

## [APNOTE06]

定期的にセンサー計測値を DB に保管、クライアント PC の  
エクセルから集計

---

ABS-9000 DeviceServer

APNOTE06 Rev A.1.2

2008/11/03



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

---

アラームデバイス (SigSensor, NetUI0) の A/D 変換値や Digital 入力ポートの値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。

DeviceServer に接続された、複数のアラームデバイス (SigSensor, NetUI0) の A/D 変換入力と Digital 入力の値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。

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て、集計やグラフの作成等を行うことができます。

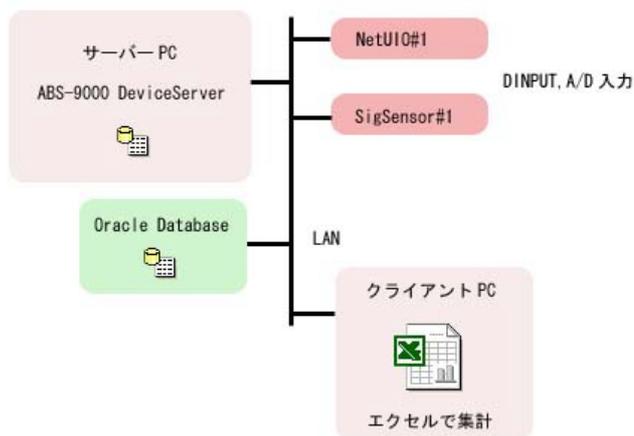
## 2 必要な機材・リソース

---

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	スタンダードライセンスもしくはエンハンスライセンスが必要になります。データを保管するためのデータベースに Oracle 10gを使用する場合は、エンハンスライセンスが必要になります。
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。
SigSensor, NetUI0デバイス	DeviceServer に接続された全てのSigSensor, NetUI0デバイスがデータ収集対象となります。(SENSOR_ACQUISITION.lua スクリプトを変更することで、データ収集の機能をカスタマイズ可能です)
Oracle10g サーバー (オプション) 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。	DeviceServer の動作するPC もしくは 別 PC にOracle10g サーバーと、DeviceServer の動作するPCに Oracleクライアントが必要です。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、アラームデバイスの A/D 変換と Digital 入力値を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 SENSOR\_ACQUISITION スクリプトを実行します。
- SENSOR\_ACQUISITION スクリプトは、アラームデバイスを検索して、A/D 変換値の取り込みと、Digital 入力値の取り込みを行います。その後、タイムスタンプ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 サーバー設定

計測データ保管用のデータベースに DeviceServer に組み込まれた Firebird を使用する場合は、特にサーバー設定プログラムで設定する項目はありません。

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	
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)

## 5.2 デバイス設定

SigSensor デバイス、NetUI0デバイスを LAN 上に設置します。複数台設置して同時に計測データを保存することも可能です。

DeviceServer のアラーム管理プログラムで、設置したアラームデバイスをDeviceServer に登録します。

また、アラーム管理プログラムで、アラームデバイス (SigSensorまたは NetUI0) の下記の項目の詳細設定を行います。

アラーム管理プログラムで設定するデバイスの詳細機能設定	
設定が必要な項目	設定内容
IPアドレス	デバイスのIP アドレスを設定する
IPネットワークマスク	デバイスを設置した LAN の環境に合わせる
デフォルトゲートウェイアドレス	デバイスを設置した LAN の環境に合わせる
DINPUT値を有効にする	チェックを付ける
A/D 入力を有効にする	チェックを付ける
A/D サンプリングレート	10msを設定 センサー値を取り込むスクリプトは、未読み込みで直近の A/D 変換値バッファの、平均値を読み込む様になっています。スクリプト実行の間隔(スケジュール間隔 5分) と比べて、ここで設定したサンプリングレートと A/D バッファの個数(1024) を掛けた値が少なくなるようにしてください。

デバイス登録と詳細機能設定の方法については“DeviceServerユーザーマニュアル”を参照してください。また、SigSensor、NetUI0 デバイスの詳しい使用方法については、“SigSensor\_NetUI0ユーザーマニュアル”を参照してください。

## 5.3 スクリプト設定

### 5.3.1 SENSOR\_LIST スクリプト作成

エクセルからデバイスリストを取得するためのスクリプトを作成します。エクセル中の VBA からこのスクリプトを実行します。DeviceServer に接続中の有効なデバイスを検索して、得られたデバイスリストをエクセルに取り込みます。ファイル名(SENSOR\_LIST.lua) で DeviceServerのスクリプトフォルダに保管します。

