

## Master's Thesis

# Entwicklung des Detektorkontrollsystems für den ITk Outer Barrel Demonstrator des ATLAS Experiments

# Development of the Detector Control System of the ITk Outer Barrel demonstrator for the ATLAS Experiment

prepared by

**Hans Joos**

from Potsdam

at the II. Physikalischen Institut

**Thesis number:** II.Physik-UniGö-MSc-2022/01

**Thesis period:** 18th February 2021 until 18th February 2022

**First referee:** Prof. Dr. Stan Lai

**Second referee:** Priv.Doiz. Dr. Jörn Große-Knetter



# Zusammenfassung

Für das Upgrade des LHC zum High-Luminosity-LHC wird der ATLAS Spurdetektor durch einen reinen Siliziumdetektor, dem Inner Tracker (ITk) mit einem Pixeldetektor und einem Streifendetektor, ersetzt. Die gestiegene Luminosität erfordert strahlungsfeste Komponenten, die mit der höheren Belegungsdichte und dem größeren Teilchenfluss zurechtkommen. Durch die große Nähe zum Kollisionspunkt ist die Umgebung besonders für den Pixeldetektor mit seinen 3D und planaren Sensoren fordernd. Für eine schnellere und zuverlässigere Auslese der Sensoren wurde ein neuer Auslese-Chip entwickelt. In der endgültigen Konstruktion werden die neuen Pixelmodule mit den Auslese-Chips in einer Reihenschaltung mit Strom versorgt und auf einer Stützstruktur montiert. Im zentralen Bereich (OB) des ITk Pixeldetektors gibt es zwei Arten von Stützstrukturen: die Longerons und die Inclined Halfrings. Um die Systemfunktionalität der Module in Reihenschaltung zu testen, wird ein ITk OB Demonstrator mit mindestens zwei Modulreihen aufgebaut.

Für den Betrieb des Demonstrators wird ein Detektorkontrollsystem benötigt. Zu diesem Zweck wurde im Rahmen dieser Arbeit ein SCADA-System, basierend auf der WinCC OA-Software, erstellt. Die Benutzeroberflächen und Überwachungsinstrumente, die im Zusammenhang mit einem WinCC OA-Projekt entwickelt wurden, um ein ausgereiftes Kontrollsystem für den ITk OB Demonstrator zur Verfügung zu stellen, werden beschrieben. Außerdem wird die Modellierung des Kontrollsystems als Zustandsmaschine diskutiert.

## Abstract

For the upgrade of the LHC to the High-Luminosity-LHC, the ATLAS tracking detector will be replaced with a pure silicon detector, the Inner Tracker (ITk) with a Pixel detector and a Strip detector. The higher luminosity requires radiation hard components that can deal with higher occupancies and fluences. Given the close proximity to the interaction point, the environment is especially challenging for the Pixel detector, which is planned to feature 3D and planar sensors and a new readout chip, allowing for a faster and more reliable readout of the sensors. In the final design, the new ITk Pixel modules with the readout chips will be combined into serially powered (SP) chains and mounted on Loaded Local Supports. In the Outer Barrel (OB) region of the ITk Pixel detector, there are two types of Loaded Local Supports: the longerons and the inclined half rings. To test the system functionalities of the modules of the SP-chains, an ITk OB demonstrator consisting of at least two SP-chains and aiming to mimic the real detector, will be set up.

To operate the demonstrator, a Detector Control System (DCS) is necessary. For this, a SCADA system based upon the WinCC OA software architecture was created in the context of this thesis. The user interfaces and monitoring tools that have been developed in the context of the WinCC OA project for the purposes of providing a well-developed DCS for the ITk OB demonstrator are described. In addition, the design of the DCS as a Finite State Machine is also discussed.



# Contents

<b>List of Terms and Acronyms</b>	<b>vii</b>
Glossary . . . . .	vii
Acronyms . . . . .	vii
<b>1. Introduction</b>	<b>1</b>
<b>2. Hadron collider physics at the LHC</b>	<b>5</b>
2.1. The Standard Model of particle physics . . . . .	5
2.2. The Large Hadron Collider . . . . .	7
<b>3. The ATLAS detector</b>	<b>13</b>
3.1. Detector systems of the ATLAS detector . . . . .	16
3.2. Inner Tracker and ITk Pixel detector (HL-LHC upgrades) . . . . .	21
3.3. ITk Pixel system tests and OB demonstrator . . . . .	26
3.4. ATLAS Detector Control System . . . . .	27
3.4.1. Hardware . . . . .	29
3.4.2. Protocols . . . . .	30
3.4.3. Software . . . . .	32
3.5. ITk Pixel DCS . . . . .	36
3.5.1. Control and Feedback path . . . . .	36
3.5.2. Safety path . . . . .	37
3.5.3. Diagnostics path . . . . .	38
<b>4. Finite State Machine</b>	<b>39</b>
4.1. Theory of Finite State Machines . . . . .	39
4.2. The State Manager Interface . . . . .	40
4.3. The Controls Hierarchy framework component . . . . .	41
4.4. ATLAS specific extensions . . . . .	43
<b>5. The DCS for the ITk Pixel OB demonstrator</b>	<b>45</b>
5.1. Requirements and the SR1 setup . . . . .	46

## Contents

5.2. System overview and project setup . . . . .	49
5.3. Watchdog scripts . . . . .	52
5.4. Graphical User Interfaces . . . . .	53
5.4.1. SP-chain monitoring and control . . . . .	55
5.4.2. Interlock monitoring . . . . .	59
5.4.3. Cooling monitoring . . . . .	63
5.4.4. Environmental monitoring . . . . .	64
5.4.5. MOPSHUB4Beginners control . . . . .	65
5.4.6. Opto Box monitoring . . . . .	66
5.4.7. Watchdog monitoring . . . . .	66
5.5. FSM control hierarchy . . . . .	70
5.5.1. SP-chain Control Unit . . . . .	72
5.5.2. User Interface . . . . .	76
5.6. Testing and performance considerations . . . . .	79
5.7. Deployment and usage . . . . .	81
<b>6. Conclusion and Outlook</b>	<b>83</b>
6.1. Conclusion . . . . .	83
6.2. Outlook . . . . .	83
<b>A. Connection diagrams</b>	<b>85</b>
<b>B. Resources</b>	<b>87</b>
<b>Bibliography</b>	<b>89</b>

# List of Terms and Acronyms

## Glossary

**bPOL12V** A radiation tolerant DC-DC converter ASIC for nominal 12 V input.

**Controller Area Network** A vehicle bus standard designed to allow microcontrollers and devices to communicate with each other in applications without a host computer. It is a message-based protocol, designed originally for multiplex electrical wiring within automobiles, but is also used in many other contexts like in ATLAS slow control.

**FELIX** Front-End Link Interface eXchange. FELIX is a network switch to convert GBT-like protocol data to Gigabit Ethernet or InfiniBand networking.

**FwFsm** Controls Hierarchy framework component.

**GBT** Gigabit Bidirectional Trigger and Data Link. Generic name of the “Gigabit transceiver”. Several radiation hard chipsets are available.

**Monitoring Of Pixel System** DCS ASIC to monitor parameters of detector modules and Opto Boxes.

**MOPSHUB** A communication interface between the MOPS and the DCS control station in the counting room.

**Opto Board** Carries all opto-electrical components. Up to eight opto boards can be installed in one Opto Box.

**Opto Box** The on-detector location of the Pixel opto-transceivers including ASICs for data handling.

**ShuntLDO** Shunt Low Drop-Out Voltage regulator.

**SP-chain** Group of Pixel detector modules which are serially powered by the same low voltage.

**systemd** System and Service Manager for Linux operating systems.

## **Acronyms**

**ADC** Analog to Digital Converter.

**ALICE** A Large Ion Collider Experiment.

**ASIC** Application Specific Integrated Circuit.

**ATLAS** A Toroidal LHC Apparatus.

**BE** Back-End.

**CAN** Controller Area Network.

**CMS** Compact Muon Solenoid.

**DC** Direct Current.

**DCS** Detector Control System.

**DIM** Distributed Information Manager.

**DIP** Data Interchange Protocol.

**DSS** Detector Safety System.

**ECal** Electromagnetic Calorimeter.

**ELMB** Embedded Local Monitor Board.

**EMCI** Embedded Monitoring and Control Interface.

**EMP** Embedded Monitoring and control Processor.

**EYETS** Extended Year-End Technical Stop.

**FE** Front-End.

**FPGA** Field Programmable Gate Array.

**FSM** Finite State Machine.

**GUI** Graphical User Interface.

**HCal** Hadronic Calorimeter.

**HL-LHC** High Luminosity Large Hadron Collider.

**HV** high voltage.

**IBL** Insertable B-Layer.

**ID** Inner Detector.

**IDE** Integrated Development Environment.

**IMC** Interlock Matrix Crate.

**ITk** Inner Tracker.

**JCOP** Joint Controls Project.

**LAr** Liquid Argon.

**LEP** Large Electron–Positron Collider.

**LHC** Large Hadron Collider.

**LHCb** Large Hadron Collider beauty.

**Linac4** Linear accelerator 4.

**LLS** Loaded Local Support.

**LS** Long Shutdown.

**LV** low voltage.

**MOPS** Monitoring Of Pixel System.

**NTC** Negative Temperature Coefficient Thermistor.

**OB** Outer Barrel.

**OPC-UA** Open Platform Communications Unified Architecture.

**PP** Patch Panel.

**PS** Proton Synchrotron.

**PSB** Proton Synchrotron Booster.

**PSU** Power Supply Unit.

**QCD** quantum chromodynamics.

**QFT** quantum field theory.

**SCADA** Supervisory Control and Data Acquisition.

**SCT** Semiconductor Tracker.

**SM** Standard Model.

**SMI** State Manager Interface.

**SML** State Manager Language.

**SNMP** Simple Network Management Protocol.

**SPS** Super Proton Synchrotron.

**TDAQ** Trigger and Data Acquisition.

**TRT** Transition Radiation Tracker.

**VME** Versa Module Eurocard.

**WinCC OA SIMATIC WinCC** Open Architecture.



# 1. Introduction

Questions such as “What is the world made of?” or “How does the universe work?” have driven people for centuries to invest time and resources into research in the field of physics. In early Greece, the philosopher Leucippus and his pupil Democritus coined the term “atom” (from the Greek word *atomos* meaning “uncuttable”) for the smallest building block of matter. The idea that matter is not a continuum that can be divided arbitrarily into smaller pieces was at first only of philosophical nature. But this idea caught on in science in later centuries, and many atomic models were developed that were able to explain chemical elements, for example. However, with the discovery of the electron, the first subatomic particle, the literal meaning of the word *atom* was proven wrong. The word is still used, but it is now known that the atom consists of, apart from the negatively charged electron, a positively charged nucleus. Scientists later discovered that the nucleus is also divisible and consists of protons and neutrons that also have substructure and consist of so-called quarks.

For the correct description of the physical quantities of the newly discovered subatomic particles and their behavior, a new theory was necessary. Quantum mechanics introduced new concepts such as quantization, the uncertainty principle and the wave-particle duality. It is now the fundamental basis to describe particles at microscopic scale. A theoretical framework that builds on quantum mechanics and also follows the principles of special relativity is quantum field theory (QFT), in which particles are treated as excitations of their respective quantum fields.

One such QFT is the Standard Model (SM) of particle physics. It includes all known elementary particles, i.e. particles without a substructure, and three of the four known fundamental forces. Although it is successful with many precise predictions that were verified in experiments, it fails to answer questions like “Why is there more matter than antimatter?” or “How does dark matter fit into the framework?” There are theories beyond the SM that could explain these phenomena. Particle physicists are interested in probing the SM, exploring further the building blocks and fundamental laws of the universe and finding signs of these new theories.

## 1. Introduction

This exploration is a continuous human effort that has already spanned several decades. To combine resources and bring together scientists from different European countries, the European Organization for Nuclear Research (CERN) was established in 1954. It houses several accelerators and collider experiments on its sites at the border of Switzerland and France. The largest collider at CERN is the Large Hadron Collider (LHC), one of the largest scientific instruments ever built. Since the start of its operation in 2010, a large community of scientists working in fundamental particle physics and high energy physics has been exploring the new energy domain made accessible by the LHC. Among the goals of the LHC are the validation of the SM, the study of known particles and the creation and study of new particles, postulated by theories beyond the Standard Model. One of the greatest success so far was the experimental discovery of the Higgs boson in 2012 by research groups of the ATLAS and CMS experiments [1, 2]. Chapter 2 gives an introduction to the LHC and the corresponding physics phenomena, including a short description of the SM.

To extend the possibilities of new scientific discoveries and to account for radiation damage in current accelerator machinery, the LHC is foreseen to be upgraded to the High Luminosity Large Hadron Collider (HL-LHC) in the late 2020s. The aim is to achieve instantaneous luminosities (rate of collisions) of a factor of five larger than the LHC nominal value [3]. During the HL-LHC installation, also the ATLAS detector will undergo a major upgrade. The Inner Detector (ID) of ATLAS will be replaced by a new all-silicon Inner Tracker (ITk). For the safe operation of the new detector subsystem, a Detector Control System (DCS) has to be developed. For the new detector elements, a comprehensive testing procedure and quality assurance program were prepared. The first tests of several detector modules assembled together into their final form will be carried out in so-called system tests. These large tests require their own DCS, where also experience can be gained for the future development of a DCS for the full ITk. Chapter 3 gives an overview of the current ATLAS detector with its subsystems and the planned ITk with focus on its Pixel detector. It then describes the upgrade efforts currently ongoing and planned with respect to testing of bigger prototypes. An important part and basis for this thesis is the description of the DCS for ATLAS and the plans for the ITk.

In Chapter 4, the concept of Finite State Machines (FSMs) is introduced. A large detector like ATLAS requires a way to conceal all the many details of the different parts of the detector that are not relevant to not overwhelm the operator in the control room. The detector operator has to have useful tools on hand to be able to correctly assess the condition of the detector, bring the detector in any desired operational state, or react

to errors. Modeling the operational parts of the detector hierarchically in an FSM tree achieves this.

The focus of this work is on the DCS that has been developed for the system tests of the ITk OB demonstrator. This includes the setup of infrastructure to allow for communication between hardware and a SCADA system as the controlling software. For easy operation, graphical user interfaces and an FSM hierarchy structure were developed. The entire setup must meet the requirements for the system test in terms of flexibility and scalability. The DCS is presented in Chapter 5. Finally, the results are summarized in Chapter 6.



# 2. Hadron collider physics at the LHC

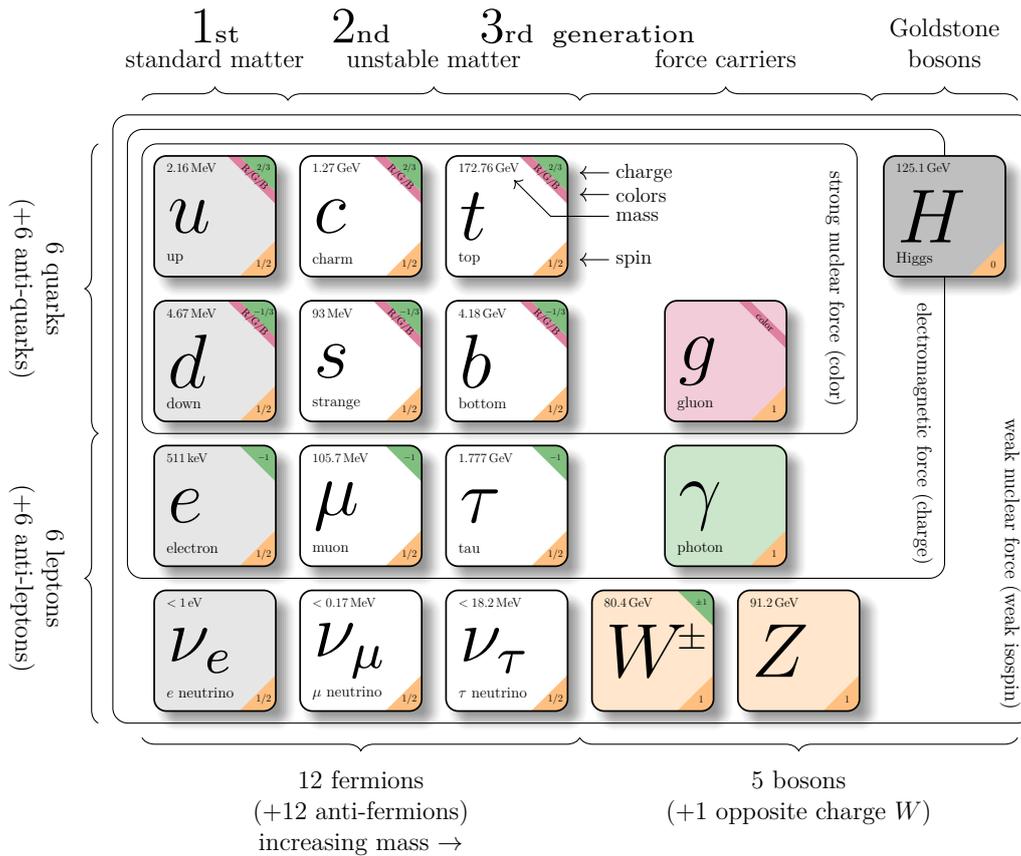
## 2.1. The Standard Model of particle physics

The Standard Model (SM) of particle physics is the theoretical framework which contains all elementary particles discovered so far and also explains the interactions of these particles with each other. It is the basis for all high-energy physics experiments and has proven hugely successful in providing experimental predictions. The elementary particles are classified as fermions if they have half-integer spin, or bosons if they have integer spin quantum numbers. All matter constitutes of fermions, while the bosons are associated with forces. The particle content is summarized in Fig. 2.1. Fermions can be grouped into three generations with an increase in their mass in each subsequent generation. Only the first generation is stable. This means that all particles from the second and third generation will eventually decay into particles from the first generation by radiating off a boson.

Each fermion has an anti-fermion of the same mass but opposite charge associated with it. The fundamental fermions can be grouped into two types, quarks and leptons. Only the quarks ( $u, d, c, s, t, b$ ) can take part in strong interactions. The leptons do not carry color charge and therefore do not experience strong interactions. From the six leptons, three carry an electromagnetic charge: the electron ( $e$ ), the muon ( $\mu$ ) and the tau lepton ( $\tau$ ). Each of these leptons has an associated neutral neutrino. The bosons associated with three of the four fundamental forces are the gluon ( $g$ ) for the strong interaction, the photon ( $\gamma$ ) for the electromagnetic interaction and the  $W^\pm$  and  $Z$  bosons for the weak interaction. The fourth fundamental force, gravity, is not included in the SM.

Mathematically, the SM is realized as a gauge quantum field theory (QFT) with global Poincaré symmetry (as in all relativistic QFTs) and local  $SU(3)_C \times SU(2)_L \times U(1)_Y$  gauge symmetry. The labels of the gauge symmetry describe the different types of interactions and generate the bosonic force carriers. The gauge group  $SU(3)_C$  corresponds to the strong interaction. It has 8 generators which correspond to the 8 types of gluons, the

## 2. Hadron collider physics at the LHC



**Figure 2.1.:** Elementary particles of the Standard Model and their properties<sup>1</sup>. The charges are in units of the elementary charge. (Data is taken from Ref. [4])

force carriers of the strong interaction. The theory that describes this interaction is called quantum chromodynamics (QCD), from the Greek word *chroma* meaning “color”, which is emphasized by the index  $C$ . Color is used to describe the “charge”, to which the gluons couple. Since the gluons themselves carry charge, they can interact with each other. This is a typical property of a non-abelian gauge theory. Another property of QCD is color confinement, meaning that quarks and gluons cannot be isolated, but always appear in color-neutral composites. These composites are called hadrons. If, for example, two quarks were to separate from each other far enough, a new quark-antiquark pair would be created to bind the free quarks. This process is called hadronization. A hadron that consists of two quarks is called meson and a hadron that consists of three quarks is called baryon.

The second major part of the SM is the electroweak theory (sometimes called the Glashow-Weinberg-Salam model [5–7]) as a unification of electromagnetic and weak interaction. The electroweak theory is a gauge theory with local  $SU(2)_L \times U(1)_Y$  symmetry. Here again,

<sup>1</sup>Adapted from graphic by © Carsten Burgard, CC BY 2.5, <https://texample.net/tikz/examples/model-physics/>

the symmetry generates the interaction types and bosonic force carriers. However, under a gauge theory, the gauge bosons are required to be massless. Experimental evidence shows that the weak interaction is short ranged and calls for massive bosons. A solution is provided by spontaneously breaking the electroweak symmetry via the associated Higgs mechanism [8, 9]. The introduced Higgs field has a vacuum expectation value  $v \neq 0$ , and while the Lagrangian is gauge invariant the vacuum state is not and the symmetry is said to be spontaneously broken. The Higgs mechanism also predicts a massive scalar boson, the Higgs boson. This was the last missing particle of the SM and discovered at the ATLAS and CMS experiments in 2012 [1, 2].

Before electroweak symmetry breaking, the generators of  $SU(2)_L$  and  $U(1)_Y$  are given the name “weak isospin” (labeled  $T$ ) and “weak hypercharge” (labeled  $Y$ ) respectively. The index on the  $SU(2)_L$  group indicates that the theory is chiral, and the coupling only occurs for left-handed particles which have a weak isospin of  $T = \frac{1}{2}$ . The generators correspond to the massless gauge bosons for the weak isospin  $W_1$ ,  $W_2$  and  $W_3$  and to the boson for the weak hypercharge  $B$ . The SM bosons are then obtained after symmetry breaking due to the Higgs mechanism. The (massless) photon and (massive)  $Z$  boson emerge as linear combinations of  $W_3$  and  $B$ , while the massive  $W^\pm$  bosons are combinations of  $W_1$  and  $W_2$ . The massive bosons acquire their mass through coupling with the Higgs field. The fermions of the SM acquire their mass through Yukawa coupling with the Higgs field. The electrical charge emerges as  $Q = T_3 + \frac{1}{2}Y$ , where  $T_3$  is the third component of the weak isospin.

All electrically charged fermions can interact via the electromagnetic force by exchanging a photon. The “charge” of the *weak* interaction is flavor, the 6 types of quarks and the 6 types of leptons shown in Fig. 2.1. Flavor is not conserved in weak interactions and allows for transformations of one particle type to another. For leptons, these transformations are only allowed within a generation. Quarks, however, can be transformed from one generation to another via the weak interaction.

## 2.2. The Large Hadron Collider

CERN’s research facilities are located near Geneva in Switzerland. The site houses several accelerators, with the Large Hadron Collider (LHC) [10] being the largest and also the most powerful one in the world. The LHC is a ring accelerator and collider inside the 26.7 km long tunnel of the old Large Electron–Positron Collider (LEP) machine [11], around 100 m underground. It mainly collides proton beams, but can also accelerate lead ions. One of the reasons for switching from light leptons at LEP to protons at the LHC is that it is

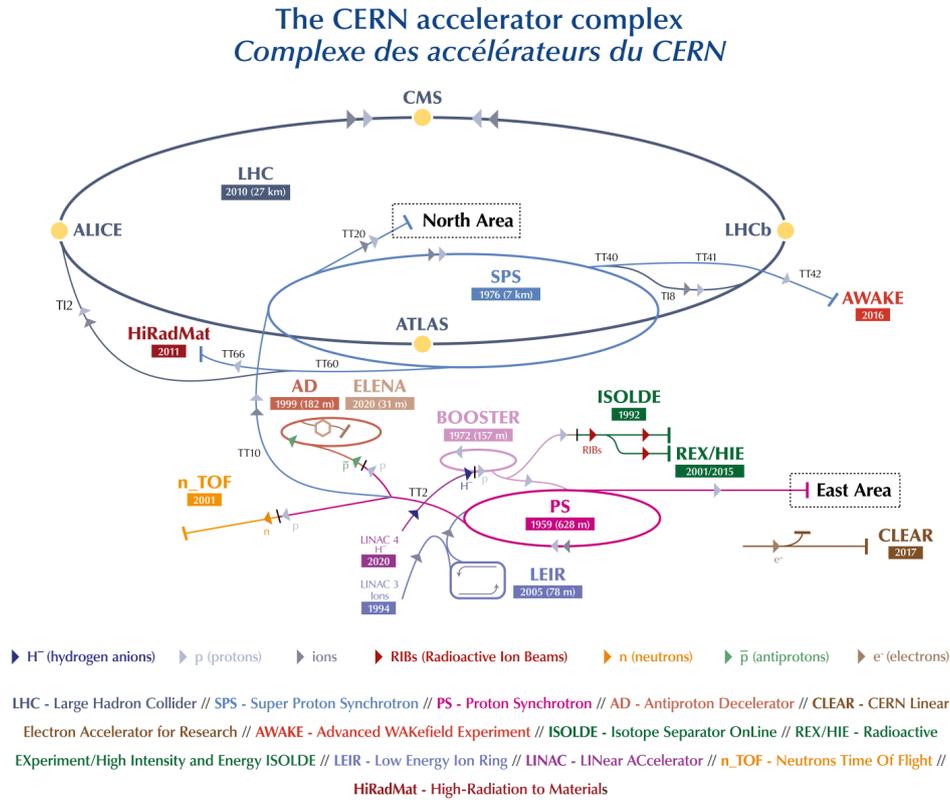
## 2. Hadron collider physics at the LHC

easier to accelerate protons to higher energies  $E$ . This is because the energy loss  $\Delta E$  due to synchrotron radiation is inversely proportional to  $m^4$  [12]:

$$\Delta E \approx \frac{1}{3\epsilon_0} \frac{q^2}{R} \left( \frac{E}{mc^2} \right)^4, \quad (2.1)$$

where  $m$  is the mass of the accelerated particle,  $q$  is the charge,  $\epsilon_0$  the vacuum permittivity and  $R$  the radius of the ring collider.

The position of the LHC in the larger CERN accelerator complex is shown in Fig. 2.2. The LHC is designed to create proton-proton collisions at center-of-mass energies of up to 14 TeV. In the acceleration chain of the protons, the LHC is only the last element.



**Figure 2.2.:** The CERN accelerator complex (© 2019-2022 CERN).

Since 2020, the Linear accelerator 4 (Linac4) [13] is the proton source for the LHC, where negative hydrogen ions ( $H^-$ ) are accelerated to an energy of 160 MeV. During injection into the Proton Synchrotron Booster (PSB), the electrons are stripped from the ions leaving only protons. The protons leave the PSB at an energy of 2 GeV and enter the Proton Synchrotron (PS) where they are accelerated to 26 GeV. The last stage before entering the LHC ring is the Super Proton Synchrotron (SPS), where the protons are

accelerated to 450 GeV. From the SPS, the beam is injected into the two beam pipes of the LHC at two points. In one beam pipe, the beam circulates clockwise around the ring and in the other beam pipe, the beam circulates anti-clockwise. Two separate pipes are needed, because the magnetic field that keeps the protons on their circular orbit must have opposite orientation for the two different directions of circulation. The magnetic field is created by 1232 dipole magnets cooled with superfluid helium to a temperature of 1.9 K. The magnets produce a magnetic field with a strength of 8.3 T to bend the protons' path into an arc. This is done in 8 areas, such that the full LHC ring consists of 8 arcs and 8 straight sections. In one of the straight sections, there are RF cavities that compensate the energy loss due to synchrotron radiation and accelerate the protons to energies of up to 7 TeV. Quadrupole magnets along the beam line are used for focusing and defocusing the beam.

The beam itself consists of 2808 bunches of protons with approximately  $1.15 \times 10^{11}$  protons per bunch. The nominal time between two bunches at a fixed point in the ring is 24.95 ns. At four points along the ring, the beams can cross and be brought to collision. The bunch crossing frequency at these interaction points is then 40 MHz. Around these four interaction points, the four detectors of the LHC have been built: ATLAS (A Toroidal LHC Apparatus) [14], CMS (Compact Muon Solenoid) [15], LHCb (Large Hadron Collider beauty) [16] and ALICE (A Large Ion Collider Experiment) [17]. The experiments take data of collision events when the beam is at a stable energy. A typical “fill” of the LHC with proton bunches is thus carried out as follows: First the beam has to be injected, that has come from the SPS. Then the energy ramps up and the particles are accelerated. This takes around 20 minutes. At the “flat top”, the particles have their final energy. After that, the bunches are squeezed to increase the so-called luminosity. For several hours following, there is a stable beam at constant energy during which the detectors can take physics data. Once the beam intensity becomes too low due to collision losses, the beam is dumped and a new fill can be started.

A full run of the LHC consists of many fills, usually only interrupted by short technical stops. Run 1 of the LHC with protons at a beam energy of 3.5 TeV started in March 2011. The beam energy was later increased to 4 TeV, corresponding to a center-of-mass energy in the head-on collisions of 8 TeV. Run 1 ended in 2012. After the first Long Shutdown (LS), Run 2 started in 2015 with protons at an energy of 6.5 TeV and ended in 2018.

