



インストールマニュアル（ビルド 009 用）

開発者：小林茂＋遠藤孝則＋増田一太郎

1 動作環境

- OS
 - Windows XP SP2/3 または Vista
 - Mac OS X Tiger 10.4 または Leopard 10.5
- ソフトウェア
 - Java 実行環境（J2SE 5.0 以上）
 - Flash Professional CS3/3・Flex Builder 3・Flex 3 SDK^{*1}
 - Processing (1.0)^{*2}
 - Ruby (1.8.6 系列・1.9 系列での動作は未確認)^{*3} + OSC ライブラリ^{*4}
- ハードウェア
 - Gainer I/O モジュール^{*5*6}または互換機
 - Arduino NG/Diecimila/LilyPad/Nano/Pro/Pro Mini^{*7}または Boarduino などの互換機
 - XBee 802.15.4 OEM RF モジュール^{*8}または ZB ZigBee PRO RF モジュール^{*9}
 - FIO (Funnel I/O) モジュール^{*10}

2 配布パッケージのディレクトリ構成

- documents/：仕様書やマニュアルなど
- hardware/：ハードウェア及びファームウェア
- libraries/：各ソフトウェアライブラリ
 - actionscript3/：AS3 用ライブラリ・サンプル・ソースコード
 - processing/：Processing 用ライブラリ・サンプル・ソースコード
 - ruby/：Ruby 用ライブラリ・サンプル・ソースコード
- LICENSE.txt：ライセンス条項
- README_en.txt：概要説明（英語版）
- README_ja.txt：概要説明（日本語版）
- server/：Funnel Server
 - funnel_server.jar：Funnel Server 本体
 - settings.txt：Funnel Server の設定ファイル

^{*1} <http://www.adobe.com/jp/products/flex/sdk/>

^{*2} <http://processing.org/download/index.html>

^{*3} <http://www.ruby-lang.org/ja/downloads/>

^{*4} <http://raa.ruby-lang.org/project/osc/>

^{*5} <http://www.triggerdevice.com/items/>

^{*6} http://www.sparkfun.com/commerce/product_info.php?products_id=8480

^{*7} <http://www.arduino.cc/en/Main/Hardware>

^{*8} <http://www.digi-intl.co.jp/digi/wireless/zigbee/xbee-series1-module.html>

^{*9} <http://www.digi-intl.co.jp/digi/wireless/zigbee-mesh/xbee-zb-module.html>

^{*10} http://www.sparkfun.com/commerce/product_info.php?products_id=8957

3 ドライバのインストール

Gainer I/O モジュール、Arduino I/O ボード、および XBee 無線モデムや FIO モジュールとの接続に使用する USB ⇔ XBee 変換モジュール^{*11}のほとんどは FTDI 社の USB-to-UART 変換ブリッジチップ FT232R シリーズを搭載しています。このため、最初にこのドライバをセットアップすればほとんどの全てのハードウェアを利用できます^{*12}。

3.1 Windows XP/Vista の場合

まず最初にドライバをダウンロードします。ウェブブラウザで次の URL にアクセスし、Operating System（オペレーティング・システム）の欄が Windows XP と書かれているドライバをダウンロードします。

<http://www.ftdichip.com/Drivers/VCP.htm>

ドライバのダウンロードが終わったら、ZIP ファイルを展開（右クリックして表示されるメニューから「全て展開…」を実行）します。展開が終わったら、I/O モジュールに USB ケーブルの片方のコネクタを接続し、もう片方のコネクタを PC 側に接続します。数秒後、ドライバのインストールを要求するダイアログが表示されますので、ダイアログにしたがってドライバのインストールを行います。

- ソフトウェア検索のため、Windows Update に接続しますか？
→ 「いいえ、今回は接続しません。」選択します
- インストール方法を選んでください
→ 「一覧または特定の場所からインストールする（詳細）」を選択します
- 次の場所で最適のドライバを検索する
→ 「次の場所を含める」をチェックし、先ほどドライバを展開したフォルダを指定します

