

## [APNOTE07]

# リモート XBee デバイス計測値を定期的に DB に保管、クライアント PC のエクセルから集計(ポーリング方式)

---

ABS-9000 DeviceServer

APNOTE07 Rev A.1.1

2010/03/31



オールブルーシステム (All Blue System)

ウェブページ: [www.allbluesystem.com](http://www.allbluesystem.com)

コンタクト: [contact@allbluesystem.com](mailto:contact@allbluesystem.com)

## 1 イン트로ダクション

複数のリモート XBee<sup>1</sup> デバイスの A/D 変換値 (DI02, DI03) を定期的に取得して、随時クライアントPCから集計を行うシステムについて説明します。スクリプトを修正することで、Digital I/O 入力値の集計等に簡単に変更できます。計測データの取得タイミングは、サーバー側でスケジュールされた間隔で行います（ポーリング方式）。

XBee デバイスは、Digi International Inc. 社製の IEEE 802.15.4 RF モジュールを使用します。計測用のリモート XBee モジュールには、外付けのマイクロコントローラ等を使用せず、XBee 自身が持つ A/D 変換機能のみを使用します。

DeviceServer の COM ポートに接続された XBee デバイスを経由して、複数の XBee デバイスをリモート操作し、A/D 変換入力を定期的に取得して、データベースに保管します。データベースは DeviceServer に組み込まれた Firebird DBMS を使用します。データ保管用のデータベースとして外部の Oracle 10g サーバーを使用することもできます。

リモート XBee デバイスから計測データを取得するときにエラーが発生した場合は、予め決められたメールアドレスに、エラー発生を知らせるメールを送信することができます。

クライアントPC のエクセルから、計測データの期間とデバイス名を指定して、データベースに保管されたデータを取得し、集計やグラフの作成等を行うことができます。

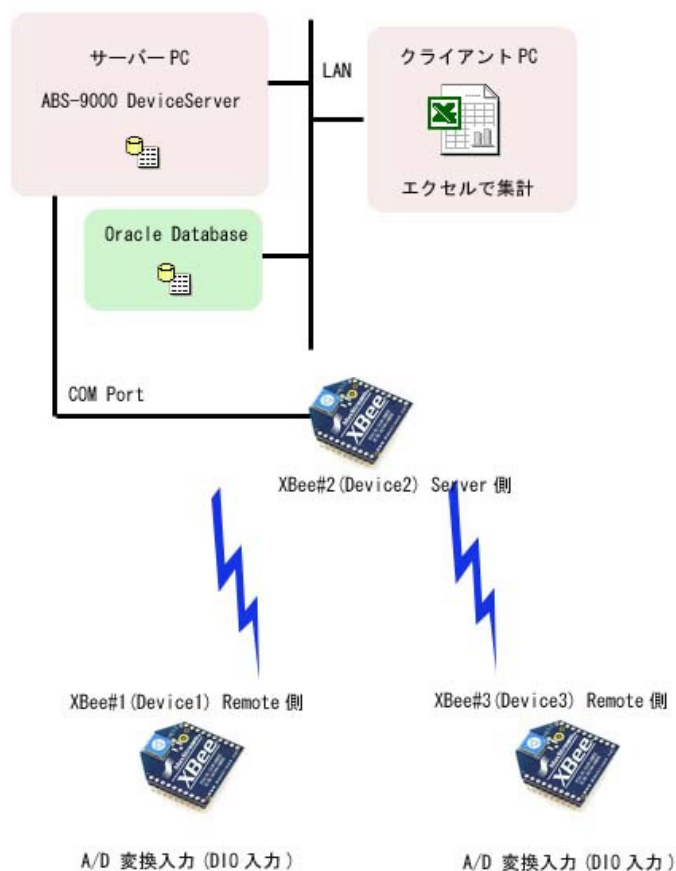
## 2 必要な機材・リソース

| 必要なシステムやデバイス等                               | 説明                                                                                                                                                                                                                                  |
|---------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ABS-9000 DeviceServerの動作しているPC              | DeviceServer の動作するPC が必要になります。                                                                                                                                                                                                      |
| ABS-9000 DeviceServerの動作しているPC の Windows OS | タスクスケジュールコマンド (schtasks) を使用するため、DeviceServer の動作している OS が WindowsXP もしくは Windows2003 である必要があります。                                                                                                                                   |
| XBee デバイス                                   | Digi International Inc. 社製 XBee IEEE 802.15.4 デバイスが 2台以上必要です。DeviceServer のCOM ポート直接接続する XBee デバイスが1台、その他はリモート計測側の XBee デバイスになります。<br><br>DeviceServer は、XBee デバイスのファームウェアバージョンの “10GD” にのみ対応しています。(必要に応じて XBee ファームウェアの更新を行ってください) |

<sup>1</sup> XBee XBee® and XBee PRO® are registered trademarks of Digi, Inc.

|                        |                                                                                                                                                                                               |
|------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SMTP メールアカウント (オプション)  | エラー発生時にメールを送信するSMTPサーバーメールアドレス。<br>(メール送信を行わない場合は必要ありません)                                                                                                                                     |
| Oracle10g サーバー (オプション) | DeviceServer の動作するPC もしくは 別 PC にOracle10g サーバーと、<br>DeviceServer の動作するPCに Oracleクライアントが必要です。<br>(保管対象のデータベースをFirebird からOracleに変更する時だけ、必要になります)<br><br>詳細は “DeviceServer ユーザーマニュアル”を参照してください。 |

### 3 システム構成図



### 4 システム動作概要

- DeviceServer の動作するPC で、リモートXBee デバイスの A/D 変換値を定期的に取り込む様にスケジュールします。スケジュール実行では Windows のタスクスケジューラによって、5 分毎に ScriptExecCmd.exe プログラム (DeviceServer のクライアントプログラム) が起動されて XBEE\_ACQUISITION スクリプトを実行します。
- XBEE\_ACQUISITION スクリプトは、予めXBee デバイス管理プログラム (XbeeConfig) に登録済みのリモート XBee デバイスを検索して、全てのデバイスに対してA/D 変換値の取り込みを行います。その後、タイムスタン

ブとデバイス名に応じたキー名で、データベースに取得した計測値を保管します。

- リモートXBee デバイスの A/D 変換値取り込みに失敗した場合は、エラーメールを送信するために XBEE\_ERROR\_MAIL\_SEND スクリプトを実行します。
- リモートPCもしくはサーバーPC のエクセルから、DeviceServer 経由でデータベースに保管された計測値を取り込みます。集計対象デバイスと期間を指定して、計測データをエクセルのワークシートのセルにロードして、集計作業やグラフ作成などを行います。

