Remotely Setting up MicroTCA Systems

Through the Example of TARLA LLRF Control

Patrick Nonn
Hamburg, 12.12.2024

HELMHOLTZ

DESY.

Motivation

Why set up MicroTCA Systems remotely?

- Shipping hardware around:
  - Is expensive
  - Takes time
  - Risks damage
- Adaption and bug fixing “in situ”
  - Alternative: Reference System at DESY
- Overall: Less headaches for all involved

Prerequisites

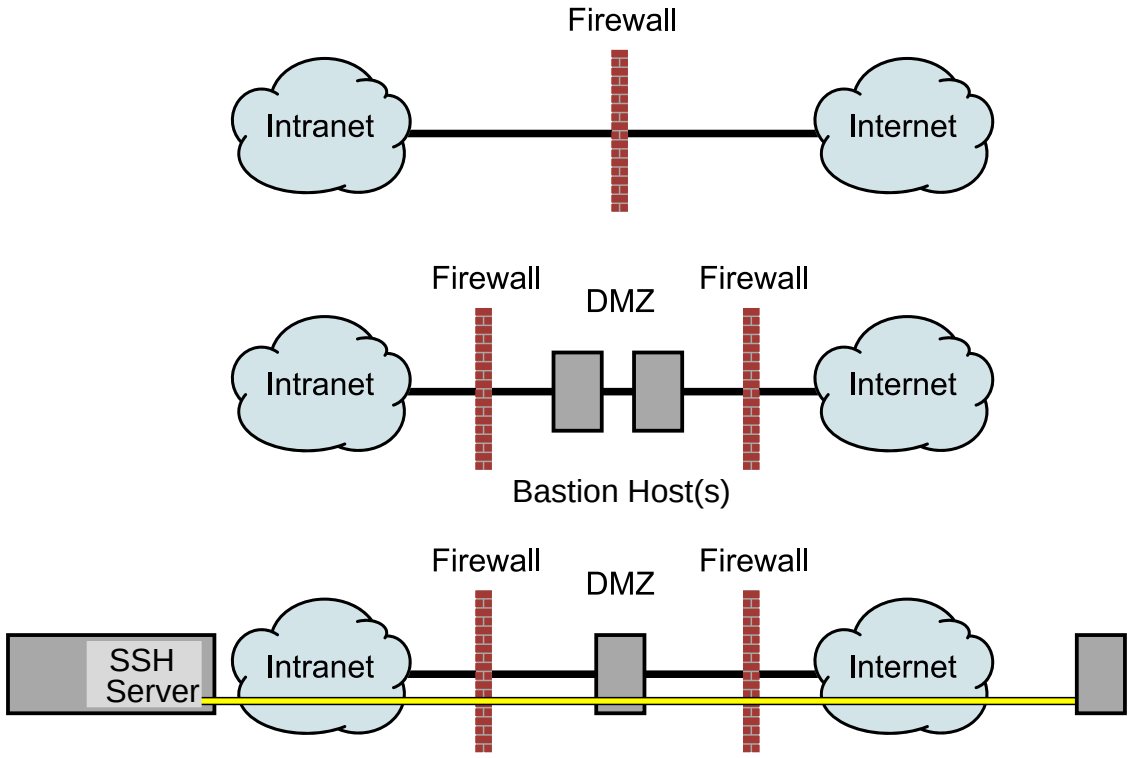
What can not be done remotely!

