BNCS Alarm System

**Atos IT Services**

# Alarm System

# BNCS Driver Status Acquisition Module

alm_acq_bncs_drv_stat.dll

Written by: Richard Kerry

BNCS

# Contents

BNCS

# 1 Alarm System – BNCS Driver Status Acquisition Module

## 1.1 Overview

This module (alm_acq_bncs_drv_stat.dll) is part of the suite of modules that form the BNCS Alarm System.

The module gathers input information for the alarm system from CSI regarding the status of BNCS device drivers.

## 1.2 Description

The module registers with CSI. It reads CSI's configuration to find the driver status infodriver number, and then polls the slots as required to monitor the Tx/Rx status of the various BNCS devices' drivers. The status values are presented to the logic module, which is part of the host application.

# 2 Configuration procedure

Use the configuration editor. See the configuration document for details.

## 2.1 Starting up

The module is started from the host application AlarmControl.exe. It loads its configuration, registers with CSI for the source device it is interested in and polls the appropriate ranges of slots. Once it has polled all the inputs it re-polls any from which it hasn't received a response from, this is repeated at increasing time intervals until the maximum time between polls is about 20 minutes. There can be more than one instance of the BNCS Driver Status Acquisition module in an Alarm System.

### 2.1.1 Dependencies

The module requires the QT dll (currently qt-mt323.dll), bncsif.dll and the Visual C runtime dll MSVCP60.dll.

# 3 Features

When revertives arrive from the source device they are analysed based on the configuration. The results are stored in memory. When the host application requests data the acquisition block sends a pointer to the data. The data sent is one of three states:

| | |
|---|---|
| 0 | OK |
| 1 | Alarm |

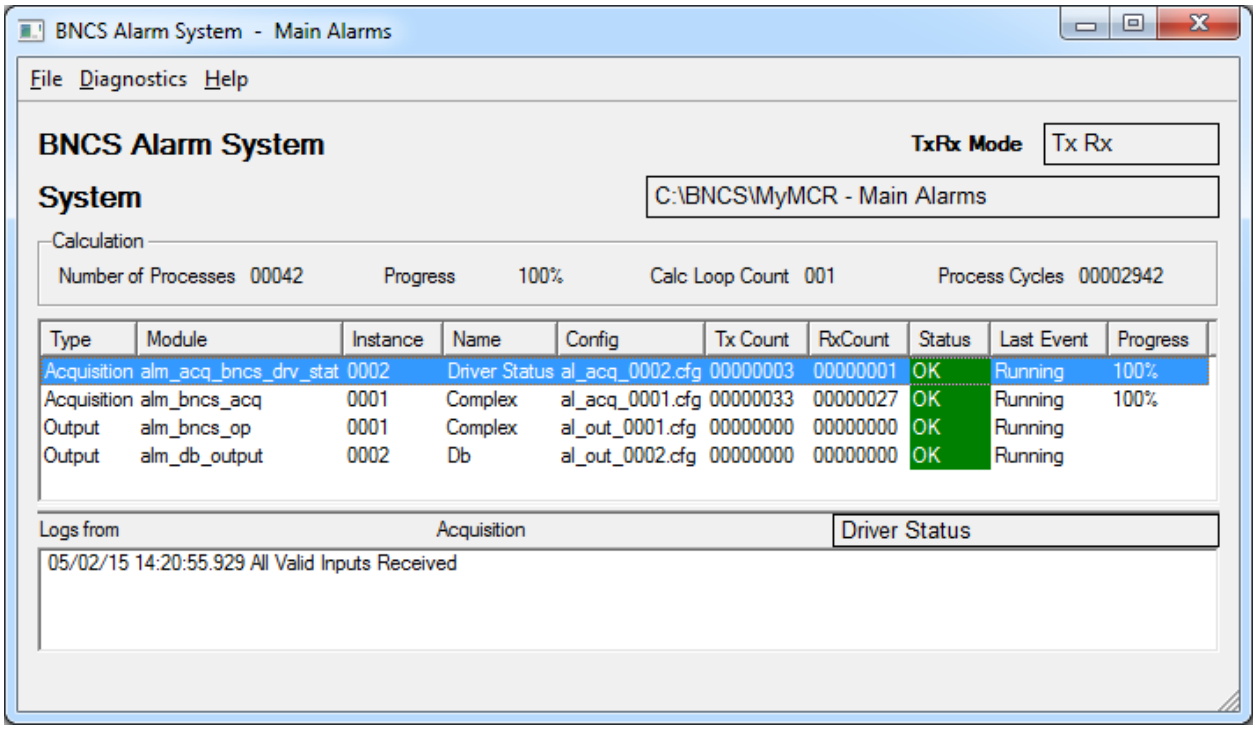
| | |
|---|---|
| -1 | Unknown |

Unknown means that the acquisition module has never received a revertive from this input.