During a run, when the bunches cross, particles can collide. Such a collision is called event or interaction. The event rate depends then on the number of particles in a bunch, on the bunch size and shape, on the bunch crossing frequency and on the cross-section of

## 2. Hadron collider physics at the LHC

the event. This can be formulated as

$$\frac{dN_{\text{event}}}{dt} = \sigma_{\text{event}} L, \quad (2.2)$$

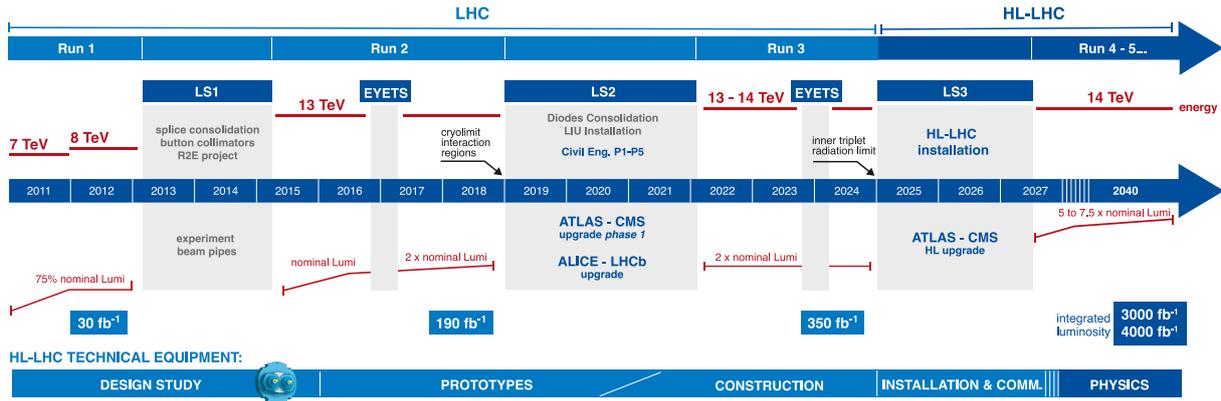
where  $\sigma_{\text{event}}$  is the cross-section of the underlying event measured in barns ( $1 \text{ b} = 10^{-24} \text{ cm}^2$ ).  $L$  is called the (instantaneous) luminosity, and summarizes the information about the beam and only depends on machine parameters. The luminosity is a measure for how many particles are put by the machine into a position to collide. The corresponding unit is  $\text{cm}^{-2} \text{ s}^{-1}$ . To obtain the total amount of events for a run, one has to integrate over time. Since the cross-section is time independent, this is given by

$$N_{\text{event}} = \sigma_{\text{event}} \int L dt = \sigma_{\text{event}} L_{\text{int}}. \quad (2.3)$$

The integrated luminosity  $L_{\text{int}}$  can be used as a measure for the performance of a particle accelerator as the number of any event of interest is directly proportional to  $L_{\text{int}}$ .  $L_{\text{int}}$  is usually expressed in inverse femtobarns ( $\text{fb}^{-1}$ ) for a complete run and measures the total amount of data that can be delivered to the experiments. In Run 1, the LHC was able to deliver an integrated luminosity of around  $30 \text{ fb}^{-1}$ , while in Run 2 it delivered around  $190 \text{ fb}^{-1}$  with peak (instantaneous) luminosities of  $2 \times 10^{34} \text{ cm}^{-2} \text{ s}^{-1}$ .

For the detectors, using hadron beams means that next to the hard parton collision (from quarks, antiquarks or gluons), many traces from underlying events overlap with the signals from the main interaction. Underlying events are all the activity in a single proton-proton collision that do not come from the main interaction, and can be for example the emission of more partons or photons directly before or after the main interaction. The usage of bunches means that several proton-proton collisions can happen per bunch crossing. This is called a *pile-up* of events. The detector can handle this, if it is able to deal with the large bandwidth for readout of the data and can reconstruct primary vertices of the interactions. The average number of interactions per bunch crossing measured by ATLAS in Run 2 was 33.7.

After Run 3, which is scheduled to start this year and end in 2024, it is projected that the LHC will have delivered an integrated luminosity of  $350 \text{ fb}^{-1}$ . Many critical components of the accelerator will have reached the end of their lifetime by then and will need to be replaced [18]. To ensure scientific progress and to exploit the full capacity of the LHC, it was decided to upgrade the LHC in the years from 2025 to 2027 to the High Luminosity Large Hadron Collider (HL-LHC) [3]. Since the statistical error for measurements scales with  $1/\sqrt{N_{\text{event}}}$ , a significant increase in (instantaneous) luminosity is desired. The HL-LHC aims for an increase of 5 to 7 times the nominal value of  $10^{34} \text{ cm}^{-2} \text{ s}^{-1}$ . The upgrade



**Figure 2.3.:** The HL-LHC upgrade schedule. The nominal luminosity of the LHC is  $10^{34} \text{ cm}^{-2} \text{ s}^{-1}$ . Note here, that the ATLAS and CMS upgrades during LS3 are also called phase 2-upgrades. EYETS stands for Extended Year-End Technical Stop.<sup>2</sup>

schedule for the HL-LHC project as of January 2021 is shown in Fig. 2.3. With the upgrade, the HL-LHC is to produce up to  $4000 \text{ fb}^{-1}$  of integrated luminosity by 2038, thereby enabling the experiments to enlarge their data sample by one order of magnitude compared to the LHC baseline program.

The increase in luminosity will be achieved, among other things, by the usage of crab cavities. These cavities can rotate the bunches right before collision to increase the overlap of the colliding bunches and increase the luminosity. It is projected that the effects through the upgrades are such that the number of events per crossing can be raised to around 130, while allowing a slight reduction of bunches from 2808 to 2760.

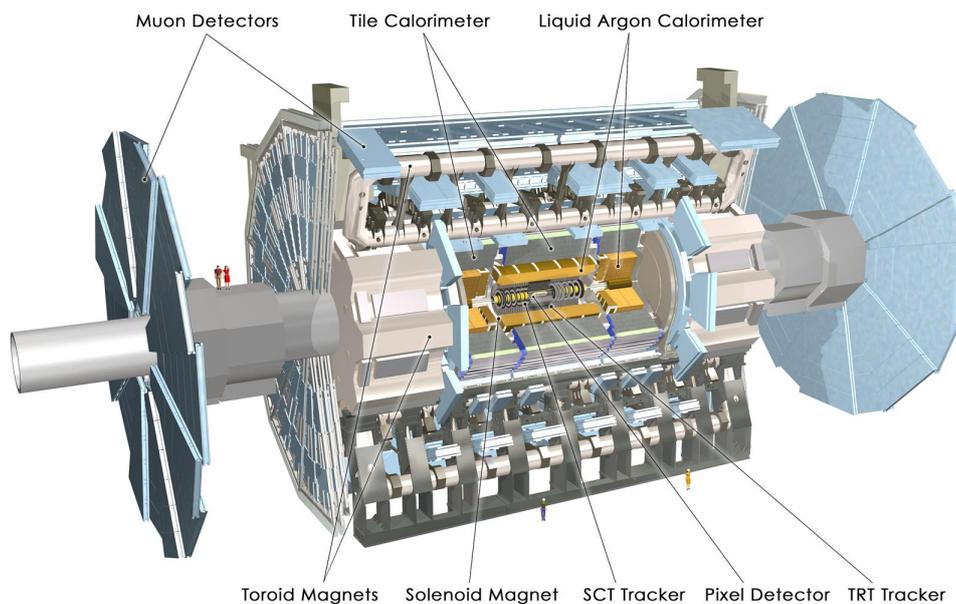
The higher radiation environment means that also the detectors will have to adjust in order to at least keep performance at the same level. The increased pile-up will lead to an increased occupancy and an increased readout bandwidth. With the larger data sample, the experiments can study rare events and perform high-precision measurements. And although the energy will not increase by a large amount, it might be possible to search for physics beyond the SM due to the significant increase in luminosity [19].

<sup>2</sup>From <https://hilumilhc.web.cern.ch/sites/default/files/images/HL-LHC-plan-2021-1.pdf> (accessed: January 5, 2021)



### 3. The ATLAS detector

The ATLAS detector [14] is a general purpose detector and the largest detector at the LHC. The detector has a cylindrical shape with a diameter of 25 m and a length of 46 m and weighs approximately 7000 t. A cut-away view is shown in Fig. 3.1. The detector is situated approximately 100 m below ground in a large cavern. As typical for a collider experiment, where lab frame and center-of-mass frame are identical, ATLAS is built with cylindrical symmetry around the beam pipe with an additional forward-backward symmetry with respect to the interaction point.

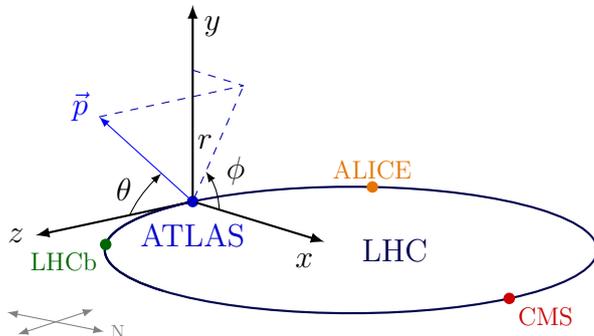


**Figure 3.1.:** Computer generated image of the whole ATLAS detector. The beam pipe crosses from left to right with the interaction point being in the center of the detector (© 2008 CERN).

For the description of the detector and the physics events that are recorded, ATLAS uses a right-handed coordinate system as shown in Fig. 3.2. The origin of the coordinate system is at the nominal interaction point in the center of the detector and the  $z$ -axis is along the beam pipe. The  $x$ -axis points from the interaction point to the center of the LHC ring, and the  $y$ -axis points upwards. Cylindrical coordinates  $(r, \phi)$  are used in the transverse

### 3. The ATLAS detector

plane, with  $\phi$  being the azimuthal angle around the  $z$ -axis. The pseudorapidity  $\eta$  is defined in terms of the polar angle  $\theta$  as  $\eta = -\ln \tan(\theta/2)$ . The A-side of the detector is defined as the side with positive  $z$  and the C-side as the side with negative  $z$ .



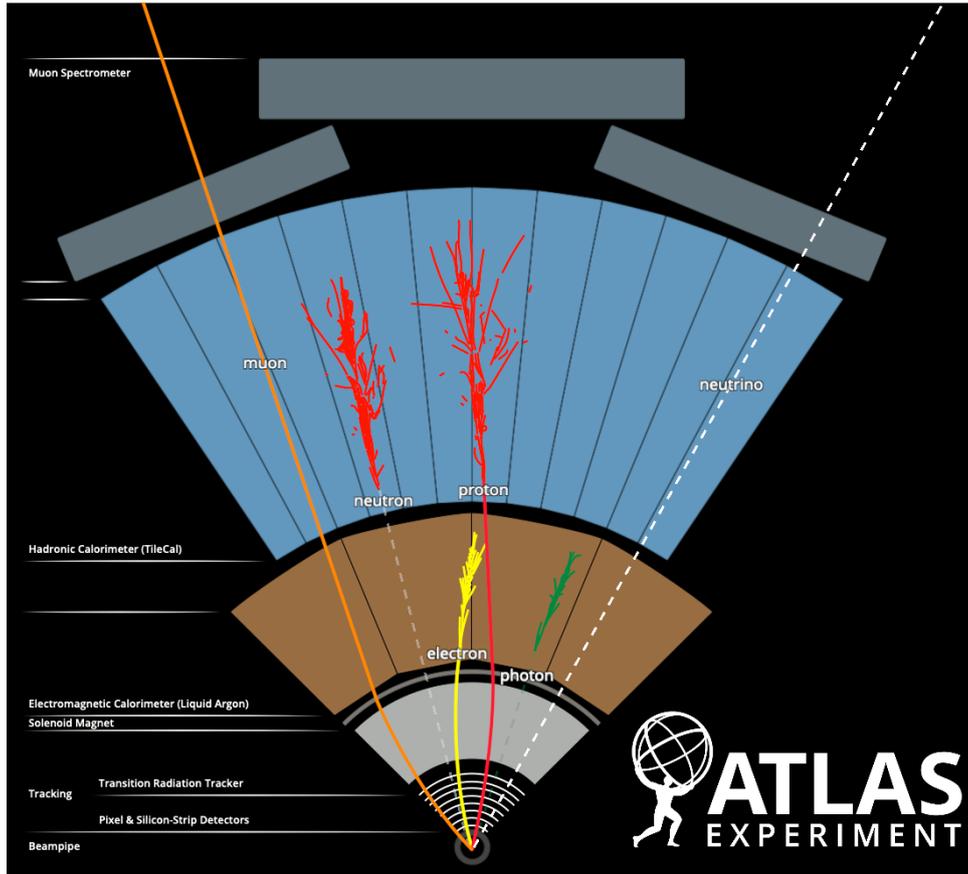
**Figure 3.2.:** Coordinate system of the ATLAS detector.

Due to its layout, the detector covers almost the full solid angle around the interaction point, with the exception of a thin cylinder for the beam pipe. As a general purpose detector, ATLAS aims for comprehensive detection of all possible reaction products. It is therefore built in layers around the beam pipe with different detector types for a particular function in each layer. The inner layers perform detection and tracking of charged particles in a magnetic field. In the outer layers are the electromagnetic and the hadronic calorimeters for measuring particle showers. The last layer is the muon spectrometer, since the muon penetrates the rest of the detector and is only identified in this last layer. An overview of the detection of different particles is given in Fig. 3.3. The electron and (relativistic) muon are considered as stable particles as they traverse the detector. The tau lepton with a lifetime of  $2.903 \times 10^{-13}$  s will travel a short distance but then decay before reaching the innermost detector layers. Hadrons with  $b$ -quarks that have a lifetime of the order of  $1.5 \times 10^{-12}$  s also behave similarly [20]. This makes vertex detectors close to the interaction point necessary to observe secondary vertices.

Quarks are detected since they hadronize as “jets”, which are collections of many color-neutral particles. It might not be possible to resolve all the constituents of a jet, but the energy and momentum of the jet can be determined from the total energy deposited in the calorimeters.

Neutrinos are not directly detected. They are very light and only interact via the weak force. It is therefore important to obtain accurate missing transverse energy measurements of events to infer the presence of neutrinos as a result of energy conservation.

With an inelastic proton-proton cross-section of  $\sigma_{\text{inel}} = 80$  mb and a luminosity of  $L = 2 \times 10^{34} \text{ cm}^{-2} \text{ s}^{-1}$  that accounts for bunch crossings every 25 ns, the LHC produces many inelastic collision events per bunch crossing. This requires fast and radiation hard



**Figure 3.3.:** Detection of different particles by ATLAS and its detector systems. The dashed tracks are invisible to the detector. The bended tracks of the charged particles are due to a magnetic field. (© 2021 CERN).

electronics and sensor elements in the detector. The inner layers are affected the most and their sensors and readout electronics must be prepared for high particle fluences. Because of technological constraints, the readout rate of event data is limited and many detected events have to be rejected. ATLAS uses triggers to deal with events produced at the bunch crossing rate of 40 MHz. The Level-1 (L1) trigger system uses a subset of the total detector information to make a decision on whether or not to continue processing an event. This hardware based trigger has a latency of  $2.5 \mu\text{s}$ . The L1 trigger accepts events at a rate up to the maximum detector readout rate (limited by the bandwidth) of 100 kHz. The final output rate of physics data, decided by the (software) High Level Trigger (HLT), during an ATLAS data-taking run is on average 1.2 kHz with an average physics throughput to permanent storage of 1.2 GB/s [21].

With the upgrade of the LHC to the HL-LHC and increase in luminosity, the pile-up will increase from on average 30 events of inelastic proton-proton collisions to 200 events per bunch crossing. This will be a highly challenging environment for particle detectors.

### 3. The ATLAS detector

To cope with these conditions, upgrades for the ATLAS detector are planned and executed during LS2 for Run 3 and an upgrade plan has been laid out for LS3 [22]. The two major upgrade areas affect the Trigger and Data Acquisition (TDAQ) system and the Inner Detector (ID). Significant improvements to the TDAQ system are needed to accomplish the physics goals of the HL-LHC upgrades. The upgrade introduces a new Level-0 (L0) trigger for the calorimeters and muon system as well as an update of the readout system utilizing FELIX<sup>1</sup> [23]. A second major decision was made to replace the current ID with a new all-silicon Inner Tracker (ITk). This decision was driven by the need for a radiation hard and high-rate capable tracking system. The new ITk will comprise two subsystems, a silicon Pixel Detector [24] surrounded by a silicon Strip Detector [25].

In the following, first the current detector systems of ATLAS are presented in more detail, with focus on the Inner Detector and then the planned ITk upgrade. During prototyping, an extensive testing schedule is set up. Some of these tests are the ITk Pixel system tests and one part is the Outer Barrel demonstrator, which is discussed next. This thesis focuses on the DCS of the demonstrator, therefore an overview of the general ATLAS DCS is given as an introduction as well as an outlook on the planned control structure for the ITk Pixel.

## 3.1. Detector systems of the ATLAS detector

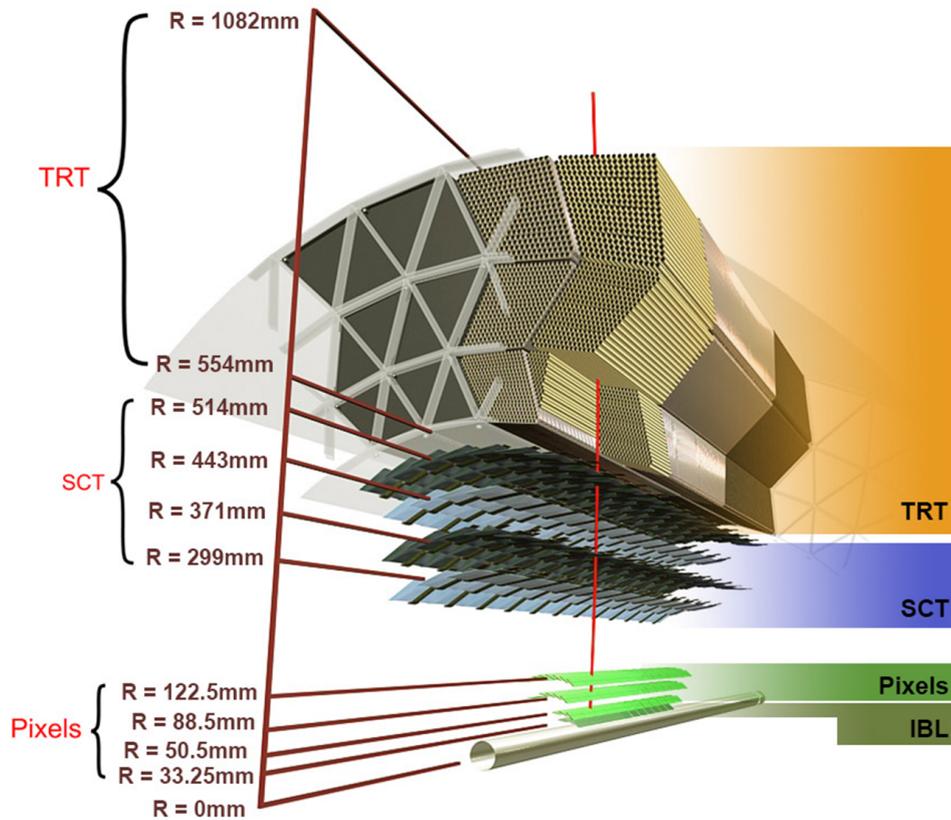
**Inner Detector** The purpose of the Inner Detector (ID) [26] is the tracking of charged particles from the interaction point until they reach the calorimeters. The ID is cylindrical around the beam pipe with a length of around 6.2 m and a diameter of around 2.1 m. The tracking starts as close as practically possible to the beam. The information from the ID is then used for track reconstruction, momentum and (primary and secondary) vertex measurement and electron identification. The ID is situated inside a solenoid magnet that creates a magnetic field of 2 T parallel to the beam axis and forces charged particles on a curved trajectory. The acceptance in pseudorapidity is  $|\eta| < 2.5$  for particles coming from the nominal interaction point, with full coverage in the azimuthal angle  $\phi$ . At nominal luminosity, the ID has to cope with approximately 1000 particles every 25 ns.

The ID consists of three independent sub-detectors: the Pixel Detector, the Semiconductor Tracker (SCT) and the Transition Radiation Tracker (TRT). The layout is shown in Fig. 3.4. During LS1, an additional layer, the Insertable B-Layer (IBL) [27], was added

---

<sup>1</sup>Front-End Link Interface eXchange. FELIX is a network switch to convert GBT-like protocol data to Gigabit Ethernet or InfiniBand networking

at a radius of 33.25 mm between Layer 0 of the Pixel Detector and a new smaller radius beam pipe.



**Figure 3.4.:** The structure of the ATLAS ID for Run 2 and Run 3 made of highly granular silicon pixels (IBL and Pixel), silicon strips (SCT) and straw tubes (TRT). The red curved line represents a charged particle traversing the various layers and bending in the 2 T magnetic field. The innermost pixel layer is called the Insertable B-Layer and was added to the detector between the first and second runs of the LHC (© 2020 CERN).

The Pixel Detector features 3 concentric layers in its barrel region with the midpoint at the interaction point and 3 end-caps on both A- and C-side. It offers high granularity and high precision measurements close to the vertex region by utilizing hybrid pixel sensors. These sensors are high in costs but needed for fast readout and low occupancy. The FE-I3 [28] readout chips in the Pixel Detector have 2880 readout cells with an area of  $50\ \mu\text{m} \times 400\ \mu\text{m}$  per cell and individual circuits for each pixel element. They cover  $16.4\ \text{mm} \times 60.8\ \text{mm}$  active area. The readout chips are oriented such that the  $50\ \mu\text{m}$  pitch is used to measure the  $\phi$  coordinate. The column-wise readout also includes buffers to store the data while awaiting the Level-1 trigger decision. Each chip is bump-bonded to

### 3. The ATLAS detector

the detector substrate and includes radiation-hard electronics as the radiation dose for the innermost Pixel layer is expected to reach 500 kGy, the lifetime dose, after approximately 5 years of LHC operation at nominal luminosity. For the other layers, it is expected that this dose is reached after around 10 years of operation [29]. Sixteen FE-I3 chips are grouped together in a module, the basic building block of the active part of the Pixel Detector. There are in total 1744 Pixel modules with approximately 80.4 million readout channels. The total active area of the Pixel Detector is 1.73 m<sup>2</sup>.

The modules of the Pixel Detector are mounted on mechanical supports, called staves, in the barrel region and disk sectors in the end-cap region. The mechanical supports also carry the services which are needed for each module: powering, monitoring, data input/output and cooling. All these add material, although it is important to keep the amount of material inside the detector small in order to reduce multiple scattering and secondary interactions.

The SCT is a silicon microstrip detector around the Pixel Detector with 4 layers of radii from 299 mm to 514 mm in the barrel region and 9 disks in each end-cap region. In total there are 4088 modules. Each module consists of 4 rectangular silicon-strip sensors [30] with a second pair of identical sensors glued back-to-back at a stereo angle of 40 mrad [31]. A module has 768 strips that are approximately 12 cm in length. The total number of readout channels is approximately 6.3 million. In the barrel region, the strips are approximately parallel to the beam axis with a pitch of 80  $\mu\text{m}$ . The small stereo angle between the first and second pair of sensors allows for more precise measurement in the  $z$  direction. The readout electronics and sensors of the SCT need cooling to keep leakage current low especially after radiation damages. The same evaporative cooling system [32] with C<sub>3</sub>F<sub>8</sub> as refrigerant fluid as for the Pixel Detector is used, with a target temperature of 0 °C for the SCT modules [33].

The TRT is in the outermost part of the ID and used for continuous tracking and electron identification. Its sensitive volume in the barrel region covers radial distances from 563 mm to 1066 mm. Together with the two end-cap regions it has an acceptance range of  $|\eta| < 2.0$  [34]. The active elements of the TRT are 298 304 proportional drift tubes (straws), 4 mm in diameter with the walls at a potential of  $-1530$  V with respect to the center wire, and read out by 350 848 channels of electronics. The TRT straw layout is designed so that charged particles with transverse momentum  $p_{\text{T}} > 0.5$  GeV and with pseudorapidity  $|\eta| < 2.0$  cross typically more than 30 straws [35]. The crossing particles ionize the gas mixture inside the straws and create an electron drift towards the grounded wire in the center of the tube. The electrons cascade close to the wire and create a detectable signal. The position measurement accuracy is about 110  $\mu\text{m}$ .

### 3.1. Detector systems of the ATLAS detector

Radiation fibers and foils between the straws cause traversing particles to create transition radiation. The emitted photons energy is proportional to the Lorentz factor  $\gamma$  of the particle and absorbed by the gas in the straws. The measurement of the deposited energy allows for particle identification and is used to distinguish electrons from pions.

The occupancy in the TRT is defined as the probability to have a straw signal above a given threshold value in the readout window of 75 ns. The value then characterizes the particle density in the sub-detector and depends on the pile-up of interactions per bunch crossing. Already at nominal luminosity, the TRT straw occupancy can reach 60 %. Large detector occupancies present many challenges for track reconstruction, including a degradation of the track parameter resolution due to incorrect hit assignments, a decrease of hit efficiencies, and an increase in the rate of fake tracks due to random hit combinations.

After detector alignment, the impact parameter resolutions for secondary vertex measurements of high-momentum tracks were found to be  $(22.1 \pm 0.9) \mu\text{m}$  and  $(112 \pm 4) \mu\text{m}$  in the transverse and longitudinal directions, respectively. In this asymptotic limit, the relative momentum resolution was measured to be  $\sigma_p/p = (4.83 \pm 0.16) \times 10^{-4} \text{ GeV}^{-1} \times p_T$  [35].

The innermost pixel layer is called the Insertable B-Layer and was added to the detector between the first and second runs of the LHC at a radius of only 33.25 mm [36]. It represents an upgrade of the Pixel Detector to improve the performance of the ATLAS tracking system after performance degradation of silicon sensors and front-end electronics in Pixel layers due to radiation damage. Due to its close proximity to the interaction point, the IBL sensors and front-end electronics must cope with high hit rates and radiation doses. To address these requirements, a new front-end readout chip, the FE-I4 [37], was developed. This chip satisfies the requirements of radiation tolerance and readout efficiency at high luminosity. In comparison to the readout chip used in the other Pixel layers, the cell size was reduced to  $250 \mu\text{m} \times 50 \mu\text{m}$  with the shorter side being in the transverse plane. In total there are 448 readout chips on 14 local support structures called staves. Each channel is bump-bonded to a single pixel of the sensor. The IBL adds  $0.15 \text{ m}^2$  of active surface and approximately 12.04 million readout channels.

The staves are mounted with the sensors facing the beam pipe and are inclined in azimuth by  $14^\circ$  to achieve an overlap of the active area and full coverage in  $\phi$ . To reduce material, the IBL uses  $\text{CO}_2$  evaporative cooling for the sensors, which is more efficient in terms of mass flow and pipe size.

The smaller layer radius and the reduced pixel cell length are crucial parameters in defining the performance improvement of the ID. The IBL improves the impact parameter resolution by approximately a factor of two for tracks with a  $p_T \simeq 1 \text{ GeV}$ .

### 3. The ATLAS detector

**Calorimeters** The calorimeters of ATLAS measure the particles' energy after they have passed the ID. This measurement is a destructive process. The particles are slowed down until they are absorbed. The particle's energy is converted into a detectable signal along the way. The size of the calorimeters is such that almost all particles are completely stopped. Only muons, which are detected in the muon chambers later, and neutrinos, which are undetected, are exceptions to this.

ATLAS uses sampling calorimeters. These have layers of high density passive material to slow particles down, alternating with layers of active material for signal generation and energy measurement.

The Electromagnetic Calorimeter (ECal) [38] is a Liquid Argon (LAr) Calorimeter used to measure the energy of electrons, positrons and photons. As the passive material, lead is used to create electromagnetic showers. The active material is liquid argon at a temperature of  $-184^\circ\text{C}$ . The layers have a characteristic accordion structure. The ECal is subdivided into a barrel and two end-caps and covers a pseudorapidity range of  $|\eta| < 3.2$  with an energy resolution of  $\sigma_E/E = 10\%/\sqrt{E(\text{GeV})} \oplus 0.7\%$ .

The Hadronic Calorimeters (HCals) [39] surround the ECal and comprise the Tile Calorimeter, the LAr hadronic end-cap calorimeter and the LAr forward calorimeter. The Tile Calorimeter [40] is divided into a barrel and two end-cap regions. Together they cover a pseudorapidity range of  $|\eta| < 1.7$ . The calorimeter uses steel as the absorber and plastic scintillating tiles as the active material. Hadrons hitting the layers of steel produce hadronic showers which in turn produce photons in the plastic scintillators. The photons are detected by photomultiplier tubes.

The LAr end-cap calorimeter extends over a range of  $1.5 < |\eta| < 3.2$ . It uses liquid argon as active material and copper as passive material. The Tile Calorimeter and the LAr end-cap calorimeter are designed to have an energy resolution of  $\sigma_E/E = 50\%/\sqrt{E(\text{GeV})} \oplus 3\%$ .

The LAr forward calorimeter covers the range  $3.1 < |\eta| < 4.9$ . It was designed to have an energy resolution of  $\sigma_E/E = 100\%/\sqrt{E(\text{GeV})} \oplus 10\%$ . Each end-cap consists of three modules. The first uses copper as passive material and is optimized for electromagnetic measurements. The other two modules use tungsten as absorber material and measure predominantly the energy of hadronic interactions.

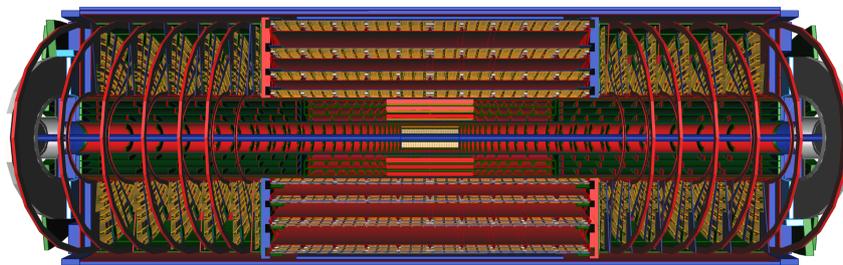
**Muon detector** The outermost part of the ATLAS detector is the Muon Spectrometer [41]. It is designed to detect muons exiting the calorimeters and to measure their momentum in the pseudorapidity range  $|\eta| < 2.7$  [14]. The magnetic field is generated by superconducting magnets (in the barrel region by the characteristic toroids) and bends the path of the traversing muons. Precision tracking is performed by Monitored Drift

Tubes (MDT) and Cathode Strip Chambers (CSC). The goal is to improve the transverse momentum resolution of high- $p_T$  muons. In addition to tracking, the Muon Spectrometer is also designed to trigger on detected particles. Resistive Plate Chambers (RPC) and Thin Gap Chambers (TGC) are used as fast trigger chambers. These chambers are capable of delivering track information within a few tens of nanoseconds after the passage of the particle.

### 3.2. Inner Tracker and ITk Pixel detector (HL-LHC upgrades)

The fivefold increase in instantaneous luminosity at the HL-LHC will pose many challenges, especially to the ATLAS detectors close to interaction region. An average of 200 proton-proton interactions per bunch crossing means that a fine granularity is needed to keep occupancy low. It was therefore decided to replace the current ID with an all-silicon Inner Tracker (ITk) [22]. The ITk will comprise a Pixel Detector [24] at small radius and a large area Strip Detector [25] surrounding it. The TRT will be removed as it is unsuitable for a high luminosity environment. The advantage of a large pixel detector close to the beam pipe is its low occupancy, since pixel cells have only a small sensitive area per readout channel.