## 5 設定手順

### 5.1 XBee デバイス初期設定

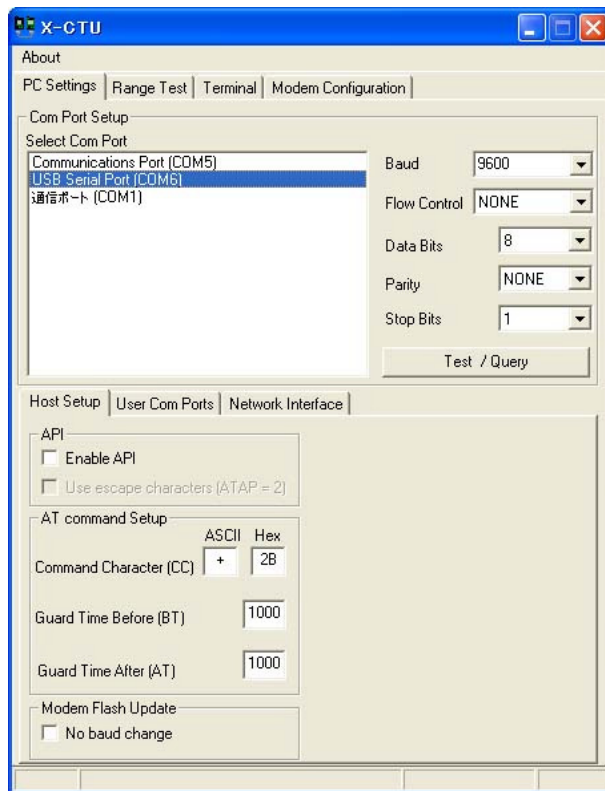
XBee デバイスを DeviceServer に接続するために、初期設定が必要です。DeviceServer の COM ポートに接続する XBee デバイスとリモート計測側の XBee デバイスで下記の設定を行ってください。

最初に DeviceServe との接続に必要な最低限の設定を COM ポート経由で行います。Sparkfun Electronics 社製の XBee Explorer USB 等を使用して、仮想 USB ポート経由で接続して設定してください。（これ以外の方法で COM ポート接続する場合も手順は同じです）

| XBee デバイス デフォルト値から変更が必要な設定値   |                                                                                                                                                                                                                                                       |
|-------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| API モード                       | 1 (default は 0)                                                                                                                                                                                                                                       |
| PAN(Personal Area Network) ID | 任意の値 (default は0x3332)<br><br>デフォルトの値のままだと、予期しないデバイスからのフレームを受信したり、間違ってデバイスを操作する恐れがありますので、適当な任意の値を設定するようにしてください。このマニュアルでは 0xAB90 を使用しています。                                                                                                             |
| 16bit Source Address          | 同一PAN ID 内でユニークな値 (default は 0x0000)<br><br>この値は、ここで設定しなくても後から XBee 管理プログラムで設定することが可能ですが、デバイス一覧から選択したデバイスがどのデバイスであるかを見分けることが容易になるように便宜的にここで設定します。<br><br>全てのデバイス間で違った値を設定してください。<br>(0x0000, 0xFFFF, 0xFFFE を除く)<br><br>例えば、0x0001, 0x0002, 0x0003 等。 |

上記3つの 初期設定のコマンドをXBee に送信するために、XBee デバイスを PC のCOM ポートに接続して Digi international Inc. 社製の X-CTU プログラム、または汎用のターミナルエミュレータプログラム等を使用します。XBee とCOM ポートのボーレートは初期設定の 9600 bps にして下さい。（ターミナルエミュレータを使用する場合は、ローカルエコー ON, 受信時の改行 CR + LF にするとコマンド実行の結果が見やすくなります）

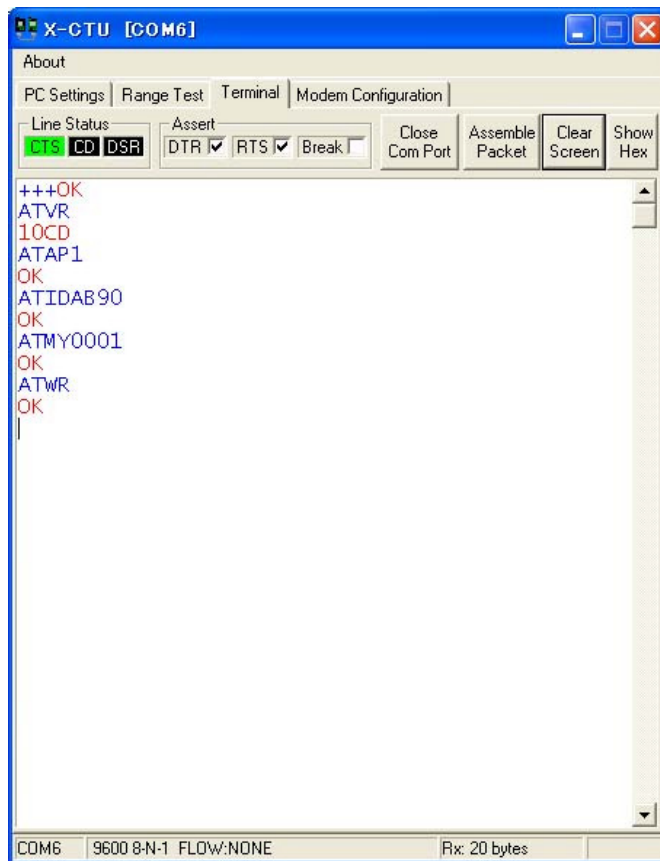
X-CTU プログラムを起動して、COM ポートを選択します。ここでは、USB Serial Port (COM6) を選択しています。



Terminal タブを選択してターミナル画面を表示します。キーボードから、“+++” を入力して、コマンドモードに入ります。コマンドモードに入ると “OK” が表示されますので、続けて以下のコマンド文字列を入力してください。コマンド入力の時間がかかりすぎると、自動的にコマンドモードから抜けてしまいますので、その場合は、“+++” を入力して最初からコマンドを入力し直して下さい。

```
ATVR
ATAP1
ATIDAB90
ATMY0001
ATWR
```

最初に ATVR でファームウェアバージョンを表示しています。“10CD” 以降になっていることを確認してください。ATAP1 は、API モードを “1” に設定しています。ATIDAB90 は PAN\_ID を 0xAB90 に設定しています。もし別の PAN\_ID を使用する場合は適宜変更してください。次に、ATMY0001 で、デバイスの16 bit Source Address を “0x0001” に設定しています。この部分は、デバイスごとにユニークな値になるように変更して下さい。最後に、ATWR で、設定値を不揮発メモリに書き込みます。実際に入力した時の画面表示は以下のようになります。



X-CTU プログラムを終了します。その後、XBee Explorer USB に接続する XBee デバイスを切り替えて、使用する全ての XBee デバイスについて同様に初期設定を行って下さい。

このときに、設定した16bit Source Address の値をデバイス機器にマーキングしておくと、後で XBee デバイス管理プログラムでデバイスを選択するときに、識別し易くなります。

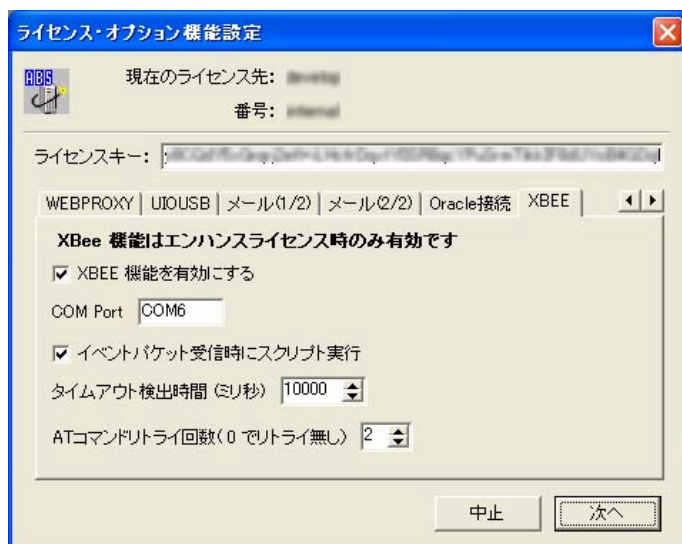
システム構成図上の3つのデバイスを接続する場合の設定値例は、以下のようになります。

| デバイス         | API モード(ATコマンド) | PAN_ID(ATコマンド)   | 16bit Address(ATコマンド) |
|--------------|-----------------|------------------|-----------------------|
| XBee#1(リモート) | 1(ATAP1)        | 0xAB90(ATIDAB90) | 0x0A01(ATMY0A01)      |
| XBee#2(サーバー) | 1(ATAP1)        | 0xAB90(ATIDAB90) | 0x0B02(ATMY0B02)      |
| XBee#3(リモート) | 1(ATAP1)        | 0xAB90(ATIDAB90) | 0x0C03(ATMY0C03)      |

