

BeeCom

シリアル通信

アプリケーションプロトコルシミュレータ

取扱説明書

2015/03/18 Ver0.92

2014/02/24 Ver0.9

2013/10/22 Ver0.8

ご使用にあたり、同梱の「ソフトウェア使用許諾契約」をよくお読みください。お客さま
にご確認頂く必要のある事項が記載されています。

はじめに

BeeCom は、依然として産業機器などで需要があるシリアル通信(RS-232C 調歩同期、将来的には HDLC など)や、UDP 通信において、アプリケーション定義のプロトコルを実行させるためのソフトウェアです。

BeeCom は、基本的に、

- ・受信データが[パターン]に[回数]マッチしたら[遅延時間]待って[データ]を送信する
- ・[パターン]を指定しないと[遅延時間]間隔で自発的に[データ]を送信する

ものです。[イタリック体]部分のパラメータを、指定されたディレクトリにある csv 形式のテキストファイルから読み込んで動作しますが、データ作成用の OpenOffice Calc ドキュメント(*.ods)を同梱しているので、そちらを使うことをおすすめします。動作中一定周期で送信データを指定したファイルを読みながら動作するので、通信中に送信データを変更することができます。(通信中に更新されるのは送信データだけで、その他のデータ(Environments や Responses など)は更新されません。)

また、プラグインを使用すると、受信データ中のあるデータを自動応答したり、送信データに FCS を自動付与したりできます。

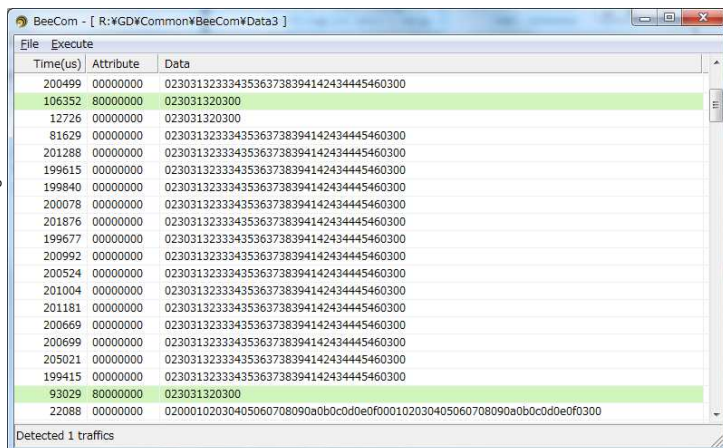
インストール・アンインストール

インストーラはありません。取得・解凍して出来たフォルダを、任意の位置に置いてご使用ください。ただし、C:\Program Files や C:\Program Files(x86)など、既定のプログラムインストールフォルダには置かないほうが無難です。

また、後述の「Tips:OpenOffice ドキュメントを右クリックすることで開けるようにする」の設定を行うと、便利にご使用になれます。(行わなくても使用はできます。)

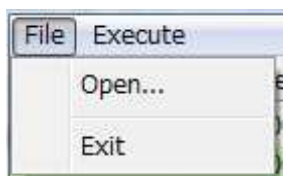
マンマシンインターフェース

メイン画面には、送受信データの時刻(一つ前の送受信データとの差分時間、単位 μsec)、属性(後述)、送受信データ(バイナリ表示)が表示されます。



The screenshot shows the BeeCom application window with the title bar 'BeeCom - [R:\GDI\Common\BeeCom\Data3]'. The window has two menus: 'File' and 'Execute'. The main area displays a table with three columns: 'Time(us)', 'Attribute', and 'Data'. The table contains multiple rows of data, with some rows highlighted in green. The 'Time(us)' column shows values like 200499, 106352, 12726, etc. The 'Attribute' column shows values like 00000000, 80000000, 00000000, etc. The 'Data' column shows binary strings like 02303132333435363738394142434445460300, 023031320300, etc. At the bottom, it says 'Detected 1 traffics'.