- MicroTCA crate assembled and powered
- MCH: network settings configured (might require serial interface)
  - DHCP or Static IP
- MCH and CPU connected to Ethernet and ping-able
- Correct OS installed on CPU
  - For DESY-supported systems: Ubuntu 24.04 noble numbat
- SSH server installed, configured and running

Remote Access

How to get to the MicroTCA crate

- Control system networks are usually not exposed to the internet:
  - Firewalls
  - Demilitarized Zone
- Thus some measures are required, to get access:
  - SSH Jumping
  - Virtual Private Network




```
ssh -j user@bastion.fqdn.edu:port user@host:port
```

Preparation

- Package manager
  - Add DESY repository to sources
  - If necessary, configure proxy for package manager
- If required: Install DOOCS infrastructure
  - LDAP-ENS server, doocstools, doocs-scripts
  - Watchdog
  - Servers (e.g. x2timer-server)

Example for apt:

```
user@host:~$ wget -O - https://doocs-web.desy.de/pub/doocs/DOOCS-key.gpg.asc | sudo gpg --dearmor -o /etc/apt/trusted.gpg.d/doocs-keyring.gpg
user@host:~$ echo 'deb http://doocs-web.desy.de/pub/doocs focal main' | sudo tee /etc/apt/sources.list.d/doocs.list
user@host:~$ echo 'Acquire::http::Proxy "http://url.of.proxy:port";' | sudo tee /etc/apt/apt.conf.d/90proxy
user@host:~$ sudo apt update
```

Firmware

- MSKs firmware delivery server (Jenkins) not accessible from outside DESY
  - Firmware installation via deployment tool only usable from inside DESY
- If current firmware from MSK framework (fast):
  - Installation via PCIe (fast)
  - Requires PCIe endpoint and the debian package: *mtca4u-fw-programmer*

```
user@host:~$ mtca4u_fw_programmer -pv -i spi -D <path/to/dmap/file> -d <devicename_in_dmap_file> -f <path/to/bit-file>
```

```
slot_6  (pci:pcieunis6?map=<path/to/mapp-file>)
slot_7  (pci:pcieunis7?map=<path/to/mapp-file>)
slot_8  (pci:pcieunis8?map=<path/to/mapp-file>)
slot_9  (pci:pcieunis9?map=<path/to/mapp-file>)
slot_10 (pci:pcieunis10?map=<path/to/mapp-file>)
slot_11 (pci:pcieunis11?map=<path/to/mapp-file>)
```

- If no firmware installed or not from framework (slow):
  - Installation via Hardware Platform Management (HPM)
  - Requires package ipmitool
  - Bit-file needs to be converted to hpm: https://github.com/MicroTCA-Tech-Lab/bin2hpm

```
user@host:~$ ipmitool -H <mch_host> -P "" -B 0 -b 7 -T 0x82 -t <ipmb_addr> hpm check

PICMG HPM.1 Upgrade Agent 1.0.9:

-------Target Information-------
Device Id          : 0x0
Device Revision    : 0x80
Product Id         : 0x8302
Manufacturer Id    : 0x92bd (Unknown (0x92BD))


--------------------------------------------------------------------------------
|ID  | Name         |                     Versions                           |
|    |              |     Active      |     Backup      |      Deferred     |
--------------------------------------------------------------------------------
|   0|ATXMEGA MMC   |   2.40 00000000 | ---.-- -------- | ---.-- -------- |
|   1|FPGA FLASH    |   0.00 EE010000 | ---.-- -------- | ---.-- -------- |
|   2|FPGA FLASH2   |   0.00 EE020000 | ---.-- -------- | ---.-- -------- |
--------------------------------------------------------------------------------
(*) Component requires Payload Cold Reset
```

Software

- Which control system is used (EPICS, TANGO, OPC-UA)
- Which servers are needed
  - Server might need to be build for control system
  - Configuration for each server instance needs to be prepared/maintained
- Servers and configurations are published as Debian packages in DESY’s public repository
- For testing purposes (or in a pinch) configurations might be “installed” via scp

*T*urkish *A*ccelerator and *R*adiation *L*aboratory in *A*nkara (TARLA)