同じような操作を 2 回要求されますので、2 回目も 1 回目と同様にドライバを展開したフォルダを指定します。

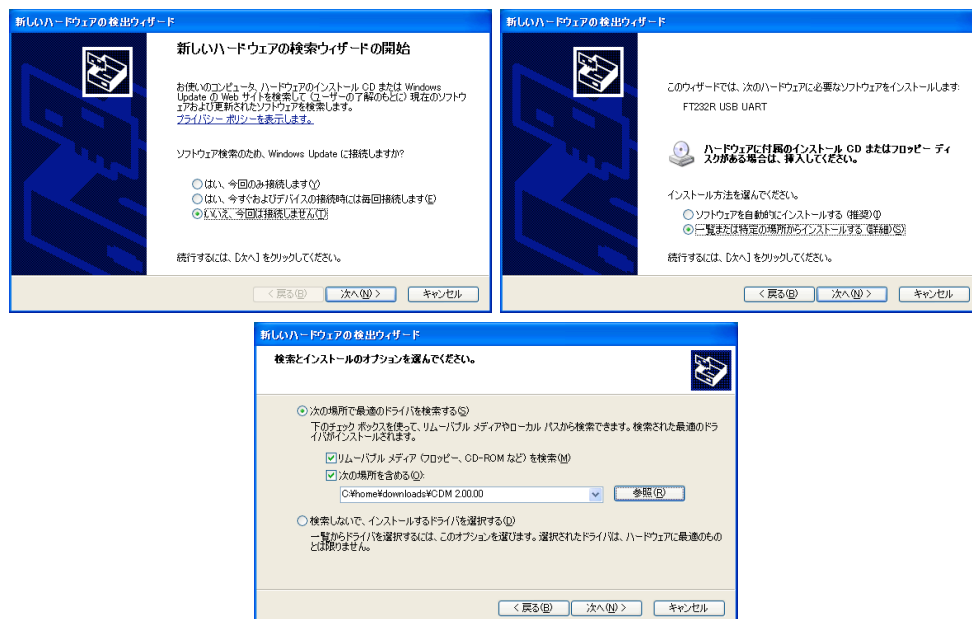


図 1 ドライバインストール時の画面例（Windows XP SP2）

^{*11} SparkFun Electronics の XBee Explorer USB（パーツナンバー WRL-08687）など

^{*12} Arduino を使用する場合には配布パッケージの drivers フォルダ（例：arduino-0012/drivers）にも含まれています。

3.2 Mac OS X の場合

まず最初にドライバをダウンロードします。ウェブブラウザで次の URL にアクセスし、Operating System（オペレーティング・システム）の欄が Mac OS X と書かれているドライバをダウンロードします。

<http://www.ftdichip.com/Drivers/VCP.htm>

ドライバのダウンロードが終わると、自動的にディスクイメージファイル（拡張子は dmg）がマウントされます。もし自動的にマウントされない場合には、ファイルをダブルクリックしてマウントします。マウントが終わると、自動的にドライバのインストーラが起動します。インストーラが起動したら、インストーラの指示に従ってドライバをインストールします。途中で管理者のパスワードを入力するように求められますので、パスワードを入力します。インストール終了時には、指示に従って再起動してください。

4 ハードウェアの準備

4.1 Gainer I/O モジュール

Gainer I/O モジュールの場合には、特にハードウェア側の準備は必要ありません。USB ケーブルで PC と接続し、後述する説明に従って Funnel Server を起動するだけですぐに利用できます。

4.2 Arduino I/O ボード

Arduino I/O ボードとの接続には「Firmata」^{*13}を使います。Firmata は Hans-Christoph Steiner が MIDI を参考に考案したプロトコルです。Arduino 0012 にはこの Firmata ライブラリが標準で含まれているため、ライブラリのサンプルを書き込むだけで簡単に I/O モジュールとして利用できるようになります。

Arduino 0012 をウェブサイト^{*14}からダウンロードし、以下の順で操作してください。

1. Tools → Board から適切なモデルを選択
2. Tools → Serial Port から該当するシリアルポートを選択
3. File → Sketchbook → Examples → Library-Firmata → StandardFirmata を選択
4. Upload ボタンを押してアップロード^{*15}