-----  
-- アラームデバイスのリストをリターン値に返す  
-----

```
local stat, name, type, sysalert, neterr = alarm_all_list()
if not stat then error() end
for key, val in ipairs(name) do
  if ((type[key] == "SIGSENSOR") or (type[key] == "NETUI0")) and (not neterr[key]) then
    script_result(g_taskid, "Device#" .. tostring(key), val)
  end
end
end
```

### 5.3.2 SENSOR\_ACQUISITION スクリプト作成

アラームデバイスに接続された I/O 装置から A/D変換入力と、Digital 入力の値を読み込んで、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	ACQ-<DeviceName>-AD-<YYYYMMDDHHMM>	<ad#0>, <ad#1>, <ad#2>, <ad#3>
Digital 入力	ACQ-<DeviceName>-DIN-<YYYYMMDDHHMM>	<dinput>

<DeviceName> には、アラームデバイス名が入ります。

<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。

<dinput> には、Digital 入力ポートの値が入ります。

A/D 変換値は、"5.2 デバイス設定" で設定したサンプリングレートで変換バッファに格納された、A/D 変換値の平均値を示します。例えば、スクリプトを5分毎に起動して、A/D サンプリングレートが10ms の場合は、5分毎に直近の約10秒間の1024 個のサンプリング値の平均を取り出すことになります。

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。

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

```
file_id = "SENSOR_ACQUISITION"
```

```
-----  
-- BEGIN SCRIPT --  
-----
```

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

```
-----  
-- アラームデバイスの A/D 変換値と、DINPUT 値を読み込んで、  
-- データベースに格納する。  
-- 全ての SIGSENSOR, NETUI0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 (APNOTE05) を参照の事  
--
```

```
-- キー名 : ACQ-<DeviceName>-<Type>-<YYYYMMDDHHMM>
```

```
--<DeviceName> デバイス名
```

```
--<Type> DINPUT 値は、'DIN', A/D 変換値は 'AD' が入る
```

```
--<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  
--
```

```
-- 一つのデバイスに、DINPUT, A/D 変換機能の両方がある場合は、1 デバイスあたり、  
-- <Type>が 'DIN', 'AD' を持った2つのデータが格納される。  
-----
```

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

```
local timestamp =
```

```
string.format("%4.4d%2.2d%2.2d%2.2d%2.2d", now["year"], now["month"], now["day"], now["hour"], now["min"])
```

```
local stat, buff1, bstat1, buff2, bstat2, buff3, bstat3, buff4, bstat4, din
```

```
local key_name, data_string
```

```
local stat, name, type, sysalert, neterr = alarm_all_list()
```

```
if not stat then error() end
```

```
for key, val in ipairs(name) do
```

```
if ((type[key] == "SIGSENSOR") or (type[key] == "NETUI0")) and (not neterr[key]) then
```

```
-----  
-- A/D データ取得。デバイスの格納バッファにある
```

```
-- 直近の平均値データを使用する
```

```

-----
stat, buff1, bstat1, buff2, bstat2, buff3, bstat3, buff4, bstat4 = alarm_adbuff_get(val)
if not stat then error() end
key_name = "ACQ-" .. val .. "-AD-" .. timestamp
data_string =
string.format("%d,%d,%d,%d", bstat1["MeanValue"], bstat2["MeanValue"], bstat3["MeanValue"], bstat4["MeanValue"])

if not set_permanent_data(key_name, data_string) then error() end
--if not set_oracle_data(key_name, data_string) then error() end
log_msg("data stored: " .. key_name .. " <- " .. data_string, file_id)

-----

-- DINPUT 取得
-----

stat, din = alarm_din_get(val)
if not stat then error() end
key_name = "ACQ-" .. val .. "-DIN-" .. timestamp
data_string = string.format("%d", din)

if not set_permanent_data(key_name, data_string) then error() end
--if not set_oracle_data(key_name, data_string) then error() end
log_msg("data stored: " .. key_name .. " <- " .. data_string, file_id)

end
end

-----

-- END SCRIPT --
-----

```

---

### 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。

---

## 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ラ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勧めします。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。