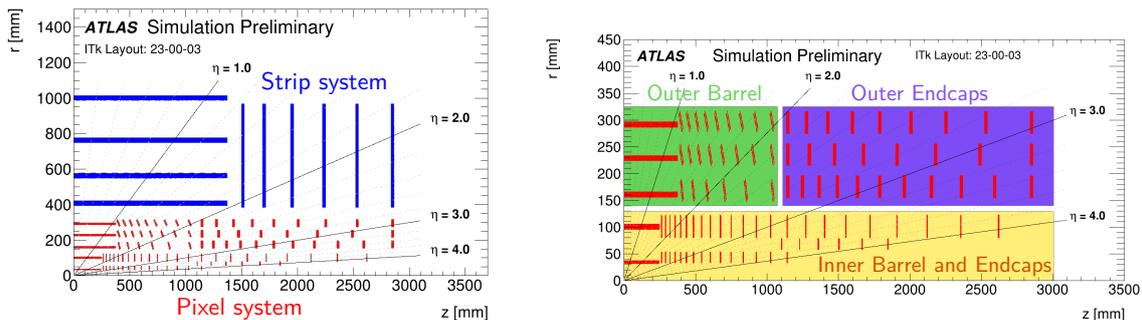
The design goal for the ITk is to have the same or better performance as the current Inner Detector but in the harsher environment of the HL-LHC. To maximize the physics potential, the tracking coverage is extended to  $|\eta| < 4.0$ . The latest layout and design choices are documented elsewhere [42]. A computer generated model is shown in Fig. 3.5. To achieve the performance goals, the deployed sensor technologies have to be radiation hard in order to withstand 1 MeV neutron equivalent fluences of up to  $1.3 \times 10^{16} \text{ n}_{\text{eq}}/\text{cm}^2$  during their lifetime. The readout electronics must achieve a readout rate of 1 MHz.



**Figure 3.5.:** Computer generated image of the ITk layout [42]. The detector is approximately 6 m long and has a radius of about 1 m.

### 3. The ATLAS detector

The Strip detector has four layers in the barrel region and six end-cap disks to cover a pseudorapidity of  $|\eta| < 2.7$ . It is complemented by a 5 layer Pixel detector extending the coverage to  $|\eta| < 4$ . The two volumes are separated by a Pixel Support Tube (PST). A schematic depiction of the ITk layout is shown in Fig. 3.6. The zoomed-in view in Fig. 3.6b shows the subsystems of the Pixel detector. The inner system (barrel and end-caps) is inside an Inner Support Tube (IST) and can be replaced.



(a) A schematic depiction of the ITk layout with the Strip and Pixel system. (b) A zoomed-in view of the Pixel detector.

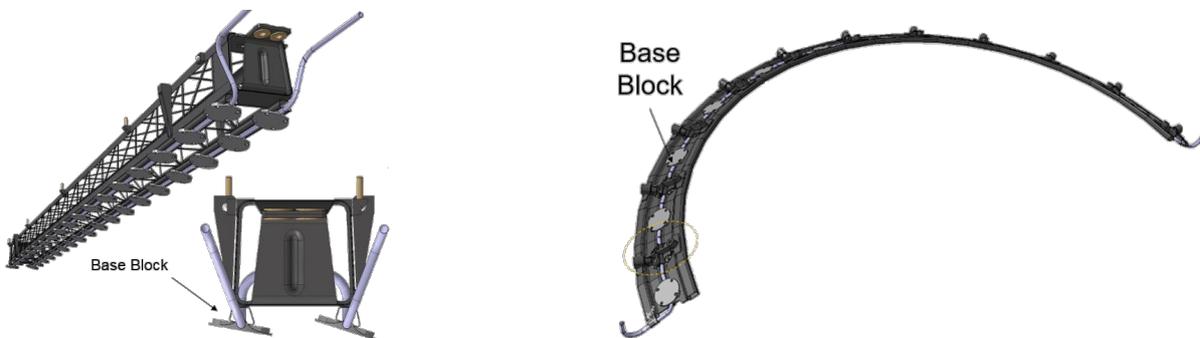
**Figure 3.6.:** A schematic depiction of the ITk Layout as presented in Ref. [42]. In each case, only one quadrant and only active detector elements are shown. The active elements of the Strip detector are shown in blue, and those of the Pixel detector are shown in red. The horizontal axis is along the beam line with zero being the nominal interaction point. The vertical axis is the radius measured from the interaction point.

The Strip system features double modules, i.e. sensors with two readout chips, each sensor with a small stereo angle to add  $z$  ( $r$ ) resolution in the barrel (end-caps), respectively. The active area of the silicon sensors is  $165 \text{ m}^2$ , 2.5 times that of the current ID SCT. The Pixel system features in total around 10 100 hybrid pixel modules with an active area of approximately  $13 \text{ m}^2$ . More detailed parameters for the Outer Barrel (OB) region of the Pixel system are shown in Table 3.1.

The modules in the OB are mounted on Loaded Local Supports (LLSs). An LLS is made of high thermal performance carbon-composite materials and includes one cooling pipe. There are two types of LLS in the barrel region: the flat longeron and the inclined half ring [43]. A computer generated image of these LLSs can be seen in Fig. 3.7. With the inclined design, the sensors in a half ring are as perpendicular as possible to particles coming from the interaction point. This reduces the traversed material. The longerons and inclined half rings are then assembled into a layer of the OB. Fig. 3.8 shows one-half of layer 4 of the Pixel detector.

**Table 3.1.:** Parameters for the Outer Barrel subsystem of the ITk Pixel detector. In the flat region, the longerons cover  $z = -372$  to  $372$  mm with modules in two rows parallel to the beam axis. The inclined rings at  $|z| = 380$  to  $1035$  mm on both A- and C-side are composed of two inclined half rings, one on top and one on bottom. All modules in the Outer Barrel region are quad modules with four readout chips [42].

Layer	Radius $r/\text{mm}$	Longerons	Modules per Longeron	Inclined Half Rings	Modules per Inclined Half Ring
2	160	16	36	$2 \times 2 \times 6$	16
3	228	22	36	$2 \times 2 \times 8$	22
4	291	28	36	$2 \times 2 \times 9$	28

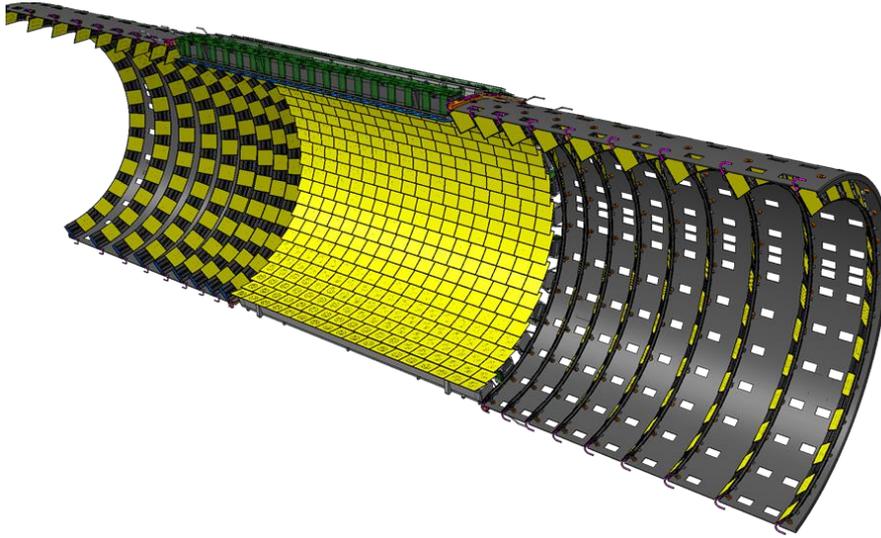


**Figure 3.7.:** CAD view of a functional longeron (left) and inclined half ring (right) [43], including the carbon fibre reinforced plastic (CFRP) truss structure, the cooling pipes and the base blocks, on which the modules will be mounted.

Everywhere in the OB, hybrid pixel detectors arranged in quad modules will be used. A quad module consists of four readout chips and a  $150 \mu\text{m}$  thick and  $4 \text{ cm} \times 4 \text{ cm}$  large silicon sensor with n-in-p technology. The pixel sensors have a pitch of  $50 \mu\text{m} \times 50 \mu\text{m}$  [42]. After irradiation, a high voltage (HV) for biasing of the sensors is needed. As the Front-End (FE) chip, a new readout ASIC is being developed by the RD53 collaboration<sup>2</sup>. Currently, the RD53A prototype chip [44, 45] is being used in tests, while the final version (ITkPix [46, 47]) will only be available at a later stage. For powering the chip, a ShuntLDO [48] is integrated. The ShuntLDO regulator is a combination of a linear Low Drop-Out voltage regulator and a shunt element. It ensures constant current consumption, even in case of failure, and stable supply voltage to the chip. The goal is to provide constant current operation with multiple FE chips connected in parallel in a module.

<sup>2</sup><https://rd53.web.cern.ch/>, Collaboration between ATLAS and CMS

### 3. The ATLAS detector



**Figure 3.8.:** CAD rendering of layer 4 of the OB [43]. On both ends are the inclined units, which will hold the inclined half rings, and in the middle are the longerons connecting them both. The modules are shown in yellow. For simplicity, the services are not shown.

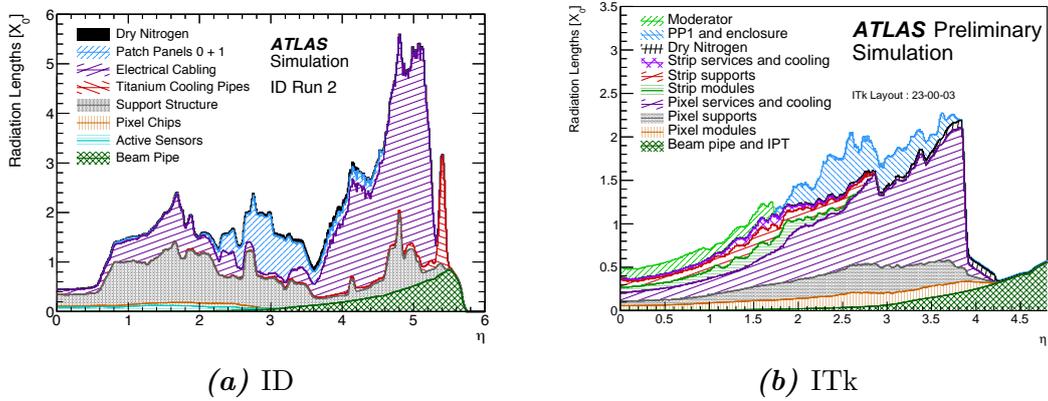
Even though the four FE chips of a quad module are connected in parallel, the powering of the *modules* will happen in series with a constant current source. Up to 14 quad modules are connected in serially powered (SP) chains to provide low voltage (LV) to the FE chips. This leads to a reduction in cable lines and material. However, it is important that the detector is robust in presence of dead modules due to possible component failures.

Using thin sensors and electronics, CO<sub>2</sub> biphase cooling and serial powering greatly reduces the material in the ITk envelope. Fig. 3.9 shows the material distribution (in terms of radiation lengths) of the ITk in comparison to the ID, where it can be seen that a large reduction of material is achieved especially for  $1 \leq |\eta| \leq 2$  and  $|\eta| > 4$ . This reduction diminishes the negative effects of multiple scattering and hadronic interactions on tracking performance and energy measurement in the calorimeters.

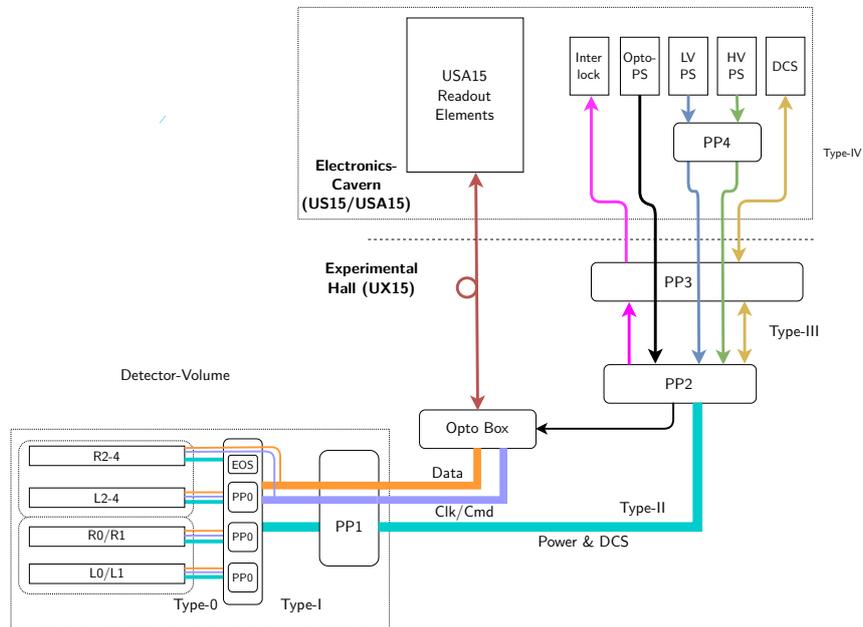
Figure 3.10 shows how the cables for data transmission, commands, monitoring and power are routed out of the detector volume. To collect cables at certain breakpoints, so-called Patch Panels (PPs) are used. Services running along the local supports up to PP0 are called Type-0 services, while the ones running from PP0 to PP1 are called Type-1 services.

The ITk will be commissioned during LS3. It is planned to fully build the detector above ground in a laboratory and transfer it as a single unit into the existing ATLAS detector. Next to a comprehensive set of quality assurance and quality control procedures, constant testing is ongoing.

### 3.2. Inner Tracker and ITk Pixel detector (HL-LHC upgrades)



**Figure 3.9.:** Material distribution in the ID (for Run 2) [25] in comparison to the ITk [42] in terms of radiation length versus the pseudorapidity  $|\eta|$ .



**Figure 3.10.:** Overview of the ITk Pixel services and their breakpoints. The starting points in the detector volume are the SP-chains. [49]

### 3.3. ITk Pixel system tests and OB demonstrator

It is crucial to study integration and system aspects in parallel to sensor and module testing. To test the on-detector components, *system tests* for all three regions of the Pixel detector are planned. These components include services, Power Supply Units (PSUs), DCS, cables, grounding and shielding. The system test should check the following:

- Check interlock functionality
- Verify the readout path
- Verify the powering scheme (LV and HV)
- Test the serial powering scheme
- Test the modules on LLSs
- Validate the services
- Check the DCS
- Run under realistic conditions
- Check for cross talk between modules
- Do source tests of modules with radioactive sources for signal generation

One system test will use the *OB demonstrator* set up in the SR1 lab at CERN to mimic the real detector. The OB demonstrator program should validate key features of the detector design. The configuration of the demonstrator is shown in Fig. 3.11.

In a first step, 27 RD53A modules will be used in the SP-chains. The capacity of the demonstrator is however 40 modules, and more modules can be installed later. The demonstrator is also prepared to take in ITkPix modules, once they become ready for system testing.

For readout of the modules, an Opto Box with Opto Boards will be set up. The Opto Boards convert the electric signal from the FE chip into an optical one and send it away from the detector to the FELIX readout system. The conversion to optical signals has to be done to ensure fast speeds over long distances. For monitoring of the module temperature and voltage drops, a dedicated ASIC, named Monitoring Of Pixel System (MOPS) [50, 51] will be used. There will be one MOPS per SP-chain and one MOPS to monitor the Opto Box. The readout of the MOPS will be done by a MOPSHUB4Beginners, an interim solution that replaces the MOPSHUB which is foreseen in the final design as described in Section 3.5.1. Both the MOPS and the MOPSHUB4Beginners specifically provide monitoring functionality and can read monitoring data independent of the current physics data taking state.



**Figure 3.11.:** CAD view of the ITk OB demonstrator with one longeron and two half rings. The longeron will have one full SP-chain with 12 modules and a smaller SP-chain, which can be filled with up to 6 modules. The inclined half rings will have one SP-chain each, with one SP-chain being full (11 modules).

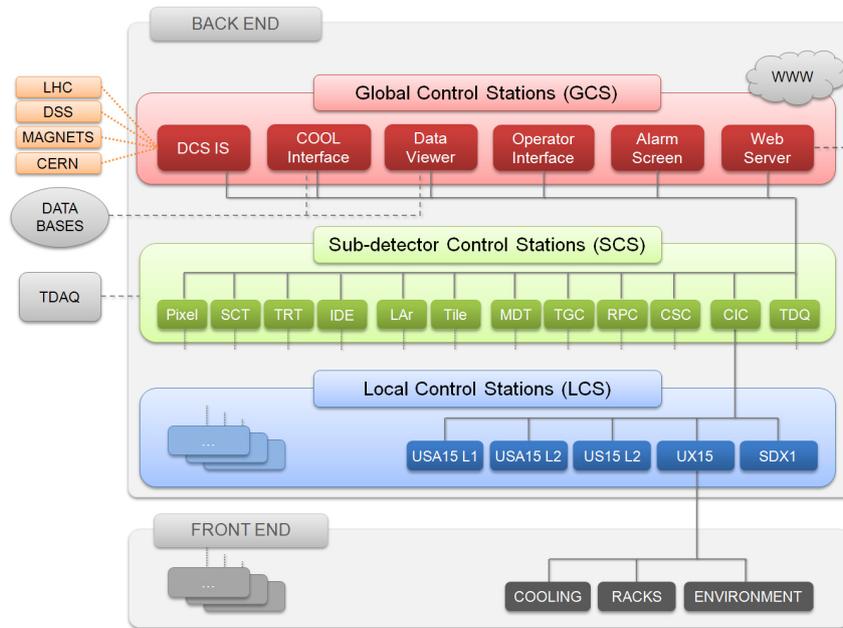
### 3.4. ATLAS Detector Control System

The purpose of the Detector Control System (DCS) is to prepare and maintain the detector in a well-defined state for high quality physics data taking. The DCS is responsible for ensuring an effective and safe configuration of the various subsystems and equipment that control and monitor all services, infrastructures and sub-detectors involved in the experiment [52].

From a high-level view, ATLAS is operated by two complementary systems, the DCS and the TDAQ. When the LHC is running, TDAQ is responsible for the readout of the detector data generated by particle collisions in the detector. During this time, TDAQ will also take control over the DCS. With more than 150 PCs, the DCS is a highly distributed system, hierarchically organized for operating the detector. The DCS has to supervise the individual detector components and the infrastructure needed for the experiment, during both “physics taking” time and during maintenance and calibration. The supervision is realized by reading and processing sensor data and the state of the active equipment or devices used for the detector. It is possible to archive these parameters for future reference. The DCS is able to analyze live data, recognize errors and warn the operator in the control room with alarms. It will allow manual and/or automatic actions to be taken to handle potential errors. All of this must be done in an accessible manner to the operator in the control room. The DCS also provides the operator with control over the detector and makes it possible to bring the detector in any desired operational state.

### 3. The ATLAS detector

For a large experiment like ATLAS that is embedded in a whole set of machinery and hardware at CERN, it is important that the DCS can communicate with external systems such as LHC run control. One of the requirements for the DCS are therefore to provide platform-independent interfaces for communication. Another requirement is that the DCS must be able to scale to a large number of parameters that can be managed.



**Figure 3.12.:** Schema of the DCS architecture. The detector hardware is interfaced to the DCS Front-End systems (sensors, PSUs etc.). The data is processed by a hierarchical Back-End system and presented to the control room operator using a single User Interface.

The implemented architecture for the DCS is shown in Fig. 3.12. It is organized into DCS Front-End (FE) equipment and Back-End (BE) systems. In general, commercial systems are preferred for the FE equipment to ensure long-term maintainability and efficient development. Only if no commercial systems can fulfill the requirements, custom designs are developed. The DCS then also consists of FE devices like power supplies and sensors. As a multipurpose I/O device, a custom DCS hardware component was developed, called the Embedded Local Monitor Board (ELMB) [52]. To enable the independent shutdown of the detector in case of an emergency, a hardware interlock system is set up. The communication between FE and BE is mostly done via CAN and the CANopen protocol or via Ethernet and the OPC-UA protocol. On the BE, there are the DCS computers (industrial rack-mounted PCs) which run a distributed system of SIMATIC WinCC Open Architecture (WinCC OA) instances to ensure high performance and scalability. The

WinCC OA projects for local control are grouped under their respective sub-detector, where each sub-detector has a certain level of operational independence.

Since a lot of the work for the BE is similar for all four LHC experiments, the Joint Controls Project (JCOP) collaboration<sup>3</sup> was founded [53]. The JCOP collaboration defines standards for the use of DCS hardware and software, provides OPC-UA servers for connecting different power supplies to the WinCC OA systems, and develops the JCOP framework software components and the software system for the Finite State Machine (FSM). The FSM is used to model the hierarchical structure of the control system. General information about the FSM software system is given in Chapter 4. Since the ATLAS detector is organized in a hierarchy tree with detectors and sub-detectors and sub-systems all having some level of independence, the distributed FSM accounts for this structure in the operational model.

The settings of the detector and DCS parameters (such as calibration constants, voltage settings, and alarm thresholds) are stored in a Configuration Database (a set of Oracle databases). The settings can be retrieved in form of “recipes”.

The detector conditions data is stored in a Conditions Database for efficient access by any offline calibration, reconstruction, and analysis applications [52].

#### 3.4.1. Hardware

To safely operate the detector Front-Ends and read out physics data, a complex power supply and control system is necessary. The environment should continuously be monitored and automatic actions should be taken to minimize risks for equipment or personnel. For this, the DCS comprises of the following hardware:

**ELMB** The ELMB houses an 8-bit microcontroller with 64 analog input channels multiplexed to a 16-bit ADC [52]. The general purpose software for the microcontroller furthermore offers access to 16 digital output and 16 digital input channels. The analog channels can be used to read out sensors for temperature (for example by NTC<sup>4</sup> sensors), humidity or dew point. Using the built-in CAN controller, the ELMB can send data asynchronously to a BE system for environmental slow monitoring.

The components were chosen to allow the ELMB to operate in high magnetic fields and under ionizing radiation. The power is distributed to the board via the CAN bus cable, with two wires for the communication and monitoring data and additional wires for power.

---

<sup>3</sup><https://jcop.web.cern.ch/>

<sup>4</sup>Negative Temperature Coefficient Thermistor

### 3. The ATLAS detector

**Power supplies** ATLAS uses commercial VME crates that house power supplies in rooms not affected by radiation. Power is needed for the ELMBs and the detector electronics. As a precaution, power supplies should provide over-voltage and over-current protections.

The DCS then monitors and controls the power supplies. As an interface, the usage of the OPC-UA protocol is common.

**Interlock** While the DCS software provides solutions for the slow control of the detector, a hardwired fast interlock system handles all situations where safety is an issue [54]. The interlock system is able to switch off all necessary power supplies and operates completely independently from the rest of the control system.

Several parameters are monitored and for example temperature sensors are wired directly into the interlock system, where an FPGA determines the current conditions from all input values and asserts the corresponding interlock alerts.

Although the interlock system is purely hardware based, it is required to communicate and explain the interlock actions to the rest of the DCS. The monitoring of the interlock signals is realized by ELMBs. They can also send test signals to the interlock system using the digital output channels. However, they do not change the decision logic of the FPGA.

#### 3.4.2. Protocols

To interface two systems, different protocols defining how the interactions should happen can be used. The ATLAS DCS uses mainly the following protocols:

**CANopen** The Controller Area Network (CAN) is a serial bus standard developed to allow communication between nodes on the bus without a master node [55]. It is a message-based protocol, where data is transmitted sequentially. Every device is identified with a node-ID and can send and receive data. If two devices want to send data at the same time, the device with the lower priority will automatically stop pushing data on the bus.

CANopen<sup>5</sup> is a high-level protocol on top of CAN. The ELMB general purpose software conforms to the CANopen DS-401 Device Profile for I/O modules [52].

The bus is physically realized over a twisted pair of wires. The advantages of using CAN are the low cost, the savings in material and the insensitivity to magnetic fields. It is therefore the preferred method for readout of slow monitoring data from detector components.

---

<sup>5</sup>CAN in Automation, <http://www.can-cia.org/>

**OPC-UA** Open Platform Communications Unified Architecture (OPC-UA) is the next generation of the Open Platform Communications protocol [56]. It is used to exchange complex data between distributed industrial automation systems. In contrast to the previous version, the new protocol version is platform independent and can use a binary protocol over TCP/IP.

OPC-UA offers extensive possibilities to describe and model the shared data. A client can send a discovery request to only a limited part of the data model and receive information about a subset of the available structure. To receive data, the client can either poll data from the server or subscribe to data changes. Connection failures are automatically recognized by both the client and the server.

Since it uses Ethernet connections, it is not usable in an environment with strong magnetic fields. However, OPC-UA has been selected as the new standard for ATLAS DCS middleware [57].

**DIP** The Data Interchange Protocol (DIP)<sup>6</sup> is used by ATLAS to connect to external control systems such as for the magnets [52]. It is a message-based communication system built on top of DIM which allows relatively small amounts of soft real-time data to be reliably exchanged between very loosely coupled heterogeneous systems.

A publisher registers its publications at a central name server. A client will search the name server for available publications and subscribe to one or more. After the client subscribed, it will continuously listen to any data changes in the publication. The data exchange is shown schematically in Fig. 3.13. An important property of the system is that connection failures are recognized and communicated to the client. In addition, connections are automatically reestablished if the network failure has passed.

**DIM** The Distributed Information Manager (DIM)<sup>7</sup> provides efficient and reliable inter-process communications across different platforms [58]. Its communication mechanism is based on the publish/subscribe method and allows for asynchronous communications and multiple destination updates. The usage of a central name server allows clients to be ignorant about the origin of the requested data before the first connection. In addition, the protocol offers the possibility to send commands from the subscribing client to the server. The underlying network communication is implemented with TCP/IP sockets.

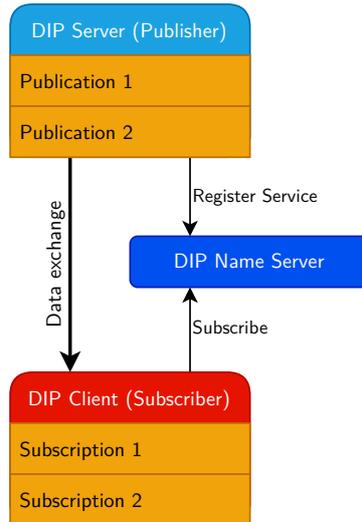
DIM is used as communication protocol for the processes of the FSM.

---

<sup>6</sup><https://dip.web.cern.ch/>

<sup>7</sup><http://dim.web.cern.ch/>

### 3. The ATLAS detector



*Figure 3.13.:* Data exchange between two systems using the DIP protocol.

#### 3.4.3. Software

The DCS Back-End runs different software components to receive information from the Front-Ends, process the information and make the information available to the operator in the control room. It also makes it possible to control the active DCS hardware.

#### OPC-UA servers

Since the creation of the quasar framework [59], a tool for efficient development of OPC-UA servers is available. By providing a server design file in XML format that describes the object-oriented information model of the target system or device, the quasar framework can generate an executable OPC-UA server application.

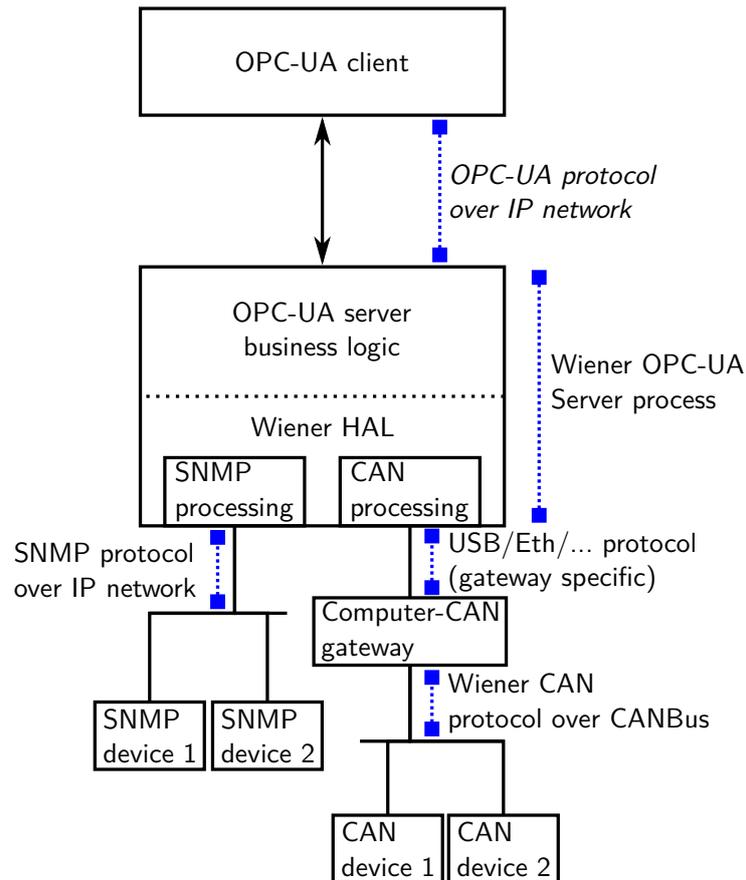
The OPC-UA server does not have to run on the target system, but can connect to it for example via CAN or over SNMP. The server may also perform conversion from raw to physical values.

A configuration file that is read by the server during runtime defines the names of all data sources which a connected OPC-UA client can use to read from and write data to.

One example is the Wiener OPC-UA server. Its usage is shown in Fig. 3.14. The server connects the Wiener power supply<sup>8</sup> to the SCADA system. Typically, the power supplies are connected through Ethernet using the SNMP protocol.

---

<sup>8</sup>power supplies by the W-IE-NE-R Power Electronics GmbH, 51399 Burscheid, Germany



**Figure 3.14.:** The Wiener OPC-UA server in a control chain. [60]

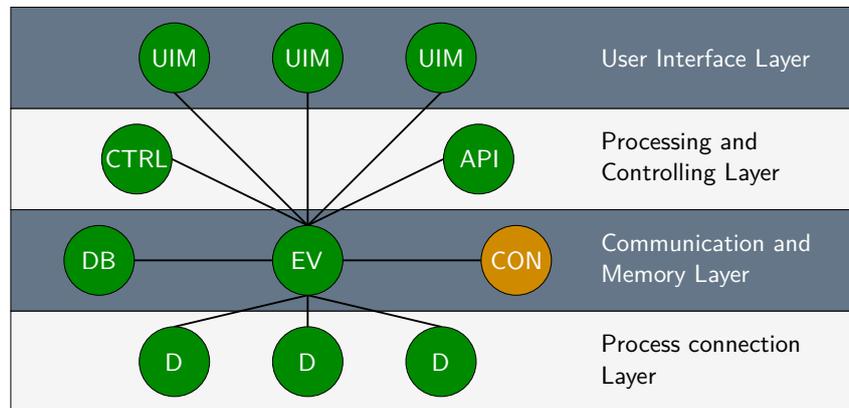
### SIMATIC WinCC Open Architecture

After a comprehensive market survey, WinCC OA<sup>9</sup> (formerly PVSS) was chosen as the Supervisory Control and Data Acquisition (SCADA) system deployed at CERN [52]. WinCC OA is a software package designed for the use in automation technology to monitor and control a system. WinCC OA is data point based and event-driven. This means that all current information is stored as data points in a database and value changes are handled by an event-based approach using multi-threaded callback routines. The software provides a scalable graphic user interface and can support redundant and distributed systems. A distributed system means that a WinCC OA project can run with its “managers” (processes) on one computer and can connect to other projects on other computers via a LAN connection. Projects then can access the data from all connected systems. The “OA” in the software package’s name stands for Open Architecture, which means, that WinCC OA offers a C++ API to connect custom software to the system and extent the functionality. This is, for example, done with the DIP and DIM software components.

