



Summer Student Report

Untangling the Cables: Streamlining Muon Detector Operation

Jonas Kann

jonas.kann@rwth-aachen.de

Supervisors: Andrea Contu, Vladimir Chulikov, Barbara Sciascia

Abstract

This project focused on streamlining the complex cabling of the LHCb muon detector by developing a relational database that efficiently stores and organises the connections between its components. The project involved cleaning and restructuring data, creating and populating the database, and designing a graphical user interface for easy access to this information within the control room. The successful implementation of these tools has enhanced the operational efficiency of the muon detector and simplified troubleshooting, which is crucial for minimising downtime and preventing data loss during experiments.

Geneva, Switzerland

September 5, 2024

Contents

1	Introduction to the muon system	2
1.1	The DAQ Chain	3
1.2	The HV Chain	3
2	The Cabling Challenge	5
3	Database Creation	6
3.1	Visualisation of the Database	7
4	Database Implementation	9
5	Conclusion	10
5.1	Additional Work	11
5.2	Prospects	11
A	Snippet of the SQL Code	13

1 Introduction to the muon system

At LHC energies muons behave as minimum ionising particles and are therefore capable of traversing the trackers and calorimeters with minimal energy loss. To account for this, the muon detector is positioned at the end of the LHCb detector and the largest sub-system within the experiment.

Although the muon detector can theoretically also be used to determine the transverse momentum, this task is left to the tracking stations, which are more suitable due to their better spatial resolution. Instead, the muon detector plays a very crucial part in the particle identification, trying to distinguish muons from other charged particles with the maximum efficiency.

Since muons are a part of the final stages of many rare decays of heavy hadrons, this identification is vital for the physics studied at LHCb.

The muon system can be subdivided into four stations, four quadrants and four regions. This is displayed in Figure 1.

Each station consists of 276 Multi Wire Proportional Chambers (MWPC), measuring the incoming charged particles. More information on the MWPCs can be found in the Muon TDR [2]. The induced signals are then collected and processed by the corresponding readout electronics.

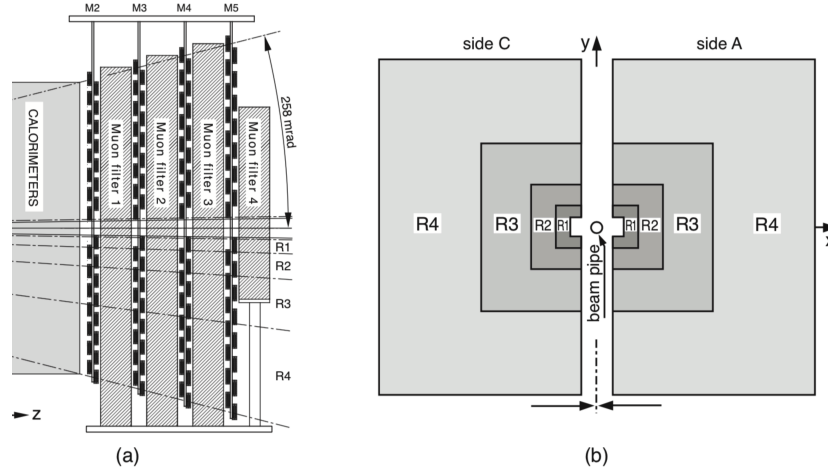


Figure 1: (a) Side view of the LHCb muon detector with the different stations. (b) Different regions within one station [1].

1.1 The DAQ Chain

The electrical signals generated in the chambers have to be processed and collected in order to get valuable information for physics analysis. This is done by the data acquisition (DAQ) chain.

When a muon traverses a Multi-Wire Proportional Chamber (MWPC), it induces a signal in the electrodes. This signal is then transmitted to the front-end board (FEB), where the initial processing, including signal shaping, amplification and discrimination, takes place. Depending on the chamber's region, each MWPC can be connected to up to 14 FEBs. The logical channels of the FEB are subsequently connected to a different component, the new off-detector electronics (nODE). There are 144 nODEs in the system, each equipped with four nSYNC chips. The nSYNC chips are responsible for further signal processing including bunch crossing alignment, clock synchronisation, buffering, measuring hit arrival times and provide the capability to delay different channels in increments of 25 ns, facilitating coarse time alignment. These nSYNC chips can handle up to 48 channels each.