## 5.2 XBee デバイスを DeviceServer に接続

XBee デバイスの初期設定後に、XBee#2(サーバー) のみを PC (DeviceServer) の COM ポートに接続します。その他のリモート XBee デバイスも電源を入れて通信可能な状態にしておきます。

デバイスが PC に接続されたら、DeviceServer から XBee デバイスを使用可能にするために COM ポートの設定を行います。サーバー設定プログラム(ServerInit.exe)を起動して、XBEE タブを選択して COM ポート番号を設定して“XBEE 機能を有効にする”にチェックをつけてください。サーバー設定プログラムの“次へ”を押して”完了”ボタンが表示されるまで進めて設定を完了して下さい。



### 5.3 マスター登録と XBee 詳細設定

XBee デバイスを DeviceServer のマスターファイルに登録します。XBee デバイスの初期設定で設定しなかった NodeIdentifier と A/D 変換ポート等の詳細設定もここでを行います。

システム構成図上の3つのデバイスを接続する場合の設定値例は、以下のようになります。

| デバイス         | 16bitAddress | Node Identifier | D102     | D103     |
|--------------|--------------|-----------------|----------|----------|
| 0x0A01(リモート) |              | Device1         | ADC      | ADC      |
| 0x0B02(サーバー) |              | Device2         | Disabled | Disabled |
| 0x0C03(リモート) |              | Device3         | ADC      | ADC      |

XBee デバイス管理プログラムを使用して、同一 PAN ID をもつ XBee デバイスを DeviceServer に登録したり、デバイス自身の設定内容を変更します。プログラムメニューから“ALL BLUE SYSTEM”->“クライアント起動”を選択・実行します。ログインするときは、管理者特権をもったユーザー（例えば DeviceServer セットアップ時に管理者アカウントとして登録したユーザー等）でログインしてください。デスクトッププログラムが起動したら、“XBee” ツールボタンを選択してXBee デバイス管理プログラムを起動します。



DeviceServer では登録済みの XBee デバイスをマスターファイルに記録しています。

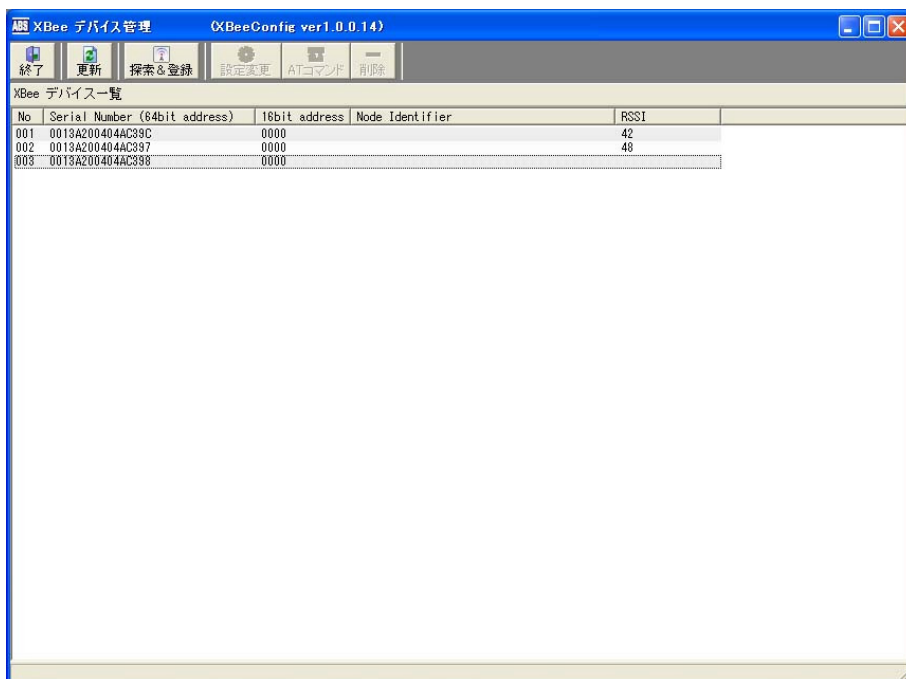
XBee デバイス登録は、“探索&登録” ボタンを押すことで、同一 PAN ID のリモートデバイスを見つけて、自動的にマスターファイルに登録します。既に、登録済みのXBee デバイスの場合は最新の情報でマスターファイルの内容が更新されます。DeviceServer に COMポートで直接接続された XBee デバイスについても同様に自動登録されます。

XBee デバイス管理プログラムの“探索&登録” ツールボタンを押します。



登録確認ダイアログが表示されますので、“OK” を押します。

DeviceServer の COM ポートに直接接続された XBee デバイスで “Node discover” が実行され、付近にある同一 PAN ID の XBee デバイス情報を取得して、自動的にマスターファイルに登録が行われます。XBee デバイス管理プログラムのデバイスイ覧には、登録済みのXBee デバイスが表示されます。



(デバイスの探索 & 登録が完了したときの画面)

XBee デバイスの詳細設定を変更するために、XBee デバイス管理プログラムのデバイスイ覧から対象デバイスを選択して、“設定変更” ツールボタンを押します。初期設定時に 16 bit Source Address を設定した場合は、その値がデバイスイ覧に表示されていますので、変更対象の XBee デバイスを確認することができます。



選択したXBee デバイスと通信を行って、現在のデバイス情報を取り込みます。

もしエラーが発生した場合は、選択したXBee デバイスとの間で通信ができない状態になっていますので、通信経路や電源等を確認してください。

XBee デバイスの現在の設定値を取り込んだ後、詳細設定変更ダイアログが表示されます。

ここでは、XBee デバイスの Node Identifier の設定を行って下さい。Node Identifier に指定可能な文字は ASCII で 20文字までです。また、ダイアログに表示されている 16 bit Source Address が、対象のデバイスであるかどうかの確認も行ってください。ここで 16 bit Source Address を任意の値に変更することも可能です。

リモート計測側の XBee デバイス (Device1, Device3) の場合のみ、DIO/PWM Config のタブ中から DIO2, DIO3 を選



択して“ADC”を選択します。DIO 設定タブ右側にある DIO ポート毎の Pullup Register Enable 設定を、ADC に選択したポートについてはチェックを外しておきます。

XBee デバイス(Device1)の詳細設定

Serial Number: 0013A200404AC39C  
Firmware Version: 10CD  
Channel: 0C  
PAN ID: AB90  
Node Identifier: Device1  
16bit Source Address: 0A01  
Destination Address High: 00000000  
Low: 00000000  
DeviceServer1に接続したXBee をDestinationに指定  
Sample Before TX: 1  
Sample Rate(x1ms): 0