アップロードは 10 数秒で終了するはずです。もしエラーメッセージが表示された場合には、再度アップロードからやり直してみてください。

^{*13} <http://www.arduino.cc/playground/Interfacing/Firmata/>

^{*14} <http://arduino.cc>

^{*15} Arduino NG などでは必要に応じてアップロード前にリセットボタンを押してください。

4.3 XBee 無線モジュール

4.3.1 802.15.4 シリーズの場合

hardware/xbec/multipoint に入っている coordinator.pro を PC に接続するコーディネータに、enddevice.pro をセンサなどに接続するエンドデバイスに（必要に応じて設定を変更した上で）それぞれ書き込んでください。Windows の場合には Digi が提供する X-CTU^{*16}がそのまま利用できます。Mac OS X の場合には、Tom Igoe が Making Things Talk のサンプルとして公開している XBee Terminal^{*17}を利用して必要なコマンドを手動で入力することにより、設定を行うことができます^{*18}。表 1 にコーディネータとエンドデバイスの設定例を示します。なお、出力側も利用するためにはファームウェア 1.0.C.D が必要になります。

	コーディネータ	エンド・デバイス	備考
ATRE	-	-	設定をリセット
ATID	1234	1234	PAN ID
ATMY	0	1	
ATDL	FFFF	0	相手側の ID
ATDO	-	2	AD0/DIO0 をアナログ入力に
ATIR	-	32	サンプリング間隔 (0x32 = 50ms)
ATIT	-	1	ここで指定したサンプル数ごとに送信
ATAP	2	-	API モードの設定
ATWR	-	-	設定を書き込む
ATCN	-	-	コマンドモードを抜ける

表 1 802.15.4 シリーズの設定例

4.3.2 ZB ZigBee PRO シリーズの場合

hardware/xbec/zbに入っている coordinator.pro を PC に接続するコーディネータに、router.pro をセンサなどに接続するルータに（必要に応じて設定を変更した上で）それぞれ書き込んでください。ファームウェアは、コーディネータ側には 21xx (Coordinator - API Operation) を、ルータ側には 22xx (Router - AT/Transparent Operation) を書き込みます。表 2 に XBee Terminal を利用して手動で設定を書き込む場合の参考となる設定例を示します。なお、これは最小限の実験用の構成ですので、実際にメッシュネットワークを構成する際にはルータとエンドデバイス用意してそれぞれ適切に設定してください。

	コーディネータ	ルータ	備考
ATRE	-	-	設定をリセット
ATID	1234	1234	PAN ID
ATJV	-	1	コーディネータのチャンネルを確認
ATDL	FFFF	0	相手側の ID
ATD1	-	2	AD1/DIO1 をアナログ入力に
ATIR	-	32	サンプリング間隔 (0x32 = 50ms)
ATAP	2	-	API モードの設定
ATWR	-	-	設定を書き込む
ATCN	-	-	コマンドモードを抜ける

表 2 ZB ZigBee PRO シリーズの設定例

^{*16} <http://www.digi.com/support/productdetl.jsp?pid=3352&osvid=57&tp=4&s=316>

^{*17} <http://www.makingthingstalk.com/chapter6/31/>

^{*18} 現時点ではファームウェアのアップデートには X-CTU が必要となるため、Boot Camp などを利用して Windows を利用できる状態にしておくことをおすすめします。

4.4 FIO (Funnel I/O モジュール)

FIO は LilyPad Arduino Main Board v1.6^{*19}をベースに XBee を搭載できるようにしたもので、Arduino IDE によるファームウェアの書き込みと XBee の設定を行う必要があります。XBee は両シリーズ共に利用できますが、設定がシンプルな 802.15.4 を推奨します^{*20}。

4.4.1 XBee の設定

X-CTU を利用する場合には、`hardware/fio/xbec` に入っている `coordinator.auto_reset.pro` (自動リセットあり) または `coordinator.pro` (自動リセットなし) を PC に接続するコーディネータに、`enddevice.pro` を FIO に搭載するエンドデバイスに (必要に応じて設定を変更した上で) それぞれ書き込んでください。

Mac OS X の場合には、XBee Terminal を利用して表 3 を参考に必要なコマンドを手動で入力してください。なお、自動リセットの設定方法は「Xbee Adapter - wireless Arduino programming^{*21}」で紹介されていた方法を参考に一部改変したものです。自動リセットを使用したくない場合には※印の項目を設定しないようにしてください。