Time(us)	Attribute	Data
200499	00000000	02303132333435363738394142434445460300
106352	80000000	023031320300
12726	00000000	023031320300
81629	00000000	02303132333435363738394142434445460300
201288	00000000	02303132333435363738394142434445460300
199615	00000000	02303132333435363738394142434445460300
199840	00000000	02303132333435363738394142434445460300
200078	00000000	02303132333435363738394142434445460300
201876	00000000	02303132333435363738394142434445460300
199677	00000000	02303132333435363738394142434445460300
200992	00000000	02303132333435363738394142434445460300
200524	00000000	02303132333435363738394142434445460300
201004	00000000	02303132333435363738394142434445460300
201181	00000000	02303132333435363738394142434445460300
200669	00000000	02303132333435363738394142434445460300
200699	00000000	02303132333435363738394142434445460300
205021	00000000	02303132333435363738394142434445460300
199415	00000000	02303132333435363738394142434445460300
93029	80000000	023031320300
22088	00000000	0200102030405060708090a0b0c0d0e0f000102030405060708090a0b0c0d0e0f0300



File -> Open... : データを読み込むディレクトリを指定します。ここでは指定されたディレクトリのファイルを読むわけではありません。実際に読み込まれるのは Execute -> Start したときです。コマンドラインでファイル名を指定すると、そのファイルがあるディレクトリを指定されたものとして動作します。