<sup>9</sup><https://www.winccoa.com/>, developed by ETM professional control GmbH

### 3. The ATLAS detector

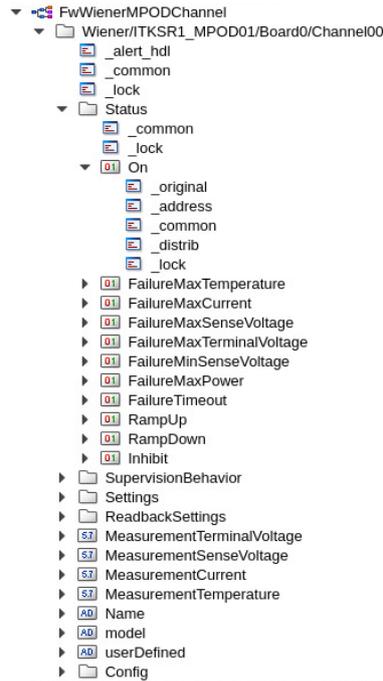
Figure 3.15 shows the managers that are employed in an example WinCC OA project. The only mandatory managers are the database manager (DB) and the event manager (EV). Other processes can be started when needed. For development of scripts and panels, an Integrated Development Environment (IDE) called Gedi is provided. Scripts use a C-like scripting language called “Control” and are executed by Control managers (CTRL).



**Figure 3.15.:** The manager/layers of a WinCC OA project. The distribution manager (here marked with CON), handles the connections in a distributed system and exchanges data point information. The drivers (marked with D) can be used to connect to hardware. For example, WinCC OA offers a built-in OPC-UA client. Control managers (CTRL) are used to run control scripts in the background and UI managers can display panels. Via the API interface, one can add custom archive managers, which write the data points into persistent storage. (From WinCC OA documentation)

Panels are run in a User Interface Manager (UIM) and can display the current state and values of monitored parameters and offer the possibility for control. WinCC OA offers a built-in OPC-UA client as driver (D) to connect to hardware. For users, an interface to the database is provided with the Para program. Para shows the data points, which are structures to store data in SCADA systems. Several data points can have the same type. The data point type includes the layout of data points. Then all data points are “instances” of a data point type and data point elements are “member variables”. Figure 3.16 shows how the data points and data point elements are displayed in Para. Via a peripheral address configuration (`_address`), the connection to the hardware is established.

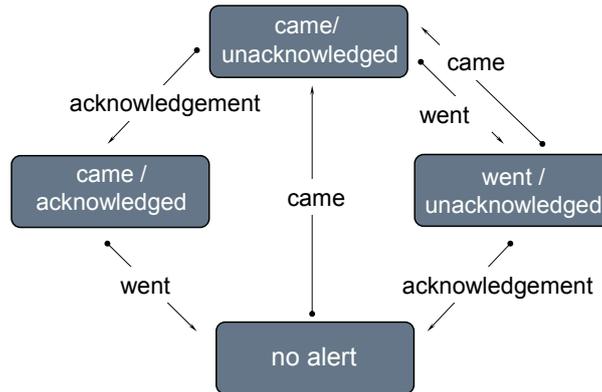
Data point elements are addressable via their name or via a freely selectable alias. This can be used to implement a hardware view, which describes the *hardware* that can be controlled, and a detector view, which describes the *usage* of the hardware in the detector. Data point elements can be configured with an alert handling with different severity levels. One can define threshold above (or below) which alerts should be triggered. The control flow for the alert handling is shown in Fig. 3.17.



**Figure 3.16.:** Example for data points in Para. At the very top is the data point type. Below, one data point of this type is shown with its data point elements. The data point name is usually derived from the hardware it represents. For a better understanding in which way the hardware will be used, one can assign aliases to data points or data point elements.

### JCOP framework components

To help with the development of WinCC OA panels and scripts (and because many functionalities are necessary for all LHC experiments), JCOP provides a framework for WinCC OA. Within the framework, it offers components and libraries for hardware access and control, especially for devices used commonly at CERN, as well as components for access control, alarm tools, trending tools and an FSM library to model the hierarchy of the control system.



**Figure 3.17.:** Alert handling for data point elements in WinCC OA. The state of the alert is shown in rectangles. “Came” and “went” transitions happen, when the condition for the alert is fulfilled or not. A user has to acknowledge the alert. Several severity levels for the alert can be defined.

## 3.5. ITk Pixel DCS

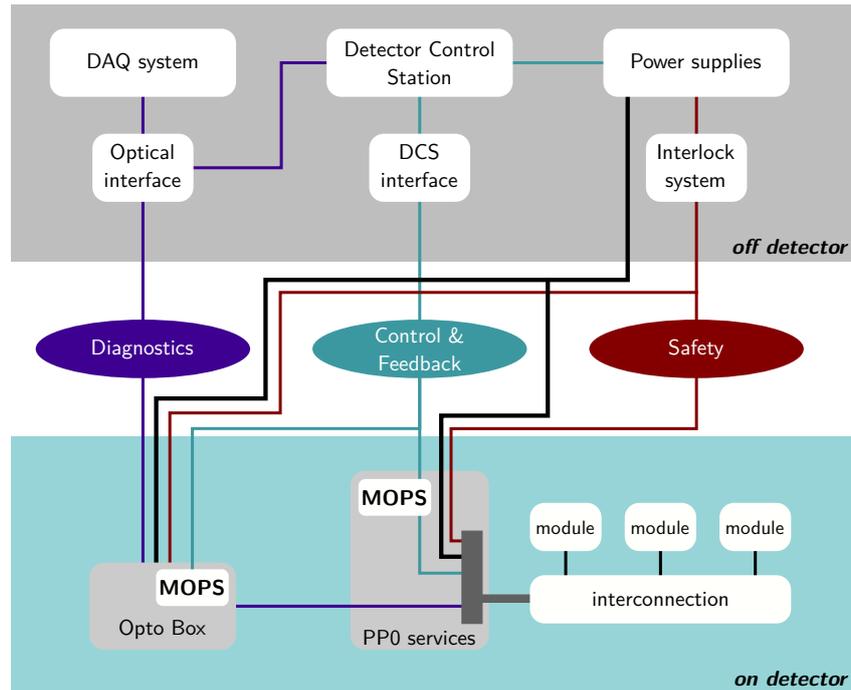
The concept of the Pixel DCS is driven by the needs of the serial powering. Four to fourteen modules are connected together into an SP-chain and powered by the same current source. In total there will be around 1000 SP-chains. Additionally, the Opto Boxes, which house the optical transceivers and the ASICs handling the data transfer on Opto Boards, require monitoring and control. About 300 Opto Boxes are foreseen for the ITk Pixel subsystem.

To ensure safe operation and reliable data taking, the Pixel DCS consists of three independent paths: Diagnostics, Control and Feedback, and Safety. This concept is shown in Fig. 3.18. The three paths are described in more detail in the following.

### 3.5.1. Control and Feedback path

The Control and Feedback path contains the power supplies for the SP-chains and the Opto Boxes. Running in serial powering means that no single module can be turned on or off but only an entire SP-chain can be controlled together. Detailed connection schemes for the SP-chain are shown in the Appendix. The control of the SP-chain includes one LV channel from a current source that powers the FE chips and two to four HV channels for depletion of the sensors. Inside the Opto Box, all Opto Boards associated to one SP-chain can be controlled together.

For the monitoring of temperatures and voltages of modules in an SP-chain as well as temperatures and voltages inside an Opto Box, the DCS connects to the Monitoring Of Pixel System (MOPS). The MOPS is a DCS ASIC developed specifically for the Pixel detector [51]. It is used for aggregation of on-detector monitoring data. There will be one



**Figure 3.18.:** Overview of the paths for the ITk Pixel DCS system. The modules are connected into an SP-chain. One module per chain has its NTC routed to the interlock system. Also NTCs inside the Opto Box are connected to the safety path. (Adapted from [61])

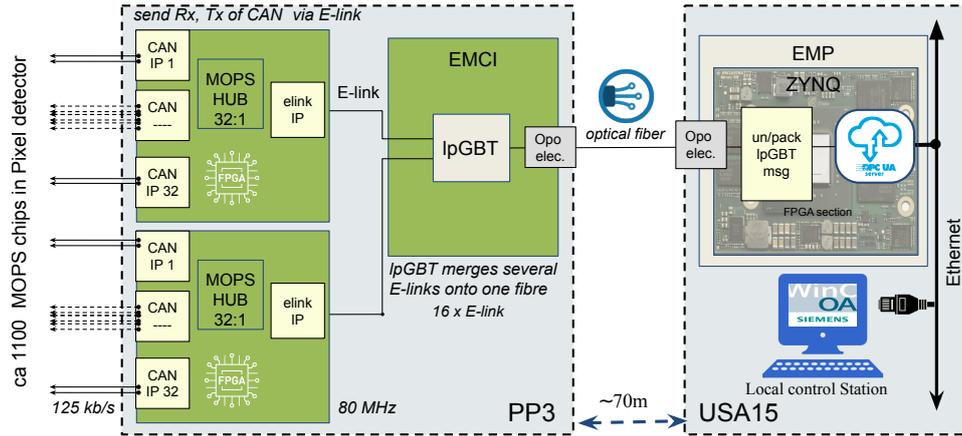
MOPS assigned to each SP-chain. For communication with the outside world, the MOPS uses a CAN bus. The data is received at PP3 by the MOPSHUB, collected there and sent to the counting room as shown in Fig. 3.19. The power for the MOPSHUB is independent of the power for the MOPS.

The Control and Feedback path is the slow control path and required during all types of operation. This path needs to be functional for the FSM to determine the state of the Pixel detector.

### 3.5.2. Safety path

The safety path comprises of the hardwired interlock system to protect the detector and humans against risks. It requires the highest reliability and runs permanently. To protect the detector against heat-ups, temperature sensors on the SP-chain and in the Opto Boxes are needed for monitoring, and are connected directly to the interlock system to ensure independence from the Control and Feedback path [63].

### 3. The ATLAS detector



**Figure 3.19.:** The MOPSHUB topology foreseen at PP3. Up to 8 MOPSHUB connect to one EMCI (Embedded Monitoring and Control Interface) which sends the aggregated data to an EMP (Embedded Monitoring and control Processor) in the counting room. (Taken from Ref. [62])

#### 3.5.3. Diagnostics path

By means of an ADC each FE chip collects monitoring data. The data contains, for example, additional temperature information, and is sent via the normal (physics) data path to FELIX, where it is separated and forwarded to DCS.

# 4. Finite State Machine

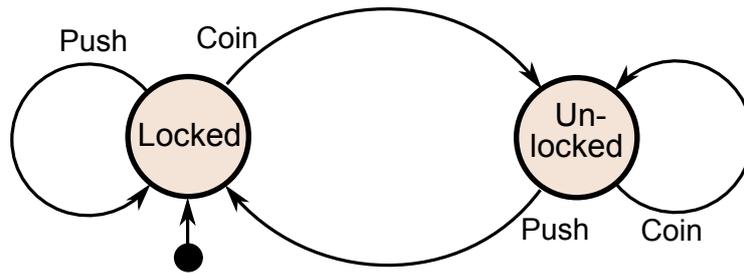
## 4.1. Theory of Finite State Machines

A Finite State Machine (FSM) is a formalism to specify the behavior of an object depending on its state [64]. It is an abstract machine that is in exactly one of a finite number of states at any given time and can change from one state to another via a transition activated because of some input. The input can be an external action and/or a reaction because some conditions are fulfilled. Common applications of FSMs are the verification of input sequences in code locks, the sequence of stops for elevators or the control of traffic lights. Control systems are typically modeled as FSMs with output. The current state can be determined by many input parameters, and depending on the current state of the system, different control actions are automatically performed or made available to an operator. This makes it possible for non-experts to control large systems.

The whole FSM can be described by the set of possible states, the initial state, the inputs which the FSM recognizes and the possible transitions, and the possible actions which are available in each state. Graphically, this can be represented by a directed graph called *state diagram*. An example for such a diagram is given in Fig. 4.1, where the two states of a turnstile and the possible transitions are shown.

Two basic types of FSMs with output exist [65]. In the Moore machine, the output depends only on the current state. In the Mealy machine, the output depends on both an input and the current state. Both types are equivalent, meaning that one can transform one machine into the other. The Moore machine is easier to understand, because of the independence on input variables. However, it usually requires more states than the corresponding Mealy machine. In practice, often a mix of both is used.

For more complex systems, simple FSMs can be nested or set up in a hierarchy.



**Figure 4.1.:** State diagram for an FSM modeling a turnstile. The turnstile has two states, “Locked” and “Unlocked”. Inserting a coin transitions the turnstile into the Unlocked state, where the bar can turn. A person pushing through the turnstile changes the state back to Locked, such that the next person cannot get through the turnstile without inserting a coin again. The arrow into the Locked node from the black dot indicates the initial state. (Chetvorno, CC0, via Wikimedia Commons)

## 4.2. The State Manager Interface

The State Manager Interface (SMI++)<sup>1</sup>, is a framework for designing and implementing distributed control systems [66]. It implements hierarchically structured objects with event-driven and rule-based FSM logic. A complex system can be decomposed into smaller, manageable entities.

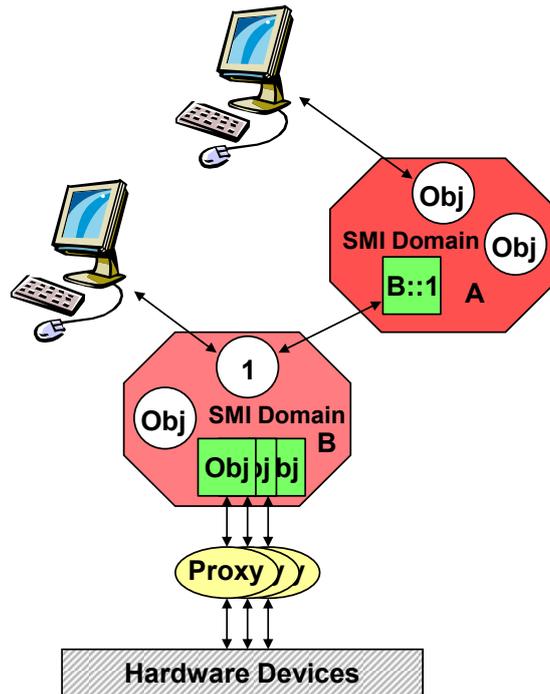
Objects can represent “concrete” entities, such as power supplies or temperature sensors, or abstract entities, like control systems of a detector. For state determination of concrete entities, objects interface through a proxy (normally WinCC OA) with the hardware. Abstract objects have their logic implemented fully within SMI++.

Logically related objects can be grouped together in a SMI domain. These domains run as separate, (possibly) distributed processes. Their communication is handled via the Distributed Information Manager (DIM) protocol. The SMI++ run-time environment is shown in Fig. 4.2.

The object model for abstract objects is described using the State Manager Language (SML). This language allows detailed specification of the class of an object such as its states, actions and rules, which when fulfilled, trigger asynchronously a change of state or the execution of an action. SML contains the code that implements the available actions. The language offers a small set of simple instructions. The most common one is the *do*

---

<sup>1</sup><https://smi.web.cern.ch/>



**Figure 4.2.:** SMI++ Run-time Environment. The quadratic objects are of type associated. These are different in that they cannot execute any action: a command received by an associated object is passed through. Conversely, a state change occurring in the entity controlled by the associated object is faithfully reflected. To address an object from a domain, “domain::object” is used. (Adopted from Ref. [67])

instruction to send commands to other objects, the *if* instruction to test the state of an object, and the *move\_to* instruction to end an action and move to a new state.

Objects, concrete or abstract, can be grouped into “object sets” to ease the manipulation of a large number of objects. It is possible to dynamically include or exclude objects from an object set.

The creation of the SML description for the experiment is commonly done by a WinCC OA script offered by the Controls Hierarchy framework component.

### 4.3. The Controls Hierarchy framework component

The Controls Hierarchy framework component for WinCC OA, named FwFsm<sup>2</sup>, provides a generic, platform-independent implementation of a hierarchical state machine toolkit utilizing SMI++ [67]. It offers tools to define FSM objects for an experiment, structure them in a hierarchy tree, generate the SML description, start and stop the SMI++ run-time

<sup>2</sup><https://lhcb-online.web.cern.ch/ecs/fw/FwFsm.html>

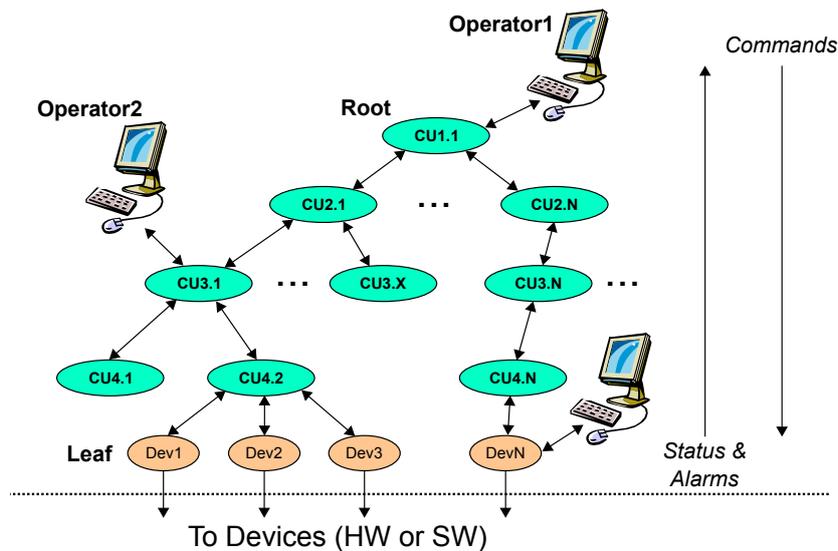
#### 4. Finite State Machine

and interact with the FSM objects from WinCC OA control scripts. WinCC OA is also used as a proxy for concrete FSM objects. This is possible, since WinCC OA can be expanded using the provided C++ API.

Not only concrete FSM objects are interfaced, however. The attributes of any instantiated FSM object inside a SMI domain are made persistent as data points in the associated WinCC OA project. This allows for archiving of the FSM states and transitions by WinCC OA, and communicating the situation of the experiment to operators and handing them control over the FSM via Graphical User Interfaces (GUIs).

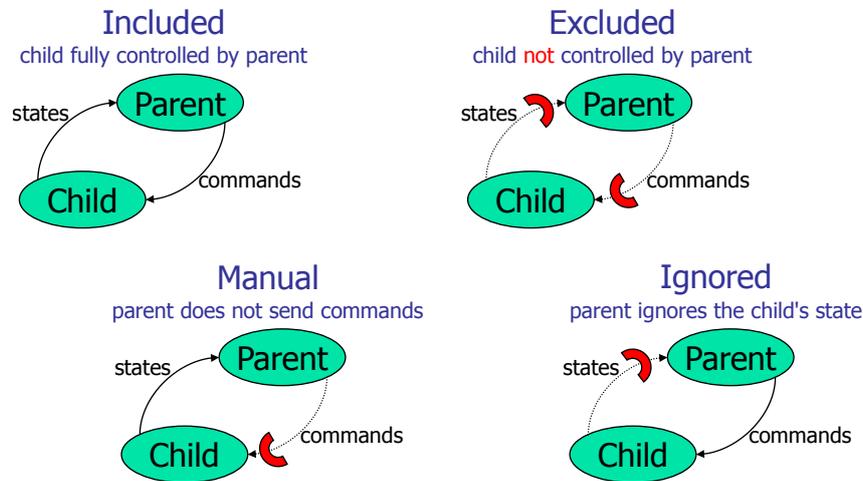
A generic FSM tree created by the FwFsm component is shown in Fig. 4.3. Commands are propagated down the tree, while a status is propagated upwards. The FwFsm component uses the following terminology for the different types of FSM object classes:

1. Device Unit (DU): These are the leaves of the FSM tree and correspond to concrete physical objects (e.g. a power supply channel) represented as a data point in WinCC OA. The state logic runs in a Control Manager in WinCC OA, which is also used as the proxy to interface the hardware with the SMI++ run-time.
2. Logical Unit (LU): These are abstract objects (e.g. a system, that groups several power supply channels). LUs run within a SMI domain process.
3. Control Unit (CU): These are also abstract objects, and the main difference to LUs is that different CUs can be controlled by different operators. One CU corresponds to one SMI domain process.



*Figure 4.3.:* Generic architecture of an FSM tree. [67]

Control Units in a system can be used in different partitioning modes to enable monitoring or control of only a subsystem independently from the rest. The possible partitioning modes are shown in Fig. 4.4.



**Figure 4.4.:** Partitioning Modes of Control Units. For each of the four partitioning modes it is shown how states and commands can propagate between a parent and a child Control Unit.

## 4.4. ATLAS specific extensions

ATLAS decided to extend the functionality of FwFsm with the fwFsmAtlas component. The partitioning modes obtain an additional attribute: ownership. Only the owner of an FSM (sub-)tree can decide if the ownership is exclusive or shared to allow control by others.

The ATLAS FSM component also introduces a second attribute to the FSM objects. Each object then has two attributes, the *state* and the *status*. The *state* represents the operational state of the object, while the *status* represents the safety status using alerts. The *status* attribute can have one of the following values or severity levels: “OK”, “WARNING”, “ERROR” or “FATAL”. Each value is associated with a specific color. Both the state and the status are propagated upwards independently, meaning that alerts do not overshadow the operational state.

For the operator in the ATLAS control room, a dedicated user interface is available. Each FSM object has a panel associated to it. Fig. 4.5 shows an example of the FSM screen and its modules. On the left-hand side are the navigation module and FSM module to freely navigate through the FSM tree and show the current object with its state and status, as well as all children with their respective states and statuses. Here, sending of

#### 4. Finite State Machine

commands and partitioning actions can be performed. Below is the secondary module to show a second FSM object independent from the main module in the middle of the panel supplying the operator with detailed information about the selected unit. At the top is the problem list, a table collecting all alerts in the system and their time stamp. Clicking on an alert navigates the main module to the origin of the alert to enable investigation of the problem.

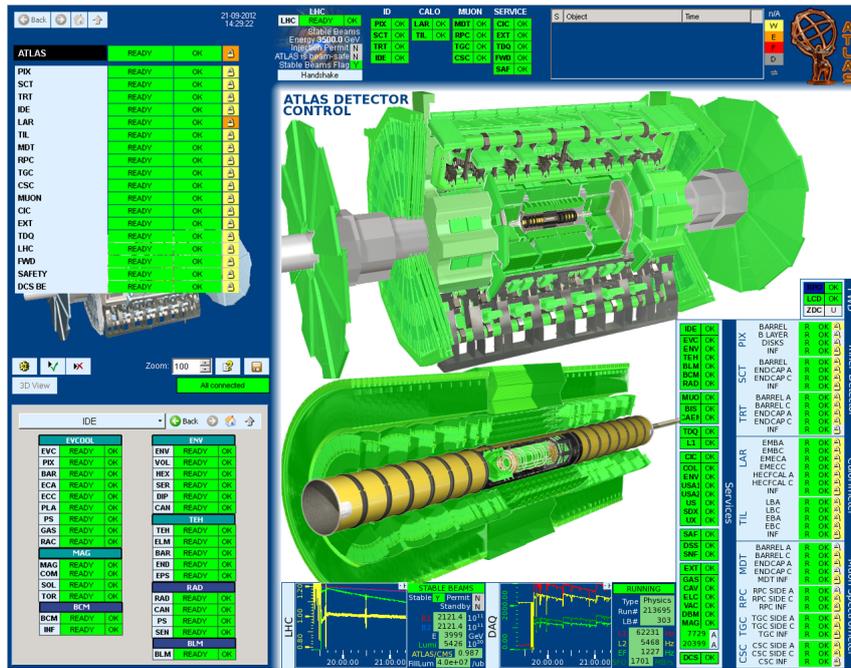


Figure 4.5.: ATLAS DCS user interface during a regular LHC fill for luminosity production with  $pp$ -collisions. It shows an overview of the state of all detector components (© ATLAS).

## 5. The DCS for the ITk Pixel OB demonstrator

For system tests with the OB demonstrator, a control system is necessary. Having an extensive DCS available simplifies the workflow and enables the operator to concentrate on the tests at hand. In addition, valuable experience can be gained early on in terms of what functionality is important for a final control system, even if the requirements are slightly different. While the demonstrator should imitate closely the conditions of the final detector, there will be differences in the setup to the final design. From the DCS point of view, the diagnostics path as described in Section 3.5 and shown in Fig. 3.18 will not be ready yet. Concerning the safety path, the interlock system will comprise an old Interlock Matrix Crate (IMC) of the IBL detector. The control and feedback path is the most complete and is where the current testing program is most active. It will be used to control several SP-chains. The Outer Barrel demonstrator setup will be close to the representative setup with respect to the powering scheme of an SP-chain. The physics readout of the detector modules is realized through an Opto Box which houses the Opto Boards that convert the electrical signals to optical ones. Up to two Opto Boards can be associated with an SP-chain. The Opto Board needs to be powered to allow readout. Both the Opto Box and the SP-chains are monitored by one MOPS chip. While it is foreseen to have the monitoring chips read out by the MOPSHUB in the final detector, an interim solution called MOPSHUB4Beginners was developed for the demonstrator.

Over the course of the work on this thesis, the power supplies, the MOPSHUB4Beginners and the Opto Box were installed. The IMC was configured and is ready to be used in the setup. Due to delays in production, no modules were yet delivered to connect to full SP-chains and mount on support structures.

In this chapter, the user interfaces and monitoring tools that have been developed in the context of a WinCC OA project for the purposes of providing a DCS for the demonstrator will be described. In addition, the design of the DCS as a Finite State Machine is discussed.

## 5.1. Requirements and the SR1 setup

The high-level setup of the OB demonstrator and the main goals of the system test were described in Section 3.3. While the modules for the SP-chains are not delivered yet, the off-detector service chain is almost fully commissioned. The service chain follows the service specifications of the ITk Pixel detector [49], as previously shown in Fig. 3.10, closely in terms of functionality.

The demonstrator is set up in the radiation laboratory of the SR1 building at CERN. The power supplies for the detector modules are placed in the rack area, and long Type-3 cables of approximately 65 m carrying the LV and HV power run from there into the lab hall, terminating at a Cable Saver Board that replaces two detector patch panels. From there, real Type-1 and on-detector (Type-0) services on an LLS are planned. The detector parts will be set up inside a Test Box, a large lightproof box to protect the sensors from stray light and to be able to flush the equipment with dry air to operate at low humidity.

The IMC is placed next to the Test Box, and the readout of the ELMBs to monitor the state of the IMC is realized with a USB-CAN controller connected to a rack-mounted PC that runs the CANopen OPC-UA server. This PC is also employed to run the other OPC-UA servers for connecting to the power supplies and the MOPSHUB4Beginners. The MOPSHUB Crate with the MOPSHUB4Beginners is located in the same rack next to the Test Box.

The Opto Box is monitored by a MOPS and powered by a power supply in the radiation laboratory. All power supplies can be interlocked by the IMC. The IMC can also act on the CO<sub>2</sub> cooling plant that is used to cool the detector modules.

All information are collected by WinCC OA projects running on a dedicated PC in the rack area. For the operator, a client PC was set up next to the Test Box to open the GUIs of the main WinCC OA project and supervise the demonstrator from there.

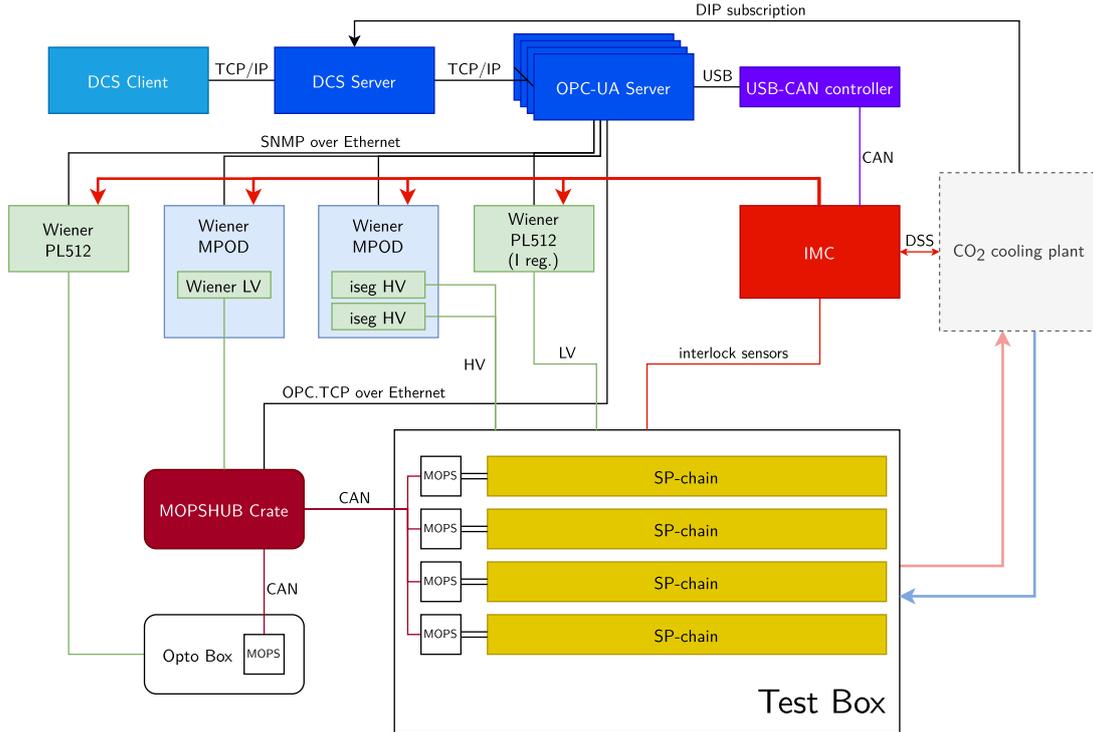
An overview of the complete setup from a DCS point of view is given in Fig. 5.1. The software infrastructure consisting of the OPC-UA servers and DCS server should provide every functionality needed to operate the demonstrator. The hardware infrastructure can be separated into four main groups: a power supply system, a monitoring system, an interlock system and a cooling system.

