

[APNOTE02]

アラームデバイスのネットワーク接続を定期監視

ABS-9000 DeviceServer

APNOTE02 Rev A.1.0

2008/10/1



オールブルーシステム (All Blue System)

ウェブページ: www.allbluesystem.com

コンタクト: contact@allbluesystem.com

1 イン트로ダクション

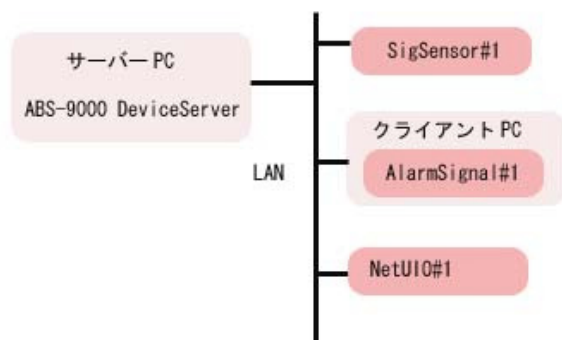
ABS-9000 DeviceServer に接続された全てのアラームデバイス (SigSensor, NetUI0, AlarmSignal) を対象に、定期的にネットワーク接続を試みて、エラーが発生しているかどうかを調べる方法を説明します。LAN ケーブルの断線やアラームデバイスの電源ダウンなどを、DeviceServerで早期に検出することができます。

アラームデバイスとの通信に失敗した場合は DeviceServer の ALARM_NETWORK_ERROR イベントが発生しますので、そのイベントハンドラにユーザーがエラー処理を組み込むことができます。

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。 |
| SigSensor, NetUI0デバイス AlarmSignal.exe プログラム | DeviceServer に接続された全てのアラームデバイスが監視対象となります。 |
| SMTP メールアカウント | エラー発生時にメールを送信するSMTPサーバーメールアカウント。 (メール送信を行わない場合は必要ありません) |

3 システム構成図



4 システム動作概要

- DeviceServer は、PERIODIC_TIMER を使用して定期的(約1分間隔)にアラームデバイスに通信を試みます。
- アラームデバイスの通信に失敗した場合は、ALARM_NETWORK_ERROR イベントが、DeviceServerで発生します。
- ALARM_NETWORK_ERRORイベントハンドラで、システムアラートデバイスとして設定されているアラームデバイスに赤ランプ点滅とエラー内容をLCD に表示します。また、予め設定したメールアドレスにエラー内容を通知し

ます。

5 設定手順

5.1 デバイス設定

DeviceServer にSigSensor, NetUI0, アームシグナルデバイスが登録済みで、LAN 上に設置されているものとします。

5.2 サーバー設定

サーバー設定プログラムで、下記の項目を設定します。

| サーバー設定プログラム | |
|-------------------------------|---|
| 設定が必要な項目 | 設定内容 |
| アラームデバイスのネットワークエラー検出時にスクリプト実行 | チェックを付ける |
| 定期的にスクリプトを実行する | チェックを付ける |
| メール機能を有効にする(*1) | チェックを付ける |
| 自分のメールアドレス(*1) | 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(*1) | 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(*1) | 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(*1) | プロバイダのSMTPサーバーの設定に合わせて下さい |

(*1 メール送信を行わない場合は必要ありません)

5.3 スクリプト・イベントハンドラ設定

5.3.1 ALARMDEVICE_TEST スクリプト作成

DeviceServer が接続されている全てのアラームデバイスの、通信テストを行うスクリプトを作成します。

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

```
file_id = "ALARMDEVICE_TEST"

-----

-- アラームデバイスが接続可能かどうかをチェックする
-- 定期的にこのスクリプトを実行することで、アラームデバイスのネットワーク異常
-- 検出することができる。エラーが発生すると ALARM_NETWORK_ERROR イベントが発生
-- する

-----

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] == "ALARMSIGNAL") and (not neterr[key]) then
    alarm_signal_get(val)
end
if (type[key] == "SIGSENSOR") and (not neterr[key]) then
    alarm_signal_get(val)
end
if (type[key] == "NETUIO") and (not neterr[key]) then
    alarm_dout_get(val)
end
end
end