この設定は、FIO のファームウェアを無線で書き換えることを想定して、Arduino のブートローダで使用される通信速度である 19200bps にしています。複数の FIO を同時に使用する場合など、コーディネータ側の通信速度が不足する時はコーディネータ側の通信速度を 57600bps などに変更してください (ファームウェアをアップデートする際には戻してください)。

	コーディネータ	エンド・デバイス	備考
ATRE	-	-	設定をリセット
ATBD	4	4	通信速度を 19200bps に
ATID	1234	1234	PAN ID
ATMY	0	1	
ATDL	FFFF	0	相手側の ID
ATD3	3	5	DIO3 を DIN に / DIO3 を HIGH に※
ATIC	8	-	DIO Change Detect で DIO3 の変化を検出※
ATIU	-	0	I/O Output を無効に※
ATIA	-	FFFF	I/O Input Address を設定※
ATWR	-	-	設定を書き込む
ATCN	-	-	コマンドモードを抜ける

表 3 FIO 用の XBee 802.15.4 シリーズの設定例

4.4.2 ファームウェアの書き込み

Arduino 0012 をウェブサイト^{*22}からダウンロードし、以下の順で操作してください。

1. Tools → Board から Arduino Pro or Pro Mini (8MHz) を選択
2. Tools → Serial Port から該当するシリアルポートを選択
3. File → Sketchbook → Open で `hardware/fio/firmware/FioStandardFirmata` を選択
4. FIO 側の電源スイッチを入れるか、リセットスイッチを押したらすぐに Upload ボタンを押す

アップロードは 10 数秒で終了するはずですが、もしエラーメッセージが表示された場合には、再度アップロードからやり直してみてください。

^{*19} http://www.sparkfun.com/commerce/product_info.php?products_id=8465

^{*20} ZB ZigBee PRO シリーズの場合には、無線で FIO ファームウェアを書き換える際にはコーディネータ側 XBee のファームウェアを変更する必要があります。

^{*21} <http://www.ladyada.net/make/xbec/arduino.html>

^{*22} <http://arduino.cc>

5 Funnel Server の設定と起動

5.1 設定 (Gainer I/O モジュール以外の場合)

Funnel Server はデフォルトで Gainer I/O モジュールを使用する設定になっています。このため、Gainer I/O モジュール以外の I/O モジュールを使用する場合には、ハードウェアに合わせて Funnel Server の設定ファイル (settings.txt) を変更して下さい*²³。サンプルとして以下の設定ファイルも配布パッケージに含まれています。

- settings.arduino.txt : Arduino I/O ボード用の設定ファイル
- settings.fio.txt : FIO 用の設定ファイル
- settings.gainer.txt : Gainer I/O モジュール用の設定ファイル
- settings.xbee.txt : XBee 用の設定ファイル

Funnel Server は 1 つのネットワークポートを利用し、デフォルト値は 9000 番です。他のアプリケーションがこのポートを使用している場合には、競合しない番号に変更して下さい。Gainer I/O モジュールの場合は、シリアルポートを指定しないと自動検出を試みますので、1 台のみ I/O モジュールを接続するときには設定不要です。Mac OS X の場合は、FTDI の USB-to-UART ブリッジを使用している場合には自動での接続を試みますので、多くの場合には Arduino などでもシリアルポートの指定無しで接続できるはずです。

リスト 1 Gainer I/O モジュール用の settings.txt の設定例

```
server:
  port: 9000

io:
  type: Gainer
  com:
  baudrate:
```

リスト 2 Arduino I/O ボード用の settings.txt の設定例 (Mac OS X)

```
server:
  port: 9000

io:
  type: Arduino
  com: /dev/cu.usbserial-A*****
  baudrate: 115200
```

リスト 3 Arduino I/O ボード用の settings.txt の設定例 (Windows)

```
server:
  port: 9000

io:
  type: Arduino
  com: COM3
  baudrate: 115200
```

*²³ Windows の場合、ノートパッドでファイルを開くと改行コードの関係で全てが 1 行に表示されてしまうかもしれません。ワードパッドであれば問題なく編集できます。

5.2 起動

設定が終わったら、`funnel.server.jar` をダブルクリックして Funnel Server を起動してください。起動するとハードウェアとの接続を行い、その結果がメッセージとして表示されます。なお、Funnel Server 自体が起動しない場合は、J2SE 5 以上の Java 実行環境がインストールされているかどうか確認してください。

5.3 Gainer I/O モジュールの場合

正常に起動した場合には次のような画面が表示されます。何らかのエラーメッセージが表示される場合には、メッセージを参考に設定ファイルやハードウェアを確認してください。

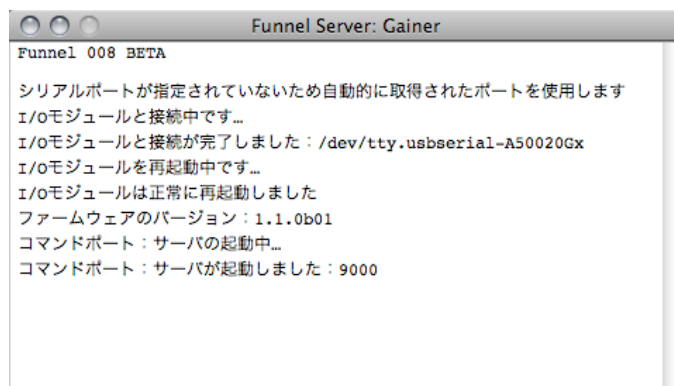


図2 Gainer I/O モジュールに接続した Funnel Server が正常に起動した時の画面例

I/O モジュールを接続しないで起動した場合、あるいは間違ってタイプを設定した場合には、次のようなメッセージが表示されます。この場合には、USB ケーブルがきちんと接続されているか、I/O モジュールの LED が点灯しているか、設定ファイルで別のタイプを設定していないかを確認してください。



図3 Gainer I/O モジュールを接続しないで Funnel Server を起動した時の画面例

5.4 Arduino I/O ボードの場合

正常に起動した場合には次のような画面が表示されます。何らかのエラーメッセージが表示される場合には、メッセージを参考に設定ファイルやハードウェアを確認してください。

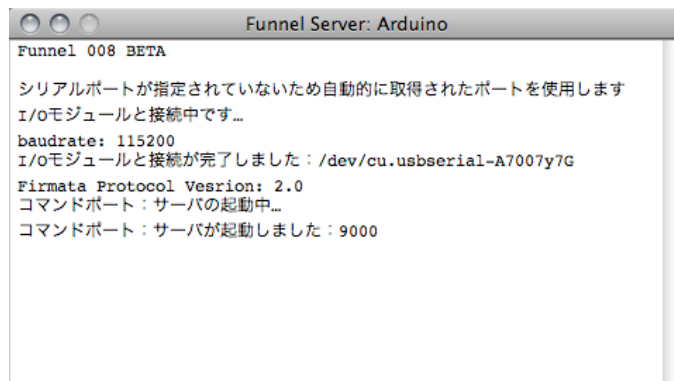


図4 Arduino I/O ボードに接続した Funnel Server が正常に起動した時の画面例

ここで「Firmata Protocol Version: 2.0」というメッセージが表示されない場合は、ファームウェアの書き込みが正しく行われていない可能性があります。セクション 4.2 の説明を参考に、ファームウェアを書き込んでください。

I/O ボードを接続しないで起動した場合、あるいは間違ってタイプを設定した場合には、次のようなメッセージが表示されます。この場合には、以下の項目について確認してください。

- USB ケーブルがきちんと接続されているか
- I/O モジュールの LED が点灯しているか
- 設定ファイルで別のタイプを設定していないか
- シリアルポートが正しく指定されているか
- シリアルポートの通信速度が正しく指定されているか

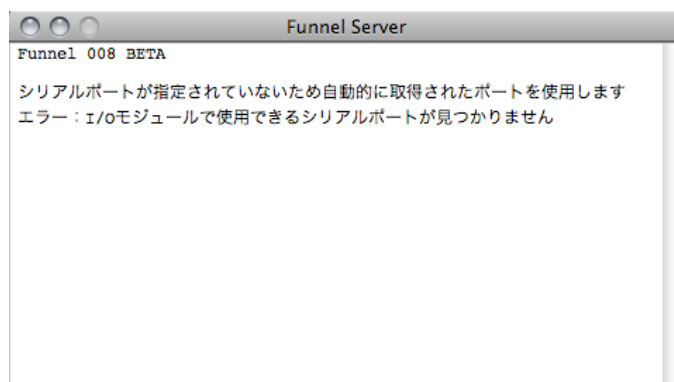


図5 Arduino I/O ボードを接続しないで Funnel Server を起動した時の画面例

5.5 XBee 無線モデムの場合

正常に起動した場合には次のような画面が表示されます。何らかのエラーメッセージが表示される場合には、メッセージを参考に設定ファイルやハードウェアを確認してください。

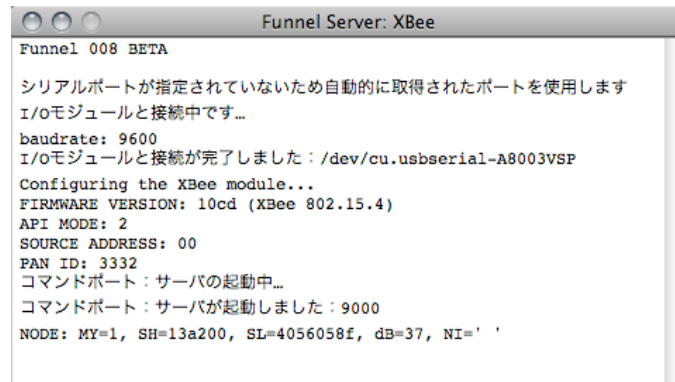


図6 XBee 無線モデムに接続した Funnel Server が正常に起動した時の画面例

XBee ⇄ USB 変換モジュールを接続しないで起動した場合、あるいは間違ってタイプを設定した場合には、次のようなメッセージが表示されます。この場合には、以下の項目について確認してください。

- USB ケーブルがきちんと接続されているか
- XBee ⇄ USB 変換モジュールの電源 LED が点灯しているか
- 設定ファイルで別のタイプを設定していないか
- シリアルポートが正しく指定されているか
- シリアルポートの通信速度が正しく指定されているか

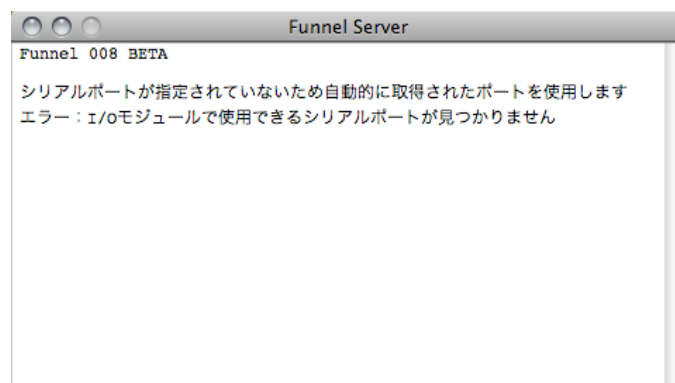


図7 XBee 無線モデムを接続しないで Funnel Server を起動した時の画面例

5.6 FIO の場合

正常に起動した場合には次のような画面が表示されます。何らかのエラーメッセージが表示される場合には、メッセージを参考に設定ファイルやハードウェアを確認してください。

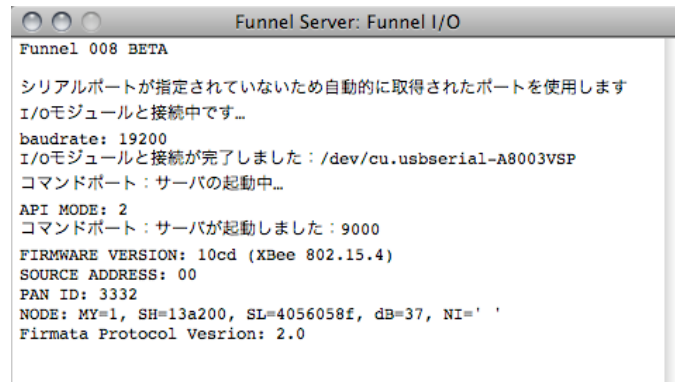


図 8 FIO (Funnel I/O モジュール) に接続した Funnel Server が正常に起動した時の画面例

ここで「Firmata Protocol Version: 2.0」というメッセージが表示されない場合は、ファームウェアの書き込みが正しく行われていない可能性があります。セクション 4.4.2 の説明を参考に、ファームウェアを書き込んでください。

