Physik - Department

Lehrstuhl E12 für Experimentalphysik

A new controller for the HADES RICH gas supply system

Bachelor thesis

Korbinian Schmidt-Sommerfeld
2011-07-25

Technische Universität München

Erstgutachter (Themensteller):	Prof. R. Krücken
Stellvertreter	Prof. S. Bishop
Zweitgutachter:	Prof. L. Oberauer
Wissenschaftlicher Betreuer:	Dr. J. Friese

Abstract

In the future, the accelerators at the GSI Helmholtzzentrum in Darmstadt, Germany, will be able to produce heavy ion particle beams with energies up to 34 $AGeV$. This offers totally new possibilities for the HADES detector. Fireballs with temperatures $T \approx 120$ MeV and baryonic densities which are six to eight times higher than nuclear ground state densities will be used to investigate the properties of hadrons in a strongly interacting medium. Before these experiments can be performed, HADES and especially the RICH detector have to be updated. An extension of the RICH gas supply system is necessary for reliable measurements at energies of more than 3 $AGeV$. This thesis reports on the installation, programming and testing of a new controller for the gas system which enables future operation in experiments at higher energies.

Contents

Chapter 1

Introduction

A hot topic in current research in particle and nuclear physics is the behavior of hadrons produced and propagating in strongly interacting nuclear matter [1] [2]. The latter may be formed as so-called cold nuclear matter when the hadrons of interest are produced by γ- or p-induced reactions of target nuclei in their ground state. Alternatively, fireballs of hot and dense baryonic matter can be produced in heavy ion reactions at relativistic and ultrarelativistic projectile energies. The **H**igh-**A**cceptance **DiE**lectron **S**pectrometer (HADES, fig. 1.1) [3] was built at the GSI Helmholtzzentrum für Schwerionenphysik in Darmstadt for the systematic study of in-medium properties of the light vector mesons ρ, ω and ϕ. Their electromagnetic decay branch into e^+/e^- pairs provides a clean signal allowing the designation of different matter phases because it is not subject to strong final-state interaction. The dilepton decay of vector mesons is strongly suppressed (by a factor of 10^{-5}) as compared to hadronic decay modes. This results in an enormous background which is suppressed to a large extent by a hadron blind **R**ing **I**maging **Ch**erenkov (RICH) detector for e^+/e^- identification [4]. Besides the RICH detector, which is the innermost component, HADES features six identical sectors which cover 85% of the azimuthal area in the forward hemisphere. Each sector consists of two sets of Mini-Drift Chambers (MDC) with four modules per sector which are placed on either sides of the superconducting coils that produce a toroidal magnetic field. The MDC are used to measure particle momenta. The outermost components are time-of-flight walls (TOF/TOFINO) and the Pre-Shower detector just behind the TOFINO [3].

The RICH detector consists mainly of a radiator and a large area photon detector. Both parts are operated with a different gas at normal pressure and separated by a thin and therefore fragile CaF_2 window, which must not be exposed to big pressure differences. Both gases have to be kept at high purity in order to enable a proper and efficient performance of the detector. This is achieved by continuous flushing, purification and recirculation of the gases. This requires a complex gas system and reliable controlling and monitoring. The presently installed system is already 12 years old and spare parts for the system controller (PLC) are no longer available. For this reason the installation of a modern PLC is needed.

Most of the present HADES experiments performed with beam energies from 1 to 2 GeV [5][6]. This corresponds to baryonic densities $\rho_B \approx 3 \cdot \rho_0$ and maximum temperatures of about 80 MeV. Temperatures of 150 MeV and higher have been reached at the RHIC and the LHC. Experiments at energies of order 10 GeV corresponding to $T \approx 120$ MeV and $\rho_B \approx 6 - 8 \cdot \rho_0$, are planned to be carried out at the **F**acility for **A**ntiproton and **I**on **R**esearch (FAIR). In this context extensions of the existing detector necessary, for higher beam energies, are foreseen, too. The upgrade of the RICH gas system for higher beam energies is going to be a major task and is discussed in this thesis.

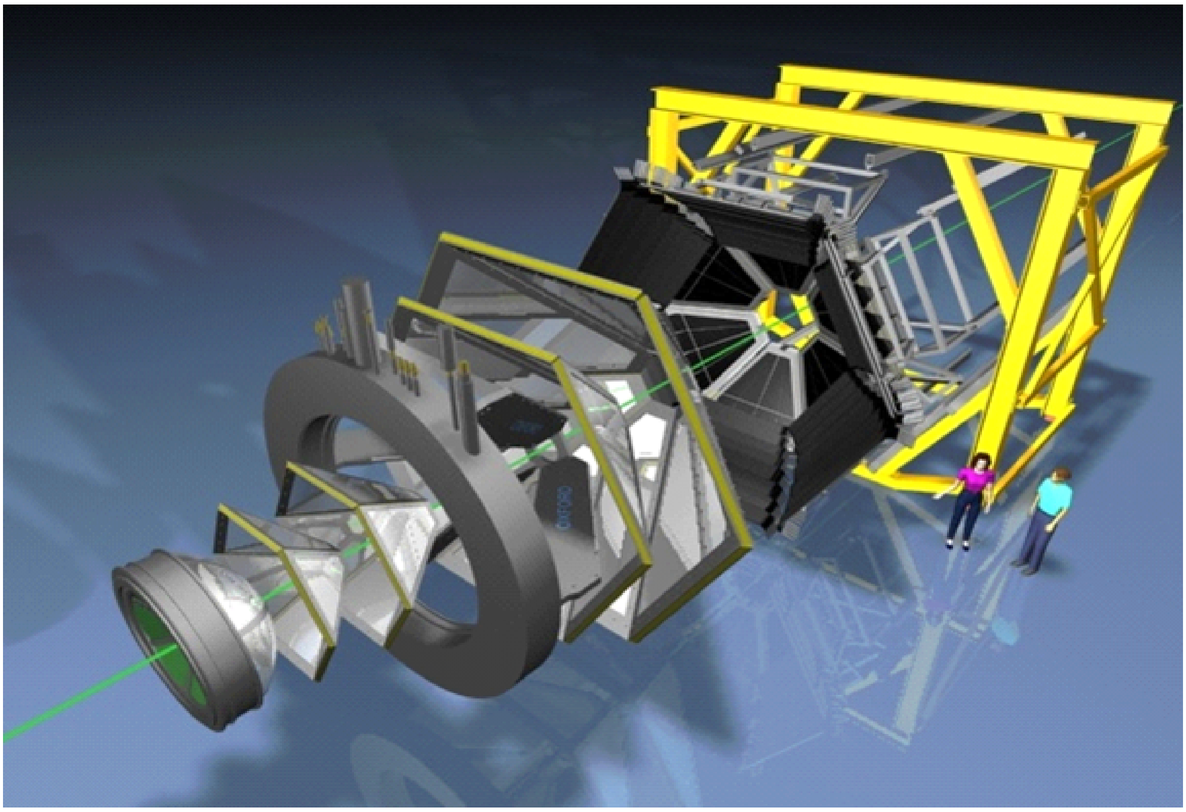


Fig. 1.1: Schematic picture of HADES with the RICH, MDC I-IV, the superconducting magnet and TOF/TOFINO detectors in a stretched view. The beam enters from the left.

Chapter 2

The RICH detector

2.1 Cherenkov radiation

The RICH detector uses the Cherenkov effect to distinguish between e^+/e^- and hadrons like pions and protons [7].

If charged particles propagate with a velocity $\beta = \frac{v}{c}$, which is larger than the phase velocity β_{thr} of electromagnetic waves in the surrounding medium,

$$\beta > \beta_{thr} = \frac{1}{n} = \sqrt{1 - \frac{1}{\gamma_{thr}^2}}, \tag{2.1}$$

they polarize the molecules of that medium. After the particle has passed these molecules then return rapidly to their ground state and thereby emit radiation. The threshold above which particles produce Cherenkov radiation depends on the refractive index n of the gas in the radiator according to

$$\gamma_{thr} = \frac{1}{\sqrt{1 - \frac{1}{n^2}}}. \tag{2.2}$$

By choosing an appropriate radiator gas, one can achieve emission of Cherenkov radiation by e^+/e^- while suppressing that of hadrons which are heavier and therefore slower at a given momentum. The detector is then called hadron blind. Presently, HADES is used for experiments at SIS18 [3]. This synchrotron accelerates protons up to energies of $E = 4.5\ GeV$ and U^{73+} ions up to $E = 1\ AGeV$. The HADES heavy ion program is focused on incident kinetic energies from 1 to 2 $AGeV$ corresponding to $\beta \approx 0.9$. In this energy regime pions, which are the lightest and therefore fastest hadrons, gain momenta up to $p_{\pi^\pm} \approx 2\ GeV/c$.