A picture of an ODE and the chambers in one station is displayed in Figure 2.

1.2 The HV Chain

Apart from the DAQ chain, the chambers also have to be connected to the HV supply chain, ensuring the operation in the proportional region. This is done in two different ways:

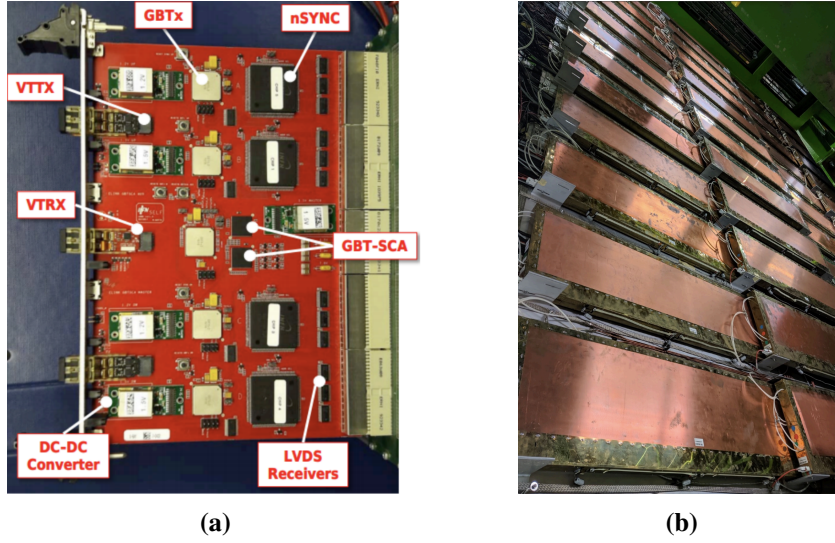


Figure 2: Components of the DAQ chain. **(a)** Structure of an ODE including the four SYNCs. **(b)** MPWCs in one station. [3]

1. **The CAEN HV-system** - This supply chain is used in the inner regions R1 and R2.
2. **The PNPI HV-system** - This system supplies the chambers in the outer regions R3 and R4.

The CAEN chain is organised as follows: The chambers are connected to a component known as the EasyBoard (CAEN A3535P). There are nine EasyBoards on each side of the detector, with each EasyBoard supplying eight chambers. These EasyBoards are then connected to an EasyCrate (CAEN 3000), which contains five slots for EasyBoards. Consequently, two EasyCrates are required per side of the detector. The EasyCrates are monitored by a Branch Controller (CAEN 3535A, one per side) and receive high voltage from the CAEN 1527 power supply.

The EasyBoard and EasyCrate are displayed in Figure 3.

The PNPI system is structured in a similar manner. In this system, the chambers are connected to Remote Distribution Boards (RDBs), which provide high voltage to the chambers, measure the currents, and regulate the voltages. Each RDB can support up to nine chambers. These RDBs are then connected to Master modules, which control the voltages supplied to the RDBs. There are four Master modules per quadrant. Finally, the Master modules are connected to the high-voltage supply, Matsusada.

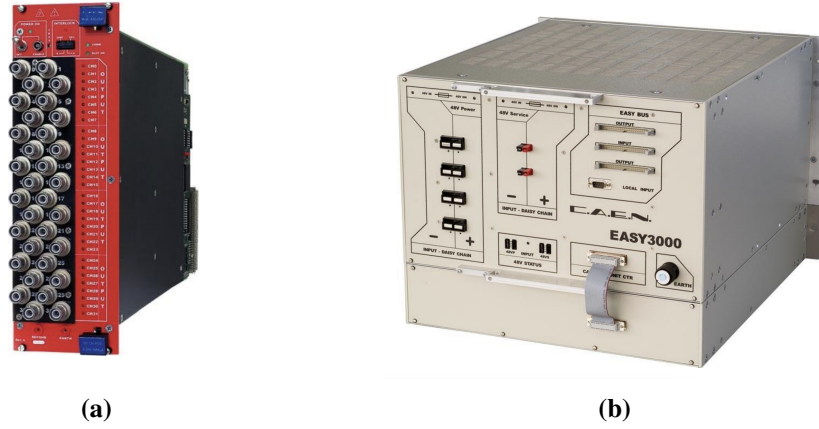


Figure 3: The CAEN HV-system. (a) The EasyBoard / CAEN A3535P. (b) The EasyCrate / CAEN 3000. [4]

2 The Cabling Challenge

To summarise the cabling configuration of the chambers, Figure 4 presents an overview of the setup.

