

This page is intended as a place to compile RCE set up and basic use instructions. Eventually this page will be replaced by SLAC documentation, but it is easier to collect content here. The first draft of this page contains informaion from email chains between Jim, Jaya John, Bruce and Stephanie.

System Requirements

- Linux machine to act as host
 - ◆ Scientific Linux 5(possibly) or 6(preferred), or RedHat Enterprise Linux 5 or 6
 - ◆ the host architecture can be either i86_32 or x86_64
 - ◆ telnet, expect, ssh, minicom (SL6 doesn't install telnet by default.)
 - ◆ Need compiler, linker and debugger already installed.
 - ◆ 2 GB RAM, and 300 MB in free disk space with two network interfaces.
 - ◆ Gigabit ethernet (1000-base T) speed in the secondary network interface card (NIC) is required for full functionality. Can do basic stuff without.

- Can get started with 1Gb ethernet connections, but eventually will need 10Gb interfaces. Here are some links:
 - ◆ <http://www.intel.com/content/www/us/en/network-adapters/converged-network-adapters/ethernet-x520-server-adapter.html>
 - ◆ <http://accessories.us.dell.com/sna/productdetail.aspx?c=us&l=en&s=gen&sku=430-4436>
 - ◆ <http://www.misco.co.uk/product/127983/Intel-Ethernet-X520-DA2-Server-Adapter>
 - ◆ <http://www.intel.com/support/network/adapters/pro100/sb/CS-030612.htm>
 - ◇ BTW: The "direct-attach" - DA - version comes with just SFP cage/s - making them cheap - then we are able to add 1Gb copper or 10Gb fibre as we please ...

- RCE Platform
 - ◆ For now most people are running RTEMS kernel on the RCE platform. Other options are (or will be) available.

- Optional: another computer to remote into the Linux host machine. The shelf is very loud, and probably physically near the Linux host machine since they need to be connected with an ethernet cable at all times and a serial cable during initial set up. I think most people work on a Windows machine and use Putty to remote into the Linux host machine to use the RCE's. Working directly on the Linux host machine should not change the instructions below (we hope).

SDK Installation

Install the SDK on the host Linux machine. The checkout command checks out a tag, not a branch. This feature is allowed in git versions xxxxx or newer. This step only needs to be done once.

```
git clone -q http://www.slac.stanford.edu/projects/CTK/SDK/arm-rtems-rceCA9.git
cd arm-rtems-rceCA9
git checkout -q rtems-V0.7.2-WS
tools/install-devtools.sh
tools/install-sdk.sh
```

```
git clone -q http://www.slac.stanford.edu/projects/CTK/SDK/i86-linux-64.git
cd i86-linux-64
git checkout -q linux-V0.7.2-WS
tools/install-sdk.sh
```

Troubleshooting tips:

- If you have a proxy server, make sure git is set accordingly
- Make sure you have enough free space to store the libraries. If it runs out of space in the installation, it does not explain the failure.

- Using `update-sdk.sh` or trying to run `install-sdk.sh` after a failed installation won't always work. Either start over with a new git clone or delete the `lib`, `include` and `tgt` folders before trying again.

Environment Variables

Add new env variables to host computer. Adjust `RPT_ROOT` to match your system install directory. This step needs to be repeated any time the environment variables on your computer are reset, so I do this every day. If it is not set, your system will not know the command `cob_dump` (and a bunch of others, but this is an easy one to check).

```
version=V0.7.2-WS

export RPT_ROOT=/home/daquser/${version}
export RTEMS_SDK=${RPT_ROOT}/arm-rtems-rceCA9
export I86_SDK=${RPT_ROOT}/i86-linux-64

source ${RTEMS_SDK}/tools/envs-sdk.sh
source ${I86_SDK}/tools/envs-sdk.sh
```

Download Workshop Examples

This step is optional, but these examples are used to demonstrate things on this twiki. Start in a folder that is on your export path so the RCE's will be allowed to mount the drive. (If you don't have an export path, you can create one later. See instructions below.)

```
wget https://confluence.slac.stanford.edu/download/attachments/184719968/workshop_examples.tgz?ap
tar xf workshop_examples.tgz?api=v2
```

Open the example you want to use. Build the example using `./build.sh`. The executables it creates will be used later.

