Polhemus Technical Note 1401c

Using Polhemus Trackers with VRPN

Products Affected:

Tracking System: All systems supported by the Polhemus PDI VRPN modules.

Software: PDI 4.2.0 and later

Technical Details:

VRPN version 7.31 and later contains components for supporting Polhemus trackers, including the G4. The Polhemus trackers are supported through the Polhemus Developer Interface (PDI). These PDI VRPN modules work on Windows only. Trackers can be accessed via USB or RS-232 when using the PDI VRPN modules.

The latest version of VRPN is 7.34. If you don't already have it, you can download VRPN 7.34 from here:

https://github.com/vrpn/vrpn/releases/download/version_07.34/vrpn_07.34.zip

When building VPRN with the Polhemus PDI modules you must have version 4.2.0 or later of the PDI installed. The latest revision of PDI (v4.4.0) can be installed from the Host Software CD that comes with the tracker or it can be downloaded from the Polhemus FTP server here:

http://ftp.polhemus1.com/pub/Software/SWD-PS08-08-**4.4.0**-PISDK.exe

When you install the PDI, you will select either **Polhemus SDK for *VC 12 (VS 2013)*** or ***VC 14 (VS 2015)*** in the **Select Components** stage of the install wizard.

Build the PDI components in the VPRN Visual Studio Main_Server project

Open the vrpn.sln solution file using Visual Studio 2013 or later; You will build the Main_Server\vrpn_server project. (If you are using VS 2017, that is fine also.)

Before building, you will need to edit the vrpn/vrpn_Configure.h file:

Uncomment line 407:

```
#define VRPN_USE_PDI
```

Comment out line 328:

```
//#define VRPN_USE_HID
```

Be sure to configure the C/C++ Additional Include Directories and the Linker Additional Library Directories project properties to include the Inc\ and Lib\ folders in your PDI installation. You will also need to add PDI.lib or PDID.lib to the Linker Input Additional Dependencies. When building vrpn_server, you will need to add these settings for the following dependent projects:

- vprn
- vrpndll
- vrpn_server

Configure Polhemus tracker in the VRPN .cfg file

The Polhemus PDI VRPN tracker components are listed and documented in detail in the server_src\vrpn.cfg file starting at line 2401. They are:

vprn_Tracker_G4.................. (supports G4)
vrpn_Tracker_FastrakPDI ...... (supports FasTrak)
vrpn_Tracker_LibertyPDI....... (supports Liberty and Patriot)

Open Source

VRPN is open source, and that means that the PDI modules for VRPN are open source too. The setup methods for the PDI components are located in the vrpn\server_src\vrpn_Generic_server_object.c file:

vprn_Tracker_G4.................. `vrpn_Generic_Server_Object::setup_Tracker_G4()`
vrpn_Tracker_FastrakPDI...... `vrpn_Generic_Server_Object::setup_Tracker_FastrakPDI()`
vrpn_Tracker_LibertyPDI ...... `vrpn_Generic_Server_Object::setup_Tracker_LibertyPDI()`

The source code for the modules' C++ classes of the same name are found in vprn\vrpn_Tracker_PDI.c/.h.

Test the PDI VRPN modules locally

Run the Server

The vrpn_Tracker_G4/FasTrakPDI/LibertyPDI classes are instantiated by the vrpn server. To demonstrate and test locally:

Review the part of the vrpn.cfg file that pertains to the tracker you are interested in. This content documents the commands needed to instantiate and initialize the tracker in the vrpn_server.

It may be easier for you to copy this content into a separate .cfg file for this test. Call the file vrpn_PDI.cfg. You will need to un-comment the sample commands at the end of the section you are working with.

Set up the commands you want to apply to the tracker at runtime. If you are not familiar with your tracker's command interface, first experiment with the PiMgr or with the sample programs installed with the PDI.

From a Windows command prompt, and with the tracker connected and running, run vrpn_server with the following command-line arguments:

```
vrpn_server -f vrpn_PDI.cfg –v –NIC localhost
```

To run this you will need to either copy the vrpn_server.exe into the PDI Lib folder that you linked to in the build or create a shortcut that runs the .exe with the PDI Lib folder as the working directory.

When you run this you will see text output that indicates the status of the connection with the tracker and the commands you specified in vrpn_PDI.cfg.

Build and Run the Client

You will test the module with the vprn_print_devices utility. First you must build it from the vrpn solution Utilities\vrpn_print_devices project.

Then run vprn_print_devices with command-line arguments:

```
<trackername>@localhost –nobutton –noanalog –nodial –notext
```

Here **<trackername>** is the logical name you gave to the tracker object in the vrpn_PDI.cfg file. This name is declared in the first line of your .cfg file commands. If you are using the default examples provided in the .cfg file, they are:

vprn_Tracker_G4	**G4**
vrpn_Tracker_FastrakPDI	**TrackerJoe**
vrpn_Tracker_LibertyPDI	**TrackerJohn**

You can run vprn_print_devices in debug mode from Visual Studio or from the command line. This program will connect to the vrpn_server on the local machine and display the P&O data captured by the server.