All power supplies must have individually floating outputs, be designed in a way that allows the interlocking of individual channels and provide connection using an OPC-UA server [68]. The inventory of available power supplies for the demonstrator is summarized in Table 5.1. For the biasing of the sensors, two iseg<sup>1</sup> HV modules of type EHS F607n-F<sup>2</sup>

---

<sup>1</sup>iseg Spezialelektronik GmbH, 01454 Radeberg, Germany

<sup>2</sup><https://iseg-hv.com/en/products/detail/EHS>



**Figure 5.1.:** Overview of the ITk Pixel OB demonstrator DCS setup. The DCS server runs the WinCC OA projects for the controlling and the monitoring of the demonstrator. Another PC runs the OPC-UA servers to enable connection with the hardware. For the operator, a DCS client PC next to the Test Box is set up where the GUIs can be opened.

are installed inside a Wiener MPOD Mini Crate<sup>3</sup>. One iseg HV module offers 16 channels with up to 700 V output. The MPOD Mini Crate is equipped with a MPOD controller card to provide Ethernet connection.

The FE chips are powered by a current source. All modules of an SP-chain will be connected in series and powered by one LV channel. The demonstrator uses a current regulated Wiener prototype in PL512 form factor for this.

The Opto Boards inside the Opto Box require a standard voltage regulated power supply. A Wiener PL512<sup>4</sup> was selected for this task. The PL512 offers 12 DC channels and a connection for sensing lines.

For the power of the MOPSHUB4Beginners and the MOPSS via a CAN bus cable, a Wiener LV MPOD module of type OMPV.8060 is used<sup>5</sup>. All power supplies offer TCP/IP communication, are DHCP capable and come with a simple web server. For integration

<sup>3</sup><https://www.wiener-d.com/product/mpod-mini-crate/>

<sup>4</sup><https://www.wiener-d.com/product/pl512-power-supply-system/>

<sup>5</sup><https://www.wiener-d.com/product/mpod-lv-module/>

## 5. The DCS for the ITk Pixel OB demonstrator

**Table 5.1.:** Available power supplies and their usage for the demonstrator.

PSU	Channels	Nominal Output	Type
iseg HV module <sup>i</sup>	4/16	150 V	HV
iseg HV module <sup>i</sup>	4/16	150 V	HV
Wiener PL512 ( <i>I</i> reg.)	4/4	6 A	LV
Wiener PL512 ( <i>U</i> reg.)	11/12	9 V/1.2 V	OPTO
Wiener LV module <sup>ii</sup>	2/8	15 V	MOPSSYS

<sup>i</sup> type EHS F607n-F    <sup>ii</sup> type OMPV.8060

into a SCADA system, OPC-UA servers are available from JCOP<sup>6</sup>. With this, one can set remotely output voltages and thresholds on voltages and currents, as well as monitor the actual output. The power supplies were connected to the General Purpose Network (GPN) of CERN and configured with a static IP address.

For the monitoring system, a MOPSHUB4Beginners is used. The MOPSHUB4Beginners is a Raspberry Pi, which will connect via a CAN bus to several MOPS chips and is able to receive data from them. For the readout from the DCS, the Raspberry Pi runs an OPC-UA client and is connected to the LAN via Ethernet. The client updates the MOPS information in an OPC-UA server that was developed using the quasar framework [69]. From there, the information can be read by WinCC OA.

For the interlock system, the IMC of the old IBL detector is used. To receive information about the state of the IMC, ELMBs monitor the input and output signals of the interlock matrix [70]. All ELMBs are connected with one CAN bus which can be monitored by a Systec USB-CAN controller<sup>7</sup>. The Systec controller is linked to a computer via a USB connection. On the computer, a CANopen OPC-UA server<sup>8</sup> provides the ELMB information via OPC-UA to WinCC OA. In contrast to the Wiener OPC-UA server, which can run in hardware discovery mode to scan the connected power supplies for their available properties, the CANopen OPC-UA server needs a full configuration file right from the start. More detailed information about the MOPSHUB4Beginners and the IMC are given in later sections, where also the corresponding GUIs are explained.

The FE chips and sensors generate a significant amount of heat and must be kept at a low temperature, which requires them to be actively cooled. The CO<sub>2</sub> cooling plant in SR1 will be used for this purpose [71]. The operational parameters of the cooling plant cannot be set via WinCC OA, but are given by the cooling team. The project can only

<sup>6</sup><https://jcop.web.cern.ch/opc-hardware-simulation>

<sup>7</sup><https://www2.systec-electronic.com/en/products/interfaces-gateways-plcs/sysworxx-usb-canmodul16/>

<sup>8</sup><https://twiki.cern.ch/twiki/bin/view/AtlasPublic/DcsCanOpenOpcUa>

receive status and monitoring information via a DIP subscription. However, the IMC can interact with the cooling plant via a dedicated Detector Safety System (DSS) line. In case the cooling fails, the power to the modules has to be switched off to prevent them from overheating. In the case that the humidity in the Test Box is too high, there is the danger of condensation and icing on the sensors. If this happens, the cooling is switched off.

The users of the demonstrator will be interested in all the details of the detector hardware and infrastructure needed to operate the demonstrator, for example if there are errors on the control side which need debugging outside normal operation. Therefore, the DCS for the demonstrator is maybe providing more detailed control than a DCS for a real detector. During normal operation, however, the highest interest lies in knowing the operational state of the different detector parts. Hence, the DCS for the demonstrator was created with panels that were built with the detector properties in mind and with state calculation of the detector parts via an FSM. The detector parts can be structured hierarchically to provide high-level control. For the system test, attention was more given to the panels in the lower levels of the hierarchy as users will be experts on detector hardware. The usage of an FSM ensures a consistent look-and-feel through the provided graphical framework and greatly simplifies the operation.

In comparison to a full and final detector, the system test setup is constantly in a state of flux. As new hardware gets integrated or old hardware upgraded, the DCS has to be flexible enough to adjust to the setup. However, with the demonstrator being close to the final detector in functionality, possible solutions for a final DCS can be explored with this prototype.

## 5.2. System overview and project setup

For the connection of the DCS to hardware, four OPC-UA servers run on a CentOS 7 machine: one server is used for the Wiener PL512 power supplies, one for the Wiener MPOD Mini Crates, one for the connection to the ELMBs via CAN, and one server for the connection to the MOPSHUB4Beginners. All OPC-UA servers are running as systemd services to enable automatic startup. This is useful after a power cut, for example, when the computer is configured to automatically start as soon as power is restored. With this, all OPC-UA servers will be back in operation after such scenarios without manual intervention.

A dedicated rack-mounted DCS PC is available to run the WinCC OA projects that will be used to control and monitor the demonstrator. The PC runs CentOS 7 and WinCC OA in version 3.16 was installed. During development of the WinCC OA project and the FSM for the demonstrator, the general ATLAS guidelines for DCS [72] and FSM [73] were

## 5. The DCS for the ITk Pixel OB demonstrator

taken into account. For the specifics of the ITk detector and HL-LHC upgrade of ATLAS, the recommendations described in Ref. [61] were followed.

There are two WinCC OA projects running on the DCS server machine, as well as a DIM name server (DNS). The name server is necessary for the SMI domains of the FSM to communicate. Both WinCC OA projects and the name server are configured as systemd services to start automatically after boot. The main WinCC OA project contains the created panels, libraries and scripts, and is used to configure and run the FSM. A second project establishes the connection to the MOPSHUB4Beginners, retains its information via data points and includes detailed panels for the MOPSHUB4Beginners. Both projects are connected in a distributed system with unique system IDs to exchange information.

The main project makes heavy use of JCOP framework components that help integrating the hardware information into WinCC OA with predefined data point types and setting of the correct peripheral address in the data points. There exist components for the ELMBs, the Wiener PL512 power supplies, the Wiener MPODs and for DIP connections, which all add their own device manager to the WinCC OA console.

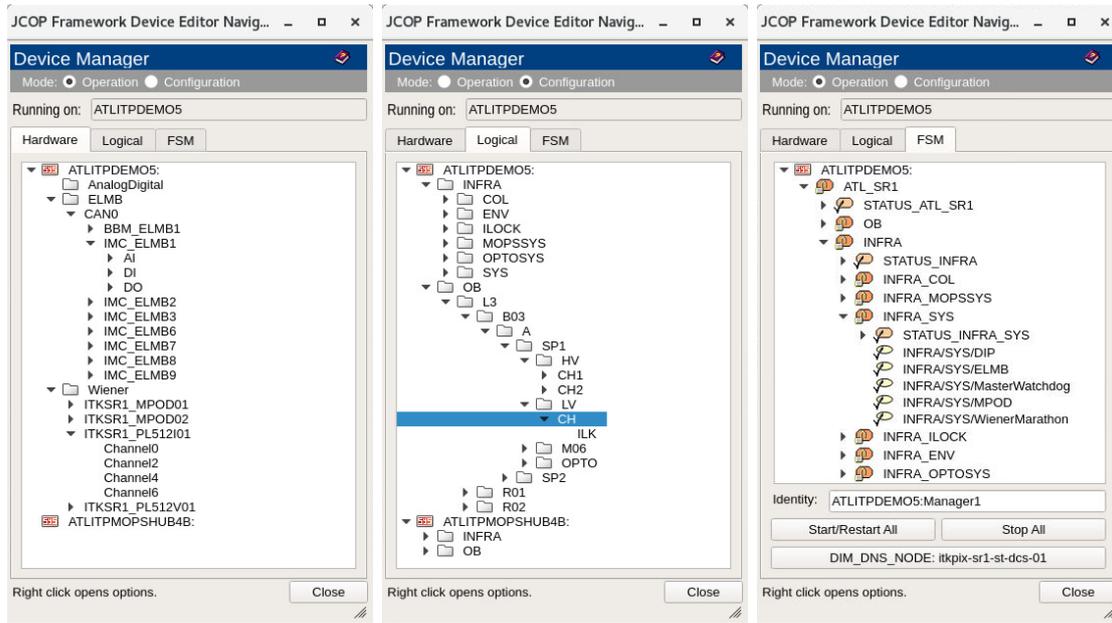
The main development tool for hardware integration is the Device Editor and Navigator. This tool can be used to integrate all hardware devices and create a “hardware” tree of the demonstrator that displays the available data points. The procedure to set up a full system that is easily controllable is to use the Device Editor and Navigator in a first step, creating a “logical” tree that represents the logical structure of the system by assigning aliases. In a second step, panels that are associated with certain levels of the hierarchy are created. In a last step, an FSM control layer is implemented. This was done for this project and the result is shown in Fig. 5.2.

The hierarchical design, both in data point names and aliases, is realized by using the slash (“/”) as hierarchy separator per JCOP convention. The first HV channel of an SP-chain (from a Wiener MPOD) can then have an alias such as:

$$\text{OB/L3/B03/A/SP1/HV/CH1} \quad .$$

Here it is indicated that the SP-chain is the first SP-chain on the A-side of the third longeron in layer 3 in the Outer Barrel region. This naming scheme loosely follows the official names in Ref. [74], but adjusts for the limitation and requirements inside the WinCC OA project.

The mapping of an alias to a data point name represents the cabling that is used for the demonstrator. The mapping shows, for example, which power supply channel is connected to which detector element. In case this cabling changes, it should be possible to implement the corresponding changes immediately in WinCC OA. For this, a panel was created to



(a) Hardware Tree

(b) Logical Tree

(c) FSM Tree

**Figure 5.2.:** The Device Editor and Navigator showing the three different trees of the project. Single operation panels can be opened from the hardware and logical view. The FSM can be started from the FSM view and the corresponding GUI can be opened from there.

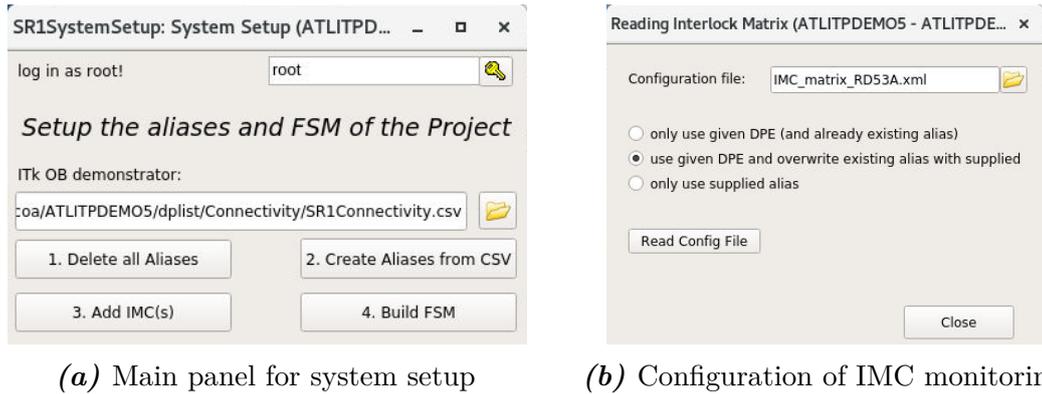
read the connectivity information from a CSV file and write the aliases into the correct data points. The configuration of the IMC monitoring needs more detailed information, which is why in this case a XML file is read, and the information is not only stored as aliases, but also in a dedicated interlock matrix data point. The system setup panels are shown in Fig. 5.3. More details about the setup of the IMC monitoring are given in Section 5.4.2.

The creation of the logical tree and common actions concerning aliases are supported by an alias handling library. This library was written to collect frequently used functions on aliases in one place. When writing an alias to a data point, additional alias elements are also written to its data point elements. For example, the “.MeasurementCurrent” data point element of the first HV channel from the previous example will receive the alias

!OB/L3/B03/A/SP1/HV/CH1.Imon ,

where the preceding “!” symbol is used to avoid the display of this data point element in the logical tree and keep the tree clean. Assigning these additional aliases is useful when displaying the value of these data point elements and when archiving the values together with the alias.

## 5. The DCS for the ITk Pixel OB demonstrator



**Figure 5.3.:** The system setup panel to read the connectivity information from a CSV file and to build the FSM by using the available aliases in the system is shown on the left. The right shows the configuration panel of the IMC monitoring, where all necessary information is read from an XML file.

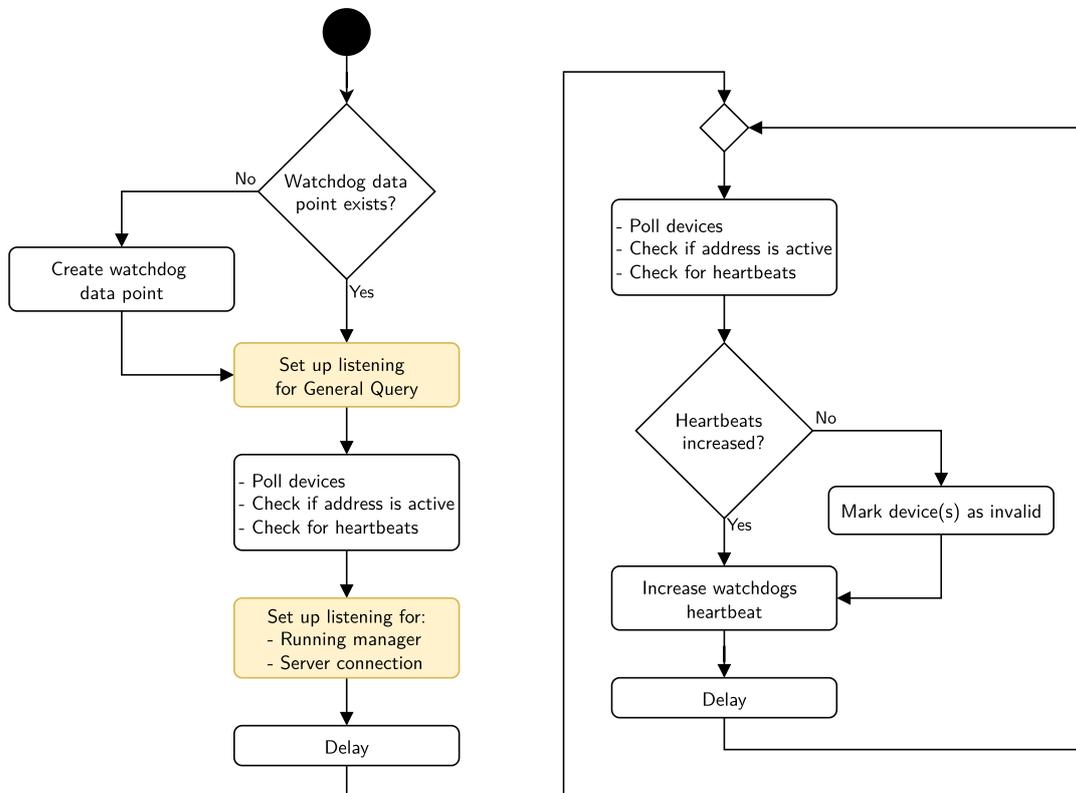
In addition, the alias handling library takes care of setting the appropriate alerts to the data points when reading the aliases from the configuration files. The logical tree in the Device Editor and Navigator shows then a “detector view” that can be already used for simple (channel-wise) control of the detector.

### 5.3. Watchdog scripts

For the monitoring of the WinCC OA projects’ connection to hardware, watchdog scripts were created. The scripts are run independently by the control managers of the project. For each hardware connection (or device manager), there is one device watchdog. This means that there is a watchdog each for the Wiener OPC-UA server connection to connect to the PL512 power supplies and the MPODs; there is a watchdog for the CANopen OPC-UA server to connect to the ELMBs of the interlock system; and there is a watchdog for the DIP connection, which receives information about the cooling plant. A master watchdog monitors the device watchdogs and summarizes heartbeats collected from all connected devices.

The watchdog control flow logic is shown in Fig. 5.4. All the information is written to a dedicated watchdog data point. The device watchdogs work by regularly checking if the peripheral addresses of data points are set to active, and if a heartbeat from the devices is received. In addition, they monitor the connection to the server, and check whether the corresponding WinCC OA client (which is a WinCC OA device manager) is running. If a problem is detected, they set the invalid bit of the affected data points. They also map status bits that might be sent by servers to the invalid bit of the corresponding data point.

If a data point has its invalid bit set, data from it will be shown in the panels with a gray background. This is to indicate to the operator that there is something wrong with the displayed data and connection problems should be investigated.



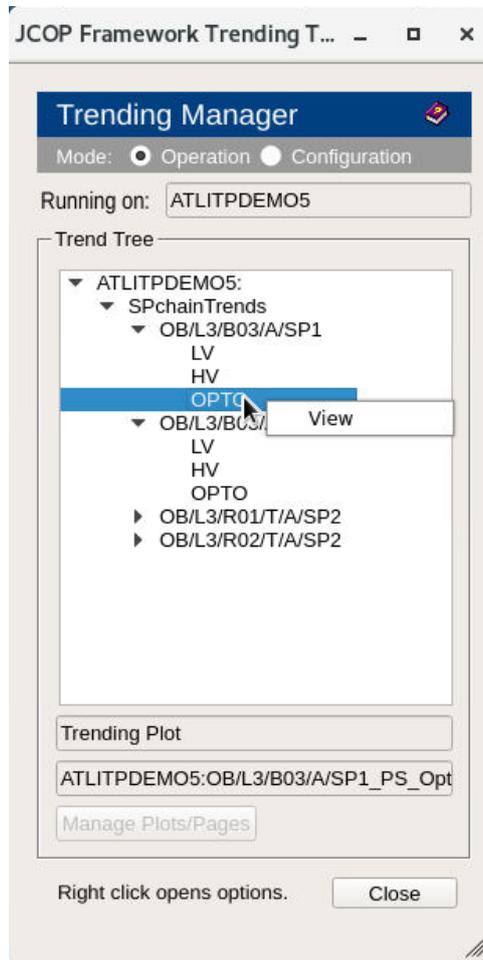
**Figure 5.4.:** Example control flow of a watchdog for OPC-UA connection. The actions with orange background are persistent actions in the sense that they create event listeners that trigger a callback function whenever an event is fired.

## 5.4. Graphical User Interfaces

The interface between an operator and the DCS BE is mainly provided by WinCC OA panels. This is a Graphical User Interface, which displays information about the detector, shows alarms and offers ways to control it. All the information that an operator might need to understand the state of the detector and to be able to debug problems should be displayed. These can be current values of sensors, parameters of devices (such as the states of power supplies) and trending plots of interesting variables over time. For many generic problems, there already exist solutions from the JCOP framework. For example, the JCOP framework offers tools to create trending plots, and an alarm screen to collect alarms raised by the system. The possibility to acknowledge these alarms is also provided.

## 5. The DCS for the ITk Pixel OB demonstrator

During project setup, some standard trends are preconfigured and made available from the JCOP trending tree shown in Fig. 5.5.



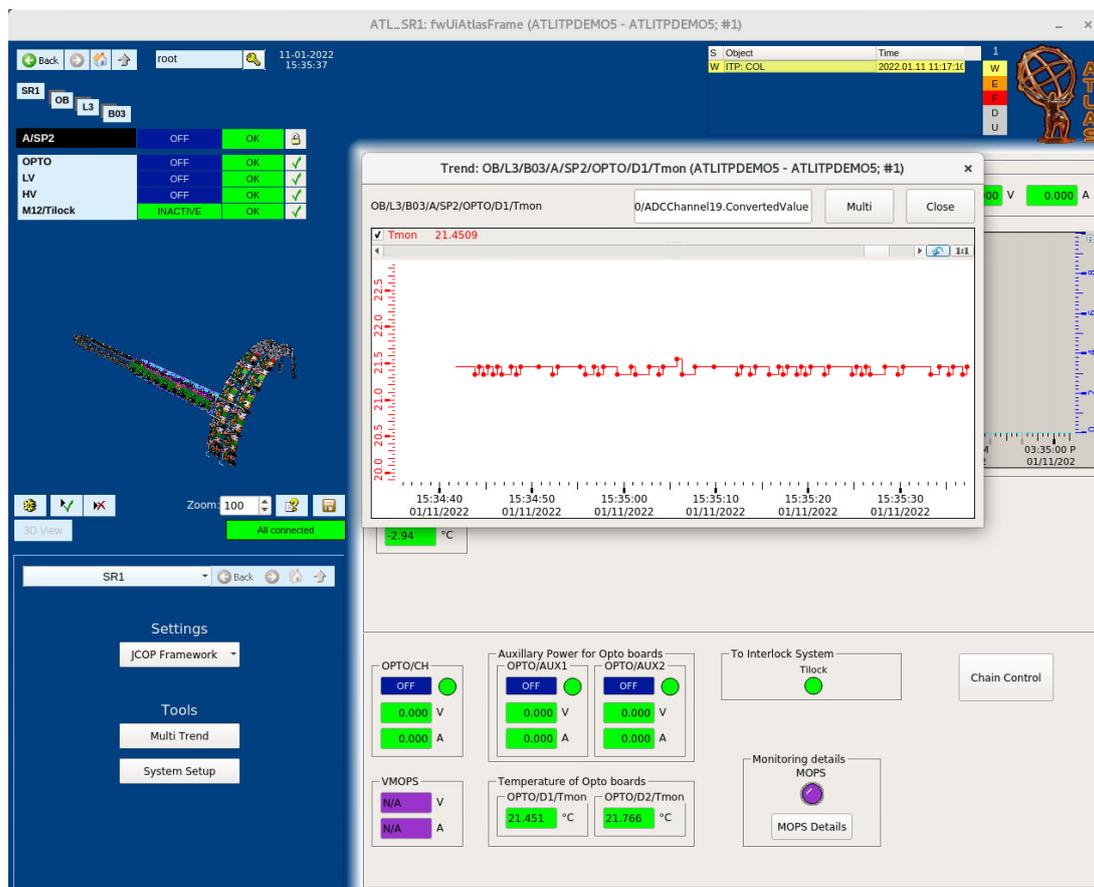
**Figure 5.5.:** JCOP trending tree with preconfigured trends for the power supplies of the SP-chains.

From the operator’s point of view, one is interested in the “detector view”, where it can easily be seen which detector parts are powered, for example. Panels should be developed with this in mind. There are three main ways to interact with information from the data points in WinCC OA to display them in a panel: `dpGet()`, `dpSet()` and `dpConnect()`. With the first function, one can obtain information stored in a data point element and the second function can be used to set a certain value manually. To display live data, the `dpConnect()` function can be used. With it, a callback function is assigned to the data point element that is called every time the value of the data point element changes. Data point elements can be addressed either by their name or by their alias.

The panels that will be shown inside the FSM GUI have a fixed size, but standalone panels can be resized. The created panels should be intuitive, and the information should

be easy to absorb on a single glance. Additional information about graphic elements can be given by using tooltips, which are small text boxes that appear under the mouse cursor when hovering over that element.

The trend panels can communicate to operators the stability of certain parameters. The single trend panel is shown in Fig. 5.6. The standard JCOP framework trend panel is limited to a maximum of eight variables that can be displayed. To mitigate this problem and to allow flexible display of different variables, a multi value trend panel was created. This offers the possibility to search for an alias or data point that should be displayed. The panel is shown in Fig. 5.7.

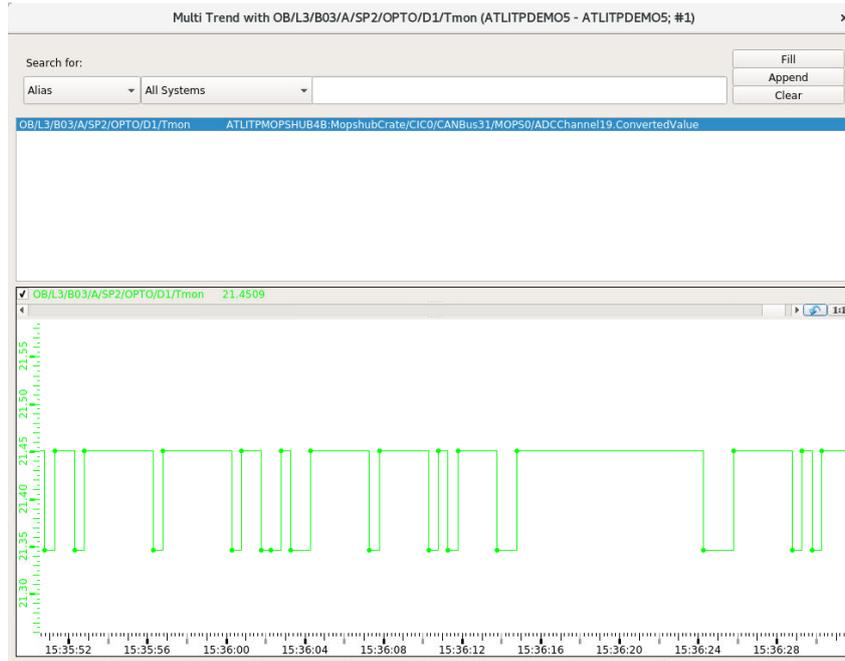


**Figure 5.6.:** A single value trend panel that opens on right-click of a monitored value and shows its time evolution.

#### 5.4.1. SP-chain monitoring and control

The SP-chain is the smallest unit that can be controlled. There are two HV channels for the sensors and one LV channel for the FE chips of the modules of an SP-chain. The parameters that need to be monitored are defined in Ref. [61] and shown in Fig. 5.8.

## 5. The DCS for the ITk Pixel OB demonstrator



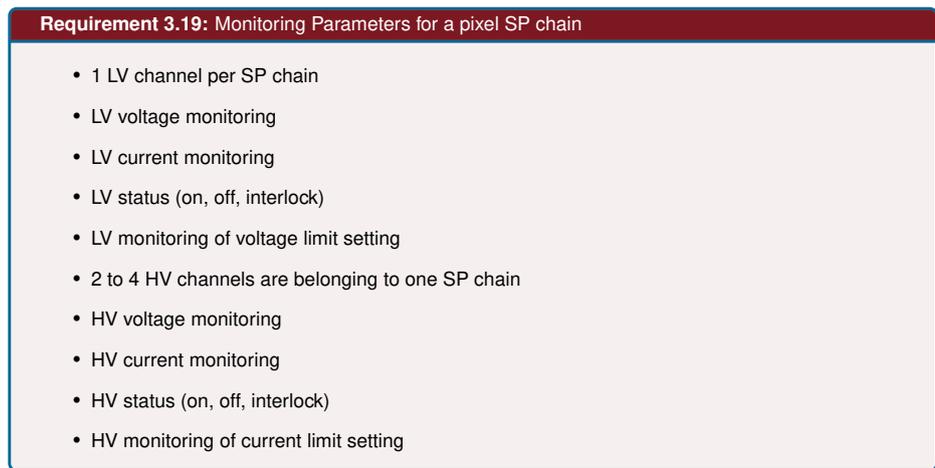
**Figure 5.7.:** A multi value trend panel, which offers a search for specific variables and selection for showing their value over time.

Two panels were created for the SP-chain: the main monitoring panel that is shown in Fig. 5.9 and the control panel that can be opened from the monitoring panel and is shown in Fig. 5.10.

The monitoring panel for the SP-chain shows the alias of the SP-chain at the top. Then there is the state of the low voltage channel and the two high voltage channels. The text box shows status information like ON, OFF, RAMPING, TRIPPED or INHIBIT. This information comes directly from the power supplies. The LED graphic next to it shows the corresponding interlock signal that comes from the IMC. A click on it opens the interlock panel, that is described in Section 5.4.2. The measured output voltage (via a sense connection) and output current are shown.

The trending plots below show the voltage and current of the low voltage channel over time on the left side and the voltage and current of the two high voltage channels over time on the right side. Below the plots is a field that shows the temperature and voltage drops of the modules. This information comes from the MOPS on the SP-chain. Since different SP-chains can have different number of modules, the panel adds them dynamically on load by checking which aliases exists in the database. However, for the moment there are no SP-chains in the experiment setup and no aliases for the modules in the database.