Shelf Hardware Setup

Add photos here

1. Set #4 dip switch on master COB board.
2. Put into shelf.
3. Using Serial port to Ethernet cord, connect the Ethernet port labeled "serial" on the front of the shelf manager to the serial port on the host linux machine. This only needs to be plugged in for set up. Doesn't seem to have any issues leaving it plugged in.
4. Using a standard Ethernet cord, connect the standard Ethernet port (the one next to the "serial" Ethernet port) to the secondary/internal network interface on the host linux machine. This can be connected through a switch, if the host computer is being used for other things, too.
5. (Optional for now, required to pass data) Connect the 1Gb or 10Gb Ethernet ports on the COB board to the host linux machine.
6. Power on using the right most power port. The shelf has triple powersupplies, but we equipped only one because we don't need 100% uptime. Just plug it in, there is no power switch. Fans will make a lot of noise at first, then quiet down a bit. Most parts of this system can be adjusted while the shelf is powered on, but I prefer to set up everything, then turn on the power just in case.

Shelf Network Setup

Linux hosts will have dual interfaces.

- The primary NIC is be hooked to your LAN however you wish.

- The secondary (red) NIC should be set up to get its address via DHCP, and the interface itself should be able to be brought up and down by the users.

The shelf will have two network interfaces.

- The red one is the COB network, which is controlled by the COB itself.
- The black one is for the shelf manager. It needs a **static** IP address, as described in the Shelf setup section of the Oxford Notes PDF.

Set Shelf IP

The shelf IP can be set by the user or the user can leave it set to whatever it was automatically set to. Users should check the shelf IP anytime it is suspected the shelf may have been assigned a new IP (for now I do this after the shelf has powered on, but it hasn't changed yet). Use the serial port to Ethernet cable to connect the "serial" Ethernet port on the self manager to the serial port on the Linux host machine.

Connect to the COB over the serial cable

First time only

Follow these instructions only the first time using this computer, RCE crate or USB port. If necessary, install minicom.

```
sudo yum install minicom
```

Determine which port serial to usb cable plugged into

```
dmesg | grep tty
```

Configure minicom.

```
sudo minicom -s
```

Select: Serial Port Set-up --> A - Serial Device : /dev/ttyUSB0 where ttyUSB0 is the port the serial to usb converter is plugged into save as default

Now minicom should start automatically.

Every time

Do the steps above once for each hardware set-up. Do the steps below each time you want to check or change shelf manager settings.

```
sudo minicom
```

```
login = root
```

```
password = [empty]
```

To exit minicom, type Ctrl+a q.

To view IP addresses, type `ifconfig`.

To view shelf name, type `clia shelfaddress`.

For information on changing settings, type `clia setlanconfig`.

Note: Minicom can also be used to connect to a specific RCE. From the host Linux machine, ssh into the DTM (bay 4 element 0, user/pw = root/root). Use `minicom bay0.2` on the DTM to connect to the RCE at bay 0, element 2. This additional usage of minicom is unrelated to setting the shelf IP, but listed here to let you know that there are multiple sets of minicom instructions that don't go together.

Network port configuration on Host:

This should only need to be done once. This allows the user to use `ifup` and `ifdown` (i think).

```
$ cat /etc/sysconfig/network-scripts/ifcfg-p4p1
DEVICE=p4p1
HWADDR=68:05:CA:2F:8F:EA
TYPE=Ethernet
UUID=78edf86d-9419-4e7b-a770-7ba02ebef87b
ONBOOT=no
NM_CONTROLLED=no
BOOTPROTO=dhcp
USERCTL=yes
```

Notice the following:

- This interface does not come on at boot. The reason for this is that at boot of the host, the shelf is not guaranteed to be powered on. This scenario would lead to a 1-5 minute delay in the host booting, and is irritating. Thus we put the port under user control.
- Since the port is user controlled, it can and should be restarted by the user when it needs a new IP address. This is typically needed when a shelf is installed, after the shelf has been powered down for some period of time, or after booting the host.
- For RHEL6/SL6, the `NM_CONTROLLED` option is off. This fixes an issue with NetworkManager.

NFS exports:

Allow the NFS disk to be exported from LinuxHostComputer on the private network. Not sure how often this needs to be done.

