LMU München - Excellence Cluster Universe

Report on Firmware Project

Stefan Rummel

Overview

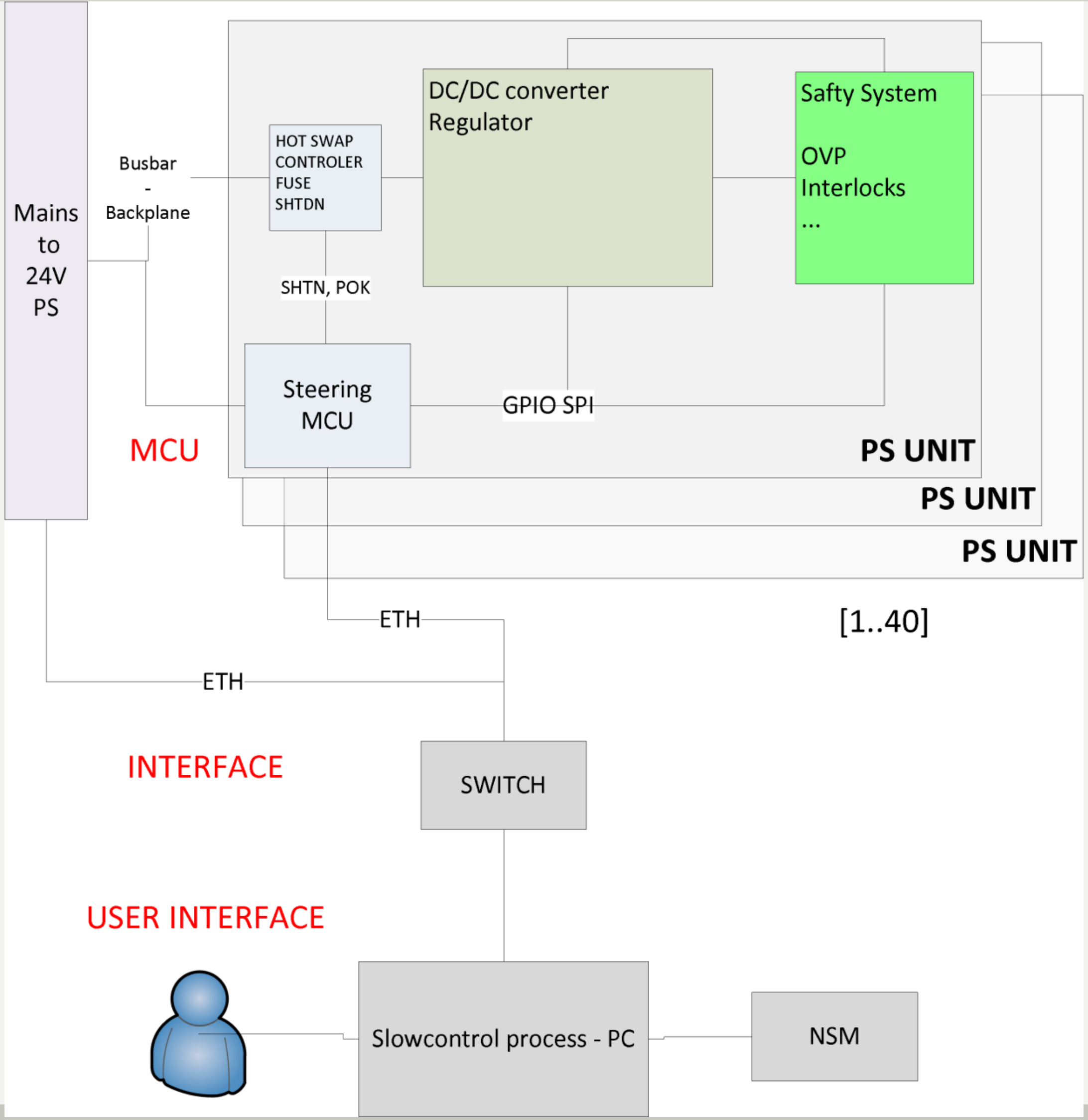
- Project overview
- Concept & functionality
    - User interface
    - MCU
    - Interface
    - Slow control

Project overview

- Via Fortiss gGmbH – see presentations in Bonn
- Local project leader Dr. Buckl, implementation Dipl. Inf. Mr. Barner
- Current status:
  - Monthly meetings
  - FMEA – "What can go wrong in the PS and what is the impact on the detector?"
  - Collection of requirements there
  - MCU selected (ARM M3, ST connectivity series)

System overview

- User Interface
- Slow control process & logging via NSM
- Interface layer via ETH
- Steering via Microcontroller



Concept User Interface

Assumption: Typical user/shifty is NOT a DEPFET expert

→ Critical things must be handled automatically

- Start Stop procedures
- Error Handling (Over Current, Regulator failures .....)
- Shutdown in case of problems
- Retry of Start procedure on user-request

→ Expert mode with full control

- Change Voltages, Current Limits, ...

