Table of Contents

RICH UKL1 Operating Information

The team responsible for the operation of the RICH UKL1 system are Nicola Mangiafave, Gareth Rogers, Hugh Skottowe and Steve Wotton.

The number of the RICH readout operations mobile is 163837

The UKL1 technical reference manual⬚ describes the hardware and firmware of the UKL1 boards. The document also describes the raw data format of the RICH detectors

Current and previous releases of the UKL1 firmware can be found in the firmware archive area⬚.

Here you can find the list of UKL1's together with the column positions to which they are connected in spreadsheet form.

⚠The UKL1 crates may normally remain on but should be switched off when not in use for extended periods or in advance of scheduled power cuts.

Running notes

- When in ALICE mode the UKL1s can not handle consequtive events. This means that when running from the top level ECS controls `Time Alignment Events`, TAE, must not be enabled. This sends 5 consequtive events which will cause the UKL1 boards to throttle after 1 trigger. With TAE enabled the `Run limited to` setting appears not to work. This should not be a problem in LHCb mode.

Recent L1 problems and solutions

- [Believed to be fixed. Nov. 2007] Sometimes a GBE card will not initialise correctly after power up and no data will arrive from the corresponding L1 module. The symptoms are no data packets from the offending module which will probably result in "Incomplete event" messages that implicate a single L1 module whose status is otherwise OK. To fix the problem run the program ukl1-cfg on every L1 board after the power is switched on and you are able to log in. You should then reload the FPGAs on each board and reconfigure. Please try to contact the expert first before trying this as there are other possible reasons for "Incomplete event" messages.

Controls (PVSS, FSM)

The UKL1 controls package for the UKL1 boards runs on the r1daq01 and r2daq01 controls PCs for RICH1 and RICH2 respectively. The UKL1 control panel can be started from the RICH console.

The L1 control interface is implemented in PVSS and now stable, with the only major feature missing being the L1 pixel masks. The project is actively being updated and new versions are still being released, most containing minor patches. This document will provide a quick start guide to using the UKL1 project, from installation to using the FSM. A more detailed overview of the UKL1 controls project can be found on the RichUkl1Controls page.

All the information below will is identical, unless otherwise stated, for both RICH1 and 2 hence either x or X will be substituted for the number (depending on which matches the case of the context best). Simply substitute for a 1 or 2 depending on which RICH is required.

#Online environment

LHCb online environment

In the LHCb online area, in the case of both RICH1 and RICH2, the UKL1 controls project is part of the RICH1 and 2 top level ECS project and can be accessed by navigating the FSM tree. It is within the RICHX_DAQ then RICHX_L1 sub-system. From here you can access the all the UKL1 boards for the given RICH and issue commands from the FSM. There is however restricted functionality in certain cases due to the nature of the distributed system, for example the start monitoring buttons do not work. This can be fixed by working locally.

The UKL1 PVSS project have been setup to start automatically when the PC boots `rxdaq01`. In the case that the project is not running it can be launched by ssh'ing into `rxdaq01` and run the command `sudo /service/pvss_mp start RXDAQL1` this should start the appropriate project. In the place of `start stop` or `restart` can also be substituted to stop or restart the project.

When started in this manor the various UKL1 panels cannot be accessed directly from the PVSS console and the projects must be accessed by one of the following shortcuts:

- `RX_UKL1_FSM`
- `RX_UKL1_DEN`
- `RX_UKL1_fwUkl1`
- `RX_UKL1_PARA`

which are located in `/group/rich/oper/RICH-X/ExpertsOnly/`. They are typically only for expert use.

Other environments

The UKL1 projects are installed outside of the LHCb online environment, such as the SSB2 lab. At present it is installed in no other distributed systems and is typically accessed directly. The PVSS administrator panel can be used to start and stop the project and the PVSS console to stop and start individual managers. For this the `Device Editor and Navigator`, `DEN` can be used to provide all the desired functionality.

Quick start

Once the project is up and running then the UKL1 boards can be accessed through the FSM tree. The minimum set of actions that need to be performed to get the UKL1 running are outlined below and it assumes that the UKL1s are opened from the `RICHX_L1` level and commands are issued from there, further that the

UKL1s are being operated 'standalone' i.e. not as part of a run orchestrated from the RICHX ECS level. Once configured the UKL1s are of course able to operate in conjunction with the TFC and L0s, however certain parameters are normally issued from the top level and are ignored here.