The momentum of a relativistic particle is

$$p = m_0 \cdot \gamma \cdot v = m_0 \cdot c \cdot \gamma \cdot \beta \xrightarrow{\beta \to 1} m_0 \cdot c \cdot \gamma. \tag{2.3}$$

With the given pion momentum the necessary γ_{thr} can be calculated to be

$$\gamma_{thr} = \frac{p_{thr}}{m_0 \cdot c} = \frac{2GeV/c}{135MeV/c^2 \cdot c} = 15. \tag{2.4}$$

The cross section of the current RICH setup is shown in fig. 2.1. The radiator is filled with C_4F_{10} ($n = 1.0015$) [7]. Charged particles propagating with

$$\gamma > \gamma_{thr} = 18 \tag{2.5}$$

through the radiator produce Cherenkov radiation. The emitted photons form a cone towards the beam direction and are reflected by the VUV-mirror towards the photon detector.

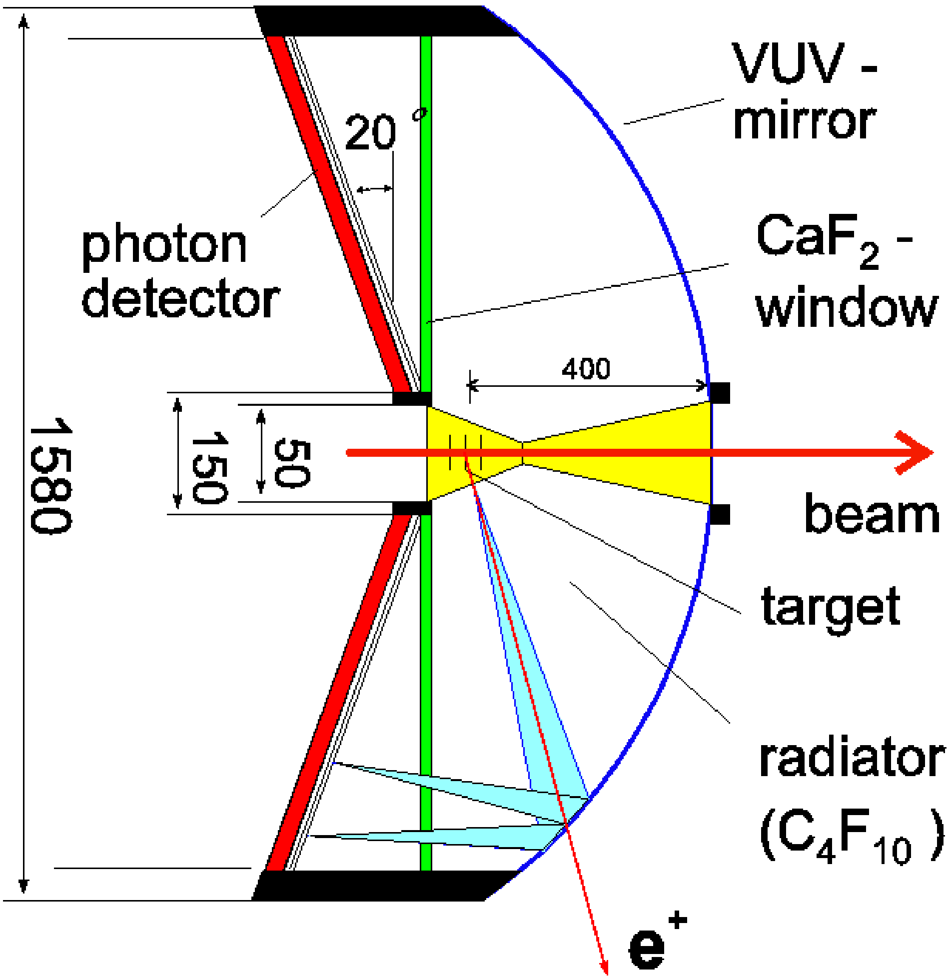


Fig. 2.1: Schematic picture of the RICH detector: In normal operation mode the radiator (green) is filled with C_4F_{10} and the photon detector (red) with methane (CH_4)

After crossing the CaF_2 window they hit the solid CsI photocathode evaporated as a thin layer onto the pad plane of the multi-wire proportional chamber. The released single photoelectrons are amplified in the multi-wire proportional chambers and the detector sees a Cherenkov ring. The detector is operated at ambient pressure with additional $15-40\ mbar$ overpressure. To protect the fragile CaF_2 window separating the two gases, the pressure difference between them has to be smaller than $\pm 4\ mbar$. The work function of the CsI photocathode is about $6\ eV$. Therefore, Cherenkov photons of a wavelength $\lambda > 210\ nm$ cannot be detected. The gases and the CaF_2 window have absorption edges at $140-150\ nm$. This limits the operating range to $150-210\ nm$ photons in the vacuum ultraviolet (VUV). In order to achieve a high efficiency of the RICH detector, the quality of all optical components must be kept close to optimum conditions. This includes especially a high degree of gas purification in the detector because the photons can easily be absorbed by H_2O or O_2 impurities. Consequently, the contamination has to be below 5ppm to guarantee the detection of at least 10 photons per Cherenkov ring [8]. Therefore, the gas in the detector and radiator volume (about 700 and 600 liters) has to be replaced once per hour. In order to monitor the gas purity and its optical quality, the transmission at the wavelengths of interest of a gas sample can be continuously measured using a VUV monochromator setup.

2.2 Gas system

For purification, circulation and pressure control of the gases in the RICH detector, the **P**urification **E**xtended **GA**s**SU**pervising **S**ystem (PEGASUS) was developed [8]. It consists of an open system for the photon detector and a closed recirculation system for the radiator. A schematic view of the existing system is shown in fig. 2.2. This setup was developed, because different standards had to be met. Methane of high purity is available at reasonable cost whereas C_4F_{10} is very expensive (> 500 €$/kg$). Furthermore, perfluorobutane is a fluorocarbon, i.e. a possible ozon killer and should therefore not be released into the atmosphere.

In periods between beam times the RICH detector is not connected to PEGASUS and operated in the so-called standby mode. N_2 from the GSI gas pipeline is used to flush the detector. A valve at the detector inlet is used to limit the nitrogen flow to about 150 l/h and the detector outlet is directly connected to the radiator inlet. At the radiator outlet a bubbler keeps the RICH detector protected against oxygen backdiffusion.

In the normal operation mode, the photon detector is filled with CH_4. The methane is taken from a bundle of twelve 50l bottles with a purity of 99.995%. After leaving the pressure reducer (72) the gas flows through the selector valve (73). If only gas of lower purity had been available, it would have been necessary to use purging cartridges (75, 76, 77). The gas flow can be monitored and controlled by a proportional magnetic valve unit (74). The pressure is regulated at the detector outlet by a regulating valve (80). A compressor pumps the gas to the exhaust (behind throttle 84) and generates a pressure for VUV transmission measurements (behind valve 83). For flushing the detector N_2 from a tank is used instead of CH_4.

The radiator is filled with C_4F_{10} in normal operation mode and can also be flushed with N_2 via valve 31 from the reservoir. This is done directly before and after beam times.

The gaseous C_4F_{10} is supplied (valve 30) from a liquid reservoir (26) in which the temperature is kept at $T \approx 30°\mathrm{C}$, i.e. well above the boiling point $T(1013mbar) = -1.7°\mathrm{C}$, in order to ensure a high vapor pressure even if a large amount of gas is needed. Valve 32 connects to the test volume for transmission measurements. The flow regulation valves and purification part is mounted in a temperature stabilized box and kept at about 30°C, to avoid condensation. The supply gas pressure is reduced at throttle 33 to $P = 2.5\ bar$. Purging cartridges can be switched into the circuit with valves 36, 37, 38. The flow is controlled with the electrical sensor 39 and the regulation valve with mechanical flow indicator (39a) before it is transported to the radiator. The gas pressure and its difference to that in the photon detector are monitored at the outlet by sensors (56, 57, 58) and regulated by valve 2. The membrane compressor (3) rebuilds the absolute pressure to a value between 3 and 4 bar. On its way back, the gas passes one port (35) to the transmission measurement volume and another one (valve 21) that allows filling from an external bottle. The gaseous C_4F_{10} is liquefied in a fridge operated at approximately $-18°\mathrm{C}$ and then pumped (22) back into the reservoir (26). Remaining gas accumulated in the fridge is blown off through valves 14, 15, 16. As the vapor pressure of C_4F_{10} is low at this temperature ($P_{C_4F_{10}}(-18°\mathrm{C}) \approx 500mbar$), the gas pressure in the fridge is stabilized via valve 43 by a permanent flow of pure N_2 from the tank to $p \approx 2.5\ bar$. In addition, valve 17 opens a small bypass ($50\ \frac{l}{h}$) to the compressor inlet which stabilizes compressor operation under extreme conditions and avoids C_4F_{10} condensation in the compressor. The nitrogen flow provides also continuous bubble washing of the liquid in the fridge and helps to remove H_2O contaminations.

After operation the C_4F_{10} must be recovered from the radiator (recovery mode). Therefore the radiator volume is flushed with N_2 from the reservoir pressing the C_4F_{10} back to the fridge for condensation and storage. These two gases differ strongly in their densities. Therefore the direction of the gas flow through the radiator can be reversed.