DIO/PWM Config

DIO0 | DIO1 | **DIO2** | DIO3 | DIO4 | DIO5 | DIO6 | DIO7 | DIO8 | DIO9

☐ 0 Disabled  
☐ 1 (n/a)  
☒ 2 ADC  
☐ 3 DI  
☐ 4 DO Low  
☐ 5 DO High

☐ Pullup Register Enable  
☐ Change Detect

☒ 設定内容をXBee デバイスの不揮発メモリに書き込む

OK Cancel

最後に、“設定内容を XBee デバイスの不揮発メモリに書き込む” にチェックを付けて“OK”を押してください。

XBeeデバイスの Node Identifier を変更した場合は、XBee デバイス管理プログラムのデバイス一覧に表示されている、マスターファイルも更新しておく必要があります。

XBee デバイス管理プログラムの“探索&登録” ツールボタンを押します。



デバイスのNode Identifier または 16bit Source Address以外の詳細設定を変更する場合には、マスターファイルの更新は必要ありません。

## 5.4 サーバー設定(Mail, Oracle 接続)

リモート XBee からのデータ取得に失敗したときに、エラーメール送信を行う場合はメール関連の設定を行います。

また、計測データ保管先をデフォルトの Firebird から Oracle に変更する場合も設定が必要です。

プログラムメニューから”サーバー設定”を選択して、サーバー設定プログラムを起動します。使用しているメール環境に対応した設定を行ってください。

| サーバー設定プログラム |                                                 |
|-------------|-------------------------------------------------|
| 設定が必要な項目    | 設定内容                                            |
| 自分のメールアドレス  | DeviceServer からメール送信するときのデフォルトメール送信元アドレスを記入します。 |
| SMTPサーバー    | プロバイダのSMTPサーバーの設定に合わせて下さい                       |
| SMTPユーザー    | プロバイダのSMTPサーバーの設定に合わせて下さい                       |
| SMTPポート     | プロバイダのSMTPサーバーの設定に合わせて下さい                       |

計測データ保管用のデータベースに Oracle10g サーバーを使用する場合は、サーバー設定プログラムで、下記の項目を設定します。このほかにもOracle サーバーに保管用テーブル作成と、オラクルクライアントの設定が必要となります。詳細は “DeviceServer ユーザーマニュアル” を参照してください。

| サーバー設定プログラム (Oracle 使用時のみ設定が必要) |                                                                            |
|---------------------------------|----------------------------------------------------------------------------|
| 設定が必要な項目                        | 設定内容                                                                       |
| Oracle接続機能を有効にする                | チェックを付ける                                                                   |
| Oracleユーザー名                     | Oracle サーバーの設定に合わせて下さい                                                     |
| Oracleパスワード                     | Oracle サーバーの設定に合わせて下さい                                                     |
| ホスト接続文字列                        | Oracle クライアントの設定に合わせて下さい                                                   |
| 同時接続数                           | 4を設定します。Oracle クライアントライセンスによっては、設定可能な接続数に制限がある場合があります。(1 以上の値を設定する必要があります) |

## 5.5 スクリプト設定

### 注意

スクリプト中に日本語を記述するときは、スクリプトファイルを UTF-8N 形式で保存してください。Shift\_JISや UTF-8 BOM付き形式などで保存すると、DeviceServer でエラーが発生します。Windows付属のワードパッドやメモ帳ではこの形式で保存できませんので、別途 UTF-8N 形式で保存可能なエディタソフト（\*1）を使用してください。

（\*1）TeraPad 等のソフトウェアがよく使用されています。

### 5.5.1 XBEE\_DEV\_LIST スクリプト作成

リモート XBee デバイス一覧を取得するためのスクリプトを作成します。XBEE\_ACQUISITION スクリプトと、エクセル中の VBA からこのスクリプトが実行されます。DeviceServer のマスターファイルに登録された XBee デバイスを検索してスクリプト呼び出し元にデバイス一覧を返します。

ファイル名 (XBEE\_DEV\_LIST.lua) で DeviceServerのスクリプトフォルダに保管します。

```
--[[
*****

アキュイジション対象のリモートXBeeデバイスのリストをリターン値に返す
DeviceServer の COM ポートに接続した XBee デバイスを除く、全ての
マスターに登録された XBee デバイスを対象に選択している。

もし、個別のデバイスを指定したい場合は、xbee_all_list() を使用しないで
下記の様にデバイスを個々にリターン値に返す様にする。

XBee の NodeIdentifier が "DeviceA", "DeviceB" の 2 つのデバイスを
アキュイジション対象にする場合は、スクリプトを下記の様にする。

script_result の第 2 パラメータ（キー名）はユニークな文字列であれば
なにを指定しても構わない。

script_result(g_taskid, "1", "DeviceA")
script_result(g_taskid, "2", "DeviceB")
*****
]]

local stat, serial, addr, name, is_local = xbee_all_list();
for key, val in ipairs(name) do
    if not is_local[key] then
        script_result(g_taskid, tostring(key), val)
    end;
end;
--[[
script_result(g_taskid, "1", "Device1")
script_result(g_taskid, "2", "Device3")
]]
```

### 5.5.2 XBEE\_ACQUISITION スクリプト作成

リモート XBee デバイス A/D 変換入力を読み込んで、データベースに格納するスクリプトを作成します。

スクリプトは、定期的に起動されることを想定しています。起動された時刻に応じてタイムスタンプ文字列を決定した後、下記のキー名でデータベースに値を格納します。1 分以内にこのスクリプトが 2 回以上起動されると、後に実行された計測値で上書きされます。（同一キー名になるため）

|     | データベースに保管するキー名                     | 値                                                                              |
|-----|------------------------------------|--------------------------------------------------------------------------------|
| A/D | ACQADC-<DeviceName>-<YYYYMMDDHHMM> | <ADC0>, <ADC1>, <ADC2>, <ADC3>, <ADC4>, <ADC5><br>ADCxx の値は 0 から 1023 までの整数が入る |

|  |                                  |
|--|----------------------------------|
|  | ADCxx のサンプリング値が無い場合は、“NA” 文字列が入る |
|--|----------------------------------|

<DeviceName> には、XBee デバイスの NodeIdentifierが入ります。

<YYYYMMDDHHMM> には、西暦日付と時刻から成るタイムスタンプ文字列が入ります。

(例 2008/1/1 6:50pm は、“200801011850” )

<ad#n> には、A/D 変換値が入ります。

スクリプト中の set\_permanent\_data() 部分を set\_oracle\_data() に変更すると、保管先のデータベースを Firebird から、Oracle DBMS にすることができます。

スクリプト中の script\_exec(“XBEE\_ERROR\_MAIL\_SEND”, “ErrorDeviceName”, device\_name) 部分で、エラー発生時にメール送信を行います。メール送信機能を使用しない場合はコメントアウトしてください。

エラー発生は、xbee\_force\_sample() 実行に失敗した場合です。これは具体的には、 リモート XBee デバイスに対して、DeviceServer のCOM に接続した XBee デバイスからリモートATコマンド “IS” を送信して、正常なステータスのフレームをリトライ回数(デフォルト2回)以内に受信できなかった場合です。