- If the UKL1s are not in the `NOT_READY` state click the L1 state and send a `Reset`, a `Stop` action may be necessary to access the `Reset` action. Now the UKL1s should be in the `NOT_READY` state.
- Send a `Configure` from the FSM menu. There are 4 parameters, only the `RUN_MODE` is necessary. This must be the recipe that is to be loaded. `PHYSICS`, `LHCb`, `LHCB`, `DEFAULT` will configure the UKL1s for LHCb triggers and `ALICE` or `CALIB` for ALICE triggers. Typically only `PHYSICS`, `LHCb` and `ALICE` are available outside the LHCb online environment. It should now be in the `READY` state.
- Send a `Start` from the FSM menu. All parameters can be ignored here. It should now be in the `RUNNING` state.

The UKL1s should now be able to receive, process and send events.

Installing the project

The RICHX UKL1 PVSS project should be installed on `rxdaq01`. The following guidelines assume that the relevant projects exist with the relevant framework components installed. How to install/update the UKL1 specific components is outlined here.

There are two packages that must be installed as part of the UKL1 installation procedure `fwUkl1` and `fwUkl1ExceptionHandling`. They can be found in `_lbRICHPackages/RXDAQL1`, in this directory the extracted folders for the framework components should exist. There is actually no difference between the projects for RICH1 and 2 from the perspective of the library and all differences are configured at run time. The components are stored in separate directories currently for historic reasons. This may be updated in the future, in which case the two packages will be found in the directory `_lbRICHPackages/RXDAQL1` where it is a literal X.

Both the `fwUkl1` and `fwUkl1ExceptionHandling` are standard framework packages and can be installed using the standard framework component tools, link⧉. They have the relevant dependencies defined and if a required framework component is not installed it should be installed before completing the installation of the component. `fwUkl1` is dependent on `fwUkl1ExceptionHandling` so `fwUkl1ExceptionHandling` should be installed first.

When installing `fwUkl1` the following screen should appear during the installation:
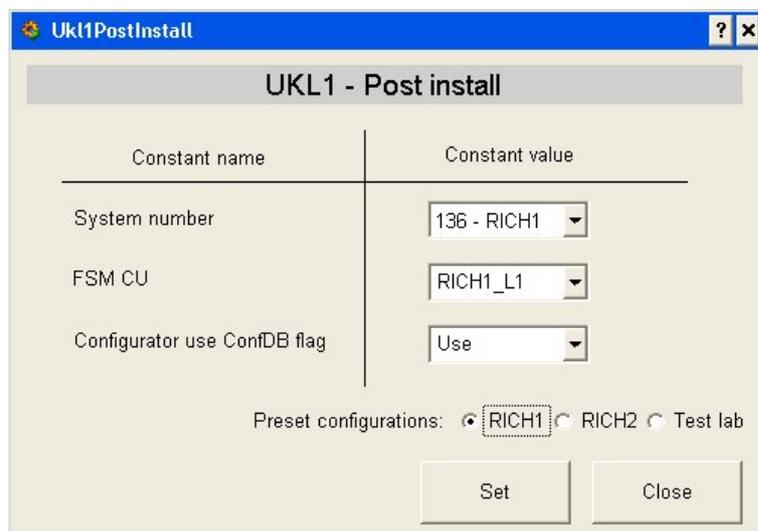
Typically one of the predefined selections is suitable, however the options can be set manually if desired.

It is advised to uninstall both packages before updating/reinstalling them and to restart the project after deletion and installation (the reset is prompted by the installation tool). Both run post install scripts, which in the case of `fwUkl1` will ensure that the FSM tree is setup correctly.

Hardware and firmware notes

Rack control

Powering the L1 crate on or off can be done through PVSS or by logging on to r2daq01:

To turn on: `/group/online/rackctrl/rackctrl start D3/D3C04U`

To turn off: `/group/online/rackctrl/rackctrl stop D3/D3C04U`

For status: `/group/online/rackctrl/rackctrl status D3/D3C04U`

CCPC

The UKL1 CCPC servers for RICH1 and RICH2 are `r1daq01` and `r2daq01` respectively. The CCPCs themselves are called `r1ukl1nn` and `r2ukl1nn`.

Programs can be cross-compiled for the CCPCs on these machines. See the makefile in `/group/rich/L1/src/ukl1cfg` for examples.

Readout network

The UKL1 network parameters can be found here.

The RICH1 and RICH2 crates each have a Hugin board which is responsible for collating the throttle signals from the UKL1s. The slot number of the UKL1 in its crate corresponds to the port input number on the Hugin board. The table also provides a map of the RICH1 and 2 UKL1 name to slot/Hugin port position.