Warning: error in one or both of the code bits below.

```
cat /etc/exports:
/opt/rce_export 10.0.1.0/16(rw,root_squash,async)
```

Or for Oxford where RCEs are set up on the 192.168/16 subnet.

```
$ cat /etc/exports
/home/daquser 192.168.0.0/16(rw,root_squash,async)
```

Make sure your compiled executable files to be run on the RCE (the `workshop_examples` folder) are located in the `rce_export` or `daquser` folder.

Start the NFS file server so RCE's will be able to mount the drive allowed above. This is normally already running, but if not, use the next line of code. `sudo /sbin/service nfs start`

Set up network

Here is Bruce's method.

ADD GRAPHIC HERE

We have lots of IP addresses and 2 sub-networks. The first network is 10.0.0.X which has LinuxHostComputer and the shelf manager on it:

- 10.0.0.200 LinuxHostComputer
- 10.0.0.212 tracker_one_sm

The other network is 10.0.1.X and this is attached to the COB. I've made a static ip for LinuxHostComputer on this one as well 10.0.1.200. The COB DHCP gives 10.0.1.254 to the DTM and normally 10.0.1.1-8 to the RCEs (I think).

```
[host]$ set_shelf_ip_info ... set FRU on SM
```

I've also changed the static IP on the shelf manager (eth1).

```
[SM]$ clia getlanconfig 2 3 # Was 192.168.2.2
[SM]$ clia setlanconfig 2 3 10.0.1.128
[SM]$ clia shelfaddress "tracker_one_sm"
```

This shelf address is the address the host computer sees for the shelf manager. Items on the shelf see a different name when they look for their own location. Normally, the shelf manager address is the same as the shelf address, but with an added _sm or -sm. I don't know how to set the shelf address as seen by items on the shelf.

And I disabled the firewall for connections from eth1.

```
iptables -I INPUT 1 -i eth1 -j ACCEPT
```

To give it two IP addresses (keeps the logical networks to COB and SM separate) I did:

```
$ sudo ifconfig eth1 add 10.0.0.200
```

Then ifconfig shows:

```
eth1 ... inet addr:10.0.1.200 Bcast:10.0.1.255 Mask:255.255.255.0
eth1:0 ... inet addr:10.0.0.200 Bcast:10.0.1.255 Mask:255.255.255.0
```

If the mask is wrong, delete eth1:0, fix the mask, then re-add eth1:0.

Oxford Notes method

Similar to Bruce's, but slightly different. Bruce's method is newer, but I think it may be missing a few things from the Oxford Notes method.

Shelf setup

Serial cable to shelf manager from the host with the i86-linux-XX SDK installed.

To set the static IP address for an ASIS shelf:

```
ssh root@<shelfmanager_ip>
pw: (there is no password)
# clia setlanconfig 1 3 10.255.5.7
```

Here is Bruce's method.

RCESetupInstructions < Sandbox < TWiki

```
# clia setlanconfig 1 6 255.255.255.0
# clia setlanconfig 1 12 10.255.255.1
# clia shelfaddress ceg
# reboot
```

Then set the shelf IP record from the Host:

```
$ set_shelf_ip_info --shelf=10.255.5.7 --vlan=1 --untagged=0 --
tagged=1 --boundary-violations=0 --group-base=192.168.209.1 --group-
end=192.168.209.254 --subnet-mask=255.255.255.0 --gateway=0.0.0.0 -v
```

Log back into the shelf manager via ssh and reboot it again. Reset the cold data on all COBs:

```
cob_cold_data_reset 10.255.5.7/6/4/0
```

Add the shelf IP / name to /etc/hosts. On pprcedaq2, the setup was:

```
[daquser@pprcedaq2 ~]$ cat /etc/hosts
127.0.0.1 localhost localhost.localdomain localhost4
localhost4.localdomain4
::1 localhost localhost.localdomain localhost6
localhost6.localdomain6
10.255.5.5 snowbird-sm
```

You may also add this to your normal DNS.

After all this, restart the network interface on the host.

Use Case Example

Here is a summary of the main commands and IP addresses used in the workshop, with some modifications. It assumes everything is set-up, as described above. If someone else already set up your equipment, you can probably start here (I hope).

Part 1: Connecting to the Linux host and checking the network connections

If using a Windows or other remote terminal, ssh into the HostLinuxMachine.