ファイル名(XBEE\_ACQUISITION.lua) で DeviceServerのスクリプトフォルダに保管します。

```
file_id = "XBEE_ACQUISITION"
log_msg("start..", file_id)

-----

-- XBeeデバイスの A/D 変換読み込んで、データベースに格納する。
-- XBEE_DEV_LIST スクリプトで取得した XBee デバイスを対象としてデータの取得を行う。
-- 取り込んだデータは、下記のキー名で DeviceServer のデータベースに格納する。
--
-- 詳細説明はアプリケーションノート (APNOTE07) を参照の事
--
-- データベースキー名 : ACQADC-<DeviceName>-<YYYYMMDDHHMM>
--
--   <DeviceName>   デバイス名
--   <YYYYMMDDHHMM> 日付と時間を表したタイムスタンプ値。2001/1/31 10:20:30に
--                   スクリプトを起動した場合は '200101311020' になる
--
-- データベース値   : <ADC0>, <ADC1>, <ADC2>, <ADC3>, <ADC4>, <ADC5>
--                   ADCxx の値は 0 から 1023 までの整数が入る
--                   ADCxx のサンプリング値が無い場合は、“NA” 文字列が入る
--
-----

local now = os.date "*t";
```

```

local timestamp =
string.format("%4.4d%2.2d%2.2d%2.2d", now["year"], now["month"], now["day"], now["hour"], now["min"]);

local stat, device_list = script_exec2("XBEE_DEV_LIST", "", "");
if not stat then error() end;

for idx, device_name in pairs(device_list) do
    local key_name;
    local value_str = "";
    local sample_stat, adc, dio = xbee_force_sample(device_name);
    if sample_stat then
        key_name = "ACQADC-" .. device_name .. "-" .. timestamp;
        for bit = 0, 5, 1 do
            if adc[tostring(bit)] then
                value_str = value_str .. adc[tostring(bit)]
            else
                value_str = value_str .. "NA"
            end;
            if (bit ~= 5) then value_str = value_str .. "," end;
        end;
        if not set_permanent_data(key_name, value_str) then error() end;
        --if not set_oracle_data(key_name, value_str) then error() end;
        log_msg("data stored: " .. key_name .. " <- " .. value_str, file_id);
    else
        if not script_exec("XBEE_ERROR_MAIL_SEND", "ErrorDeviceName", device_name)
            then error()
        end;
    end;
end;
end;

```

### 5.5.3 XBEE\_ERROR\_MAIL\_SEND スクリプト作成

リモート XBee からのデータ収集時に、エラーが発生した場合にメールで知らせるためのスクリプトを作成します。  
スクリプト中の `error_mail_addr` にエラーメールの宛先を設定してください。

ファイル名 (XBEE\_ERROR\_MAIL\_SEND.lua) で DeviceServer のスクリプトフォルダに保管します。

```

file_id = "XBEE_ERROR_MAIL_SEND";
--[[
*****

```

```

XBee デバイスでエラーが発生した時に、メールを送信する。

エラーが発生した XBee デバイス名を、スクリプトパラメータ "ErrorDeviceName"
で受け取る。

error_mail_addr に送信先メールアドレスを指定する。

*****

]]

error_mail_addr = "エラーメール宛先 <your-mail-addr@mail-domain.com>";

log_msg("start..", file_id);
local error_device_name = g_params["ErrorDeviceName"];
if error_device_name then
    local body = {};
    table.insert(body, "XBee デバイスでエラーが発生しました");
    table.insert(body, "デバイス名:" .. error_device_name);
    stat = mail_send(error_mail_addr, "", "XBee デバイスでエラーが発生", unpack(body));
    if not stat then error() end;
else
    log_msg("ErrorDeviceName パラメータが指定されていません", file_id);
end;

```

## 6 スケジュール実行用の DeviceServer ユーザー登録

定期的にスクリプト実行を行うときに使用する ScriptExecCmd.exe プログラムで使用するユーザーアカウントを DeviceServer に作成します。ユーザーのアプリケーション許可フラグには、デフォルトの "AllowLogin" を指定します。（他のオプションを追加指定しても構いません）

ここでは、ユーザー名: "data" パスワード: "data" として作成します。別のユーザー名やパスワードにしても構いません。その場合は、後で説明する ScriptExecCmd.ini ファイルに指定する内容も変更してください。

DeviceServer のユーザーアカウント作成の詳しい説明は "DeviceServer ユーザーマニュアル" 中の "動作確認と簡単な使い方" の章と、"クライアントソフトウェア" の章を参照してください。

---

### 注意

---

スクリプト実行をスケジュールを使用して行う場合は、スケジュール毎に別ユーザーアカウントを作成することをお

---

---

勤めします。これは、同一ユーザーが完全に同一のタイミングでログインする場合（複数のスケジュール実行が同一時刻で実行される場合に発生します）には、ログイン処理中に、ユーザー情報がロック中の為にエラーになる場合があるためです。

---

## 7 Windows タスクスケジューラの設定

---

定期的に決められた時刻でスクリプトを実行するために、WindowsXP もしくは、Windows2003 Server のタスクスケジューラとスケジュール用コマンド“schtasks”を使用します。また、コマンドプロンプトからスクリプトを実行するためのプログラムは、ABS-9000 DeviceServer インストール時に保管されている ScriptExecCmd.exe を使用します。

XBEE\_ACQUISITION スクリプトを実行する、タスクスケジューラのジョブ登録は、DeviceServer の動作している PC で行います。ScriptExecCmd.exe の設定ファイルの HostName タグに、“localhost”以外を記述して、スケジュール実行を行うPC と DeviceServer の動作するPC を分けることも可能です。ただしこの場合 XBEE\_ACQUISITION.lua 内でタイムスタンプを取得しているため、スケジューラで起動された時刻とDeviceServer の動作するPC の時刻が一致していないと、集計時にデータ取得を行うためのキー名指定（タイムスタンプとデバイス名を元に指定する）が困難になります。

### 7.1 ScriptExecCmd.exe プログラムの設定

---

ScriptExecCmd.exe プログラムを “C:¥Tools¥Bin” に配置してください。また、同一フォルダに ScriptExecCmd.ini ファイルを作成して、スクリプト名とユーザー名、パスワードを予め指定しておいてください。

“C:¥Tools¥Bin” フォルダは、フォルダ名に空白文字や日本語を含まないようにするために作成しています。このフォルダ以外の場所に ScriptExecCmd.exe と ScriptExecCmd.ini ファイルを配置する場合は、schtasks コマンドを実行する時のパス名指定部分を適宜変更してください。

ScriptExecCmd.ini ファイルは下記の内容で作成します。ScriptExecCmd.exe を実行したときに、iniファイルが見つからない場合は、自動的にiniファイルがデフォルト値で作成されます。

ScriptName には XBEE\_ACQUISITION を指定します。HostName には “localhost”を指定します。

UserName, Password は、“6. スケジュール実行用の DeviceServerユーザー登録” で設定した値を指定します。

```
[ScriptExecCmd]
ScriptName=XBEE_ACQUISITION
HostName=localhost
UserName=data
Password=data
KeyList=
ValList=
```

ScriptExecCmd.exe プログラムと ScriptExecCmd.ini ファイルの設定方法については “DeviceServer ユーザーマ

ニュアル”中の”その他のプログラム”の章を参照してください。

## 7.2 タスクスケジューラへの JOB 登録

---

ここでは、5 分毎にデータを取得してデータベースに格納する例で説明します。

DeviceServer の動作する PC で、コマンドプロンプトを起動して、下記のコマンドを実行してください。必ず、システム管理者権限を持った Windows アカウントでログインしてから実行してください。

下記のコマンドは、Windows のシステム管理者特権のユーザー名が “Administrator” でパスワードが “xxxxx” の場合の例になります。2 行で表示されていますが、実際には 1 行で入力してください。

```
schtasks /create /tn ACQUISITION_TASK /sc minute /mo 5 /st 00:00:00 /ru Administrator /rp xxxxx  
/tr C:\tools\bin\scriptexecmd.exe
```