At the bottom left, there are the power channels for the Opto Boards inside of the Opto Box. In the current setup, the powering consists of two stages: first a 9 V general



**Figure 5.8.:** Requirements that the DCS has to fulfill with respect to the monitoring of an SP-chain [61].

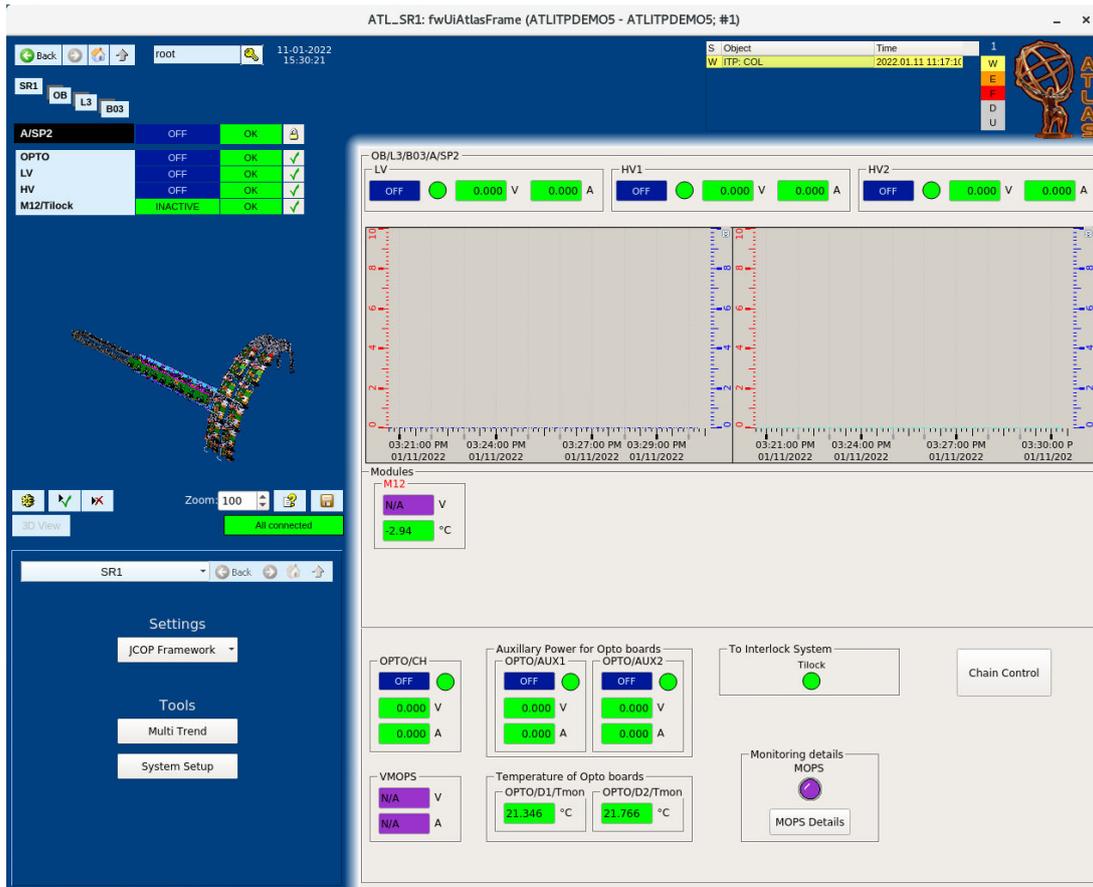
power channel and then auxiliary low voltage power for each Opto Board. Up to two Opto Boards can be assigned to one SP-chain. The temperatures of the Opto Boards are also shown. These temperatures are read out by another MOPS inside the Opto Box.

The supplied power on the CAN bus cable to which the MOPS on the SP-chain is connected is shown in the bottom left. The state of this MOPS is indicated with an LED graphic on the right side. Lastly, the generated interlock (input) signal from an NTC on one module that is connected to the IMC is shown with label “Tilock”. In case this temperature sensor generates an interlock signal, the LED graphic turns red and an alert is triggered. A right-click on the LED graphic opens a panel to acknowledge the alert.

By clicking on the “Chain Control” button in the monitoring panel, a new panel opens with the possibility to control the power lines of an SP-chain (Fig. 5.10). The JCOP framework handles most of the remote control via two data points or connections. The OPC-UA client of WinCC OA provides the possibility to add peripheral addresses to data points in three modes: input, output and bidirectional. For more reliability, settings of devices are then added as two data points, once with an output and once with an input address. Writing into a data point with output address changes the settings on the device. Using the readback data point with an input address, one can check if the new setting was accepted.

In this control panel for the power supplies of one SP-chain one can individually turn the channels on or off and set operating values. The settings are shown using the readback data points. A click on them opens a panel to set a new value. The LED graphic shows the interlock state coming from the IMC. The state of the power channel is shown in the

5. The DCS for the ITk Pixel OB demonstrator

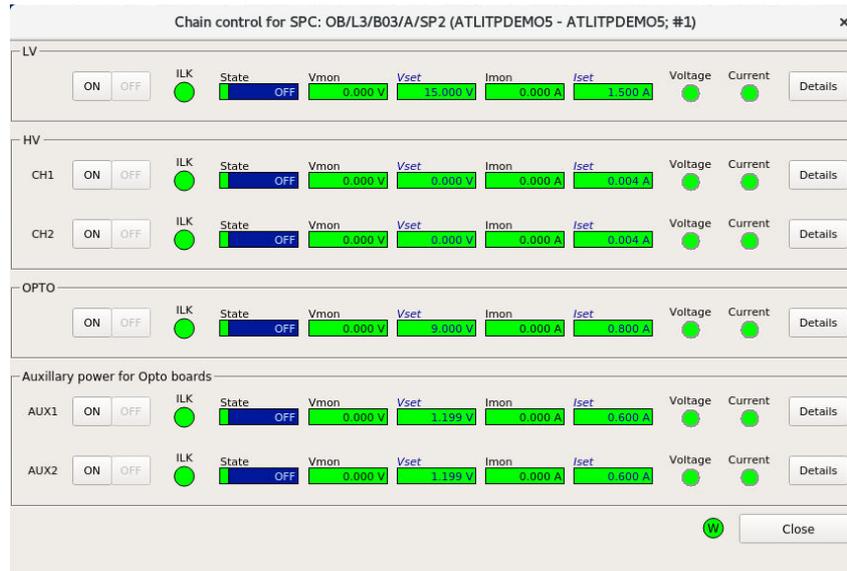


**Figure 5.9.:** The SP-chain monitoring panel, here for the OB/L3/B03/A/SP2 SP-chain. Note, that since no actual SP-chain is available yet, some information is missing. Missing information is displayed with a purple background, for example for the MOPS. With the exception of a dummy temperature sensor for the interlock system, no module information is shown.

text box next to it. In the left corner of this text box, the status of the power channel is shown. In case that the power channel triggers an alert, the corner turns red. A left-click on this corner opens a panel for alert actions, such as acknowledging or masking.

Two error flags are shown on the right, which can indicate why a channel tripped, i.e. *maximum voltage exceeded* or *maximum current exceeded*. A click on the “Details” button opens the device panel for the hardware channel. This panel is offered by the JCOP framework and contains the most detailed information. At the bottom, next to the close button, is a status LED graphic for the watchdogs. A click opens the watchdog panel.

For dealing with SP-chains and their aliases, an SP-chain library was written. Functions in this library provide quick access to the metadata contained in the alias. In addition, finding the interlock module for an SP-chain is efficient, since a caching mechanism is used to avoid repeated searches in the database.



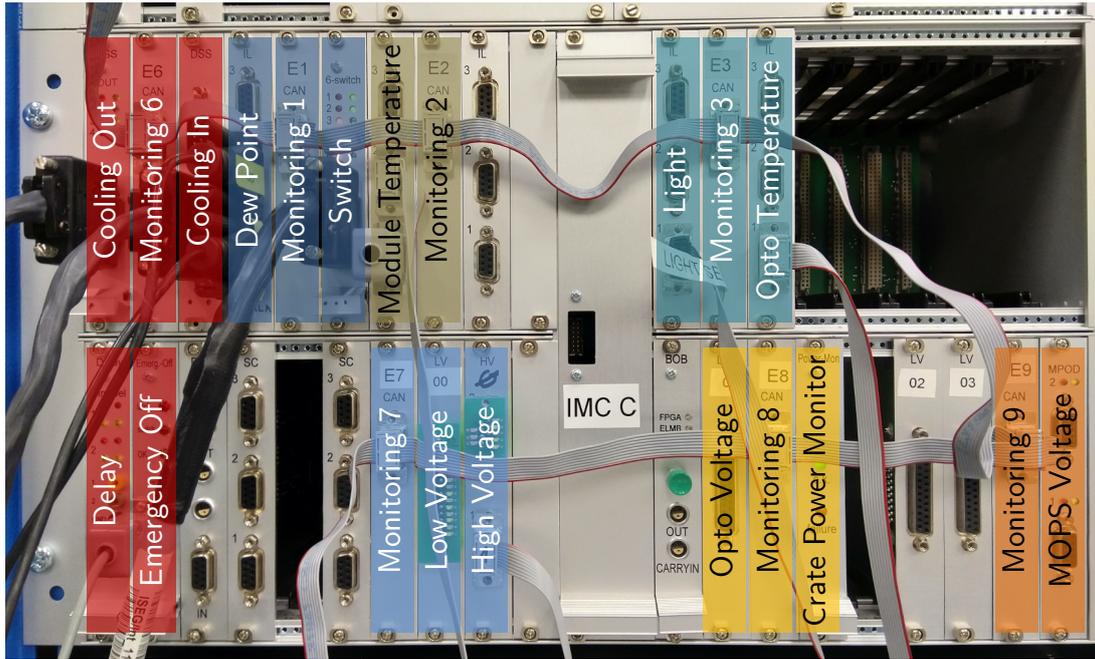
**Figure 5.10.:** Detailed control for the power supplies of an SP-chain.

### 5.4.2. Interlock monitoring

The interlock will act as the last line of defense against overheating of modules and risks to humans. It is supposed to take measures when a monitored value goes above a given threshold and work independently of the other systems. The values monitored are temperatures and dew points at different positions in the setup of the demonstrator, as well as signals from light sensors and states of switches. The main element of the interlock system is the IBL Interlock Matrix Crate (IMC) [70] with slots filled with modules as shown in Fig. 5.11. Signals between the modules and the FPGA in the middle of the crate are routed via the backplane. The FPGA includes the interlock logic that defines how the system should react. The input cards have a discriminator to generate a digital signal for the FPGA if a value exceeds a certain threshold. The input cards are adapted for the various kinds of sensors. In the same way, the output signals from the FPGA are converted in the output cards to an interlock signal for power supplies. The FPGA's task is to connect the correct input signals with the corresponding output signals. Normally, each output signal is a logical OR connection of several input signals. The entire logic can therefore be represented by a matrix.

The different types of sensors for an input signal are NTCs, box switches (to check whether the Test Box in which the demonstrator is situated is open or closed), an emergency off button, light sensors and dew point sensors. The NTCs are used to measure the temperature. An NTC from the last module in each SP-chain from the point of view of a cooling loop is connected to the IMC instead of the MOPS. Additional NTCs are to

## 5. The DCS for the ITk Pixel OB demonstrator



**Figure 5.11.:** The Interlock Matrix Crate (IMC) in the SR1 setup for use with the OB demonstrator. Behind the plate labeled “IMC C” is the FPGA. Different interlock cards are used for different input and output signals. The signals are monitored by ELMBs, which are all connected via a CAN bus.

be placed on the cooling pipes itself. The interlock signal from the NTCs is used to control the power supplies for all SP-chains along a cooling line. This is to prevent the modules from overheating should the cooling be inoperable or in case of a bad thermal contact.

The NTCs are connected via an IL (interlock) board to the FPGA. This is where the threshold for the monitored value is set. The IL board allows (passive) monitoring of the values passed to the FPGA via an ELMB. The switches on the Test Box can raise an interlock signal if the box is open. With light sensors inside the box, one can detect if the box is properly closed.

Two Vaisala DMT143L dew point sensors<sup>9</sup> are used to measure the dew point of the dry air inlet and the Test Box. The signal from the dew point sensors is used by the interlock logic to act directly on the cooling plant via the DSS. This is to prevent icing on the sensors should the air become too humid. As the power supplies for the FE chips and sensors are only allowed to work when the cooling is on, an interlock signal from the dew point sensors will then consequently switch off all power supplies.

The IMC acts with output signals only on power supplies and the CO<sub>2</sub> cooling plant. A special module is the Delay module that can be used to delay an interlock signal to first

<sup>9</sup><https://www.vaisala.com/en/products/instruments-sensors-and-other-measurement-devices/instruments-industrial-measurements/dmt143>

act on the HV channels, and then, by delaying the interlock, on the LV channels. The Delay module is neither used by the box switches, nor the emergency off button where power should be cut immediately.

Although the interlock system is purely hardware based, it is required to communicate the interlock actions to the rest of the DCS. This is realized by the use of ELMBs that read out the incoming and outgoing signals and report them to the DCS. All monitoring signals are latched, meaning the information about a value exceeding a threshold is kept until manually reset, to allow the user to reconstruct the behavior of the system even in case of very short interlock signals. All interlock signals follow negative logic where 0 V corresponds to an interlock signal, meaning that in case of a power loss or a broken cable, the interlock is active and the corresponding channels cannot be powered. The *monitoring* latched signals, that is the signals that are read out by ELMBs, follow a logic where  $-0.01$  to  $0.1$  V is interpreted as safe, while  $3.2$  to  $3.4$  V is interpreted as an interlocked state and everything else is not a valid signal [70]. In addition, the digital outputs of the ELMBs are used to provide test signals to the interlock system.

The panel for monitoring the IMC should display the matrix that was used to program the FPGA in the IMC and that connects input and output signals. In addition, the state of every output and every input signal should be shown. One FPGA input can collect up to four sensor signals that are connected via an OR operation. This means that if three temperature sensors share one FPGA input and if any of them has a value that exceeds a certain threshold, a collective interlock signal is sent. To help with debugging of the detector, the operands of the OR connection for the final input signal should also be displayed. This makes it clearer, where the interlock signal came from.

Since the FPGA can be reprogrammed and the cabling can change, the interlock matrix is subject to change as well. The monitoring panel uses a table to display the matrix which is filled by information stored in a interlock matrix data point. This information can be updated by reading the interlock matrix from an XML file. The structure of such a configuration file is shown in Fig. 5.12. The main defined element is an interlock matrix. The children of this matrix node are input signal nodes. Each input signal can have several attributes. It is mandatory to specify the FPGA signal name, the data point element that is associated with the ELMB channel used to monitor the signal, and the alias of the data point. Optionally, one can define a “hidden” attribute to hide this signal in the interlock panel, and an “inverted” attribute, which considers that the threshold for the signal should be interpreted with switched regions for safe and interlock.

Next, the dependencies of the input signal can be defined. One dependency consists of the single interlock signal and the sensor signal. The sensor signal is the (physical)

## 5. The DCS for the ITk Pixel OB demonstrator

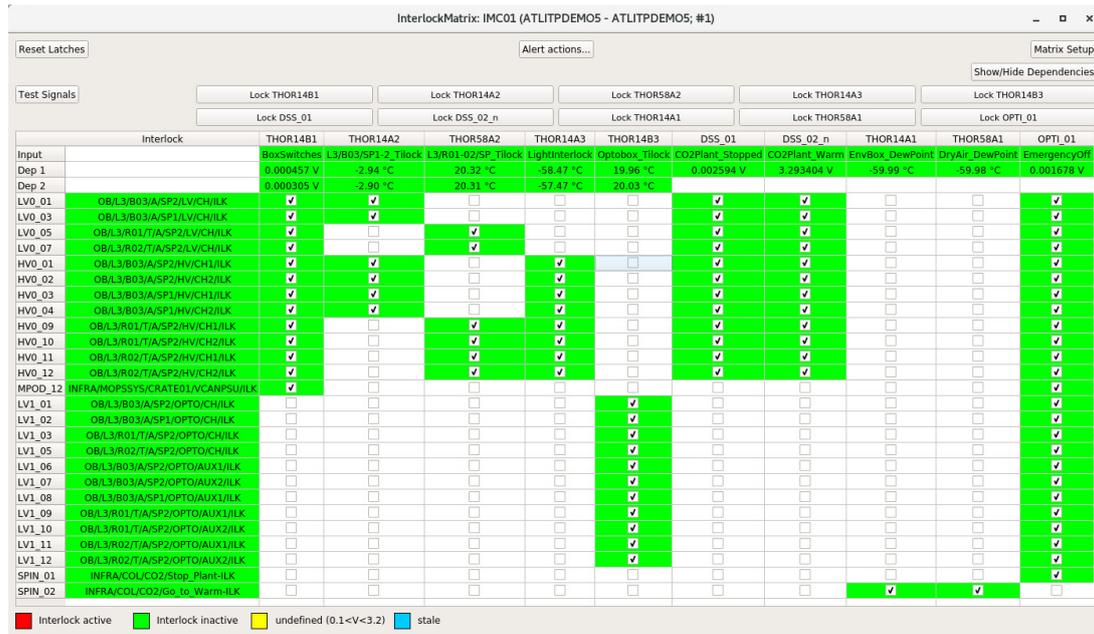
```
<?xml version="1.0" encoding="UTF-8"?>
<matrix name="InterlockMatrixCrate" alias="INFRA/ILOCK/IMC01">
  <input signal="THOR14A2" element="ELMB/CAN0/IMC_ELMB2/DI/di_F_0" alias="L3/B03/SP1-2_Tilock">
    <depends>
      <interlock>
        <element inverted="false">ELMB/CAN0/IMC_ELMB2/AI/THI_A01</element>
        <alias>OB/L3/B03/A/SP2/M12/Tilock</alias>
      </interlock>
      <value>
        <element>ELMB/CAN0/IMC_ELMB2/AI/NTC_A01</element>
        <alias>OB/L3/B03/A/SP2/M12/Tmon</alias>
      </value>
    </depends>
    <depends>
      <interlock>
        <element>ELMB/CAN0/IMC_ELMB2/AI/THI_A02</element>
        <alias>OB/L3/B03/A/SP1/M06/Tilock</alias>
      </interlock>
      <value>
        <element>ELMB/CAN0/IMC_ELMB2/AI/NTC_A02</element>
        <alias>OB/L3/B03/A/SP1/M06/Tmon</alias>
      </value>
    </depends>
  </input>
  <output signal="LV0_01" element="ELMB/CAN0/IMC_ELMB7/AI/LV0_01" alias="OB/L3/B03/A/SP2/LV/CH/ILK">x</output>
  <output signal="LV0_03" element="ELMB/CAN0/IMC_ELMB7/AI/LV0_03" alias="OB/L3/B03/A/SP1/LV/CH/ILK">x</output>
  <output signal="LV0_05" element="ELMB/CAN0/IMC_ELMB7/AI/LV0_05" alias="OB/L3/R01/T/A/SP2/LV/CH/ILK"></output>
  <output signal="LV0_07" element="ELMB/CAN0/IMC_ELMB7/AI/LV0_07" alias="OB/L3/R02/T/A/SP2/LV/CH/ILK"></output>
  <output signal="HV0_01" element="ELMB/CAN0/IMC_ELMB7/AI/HV0_11" alias="OB/L3/B03/A/SP2/HV/CH1/ILK">x</output>
  <output signal="HV0_02" element="ELMB/CAN0/IMC_ELMB7/AI/HV0_12" alias="OB/L3/B03/A/SP2/HV/CH2/ILK">x</output>
  <output signal="HV0_03" element="ELMB/CAN0/IMC_ELMB7/AI/HV0_13" alias="OB/L3/B03/A/SP1/HV/CH1/ILK">x</output>
  <output signal="HV0_04" element="ELMB/CAN0/IMC_ELMB7/AI/HV0_14" alias="OB/L3/B03/A/SP1/HV/CH2/ILK">x</output>
</matrix>
```

**Figure 5.12.:** Structure of an interlock matrix configuration file. The matrix name will be used as data point name for the interlock configuration data point, the alias value will be made its alias.

value from which the interlock signal was derived. Finally, the list of output signals that the input signal is connected to is specified. The connection is marked with an “x”. One can read this configuration file during system setup or directly from the interlock panel and then display the interlock matrix. During reading of the configuration file, the appropriate alerts are set to the data points with the monitoring latched signals. Note that the configuration file should represent the programmed state of the FPGA to show the correct configuration. The panel cannot be used to change the FPGAs state.

In case that there are several IMCs simultaneously in use in a setup, one can read several configuration files. These files should differ in the specification of the matrix name, however. The alias should be used to number the IMCs. All interlock related functions are outsourced into an interlock library.

Figure 5.13 shows the interlock monitoring panel with a table displaying the interlock matrix. One advantage in using the table approach for the matrix is that it is easy to hide certain rows or columns. This is performed when opening the panel from an interlock LED graphic on the SP-chain panels. By clicking on the interlock LED graphic of a specific power channel, the interlock panel opens showing only the interlock signals for this power channel. This should help in quickly investigating reasons why a power channel is inhibited. The interlock panel also offers a button to reset latches. The resetting of latches has to be done every time an interlock was produced. The monitored inputs are configured such



**Figure 5.13.:** Monitoring panel for the IMC. The input signals are shown as columns, the output signals are on the left side as rows. The background colors of the signals show the interlock state. It is possible to show the dependencies of the input signals below their name and to send a test signal to every input channel. Lastly, the panel provides the possibility to reset latches..

that once a threshold is exceeded, it is signaled to the DCS until the latches are reset, even if the value drifts again below the defined threshold. Finally, a button provides group actions on all alerts for output signals and can be used to acknowledge all alerts coming from interlock output signals.

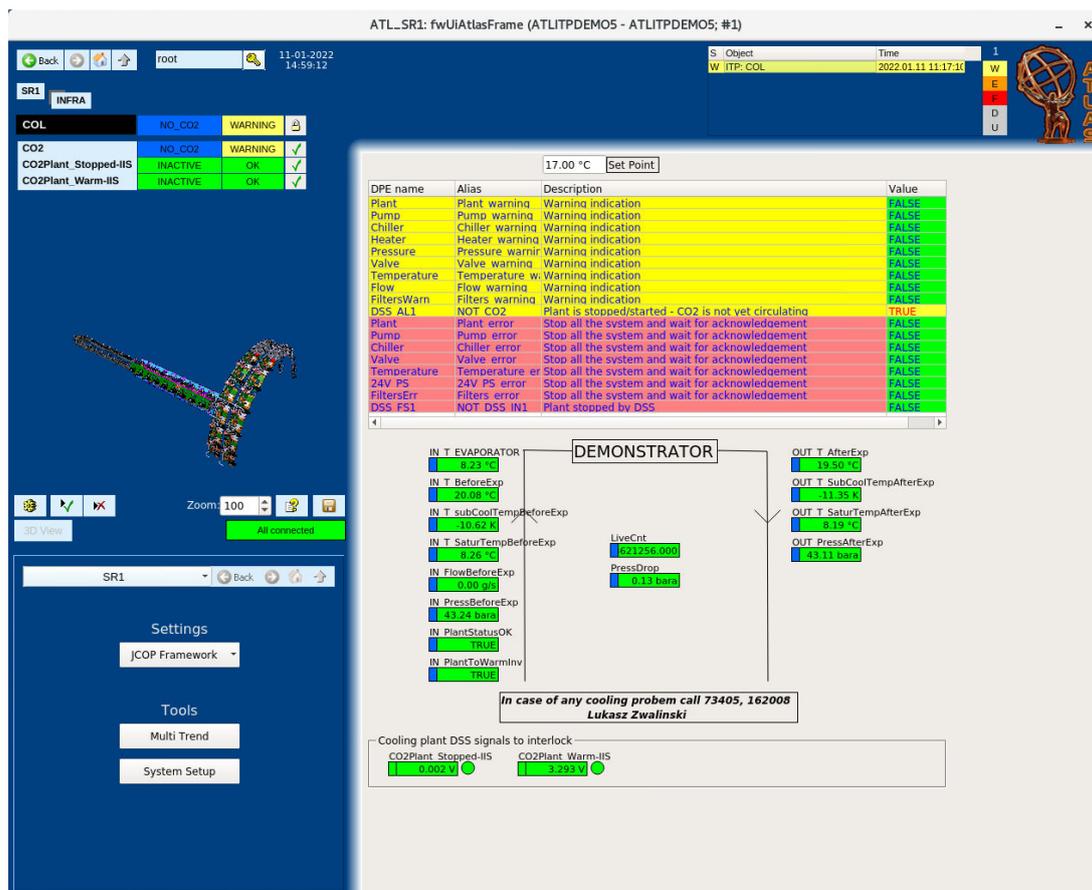
### 5.4.3. Cooling monitoring

The monitoring information of the CO<sub>2</sub> cooling plant is received via DIP. A DIP subscription is requested from the DIP name server in the General Purpose Network (GPN) of CERN. The information contains binary warning and error flags about the status of the cooling plant. An alert handling for data point elements with these flags is configured.

The panel for the information on the cooling plant is shown in Fig. 5.14. Background colors in the last column correspond to the alert state of the warning and error information. A panel for alert actions can be opened by double-clicking a cell in the last column.

Below the status some operational values are shown. The color in the left corner of the value box represents the FSM state of the cooling plant. At the very bottom are the DSS signals that the cooling plant sends to the interlock system as input signals. The LED

## 5. The DCS for the ITk Pixel OB demonstrator

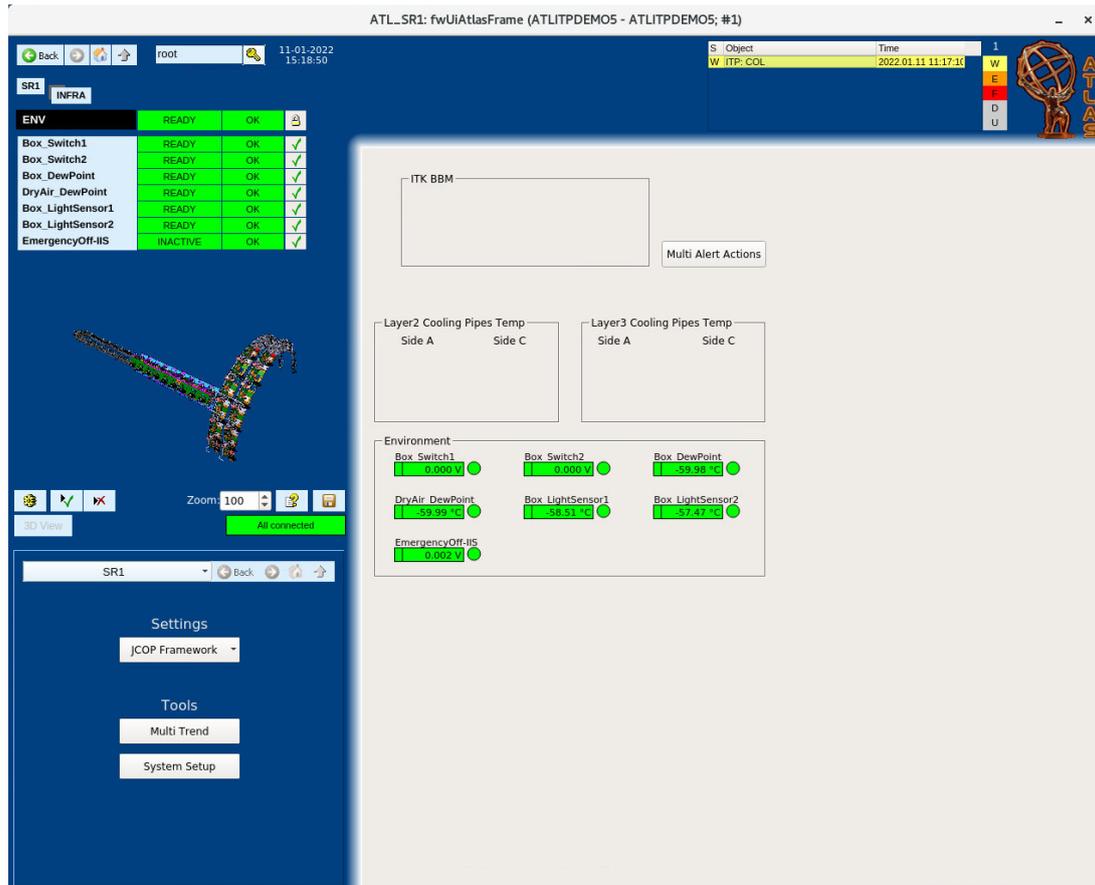


**Figure 5.14.:** Monitoring panel for the CO<sub>2</sub> cooling plant. At the top a list of possible warnings and errors and if they are active is shown. Below there are operational values displayed. These are the values received via DIP. At the very bottom are the signals that the cooling plant sends to the interlock system. In this case however, a dummy “good” signal is sent to the IMC. This panel is used only for monitoring and no commands can be sent to the cooling plant.

graphic represents the interlock status. The alert actions panel for the interlock signals can be opened by clicking on the LED graphic.

### 5.4.4. Environmental monitoring

In the environmental monitoring panel in Fig. 5.15 currently only the sensors that are connected to the IMC are shown. A value is displayed with the interlock state as an LED graphic to the right of it and the FSM state in the left corner. This panel is needed to enable alert actions on the IMC input signals and to signal the operator where the interlock signals have their origin. Additional environmental sensors like humidity sensors in the Test Box are planned to be added later once they are available.