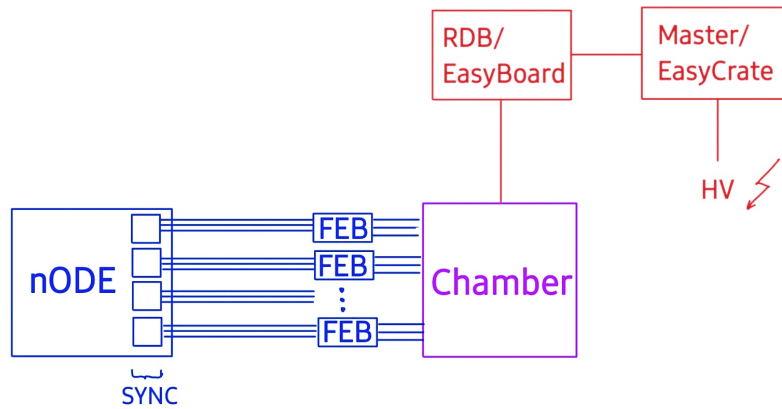


Figure 4: Cabling scheme - simplified

However, comparing the numbers — 144 ODEs and 1,104 chambers — it becomes evident that a one-to-one mapping between ODEs and chambers is not realisable. In practice, the wiring is significantly more complex, as depicted in Figure 5.

Generally, one nODE is connected to multiple chambers, but this only captures

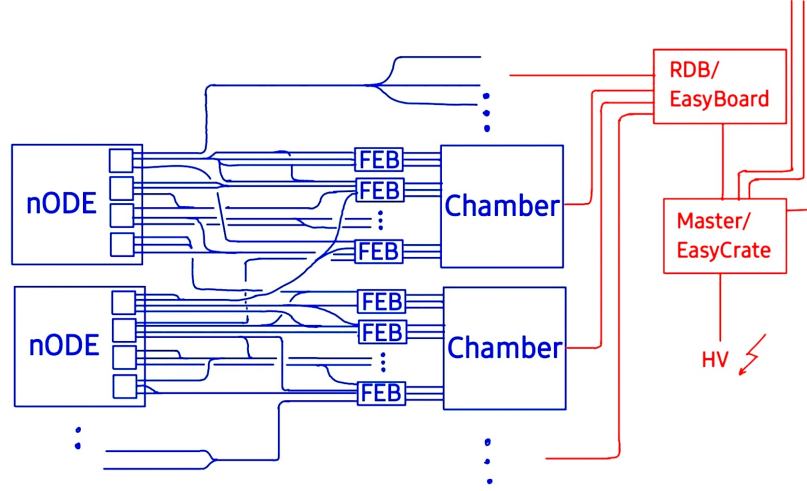


Figure 5: Wiring scheme - less simplified

part of the full complexity. Some specific chambers in region 3 are connected to multiple nODEs. The wiring between the nSYNCs and the FEBs is particularly complicated, involving approximately 50,000 connections in total.

The relationship between the logical channels of the FEBs and the channels of the SYNCs is a many-to-many relationship, meaning that a single FEB channel could be connected to multiple SYNC channels, and conversely, multiple FEB channels could be linked to a single SYNC channel. These connections are managed through logical operations such as OR, as illustrated in Figure 5, although the actual number of wires and connections is far larger.

This complex wiring between the nODEs and chambers poses significant challenges for maintenance and troubleshooting, as the connections are highly tangled. However, the relationships between the physical components are well-defined, allowing for the development of a relational SQL database that contains detailed information about the entire wiring configuration.