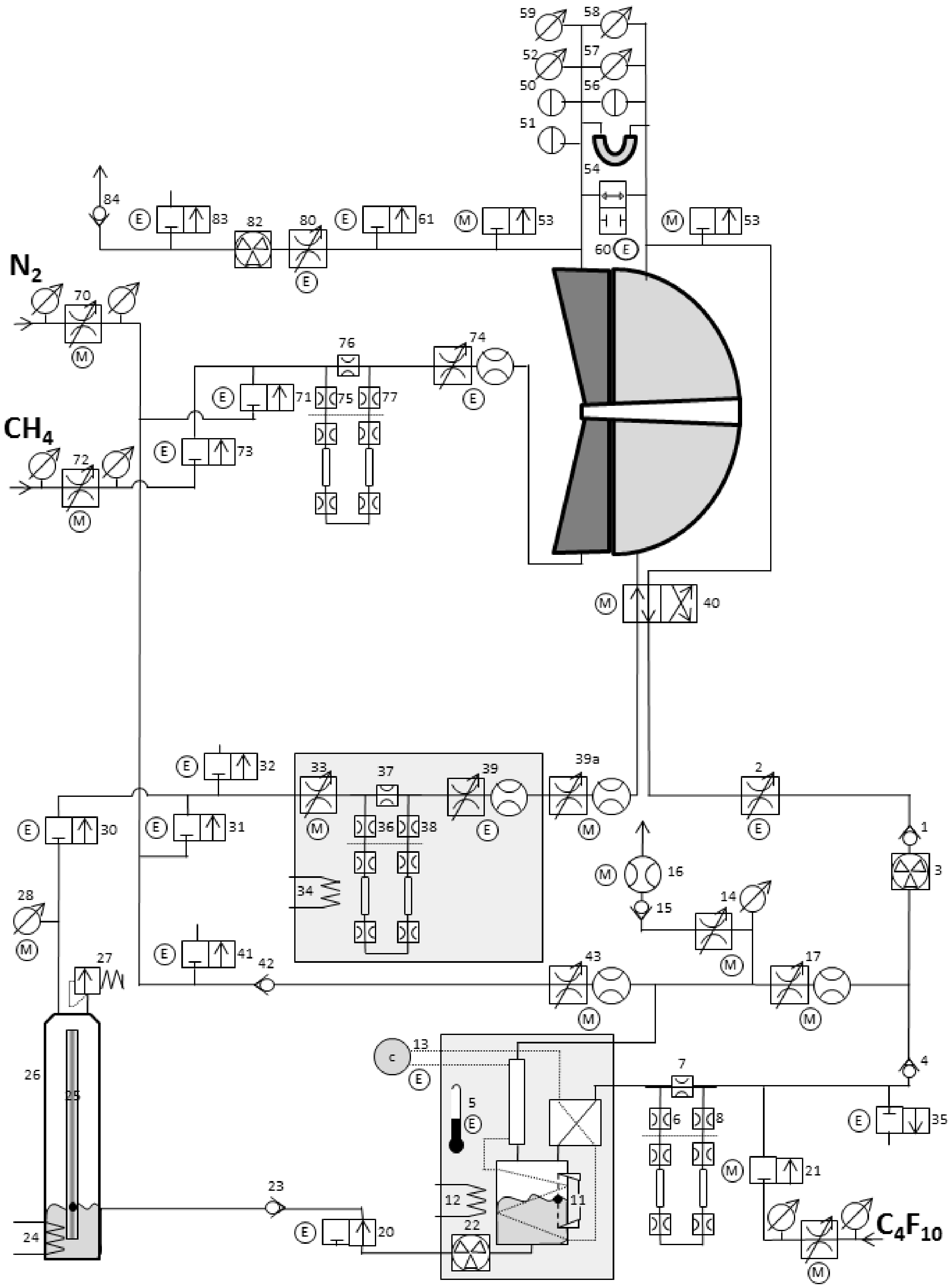


Fig. 2.2: Schematic layout of PEGASUS: one open system for the photon detector and a closed recirculation system for the radiator

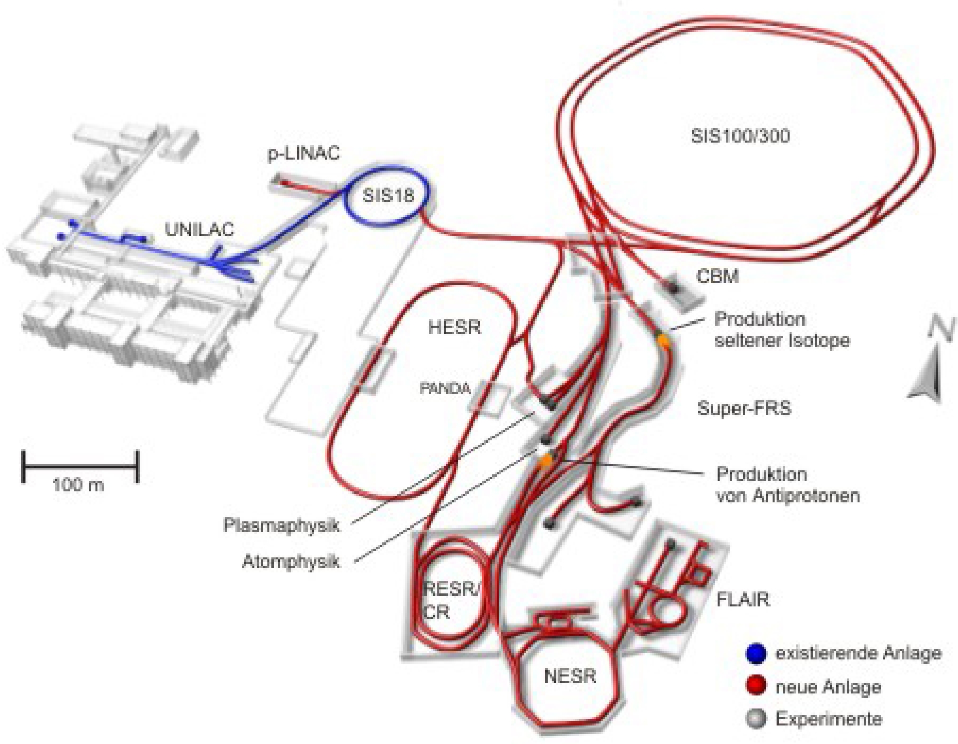


Fig. 2.3: The FAIR project at GSI. The existing facilities (blue) serve as injector for the planned synchrotrons SIS 100 and SIS 300.

2.3 System extensions for measurements at higher energies

For future research with HADES at higher energies the beams of SIS100 and SIS300 can be used. These synchrotrons are currently under construction at the GSI in Darmstadt for the FAIR-Project (fig. 2.3). SIS100 will be able to accelerate U^{28+} ions to 2.7 $AGeV$ and SIS300 U^{92+} to 34 $AGeV$ [7]. They could be used to measure e^+/e^- pairs from the decay of light vector mesons (ρ, ω, ϕ) and hadrons in collisions at $E = 2 - 15\ AGeV$. At these energies fireballs with temperatures $T \approx 120\ MeV$ are created. The momentum distribution of the emitted secondary particles can be described by the Maxwell-Boltzmann distribution. Therefore the number of secondary particles with large momenta increases with the projectile energy. Thus, light hadrons, especially pions, can exceed the given threshold of $\gamma_{thr} = 18$ and produce photons in the RICH detector, which would then no longer be hadron blind.

The expected momentum distribution of e^+/e^- and pions have been simulated in the diploma thesis of M. Weber [7] and are shown in fig. 2.4. One can see that for incident kinetic energies above about $E \approx 3\ AGeV$ a significant part of the pions exceeds the threshold of the currently used C_4F_{10}. Consequently, the threshold has to be modified by changing the radiator gas.

Usable gases and the resulting thresholds are shown in tab. 2.1. As the thresholds for electrons and positrons of all these gases are around $p = 10\ MeV/c$, almost all produced e^+/e^- would be detected. Thus, the choice of the right gas depends mainly on the maximum momentum of the produced pions.

