ASR-2300 GNU Radio / Eclipse Installation Notes

A2300 Open Source Project

MANAGED DOCUMENT	
Document Version:	0.5
Filename:	ASR-2300 GNU Software Document.docx
Last Revised	9-Sep-13

Document Control

Revision History

Revision	Date	Description	Author
R.00	*9/9/2013*	Initial specification draft	C. Center

Contents

1. Introduction

This document details the gnuradio communication interface for the ASR-2300. The ASR-2300 implements a USB communication interfaces for gnuradio. Internally, these interfaces support the Loctronix Device Communications Interface (DCI) protocol. Additionally, Loctronix provides an enhanced Loctronix Advanced Software Radio™ (ASR) platform targeting navigation applications for GPS/GNSS challenged environments.

1.1 Intended Use

The information contained in this document is the proprietary property of Loctronix Corporation. Any unauthorized use of this information and related protocols stripping prohibited. The specification is managed by Loctronix and may change without notice. Every attempt will be made to maintain backward compatibility.

This document is intended for use by Loctronix and its development partners who have authorized access to detailed design documentation for ASR and the WCA. This information is not for general circulation and requires a nondisclosure agreement before access.

1.2 Related Documents

- GNU Radio

2. Communications Overview

GNU Radio is a free & open-source software development toolkit that provides signal processing blocks to implement software radios. It can be used with readily-available low-cost external RF hardware to create software-defined radios, or without hardware in a simulation-like environment.

GNU Radio applications are primarily written using the Python programming language, while the supplied performance-critical signal processing path is implemented in C++ using processor floating-point extensions, where available.

3. GNU Radio

Post Installation Notes
After installing the gnuradio package I added the following lines to .bash_profile:
export PATH
export LD_LIBRARY_PATH=/home/chris/Projects/gnuradio/src/uhd/host/build/lib
export PYTHONPATH=/usr/local/lib64/python2.7/site-packages

We probably have to add to the LD_LIBRARY_Path our new stuff:
Ours is libUhdAdaptor.so and lives here:
WksAsr2300/pkgs/OpenSource/host/lib/UhdAdaptor/Debug

This section defines the details for working with the proprietary interface. The interface is intended for use via the ASR-2300 WCA driver implemented in support of the WCA platform available for Linux and Windows -based devices. The WCA driver for the ASR-2300 accesses the USB interface directly using lib-usb 1.0(Linux) or Cypress APIs (Windows), enabling high-speed bulk transfer of data between the host platform and the ASR-2300. The minimum configuration defines seven (ep0 through ep6) and points for configuration, control, and RAM operations. Additional endpoints supporting WCA ports may be defined depending on the particular FPGA logic loaded.

This section details the specific specifications for each of the endpoints and implementation considerations.

3.1 Installation

Software Installation Notes:
gnuradio source is can be downloaded from the github using the following command:
> git clone http://git.gnuradio.org/git/gnuradio.git

We installed gnuradio-0:3.6.5-2
Which version of boost am I using? (mine says 0:1.50)
> repoquery yum repository boost<CR>
gnuradio source
gnuradio/gnuradio = build directory
Refer to the gnuradio.org website for build instructions.

3.1.1 Build Option 1 (with manual Debug entry).

Download the gnuradio 3.7.0 or later software and perform a build on you system.

3.1.2 Build Option 2

pwd> /home/chris/Projects/gnuradio/src/gnuradio/build
cmake -DCMAKE_BUILD_TYPE=Debug ../
make && make test

98% tests passed, 3 tests failed out of 175

Total Test time (real) = 80.61 sec

The following tests FAILED:
4 - qa_tag_utils (Failed)
5 - qa_feval (Failed)
6 - qa_kludged_imports (Failed)
Errors while running CTest

3.1.3 Post Build Options

After installing and building the gnuradio package I added the following lines to .bash_profile (note that you may have to "source .bash_profile" after boot.