3 Database Creation

In order to build the database, I was provided with multiple txt-files, which contained the information about the different connections between the physical components.

The first stage of processing was to clean and reformat the data using python and pandas. Unique identifiers for each components were created and the data was restructured into different tables.

The subsequent step was to construct the database. The **Structured Query Language (SQL)** is a language used to store and manage data, especially in a relational database management system (RDBMS). Since the database had to be integrated into the muon operation system at a later stage (see section 4) and the already existing databases were running in the Oracle database management system, Oracle was the natural choice as RDBMS.

In this system and on a dedicated server, the database was created and the tables and their corresponding columns were inserted. A snippet of the relevant SQL code is given in the Appendix A.

Every table needs a primary key, a unique identifier by which every component can be accessed. Additionally the relationships between the tables are realised by foreign keys, a set of attributes that refer to the primary key of another table, linking these two tables making cross-referencing possible.

Finally, the database was filled with data by creating a connection to the database via python and pushing the processed data to the database.

3.1 Visualisation of the Database

Relational databases can be visualised by an Entity-Relationship-Diagram (ERD). An ERD shows the different tables (entities) as blocks that contain the corresponding columns (entity attributes). Between these different entities, the relationships are displayed by arrows. The ERD for the newly created database is shown in Figure 6.

When looking at the ERD, one might notice similarities to the wiring scheme in Figure 5. This is not a coincidence, since the relational database serves as a virtual representation of the physical components. The relationships between the tables in the database mirror the cables and connections between the electrical components.