Upgrading the RICH detector with a setup to control the refractive index and thereby the Cherenkov threshold in the radiator will be one of the main tasks of modifying HADES for higher energies. A comparatively simple way to realize this is to mix different gases. As can

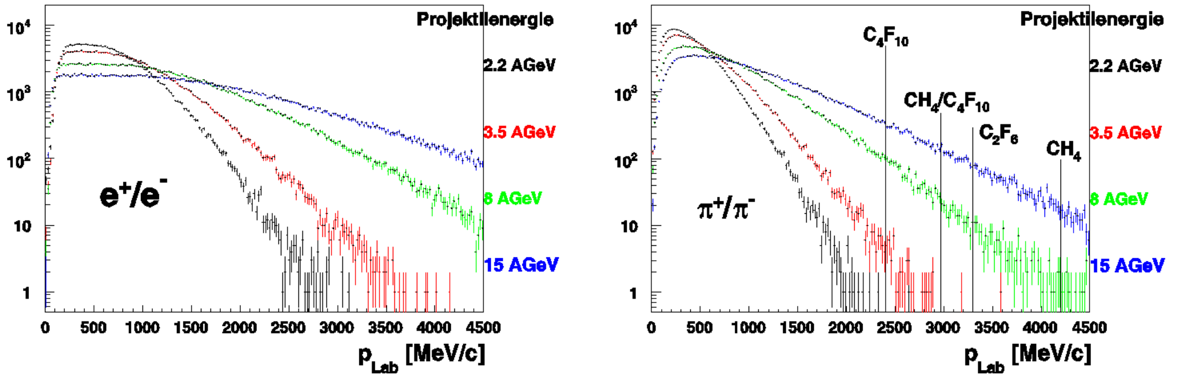


Fig. 2.4: Momentum distribution of electrons, positrons and charged pions from $\omega \rightarrow e^+ + e^-$ (left) and $\omega \rightarrow \pi^+ + \pi^- + \pi^0$ (right) with the Cherenkov thresholds of different gases for charged pions [7].

Radiator gas	Refractive index n	γ_{thr}	$p_{thr,\pi^+\pi^-}$ [GeV/c]	p_{thr,e^+e^-} [MeV/c]
C_4F_{10}	1.00155	18	2.4	9.2
$C_4F_{10}/CH_4(1:1)$	1.00103	22	2.97	11.2
C_2F_6	1.000793	25	3.3	12.7
CH_4	1.00051	31	4.2	15.8

Tab. 2.1: Cherenkov thresholds of different gases [7].

be seen from fig. 2.4 the thresholds can be adjusted by mixing C_4F_{10} and CH_4. Since both these gases are already used in the present system, only a mixing volume, one switching and two control valves need to be added. The proposal for a new scheme is depicted in fig. 2.5.

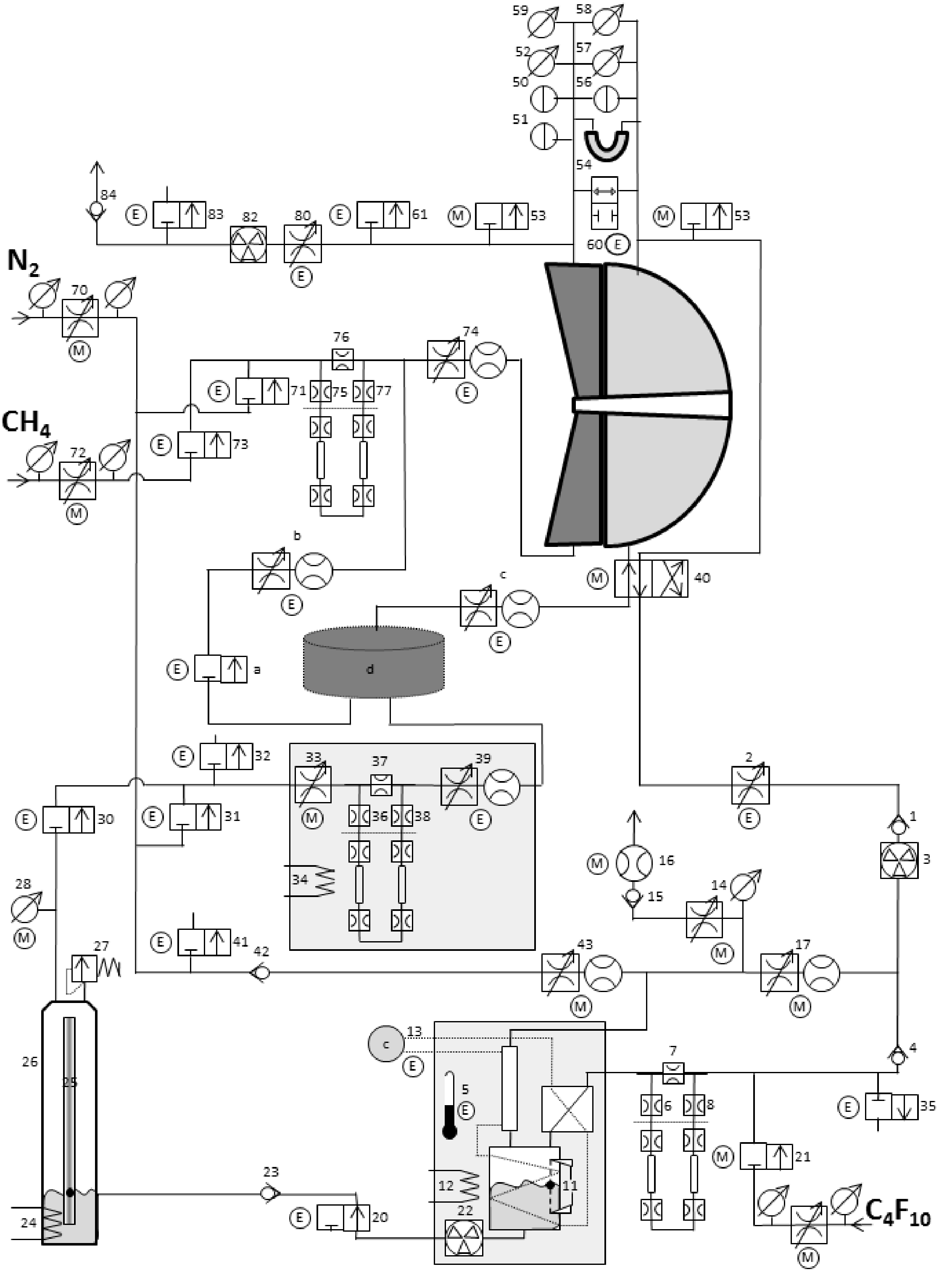


Fig. 2.5: Proposed upgrade setup of the RICH gas system for measurements with C_4F_{10}/CH_4 gas mixtures.

Chapter 3

The programmable logic controller

3.1 Present setup

The most important parts of PEGASUS (see Sec. 2.2) are the valves, sensors and compressors for pressure regulation, flow control and purification of the gases. There are altogether 33 devices to be monitored and regulated. The actuators are controlled by a programmable logic controller (PLC), which also reads out the sensors' signals.

The switching valves and sensors work with a binary $0/24\ V$ logic and so do the digital in- and output modules of the PLC. The magnet valves are open as long as the corresponding channels are high (which cooresponds to '1' in the binary logic of the PLC). As some actuators are run with $230\ V$ AC or even $400\ V$ three-phase electric power (e.g. the compressors), relays or contactors are used to switch their load current by the $24\ V$ output of the PLC. The temperature and pressure sensors display an analog $0 - 20\ mA$ signal, which is read out by a 12-bit analog-to-digital converter (ADC). The flowmeters are also connected to an ADC of the same resolution, but they display a $0 - 10\ V$ signal, which is what the analog actuators need and has therefore to be provided by the PLC as analog output.

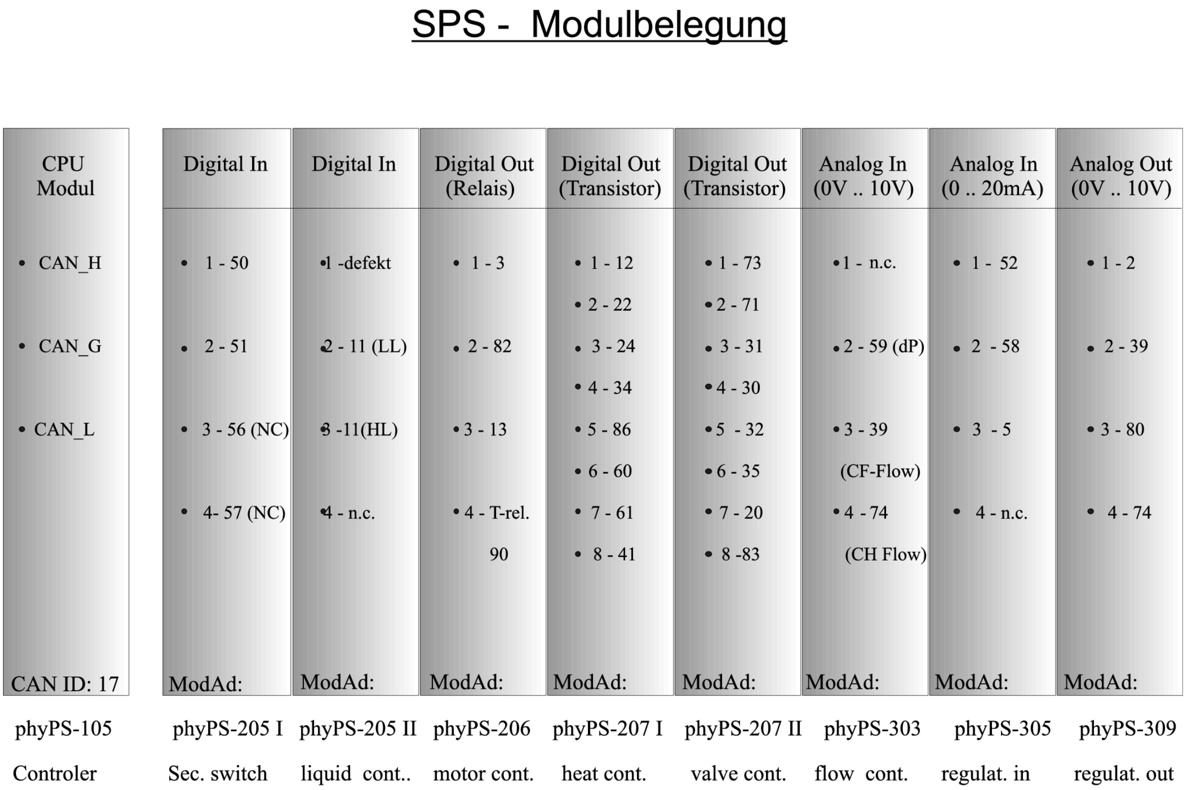
The presently installed PLC is a twelve year old PHYTEC[1] system [8]. It consists of a CPU (phyPS-105) that features RS-232 and CAN bus as well as eight in- and output modules with a total of 40 channels (tab. 3.1). All modules are plugged into one crate (fig. 3.1, fig. 3.2). If one module or further channels fail, the complete PLC has to be replaced, because spare parts are no longer available. As already mentioned in Sec. 2.3 it is planned to upgrade HADES for higher beam energies. Therefore it is necessary to add the possibility of gas mixing for the radiator gases of the RICH detector. Since no more modules are available and this new function must also be controlled, which necessitates more channels, the old PLC has to be replaced anyway.

