加されてしまうとRBT-1000はコマンドモードに移行せずデータモードを維持します。READYイベントを確認できない場合には、上図のようにオシロスコープ等で必ずUARTホストからの送信データの波形を確認して、2Bh以外のデータが前後に付加していないかどうかを確認してください。

◎現在接続しているリンク先を調べて再度データモードに移行する

RBT-1000が現在接続しているリンク先をすべて表示させることができます。ここで接続されているリンクを表示させることでリンクIDが分かります。そのリンクIDを使うことで希望のリンク先デバイスとの間でデータモード通信ができます。

リンクデバイスをリスト表示させるには"LIST"コマンドを使います。

```
LIST<cr><LF>
```

戻り値は2行になります。1行目は現在RBT-1000に接続しているコネクションの数を返します。2行目はそれぞれのリモートデバイスの概要を表示します。2行目は引数が多いですが、特に知りたい情報は、そのデバイスを示すBluetooth IDと、接続指定する時に使うリンクIDです。

```
LIST {num of connections}

LIST {link_id} CONNECTED {mode} {blocksize} 0 0
{elapsed_time} {local_msc} {remote_msc} {addr}
{channel} {direction} {powermode} {role}
{crypt} {buffer} [ERETX]
```

例えば現在RBT-1000が1台のリモートデバイスと接続を確立している場合には次のように表示されます。

```
LIST 1
LIST 0 CONNECTED RFCOMM 666 0 0 3332 8d 1
00:xx:xx:xx:xx:xx 1 INCOMING ACTIVE SLAVE
PLAIN 0
```

ここで分かる主なことは次のような内容です。

- ・現在のコネクション数は1
- ・リンクIDが0番のデバイスはRFCOMMによって接続されている
- ・1回のパケットで送信できるデータサイズは最大で666バイト
- ・Bluetooth IDは 00:xx:xx:xx:xx:xx
- ・リンクの方向はインカミングであること  
→リンクはリモートデバイスによって初期化された
- ・パワーモードはActive  
→Activeは省電力モードでないことを示す
- ・RBT-1000はスレープデバイスとして登録されている
- ・接続は暗号化されていない
- ・インカミングデータ用バッファはサイズ0(データ無し)

特にデータモードに移行する場合使うのは最初のリンクIDです。コマンドモードからデータモードへ移行する場合には、"SELECT"コマンドを使用します。SELECTコマンドでデータモードに移行したいリンクIDを指定します。

```
SELECT {link_id}<cr><LF>
```

例えば、リンクIDが0のコネクションを指定したい場合には次のようにします。

```
SELECT 0
```

これでコマンドモードからデータモードに移行します。このようにRBT-1000では、コマンドモードからデータモードへ、逆にデータモードからコマンドモードにそれぞれ移行することが可能です。

## RBT-1000の主な基本設定

パスキーの設定や本体デバイス名の変更方法、UART通信速度の変更、エコー設定などRBT-1000の基本的な設定方法を紹介します。その他にもRBT-1000には様々な設定がありますが、一般的にはあまり使用しないため日本語マニュアルでは紹介していません。その他の設定コマンドについては英語版のマニュアルをご参照下さい。

### ■コマンドの基本

RBT-1000では、各種設定を行うコマンドには最初に必ず"SET"がつきます。更に多くの場合、その次にBluetoothの設定では"BT"が、本体そのものの設定では"CONTROL"がつきます。

例えばデバイス名の設定は、"SET BT NAME"とというコマンドになります。なお、表示上スペースのある部分は20hのスペースがあります。

### ■各種コマンド

コマンド内容	デバイス名の変更
コマンド	SET BT NAME {friendly_name}
引数	friendly_name デバイス名を256文字以内で指定
応答	なし
発生イベント	なし
コマンド詳細	デバイス名を変更します。 デバイス名の引数を何も入力しないとデバイス名が空になり、デバイス検索の時、問題が発生することがあります。 設定後は"RESET"コマンドで本体を再起動してください。