Add environment variables. (May have already been done, but sometimes they get reset.) For Oxford:

```
version=V0.7.2-WS
export RPT_ROOT=/home/daquser/${version}
export RTEMS_SDK=${RPT_ROOT}/arm-rtems-rceCA9
export I86_SDK=${RPT_ROOT}/i86-linux-64
source ${RTEMS_SDK}/tools/envs-sdk.sh
source ${I86_SDK}/tools/envs-sdk.sh
```

For RAL (Just different paths.)

```
export RPT_ROOT=/opt/rce_platform/steph
export RTEMS_SDK=${RPT_ROOT}/arm-rtems-rceCA9
export I86_SDK=${RPT_ROOT}/i86-linux-64
source ${RTEMS_SDK}/tools/envs-sdk.sh
source ${I86_SDK}/tools/envs-sdk.sh
```

Compile the executables you want to run. Save to a folder on the HostLinuxMachine. For the this example, the executables are stored at /home/daquser/participants/jjj/test/workshop_examples/compiled

Next check that the second network connection to the COB (p4p1, p#k# or eth1, depending on your network, we will use p4p1 for this example) is good:

```
ifconfig
```

What we want to see is that p4p1 is listed and has an inet addr assigned this will be 192.168.209.x, probably 192.168.209.30.

If p4p1 is not listed, bring up the interface:

```
ifup p4p1
```

Wait for completion, then check again with ifconfig.

If p4p1 is listed but has fewer lines of information than shown above, especially if the line beginning with inet addr does not appear, which means the COB has not given an IP address to PPRCEDAQ1's second network interface, then reset the interface:

```
ifdown p4p1
ifup p4p1
```

Wait for completion, then check again with ifconfig.

You can ping the shelf manager to check that the shelf is powered up and visible:

```
ping isis-sm
```

At this point, we can talk to the shelf manager.

Part 2: Checking the state of the COB

Next, check the state of the shelf, as the RCEs tend to boot up incompletely after powering up (due to the known XAUI race condition issue in firmware):

```
cob_dump isis-sm --rce
```

What we want is all RCEs coming up in state 0x00 i.e. successfully booted up and ready:

```
=====
|           RCE   |Ena|Rst|Rdy|Dne|Vok|BTemp|JTemp| RCE State |
|=====|
| isis-sm/6      |
|-----|
| DPM0: RCE0   | Y | N | Y | Y | Y | 43 | 57 | 0x00 |
|           RCE2 | Y | N | Y | Y | Y | 47 | 56 | 0x00 |
| DPM1: RCE0   | Y | N | Y | Y | Y | 46 | 56 | 0x00 |
|           RCE2 | Y | N | Y | Y | Y | 50 | 54 | 0x00 |
| DPM2: RCE0   | Y | N | Y | Y | Y | 40 | 56 | 0x00 |
|           RCE2 | Y | N | Y | Y | Y | 47 | 50 | 0x00 |
| DPM3: RCE0   | Y | N | Y | Y | Y | 40 | 56 | 0x00 |
|           RCE2 | Y | N | Y | Y | Y | 45 | 49 | 0x00 |
| DTM:  RCE0   | Y | N | Y | Y | Y | 46 | 56 | 0x00 |
|-----|
|=====
```

If any RCEs are in state 0x14 i.e. stuck due to race condition, then we need to reset them. On our shelf, all of the DPM RCEs come up stuck after power-up, but the DTM will be fine. This command resets all the DPMs on the COB in slot 6:

```
cob_rce_reset isis-sm/6/dpm
```

Then check again:

```
cob_dump isis-sm --rce
```

Part 3: Connecting to an RCE with telnet:

Telnet to a particular RCE: (here slot 6, bay 0, RCE 0)

```
telnet $(atca_ip isis/6/0/0 --ifname p4p1)
```

Note: The slot number is the physical slot the COB board is in on the shelf, 1-16 allowed, but look at your shelf. Allowable bay numbers are 0-4. Allowable RCE element numbers are 0 or 2.