Quantity	Name	Channels	Specification
2	phyPS-205	4	$0/24\ V$ In
1	phyPS-206	4	Relay
2	phyPS-207	8	$0/24\ V$ Out
1	phyPS-303	4	$0 - 10\ V$ In
1	phyPS-305	4	$0 - 20\ mA$ In
1	phyPS-309	4	$0 - 10\ V$ Out

Tab. 3.1: Technical data of the old in- and output modules [8].

---

[1] PHYTEC Messtechnik GmbH, Robert-Koch-Straße 39, 55129 Mainz



SPS - Modulbelegung

CPU Modul	Digital In	Digital In	Digital Out (Relais)	Digital Out (Transistor)	Digital Out (Transistor)	Analog In (0V .. 10V)	Analog In (0 .. 20mA)	Analog Out (0V .. 10V)
• CAN_H • CAN_G • CAN_L	• 1 - 50 • 2 - 51 • 3 - 56 (NC) • 4- 57 (NC)	• 1 -defekt • 2 - 11 (LL) • 3 -11(HL) • 4 - n.c.	• 1 - 3 • 2 - 82 • 3 - 13 • 4 - T-rel. 90	• 1 - 12 • 2 - 22 • 3 - 24 • 4 - 34 • 5 - 86 • 6 - 60 • 7 - 61 • 8 - 41	• 1 - 73 • 2 - 71 • 3 - 31 • 4 - 30 • 5 - 32 • 6 - 35 • 7 - 20 • 8 -83	• 1 - n.c. • 2 - 59 (dP) • 3 - 39 (CF-Flow) • 4 - 74 (CH Flow)	• 1 - 52 • 2 - 58 • 3 - 5 • 4 - n.c.	• 1 - 2 • 2 - 39 • 3 - 80 • 4 - 74
CAN ID: 17	ModAd:	ModAd:	ModAd:	ModAd:	ModAd:	ModAd:	ModAd:	ModAd:
phyPS-105	phyPS-205 I	phyPS-205 II	phyPS-206	phyPS-207 I	phyPS-207 II	phyPS-303	phyPS-305	phyPS-309
Controler	Sec. switch	liquid cont..	motor cont.	heat cont.	valve cont.	flow cont.	regulat. in	regulat. out

Fig. 3.1: Module and port configuration of the old PLC.

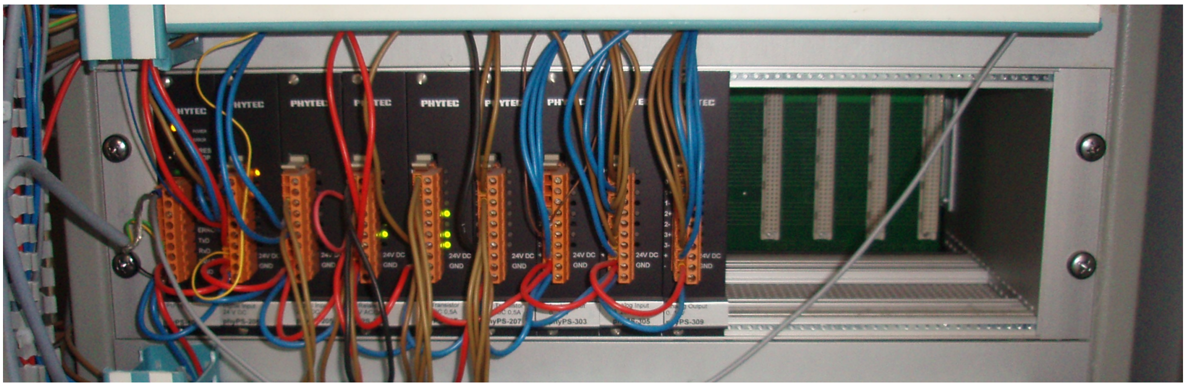


Fig. 3.2: Picture of the old PHYTEC PLC with the CPU module (left), five digital (middle) and three analog in- and output modules (right).

3.2 The new controller

To ensure future operation of the RICH gas system a new PLC shall meet several demands:

- **Compatibility:** The new PLC must be able to process the signals of the installed sensors and control the actuators (tab. 3.1).
- **Modularity:** Future extensions shall be implemented by adding modules with the necessary in- and output channels.
- **Simple programming language:** In order to enable future users to understand the then running program, a simple line based language is preferred for programming the PLC.
- **Support:** In case of system failures and hardware defects instant support from local experts at GSI is essential.

At GSI some similar systems based on the EtherCAT system manufactured by Beckhoff[2] are already installed and supported. Therefore a Beckhoff controller system utilizing an embedded PC with the required modules (fig. 3.3) was chosen.

The basic modules of the new system are CPU and power supply. In- and output modules can be attached easily. The CPU module (CX1010-0012) comprises an ethernet port, an internal flash memory, a CPU and a Compact Flash insert for memory extension. It is fixed permanently to the power supply module (CX1100-0004), which features an E-bus connection to attach in- and output modules. A separate Windows XP computer is connected to the PLC by ethernet for controlling and monitoring with Beckhoff's TwinCAT software. The software meets international standard IEC 61131-3, that deals with programming languages and defines five PLC programming language standards [9]. The power supply module is connected to an external 24 V source. An overview of the 13 installed in- and output modules with a total of 44 channels is given in tab. 3.2. The analog signals from the sensors are converted by 12-bit analog-to-digital converters (ADC) and displayed as 16-bit integers. The analog outputs also have a 12-bit resolution. All modules are mounted onto 35 mm DIN rail (fig. 3.4).

Quantity	Name	Channels	Specification
2	EL 1002	2	0/24 V In
1	ES 1002	2	0/24 V In
1	ES 1004	4	0/24 V In
2	EL 2042	2	0/24 V, 4 A Out
2	EL 2004	4	0/24 V, 0.5 A Out
1	EL 2008	8	0/24 V, 0.5 A Out
2	ES 3061	1	$0 - 10$ V In
1	ES 3048	8	$0 - 20$ mA In
1	EL 4004	4	$0 - 10$ V Out

Tab. 3.2: Technical data of the new Beckhoff in- and output modules. For digital output modules the maximum current per channel is given.