```

5.3.2 PERIODIC_TIMER イベントハンドラの作成

DeviceServer から定期的（約1分毎）に発生するPERIODIC_TIMER イベントのイベントハンドラを作成します。イベントハンドラでは、先に作成したアラームデバイスの通信テストを行うスクリプト (ALARMDEVICE_TEST) をコールします。

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

```

file_id = "PERIODIC_TIMER"
-----
-- アラームデバイスの通信エラーを定期的にチェックする
-----
if not script_exec("ALARMDEVICE_TEST","", "") then error() end

```

5.3.3 NETWORK_ERROR_MAIL_SEND スクリプト作成

エラー発生時に（エラーの発生した）アラームデバイスの名前を、電子メールで送信するスクリプトを作成します。ファイル名 (NETWORK_ERROR_MAIL_SEND.lua) で DeviceServerのスクリプトフォルダに保管します。メールの宛先部分やメール本文は、適宜環境に合わせて修正してください。

```

file_id = "NETWORK_ERROR_MAIL_SEND"
error_mail_addr = "エラーメール宛先 <your_mail_address@your_mail_domain.com>"
log_msg("start..", file_id)
local error_device_name = g_params["ErrorDeviceName"]
if error_device_name then
    body = {}
    table.insert(body, "アラーム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, "", "アラームデバイスでエラーが発生", unpack(body))
    if not stat then error() end
end

```

```

else
    log_msg("ErrorDeviceName パラメータが指定されていません", file_id)
end

```

5.3.4 ALARM_NETWORK_ERROR イベントハンドラの作成

DeviceServer でアラームデバイスの通信エラーを検出した時に発生する、ALARM_NETWORK_ERROR イベントのイベントハンドラを作成します。

スクリプトでは、アラーム管理プログラムでシステムアラートデバイスとして設定されているアラームデバイスを取得します。そのアラームデバイスがシグナル出力機能のある SigSensor または AlarmSignal の場合には、赤ランプ点滅とブザー出力を行います。また SigSensor の場合は、エラー内容をLCD に表示します。

同時に予めスクリプト中に記述したメールアドレスに、エラー内容を電子メールで通知します。これは、エラー発生をユーザーに知らせるためのシグナル出力機能を持ったアラームデバイスが無い場合や、エラー発生をユーザーに知らせるためのデバイス自身に、通信エラーが発生する可能性があるためです。システム管理者にエラー発生を知らせるための方法を、アラームデバイス以外に別途用意しておくことが必要な場合に使用します。

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

```

file_id = "ALARM_NETWORK_ERROR"
log_msg("start..", file_id)
-----
-- システムアラートとしてフラグが設定されたデバイスに、エラーが発生したことを
-- Red ランプと Beep1 ブザー、LCD メッセージで知らせる
-- エラー発生を通知する対象のデバイスで再びエラーが発生した場合には
-- LCD の内容は最後に検出したエラーの内容で上書きされる
-----

local stat, name, type = alarm_sysalert_list()
if not stat then error() end
for key, val in ipairs(name) do
    if (type[key] == "SIGSENSOR") or (type[key] == "ALARMSIGNAL") then
        if (type[key] == "SIGSENSOR") then
            stat = alarm_signal_message(val, "**DEVICE ERROR**" .. g_params["DeviceName"])
            if not stat then error() end
        end
        stat = alarm_signal_set(val, "BlinkingRed", true)
        if not stat then error() end
        stat = alarm_signal_set(val, "Beep1", true)
        if not stat then error() end
    end
end
end

```

```
end
```

```
-----  
-- メールでエラーが発生したデバイスの内容を通知する  
-----
```

```
if not script_exec("NETWORK_ERROR_MAIL_SEND", "ErrorDeviceName", g_params["DeviceName"]) then error() end
```

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 備考

ここで設定したスクリプトや設定内容は、他のDeviceServer のシステムを構築するときに、同時に組み込んで利用できます。これによって、アラームデバイスの通信監視機能を追加することができます。

7 このドキュメントについて

7.1 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。

7.2 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

7.3 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。