コマンド内容	PINコード(パスキー)の変更
コマンド	SET BT AUTH * {pin_code}
引数	pin_code 設定するPINコードを16文字以内で指定します。 これを省略するとPINコードなしでコネクションが確立するようになります。(デフォルト状態) セキュリティのためにここに文字列を設定することをお奨めします。
応答	なし
発生イベント	なし
コマンド詳細	PINコードはリモートデバイスと接続を確立する時に本機とリモートデバイス間での通信に一定のセキュリティを保つ仕組みです。接続時PINコードを入力しないと、通信が確立できません。 pin_code引数を省略すると、PINコードの入力なしで接続が確立できるようになります。

コマンド <sup>※</sup> 内容	UART通信の通信速度変更
コマンド <sup>※</sup>	SET CONTROL BAUD {baud_rate},8{parity}{stop_bits}
引数	<p>baud_rate 通信速度をbpsで指定します。9600、115200など1200の倍数を指定します。 ※設定値には注意が必要です。</p> <p>parity パリティの設定。下記のパラメータを指定します。 n ノンパリティ e 偶数パリティ o 奇数パリティ</p> <p>stop_bit ストップビットを設定します。 1 1ストップビット 2 2ストップビット</p>
応答	なし
発生イベント	なし
コマンド <sup>※</sup> 詳細	<p>コマンドを実行後、エコー設定が有効の場合には送信文字列が返った後、設定した内容が有効になります。</p> <p>工場出荷時の状態では通信速度は115.2kbpsになっていますので、9.6kbpsなどに変更する場合には、一度115.2kbps通信の可能なUARTホスト機器から本コマンドを実行して希望の内容に設定を変更します。</p> <p>コマンド指定時はカンマ(,)の位置や、数字8が挿入されることに注意してください。例えば通信速度を9.6kbpsに、ノンパリティ、1ストップビットに設定したい場合には次のように指定します。 <b>SET CONTROL BAUD 9600,8n1</b></p> <p>本通信速度は、UART通信の速度です。Bluetooth通信をしているリモートデバイス間との通信速度ではありません。</p> <p>通信速度の値を間違えてたり、本機で不可能な通信速度に設定した場合、本機にアクセスできなくなります。その場合、本機の復旧ができなくなりますので設定には十分にご注意ください。設定値の誤りによって使用が出来なくなった場合、当方でも修理は不可能です。</p>

コマンド <sup>※</sup> 内容	コマンドエコーの設定
コマンド <sup>※</sup>	SET CONTROL ECHO {echo_mask}
引数	<p>echo_mask ビットマスクでエコーの有効/無効を設定します。</p> <p>Bit0 このビットを1にセットすると起動時のバナー文字列が有効になります。</p> <p>Bit1 このビットを1にセットするとコマンドモード時に文字列がエコーバックします。</p> <p>Bit2 このビットを1にセットするとコマンドモード移行時又はデータモード移行時にイベントが発生してイベント発生を通知する文字列を返します。</p>
応答	なし
発生イベント	SYNTAX ERROR
コマンド <sup>※</sup> 詳細	<p>ビットマスクとは、該当ビットを0か1にセットすることで。例えばBit0を0、Bit1を1、Bit2を1に設定する場合、ビットで示すと110となります。これを10進数で表すと6ですので送るコマンドは下記ようになります。 <b>SET CONTROL ECHO 6</b></p> <p>なおこの設定はBit0が0ですので本体起動時のバナー文字列が返されなくなります。</p> <p>すべてのビットを0にしてしまうと、RBT-1000の状態を確認することが難しくなります。また、Bit2を0にセットすると、RBT-1000がコマンドモードに移行したか、又はデータモードに移行したのかを確認することができなくなります。</p> <p>初期設定値は111(10進数で7)です。</p>