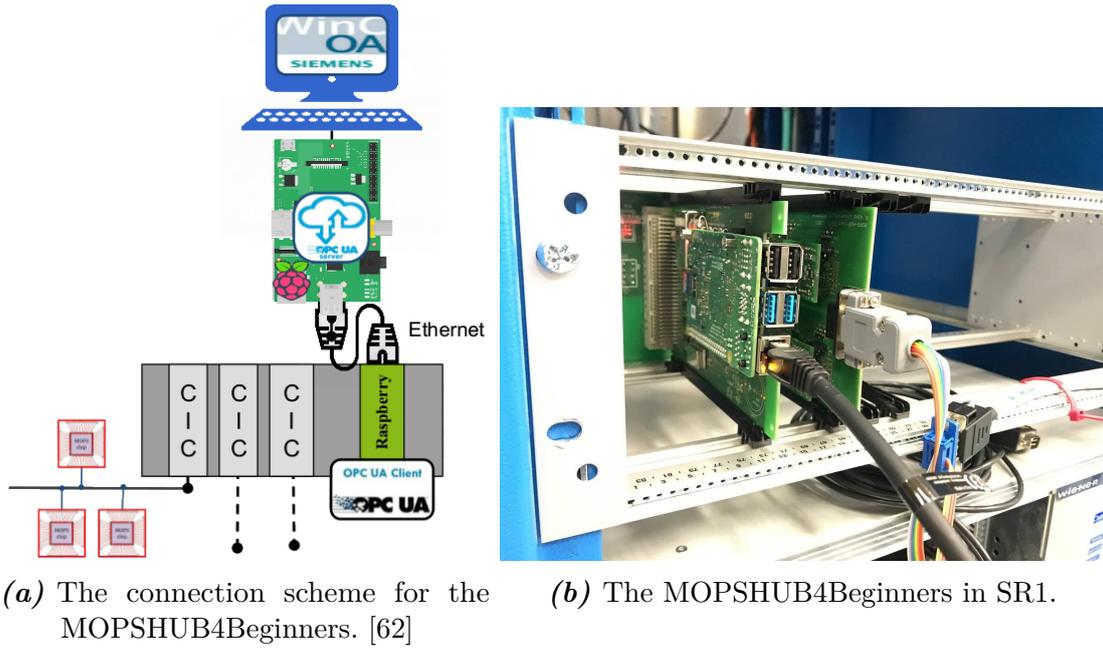


**Figure 5.15.:** Environmental monitoring with sensors in the Test Box. Currently, there are only the sensors for the interlock system connected. These include box switches, light sensors and dew point sensors. The status of the emergency button is also shown.

#### 5.4.5. MOPSHUB4Beginners control

To enable the readout of the MOPS, the MOPSHUB4Beginners and the CAN Interface Cards (CIC) as shown in Fig. 5.16 need power. Power is provided by two LV MPOD channels. The power channel for the MOPSHUB4Beginners is labeled INFRA/-MOPSSYS/CRATE01/VPP3, while the power channel for the CICs is labeled INFRA/-MOPSSYS/CRATE01/VCANPSU. This naming scheme follows closely the official one for an FPGA MOPSHUB.

The control panel shown in Fig. 5.17 shows simply the two power channels. The VPP3 power has to be turned on to boot the Raspberry Pi and start the OPC-UA client which enables the readout of MOPS data. The OPC-UA client was configured as a systemd service to start automatically.



**Figure 5.16.:** The MOPSHUB4Beginners and its connection to the WinCC OA projects. Note that the OPC-UA server does not run on another Raspberry Pi, but rather a normal computer.

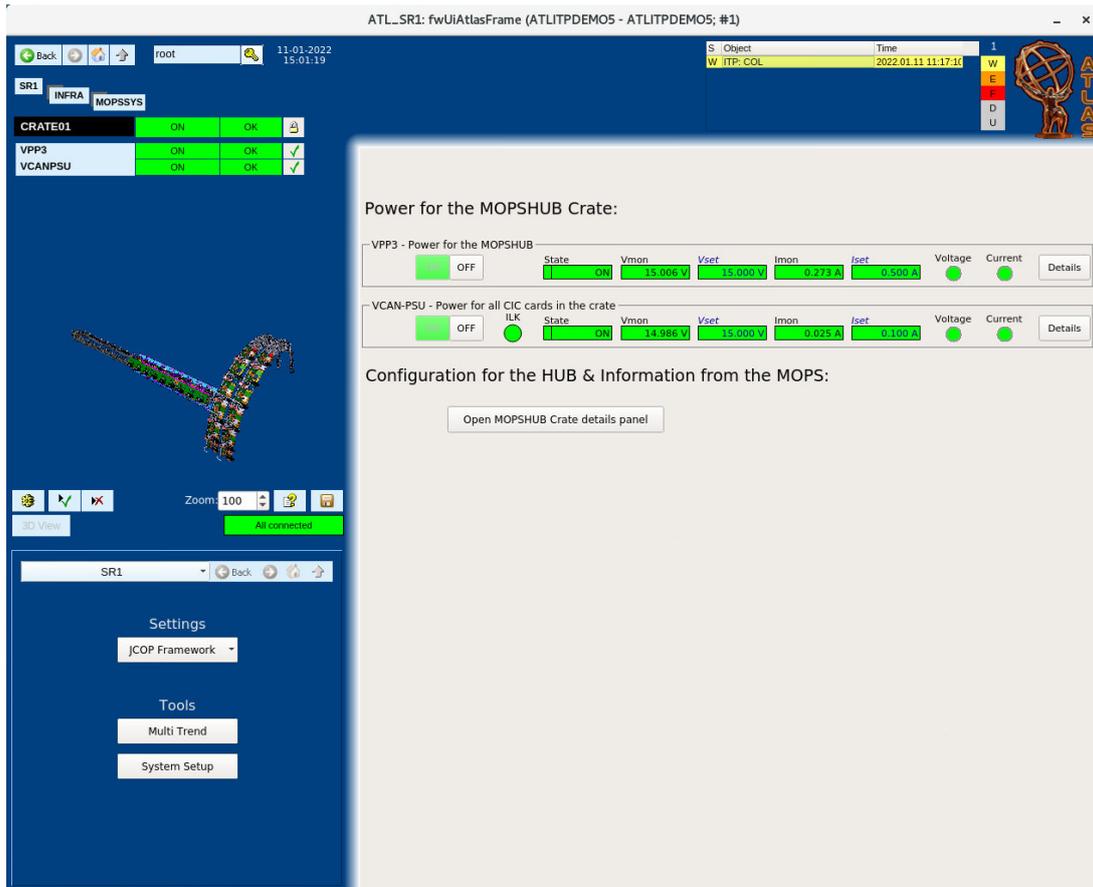
It should be mentioned that only the VCANPSU needs an interlock connection, since this is the PSU that supplies power into the Test Box. The data that is received from the MOPS is provided to the main WinCC OA project by an external distributed project [69]. A panel with details to these information can be opened from the control panel.

### 5.4.6. Opto Box monitoring

The monitoring of the Opto Box consists of the information that is sent to the interlock system and read out by an ELMB in the IMC, and information read out by a MOPS in the Opto Box. As shown in Fig. 5.18, the interlock information is displayed in an extra frame with LED graphics that indicate the interlock status. The MOPS information consists of temperatures of the Opto Boards and voltages of the DC-DC converters called bPOL12V. In addition, the state of the OPTO power of the associated SP-chains is shown.

### 5.4.7. Watchdog monitoring

The panel for the watchdogs is shown in Fig. 5.19. The information it shows is collected by the watchdog scripts. The scripts check the connection to the hardware. If any errors are detected, an alert is raised. This indicates to the operator that the issue should be



**Figure 5.17.:** Control panel for a MOPSHUB Crate with a MOPSHUB4Beginners and CICs.

fixed immediately, as the interruption of the connection prevents the actual state of the hardware from being accessed.

5. The DCS for the ITk Pixel OB demonstrator

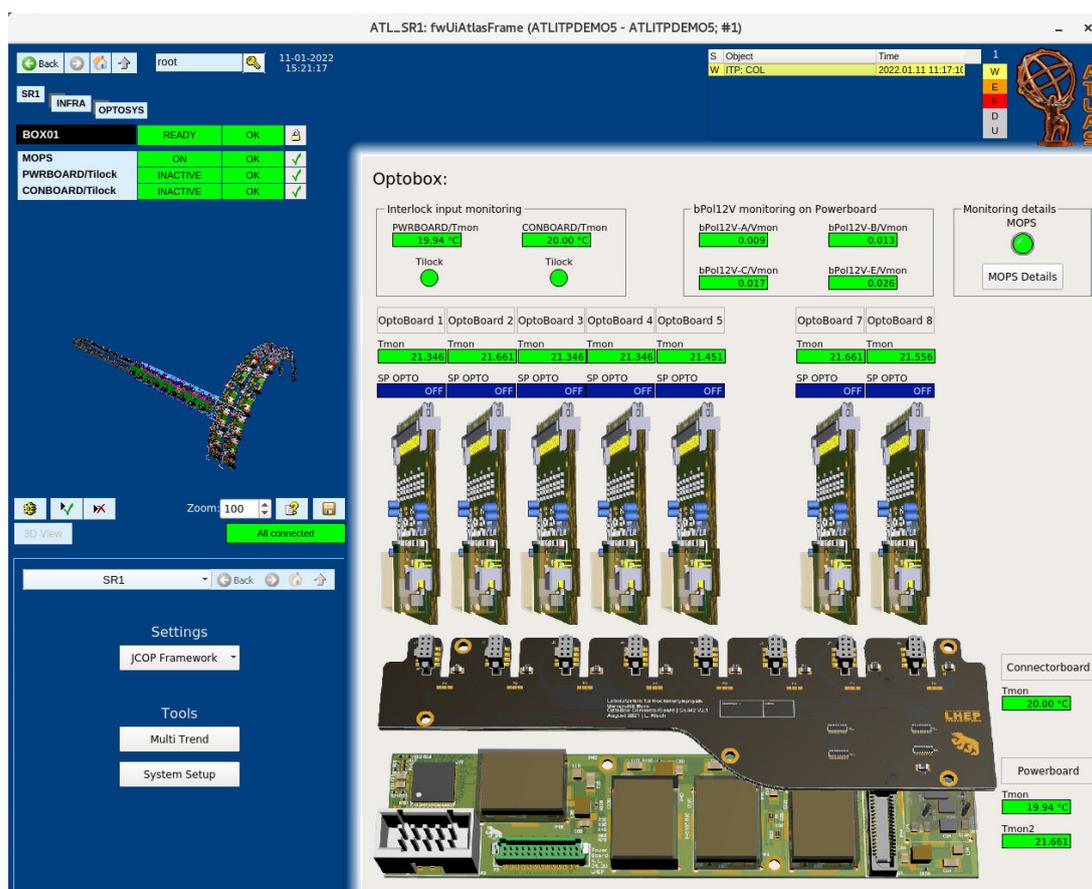
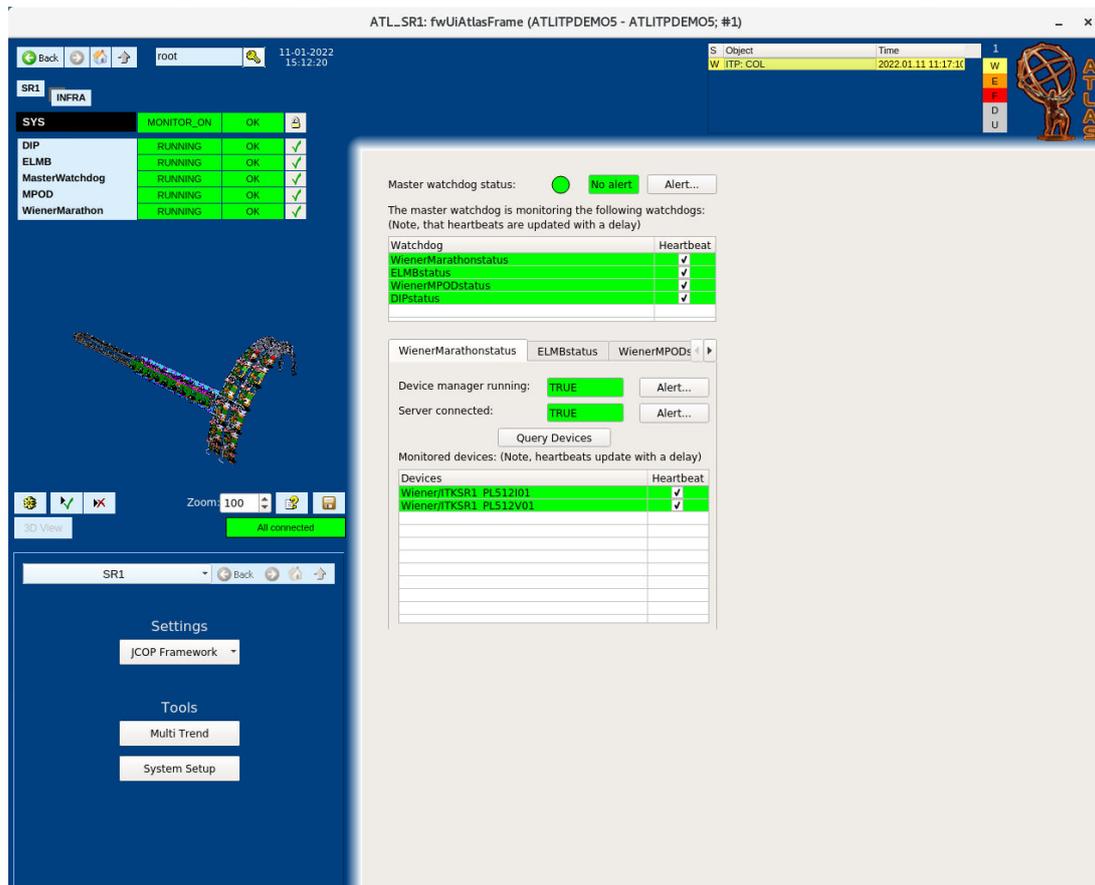


Figure 5.18.: The Opto Box monitoring panel with information from the MOPS and information that is sent to the interlock system.



**Figure 5.19.:** Monitoring panel for the watchdogs. Information from the master watchdog is shown in the upper half. The master watchdog monitors the device watchdogs and lists their state. The device watchdogs monitor the connection to the hardware. For each hardware connection, the watchdog monitors if the WinCC OA device manager is running, whether a connection to the server is established, and whether heartbeats from the devices are being received.

## 5.5. FSM control hierarchy

To ensure safe operation and enable supervision of the detector by non-experts, an FSM control hierarchy is set up for the demonstrator DCS. It acts as a layer between the low level hardware oriented parts of the DCS and the demonstrator operation and supervision. With it, large parts as well as the entire demonstrator can be controlled by simple commands, and together with the alert system, the debugging of problems is made easier.

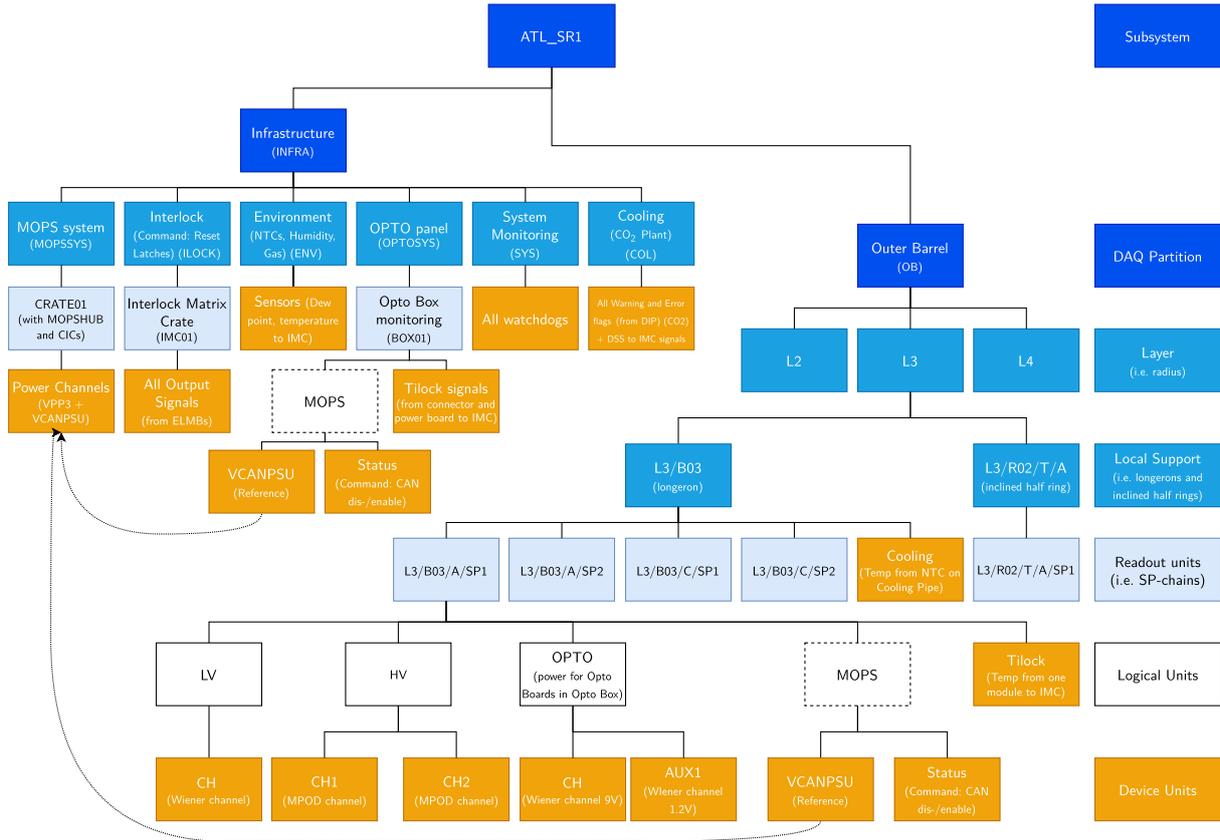
The FSM control hierarchy is realized as a tree as shown in Fig. 5.20. Commands are propagated downwards in the tree and states are propagated upwards. The leafs of the tree are the Device Units (DUs), which are shown in orange. The white nodes are Logical Units (LUs) and the blue colored nodes in the tree are Control Units (CUs) as defined in Section 4.3. The tree consists of two main subtrees, the infrastructure tree and the detector tree. In the detector tree, the DUs are mainly power supply channels for powering the detector parts. The DUs in the infrastructure tree are mainly sensors and status signals.

For the demonstrator setup, the granularity of the DUs was chosen to be very fine, such that detailed information during tests is available. In particular, channel-wise information for the power supplies is available. The non-expert operator will usually use the lowest CU to gather detailed information about the state of the detector parts, and there is no need for knowledge of the specifics of the hardware components.

The lowest CU in the detector tree is an SP-chain. It is the smallest unit that can be controlled independently from other parts of the detector. A detailed description of the SP-chain CU as an example for the implementation of FSM logic is given in the following Section 5.5.1.

The general creation of CUs, LUs and DUs followed the conventions and an example implementation of an FSM in ATLAS [73]. For the DUs, their operational state is computed inside a control script in WinCC OA. The status of a DU is usually obtained from the alert handling of the associated data points. All state, status and command labels have upper case letters. The ATLAS specific extension for the FSM components for WinCC OA predefines a color coding scheme of 8 different colors for the state of FSM objects. An orange or red color indicates a problem. A light blue color indicates a transition state that is expected to transition into a static state after some time. Objects in transition usually cannot receive commands.

For the time being, the FSM logic implemented in the objects does not contain automatic actions, such as a software interlock. If an error is detected, the FSM or alert system will signal the error to the operator and allow the operator to take action. This concerns mainly the status received from the power supplies. In case of dangerous conditions in the environment, the IMC will act directly on the power supplies, independently from



**Figure 5.20.:** FSM hierarchy tree of Control Units, Logical Units and Device Units for the demonstrator. The tree consists of two main subtrees, the infrastructure tree and the detector tree. Due to lack of space only one SP-chain Control Unit is shown in full detail.

the FSM. However, the FSM will reflect the changed state immediately. If, for instance, a power supply trips, the FSM could be programmed to take automatic actions such as turning other power supplies off as well.

The FSM tree for the project is created by using the aliases in the system. A library was written that automates the creation of the FSM tree in the main system by traversing down the logical tree in all connected, distributed systems. This means that the FSM only runs on the main system, but the DUs will access data points on remote systems if so instructed. To make this possible, the FwFsm framework component was patched locally to allow the access of remote data points by their alias. The mapping of aliases to FSM object types is then realized by pattern matching. Using this approach, the FSM tree can be easily adjusted to the configuration of the system. The DUs name is then the alias of the associated data point and already includes the tree structure. In the names for the LUs and CUs the “/” symbol is replaced by an underscore (  ), since the slash is a reserved character in these cases.

## 5. The DCS for the ITk Pixel OB demonstrator

In contrast to the simple logical tree, the FSM tree allows for object references that can be used to create other views into the hierarchy. For the demonstrator, references are used to distribute the MOPS power to the different MOPS LUs as shown in Fig. 5.20. The MOPS LUs are also shown with dashed rectangles to indicate that only their status is propagated upwards in the tree. A MOPS that is switched off raises an alert without changing the state of the parent CU.

As an example for the FSM logic of a CU, the SP-chain CU is described in more detail in the following section.

### 5.5.1. SP-chain Control Unit

An SP-chain is supplied by a constant current source, generating the LV to power the FE chips of all modules, and by a voltage source providing two HV lines in the Outer Barrel for the depletion of the sensors. For the readout via the readout system, the FE chips send their data to the Opto Box, where the electric signals are converted into optical signals. Up to two Opto Boards inside an Opto Box are associated with an SP-chain to take on the task of converting the signals. There is one main OPTO power associated to each SP-chain, and in the case of the prototype Opto Box, for each Opto Board there is an auxiliary lower voltage power channel to replace the missing voltage converters in the Opto Box.

The design of the SP-chain CU can also be seen in Fig. 5.20. The figure shows that the CU includes three LUs as children for the LV, HV and OPTO powering. The DUs are then the associated channels of the power supplies. The HV unit groups together two power channels and the OPTO unit one main channel, along with either one or two auxiliary channels. Currently, no individual modules are added as children, since they are powered in serial by LV.

The FSM logic for the DUs of the power supply channels is implemented using scripts in WinCC OA. The behavior of the DUs for different power supply types is almost identical. The possible states of these DUs are listed in Table 5.2. Their status is derived from the alert handling of the associated data points. The higher-level LV unit replicates the state behavior of its single DU, while the HV unit adds an additional MIXED state, in case that the two HV power channels are in different states. The OPTO unit also distinguishes between the main channel and the auxiliary channels, and gives state information if the main channel is on while the auxiliary channels are turned off.

The SP-chain CU then consists of its own set of states, the transition between these states, and actions that are available in each state. The state of each SP-chain CU should summarize the states of the LUs in a way that highlights the operational procedure. The

**Table 5.2.:** FSM states of a power supply channel DU.

State	Description
ON	Static state: The channel is switched on and outputting constant voltage/current.
OFF	Static state: The channel is switched off.
RAMPING_UP/DOWN	Transition state: The channel is ramping up or down its output.
UNKNOWN	Error state: Connection to the power supply is lost.
INHIBIT	Static state: Power channel receives external inhibit signal. The channel is switched off and cannot be turned on in this state.
TRIPPED	Severe error state: The channel has tripped after exceeding operational limits.

number of states should be kept relatively small. The states available for the SP-chain CU are described in Table 5.3. The normal states are the states where the OPTO power is first supplied to configure the Opto Boards and modules, and then the LV power is turned on to start the FE chips and lastly the HV power to deplete the sensors. Concerning the number of states, it was decided to create all necessary states such that to move from one state to the next, only one command has to be sent to exactly one child. This results in many error states, which can however be grouped together on the next higher level of the FSM tree. The state transitions that are foreseen during a normal operation are shown in the state diagram in Fig. 5.21. The arrows in the diagram indicate the direction of commands. One can see that the commands only act on one child at a time.

It is foreseen to operate the SP-chain from its CU. To prevent incorrect operation, the commands directly onto the LUs that are available from the FSM UI are marked as “expert” commands.

The three different LUs have to be handled cautiously, since they are not equal and need commands sent in the correct order. To be able to go with one command through several states in the state diagram of the CU, a few options are available.

1. The SML language offers a `WAIT_FOR` instruction that can be used when declaring an action. A second command can then only be sent if a certain condition is fulfilled. During execution of the chain command however, the state propagation is blocked and intermediate states are not captured.
2. One can add an extra DU under the CU and let it handle all the commands. A reliable connection then has to be guaranteed. This method was chosen for the smallest CU in the FSM of the current ATLAS Pixel detector.

5. The DCS for the ITk Pixel OB demonstrator

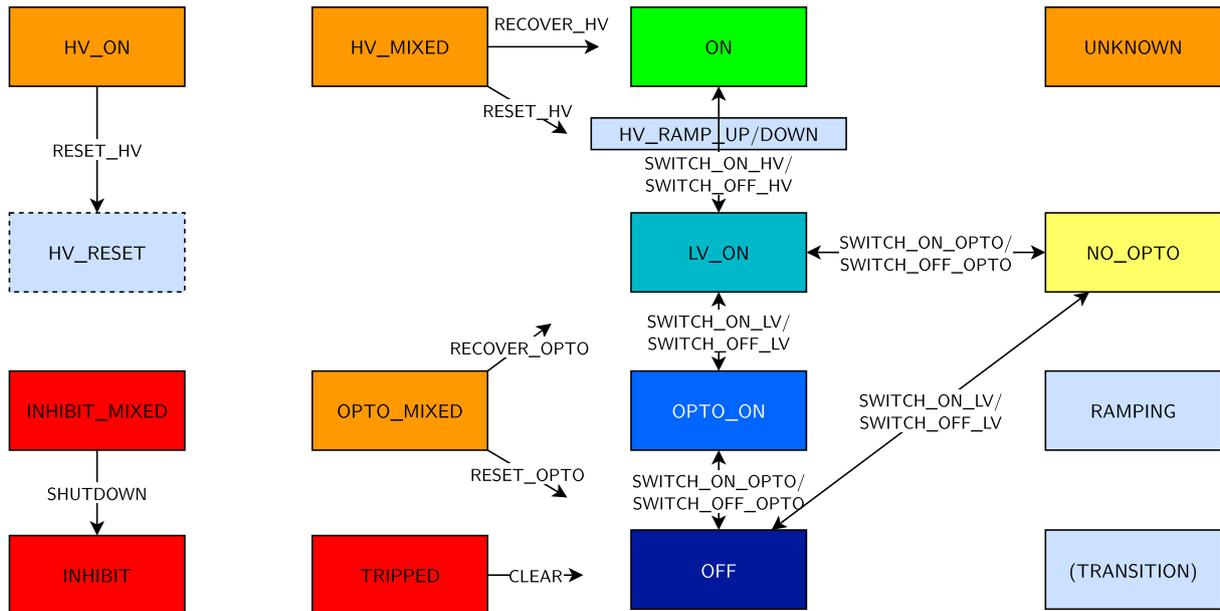
**Table 5.3.:** FSM states of the SP-chain Control Unit.

State	Description
ON	Static state: OPTO, LV and HV are ON.
LV_ON	Static state: OPTO and LV are ON while HV is OFF.
OPTO_ON	Static state: OPTO is ON while LV and HV are OFF.
OFF	Static state: OPTO, LV and HV are OFF.
NO_OPTO	Static state: LV is ON while OPTO and HV are OFF.
RAMPING	Transition state: OPTO or LV are ramping up or down.
HV_RAMP_UP/DOWN	Transition state: HV is ramping up or down.
HV_MIXED	Error state: OPTO and LV are ON while only one HV channel is ON.
HV_ON	Error state: Either LV or OPTO are OFF, while HV is on.
OPTO_MIXED	Error state: OPTO is in a mixed state.
UNKNOWN	Error state: One of OPTO, LV or HV is in state UNKNOWN.
INHIBIT	Error state: One of OPTO, LV or HV is in state INHIBIT, while others are OFF.
INHIBIT_MIXED	Error state: One of OPTO, LV or HV is in state INHIBIT, while others are ON.
TRIPPED	Error state: One of OPTO, LV or HV is in state TRIPPED.

3. One can add additional states for each possible next command in a command chain. These additional states have the same conditions for entering than the stable states. The difference is that on reaching the additional state, they automatically execute the next command in the command chain. Each additional state has one command associated with it. This is akin to the Moore machine discussed in Section 4.1 and would result in many more states.
4. One can use object parameters that persist over state changes. These parameters can be used to trigger the next action. This is akin to the Mealy machine.

For the implementation of chain commands for the SP-chain CU in the demonstrator project, option 4 with object parameters was chosen.

There are three parameters, the previous state, the final state and the command type. When a chain command is issued, these three parameters are set and the first action is executed. Once the CU is in the next static state, it checks the parameters again. By



**Figure 5.21.:** State diagram of the SP-chain Control Unit. The normal operational states are OFF, OPTO\_ON, LV\_ON, NO\_OPTO and ON. The arrows indicate commands.

knowing the final state, the object knows where to proceed next. By knowing the previous state and comparing to the current one, the object knows, whether it is proceeding in the right direction or should stop the chain commands because it is stuck in a state. The command type for this scenario is either RECOVER or GOTO. RECOVER commands do not need a final state parameter and stop as soon as a normal state is reached. GOTO commands can be used to continue when a transition from an irregular state to a normal state occurs. At the end of a chain of commands, the unit goes into a transition state called CMD\_FINISHED, in which it resets all parameters. This state will also appear in logs and can help to understand and retrace the transitions of a unit. A parent of the unit will also be informed of the state changes.

The simple SWITCH\_ON\_\* commands for the CU are then mostly hidden in the FSM UI and GOTO\_\* commands are used, where the name of the command is independent of the current state.

Currently, no delay is implemented between two actions in a chain of commands. This might pose a problem if a device (FE chip or Opto Board) is not immediately ready, once the supply voltage is provided, but the next action has already started. If more information for the DCS is needed about the state of an FE chip or Opto Board configuration, it can be added under the LV LU or OPTO LU respectively, and then be taken into account during chain commands.

## 5. The DCS for the ITk Pixel OB demonstrator

In addition to the previously discussed LV, HV and OPTO children, the SP-chain CU also has a Tilock DU and a MOPS DU. These two units, however, do not play a role in the state calculation for the SP-chain. They only propagate their *status* up to the CU. The Tilock raises an alarm if the module temperature, that is routed to the interlock system, is too high. The MOPS LU will raise an alarm if the DCS monitoring of the SP-chain is defective.

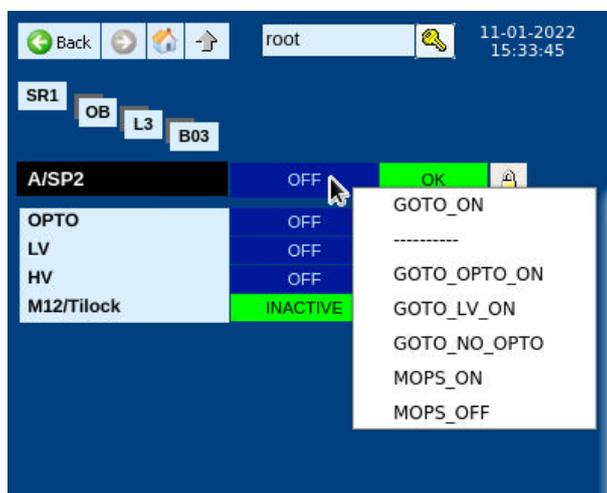
### 5.5.2. User Interface

The control of the FSM is provided via an ATLAS customized UI framework in which the previously created panels are embedded. The FSM UI frame shows information about the FSM tree and offers control of and navigation through the tree. As an example, the start screen of the FSM UI is shown in Fig. 5.22. One can navigate through the tree by clicking on the names of the different units on the left. The alarm table at the top of the panel collects all the statuses in the FSM tree. By clicking on the alarm, the FSM UI navigates directly to the unit where the alarm was raised. With this, one can quickly investigate the cause of an alarm.