$ export LD_LIBRARY_PATH=/home/chris/Projects/gnuradio/src/uhd/host/build/lib
$ export PYTHONPATH=/usr/local/lib64/python2.7/site-packages

We probably have to add to the LD_LIBRARY_PATH with our new stuff as follows:
$ export $LD_LIBRARY_PATH:<new_stuff>
Ours is libUhdAdaptor.so and lives here:
WksAsr2300/pkgs/OpenSource/host/lib/UhdAdaptor/Debug
$ export $LD_LIBRARY_PATH:/
WksAsr2300/pkgs/OpenSource/host/lib/UhdAdaptor/Debug

3.2 Custom Motherboard

Create a new device driver when the driver in lib/usrp/ cannot support your custom FPGA or hardware modifications. Make a copy of the relevant driver code in lib/usrp/, make mods, and rename the class. The new device code should register itself into the discovery and factory system.
(see /Projects/gnuradio/src/uhd/host/lib/usrp/)

3.3 Custom Daughterboard

Use code from an existing daughterboard in lib/usrp/dboard/* as an example. Your daughterboard code should subclass an rx dboard, rx dboard, or xcvr dboard; and it should respond to calls to get and set properties. The new daughterboard code should register itself into the dboard manager with a unique rx and/or tx 16 bit identification number.

Installation Notes (Step 1).

/home/chris/Projects/gnuradio/src

wget http://www.sbrac.org/files/build-gnuradio && chmod a+x ./build-gnuradio && ./build-gnuradio

edit .bash_profile to change
export PATH
to:
export PATH
export PYTHONPATH=/usr/local/lib64/python2.7/site-packages

source .bash_profile

UHD
Proprietary software can link to the UHD and not have to open source their code.

UHD Requirements:
> Git Account
> C++ compiler (gcc, msvc)
> CMake 2.6 (or higher)
> Boost 1.36 (or higher)
> LibUSB 1.0 (or higher)
> Python 2.6 (or higher)
> Cheetah 2.6 (or higher
> Doxygen
> Docutils

3.4 Python

This section defines a simple python example.

1. How to find a uhd device:

```
d = uhd.find_devices(uhd.device_addr(options.args))
uhd_type = d[0].get('type')
if( uhd_type == "asr-2300")
   tr0 = uhd.tune_request(freq0)
```

4. USB Interfaces

This endpoint is a messaging type endpoint supporting configuration, command and control messages defined by the USB specification for enumerating and configuring USB devices. Message protocol supported on this interface is in compliance with USB 2.0 and 3.0 standards. Standard support implemented by the FX-3 supports endpoint 0 communications. At this time no specific modifications of this command protocol are expected.

4.1 Fedora for Cypress FX3

Run "lsusb" to get a list of USB devices on the system. Observe something like:
$ lsusb
Bus 001 Device 002: ID 8087:0024 Intel Corp. Integrated Rate Matching Hub
Bus 002 Device 002: ID 8087:0024 Intel Corp. Integrated Rate Matching Hub
Bus 003 Device 002: ID ffff:00ff
Bus 003 Device 004: ID 0781:5530 SanDisk Corp. Cruzer

Run "dmesg" to get a list of all device and see this.
$ dmesg
[1196.921888] usb 3-2: new high-speed USB device number 4 using xhci_hcd
[1196.955469] usb 3-2: New USB device found, idVendor=ffff, idProduct=00ff
[1196.955476] usb 3-2: New USB device strings: Mfr=1, Product=2, SerialNumber=0
[1196.955480] usb 3-2: Product: FX3
[1196.955483] usb 3-2: Manufacturer: Cypress **FX3**

4.2 Fedora USB installation

On Linux, Udev handles USB plug and unplug events. The following commands install a Udev rule so that a non-root user may access the device (for a UHD device):

$ cd <install-path>/share/uhd/utils
$ sudo cp uhd-usrp.rules /etc/udev/rules.d/
$ sudo udevadm control --reload-rules

After the update you can run the following command:
$ lsusb -v -d:00ff
The first few lines indicate here that you do not have permission to access the device. This will result in device open failures.

Bus 003 Device 011: ID ffff:00ff
Couldn't open device, some information will be missing
(this output indicates you do not have "driver" access.

The udev daemon is used by some Linux distributions to handle plug/unplug events for usb devices.