コマンド内容	エスケープシーケンス文字の変更
コマンド	SET CONTROL ESCAPE {esc_char} {dtr_mask} {dtr_mode}
引数	<p>esc_char</p> <p>エスケープシーケンスに設定する文字のASCIIコードを10進値で指定します。10進値で指定することに注意してください。デフォルト設定は + なので43が設定されています。</p> <p>なお、- (45)に設定するとエスケープシーケンスは無効になります。</p> <p>dtr_mask</p> <p>RBT-1000では使用できません。 0に指定してください。</p> <p>dtr_mode</p> <p>RBT-1000では使用できません。 0に指定してください。</p>
応答	なし
発生イベント	SYNTAX ERROR
コマンド詳細	エスケープシーケンスの使い方は本書の6ページをご覧ください。

コマンド詳細	<p>本機と通信可能な範囲内に存在するデバイスを検索するコマンドです。通常デバイスが発見されると、INQUIRY_PARTIALイベントが発生してBluetoothアドレスを続けて返します。</p> <p>検索時間がタイムアップすると、レスポンスが返り発見したデバイスの数と、それに続いてBluetoothアドレス及びデバイスのサポートしているクラスを番号(16進数)で返します。デバイスのクラス番号(CoD)は、Bluetoothの仕様書で定められた24ビットの値となります。この値は下記URLのようなサイトで調べることが可能です。 <a href="http://www.ampedrftech.com/cod.htm">http://www.ampedrftech.com/cod.htm</a></p> <p>タイムアウト時間は長いほど多くのデバイスを検出できます。一方で短いとほとんどのデバイスを発見できません。適当な値は5~10くらいまでを指定するとよいでしょう。</p>
--------	--

コマンド内容	本体のソフトウェアリセット(再起動)
コマンド	RESET
引数	なし
応答	なし
発生イベント	なし
コマンド詳細	本体を再起動します。各種設定コマンドなどで本体を再起動する時に使えます。

コマンド内容	近傍Bluetoothデバイスの検索
コマンド	INQUIRY {timeout}
引数	<p>timeout</p> <p>1~48の値で検索タイムアウト時間を指定します。値は1.28秒単位です。</p>
応答	<p>INQUIRY {num_of_devices} 及び INQUIRY {addr} {class_of_device}</p> <p>num_of_devices 見つかったデバイスの数を表示します。</p> <p>addr 見つかったデバイスのBluetoothアドレス</p> <p>class_of_device デバイスのBluetoothクラスを示します</p>
発生イベント	<p>INQUIRY_PARTIAL デバイスが見つかった時、このイベントが発生します。</p> <p>INQUIRY_EXTENDED 見つかったデバイスがEIR(Extended Inquiry Response)をサポートしたBluetooth2.1+EDRの時発生します。</p> <p>↓続きます</p>

## RBT-1000を色々なデバイスと通信させる

### ■RBT-1000同士で通信させる

RBT-1000を2台用意して通信させる方法をご紹介します。  
RBT-1000にUARTホストデバイスを接続してそれぞれ通信できるようにします。両方のデバイスの電源を入れてください。

- どちらか一方のデバイスから接続要求を相手方のデバイスに対してリクエストし接続を開始します。  
例えば下のような場合を想定してデバイスAからデバイスBに対して接続のリクエストをすることとして解説します。



※上記のBluetooth IDは例として付けたものです。実際のデバイスのBluetooth IDを使ってください。

- デバイスA側で"INQUIRY"コマンドを実行してデバイスB側のBluetoothアドレスを把握します。

```
INQUIRY 5
INQUIRY_PARTIAL 00:0b:xx:xx:xx:xx 001f00
INQUIRY_PARTIAL 00:xx:xx:xx:xx:xx 180104
INQUIRY 2
INQUIRY 00:0b:xx:xx:xx:xx 001f00
INQUIRY 00:xx:xx:xx:xx:xx 180104
```

上の例では2つのデバイスが検出されました。ここでは1行目が該当するリモートデバイスのアドレスなのでこのアドレスを控えます。

※RBT-1000に使用されているWT41モジュールはBluegiga社のアドレスなので最初が"00:07:80~"で始まります。よって複数のデバイスが検出された場合、"00:07:80~"のアドレスを持つデバイスがRBT-1000であることが把握できます。ここではわかりやすいようにBluetoothアドレスを 00:0b:xx:xx:xx:xx としています。