Windows タスクスケジューラと “schtasks コマンド” についての詳しい説明はマイクロソフト社のドキュメントを参照してください。

## 8 エクセルからデータ取得・集計

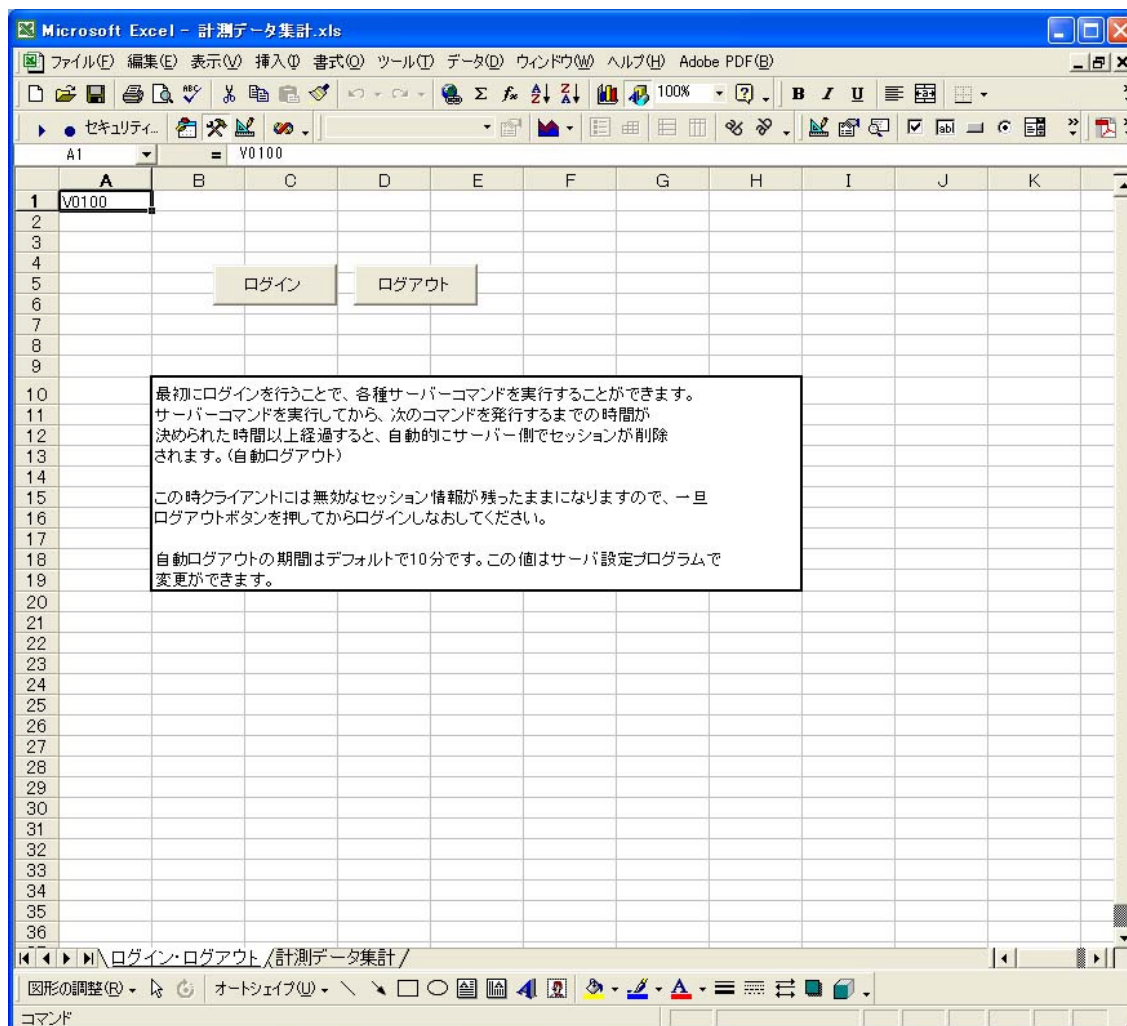
---

クライアント PC のエクセルから、データベースに保存された計測データを取り込んで集計を行います。

データの取得とデータベース保管は、スケジューラによってバックグラウンドで実行されていますので、クライアントからの集計は任意のタイミングで行えます。

以降で説明している内容は、添付のエクセルファイル (計測データ集計. xls) のワークシートを実行したものです。全てのマクロ (VBA スクリプト) とワークシートシートはエクセルファイル (計測データ集計. xls) 内に記述されていますので、詳細はエクセルをデザインモードにして、”Visual Basic Editor” で開いてください。以下に、データ取り込み用のワークシートのマクロ実行画面と、マクロ (VBA スクリプト) の動作について説明します。





(計測データ集計.xls を開いた画面。起動時にエクセルのマクロを有効にして下さい)

クライアントPC でエクセルを実行する場合には、予め XASDLCMD.DLL をクライアントPC のシステムフォルダにコピーしておく必要があります。エクセルを実行するPC が DeviceServer の動作しているPC と同一の場合は DLL のコピーは必要ありません。詳細は“DeviceServerユーザーマニュアル”の“インストール”章の“ユーザーアプリケーションを利用する場合”の項目を参照してください。

## 8.1 DeviceServer にログイン

計測データ集計.xls をエクセルで開いて、“ログイン・ログアウト”ワークシートを選択します。

ログインボタンを押して、DeviceServer にログインします。この時に指定するユーザーは、DeviceServer に登録済みのユーザーを指定してください。スケジューラでスクリプト実行用に作成したユーザー以外も指定することができます。ホスト名には、DeviceServer の動作しているPC のホスト名を入力します。



ワークシート内のログインボタンを押した時に実行されるマクロ (VBA) 部分は以下の様に記述されています。ログインに成功するとLastSessionToken にセッショントークン文字列が入っています。未ログインの場合は空文字列が入っています。

```
Private Sub LoginBtn_Click()
    If LastSessionToken <> "" Then
        MsgBox ("現在ログイン中です。一旦ログアウトしてからやり直してください")
        Exit Sub
    End If
    LoginForm.LoginPasswordEdit.Text = ""
    LoginForm.Show
End Sub
```

ログインフォームの”OK” ボタンを押した時に実行されるマクロ (VBA) 部分は以下の様に記述されています。

```
Private Sub CommandButton1_Click()
    If Not LoginUser(ServerEdit.Text, LoginNameEdit.Text, LoginPasswordEdit.Text) Then
        MsgBox ("ログインに失敗しました")
    End If
    LoginForm.Hide
End Sub
```