The control of the demonstrator should mainly be realized by using FSM commands only. The FSM logic prevents errors and makes handling of the hardware easier. Fig. 5.23 shows the usage of FSM commands for an SP-chain.



5. The DCS for the ITk Pixel OB demonstrator



**Figure 5.23.:** Sending commands to an SP-chain Control Unit in the FSM module.

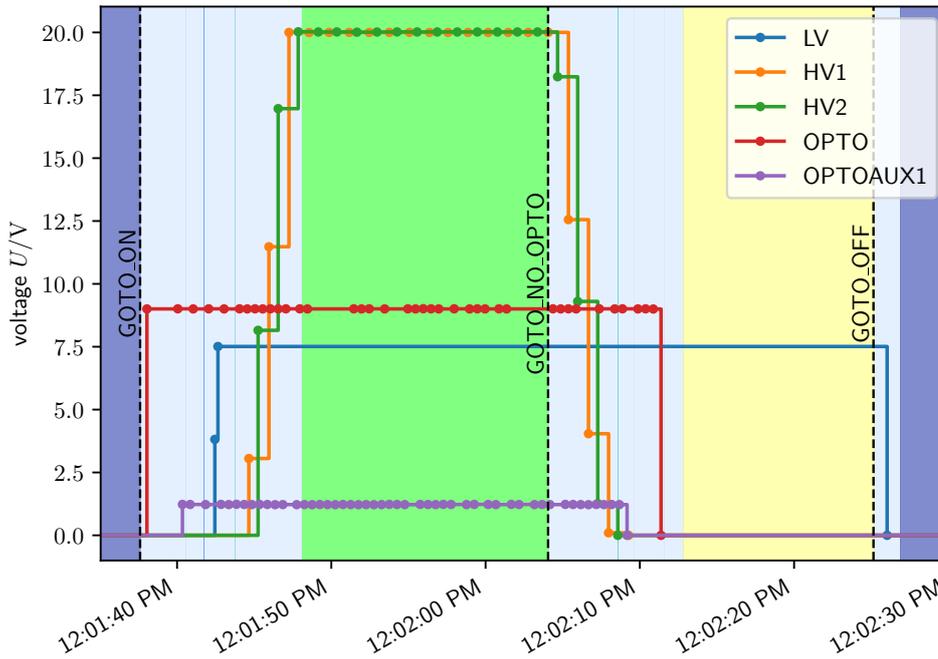
## 5.6. Testing and performance considerations

Since no SP-chain is commissioned yet, no operation under real conditions could be realized. Using a dummy electrical load, however, it was possible to test the off-detector service chain up to the Cable Saver Board. With this test, also the FSM behavior for controlling the power channels of one SP-chain could be checked.

A high-performance resistor was used to test the output of the LV current source. The result of the test is shown in Fig. 5.24. The diagram depicts the time evolution of the voltage output of all channels. It shows how the SP-chain Control Unit transitions from one state to the next after having received a command. From the initial OFF state, the command GOTO\_ON was executed. Through the monitoring of the output voltage, it is visible how first the main OPTO power turns on and then the auxiliary power channel. Once both channels are on, the LV channel is switched on. As the last step in the command, with the LV channel on, the two HV channels are switched on and ramp up to their set value. Having reached the set value, the SP-chain Control Unit goes into the ON state which is indicated by the green background. On the way, the SP-chain Control Unit transitions into every intermediate state according to the state diagram. The Control Unit also executes the GOTO\_NO\_OPTO action and reached the NO\_OPTO state which is indicated by the yellow background. Lastly, the SP-chain is turned off by switching off the LV channel.

Considering performance and scalability of the system, the system setup was designed to enable quick addition of extra hardware by reading a CSV file. If the hardware's alias follows the standard naming scheme, it will automatically be included in the generation of the FSM tree. The FSM library was written in a way such that all connected distributed systems are scanned for aliases according to the naming scheme per default, in order to check if they should be included in the FSM tree. During the system setup, it is not important if the time needed to create the FSM tree is short. Once the system is in operation, however, look-ups of data points in the local database by their alias proceed quickly. The panels are built in a way where they utilize aliases to display information. During initialization of a panel, different parts might only be added after a data point look-up was performed, which can delay the moment until the complete panel is shown. If the full alias of a data point is known, the look-up of the corresponding data point name is instantaneous, even if it is in a remote system (on the same computer). With native time measurement functions, the time needed to execute `dpAliasToName()` is less than the measurement precision. If only a part of the alias is known and the look-up is performed with wildcards, e.g. `OB/L3/B03/A/SP1/HV/CH*` with the `dpGetAllAliases()` function, the time needed for finding the corresponding aliases increases drastically. Native time measurement functions

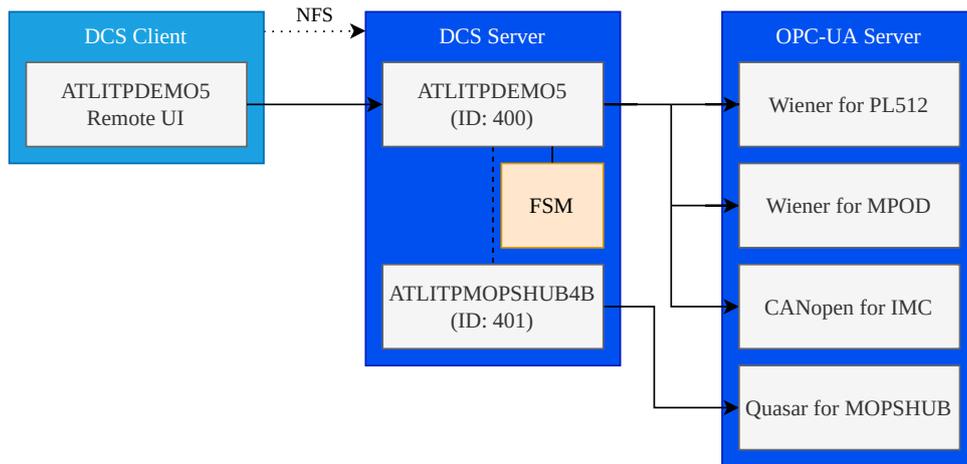
## 5. The DCS for the ITk Pixel OB demonstrator



**Figure 5.24.:** FSM tests for an SP-chain showing the output voltage of all channels needed for operation. The dashed vertical lines are the times when a command was sent to the SP-chain Control Unit. The background color represents the state of the SP-chain.

give an execution time for one function call of this type of approximately 70 ms. In this simple example, where one can guess the range of possible aliases that will match the given search pattern, it is much more efficient to simply check these possible aliases (here CH1 and CH2) individually. Executing 1000 calls to `dpAliasToName()` currently costs approximately the same time as one call to `dpGetAllAliases()`, as was seen in a quick test for the system setup. The usage of `dpGetAllAliases()` should therefore be kept to a minimum and kept as a last resort. In some situations, for example for a function checking for the module in an SP-chain with the interlock connection, the usage of a cache system was realized, where after the first successful search, the results are reused.

In terms of computational stress for the event manager of the WinCC OA project, the current project is too small to show any performance degradation due to event queue sizes being too large. The computer never ran out of available memory.



**Figure 5.25.:** Overview of the DCS software deployed for the SR1 demonstrator setup. The DCS Client is a mini PC in the lab next to the demonstrator, the DCS Server and OPC-UA Server are rack-mounted PCs. All (software) OPC-UA servers needed for the communication with the hardware are running on the OPC-UA Server, while the two WinCC OA projects are running on the DCS Server.

## 5.7. Deployment and usage

As described in the system overview in Fig. 5.1 in Section 5.1, the WinCC OA projects run on a server computer in the rack area of SR1. For the operator in the lab, who is working on the OB demonstrator, WinCC OA was installed on a local PC and a project was set up to operate as a Remote UI for the main WinCC OA project on the DCS server. The Remote UI needs to have a connection to the event and data manager of the host project for access to the local database. Code in panel scripts is however executed locally by UI managers of the Remote UI project. To have the same panels available in the DCS server and the client, an NFS share on the server was set up that was mounted by the client. An overview of the deployed software is given in Fig. 5.25.

The desktop shortcut on the client opens the Device Editor and Navigator of the project. The operator does not have deal with the WinCC OA development tools. The Device Editor and Navigator was chosen over the FSM GUI, because it offers more expert information on the hardware that is connected. The FSM panels can be opened from the Navigator and also the FSM itself can be started and stopped. All other components of the setup were configured to start automatically after boot. This includes also all necessary OPC-UA servers on the OPC-UA Server machine.



# 6. Conclusion and Outlook

## 6.1. Conclusion

A DCS is important for the safe operation and high-level control of a detector. The OB demonstrator that is planned in advance of the upgrade of the ATLAS detector for the HL-LHC requires a DCS to enable the efficient tests of system functionalities for the ITk Pixel OB subsystem. In the course of this thesis, a small but comprehensive DCS was developed for usage by detector experts to operate the demonstrator.

Via a SCADA system where all live data relevant to the operation of the demonstrator is recorded, graphical interfaces provide access to this data. User panels were developed that display the information about the monitored environment and the state of the interlock system. The detector hardware can be monitored and finely controlled in dedicated panels. Scripts running constantly in the background check for communication errors between the software and the hardware.

Special attention was placed to design a system that enables easy operation and gives the operator the tools at hand for quick debugging and troubleshooting. Requirements for operational procedures were identified and the operational model translated into the state model of an FSM. The FSM provides a view into the state of the detector hardware and enables high-level control with well-defined transitions between these states. Together with the alert system, all relevant information is always present to the user.

The system was tested and validated as extensively as possible with a dummy electrical load and made available to the user in the laboratory. To account for future changes in the setup, configuration files can be used to specify the connectivity of the detector hardware and to change the structure of the system in order to adopt to the new conditions.

## 6.2. Outlook

The delivery of the Pixel modules and Loaded Local Supports is planned to be carried out soon, and preparations for first tests are ongoing as part of a detailed test program. This means that the developed DCS project will be used to support system tests with SP-chains

## 6. Conclusion and Outlook

in the ITk Pixel OB demonstrator. It is important to observe the DCS performance and reliability under real world conditions, in order to learn about how well common use cases are covered and where some revision are needed.

As the demonstrator is constantly changing, the DCS also has to constantly adapt. A change planned for the near future is that the Opto Box prototype will receive voltage converters as foreseen in the final design, which will make the auxiliary voltage supply for the Opto Boards obsolete. This will require a change in the FSM logic of the OPTO power for the SP-chain.

Aside from changes that are necessary due to demonstrator upgrades, the DCS is constantly being improved. Development to enable archiving of the monitored parameters into an external database has started.

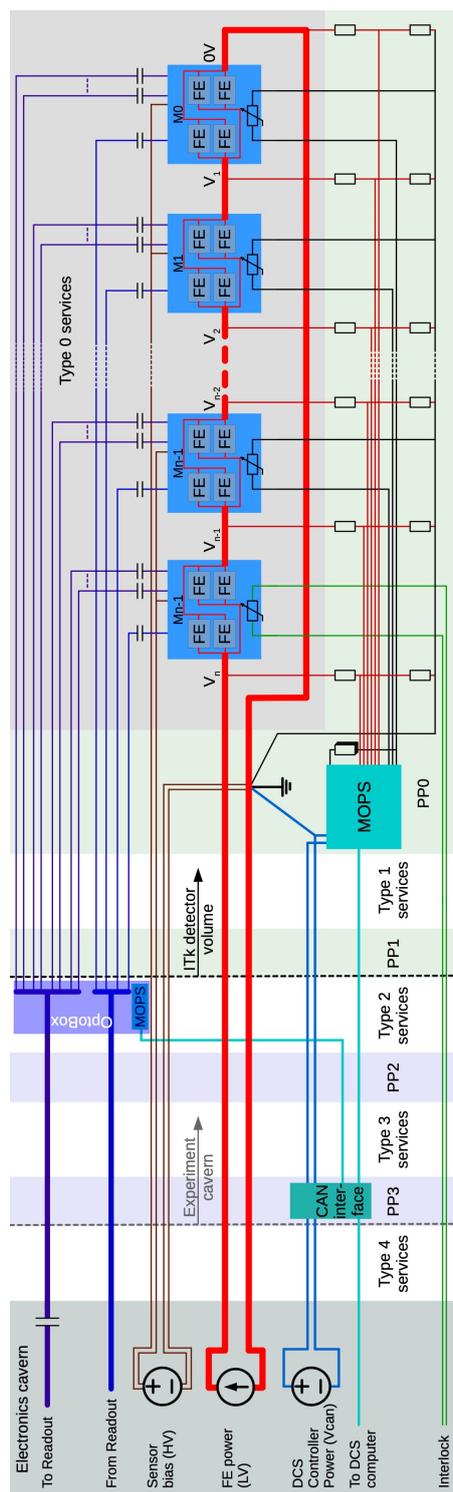
Attention was paid to make the WinCC OA project compatible with different configurations. This accounts for the plan to develop this project further and deploy it for the LLS integration tests, where the two different types of LLS are tested in all available configurations before being integrated into the final detector.

Currently, the DCS project focuses mostly on the control and feedback path of the ITk Pixel DCS, but at some point in the future, the MOPSHUB4Beginners will be replaced by the MOPSHUB. The safety path that is in use right now is very similar to the safety path of the old Pixel detector and will not change significantly in the final design. The diagnostics path is the path with the most undetermined future.

However, all three DCS paths will be necessary to ensure the safe and successful operation of the Pixel detector as a subsystem of the ITk. For the development of the final DCS for the ITk, the project described in this thesis could be used as a basis for inspiration. The aim is to have a fully functional and well-developed system ready for operation in Run 4. Only then can the potential of the HL-LHC be fully exploited. The presented DCS project is a first step in this direction.



A. Connection diagrams



**Figure A.2.:** DCS connection diagram of an SP-chain including the MOPS chip for the ITk Pixel detector. The CAN interface at PP3 is called MOPSHUB and will connect to several MOPS. One NTC of a module will be routed directly to the interlock system. [49]

## B. Resources

A TWiki page with information about the DCS of the ITk Pixel OB demonstrator and the FSM logic was set up at:

<https://twiki.cern.ch/twiki/bin/view/Atlas/ITkPixelOBDemonstratorDCS>

The code for the main WinCC OA project is available in the following repository:

[https://gitlab.cern.ch/atlas-itp-dcs/dcs\\_itkpixsr1st](https://gitlab.cern.ch/atlas-itp-dcs/dcs_itkpixsr1st)



# Bibliography

- [1] ATLAS Collaboration, *Observation of a new particle in the search for the Standard Model Higgs boson with the ATLAS detector at the LHC*, Phys. Lett. B **716** (2012) 1.
- [2] CMS Collaboration, *Observation of a new boson at a mass of 125 GeV with the CMS experiment at the LHC*, Phys. Lett. B **716** (2012) 30.
- [3] O. Aberle et al., *High-Luminosity Large Hadron Collider (HL-LHC): Technical design report*, CERN Yellow Reports: Monographs **10** (2020), ed. by I. B. Alonso et al.
- [4] P. A. Zyla et al., *Review of Particle Physics*, Prog. Theor. Exp. Phys. **2020** (2020) 083C01.
- [5] S. L. Glashow, *Partial-symmetries of weak interactions*, Nucl. Phys. **22** (1961) 579.
- [6] S. Weinberg, *A Model of Leptons*, Phys. Rev. Lett. **19** (1967) 1264.
- [7] A. Salam, *Weak and Electromagnetic Interactions*, Conf. Proc. C **680519** (1968) 367.
- [8] P. W. Higgs, *Broken Symmetries and the Masses of Gauge Bosons*, Phys. Rev. Lett. **13** (1964) 508.
- [9] F. Englert and R. Brout, *Broken Symmetry and the Mass of Gauge Vector Mesons*, Phys. Rev. Lett. **13** (1964) 321.
- [10] L. Evans and P. Bryant, *LHC Machine*, JINST **3** (2008) S08001.
- [11] S. Myers and E. Picasso, *The design, construction and commissioning of the CERN Large Electron–Positron collider*, Contemp. Phys. **31** (1990) 387.
- [12] B. Povh, K. Rith, C. Scholz, F. Zetsche, and W. Rodejohann, *Teilchen und Kerne, Eine Einführung in die physikalischen Konzepte*, 9th ed., Springer Berlin Heidelberg, 2014.
- [13] J. Vollaire et al., *Linac4 design report*, CERN Yellow Reports: Monographs **6** (2020), ed. by M. Vretenar.
- [14] ATLAS Collaboration, *The ATLAS Experiment at the CERN Large Hadron Collider*, JINST **3** (2008) S08003.

## Bibliography

- [15] CMS Collaboration, *The CMS Experiment at the CERN LHC*, JINST **3** (2008) S08004.
- [16] LHCb Collaboration, *The LHCb Detector at the LHC*, JINST **3** (2008) S08005.
- [17] ALICE Collaboration, *The ALICE experiment at the CERN LHC*, JINST **3** (2008) S08002.
- [18] O. Brüning and L. Rossi, *The High Luminosity Large Hadron Collider*, WORLD SCIENTIFIC, 2015.
- [19] A. Dainese et al., *Report on the Physics at the HL-LHC, and Perspectives for the HE-LHC*, tech. rep. CERN-2019-007, CERN, 2019, CDS: 2703572, Addendum: CMS Collaboration, ATLAS Collaboration, *Addendum to the report on the physics at the HL-LHC, and perspectives for the HE-LHC: Collection of notes from ATLAS and CMS*, tech. rep. CERN-2019-007-ADD, 2019, CDS: 2651134.
- [20] M. Thomson, *Modern Particle Physics*, Cambridge University Press, 2013.
- [21] ATLAS Collaboration, *Operation of the ATLAS trigger system in Run 2*, JINST **15** (2020) P10004.
- [22] ATLAS Collaboration, *ATLAS Phase-II Upgrade Scoping Document*, tech. rep. CERN-LHCC-2015-020, 2015, CDS: 2055248.
- [23] ATLAS Collaboration, *ATLAS TDAQ Phase-II Upgrade: Technical Design Report*, tech. rep. CERN-LHCC-2017-020, 2017, CDS: 2285584.
- [24] ATLAS Collaboration, *ATLAS Inner Tracker Pixel Detector: Technical Design Report*, tech. rep. CERN-LHCC-2017-021, 2017, CDS: 2285585.
- [25] ATLAS Collaboration, *ATLAS Inner Tracker Strip Detector: Technical Design Report*, tech. rep. CERN-LHCC-2017-005, CERN, 2017, CDS: 2257755.
- [26] ATLAS Collaboration, *ATLAS Inner Detector: Technical Design Report, Volume 1*, tech. rep. CERN-LHCC-97-016, 1997, CDS: 331063.
- [27] ATLAS Collaboration, *ATLAS Insertable B-Layer Technical Design Report*, tech. rep. CERN-LHCC-2010-013, 2010, CDS: 1291633, Addendum: *ATLAS Insertable B-Layer Technical Design Report Addendum*, tech. rep. CERN-LHCC-2012-009, 2012, CDS: 1451888.
- [28] I. Perić et al., *The FEI3 readout chip for the ATLAS pixel detector*, Nucl. Instrum. Meth. A **565** (2006) 178.
- [29] ATLAS Collaboration, *ATLAS pixel detector electronics and sensors*, JINST **3** (2008) P07007.

- [30] A. Ahmad et al., *The silicon microstrip sensors of the ATLAS semiconductor tracker*, Nucl. Instrum. Meth. A **578** (2007) 98.
- [31] ATLAS Collaboration, *Operation and performance of the ATLAS semiconductor tracker*, JINST **9** (2014) P08009.
- [32] D. Attree et al., *The evaporative cooling system for the ATLAS inner detector*, JINST **3** (2008) P07003.
- [33] ATLAS Collaboration, *Operation and performance of the ATLAS semiconductor tracker in LHC Run 2*, JINST **17** (2022) P01013.
- [34] ATLAS Collaboration, *Performance of the ATLAS Transition Radiation Tracker in Run 1 of the LHC: tracker properties*, JINST **12** (2017) P05002.
- [35] ATLAS Collaboration, *The ATLAS Inner Detector commissioning and calibration*, Eur. Phys. J. C **70** (2010) 787.
- [36] ATLAS Collaboration, *Production and integration of the ATLAS Insertable B-Layer*, JINST **13** (2018) T05008.
- [37] M. Garcia-Sciveres et al., *The FE-I4 pixel readout integrated circuit*, Nucl. Instrum. Meth. A **636** (2011) S155.
- [38] ATLAS Collaboration, *ATLAS Liquid Argon Calorimeter: Technical Design Report*, tech. rep. ATLAS-TDR-2, 1996, CDS: 331061.
- [39] ATLAS Collaboration, *ATLAS Calorimeter Performance: Technical Design Report*, tech. rep. CERN-LHCC-96-040, 1996, CDS: 331059.
- [40] ATLAS Collaboration, *Operation and performance of the ATLAS Tile Calorimeter in Run 1*, Eur. Phys. J. C **78** (2018) 987.
- [41] ATLAS Collaboration, *ATLAS Muon Spectrometer: Technical Design Report*, tech. rep. CERN-LHCC-97-022, 1997, CDS: 331068.
- [42] ATLAS Collaboration, *Expected tracking and related performance with the updated ATLAS Inner Tracker layout at the High-Luminosity LHC*, tech. rep. ATL-PHYS-PUB-2021-024, CERN, 2021, CDS: 2776651.
- [43] D. Álvarez, *Design Overview of the Bare Local Supports for the ITkPixel Outer Barrel, ITk Pixel –Local Supports for the Outer Barrel*, tech. rep. AT2-IP-ER-0013, CERN, 2021, EDMS: 2632352 v.1.
- [44] M. Garcia-Sciveres, *RD53A Integrated Circuit Specifications*, tech. rep. CERN-RD53-PUB-15-001, CERN, 2015, CDS: 2113263.

## Bibliography

- [45] M. Garcia-Sciveres, *The RD53A Integrated Circuit*, tech. rep. CERN-RD53-PUB-17-001, CERN, 2017, eprint: [https://cds.cern.ch/record/2287593/files/RD53A\\_Manual\\_V3-51.pdf](https://cds.cern.ch/record/2287593/files/RD53A_Manual_V3-51.pdf).
- [46] RD53 Collaboration, *RD53B users guide*, tech. rep. CERN-RD53-PUB-21-001, CERN, 2020, CDS: 2754251.
- [47] RD53 Collaboration, *RD53B manual, Detailed manual for ATLAS*, tech. rep. CERN-RD53-PUB-19-002, version 2.18, CERN, 2022, CDS: 2665301.
- [48] M. Karagounis et al., *An integrated Shunt-LDO regulator for serial powered systems*, 2009 Proceedings of ESSCIRC (2009) 276.
- [49] D. Giugni et al., *Specifications for the ATLAS ITk Pixel Services, Technical Requirements*, tech. rep. AT2-IP-EP-0007, version 2, ATLAS Collaboration, 2020, EDMS: 1817540 v.3.
- [50] A. Walsemann et al., *A CANopen based prototype chip for the Detector Control System of the ATLAS ITk Pixel Detector*, PoS **TWEPP2019** (2020) 013.
- [51] R. Ahmad, M. Karagounis, S. Kersten, N. Lehmann, and C. Zeitnitz, *Specification for the Pixel DCS ASIC: Monitoring Of Pixel System*, tech. rep. AT2-IP-ES-0001, 2021, EDMS: 1909505 v.3.
- [52] A. B. Poy et al., *The detector control system of the ATLAS experiment*, JINST **3** (2008) P05006.
- [53] O. Holme, M. González-Berges, P. Golonka, and S. Schmeling, “The JCOP Framework”, *10th ICALEPCS Conference*, Geneva, 2005, CDS: 907906.
- [54] T. Henss et al., *The hardware of the ATLAS Pixel Detector Control System*, JINST **2** (2007) P05006.
- [55] G. Schnell and B. Wiedemann, *Bussysteme in der Automatisierungs- und Prozesstechnik*, 9th ed., Springer Fachmedien Wiesbaden, 2019.
- [56] W. Mahnke, S.-H. Leitner, and M. Damm, *OPC Unified Architecture*, Springer Berlin Heidelberg, 2009.
- [57] P. P. Nikiel, B. Farnham, S. Schlenker, H. Boterenbrood, and V. Filimonov, “OPC Unified Architecture within the Control System of the ATLAS Experiment”, *14th ICALEPCS Conference*, 2014, CDS: 1696973.
- [58] C. Gaspar, M. Dönszelmann, and P. Charpentier, *DIM, a portable, light weight package for information publishing, data transfer and inter-process communication*, Comput. Phys. Commun. **140** (2001) 102.

- [59] P. P. Nikiel, B. Farnham, V. Filimonov, and S. Schlenker, *Generic OPC UA Server Framework*, J. Phys. Conf. Ser. **664** (2015) 082039.
- [60] JCOP project, *Wiener OPC-UA User Manual*, ed. by B. Farnham, version 1.2.0, 2021, eprint: [https://edms.cern.ch/ui/file/2169631/1.2.0/Wiener\\_OPCUA\\_Server\\_User\\_Manual.pdf](https://edms.cern.ch/ui/file/2169631/1.2.0/Wiener_OPCUA_Server_User_Manual.pdf).
- [61] R. Kopeliansky and S. Schlenker, *DCS: Requirements Document for HL-LHC*, tech. rep. ATU-GE-ES-0004, ATLAS Collaboration, 2020, EDMS: 2276493 v.1.
- [62] A. Qamesh, *MOPSHUB status report*, ATLAS Upgrade Week, 2021, URL: <https://indico.cern.ch/event/1092015/>.
- [63] C. Gemme et al., *ITk Interlock System Strategy*, tech. rep. AT2-IE-ES-0005, ATLAS Collaboration, 2021, EDMS: 2387552 v.7.
- [64] J. Wang and W. Tepfenhart, *Formal Methods in Computer Science*, Chapman and Hall/CRC, 2019.
- [65] J. H. Conway, *Regular Algebra and Finite Machines*, Dover Publications, 2012, ISBN: 9780486485836.
- [66] B. Franek and C. Gaspar, *SMI++ object oriented framework for designing and implementing distributed control systems*, IEEE Trans. Nucl. Sci. **45** (1998) 1946, ed. by J. P. Dufey.
- [67] C. Gaspar and B. Franek, *Tools for the automation of large distributed control systems*, IEEE Trans. Nucl. Sci. **53** (2006) 974.
- [68] M. Hamer, *Requirements for the Power Supply System of the ATLAS ITk Pixel Detector*, tech. rep. AT2-IP-ES-0014, ATLAS Collaboration, 2019, EDMS: 2192745 v.2.
- [69] D. Ecker, *Implementation of the communication and data visualization for the ITk Pixel monitoring chip of the ATLAS experiment at CERN*, MA thesis, Bergische Universität Wuppertal, 2021.
- [70] S. Kersten, P. Kind, C. Riegel, and N. Lehmann, *The Interlock Matrix Crate of the ATLAS IBL detector*, tech. rep. ATL-IP-ES-0203, ATLAS Collaboration, 2021, EDMS: 1302654 v.1.
- [71] S. Schäpe, *ITk SR1 CO2 Cooling Plant for Pixel*, 2019, URL: <https://twiki.cern.ch/twiki/bin/view/Atlas/ITkSR1CO2PlantPixeloperation>.
- [72] ATLAS DCS, *ATLAS DCS Integration Guidelines*, tech. rep. ATLAS-DQ-ON-0013, 2007, EDMS: 685451 v.3.

## Bibliography

- [73] ATLAS DCS, *FSM Integration Guideline*, tech. rep. ATL-DQ-ON-0010, 2021, EDMS: 685114 v.3.0.
- [74] M. Hamer, *ITk Pixel Connectivity Spreadsheet for Electrical Services*, tech. rep. AT2-IP-ES-0016, version 1.9, ATLAS Collaboration, 2021, EDMS: 2423241 v.1.
- [75] S. Gonzalez, *ITk Pixel Electronics ICD - Outer Barrel interfaces*, tech. rep. AT2-IP-MG-0008, CERN, 2021, EDMS: 2379179 v.1.

# Danksagung

Das Verfassen einer Masterarbeit ist nicht möglich ohne die Unterstützung und Hilfe von anderen. So auch hier. Ich möchte mich darum bei den Personen bedanken, die mir im Verlauf der Arbeit mit Rat und Tat zur Seite gestanden haben. Andrés, vielen Dank für die Einführung in das Thema und in die Arbeitsgruppe. Mein herzlicher Dank geht auch an Susanne Kersten für die Erklärungen zu Feinheiten der verwendeten Hardware und für das Herstellen von Kontakten im ATLAS Umfeld. Zu großem Dank bin ich auch Benedikt Vormwald verpflichtet, der wirklich jede Frage zu seinem Demonstrator aber auch allgemein zum ATLAS Detektor beantwortet hat und mich das große Ganze hat sehen lassen. Es ist eine Freude mit ihm zusammenzuarbeiten. Die Arbeit erst ermöglicht hat Prof. Dr. Stan Lai. Man kann sich keinen besseren Betreuer vorstellen. Durch ihn sind auch die Aufenthalte am CERN zustande gekommen. Ganz herzlichen Dank dafür.

Auch meinen Eltern bin ich für ihre Unterstützung insbesondere im letzten Monat dankbar. Zu guter Letzt, und mit am wichtigsten, ein ganz herzliches Dankeschön an meine Freundin Daria. Ohne dich hätte ich es nicht so weit geschafft.

**Erklärung**

nach §17(9) der Prüfungsordnung für den Bachelor-Studiengang Physik und den Master-Studiengang Physik an der Universität Göttingen:  
Hiermit erkläre ich, dass ich diese Abschlussarbeit selbständig verfasst habe, keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe und alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten Schriften entnommen wurden, als solche kenntlich gemacht habe.

Darüberhinaus erkläre ich, dass diese Abschlussarbeit nicht, auch nicht auszugsweise, im Rahmen einer nichtbestanden Prüfung an dieser oder einer anderen Hochschule eingereicht wurde.

Göttingen, den 8. Juni 2022

(Hans Joos)