The UKL1s are assigned the following source IP addresses from the range indicated above:

CCPC	IP address	RICH	Slot	Alias
rxukl102	??.??.??.??	SSB2		rxukl102
rxukl110	??.??.??.??	SSB2		rxukl110
r1ukl101	192.169.14.1	RICH1	D3C01U/06	rxukl103
r1ukl102	192.169.14.2	RICH1	D3C01U/09	rxukl113
r1ukl103	192.169.14.3	RICH1	D3C01U/11	rxukl114
r1ukl104	192.169.14.4	RICH1	D3C01U/13	rxukl116
r1ukl105	192.169.14.5	RICH1	D3C01U/15	rxukl117
r1ukl106	192.169.14.6	RICH1	D3C01U/17	rxukl118
r1ukl107	192.169.14.7	RICH1	D3C01U/19	rxukl120
r2ukl101	192.169.1.1	RICH2	D3C04U/04	rxukl109
r2ukl102	192.169.1.2	RICH2	D3C04U/06	rxukl108
r2ukl103	192.169.1.3	RICH2	D3C04U/08	rxukl106
r2ukl104	192.169.1.4	RICH2	D3C04U/10	rxukl111
r2ukl105	192.169.1.5	RICH2	D3C04U/13	rxukl112
r2ukl106	192.169.1.6	RICH2	D3C04U/15	rxukl104
r2ukl107	192.169.1.7	RICH2	D3C04U/17	rxukl105
r2ukl108	192.169.1.8	RICH2	D3C04U/19	rxukl101

Special settings

A suitable value for the L1 TFC latency compensation register on the L1 boards (register 22=0x16) is 0x67 at the pit, instead of the 0x1e used in the SSB2 and in Cambridge.

For ALICE mode always use MEP packing factor equal to 1. To prevent buffer overflow always make sure that the throttle is working and that the ODIN trigger gap generation is set to the maximum value (15) instead of the value 1 used for normal LHCb running.

Firmware updates

Firmware can be updated using the Xilinx Platform USB cable connected to the front-panel JTAG port with the JTAG port select switch set to `Ext`. To update using the CCPC/GC interface put the switch into the `GC` position and run the following command on the CCPC:

- `jam -n3 -aRUN_XILINX_PROC filename`

The process is slow and may take 1.5 hours but can be done on all boards simultaneously.

Tools and diagnostics

RICH L1 related software tools are maintained in the `lbgw:/group/rich/L1/` tree:

- `src/` contains program source files
- `bin/` contains executable programs and shareable libraries
- `include/` contains common header files
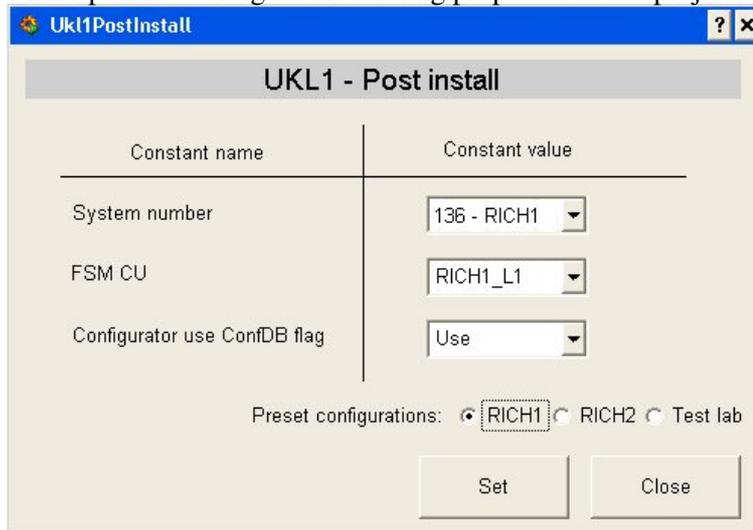- `java/jar/` Java applications

Here is a summary of a few existing tools:

- `ukl1-init` - performs hardware initialisation and is required whenever the UKL1 board is powered (also done at CCPC boot time).
- `ukl1-cfg` - configures the UKL1 board. Formerly `ukl1cfg`.
- `ukl1-status` - dumps UKL1 status or configuration data. Formerly `showStatus`.
- `mdfReader` - reads and summarises the content of MDF files.
- `ukl1-reload` - trigger reload of UKL1 FPGAs. Formerly `reloadL1`.
- `ukl1-reset` - pulse L1 reset signal. Formerly `resetL1`.
- `ukl1-rw` - read or write registers in the UKL1 egress FPGA. Formerly `l1fpga`.
- `ukl1-gbereset` - reset GBE card. Formerly `resetGBE`.
- `ukl1-gbeflush` - flush GBE buffers. Formerly `gbeFlush`.
- `ukl1-gbestatus` - dump some GBE status. Formerly `gbeStatus`.
- `ukl1-datacapture` - capture L1 data from raw socket.
- `pbw.jar` - An event display that can read non-compressed MDF files.

Note the new rationalised names for UKL1 utilities.

-- Steve Wotton - 01 Apr 2007

- UKL1 panel to configure the running properties of the project.: