RC Status, Issues, Changes

RC Summary @ TB 2017

- PXD (ONSEN, DATCON, later DHC) included
  - sub-system and single board can be in/excluded individually
  - DHC connection check and reset sequence included in RC during TB → fully working in last week
- IOCs completely stable, restarts only for update reasons (DHC)
- Stable (after shifter training → ABORT on error, don't “just” stop)
- PXDRC recognizes and handles missing ONSENRC, DATCONRC and DHCRC correctly. ONSENRC recognizes and handles missing board RCs correctly.

RC Summary – Issues

- Interface definitions between NSM (Master RC/DAQ) and EPICS RC not complete/completely documented
  - NSM RC did expect intermediate states which are not reachable from some states (this could be solved within minutes by a f2f discussion)
  - “expected behavior” not defined/documented
  - Better definitions on the way (→ confluence discussion)
- “Hard” reset of IOC (on FPGA) seem to have long recover time
  - EPICS feature, TCP/IP timeout
  - As long as IOC is missing, RC cannot change system state correctly
    - Request go to nirvana.
  - no interface defined to report missing boards to global RC (→ no shifter feedback in global screen)
- “Recovery” confusing for shifter → will be removed

RC Summary – Issues

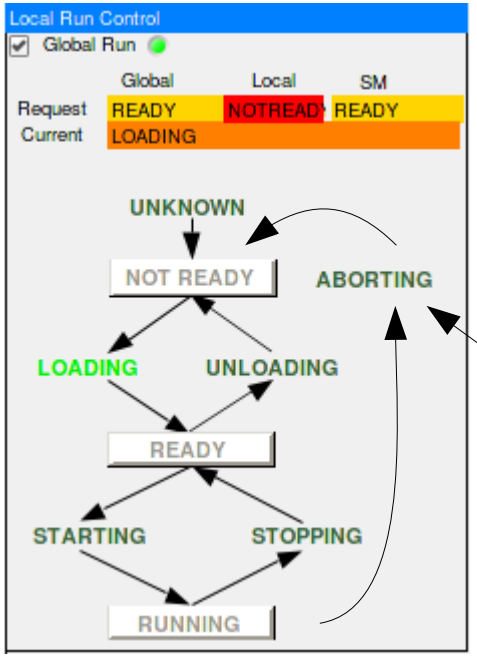
- Testing in dry run system (November) not possible because NSM RC/bridge not available until ~mid January
  - Testing of PXDRC – NSM interface not possible
- Found Problems:
  - Unexpected returned states not accepted by (hidden) NSM-PXDIOC (NSM gateway)
    - Example: At least the LOADING state is missing error checks
  - UNLOADING (EPICS) was mapped to ABORTING (nsm) (which is o.k.) but ABORTING was NOT mapped (might be introduced by ad-hoc change during TB)

RC ↔ Firmware Issues (Low Level)

- (see talk by Klemens Lautenbach)
- Rework activation + deactivation of links (TCP/IP and Aurora) to prevent race conditions between sub-systems and disconnected TCP/IP connections (timeouts).
- Same issue within ONSEN (between Merger and Selector) – was (mostly?) not detected because of 1s sampling time >> link down time.

Discussion since TB

GUI Interface (no significant change)



only on „NOTREADY“ request

ERROR

any state can go to error

Discussion points:

Graphical overview for returned state depening on internal state and sub-system state:

SM state	Allowed/Forbidden Subsystem states											
	UNKNOWN	NOREADY	LOAD	UNLOAD	READY	START	STOP	RUNNING	ABORT	ERROR	If not return	If not change state
UNKNOWN	yes	yes	yes	yes	yes	yes	yes	yes	yes	no	ERROR	
NOTREADY	no	yes	maybe	maybe	maybe	no	no	no	no	no	UNKNOWN/ERROR	UNKNOWN?
LOADING	no	yes	yes	no	yes	no	[illegible]	no	no	no	ERROR	
UNLOADING	no	yes	yes	yes	yes	no	[illegible]	no	no	no	ERROR	
READY	no	no	no	no	yes	no	no	no	no	no	ERROR	
START	no	no	no	no	yes	yes	no	yes	no	no	ERROR	
STOP	no	no	no	no	[illegible]	yes	yes	yes	[illegible]	no	ERROR	
RUNNING	no	no	no	no	no	no	no	yes	no	no	ERROR	
ABORT	yes	yes	yes	yes	yes	yes	yes	yes	yes	no	ERROR	
(ERROR)	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	---	

- UNKNOWN and DISCONNECTED same time
- Allow system to be ready in advance, advantages/disadvantages??
- ERROR can be a SM state, but must not. Example: in my mother, network can come from hardware int.
- ERROR is raised from aggregated state of local and daugther SM states. The local will/may(?) advance to ABORT in case of ERROR (or, stay in its current s

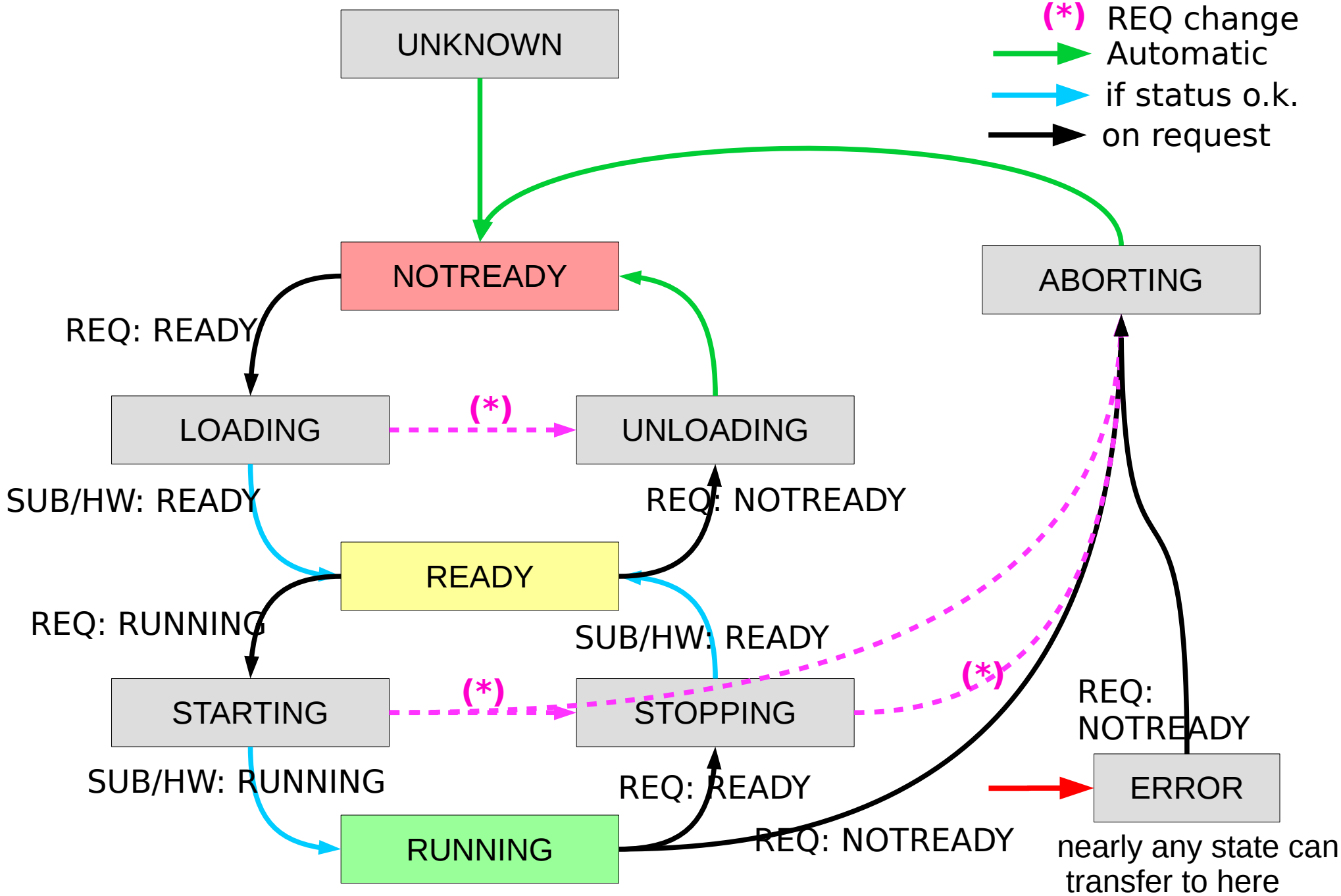
To discuss: more defined state changes including error state

from	to	intermediate state	hardware !OK
UNKNOWN	NOTREADY	automatic	automatic
NOTREADY	LOAD	target READY	target READY
LOAD	READY	all subsys are READY	hardware !READY cond
LOAD	UNLOAD	target NOTREADY	target NOTREADY
LOAD	ERROR	---	hardware load failed
UNLOAD	NOTREADY	automatic	automatic
UNLOAD	ERROR	---	hardware unload failed
READY	UNLOAD	target NOTREADY	target NOTREADY
READY	ERROR	---	hardware !READY cond
READY	START	target RUNNING	target RUNNING
START	STOP	target READY	target READY

Documentation on Confluence; DISCUSSION POINTS (before, during, after TB)

https://confluence.desy.de/pages/viewpage.action?pageId=47811977

States and Transitions



States and Transitions in PXDRC and ONSENRC

from	to	transition if
UNKNOWN	NOTREADY	automatic
NOTREADY	LOAD	target **READY**
LOAD	READY	all subsys are READY
LOAD	UNLOAD	target **NOTREADY**
LOAD	ERROR	Any SubSys in invalid state
UNLOAD	NOTREADY	automatic
UNLOAD	ERROR	Any SubSys in invalid state
READY	UNLOAD	target **NOTREADY**
READY	ERROR	Any SubSys in invalid state
READY	START	target **RUNNING**
START	STOP	target **READY**
START	RUNNING	all subsys are RUNNING
START	ERROR	Any SubSys in invalid state
START	ABORT	target **NOTREADY**
STOP	READY	all subsys are READY
STOP	ERROR	Any SubSys in invalid state
STOP	ABORT	target **NOTREADY**
RUNNING	STOP	target **READY**
RUNNING	ABORT	target **NOTREADY**
RUNNING	ERROR	Any SubSys in invalid state
ERROR	ABORT	target **NOTREADY**
ABORT	NOTREADY	automatic
ABORT	ERROR	---
NOTREADY	DISCONNECTED	??
DISCONNECTED	NOTREADY	??

Remark: Disconnect in any other state than NOTREADY (+UNKNOWN) is an ERROR. (how about ABORT?)

Does not make sence

Depends how global RC handle this

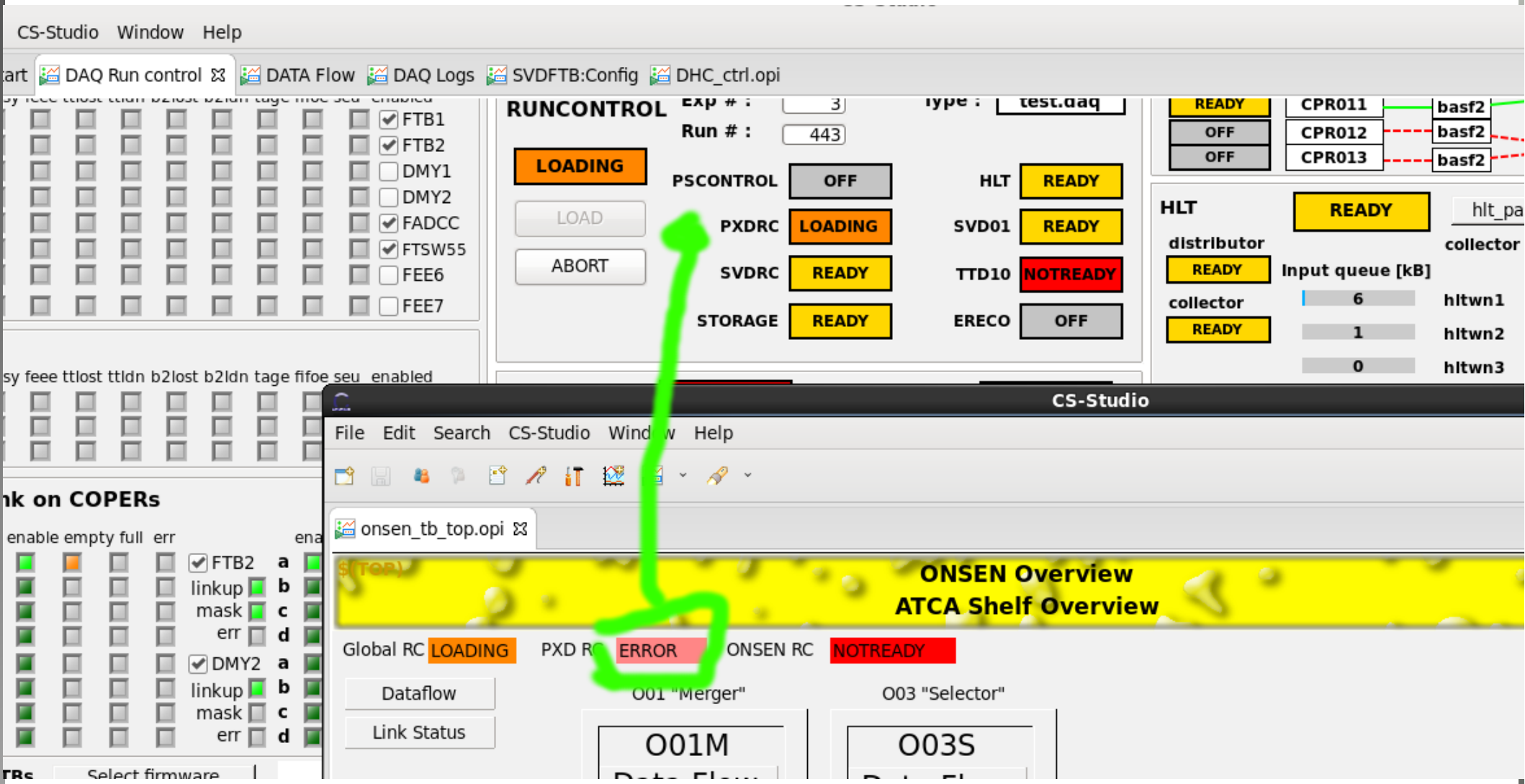
Backup

Invalid States in PXDRC and ONSENRC

SM state	Allowed/Forbidden Subsystem states										
	UNKNOWN	NOTREADY	LOADING	UNLOADING	READY	START	STOP	RUNNING	ABORT	ERROR	**if not change state**
UNKNOWN										XXXXX	ERROR
NOTREADY	XXXXX		XXXXX	XXXXX	XXXX	XXXXX	XXXXXXXXXX		XXXXX	XXXXX	(UNKNOWN/ERROR)
LOADING											ERROR
UNLOADING											ERROR
READY											ERROR
START											ERROR
STOP											ERROR
RUNNING											ERROR
ABORT										->Loop	(ERROR)

DISCONNECT – is not implemented, ongoing discussion
XXXXX – check not implemented, because
(a) unclear what transition should go to (ERROR?)
(b) glitch because any transitions to NOTREADY is automatic without check
(c) ERROR might not be permanent and not requiring abort (glitch or DISCONNECT...)

NSM Gateway: Hidden Local State



RC Tasks

- Run Control is used to coordinate/synchronize Loading/Configuring, Starting, Running Stopping, Aborting for all sub-systems.
- Run control is steering / collecting status information from major systems. The system themselves have to implement the steering of all sub-parts.
- Tree structure
  - Main NSM RC
    - SVD
      - ...
    - PXD
      - DATCON
        - all DATCON boards
      - ONSEN
        - all ONSEN boards
      - (DHC)
        - all DHC boards
    - other detectors
    - DAQ, TRIGGER, HLT, ...