File -> Exit : 終了します。

Execute -> Start : 動作を開始します。

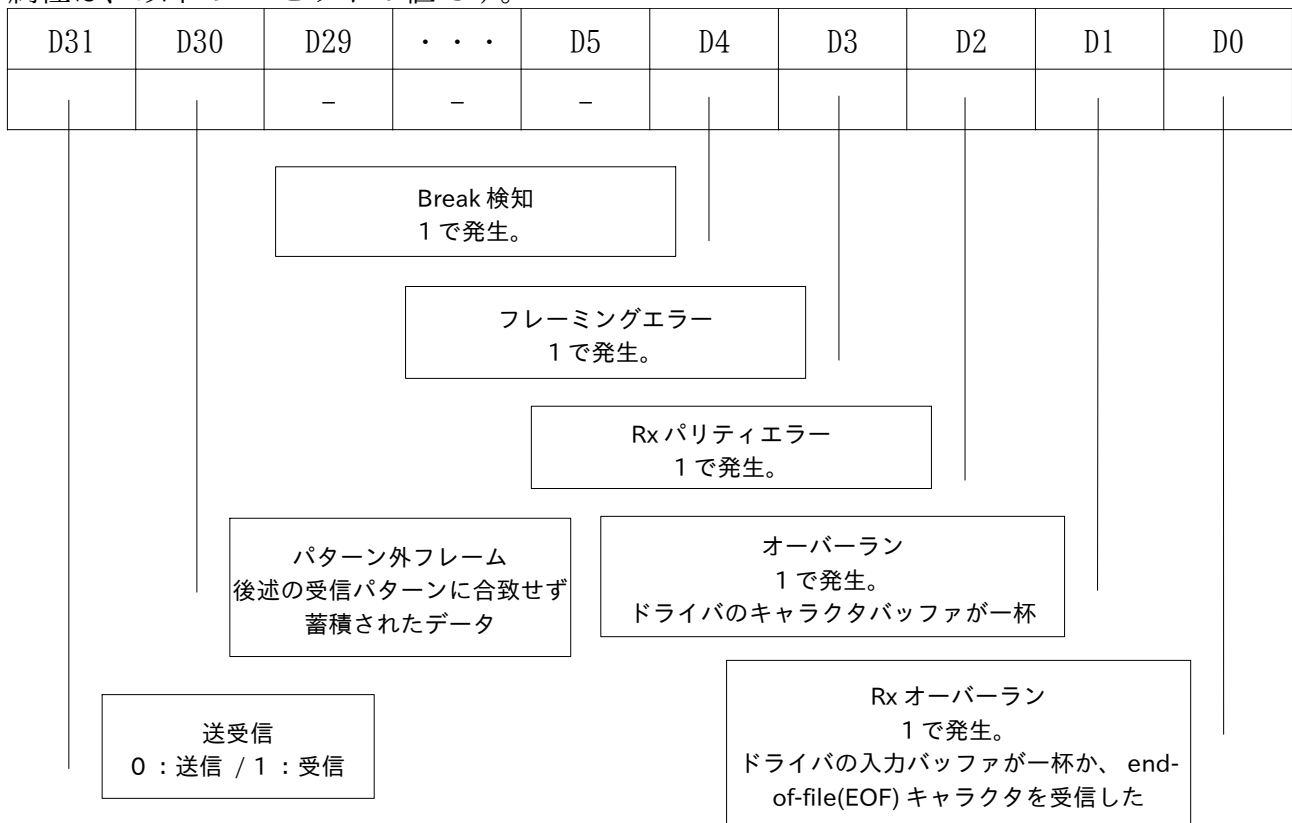
Execute -> Stop : 動作を停止します。

Execute -> Follow : スクロール追従の ON/OFF を切り替えます。ON 時は送受信データが表示されるとその位置にスクロールします。



属性と背景色

属性は、以下の 32 ビットの値です。



D0～D4 と D30 が「1」であるデータは、異常として背景赤で表示します。

Time(us)	Attribute	Data
3000220	00000000	023031320300
3001258	00000000	023031320300
3000816	00000000	023031320300
2999543	00000000	023031320300
3001106	00000000	023031320300
3000914	00000000	023031320300
2999920	00000000	023031320300
3001362	00000000	023031320300
2367638	c0000000	0230313233343536373839414243444546030002303132333435363738394142434445460300023031...
634394	00000000	023031320300
3001143	00000000	023031320300
3000277	00000000	023031320300
3001515	00000000	023031320300
3000950	00000000	023031320300
3001778	00000000	023031320300
3001329	00000000	023031320300

Time(us)	Attribute	Data
200499	00000000	02303132333435363738394142434445460300
106352	80000000	023031320300
12726	00000000	023031320300
81629	00000000	02303132333435363738394142434445460300
201288	00000000	02303132333435363738394142434445460300
199615	00000000	02303132333435363738394142434445460300
199840	00000000	02303132333435363738394142434445460300
200078	00000000	02303132333435363738394142434445460300
201876	00000000	02303132333435363738394142434445460300
199677	00000000	02303132333435363738394142434445460300
200992	00000000	02303132333435363738394142434445460300
200524	00000000	02303132333435363738394142434445460300
201004	00000000	02303132333435363738394142434445460300
201181	00000000	02303132333435363738394142434445460300
200669	00000000	02303132333435363738394142434445460300
200699	00000000	02303132333435363738394142434445460300
205021	00000000	02303132333435363738394142434445460300
199415	00000000	02303132333435363738394142434445460300
93029	80000000	023031320300
22088	00000000	02000102030405060708090a0b0c0d0e0f000102030405060708090a0b0c0d0e0f0300