標準モジュールに定義された LoginUser () は以下の様になっています。

```
' この関数を直接コールしないで、代わりに LoginUser () を使用してください
Public Declare Function SX_LoginUser Lib "XASDLCMD.dll" (ByVal Host As String, ByVal Port As Integer, ByVal
UserName As String, ByVal Password As String, ByVal Session As String) As Integer

Public LastSessionToken As String
Public LastLoginName As String
Public LastLoginHost As String
```

```

Public Function LoginUser (ByVal Host As String, ByVal UserName As String, ByVal Password As String) As Boolean

    Dim NewSession As String * 128

    If SX_LoginUser(Host, DefPort, UserName, Password, NewSession) = 0 Then

        LastLoginHost = Host

        LastLoginName = UserName

        LastSessionToken = NewSession

        LoginUser = True

    Else

        LoginUser = False

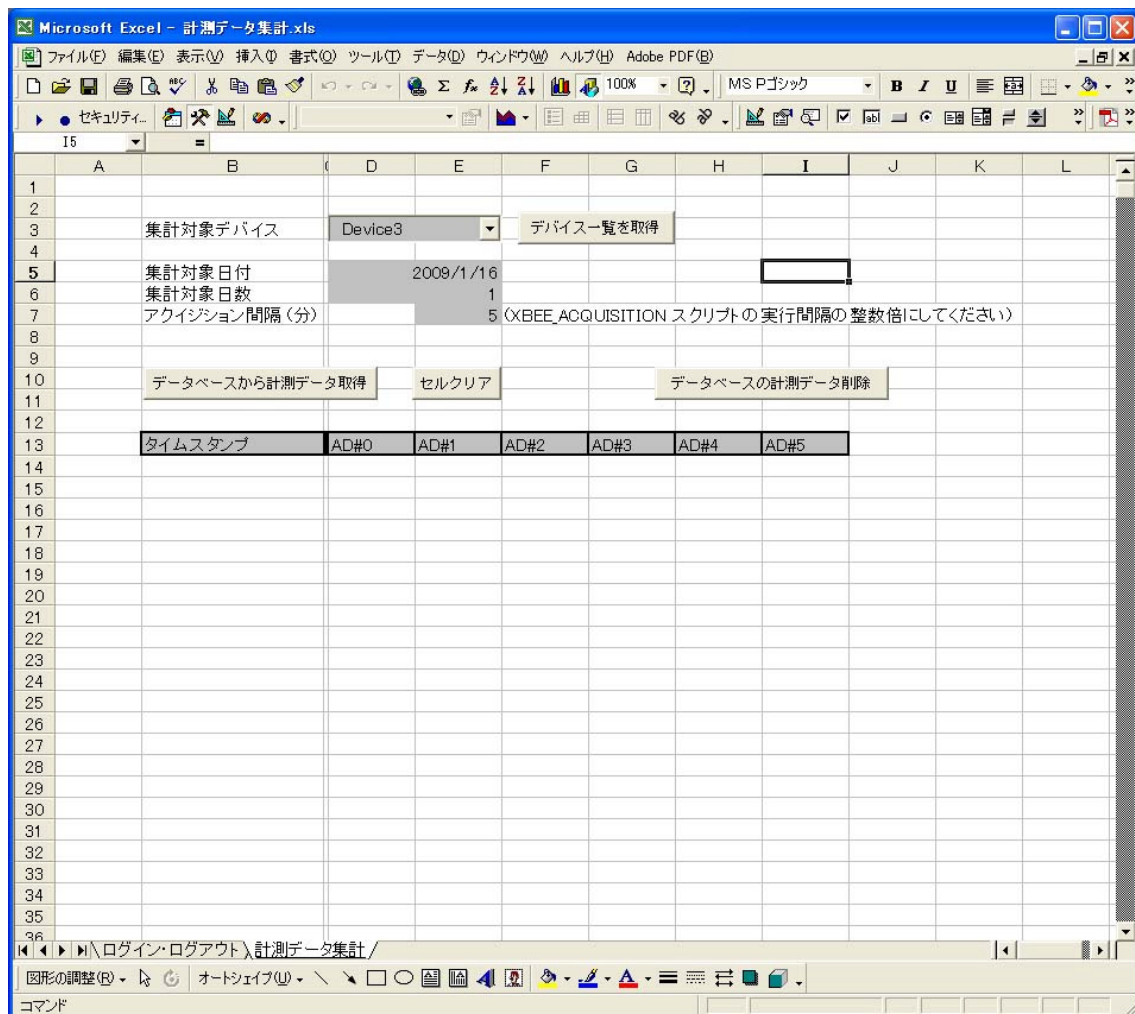
    End If

End Function

```

## 8.2 データ取り込み範囲指定

ログインに成功したら、“計測データ集計” ワークシートを選択します。



最初にデータベースから計測データを取得する条件を設定します。デバイス名と日付範囲、計測データ間隔を指定します。デバイス名はコンボボックスになっていますので、直接 XBee デバイスの名前(NodeIdentifier)を指定することができます。”デバイス一覧を取得” ボタンを押すと、“XBEE\_DEV\_LIST スクリプト作成” で作成したスクリプトを DeviceServerで実行して、コンボボックスのリストに最新のデバイス名をロードすることができます。

集計対象日付には、集計対象の開始日を指定します。集計対象日数には開始日から何日分のデータを取得するかを日数で指定します。アキュイジション間隔は、集計対象日付の午前 00:00 から何分単位にデータを取得するかを指定します。ここで指定する分間隔は、“タスクスケジューラへのJOB登録” で指定した値の整数倍を指定してください。

|               |           |                                          |
|---------------|-----------|------------------------------------------|
| 集計対象デバイス      | Device3   | デバイス一覧を取得                                |
| 集計対象日付        | 2009/1/16 |                                          |
| 集計対象日数        | 3         |                                          |
| アキュイジション間隔(分) | 30        | (XBEE_ACQUISITION スクリプトの実行間隔の整数倍にしてください) |

(XBeeデバイス名 “Device3” の 2009/1/16 日から3日分のデータを30 分間隔で取得する例)

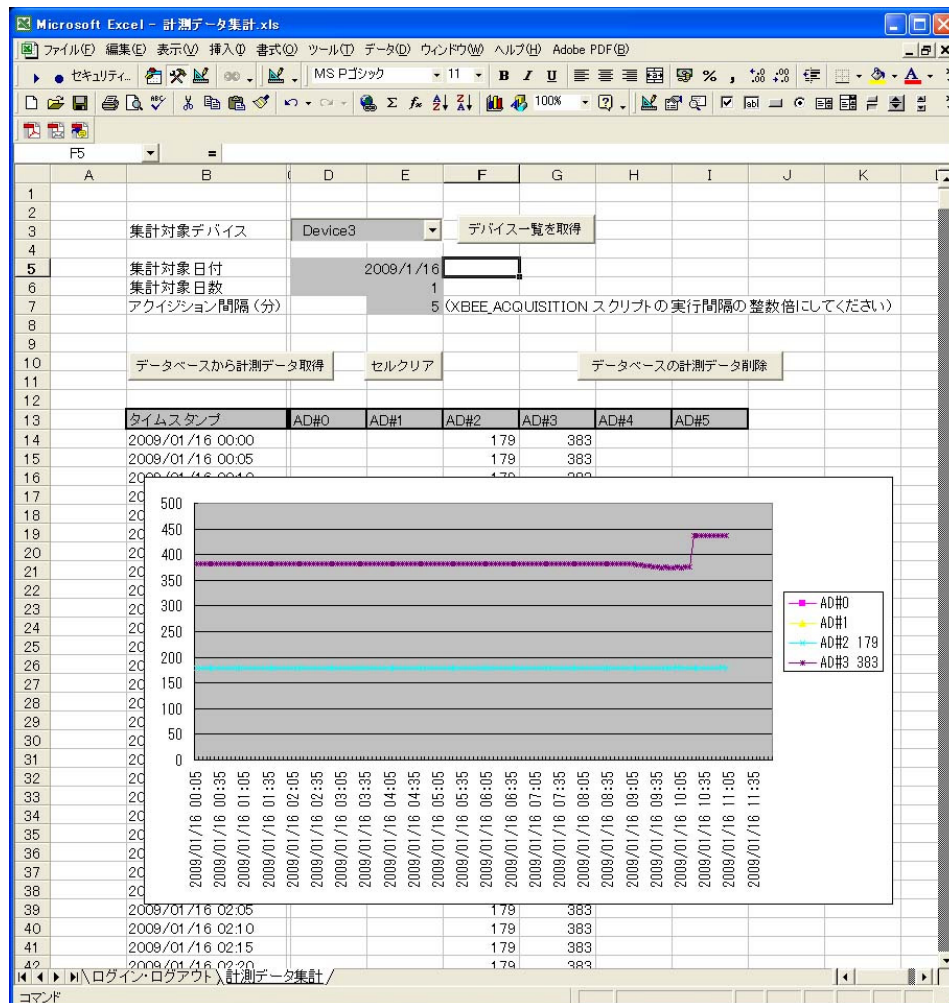
### 8.3 データ取得と集計

集計対象条件を入力した後に、“データベースから計測データ取得” ボタンを押します。集計対象に指定された計測データを DeviceServer のデータベースから取得して、ワークシートのセルに入力されます。

