

RCE Teststand Software

This page describes how to compile and run the RCE teststand software at SLAC and CERN. Checking out the packages from SLAC CVS requires a SLAC Unix login account. Required steps are identical for the SLAC and CERN environment unless otherwise mentioned.

Recommended host settings used in examples

	SLAC (Lab1)	SLAC (cosmic telescope)	CERN
[BUILD_HOST]	rddev101	rddev101	pcslac02a
[IPC_HOST]	atca01	rdcds105	pcslac02a
[RCE]	rce20	rce22	rce27
[RCE_INTERFACE]	0	0	1

SLAC module setup:

Cosmic Telescope (rce20)

Module	Channel	TurboDAQ Configuration File
M511958	2	/reg/g/atlas/rcecalib/moduleconfigs/M511958/configs/CosTel.cfg
M512819	3	/reg/g/atlas/rcecalib/moduleconfigs/M512819/configs/CosTel.cfg
M512737	6	/reg/g/atlas/rcecalib/moduleconfigs/M512737/configs/CosTel.cfg
M512855	7	/reg/g/atlas/rcecalib/moduleconfigs/M512855/configs/CosTel.cfg

SLAC Lab1 Teststand (there is no known good module configuration available: M511958 can be used instead)

Module	Channel	TurboDAQ Configuration File
M12898	4	/reg/g/atlas/rcecalib/moduleconfigs/M511958/configs/CosTel.cfg

A teststand can be booked here:

<https://pc-sct-www01.cern.ch/cgi-bin/workplan.cgi?clIndex=1>

Time slots are CERN time (CET or CEST).

Setting the RCE development environment (C-Shell)

```
source /reg/g/atlas/rcecalib/setup/setup_rce.csh
```

To execute this setup at login the following lines should be added to \$HOME/.cshrc

```
if ( -r /reg/g/atlas/rcecalib/setup/setup_rce.csh ) then
source /reg/g/atlas/rcecalib/setup/setup_rce.csh >& /dev/null
endif
```

Setup for Bourne-Shell

```
source /reg/g/atlas/rcecalib/setup/setup_rce.sh
```

Or adding the following lines to \$HOME/.bash_profile

```
if [ ! -r /reg/g/atlas/rcecalib/setup/setup_rce.sh ]; then
source /reg/g/atlas/rcecalib/setup/setup_rce.sh 2>&1 > /dev/null
fi
```

Checking out and setting up the RCE and STControl/PixLib packages into a directory. The setup scripts assume that top RCE release directory is found in \$HOME/rce. HOME should point to /reg/lab1/home/\$USER. Several release installations should be managed by symbolic links.

```
[BUILD_HOST] mkdir $HOME/RCE
[BUILD_HOST] cd $HOME/RCE
[BUILD_HOST] cvs co -d rcecalib-CVS-HEAD release
[BUILD_HOST] cd $HOME
[BUILD_HOST] ln -s RCE/rcecalib-CVS-HEAD rce
[BUILD_HOST] cd rce
[BUILD_HOST] cvs co pixlib rce rceapp rceers rceowl rcecalib rceipc rceis rceoh rceusr
```

Building the RCE and STControl/pixlib packages.

Note: for some RCE the code needs to be compiled with the RCE_INTERFACE=1. The default setting is RCE_INTERFACE=0 (see table above)

```
[BUILD_HOST] make i386-linux
[BUILD_HOST] make ppc-rtems-rce405-opt RCE_INTERFACE=1
[BUILD_HOST] cd pixlib
[BUILD_HOST] make
```

Running a threshold scan and retrieve S-Curve-Fit, Occpancy and ToT mean/sigma

- Start ipc initial and 'rcetest' partition

```
[IPC_HOST] ipc_server &
[IPC_HOST] ipc_server -p rcetest &
[IPC_HOST] is_server -p rcetest -n RceIsServer &
```

- Download the calibration server application to the RCE.

```
host_bootloader -r rce31 -s cmdcalib.txt -i ~/ipc_root.ref -l ~/rce/build/rcecalib/mod/ppc-rtems-rce405-
opt/calibservermod.1.0.prod.so
```

- cmdcalib.txt contains commands executed by host_bootloader. For example

```
reboot
setenv TDAQ_PARTITION rcetest
setenv TDAQ_IS_COMPRESSION_THRESHOLD 100000000
reboot
```

- Run the command line calibration client on the SLAC Lab1 teststand; the module is connected to channel 4.

```
[IPC_HOST] calibclient -p rcetest -c 4 -m /reg/g/atlas/rcecalib/moduleconfigs/M511958/configs/CosTel.cfg
```

Running STcontrol

- To start STcontrol

```
[IPC_HOST] STcontrol
```