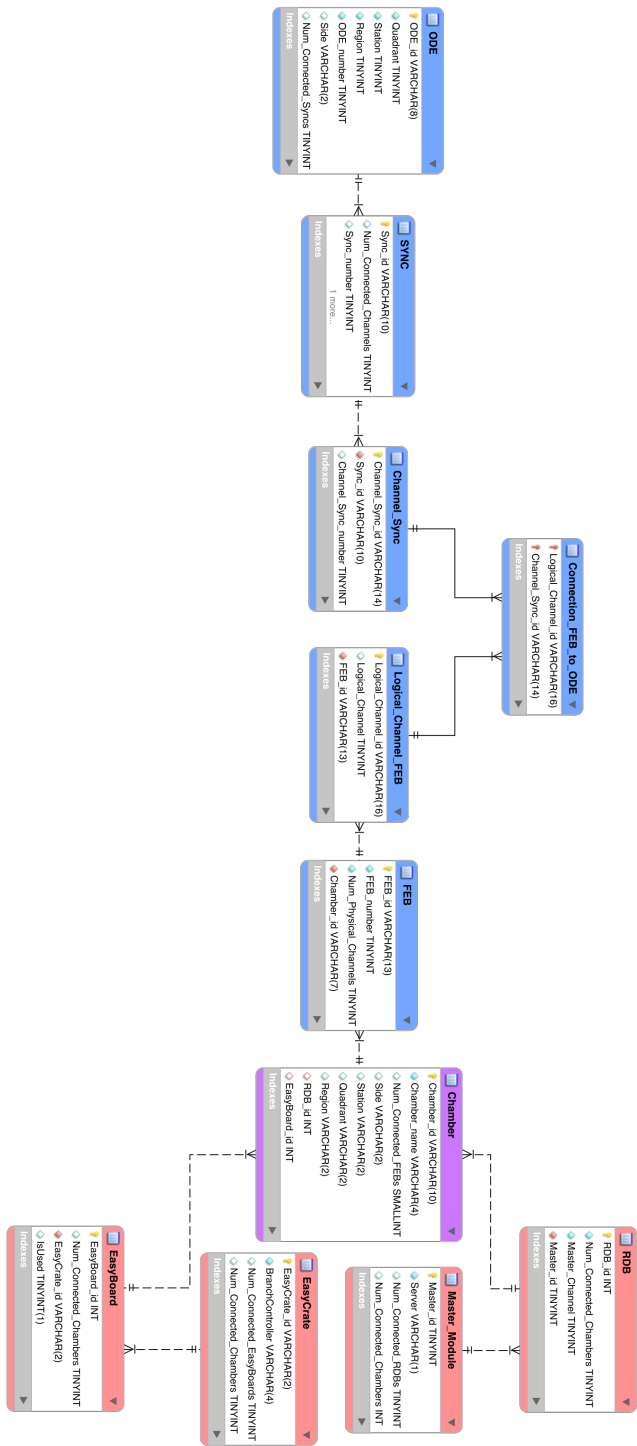


Figure 6: Entity-Relationship-Diagram

4 Database Implementation

The database has been successfully completed and is now running on a dedicated server. However, the objective goes beyond simply storing information about the connections; it is also crucial to make this information accessible where it is most needed: the control room. To achieve this, the next phase of my summer student project involved designing a graphical user interface (GUI) that allows users to query information from the database efficiently.

The tool selected for this task is the WinCC SCADA (Supervisory Control and Data Acquisition) software, which provides the capability to create custom control panels and is the standard framework for control software all around CERN. The graphical editor of the WinCC software, as shown in Figure Figure 7, enables users to design the panel using various widgets such as buttons, tables, and text fields. Additionally, each widget can be associated with specific actions, such as a button click, which can be programmed using a C-like scripting language called CTRL. These scripts are executed when the corresponding action is performed, allowing for the creation of a fully functional control panel.

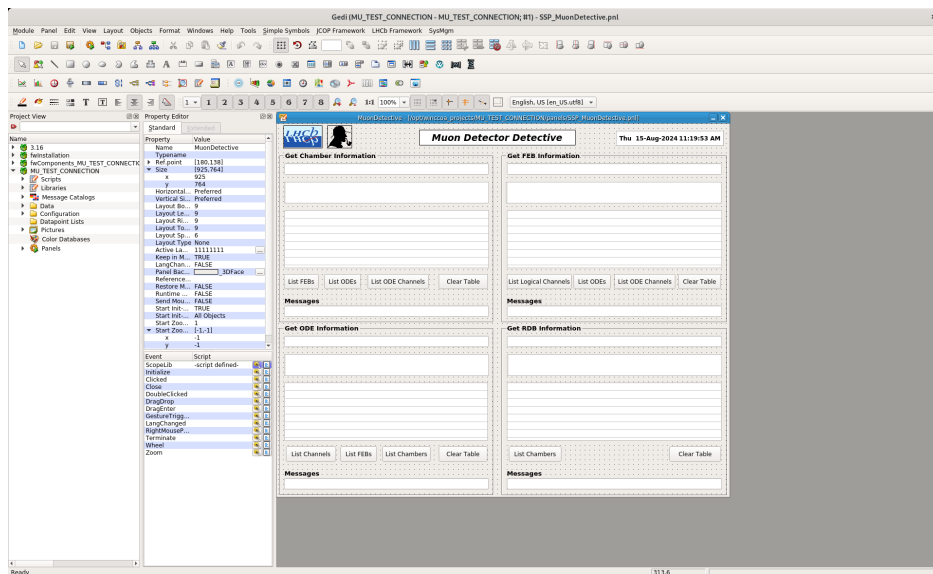


Figure 7: WinCC Graphical Editor

Finally, the panel ended up looking like depicted in Figure 8 and works as follows:

- The user can enter the identifier of the corresponding component into a text field.

Untangling the Cables: Streamlining Muon Detector Operation Jonas Kann

- As soon as the component is found in the database, a corresponding feedback message is given and the fundamental information about that component (location, number of channels, high voltage supply et cetera) is displayed in a table.
- Additionally, the user has the option to list all the different devices connected to the given component.

With this tool, the muon experts and piquets can obtain relevant information about the different components by entering the relevant identifier giving the panel its name: The Muon Detector Detective.

Get Chamber Information

M3A22B

Quadrant	Station	Region	RDB	Master_CH	Master_Module	Server
1	3	3	84	7	12	A

FEB id	FEB number	# Physical Channels
M3A22B-06	6	2
M3A22B-07	7	2
M3A22B-08	8	2
M3A22B-09	9	2
M3A22B-10	10	2
M3A22B-11	11	2

List FEBs List ODEs List ODE Channels Clear Table

Messages

Chamber found.