- 接続を確立するには、"CALL"コマンドを使用します。このコマンドでは、接続先のリモートデバイスのBluetoothアドレス、プロファイル、接続モードを指定して接続を試みます。書式は下記の通りです。

コマンド内容	指定したリモートデバイスに接続する
コマンド	CALL {address} {target} {connect_mode}
引数	<p>address リモートデバイスのBluetoothアドレスを指定します。 xx:xx:xx:xx:xx:xx の形式で指定します。</p> <p>target プロファイルに応じた値を指定します。 SPPで接続しますので、ここは 1101 を指定してください。(別の値を指定するとシリアル通信ができなくなります。1101はuuid16でSPPを示します。)</p> <p>connect_mode 接続するモードです。ここは "RFCOMM" を指定します。</p> <p>↓続きます</p>

応答	CALL {link_id} link_id 接続が確立できた場合、その接続を指し示すリンクIDが返ります。通常0から順番に返ります。
発生イベント	CONNECT CALLコマンドで接続が確立すると発生します。  NO CARRIER 接続に失敗すると発生します。
コマンド詳細	<p>指定したBluetoothアドレスのリモートデバイスと接続を確立するコマンドです。本機で採用されているBluetoothモジュールのWT41にはSPP以外にも多くの対応プロファイルがあるため、本来は設定が複雑ですが本機はシリアル通信を無線化することに特化していますので、設定項目を簡略化して記載しています。</p> <p>接続が確立すると、RBT-1000は自動的にコマンドモードからデータモードに移行します。 再度コマンドモードに移行したい場合にはエスケープシーケンスによってコマンドモードに戻ることができます。</p>

例えば次のようにコマンドを実行します。

```
CALL 00:0b:xx:xx:xx:xx 1101 RFCOMM
CALL 0
CONNECT 0 RFCOMM 1
```

接続が正しく確立できるとCALLイベントが発生します。  
"CALL"の文字列の後ろにはリンクIDが示されます。このリンクIDは接続を閉じる時必要ですので記録しておく必要があります。

- "CALL"コマンドで接続が確立するとリモートデバイス側ではRINGイベントが発生し、両方のデバイスが自動的にデータモードに移行します。データモード中は、両デバイスのシリアルデータはすべて無線化されてリモートデバイスに送信されます。

エスケープシーケンスによってコマンドモードに戻ることができます。

- 通信を終了したい場合には一度エスケープシーケンスによってコマンドモードに移行した後、"CLOSE"コマンドで通信を閉じます。エスケープシーケンスを発行して正しくコマンドモードに移行できると、READYイベントが発生します。READYイベントが発生したということは、コマンドモードに正しく移行できたということですので、各種コマンドが使用できます。  
"CLOSE"コマンドで接続を切断します。

```
CLOSE 0
NO CARRIER 0 ERROR 0
```

なお"CLOSE"コマンドの後ろの引数0は、切断するリンクIDです。

これで接続はクローズされました。再び接続する場合には、手順3から実行します。



## パスキーによる認証とSSP認証について

Bluetoothにはパスキーによる認証と、Bluetooth2.1以降の規格で策定されたSecure Simple Paring (SSP)という2つのセキュリティシステムがあります。RBT-1000はどちらにも対応しており、次のような方法を使うことが推奨されています。

・Bluetooth2.1以降のデバイスとRBT-1000が通信を行う場合にはできる限りSSPを使うことが望ましい。

・Bluetooth2.1より古い仕様のデバイスとRBT-1000が通信する場合や、意図的にユーザー(開発者)が決めた数字(4桁)をパスキーとして使用したい場合にはSSP機能を無効にした上で、パスキー設定による認証を使う。

RBT-1000ではデフォルトでSSPが有効になっており、RBT-1000同士で通信をする場合にはこちらを使うことになります。(この場合ユーザーはパスキーなどを入力する必要はありません。)SSPには4つの方式があります。