Located south of Ankara

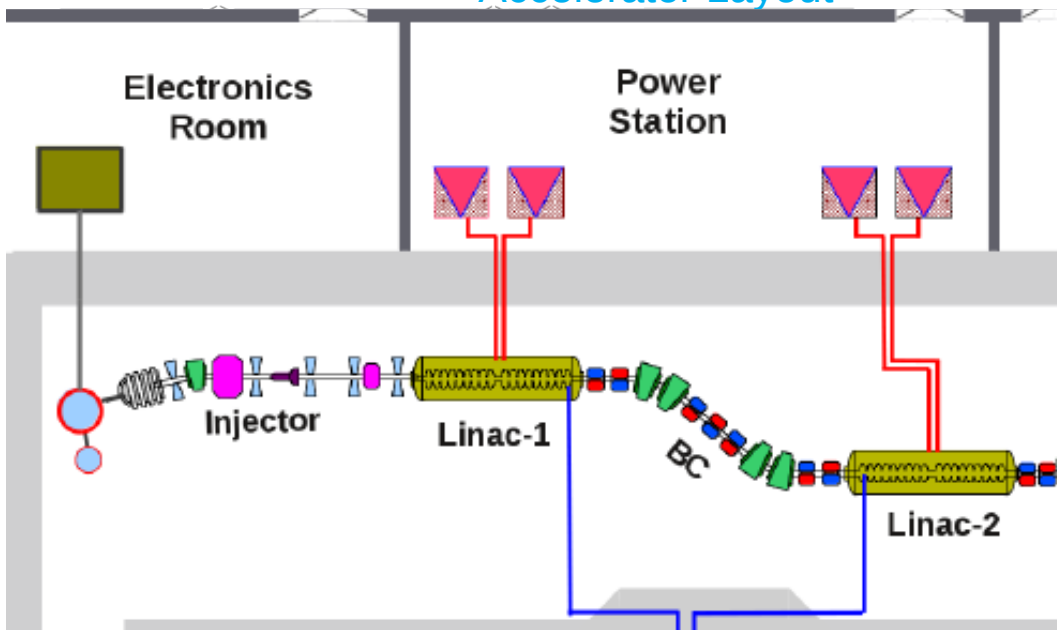
Facility site

- At the beginning of the project part of the Ankara university, still located at the campus

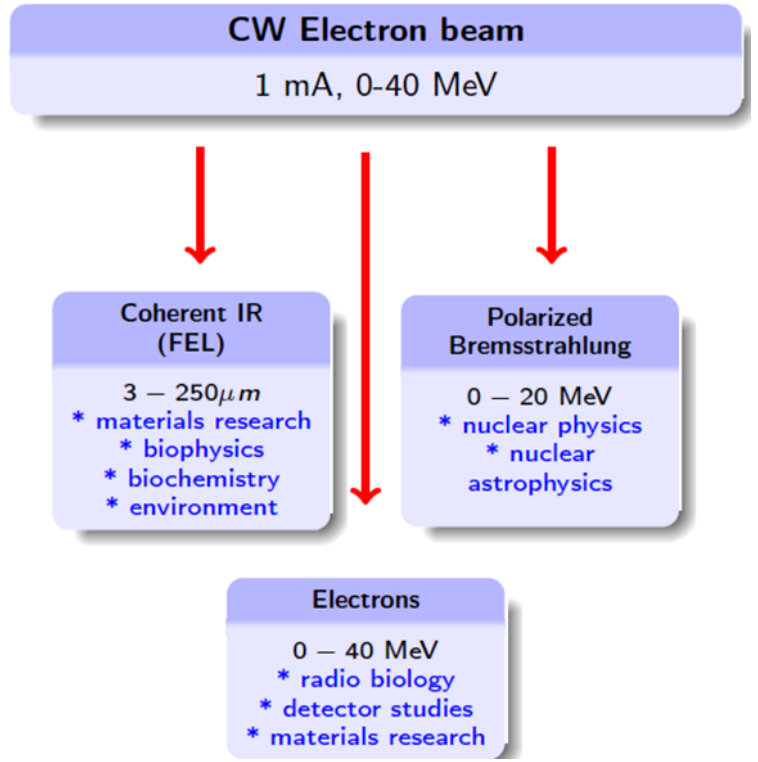
Main facility objectives

- Free Electron Laser between 3 - 250 nm using 15 - 40 MeV electron beam
  - 2 different optical resonators with 2.5 and 9 cm period length undulators
- Additional beam applications planned
  - Material and detector research
  - Bremsstrahlung for nuclear structure and gamma radiation studies