Get FEB Information

M3A22B-06

Quadrant	Station	Region	Chamber	FEB	# Physical Channels
1	3	3	22B	6	2

ODE id	ODE number	# Used Channels
Q1M3R32	2	168

List Logical Channels List ODEs List ODE Channels Clear Table

Messages

FEB found.

Get ODE Information

Q1M3R32

Quadrant	Station	Region	# Connected Sync	# Connected Channels
1	3	3	4	168

Sync	Channel	# Connected Chambers
3	41	2
3	42	2
3	43	1
3	44	1
3	45	1
3	46	1

List Channels List FEBs List Chambers Clear Table

Messages

ODE found.

Get RDB Information

84

Master Module	Master Channel	Server	# Connected Chambers
12	7	A	9

List Chambers Clear Table

Messages

RDB found.

Figure 8: WinCC Running Panel

5 Conclusion

The aim of this summer student project was to systematically reorganise the complex wiring, thereby improving the clarity and manageability of the various connections. This task was completed successfully by creating a relational database

that represents the physical wiring and efficiently stores all this information.

Moreover, the designed control panel is functioning as intended, providing users with quick and convenient access to relevant information. This advancement significantly improves the operational efficiency of the muon detector and facilitates more effective troubleshooting in the event of errors during runtime.

5.1 Additional Work

The newly created database can also be used to design new panels. In addition to the Muon Detector Detective panel, I was also working on a panel with the purpose of simplifying the process of masking nODE channels.

When masking a nODE channel, we decide to deactivate the channel ignoring the data coming via that channel. This can have different reasons, for example that the channel is noisy or faulty. Therefore, we avoid errors being up thrown by the ECS with the cost of data loss.

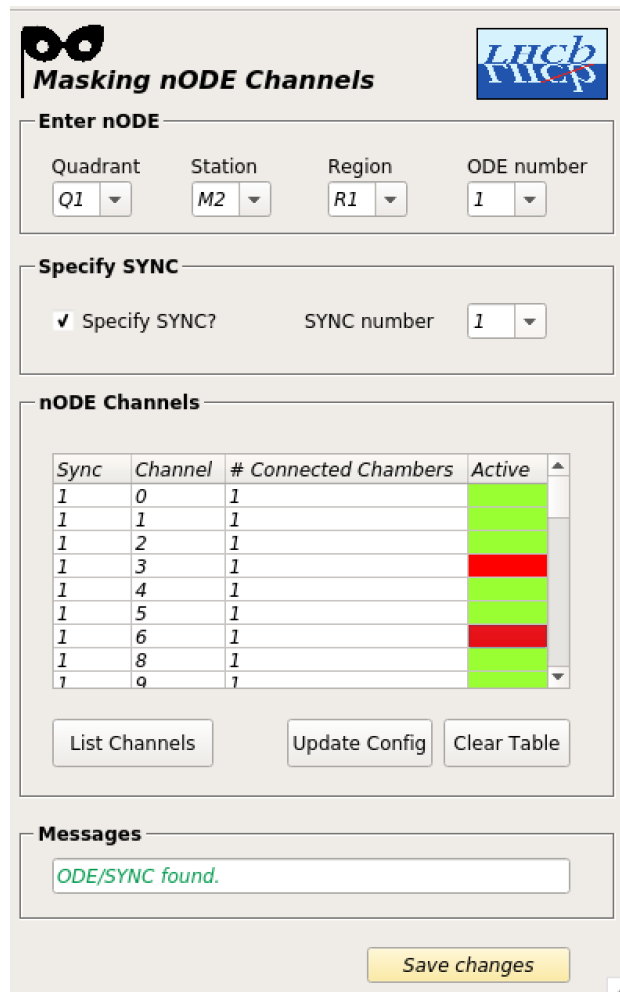
The information if a channel is masked or not is contained by a config file listing all the channels with their masking state, 0 or 1. The newly designed panel (shown in Figure 9) allows the user to select a specific nODE/nSYNC and list all their channels (queried from the database) along with their masking state (queried from the config file). By clicking on the masking state, the value gets flipped between masked (red) and unmasked(green). When then clicking the “Update config” button, a backup of the config file will be created and then the file will be modified based on the changes made in the panel. In the end, a C program has to be run to apply the changes to the hardware.