受信データ(D31 が1)は、背景緑で表示します。

送信データ、その他のデータは背景白です。

データ作成方法

BeeCom の受信データフレーム認識

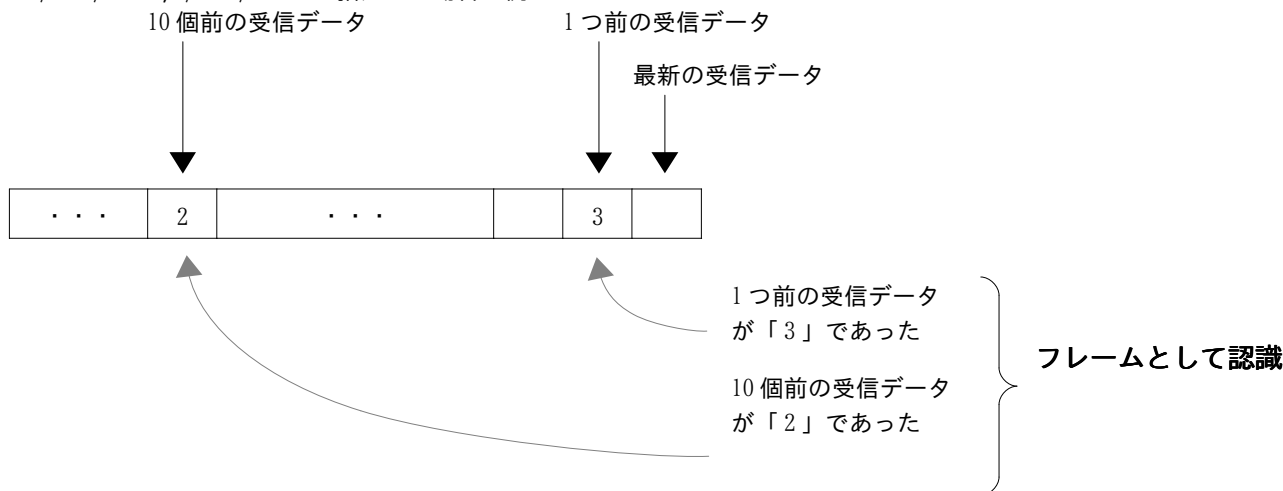
BeeCom は、

受信データをためていき、最新の受信データからさかのぼってパターンにマッチするか判断し、指定された全てのパターンにマッチしたらフレームとして認識する。

という方法で受信データを意味のある塊(フレーム)として認識します。

この動作は BeeCom を理解する上で重要なので、覚えておくようにしてください。

10/0xff/0x02;1/0xff/0x03 と指定した場合の例：



この方法は、理解するのが若干面倒ですが、

- ・フレーム中に ASCII コードでないバイナリデータが含まれる
- ・各々のデータ長は固定だが、違うデータ長を持つ複数のフレームがある
- ・しかも高速にどんどん送られてくる
- ・なおかつ通信環境が悪い、あるいはケーブルを抜き差しする

ような状況において、極めて高い確率でフレームを認識します。これを

index/mask/value;index/mask/value;...

index : 最新の受信データからのインデックス。0 で最新

mask : 受信データにかけるマスク。

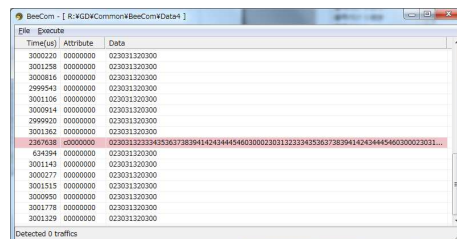
value : 比較値。受信データと *mask* を論理 AND したデータと比較される。

上記 *index/mask/value* のセットを ; で区切って複数を指定できる。

という書式で指定することでパターンを指定し、受信データをフレームとして認識させます。このとき、

- ・一番最初のパターンよりも前のデータ(この例では「2」よりも前のデータ)
- ・4096 バイト蓄積された
- ・受信タイムアウトを検知した

場合、そのデータは「パターン外」として背景赤で表示されます。



環境設定(Environments.csv)

環境設定は、以下の行・列のデータを読み込んで動作します。それ以外の行列は無視するのでコメント等記述しても問題ありませんが、行列がずれないようにご注意ください。

A	B	C
No.	設定項目	設定値
1	ポート名称	COM1
2	ボーレート	9600
3	パリティ	Even
4	ストップビット数	1
5	キャラクタビット幅	7
6	フロー制御	None
7	無受信タイムアウト	3000
8	無受信による Match Count クリアタイムアウト	10000