1. Add a group: su /usr/sbin/groupadd Asr2300
2. Add a user to the group: /usr/sbin/usermod -G Asr2300 -a chris
3. Create a file in the directory "/etc/udev/rules.d" and name it "99-Asr2300.rules"

ATTRS{idVendor}=="ffff",ATTRS{idProduct}=="00ff",MODE="0666",GROUP="Asr 2300"

You could echo that above text for the following...
echo 'ATTR... ' > tmpFile
su root
chown root.root tmpfile
sudo mv tmpfile /etc/udev/rules.d/99-Asr2300.rules
sudo udevadm control --reload-rules (load new stuff)
lsusb | grep ffff:00ff (show you installed it ok).

Note my groups with "Asr2300" as the primary. (note that the Asr2300 matches that found in the 99-Asr2300.rules name. I had to reboot Fedora after all of this.

$ groups chris

chris : Asr2300 wheel users dialout usrp

After the driver is installed issue the command "lsusb". To see additional information use the command "lsusb -v -d:00ff

More information can be found at this link (at the time of this document posting):
http://gnuradio.org/redmine/projects/gnuradio/wiki/UdevConfig

Run "lsusb" to get a list of USB devices on the system. Observe something like:
[chris@loc-wdn-dev02 ~]$ lsusb
Bus 001 Device 002: ID 8087:0024 Intel Corp. Integrated Rate Matching Hub
Bus 002 Device 002: ID 8087:0024 Intel Corp. Integrated Rate Matching Hub
Bus 003 Device 002: ID ffff:00ff
Bus 003 Device 004: ID 0781:5530 SanDisk Corp. Cruzer

Run the following command to get a list of all USB devices:
> dmesg
The output from a typical execution is here:

[1196.921888] usb 3-2: new high-speed USB device number 4 using xhci_hcd
[1196.955469] usb 3-2: New USB device found, idVendor=ffff, idProduct=00ff
[1196.955476] usb 3-2: New USB device strings: Mfr=1, Product=2, SerialNumber=0
[1196.955480] usb 3-2: Product: FX3
[1196.955483] usb 3-2: Manufacturer: Cypress

4.3 USB Transport

USB Transport (LibUSB)

The USB transport is implemented with LibUSB. LibUSB provides an asynchronous API for USB bulk transfers.

Transport Parameters

The following parameters can be used to alter the transports default behavior:

- recv_frame_size: The size of a single receive transfer in bytes.
- num_recv_frames: The number of simultaneous receive transfers.
- send_frame_size: The size of a single send transfer in bytes.
- num_send_frames: The number of simultaneous send transfers.

5. Eclipse

Eclipse software build for a simple application (no gnuradio).

The gnuradio eclipse project "workspace" is here: "/home/chris/Projects/WksAsr2300/pkgs/OpenSource/host/". This project includes the Directories of:
A2300Identify
A2300Usb
Common
UhdAdaptor

Creators of custom hardware can create drivers that use the UHD API. These drivers can be built as dynamically loadable modules that the UHD will load at runtime.

For a module to be loaded at runtime, it must be:

- Found in the UHD_MODULE_PATH environmental variable
- Installed into the "<prefix>/share/uhd/modules" directory
- Installed into the "/usr/share/uhd/modules" directory (linux).

5.1 Linkage

Specify the following libraries:

Uhd
boost_system
boost_program_options
dl

Specify the following Library Paths:
/../Common/Debug
/home/chris/Projects/gnuradio/src/uhd/host/build/lib

6. Applications

This endpoint provides bulk transfer of data to a specific location in memory. Each operation specifies a beginning and number of 16 bit words to be transferred. Details of how this exactly is done will be specified in a later version of this document. Memory operations will require a DCI property message to set up memory transfer operation and then this endpoint to write bulk data to the specified memory locations. Writing continues until the specified number of 16-bit words are written.

6.1 A2300Identify

This project requires you to a) run as root and b) identify the library location as follows:

> Under Eclipse | Run | Run Configurations:
1. Select C/C++ Application.
2. Select the "Environment" Tab
3. Enter a variable called "LD_LIBRARY_PATH"
4. Add a value to the library location, for example:
/home/chris/Projects/WksAsr2300/pkgs/OpenSource/host/lib/A2300Usb/Debug/