To conclude, this panel works as intended and serves as a graphical interface making the process of masking nODE channels accessible.

5.2 Prospects

Regarding the prospects, there are several ways to extend the project and the functionalities of the developed tools.

In the near future, it may be of interest to connect the panel to the rest of the muon experimental control system (ECS) meaning that the user can easily get forwarded between the different panels. Additionally, the database can be used to design even more panels for different tasks, e.g. setting time delays of the different channels for improved time alignment. Furthermore, there are different physical components in the DAQ chain (e.g. intermediate boards) that could still be included in the database.



Masking nODE Channels

Enter nODE

Quadrant: Q1 Station: M2 Region: R1 ODE number: 1

Specify SYNC

☒ Specify SYNC? SYNC number: 1

nODE Channels

Sync	Channel	# Connected Chambers	Active
1	0	1	Active
1	1	1	Active
1	2	1	Active
1	3	1	Masked
1	4	1	Active
1	5	1	Active
1	6	1	Masked
1	8	1	Active
1	9	1	Active

List Channels Update Config Clear Table

Messages

ODE/SYNC found.

Save changes

Figure 9: WinCC Panel for masking nODE channels

Acknowledgements

I would like to sincerely thank my supervisors Andrea Contu, Vladimir Chulikov and Barbara Sciascia for their support and mentorship this summer. Their guidance and willingness to address every question greatly contributed to my learning experience. I also want to thank Sofia Katriakhova for her help with the integration of the database into the server. Lastly, I want to thank Professor Johannes Erdmann from RWTH Aachen for supporting me during the application process, making this unique experience possible.

References

- [1] P. Albicocco, W. Baldini, S. Belin, V. Bocci, N. Bondar, D. Brundu, S. Cadeddu, A. Cardini, M. Carletti, G. Chiodi, et al., *A Method Based on Muon System to Monitor LHCb Luminosity*, Symmetry 2022, 14, 860. <https://doi.org/10.3390/sym14050860>
- [2] LHCb Collaboration, *LHCb muon system : Technical Design Report*, Geneva : CERN, 2001, <https://cds.cern.ch/record/504326?ln=de>
- [3] L. Paolucci, *The LHCb Muon Detector Upgrades*, ICHEP 2022. https://agenda.infn.it/event/28874/contributions/169040/attachments/94683/129724/ICHEP_2022_LPaolucci.pdf
- [4] O. Maev, *Piquet Training for Run 3 - HV system*, 26.07.2022.

A Snippet of the SQL Code

```
-- Table structure for EasyBoard table
CREATE TABLE "SSP_EasyBoard" (
  "EasyBoard_id" NUMBER(19) NOT NULL,
  "Num_Connected_Chambers" NUMBER(5) DEFAULT NULL,
  "EasyCrate_id" VARCHAR2(2) NOT NULL,
  "IsUsed" NUMBER(1) DEFAULT NULL,
  PRIMARY KEY ("EasyBoard_id"),
  CONSTRAINT "FK_EasyCrate_TO_EasyBoard"
    FOREIGN KEY ("EasyCrate_id")
    REFERENCES "SSP_EasyCrate" ("EasyCrate_id")
);

CREATE INDEX "idx_EasyCrate_id"
  ON "SSP_EasyBoard" ("EasyCrate_id");

-- Dumping data for table "SSP_EasyBoard"
LOCK TABLE "SSP_EasyBoard" IN EXCLUSIVE MODE;
INSERT ALL
  INTO "SSP_EasyBoard" ("EasyBoard_id", "Num_Connected_Chambers",
    "EasyCrate_id", "IsUsed")
    VALUES (53,8,'A2',1)
  INTO "SSP_EasyBoard" ("EasyBoard_id", "Num_Connected_Chambers",
    "EasyCrate_id", "IsUsed")
    VALUES (67,8,'C2',1)
  ...
SELECT * FROM dual;
COMMIT;
```