- ・確認なし(Just Works)
- ・6桁の認証コードを表示しての一致確認 (Numeric Comparison)
- ・6桁の認証コードを打ち込むことによる確認 (Passkey Entry)
- ・Bluetooth以外の通信による確認 (Out-Of-Band, OOB)

RBT-1000ではデフォルトでJust WorksになっているためSSPを使って接続する場合、確認は必要ありません。もしRBT-1000同士での通信やリモートデバイスがBluetooth2.1以前のデバイスの場合に次のように設定してパスキーによる認証を行います。またリモートデバイスのRBT-1000にパスキーが設定されている(SET BT AUTHコマンドによる)場合には次のようにします。

- ① ローカルデバイス側のRBT-1000にて、リモートデバイスと同じパスキーを設定します。(ローカルのRBT-1000のパスキーをリモートデバイスと同じに設定します。)

```
SET BT AUTH * 1111
```

※ローカルデバイスのパスキーが1111の場合

- ② ローカルデバイスのSSPを次のコマンドで無効にします。

```
SET BT SSP 3 0
```

- ③ "CALL"コマンドでリモートデバイスと接続します。パスキーが正しければ通信が確立してデバイスはデータモードに移行します。

## ■Android端末と接続する

BluetoothのSPPに対応したAndroid端末(スマートフォンやタブレット端末)と接続してRBT-1000と通信をすることができます。Android端末では、様々なSPP対応のソフトウェアがGoogle Playからダウンロードすることができ、Android端末とマイコンなどの接続がBluetoothで簡単にいきます。

※Androidのバージョンによって操作方法は異なります。

- 1 あらかじめRBT-1000に"SET BT AUTH"コマンドを使ってパスキーを割り当ててください。  
ここでは例として"0000"に設定します。
- 2 Android端末のBluetooth通信機能を有効にします。
- 3 周辺のBluetoothデバイスを検索してください。  
RBT-1000は"WT41-A"というデバイス名で検出されます。



"WT41-A"をタップします。

- 3 Android端末側でパスキーの入力が求められますので、手順1で設定したパスキーを入力します。  
→ペアリングが完了します。
- 4 Androidからデータの送受信をする場合には、SPPに対応したソフトウェアを使います。例えば下記のようなソフトがあります。

- ・S2 Terminal for Bluetooth
- ・S2 BT Terminal2
- ・SENA BTerm
- ・UNI BT remote LITE
- ・uConnect Bluetooth Terminal
- ・Bluetooth Serial Controller
- ・microController BT

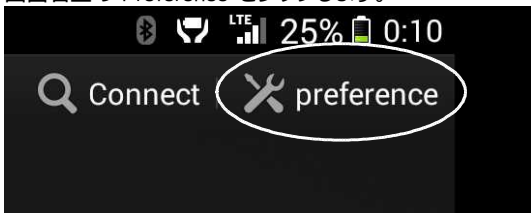
Google Playからダウンロードできます。その他にも"SPP Bluetooth"で検索するといっぱい類似ソフトが出てきますのでお気に入りのソフトを見つけてください。  
ここではBluetooth Serial Controllerを使ってサンプルを作ってみましょう。



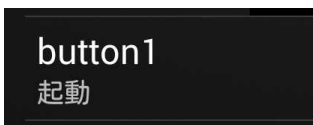
- 5 Bluetooth Serial Controllerは画面上にボタンを自由に配置し、各ボタンに対して、ボタンを押した時に出力するデータを割り当てることができるソフトウェアです。  
アプリケーションを自作しなくてもBluetoothで簡単にシリアルコマンドを送れます。

画面右上の"Connect"ボタンを押して、ペアリング済みデバイス一覧から"WT41-A"を選択します。

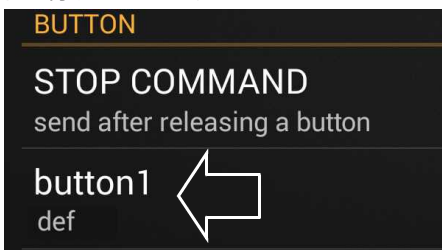
- 6 接続が完了するとRBT-1000はRINGイベントを発生します。ボタンの表示名と、押した時に出力するコマンドを設定します。画面右上の"Preference"をタップします。