SENSOR\_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 SENSOR\_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 SENSOR\_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=SENSOR_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%scriptexeccmd.exe
```

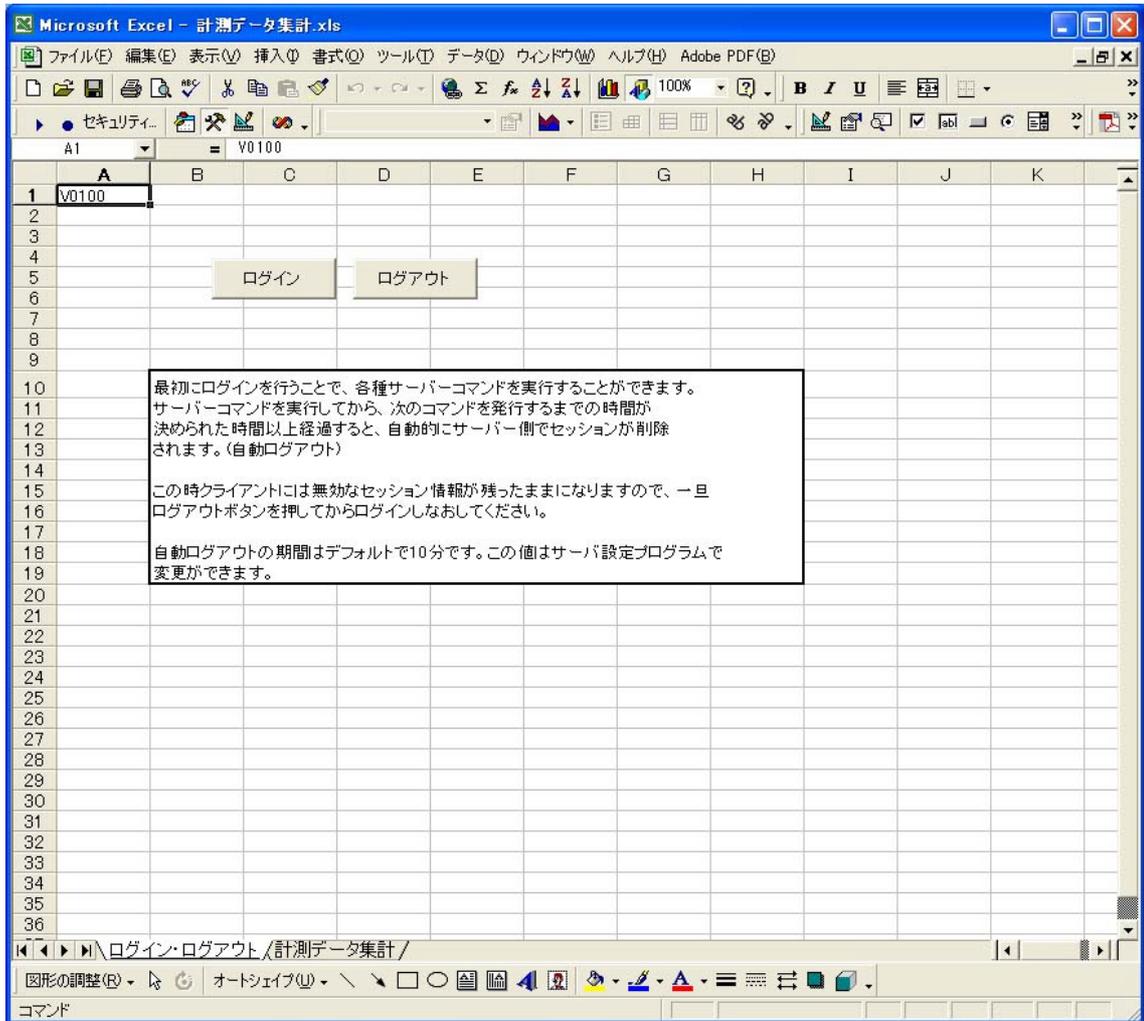
Windows タスクスケジューラと “schtasksコマンド” についての詳しい説明はマイクロソフト社のドキュメントを参照してください。

## 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。ここでは、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
```

