INTERFACING OPENJAUS WITH THE BYWIRE XGV

This application note outlines a limitation of OpenJAUS and explains the process of patching OpenJAUS so that it will work with the ByWire XGV ("XGV").

WARNING: This application note alone does not contain sufficient information to operate the ByWire XGV. Read the entire XGV User Manual before operating.

What is OpenJAUS?

OpenJAUS is a freely available, open-source implementation of the JAUS standard. Originally developed by graduate students at the University of Florida, it is now maintained by Danny Kent and Tom Galluzzo.

For an organization that does not already have a JAUS implementation, attaining even basic interoperability requires considerable infrastructure – types, message parsing and building, message routing, basic component state logic, and protocol for exchanging messages. By using OpenJAUS, XGV users can leverage a considerable amount of already-developed and already-tested JAUS software.

So, why an App Note?

There is a minor limitation in OpenJAUS that precludes it from interoperating with the XGV. This application note explains how to apply a patch to correct this problem.

The Problem: UDP Broadcast versus UDP Multicast

The commonly accepted protocol for dynamic discovery in JAUS requires each component to produce a periodic Report Heartbeat Pulse message. When one node detects a heartbeat from another node, it can register its IP address and query it for further information. Typically each node subscribes to the same UDP multicast group, and the heartbeat is sent via multicast. In other implementations, the heartbeat is sometimes sent via UDP broadcast (a network address such as 192.168.1.255).

For maximum flexibility, the XGV sends its heartbeat via UDP broadcast, rather than UDP multicast. The issue with the OpenJAUS Node Manager is that it cannot receive broadcast packets, only packets sent via either multicast or directly to the IP address of the OpenJAUS node.

Affected Versions of OpenJAUS

The bug is present in OpenJAUS v3.3.0.a and prior versions. A patch developed by TORC has been submitted to the developers of OpenJAUS, and may be included in a future release of OpenJAUS.

Patching OpenJAUS

TORC has developed a patch that will correct the UDP broadcast issue, enabling OpenJAUS to receive UDP broadcast packets from the XGV. Execute the following steps to apply the patch. This patch has been tested and verified in Linux and Macintosh, so steps will be slightly different for Windows.

1. Download OpenJAUS from www.openjaus.com.
2. Obtain the patch from TORC (support@torctech.com). For reference, it is included at the end of this file.
3. [Linux/Mac] We will assume you have downloaded the tar.gz file (from OpenJAUS.com) to /home/xgvuser/ and extracted OpenJAUS to /home/xgvuser/OpenJAUSv3.3, and that you have downloaded the patch file (openjaus_patch.diff, from TORC) to /home/xgvuser/. We'll also assume you are entering all commands starting in the working directory /home/xgvuser.
    i. Backup the file multicastSocket.c

```
$ cp OpenJAUSv3.3/libopenJaus/src/utils/multicastSocket.c multicastSocket.c.backup
```

    ii. Patch the file multicastSocket.c, which is located in the folder /home/xgvuser/OpenJAUSv3.3/libopenJaus/src/utils/

```
$ patch OpenJAUSv3.3/libopenJaus/src/utils/multicastSocket.c openjaus_patch.diff
```

4. Rebuild libopenJaus and the ojNodeManager. For simplicity's sake, simply build the entire OpenJAUS project.

```
$ cd OpenJAUSv3.3

~/OpenJAUSv3.3$ make
```

Running the OpenJAUS Node Manager

In this section, we briefly outline the steps necessary to run the OpenJAUS Node Manager and get it communicating with the XGV.

1. First, decide whether the OpenJAUS Node Manager will be in the same JAUS subsystem as the XGV (communicating with the XGV controller as another node in the same

subsystem), or a different subsystem (communicating with the XGV as a different subsystem).

2. Next, decide what the JAUS subsystem number and JAUS node number will be for your OpenJAUS node. The subsystem ID and node ID must be different from that of the XGV. If the XGV is subsystem 1, node 1 (default), the following are valid configurations for the Node Manager:

    i. XGV: subsystem = 1, node = 1. OpenJAUS: subsystem != 1, node = 1. XGV and OpenJAUS in different subsystems. This is the recommended mode of operation.

    ii. XGV: subsystem = 1, node = 1. OpenJAUS: subsystem = 1, node != 1. XGV and OpenJAUS in same subsystem.

    iii. Note, you can change the subsystem and node number for the XGV via the System Configuration Utility. See the XGV User Manual for more information.

3. Edit the configuration file for the OpenJAUS Node Manager. Note you may have to create a new configuration file from the template file in OpenJAUSv3.3/ojNodeManager/. The file should be called nodeManager.conf and should be in the working directory that the Node Manager will be run from.

    i. Once you've created the file, open it in a text editor.

    ii. Set key "SubsystemId" under "[JAUS]" to the value chosen above.

    iii. Set key "NodeId" under "[JAUS]" to the value chosen above.

    iv. If the XGV and OpenJAUS will be communicating as different subsystems, enable Subsystem Communications. There are three keys under "[Subsystem_Communications]". Set "Enabled" to "true", "JUDP_Interface" to "true", and "JUDP_IP_Address" to the IP address of the computer on which the OpenJAUS Node Manager will run.

    v. If the XGV and OpenJAUS will be communicating as two nodes in the same subsystem, enable Node Communications. There are three keys under "[Node_Communications]". Set "Enabled" to "true", "JUDP_Interface" to "true", and "JUDP_IP_Address" to the IP address of the computer on which the OpenJAUS Node Manager will run.

4. Run the OpenJAUS Node Manager.

5. Hit "t" to display the current subsystem configuration.