- 7 "BUTTON"の一覧に表示されている"Name"をタップします。ここでボタンに表示される名前を設定します。例えば"button1"に"起動"というボタン名を付けることにします。最初は,"button1"でその下に"def"と記載されていますので、ここをタップします。入力ができるようになりますので、日本語で"起動"と入力してOKをタップします。名前は日本語が付けられます。



- 8 戻るボタンをタップして先ほどのPreference画面まで戻り、続いて"Command"をタップします。ボタンを押した時に出力できるデータはASCII、すなわち文字列か、HEXの16進数かを選択できます。ここでは"ASCII"を選択しておきます。その下"BUTTON"の一覧に"button1"がありますのでタップします。これは先ほど"起動"という名前を付けたボタンです。



- 9 文字の入力画面になりますので、ここでボタンをタップした時に出力したい文字列を入力します。ここでは例として"START"と入力します。

基本的な設定はここまでです。戻るボタンを何回かタップしてボタンが表示されている一番最初の画面まで戻ってください。

- 10 "起動"というボタンがあることが確認できます。



このボタンをタップしてみてください。RBT-1000で"START"という文字列が受信できるはず。このような要領でボタンを作っていけば簡単に自分だけのコマンド送信ソフトが作れます。その他、ボタン画面の下にあるテキストボックスに文字列を入力して"Send"ボタンをタップすると任意の文字列も送信できます。

その他、似たようなソフトウェアとしてはmicroController BTというソフトウェアがありこちらは自由にボタンを配置したりスライダーなども利用できます。

またこのようなボタンアプリケーションではなくシリアルターミナルのような送受信ソフトウェアではS2 Terminal for Bluetoothなどが便利でわかりやすいです。

## 主な仕様

電源電圧:	DC3.3V(絶対最大定格3.7V)
消費電流: (下記参照)	待機時 4mA 待機時間欠 最大20mA SPPリンク確立時 最大170mA SPPリンク確立時 平均60mA
Bluetooth(R)準拠規格	BluetoothVer.2.1+EDR BluetoothVer.1.xとの互換性有り
無線伝送方式	周波数ホッピング方式 スペクトラム拡散(FS-SS)方式 送信周波数範囲 2.4GHz~2.4835GHz
通信範囲	Class1準拠 見通し内通信で約1km以内 ※通信可能実効距離800m程度
TXパワー/RX受信感度	19dBm/-92dBm
周囲動作温度	-40°C~+85°C
UARTインターフェイス	CTS/RTSハンドシェイク通信方式 但し、2線式通信でも通信可能
UART通信速度	デフォルト設定通信速度 115200bps 1200bps~3Mbps
データ形式	パリティ偶数、奇数又はなし ストップビット1又は0 データ長8ビット長
UART電圧レベル	ロジック電圧レベルL Vss-0.4V(min) ロジック電圧レベルH Vdd+0.4V(max)
動作環境:	5°C~60°C (動作保証範囲) 但し結露無きこと
外形寸法	52mm×16mm
使用Bluetoothデバイス	BlueGiga Technologies Inc社製 WT41-A
認証コード	技適認証(日本) R209-J00047
製造国	セルビア、アメリカ、中国

※更に詳しい仕様が必要な場合にはWT41-Aのデータシートをご参照ください。

## 使用上の注意

①RBT-1000を使用するに際し、当方は明示的及び潜在的な使用したことによる危険性や、不確実性については予見することができません。使用する際には、お客様の責任においてこの製品を正しくお使いいただけますようお願い致します。

②RBT-1000は、Bluetooth規格に基づきデータ通信を空間伝送できるデータ通信モジュールですが、性能及び信頼性は一般的な使用の範囲に限定されます。本製品を宇宙、航空、原子力、燃焼制御、運輸、交通、各種安全装置、ライフサポート関連の医療機器等のように、特別な品質・信頼性を要求され、その故障や誤動作が直接人命を脅かしたり、人体に危害を及ぼす恐れのある用途向けには使用できません。