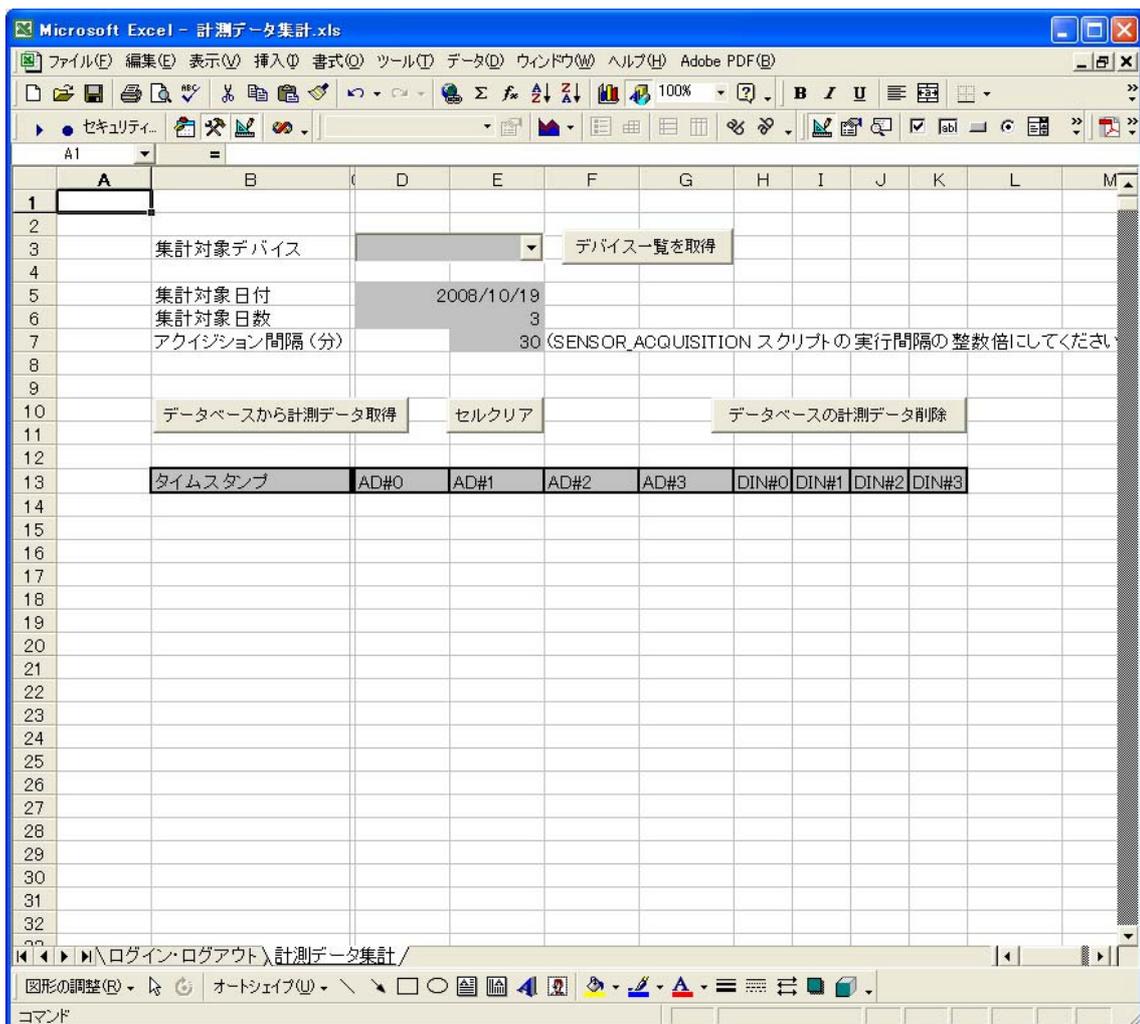
```

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、直接アラームデバイスの名前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、“5.3.1 SENSOR\_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、“7.2 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	SigSensor	デバイス一覧を取得
集計対象日付	2008/10/19	
集計対象日数	3	
アキュイジション間隔(分)	30	(SENSOR_ACQUISITION

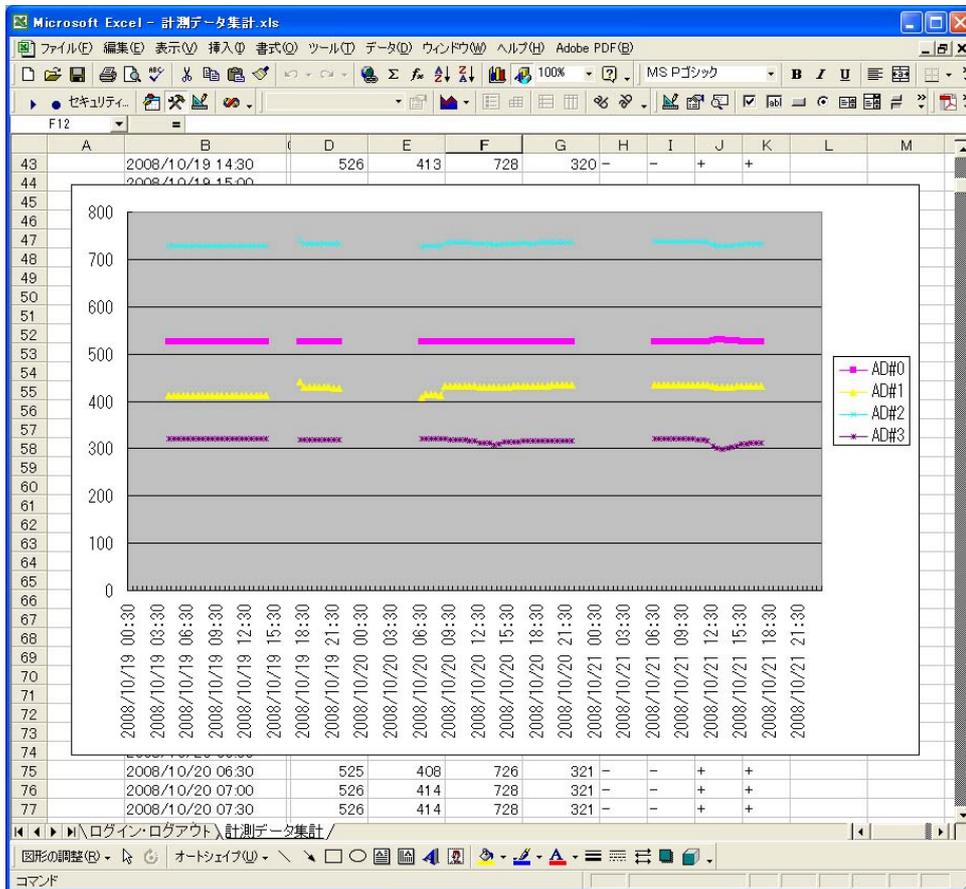
(アラームデバイス名 “SigSensor” の 2008/10/19 日から3日分のデータを30 分間隔で取得する例)

### 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	DIN#0	DIN#1	DIN#2	DIN#3
2008/10/19 00:00								
2008/10/19 00:30								
2008/10/19 01:00								
2008/10/19 01:30								
2008/10/19 02:00								
2008/10/19 02:30								
2008/10/19 03:00								
2008/10/19 03:30								
2008/10/19 04:00								
2008/10/19 04:30	526	413	728	321	-	-	+	+
2008/10/19 05:00	526	413	728	321	-	-	+	+
2008/10/19 05:30	526	413	728	321	-	-	+	+
2008/10/19 06:00	526	413	728	321	-	-	+	+
2008/10/19 06:30	526	413	728	321	-	-	+	+
2008/10/19 07:00	526	413	728	321	-	-	+	+
2008/10/19 07:30	526	413	728	321	-	-	+	+
2008/10/19 08:00	526	413	728	321	-	-	+	+
2008/10/19 08:30	526	413	728	321	-	-	+	+
2008/10/19 09:00	526	413	728	321	-	-	+	+
2008/10/19 09:30	526	413	728	321	-	-	+	+
2008/10/19 10:00	526	413	728	321	-	-	+	+
2008/10/19 10:30	526	413	728	320	-	-	+	+
2008/10/19 11:00	526	413	728	320	-	-	+	+

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。Digital 入力ポート(合計 4bit) は、ビット毎にセルに値が入り、“-“ が Low レベルで “+” が High レベルを示します。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 TmpVal As Integer

TimeStamp = CDate(ActiveSheet.Range("D5").Value) + #12:00:00 AM#