---

[2] Beckhoff Automation GmbH, Eiserstraße 5, 33415 Verl

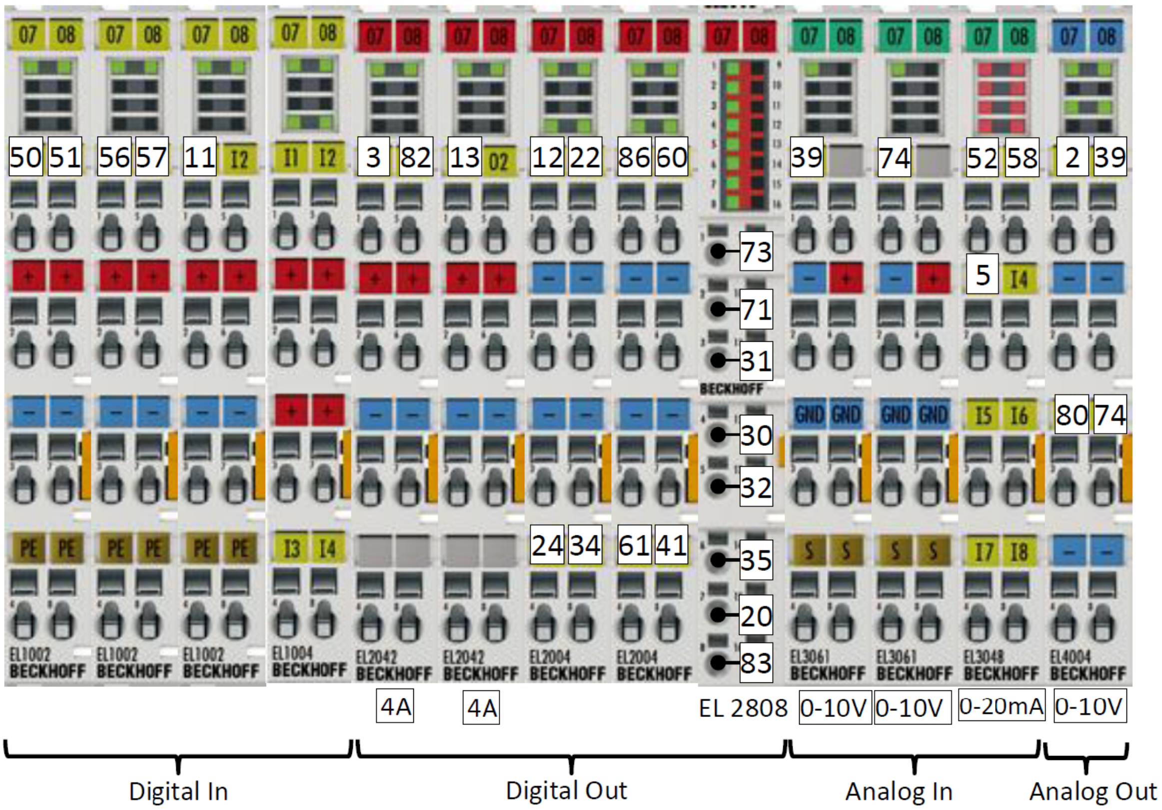


Fig. 3.3: Module and port configuration of the new PLC

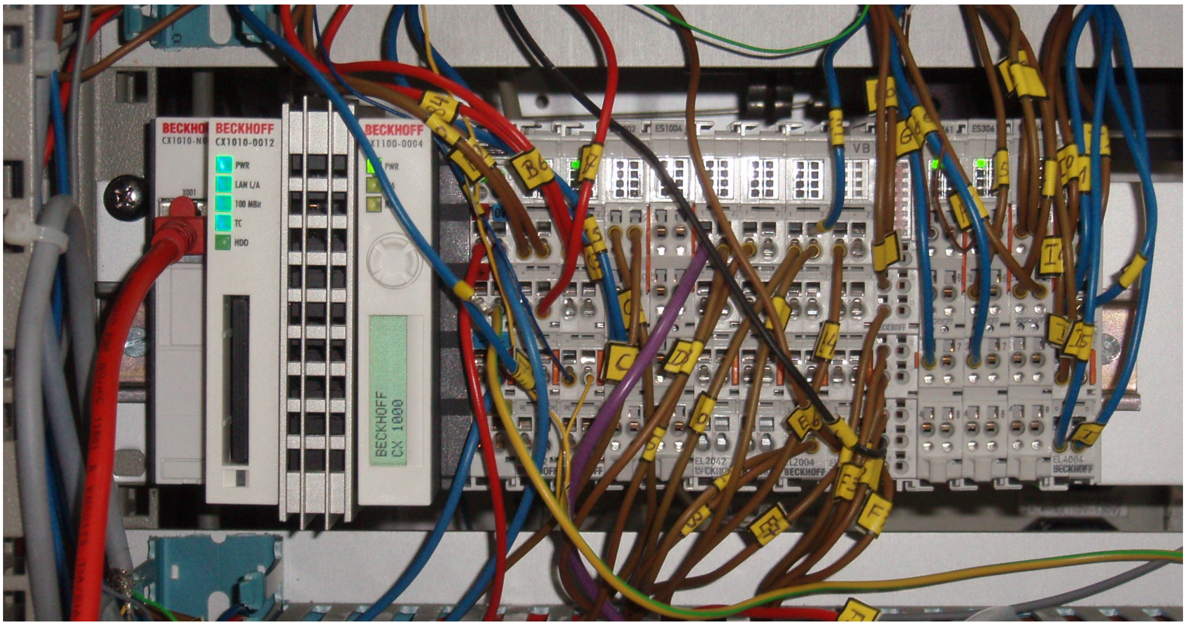


Fig. 3.4: The new Beckhoff PLC. The CX with ethernet port on the left and different analog and digital in- and output modules.

3.3 Source code

When the gas system was developed, a line based source code was written [8]. TwinCAT supports the IEC 61131-3 standard, which includes structured text (ST), a high level language that is block structured and syntactically resembles Pascal. As some parts of the old code are not needed any longer and the syntax of ST differs from the one used in the old program [9], the source code was slightly changed. Its essential parts are now briefly commented on, the complete version (more than 300 lines) can be found in the appendix (A.1).

- **Setting default values:** During a system start the variables (for a detailed overview of all variables see A.2) which are given by the user in normal operation mode are set to their default values. This affects operation modes, desired pressures and flows.
- **Checking set values:** Whenever a variable is set by the user, the program checks the sensor signals whether the new value or mode endangers the detector. The set value is then either changed automatically or the gas flow is stopped.
- **Comparing desired and measured values:** The sensor values of temperature, flows and pressures are compared to the desired values in every program loop ($t = 10\ ms$). Therefore, the program converts the ADC values with calibration constants.
- **Checking differential pressure:** A pressure difference between the radiator and the photon detector endangers the window in between. Therefore, the differential pressure is constantly controlled by the program with several sensors. In case of danger, emergency valves are opened and the gas flow is stopped.
- **Switching valves:** The modes of operation differ mainly in the switching position of valves. Whenever the mode is changed, the corresponding valves are switched. The most important operation modes are:
  1. Shutdown: no gas flow, static state
  2. N_2/N_2: flushing both chambers with nitrogen
  3. CH_4/C_4F_{10}: normal operation mode
  4. Recovery: getting back C_4F_{10} and filling the radiator with N_2
- **Regulation:** The program regulates the flows and pressures so that they stay close to the desired values. For pressure regulation two function blocks are used. They can be found in the appendix (A.3).
- **Error control:** If a pressure value is outside of its normal range, an error is reported. As this might be a singular malfunction, the system is not always immediately shut down. Instead of that, error reports are managed in a way that the system is only shut down in case of a systematic error. Furthermore, the error handling allows to find the critical value in case of an emergency shut down.

3.4 Tests of the new PLC

After implementation of the new software the TwinCAT package was used to test the PLC operation in three steps.

1. The source code was tested on a virtual PLC. The program was uploaded into a virtual system emulated by the TwinCAT software PLC on the host computer. In this way the syntax could be successfully checked and it was assured that the source code runs error-free.

2. The real PLC was tested with all modules. The in- and output modules were connected to a test board with LEDs, switches and potentiometers which simulated most of the sensor signals and actuator states expected for the real gas system. These tests demonstrated the correct operation of the program including the error handling for out of range values of temperature and pressure sensors.

3. Finally the new PLC was connected to the RICH gas system. After testing the sensors and actuators in order to assure their correct operation, the wiring had to be modified because the old and new power supplies differ slightly.

   During the following tests the Beckhoff PLC controlled all components of the gas system smoothly. The overpressure in the RICH detector was varied and monitored between $0 - 30$ $mbar$ at different flow rates. All control loops worked fine and the pressure difference between the radiator and the photon detector was kept close to zero during the complete testing period. The gas flow through both chambers could be controlled and monitored as desired. The C_4F_{10} modes have not yet been tested, since a full cycle (filling, operation and recovery) should only be done in the course of a real experiment during beam time. A few minor problems were observed which have to be solved but it was demonstrated successfully that the new PLC is able to read out the sensors' signals and control the actuators.

   In order to ensure the operational availability of the system for the upcoming important experiment, the new PLC was removed again. The old PHYTEC system was installed and tested in order to ensure a smooth operation during the next beam time.

Chapter 4

Summary and Outlook

This thesis is concerned with a new gas system controller that has been assembled and programmed. It has been successfully demonstrated that the gas system of the HADES RICH detector can be controlled and monitored by a Beckhoff PLC system running on a CPU with Windows CE operating system and the TwinCAT software package. The final test at the GSI in Darmstadt showed that only slight modifications of the existing system were necessary before switching to the new PLC. The new system is ready to replace the old PHYTEC controller.

However, there are still some tasks to be carried out.