**Note**: All the BNCS Driver Status Acquisition modules in the same application use the same polling object. This object sorts all the devices to be polled numerically by device and index. It polls for a range of inputs where this is more efficient than a single poll. This is to maximise efficiency and minimise the number of messages that CSI has to process.

# 4 Diagnostics

As for AlarmControl.exe.




There are counters for messages to and from CSI, and various other information about the module. The Status field will be 'FAIL' if CSI is not present. . 'Progress' shows how much of the polling process has completed.

Error and diagnostic messages are shown in the lower display.

If the acquisition module has received revertives from all its inputs the message "All Valid Inputs received" will be shown in the lower display – as seen here.

## 4.1 Test Procedures

When polling BNCS devices the BNCS Comms Tx count increases and when receiving revertives the Rx count increases. Figures are written to the screen approximately once per second.

## 4.2 Polling

On start-up if there are devices that don't respond to the first poll they are re-polled until they respond. The time interval between polls increases to a maximum of about 20 minutes.

At present after the first revertive has been received there is no automatic re-polling. Repolling may be started manually either for all modules using the Diagnostics|Restart All Polls, or for a single module using the context (right-click) menu item Restart Poll.

The context (right-click) menu item Next Poll Now will trigger the next poll immediately if selected during a pause between polls.

(*** not yet available)

# 5 Operation

CSI uses heartbeat messages to share driver status information. It maintains an internal infodriver which clients can read data from so any client can establish the Tx/Rx status for any driver number. Each slot in the infodriver corresponds to the driver of the same number.

The revertive data comprises one or more comma-separated strings in the format "nc", where "n" is a workstation number, and "c" is a character representing the Tx/Rx state of that driver on that workstation.

The Tx/Rx state code will be one of the following:

| *Code* | *State* |
|---|---|
| R | Tx/Rx |
| T | Rx Only |
| B | Rx Broken |

From the revertive string the system counts how many instances of the driver are running, and how many are in each state. Based on these counts and the alarm configuration the system works out the appropriate alarm state and passes it to AlarmControl.

## 5.1 Presence Alarm

A presence fault is generated if there are no workstations reporting that the driver is in the Tx/Rx state.

## 5.2 Resilience Alarm

A resilience fault is generated if the number of workstations with the driver in the RxOnly state is not one less than the resilience level configured.

For example, if normal resilience is selected (ie the level is two) then this fault will be signalled if there is not precisely one workstation reporting that the driver is in the RxOnly state.

This alarm is not relevant if resilience is not selected (ie the level is one).

### 5.3 Configuration Alarm

A configuration fault is generated if the list of workstations on which the driver is reported as running doesn't match the list of workstations the driver is configured to be running on (ie from the configuration in launch.xml).

### 5.4 Breakage Alarm

A breakage fault is generated if the driver is in the RxBroken state on any workstation.

### 5.5 Treat RxBroken As RxOnly

If this property is set then any instances of the driver reported as RxBroken will be treated as if they were RxOnly.

# 6 Logging

When Logging is enabled (via AlarmControl's GUI or arguments), selected messages are logged into a file at a path determined by the BNCS settings.

Log files are written in %CC_ROOT%\%CC_SYSTEM%\logs\alarms.

# 7 Documents referenced

This document should be read in conjunction with other documents in the tree.

In particular:

Alarm – overview

alarm - configuration-Bncs driver status acquisition

alarm – mainapplication

The documentation relating to file formats may also be useful.

# 8 Version history

### 8.1 Software Version

Version numbers shown here may not be seen within the software itself. The implementation date is a more reliable way of determining whether a particular issue is present in any particular instance of the software.

| Version | Date | State / Changes | Author |
|---|---|---|---|
| 1.00.00 | March 2012 | Original Release | Richard Kerry |
| | | | |
| | | | |

## 8.2 Document Version

| Version | Date | State / Changes | Author |
|---|---|---|---|
| 1.00.00 | 26 January 2015□ | First version of documentation for this module. | Richard Kerry |
| | | | |
| | | | |

Atos IT Services Limited
4 Triton Square
Regent's Place
London NW1 3HG, UK
http://uk.atos.net/

BNCS
Room G300
Stadium House
Wood Lane
London W12 7TA
colleadiacontrol.it-solutions.gb@atos.net