③RBT-1000では様々な外的要因、特に使用する周囲の電磁界環境によって、性能を仕様どおり発揮できない場合があります。また、場合によっては正しく動作しない場合や期待した動作をしない場合があります。本製品を使用することによって生じた、もしくはこれに関連するいかなる直接・間接損害、懲罰的損害、その他データの破損や消失等を含むいかなる損害、損失についても、当方では一切責任を負いかねます。あらかじめご理解とご了承頂きますようお願い致します。

④RBT-1000を使用した製品等を製造させる場合には、様々なフェイルセーフ機能(安全設計)を施して頂き、十分に機器のテストをした上で運用されますようお願い致します。また、データの損失や予期しない事態に備え、データのバックアップを行って頂きますようお願い致します。

⑤RBT-1000は電波法に基づく無線設備として技術基準適合証明を受けています。本製品は日本国内で使用することができます。

⑥本製品は米国再輸出管理規制(EAR)に該当する製品です。本製品は最終仕向国を日本国内として輸入・販売している製品のため本製品を国外へ輸出することは原則としてできません。いったん米国から輸出された本製品が、その後、当国から第三国あるいは第三国の特定の使用者向けに再輸出される場合、その相手先が米国からの直接輸出が規制されている場合、再輸出においても同等の規制を受けることとなります。当方ではEAR該当品につき、製品の輸出については禁止しております。

⑦輸出に関わる各種証明書、非該当判定書などの発行は行っておりません。また輸出に関するお問い合わせも承っておりません。

⑧RBT-1000は技適認証をうけた製品ですので、本製品を分解・改造することは法律で禁止されています。アンテナを改造したり電波の出力強度を変更することなどは法律で禁止されています。

⑨RBT-1000は以下の無線局と同じ周波数帯を使用します。産業、科学、医療用機器・工場の製造ライン等で使用されている移動体識別用無線局・無線LAN関連製品全般上記の機器や無線局の近くで本製品を使用すると電波干渉が生じる恐れがあります。特に医療機器の近くでの使用は、他の機器に影響を与える場合がありますので使用に際しては十分ご注意ください。

## 製品の技術的なサポートについて

本製品の技術的なサポートはモジュール製造メーカーのBluegiga Technologies社が直接行います。技術的なサポートが必要な場合には、下記へ直接ご連絡頂きます。原則と致しまして当方での技術的なサポートは致しておりません。

技術サポートはすべて英文となります。あらかじめご了承頂きますようお願い申し上げます。なお、日本語補足マニュアル及びFAQにつきましては、当方のサイトより最新の情報をご提供致します。

なお、サポートはいずれも技術者専門となっております。より技術的なご質問をするためのものとなっております。マニュアルに既に記載されている内容などについては回答が得られない場合がございます。

<https://www.bluegiga.com/en-US/support/>


サポートはチケット制になっており、WEBフォームからご質問頂く形式です。まずはユーザー登録が必要ですので上記サイトよりユーザー登録をしてアカウントを作成してください。アカウント作成後、そのアカウントでログインし"Contact Tech Support"のページからご質問ください。その際次の項目については下記のように選択してください。

- ・Productの項目 "Bluetooth Classic"→"WT41"
- ・Countryの項目 "APAC"→"Japan"

なおご質問事項はできるだけ簡潔にして頂き、使用環境、接続状況、エラーコードの内容などは具体的にお書き下さい。

当方では、最新のFAQを下記サイトで公開しております。よくアルご質問はFAQにまとめておりますので、予めこちらをご確認頂きますようお願い致します。

<http://www.microtechnica.tv/faq/faq.cgi>

マイクロテクニカ   
〒158-0094 東京都世田谷区玉川1-3-10  
TEL: 03-3700-3535 FAX: 03-3700-3548  
(C)2014 Microtechnica All rights reserved