1. The calibration of the pressure sensors and the linearity of the analog input module ES3048 have to be checked. This will lead to new calibration parameters in the code.
2. The large current load of the ten magnet valves (20, 30, 31, 32, 35, 41, 61, 71, 73, 83) may damage the digital output modules EL2004 and EL2808. They should be exchanged by EL2042 modules, which offer higher output currents.
3. The new PLC has still to be tested with the different C_4F_{10} operation modes (normal operation, refilling, recovery).
4. The existing detailed PEGASUS operation manual has to be updated as soon as the new PLC is finally installed.
5. The user interface of the TwinCAT project has to be adapted for easy and user friendly monitoring of the gas system status.
6. The TwinCAT software should be installed on a virtual machine (VMware) with Windows XP or Windows 7 operating system. The virtual machine can then be played on any other computer irrespective of its operating system. The control system is then platform independent and can be safely operated for the coming 10-15 years.

As the Beckhoff PLC is the only backup for the current PHYTEC modules, it is desirable to get it to an operational state as soon as possible. The exchange of the system is planned for fall 2011, shortly after the forthcoming beam time.

As soon as the new PLC is in operation, one can start to realize the planned modifications of the RICH detector for higher beam energies. For controlling the additional components, only one digital output channel, one in- and one output module each with at least two analog $0-10\ V$ channels for the additional flowmeters and valves are needed. The additional modules can easily be added to the system due to the modularity inherent in the Beckhoff EtherCAT bus system. In the control software the additional program lines have to be added to the existing source code and uploaded to the CX. The old function blocks and control loops will help to find an efficient way to mix CH_4 and C_4F_{10} gases to the desired precision. Once these remaining tasks have been completed, the gas supply system of the HADES RICH detector is prepared for a safe and reliable operation in the decade to come.

Appendix A

Program on the new PLC

A.1 Source code

```
dcount := dcount  + 1;

IF bReset THEN
        bOnMain := 0 ;
        iModePhoton := 1 ;
        iModeRadiator := 1 ;
        bReset := 0 ;
        bMonitorV32 := 0 ;
        bMonitorV35 := 0 ;
        bMonitorV83 := 0 ;
        bMonitorV41 := 0 ;
        bMonitorV86 := 0 ;
END_IF;

IF dPabsSet > 1040000 THEN
        dPabsSet := 1040000 ;
END_IF;
IF dPabsSet < 980000 THEN
        dPabsSet := 980000 ;
END_IF;
IF iFlowPhotonSet > 20000 THEN
        iFlowPhotonSet := 20000 ;
END_IF;

(*delayed shut-off  and start of the compressors:
bFinished 1s delayed on (not bOnMain)
TimeStartRegulation 10s delayed on (bOnMain)
TimeStartCompress 1s delayed on (TimeStartRegulation)*)
TimeFinish(IN := bOnMain , PT := t#1s) ;
bFinished := NOT TimeFinish.q ;
TimeStartRegulation(IN := bOnMain  , PT := t#10s) ;

IF bFinished THEN
        bCompress3 := 0 ;
```

```
        bCompress82 := 0 ;
ELSE
        TimeStartCompress(IN := TimeStartRegulation.q , PT := t#1s) ;
        IF TimeStartCompress.q THEN
                bCompress3 := 1 ;
                bCompress82 := 1 ;
        END_IF ;
END_IF ;

(*pressure regulation*)
dPabsMeas52 := (1000000/22880) * UINT_TO_DINT (iPabsSens52) ;
dPrelInterl := ( (25 * ( ( ( UINT_TO_DINT (iDpSensI57) ) / 8 ) - 2450 ) ) / 4 ) ;
dPrelMeas :=  - 2 * ( ( ( UINT_TO_DINT ( iPrelSens58 ) ) / 8 ) - 2800 ) ;
dPabsRadiator := dPabsMeas52 + dPrelMeas ;

IF dPabsSet > dPabsRadiator + 2000 THEN
        dPabsSoll := dPabsRadiator + 2000;
ELSIF dPabsSet < dPabsRadiator - 2000 THEN
        dPabsSoll := dPabsRadiator - 2000;
ELSE
        dPabsSoll := dPabsSet ;
END_IF

IF TimeStartRegulation.Q THEN
        IF dRamp > 0 THEN
                dRamp := dRamp - 400;
        END_IF
        IF DPabsMeas52 > dPabsSoll THEN
                dPabsRegDiff := dPabsRegDiff + 1;
        ELSIF DPabsMeas52 < dPabsSoll THEN
                dPabsRegDiff := dPabsRegDiff - 1;
        END_IF;
        IF dPabsRegDiff > 3000 THEN
                dPabsRegDiff :=  3000;
        ELSIF dPabsRegDiff < -3000 THEN
                dPabsRegDiff := -3000;
        END_IF;
        IPabsReg := ( 8* ( regulknick ( SOLL := dPabsSoll , IST := dPabsMeas52 , DEV := 2000 ) ) ) ;
        dPabsReg := UINT_TO_DINT ( IPabsReg ) ;
        IPabsReg80 := DINT2UINT ( IPabsReg + dPabsRegDiff + 600 - dRamp ) ;

        IF dPrelMeas > 100 THEN
                dPrelRegDiff := dPrelRegDiff + 1;
        ELSIF dPrelMeas < -100 THEN
                dPrelRegDiff := dPrelRegDiff - 1;
```

```
        END_IF
        IF dPrelRegDiff > 3000 THEN
                dPrelRegDiff := 3000 ;
        ELSIF dPrelRegDiff < -6000 THEN
                dPrelRegDiff := -6000;
        END_IF
        IPrelReg := ( 6* ( regulator ( SOLL := 0 , IST := dPrelMeas , DEV := 1000 ) ) ) ;
        dPrelReg := UINT_TO_DINT ( IPrelReg );
        IPrelReg2 := DINT2UINT (dPrelReg + dPrelRegDiff + 5600 - dRamp ) ;

        IF iFlowPhotonSens74 > iFlowPhotonSet THEN
                iValveSoll74 := iValveSoll74 - 10;
        ELSIF iFlowPhotonSens74 < iFlowPhotonSet THEN
                iValveSoll74 := iValveSoll74 + 10;
        END_IF
        iFlowPhotonSoll74 := iValveSoll74 - DINT2UINT ( dRamp);
        dFlowRadiatorSens39 := UINT_TO_DINT(iFlowRadiatorSens39);
        iFlowRadiatorSoll39 := iFlowRadiatorSet - DINT2UINT(dRamp) ;
ELSE
        dRamp := 8000;
        iValveSoll74 := 20000;
        iFlowPhotonSoll74 := 0;
        iPabsReg80 := 0;
        iPrelReg2 := 0;
END_IF

(* states of the photon detector *)
IF iModePhoton = 2 THEN
        bNPhotonV71 := 1;
ELSE
        bNPhotonV71 := 0;
END_IF

IF iModePhoton = 3 THEN
        bCHPhotonV73 := 1;
ELSE
        bCHPhotonV73 := 0;
END_IF

(* states of the radiator *)
CASE iModeRadiator OF
        1 (* no gas *) :
                bCFRadiatorV30 := 0;
                bNRadiatorV31 := 0;
                bHEAT12 := 0;
                bFridge13 := 0;
        2 (* N2 *) :
```

```
            bCFRadiatorV30 := 0;
            bNRadiatorV31 := 1;
            bHEAT12 := 1;
            bFridge13 := 0;
    3 (* C4F10 (detector filled with N2) *) :
            IF bFridgeCold THEN
                iModeRadiator := 4;
            ELSE
                bCFRadiatorV30 := 0;
                bNRadiatorV31 := 1;
            END_IF
            bHEAT12 := 0;
            bFridge13 := 1;
    4 (* C4F10 *) :
            IF bFridgeCold THEN
                bCFRadiatorV30 := 1;
                bNRadiatorV31 := 0;
            ELSE
                bCFRadiatorV30 := 0;
                bNRadiatorV31 := 0;
                bOnMain := 0;
                IF iError = 0 THEN
                    iError := 5;
                    dError := UINT_TO_DINT ( iTempFridgeSens5 );
                END_IF
            END_IF
            bHEAT12 := 0;
            bFridge13 := 1;
    5 (* get back C4F10 *) :
            IF bFridgeCold THEN
                bCFRadiatorV30 := 0;
                bNRadiatorV31 := 1;
            ELSE
                bCFRadiatorV30 := 0;
                bNRadiatorV31 := 0;
                bOnMain := 0;
                IF iError = 0 THEN
                    iError := 5;
                    dError := UINT_TO_DINT ( iTempFridgeSens5 );
                END_IF
            END_IF
            bHEAT12 := 0;
            bFridge13 := 1;
    6 (* C4F10 (already C4F10 in the detector, but fridge not cold; only for an emergency!!! *) :
            IF bFridgeCold THEN
                iModeRadiator := 4;
            ELSE
```

```
            bCFRadiatorV30 := 1;
            bNRadiatorV31 := 0;
        END_IF
        bHEAT12 := 0;
        bFridge13 := 1;
    9 (* filling storage *) :
        bCFRadiatorV30 := 0;
        bNRadiatorV31 := 0;
        bHEAT12 := 0;
        bFridge13 := 1;
        bOnMain := 0;
END_CASE

(* fridge cold? *)
IF ( iTempFridgeSens5 < 11500 ) OR ( ( iTempFridgeSens5 < 15000 ) AND bFridgeCold ) THEN
    bFridgeCold := 1 ;
ELSE
    bFridgeCold := 0 ;
END_IF

(* heater and pump *)
IF (bFinished AND NOT ( iModeRadiator = 9 ) ) THEN
    bHeat24 := 0 ;
    bHeat34 := 0 ;
    bPump22 := 0 ;
    bLiquidV20 := 0 ;
ELSE
    bHeat24 := 1 ;
    bHeat34 := 1 ;
    TimePumpOn ( IN := bLiquidSw11 , PT := t#45s );
    IF ( TimePumpOn.Q OR bPumpManual ) THEN
        bPump22 := 1 ;
        bLiquidV20 := 1 ;
    ELSE
        bPump22 := 0 ;
        bLiquidV20 := 0 ;
    END_IF
END_IF

(* VUV-monitor *)
IF (bTransRemote) THEN
    IF ( bTransRem1 AND bTransRem2 ) THEN
        iModeTrans := 2 ;
    END_IF
    IF ( bTransRem1 AND NOT bTransRem2 ) THEN
        iModeTrans := 4 ;
    END_IF
    IF ( NOT bTransRem1 AND bTransRem2 ) THEN
        iModeTrans := 1 ;
```

```
        END_IF
        IF ( NOT bTransRem1 AND NOT bTransRem2 ) THEN
                iModeTrans := 3 ;
        END_IF
END_IF

IF NOT ( iModeRadiator = 2 OR iModeRadiator = 4 ) OR NOT ( iModePhoton = 2 OR iModePhoton = 3 ) OR bOnMain = 0 THEN
        IF ( iModeTrans = 1 ) OR ( iModeTrans = 2 ) OR ( iModeTrans = 4 ) THEN
                iModeTrans := 0 ;
        END_IF
END_IF

IF iModeTrans = 1 THEN
        bMonitorV32 := 1;
ELSE
        bMonitorV32 := 0;
END_IF
IF iModeTrans = 2 THEN
        bMonitorV35 := 1;
ELSE
        bMonitorV35 := 0;
END_IF
IF iModeTrans = 3 THEN
        bMonitorV41 := 1;
ELSE
        bMonitorV41 := 0;
END_IF
IF iModeTrans = 4 THEN
        bMonitorV83 := 1;
ELSE
        bMonitorV83 := 0;
END_IF
IF iModeTrans > 0 THEN
        bMonitorV86 := 1;
ELSE
        bMonitorV86 := 0;
END_IF

(* emergency switch off *)
IF NOT bDisabESO THEN
        IF ( ( dPabsMeas52 > 1045000) OR ( dPabsMeas52 < 970000 ) ) THEN
                dErrCount := dErrCount + 10 ;
                IF iError = 0 THEN
                        iError :=52 ;
                        dError := dPabsMeas52 ;
                END_IF
        END_IF
```

```
        IF ( ABS (dPrelMeas) > 2300 ) THEN
                dErrCount := dErrCount + 10 ;
                IF iError = 0 THEN
                        iError :=58 ;
                        dError := dPrelMeas ;
                END_IF
        END_IF
        IF ( ABS (dPrelInterl) > 4000 ) THEN
                dErrCount := dErrCount + 2 ;
                IF iError = 0 THEN
                        iError :=57 ;
                        dError := dPrelInterl ;
                END_IF
        END_IF
        IF ( dErrCount > 0 ) THEN
                dErrCount := dErrCount - 1 ;
        END_IF
        IF ( dErrCount > 10 ) THEN
                bReset := 1 ;
        END_IF
        IF ( ( ABS(dPabsMeas52 - dPabsSet) > 4000 ) AND bOnMain) THEN
                TimeEmergOff ( In := TRUE, PT := t#100s ) ;
        ELSE
                TimeEmergOff ( In := FALSE, PT := t#100s ) ;
        END_IF
        IF TimeEmergOff.q THEN
                bReset := 1 ;
                IF iError = 0 THEN
                        iError :=522 ;
                        dError := dPabsMeas52 ;
                END_IF
        END_IF
END_IF

(* emergency hradware valves 50 and 51 *)
IF NOT bPabsSw50 THEN
        bReset := 1 ;
        IF iError = 0 THEN
                iError := 50 ;
        END_IF
END_IF
IF NOT bPabsSw51 THEN
        bReset := 1 ;
        IF iError = 0 THEN
                iError := 51 ;
        END_IF
END_IF
```

