NA62 RunControl

Nicolas Lurkin

January 27, 2014

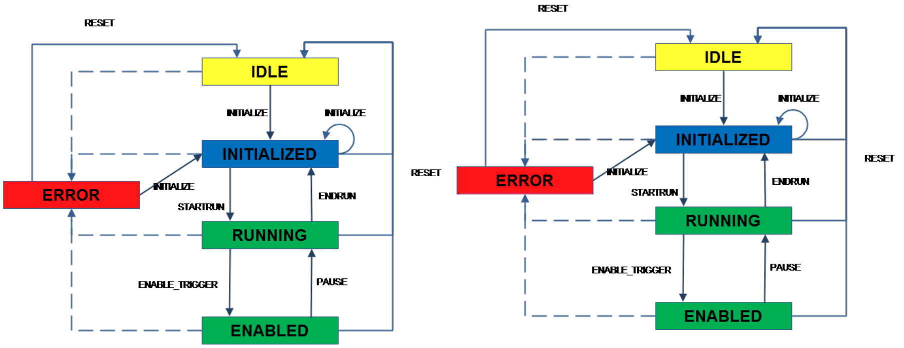
1 Finite State Machine

THe NA62 RunControl is based on a three levels tree-like hierarchy of Finite State Machine (FSM).

Each constituting element of the DAQ (device) is internally modeled as an FSM. Each state of the FSM is defined by the evaluation of logical expressions depending on a set of parameters that are provided by the device. The figure 1a shows the FSM diagram followed by most of the devices.

The device nodes are forming the leaves of the tree that are grouping the devices into logical entities representing subsytems of the experiment. These nodes are also modeled as FSM, summarizing the states of the devices belonging to this group according to a set of rules.

Finally the root of the tree is an FSM node that represents the global state of the Data Acquisition by further summarizing the states of all the logical nodes. The figure 1b shows the FSM diagram for this root node which is derived from the device FSM. The possible states are described below:



(a) Generic FSM diagram for the devices.

(b) FSM diagram of the root node representing the global state of the NA62 DAQ

- **IDLE**: This is the initial state after starting or resetting the FSM and the devices.
- **INITIALIZED**: When all the devices have been configured and the DAQ is ready to take data.
- **RUNNING**: All the devices are completely running and waiting for triggers. The only exception being the trigger processor, in a paused state, waiting for further command to generate the triggers.

- **ENABLED**: The trigger processor is out of the paused state and running.
- **ERROR**: This state can be reached from any other state whenever a problem occurs on any device.

Each state allows a list of abstract commands that are propagated downward in the hierarchy to the device nodes where it is transmitted through the network. The list of commands is described hereafter:

- **INITIALIZE**: Request to initialize all the devices with a specific configuration.
- **STARTRUN**: Request to all the devices to start the run and move in a ready state where they are able to take data.
- **ENABLE_TRIGGER**: Request to the trigger processor to start generating triggers.
- **PAUSE**: Request to the trigger processor to stop generating triggers.
- **ENDRUN**: Request to all devices to stop taking data and end the current run.
- **RECOVER**:
- **RESET**: Immediately move to an idle/initialized state, stopping the current run if it was ongoing.

2 Interface

The link between the device nodes modelling a specific device and the hardware itself is established by DIM[1][2] that will take care of transmitting the commands from the RunControl to the device and the state parameters from the device to the RunControl, along with any other relevant information that should be known by the RunControl or made availabe to the shifters.

As the RunControl is not aware of the internal operation of any device the commands are very generic and the device is expected to understand it and execute the appropriate sequence of actions specific to itself. After the execution of the associated action the devices should answer back to the RunControl, notifying the success or failure of the action.

The minimum set of commands to be understood and implemented is the following:

-

The state changes are triggered
(currently PC farm node, L0TP)

3 Configuration

The configuration mechanism of the RunControl will take advantage of the existing recipe mechanism of the JCOP[3] framework. The configuration of the different devices is done through configuration files. The content of the files will be written in the JCOP database and associated to recipes.

[1] Distributed Information Management System

[2] TODO:link to dim

[3] TODO:Link to JCOP

4 Logging