| タイムスタンプ          | AD#0 | AD#1 | AD#2 | AD#3 | AD#4 | AD#5 |
|------------------|------|------|------|------|------|------|
| 2009/01/16 00:00 |      |      | 223  | 1021 |      |      |
| 2009/01/16 00:30 |      |      | 223  | 1021 |      |      |
| 2009/01/16 01:00 |      |      | 224  | 1021 |      |      |
| 2009/01/16 01:30 |      |      | 224  | 1021 |      |      |
| 2009/01/16 02:00 |      |      | 224  | 1021 |      |      |
| 2009/01/16 02:30 |      |      | 224  | 1021 |      |      |
| 2009/01/16 03:00 |      |      | 224  | 1021 |      |      |
| 2009/01/16 03:30 |      |      | 224  | 1021 |      |      |
| 2009/01/16 04:00 |      |      | 224  | 1021 |      |      |
| 2009/01/16 04:30 |      |      | 224  | 1021 |      |      |
| 2009/01/16 05:00 |      |      | 224  | 1021 |      |      |
| 2009/01/16 05:30 |      |      | 224  | 1021 |      |      |
| 2009/01/16 06:00 |      |      | 224  | 1021 |      |      |
| 2009/01/16 06:30 |      |      | 224  | 1021 |      |      |
| 2009/01/16 07:00 |      |      | 224  | 1021 |      |      |
| 2009/01/16 07:30 |      |      | 224  | 1021 |      |      |
| 2009/01/16 08:00 |      |      | 224  | 1021 |      |      |
| 2009/01/16 08:30 |      |      | 224  | 1021 |      |      |
| 2009/01/16 09:00 |      |      | 224  | 1021 |      |      |
| 2009/01/16 09:30 |      |      | 224  | 1021 |      |      |
| 2009/01/16 10:00 |      |      | 224  | 1021 |      |      |
| 2009/01/16 10:30 |      |      | 224  | 1021 |      |      |
| 2009/01/16 11:00 |      |      | 224  | 1021 |      |      |

計測データが見つからない場合は、セルは空になります。A/D 変換値はそれぞれのチャンネル毎にセルに値が入ります。計測データをセルに取り込んだ後は、エクセルで自由に加工して集計を行うことができます。

計測データの A/D 変換値をグラフ化した例は以下のようになります。



”データベースから計測データ取得” ボタンを押した時に実行されるマクロ (VBA) 部分は、以下の様に記述されています。マクロ (VBA) 中の `SX_get_permanent_data()` を `SX_get_oracle_data()` に変更すると、計測データの取得先のデータベースを Firebird から、Oracle DBMS にすることができます。

`CsvToList()` 関数等、詳しい内容はエクセルファイル (計測データ集計.xls) の内容を参照してください。

```
Private Sub DoSummarizeBtn_Click()
    Call fetch_data
End Sub

Public Sub fetch_data()
    Dim TimeStamp As Date
```

```

Dim ADKeyName As String
Dim DINKeyName As String
Dim i As Integer
Dim cntr As Integer
Dim MaxSample As Integer
Dim SharedValue As String * 256
Dim ChackStr As String
Dim TmpVal As Integer

TimeStamp = CDate(ActiveSheet.Range("D5").Value) + #12:00:00 AM#
MaxSample = 24 * (ActiveSheet.Range("D6").Value) * (60 / ActiveSheet.Range("E7").Value) ' アクイジション間隔が 60 で割り切れないと正しく動作しない
For i = 1 To MaxSample
    ' タイムスタンプを記入
    ActiveSheet.Cells(start_line + i - 1, start_column).Value = Format(TimeStamp, "yyyy/mm/dd hh:nn")
    ' A/D データ取得
    ADKeyName = "ACQADC-" & DeviceListCbx.Text & "-" & Format(TimeStamp, "yyyymmddhhnn")
    If SX_get_permanent_data>LastSessionToken, LastLoginHost, DefPort, ADKeyName, SharedValue) <> 0 Then
        ' If SX_get_oracle_data>LastSessionToken, LastLoginHost, DefPort, ADKeyName, SharedValue) <> 0 Then
            MsgBox (SVR_ERR_MSG)
            Exit Sub
        Else
            CheckStr = Trim(Left(SharedValue, InStr(SharedValue, vbNullChar) - 1))
            If CheckStr <> "" Then
                ' A/D データリスト(CSV)の各フィールドをセルに代入
                Call CsvToList(SharedValue)
                For cntr = 0 To (result_csv_clmcnt - 1)
                    If result_csv(cntr) <> "NA" Then
                        ActiveSheet.Cells(start_line + i - 1, start_column + 2 + cntr) = result_csv(cntr)
                    End If
                Next cntr
            End If
        End If

        TimeStamp = DateAdd("n", ActiveSheet.Range("E7").Value, TimeStamp)
    Next i

    MsgBox ("データ取得 完了しました")
End Sub

```



## 9 このドキュメントについて

---

### 9.1 著作権および登録商標

---

Copyright© 2009 オールブルーシステム

このドキュメントの権利はすべてオールブルーシステムにあります。無断でこのドキュメントの一部を複製、もしくは再利用することを禁じます。

Windows、Visual Basic および Excel は米国Microsoft Corporationの米国およびその他の国における登録商標または商標です。ここではExcel® をエクセル、Visual Basic® for Applications をVBAと表記する場合があります。

### 9.2 連絡先

---

オールブルーシステム (All Blue System)

ウェブページ <http://www.allbluesystem.com>

メール [contact@allbluesystem.com](mailto:contact@allbluesystem.com)

### 9.3 このドキュメントの使用について

---

このドキュメントは、ABS-9000 DeviceServer の一般的な使用方法と応用例について解説してあります。お客様の個別の問題について、このドキュメントに記載された内容を実際のシステムに利用するときには、ここに記載されている以外にも考慮する事柄がありますので、ご注意ください。特に安全性やセキュリティ、長期間にわたる運用を想定してシステムを構築する必要があります。

オールブルーシステムでは ABS-9000 DeviceServer の使用や、このドキュメントに記載された内容を使用することによって、お客様及び第三者に損害を与えないことを保証しません。ABS-9000 DeviceServer を使用したシステムを構築するときには、お客様の責任の下で、システムの構築と運用が行われるものとします。

## 10 更新履歴

---

REV A. 1.1 2010/3/31

XBee デバイスを DeviceServer 全てのライセンスで扱える様に記述を変更

REV A. 1.0 2009/1/20

初版作成