Accelerator Layout

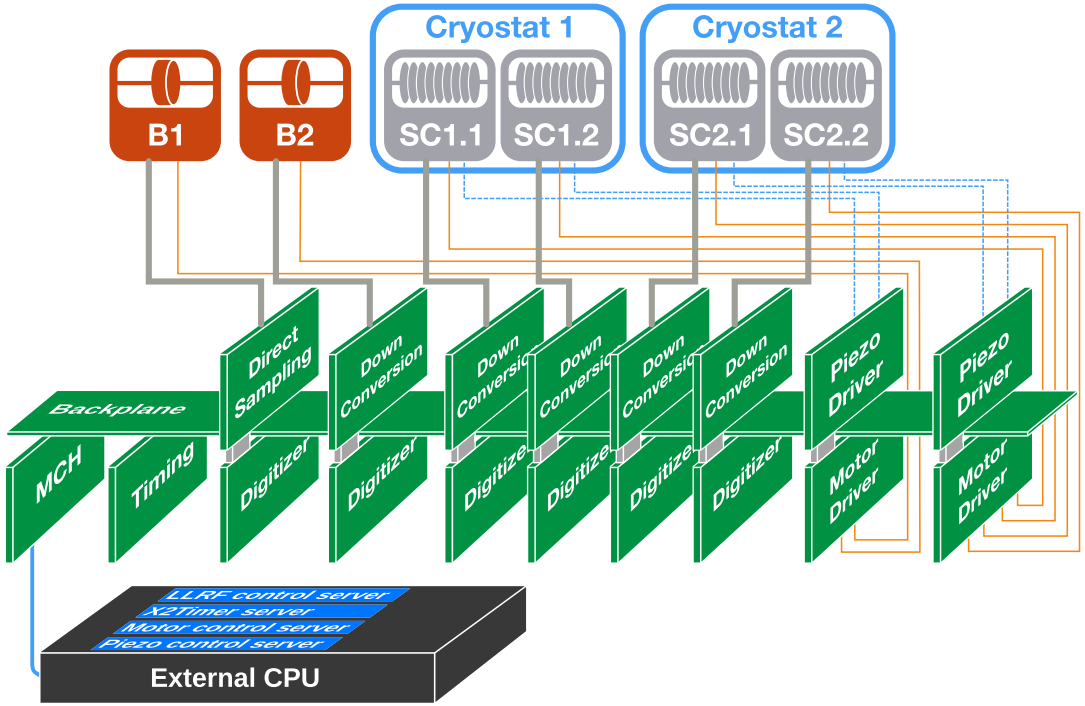


Very much comparable to the ELBE facility at HZDR



LLRF control system at TARLA

- Based on MicroTCA.4, single cavity control (no vector sum)
- Two normal conducting cavities for bunching
  - One 260 MHz sub-harmonic, one 1.3 GHz
- Four Tesla-type superconducting 9-cell cavities



- Master Oscillator
- LO-Generation Module
- LLRF crate
- External CPU
- Power Supply for LOGM
- EPICS-based LLRF control server

Remote Setup at TARLA

Firmware

- Three different firmware builds are needed:
  - Normal conducting, 260 MHz
  - Normal conducting, 1.3 GHz
  - Superconducting, 1.3 GHz
- Used HPM to restore flash-image after accidental mis-programming

Software

- First LLRF system on Ubuntu 24.04
- EPICS: 6 llrfctrl-server instances, 1 motor controller (new)
- DOOCS: x2timer-server, watchdog
- VNC-server for remote access

Thank you

Contact

Deutsches Elektronen-
Synchrotron DESY

www.desy.de

Patrick Nonn
MSK
patrick.nonn@desy.de
040 8998 1962