RC Example for ONSEN Branch

RunControl overview:

(DHCRC)
DATCONRC

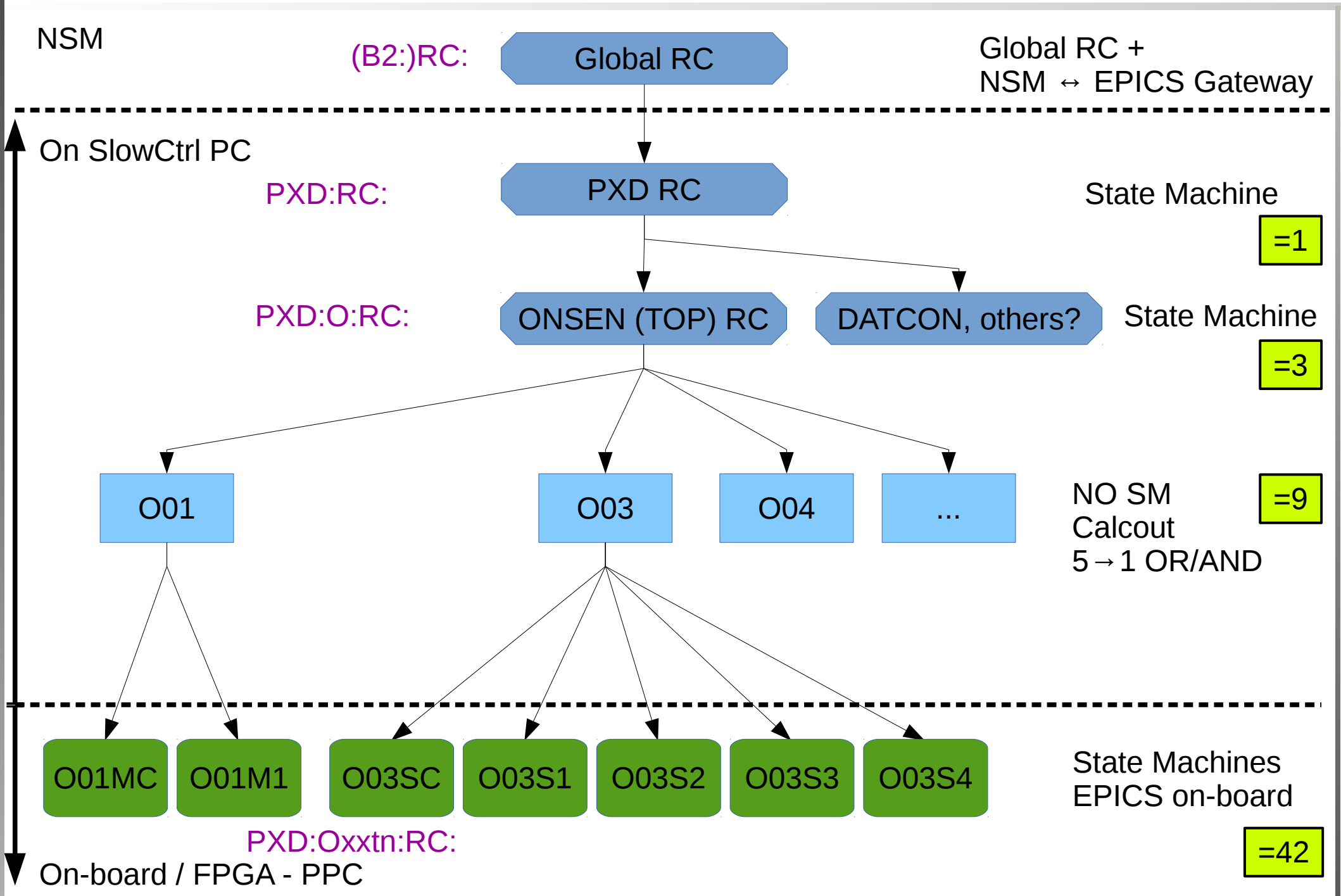
NSM ↔ NSM-PXDIOC ↔ PXDRC ↔ ONSENRC ↔ ONSENHW ↔ FPGA Cores
(„gateway“) (on board)

KEK	Mainz	Giessen

Interface defined, but not
the expected behaviour (hidden)

- PXDRC and ONSENRC take the status of their sub-systems and decide the overall status from this. Unexpected states result in ERROR
- PXDRC and ONSENRC propagate the request to their sub-systems.
- ONSENHW get the status from the hardware (PVs) and calculate the single board state from that. Requests/state transitions call a program which do the actual hardware interaction.
- An additional SM manages the resets or DHC (as long as under RC control). For DATCON there is no interface to the hardware defined yet. RC just reports the status of the hardware bus cannot interact.