The console text should be similar to:

```
Trying 192.168.209.20...
Connected to 192.168.209.20.
Escape character is '^['.
```

```
RTEMS SHELL (Ver.1.0-FRC):/dev/pty0. Feb 16 2014. 'help' to list commands.
[/] #
```

If there is no output, it may be a firewall problem. See the troubleshooting section on how to set iptables. If that is not the problem, see if you are confusing the shelf manager with the self. Note that to address the shelf manager from the host computer, the name `isis-sm` was used. Here we are using `isis` to address hardware on the shelf.

If there is a delay, then we see only a telnet prompt:

```
telnet>
```

Then it means that this RCE has frozen (or that interface p4p1 is not up, but the earlier steps should have checked this).

Type q and ENTER to quit telnet, then reset the RCE (slot 6, bay 0, RCE 0 here) and try again:

```
cob_rce_reset isis-sm/6/0/0
telnet $(atca_ip isis/6/0/0 --ifname p4p1)
```

Then, assuming that you have previously done the workshop exercises on pprcedaq1 (the linux host machine), you can mount your working directory from pprcedaq1 and assign a namespace: (replace `/jjj` with the name of your own directory, created under `/home/daquser/participants/` during the workshop)

```
mkdir /test
mount -t nfs 192.168.209.30:/home/daquser/participants/jjj /test
ns_assign examples /test/workshop_examples/compiled
```

If you get the following error with mounting, check to ensure that your directory is on the approved export path list for the host machine. The newly created test directory will get deleted, I think on reboot of the COB.

```
Unable to contact NFS server - invalid port? (RPC: Program not registered)
```

```
error: Unknown protocol
```

Then we can run executables from the workshop. First clear the system log, then run an executable, then view the system log to see the programme's output:

```
syslog -c
run examples:hello_1.exe
syslog
```

If you get the following error, check to make sure your ns_map path is correct:

```
1988/01/01 00:02:23.540441: lnk_load failed with error 0x10002=
```

Once done, type exit to close the telnet connection.

Note that the workshop's pseudo-ARP examples (rx.exe and tx.exe) time out due to the larger 6-slot shelf.

Part 4: Connecting to an RCE through the serial link from the DTM:

For using the minicom serial link to the RCEs, from pprcedaq1, start by telnetting to bay 4, rce 0, which is the DTM:

```
ssh root@$ (atca_ip isis/6/4/0 --ifname p4p1)
```

Then open a serial session to a particular RCE using minicom: (here bay 3, RCE 2)

```
minicom -w bay3.2
```

Once done, press Ctrl-A, then Q, to exit the minicom session. Don't type exit while in the minicom session, otherwise the RCE hangs.

Misc Tips and Troubleshooting

- What I found when I renamed our shelf from ceg to isis was that atca_ip was the last thing I got working. It took a few reset/restart cycles to work, more than I had expected. I alternated between restarting the PC, cob_cold_data_reset, cob_rce_reset isis-sm/6/dpm, and finally powered down the shelf for a moment. After that, atca_ip started working.
- When comparing examples between Oxford, SLAC and RAL, different names are used. Here is a mapping of names that are functionally equivalent. I think this is right, but not confirmed.
 - ◆ Network Interface: eth1 = p#k#
 - ◆ Host Machine: LinuxHostComputer = heplnw180 = pprcedaq2
 - ◆ Shelf Name: tracker_one_sm = Snowball = ceg = isis
- Sometimes the RCE gets stuck configuring the ethernet XAUI interface. Errors from BOOTP/ DHCP will be seen in the console. If this happens use cob_rce_reset to reset the RCE until it comes up all the way. Sometimes this must be done repeatedly.
- Don't do the UDP exercise from a telnet session. This tickles a bug in the XAUI plugin and locks up the network.
- If atca commands are not working (not returning anything), it could be a firewall problem. Try this line of code on the Linux host machine:
 - ◆ iptables -A INPUT -i eth2 -m state --state NEW -j ACCEPT
 - ◆ Where eth2 is the network interface card on the Linux host machine that is connected to the shelf.

- ◆ This command requires sudo.

Useful Links

SLAC Workshop at Oxford, Jan 2015 [↗](#)

SLAC Documentation Site [↗](#)

-- StephanieWSullivan - 2015-02-02

This topic: Sandbox > RCESetUpInstructions

Topic revision: r15 - 2015-10-21 - StephanieWSullivan



Copyright &© 2008-2021 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.
or Ideas, requests, problems regarding TWiki? use [Discourse](#) or [Send feedback](#)