MaxSample = 24 * (ActiveSheet.Range("D6").Value) * (60 / ActiveSheet.Range("E7").Value) ' アクイジシヨ
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 = "ACQ-" & DeviceListCbx.Text & "-AD-" & 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
            ' 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)
                ActiveSheet.Cells(start_line + i - 1, start_column + 2 + cntr) = result_csv(cntr)
            Next cntr
        End If

        ' DINPUT データ取得
        DINKeyName = "ACQ-" & DeviceListCbx.Text & "-DIN-" & Format(TimeStamp, "yyyymmddhhnn")
        If SX_get_permanent_data(LastSessionToken, LastLoginHost, DefPort, DINKeyName, SharedValue) <> 0 Then
            ' If SX_get_oracle_data(LastSessionToken, LastLoginHost, DefPort, DINKeyName, SharedValue) <> 0 Then
                MsgBox (SVR_ERR_MSG)
                Exit Sub
            Else
                ' 取得データが空でなかったものについて処理する
                SharedValue = Left(SharedValue, InStr(SharedValue, vbNullChar) - 1)
                If Trim(SharedValue) <> "" Then
                    TmpVal = Val(SharedValue)
                    TmpVal = TmpVal Mod 16
                End If
            End If
        End If
    End If

```

```
For cntr = 3 To 0 Step -1
  If (TmpVal >= (2 ^ cntr)) Then
    ActiveSheet.Cells(start_line + i - 1, start_column + 9 - cntr) = "+"
  Else
    ActiveSheet.Cells(start_line + i - 1, start_column + 9 - cntr) = "-"
  End If
  TmpVal = TmpVal Mod (2 ^ cntr)
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© 2008 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。