行列	項目	説明
B2	ポート名称	ポート名称を指定します。COM1 など。 ただし、ここで ホスト名: ポート番号 という指定を行うと、シリアル通信の代わりに指定したホスト名: ポート番号に Bind して UDP 通信を行います。ホスト名は省略可能で、省略した場合には自ホストです。
B3	ボーレート	ボーレートを指定します。設定可能な範囲は実デバイスに依存しますが、一般的に 1200～115200 程度です。
B4	パリティ	None、Even、Odd が指定可能です。
B5	ストップビット数	1、1.5、2 が指定可能です。
B6	キャラクタビット幅	8、7 が指定可能です。
B7	フロー制御	None、RTS/CTS が指定可能です。
B8	無受信タイムアウト	BeeCom の受信データフレーム認識で述べたように、BeeCom はパターンが認識されるまで(最大 4096 バイト)受信データを蓄積します。ここに数値を指定すると、受信がなくなってから指定した時間経過することで蓄積されたデータをクリアします。クリアされたときデータが蓄積されていれば、属性の D30 を立てて表示します。
B9	無受信による Match Count クリアタイムアウト	ここに数値を指定すると、受信がなくなってから指定した時間経過することで後述の Match Count を全てクリアします。BeeCom をスタートしなすことなく、指定した動作を最初からさせたい場合に使用します。

応答設定(Responses.csv)

応答設定は、以下の行列を読み込んで動作します。それ以外の行列は無視するのでコメント等記述しても問題ありませんが、行列がずれないようにご注意ください。

	A	B	C	D	E	F	
1	No.	TxD	Delay	Pattern	Match Count	Reset Counter	Plugins
2	1	Sample1	200		0	1	TextBcc.dll();
3	2	Sample2	10	5/0xff/0x02;1/0xff/0x03	1		TextBcc.dll();
4	3	Sample3	20	5/0xff/0x02;1/0xff/0x03	2		
5	4	Sample4	30	5/0xff/0x02;1/0xff/0x03	3	1	
6							

行列	項目	説明
B2～	TxD	送信データを定義した別ファイルの名前を指定します。送信データは、ここで指定した名前.csvが読み込まれます。
C2～	Delay	送信遅延時間を msec で指定します。 受信データへの応答である (Pattern が空でない) 場合には、受信してから送信するまでの遅延時間として使用され、自発的な送信である場合には送信の間隔として使用されます。
D2～	Pattern	パターンを指定します。パターンについては BeeCom の受信データフレーム認識を参照してください。パターンが空欄の場合、自発的に送信を行います。
E2～	Match Count	BeeCom は、単にパターンにマッチしただけでは応答しません。ここで指定した回数マッチして初めて応答します。パターンが空欄の場合、使用されません。
F2～	Reset Counter	パターンが Match Count 回数マッチし、応答を返すとき、ここで指定した回数に Match Counter をセットします。例えば、 <ul style="list-style-type: none"> ・ 1 回めは TxD1 を返す ・ 2 回めは TxD2 を返す ・ 3 回めは TxD3 を返す ・ 以降は 2 回めと 3 回めの動作を繰り返したい というようなとき、3 回めの TxD3 で Reset Counter を「1」とセットすると、次のパターンマッチで Match Counter が 2 となり、2 回め以降の動作を繰り返させることができます。 パターンが空欄 (自発送信) の場合、ここで 0 以上の数値を指定すると、再度送信を行います。結果的に一定時間間隔で送信が繰り返されることになります。
G2～	Plugins	プラグインを参照ください。

送信データ([TxD.]csv)

送信データは、以下の行列を読み込んで動作します。それ以外の行列は無視するのでコメント等記述しても問題ありませんが、行列がずれないようにご注意ください。また、ファイル名はResponsesで指定したファイル名になるようにしてください。

	A	B	C	D	E	
1		:30010				
2	Ofs	Bin	*ASCII	D7	D6	
3	0	0x02				
4	1	48	0			
5	2	49	1			
6	3	50	2			
7	4	0x03				
8	5	0				
9						

行列	項目	説明
A1	コメント	この部分に「//」を入力すると、送信がペンディングされます。
B1	UDP 送信時の宛先	環境設定(Environments.csv)でUDP通信の設定とした場合、ここで指定したホスト名:ポート番号に対して送信されます。ホスト名は省略すると自ホストです。通常のシリアル通信時は空欄でかまいません。
B3～	送信データ	送信データを設定します。指定できるのは10進数、16進数、8進数で、先頭が「0x」で始まるものは16進数、先頭が「0」以外の数字で始まるものは10進数で解釈されます。先頭を「0」で始め、数値を入力すると、8進数として解釈されるので注意してください。(例: 015と指定すると10進数の13(16進数の0x0d)として解釈される) 数値として解釈できないような指定がされた場合であってもデータ 0x00 を送信します。

データ作成用 OpenOffice Calc ドキュメント

データ作成用にOpenOffice Calcドキュメントを添付しています。これはEnvironmentsやResponsesをファイル保存と同時にCSV Exportするものです。データ作成にご利用ください。CSV Exportにマクロを使用しているため、起動時にマクロの警告が表示される場合がありますが、許可して実行してください。

プラグイン

BeeCom は、実行ディレクトリに配置された DLL を読み込み、その DLL によって機能拡張を行うことができます。DLL による機能拡張は、応答設定の Plugins 列によって制御されます。

Plugins 列は、以下のフォーマットで、「;」（コロン）で区切って指定します。

DLL 名称. 機能名称; DLL 名称. 機能名称; ...

DLL 名称 : DLL ファイルの拡張子を除いた部分。

機能名称 : 以下を参照ください。

No.	機能名称	機能
1	OnRx	受信時、DLL に配置された OnRx をコールします。 呼ばれた側の DLL にてカスタマイズされた動作を実装します。
2	OnTx	送信時、DLL に配置された OnTx をコールします。 呼ばれた側の DLL にてカスタマイズされた動作を実装します。

プラグインを実装することにより、例えば送信時に FCS を自動設定する、あるいは受信したデータ中のある要素をそのまま返送データに適用する、といった拡張が可能です。同梱のサンプル(Replyer、指定位置のデータを返送するだけの単純なもの)を参考に、プラグインを作成してみましょう！

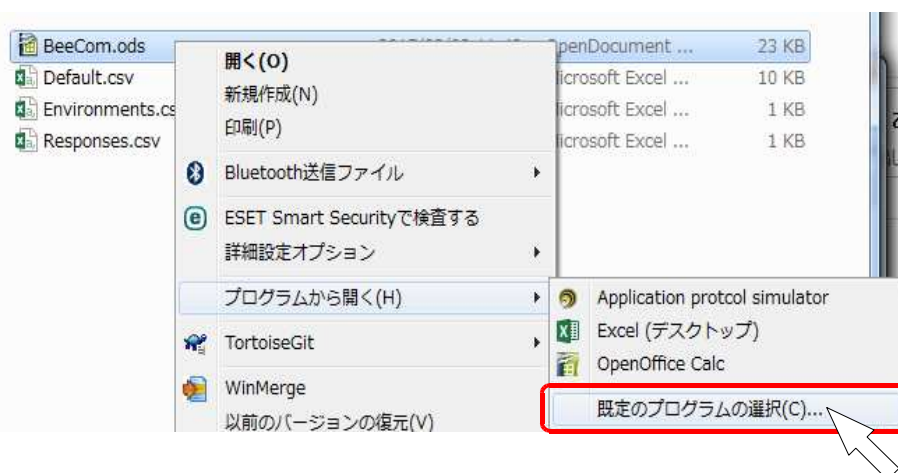
Tips:OpenOffice ドキュメントを右クリックすることで開けるようにする

BeeCom に、コマンドラインパラメータで OpenOffice ドキュメントを指定すると、あたかもそのドキュメントを開いたかのように動作させることができます。(実際の仕組みは右記参照。)

*.ods 拡張子に関連付けてしまえば早いのですが、OpenOffice Calc との関連付けがなくなるのも不便なので、以下の設定をお勧めします。

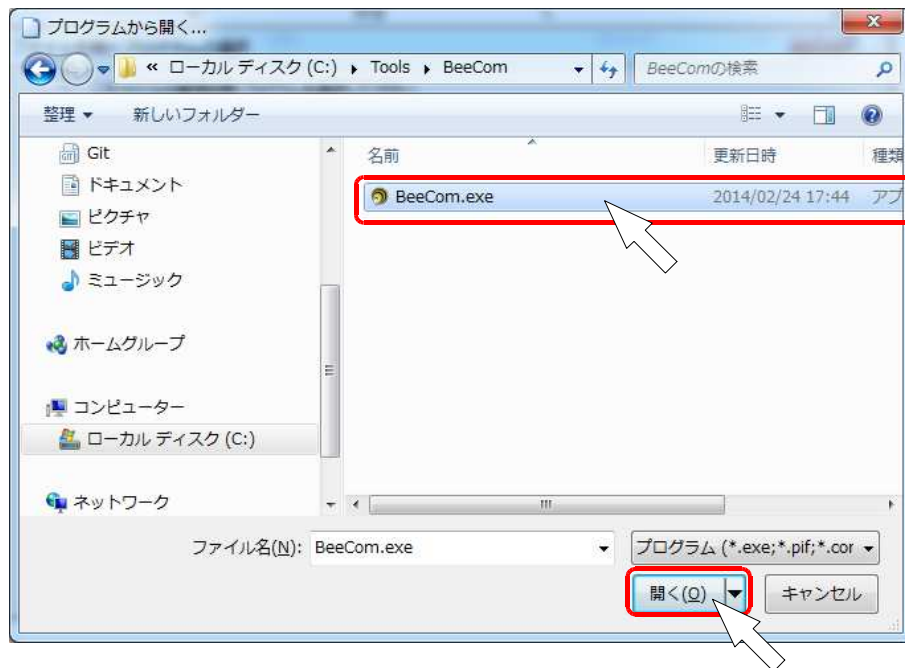
・ OpenOffice ドキュメントのマクロは、ドキュメントと同じディレクトリに csv をエクスポートする
・ BeeCom は、コマンドラインでファイルを指定すると、そのファイルのあるディレクトリが指定されたものとして動作する
コマンドラインで OpenOffice ドキュメントを指定すると、そのドキュメントの条件で動作するようになる

(1) OpenOffice Calc
ドキュメント(*.ods)を
右クリックし、「プログラムから開く」→
「既定のプログラムの
選択」をクリック



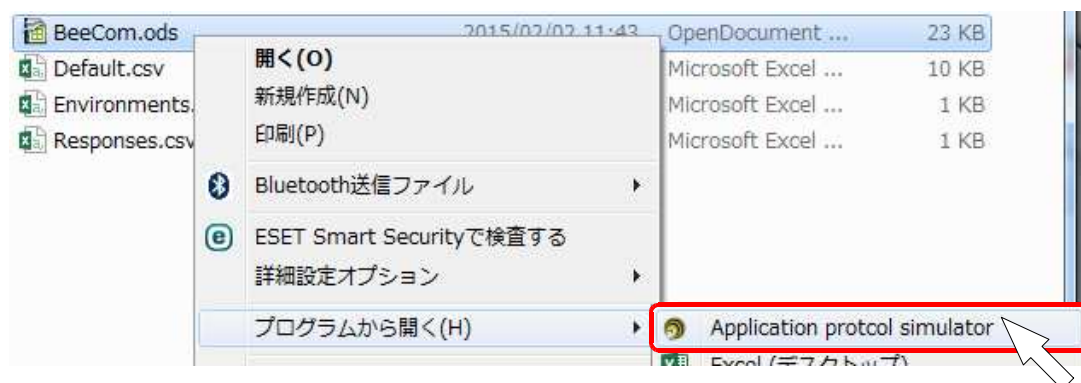
(2) 「ファイルを開くプログラムの選択」で「参照」をクリック

(3) インストールした
BeeCom.exe を選択し、
「開く」



(4) 「ファイルを開くプログラムの選択」に戻るので、「Application protocol simulator」が選択されていることを確認、また、「この種類のファイルを開くときは、選択したプログラムをいつも使う」のチェックが外れていることを確認し、OK。

(5) 次回以降は、OpenOffice Calc ドキュメントを右クリック→「プログラムから開く」に「Application protocol simulator」が追加され、それを選択することで起動できるようになります。



連絡先

ご意見、ご要望など、以下にて承ります。

beecom.hcysvc@gmail.com

Copyright(C) 2014- Hachiya-production LLC.