Functionality – MCU level

- Implementation – changes should be possible with free tools
  - GNU tool chain
  - Free RTOS – deterministic behavior
  - Boot loader via ETH (JTAG as backup)
  - Documented and commented code

- Startup- Shutdown sequence / (ramping up/down voltages)
  - Readout (DHP, DCD, SW_DVDD)
  - Confirm Switcher Configuration
  - Steering Voltages
  - Source/HV

- Periodic monitoring of status (voltages, current …)

- Nominal parameters stored locally

  → Periodic checking

- Reporting of status (Voltages, Currents, Status Bits, Secondary Voltages, Temperature)

Functionality – MCU level

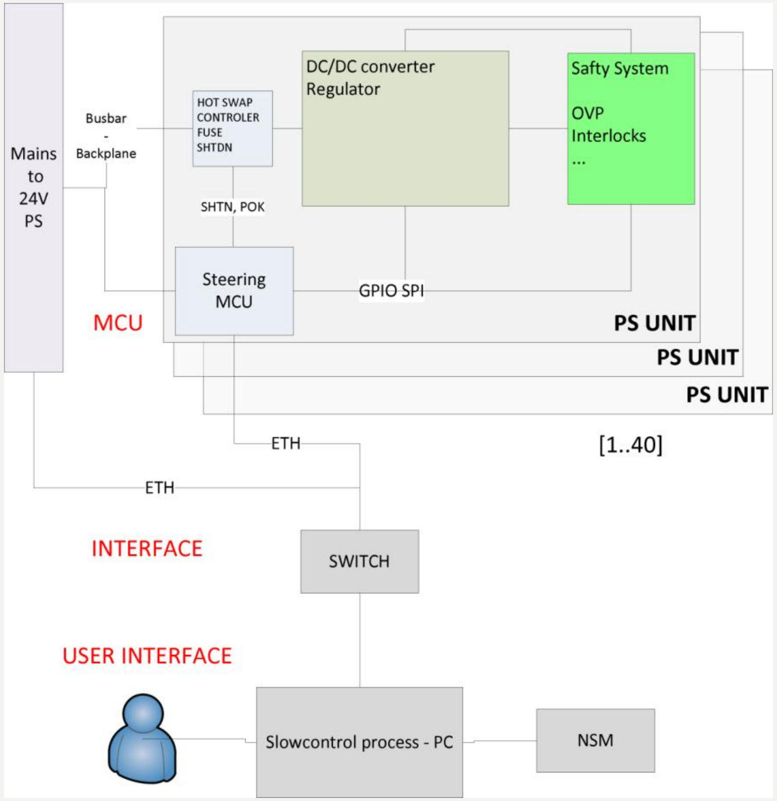
- Self tests @ Startup / Running
  - ADC (sanity check of code)
  - DAC (via integrated MUX, read back of registers)
  - Current limit amplifier (status bit @ current limit 0A, small test voltage)
  - OVP & Shutdowns (shutdown functionality via voltage)
  - Check secondary voltages and current (per 4 Channel module)
- MCU watchdog via edge triggered periodic reset – initiates hardware shutdown in case of MCU failure
- Interlock for cooling (proposal: TTL COOLING_IS_ALIVE signal)
- Reasonable behavior in case of connection loss to Slow Control

Functionality – Interface

- Interface between MCU's and Slow Control done via Ethernet
- Status messages send via UDP
- Commands TCP/IP
- Supports broadcasts e.g. Start, Stop, Calibration of steering voltages (Gate_ON, CCG, Clear_Low…)

Functionality – Slow Control

- Receiving status messages
- Send / Broadcast of commands
  - Start, shutdown, calibration of steering voltages
  - Adapt voltages and current limits
- User frontend
  - Visualization of PS-system
    - System status
    - Module status
      - Voltages, Currents
      - Secondary Voltages
      - Temperatures (MCU, sel. Voltages)
      - Status Bits (Regulator, OVP)
      - Error messages
  - Different levels of control – Expert / User
- Interface to
  - BELLE II slow control - NSM for logging
  - Interface to DAQ – to receive status of Switcher configuration – TBD
  - Cooling plant



Summary

- Firmware effort has started
- Collection of requirements is there – still open for changes - contributions

- Timeline
  - Implementation till next year than debugging/testing phase
  - Implementation of Slow Control starts in parallel
  - Next steps – next month – first steps with evaluation boards (MCU, ADC, DAC)
  - Late summer test with actual hardware