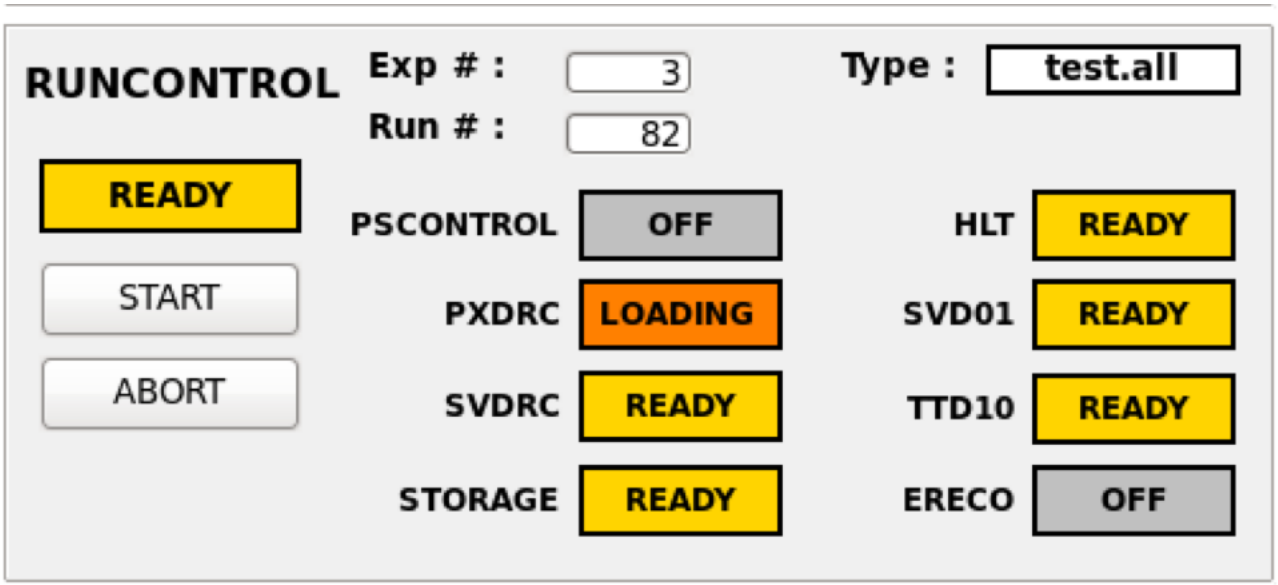
PXDRC Tree


NSM
(B2:)RC:
Global RC
Global RC +
NSM ↔ EPICS Gateway
On SlowCtrl PC
PXD:RC:
PXD RC
State Machine
=1
PXD:O:RC:
ONSEN (TOP) RC
DATCON, others?
State Machine
=3
O01
O03
O04
...
NO SM
Calcout
5→1 OR/AND
=9
O01MC
O01M1
O03SC
O03S1
O03S2
O03S3
O03S4
State Machines
EPICS on-board
PXD:Oxxtn:RC:
=42
On-board / FPGA - PPC

Backup

Invalid State (by Recovery?)

„invalid“ state …
How can global RC state READY while not all systems are ready.



(Have some idea, maybe operator made a mistake, and clicked LOAD on the EXCLUDED PXD).
But then the GUI should prevent that!
Presentation should be clearer.

<u>or</u>

PXD (or ONSEN) aborted while in READY and RC missed it…
(ABORT in LOADINg where its not checked for?)

„so called“ Run Control problems (no joking!)

- Some systems need to be re-initialized each run.
  - This was called a Run Control problem.
- ONSEN stayed in „Loading“ state because EB or HLT did not connect
  - This was called a Run Control problem, because the RC did not switch ONSEN to READY
- Some link status (EB?) was not displayed correctly
  - Called RC issue
- Inactive links (EB?) was showing data rate
  - “RC is wrong”
- IOC on ONSEN boards took a long time to connect to slow control after a board reset
  - RC shows disconnected → RC problem