XBee ⇄ USB 変換モジュールを接続しないで起動した場合、あるいは間違ってタイプを設定した場合には、次のようなメッセージが表示されます。この場合には、以下の項目について確認してください。

- USB ケーブルがきちんと接続されているか
- XBee ⇄ USB 変換モジュールの電源 LED が点灯しているか
- 設定ファイルで別のタイプを設定していないか
- シリアルポートが正しく指定されているか
- シリアルポートの通信速度が正しく指定されているか

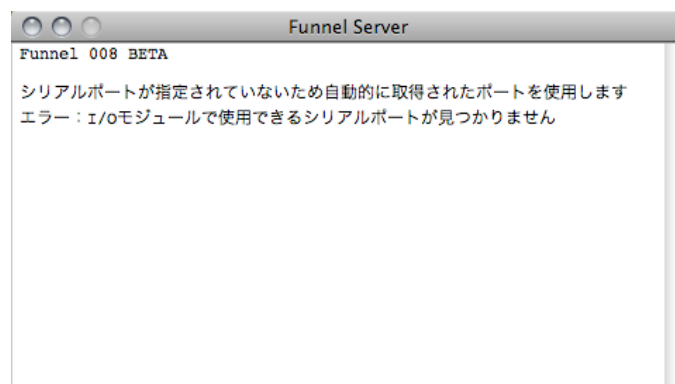


図 9 FIO (Funnel I/O モジュール) を接続しないで Funnel Server を起動した時の画面例

6 各ライブラリごとの準備と使用方法

6.1 ActionScript 3

6.1.1 準備

libraries/actionscript3/examples/に各ハードウェア用のサンプルがあります。例えば、Gainer I/O モジュール用のサンプルは GainerTest.as、Flash IDE 用のファイルは GainerTest fla、パブリッシュ済みの Flash Player 用ファイルは GainerTest.swf になります。Flash IDE 以外でパブリッシュする場合には、それぞれの環境に合わせて libraries/actionscript3/src/をソースパスに設定して下さい。

リスト 4 mxmhc でサンプルをコンパイルする際のオプションの設定例

```
$ mxmhc GainerTest.as -sp ../src
```

次に、ウェブブラウザで <http://tinyurl.com/2rg3lq>^{*24} にアクセスして、「グローバルセキュリティ設定」の「これらのファイルとフォルダを常に信頼する→追加→フォルダを参照...」でサンプルのフォルダを追加してください。この設定を行わないと、Funnel Server と Flash Player の通信ができません。

6.1.2 動作確認

libraries/actionscript3/examples/*Test.swf を Flash Player で再生してください。それぞれのサンプルで実装されている内容に関しては、*Test.as のコメントを参照してください。

6.2 Processing

6.2.1 準備

1. funnel.009/libraries/processing/library/をマイドキュメント/Processing/libraries/funnel/library/ (Windows の場合) 書類/Processing/libraries/funnel/library/ (OS X の場合) としてコピーしてください。
2. この状態で Processing を起動し、「Sketch → Import Library...」のメニュー項目として「funnel」が表示されるのを確認してください。確認が終わったら Processing を終了してください。
3. 必要に応じて、Processing のスケッチブックフォルダ (Windows の場合にはマイドキュメント/Processing、OS X の場合には書類/Processing) の中に libraries/processing/sketch_samples/をコピーしてください。

6.2.2 動作確認

Open ボタンをクリックし、sketch_samples フォルダの中からそれぞれのハードウェアのスケッチを開いて実行してください。それぞれのサンプルで実装されている内容に関しては、スケッチ中のコメントを参照してください。

6.3 Ruby

6.3.1 準備

libraries/ruby/examples に Gainer、Arduino、XBee、Funnel I/O 用のサンプルがあります。また、libraries/ruby/examples/action-coding/にある action-coding 用のサンプルを実行する場合には、<http://code.google.com/p/action-coding/wiki/Tutorial> を参照して action-coding の設定を済ませてください。

^{*24} http://www.macromedia.com/support/documentation/jp/flashplayer/help/settings_manager04.html

6.3.2 動作確認

Ruby から `libraries/ruby/examples/` フォルダの中からそれぞれのハードウェアのスクリプトを開いて実行してください。それぞれのサンプルで実装されている内容に関しては、スクリプト中のコメントを参照してください。