A.2 Variables in the source code

The following list in alphabetical order gives an overview of the variables in the source code. The different types are:

- **INT**: 16 bit integer (values −32768 to 32767)
- **UINT**: 16 bit unsigned integer (values 0 to 65535)
- **DINT**: 32 bit integer
- **BOOL**: Boolean variable
- **TON** and **TOF**: Timing variable

Name	Channel	Typ	Description
bCFRadiatorV30	Digital Out	BOOL	Valve 30
bCHPhotonV73	Digital Out	BOOL	Valve 73
bCompress3	Digital Out	BOOL	Compressor 3
bCompress82	Digital Out	BOOL	Compressor 82
bDisabESO	-	BOOL	Disable emergency switch off
bFinished	-	BOOL	Ready to switch off compressors
bFridge13	Digital Out	BOOL	Fridge 13
bFridgeCold	-	BOOL	Fridge cold
bHeat12 AT	Digital Out	BOOL	Heater 12
bHeat24 AT	Digital Out	BOOL	Heater 24
bHeat34 AT	Digital Out	BOOL	Heater 34
bLiquidSw11	Digital In	BOOL	Liquid switch 11
bLiquidV20	Digital Out	BOOL	Valve 20
bMonitorV32	Digital Out	BOOL	Valve 32
bMonitorV35	Digital Out	BOOL	Valve 35
bMonitorV41	Digital Out	BOOL	Valve 41
bMonitorV83	Digital Out	BOOL	Valve 83
bMonitorV86	Digital Out	BOOL	Valve 86
bNPhotonV71	Digital Out	BOOL	Valve 71
bNRadiatorV31	Digital Out	BOOL	Valve 31
bOnMain	-	BOOL	Main switch: system on/off
bPabsSw50	Digital In	BOOL	Sensor 50: 50mbar overpressure
bPabsSw51	Digital In	BOOL	Sensor 51: 8mbar underpressure
bPump22 AT	Digital Out	BOOL	Pump 22
bPumpManual	-	BOOL	Manual pumping of the liquid
bReset	-	BOOL	Switches off gas system
bTransRem1 AT	Digital In	BOOL	Remote transmission control I
bTransRem2 AT	Digital In	BOOL	Remote transmission control II
bTransRemote	-	BOOL	Remote transmission control on / off
dcount	-	DINT	Number of program runs
dErrCount	-	DINT	Counts dangerous pressure values
dError	-	DINT	Value which caused the error
dFlowRadiatorSens39	-	DINT	Sensor 39
dPabsMeas52	-	DINT	Absolute pressure in the detector [dPa]

Name	Channel	Typ	Description
dPabsRadiator	-	DINT	Absolute pressure in the radiator
dPabsReg	-	DINT	Strengh of absolute pressure regulation
dPabsRegDiff	-	DINT	Deviation of absolute pressure
dPabsSet	-	DINT	Desired absolute pressure [dPa]
dPabsSoll	-	DINT	Desired absolute pressure [dPa] (limited)
dPrelInterl	-	DINT	Pressure difference
dPrelMeas	-	DINT	Relative pressure [dPa]
dPrelReg	-	DINT	Strengh of relative pressure regulation
dPrelRegDiff	-	DINT	Counts discrepancys of relative pressure
dRamp	-	DINT	Slows starting regulation
iDpSensI57	Analog In	UINT	Sensor 57
iError	-	INT	Errorcode for emergency reset
iFlowPhotonSens74	Analog In	UINT	Sensor 74
iFlowPhotonSet	-	UINT	Desired value for flow into the detector
iFlowPhotonSoll74	Analog Out	UINT	Valve 74
iFlowRadiatorSens39	Analog In	UINT	Sensor 39
iFlowRadiatorSet	-	UINT	Regulation value flow for radiator
iFlowRadiatorSoll39	Analog Out	UINT	Valve 39
iModePhoton	-	INT	Mode of photondetector
iModeRadiator	-	INT	Mode of radiator
iModeTrans	-	INT	Mode of transmission measurement
IPabsReg	-	UINT	Strength of absolute pressure regulation
IPabsReg80	Analog Out	UINT	Valve 80
iPabsSens52	Analog In	UINT	Sensor 52
IPrelReg	-	UINT	Strengh of relative pressure regulation
IPrelReg2	Analog Out	UINT	Valve 2
iPrelSens58	Analog In	UINT	Sensor 58
iTempFridgeSens5	Analog In	UINT	Sensor 5
iValveSoll74	-	UINT	Value for flow regulation into the detector
TimeEmergOff	-	TON	Delay for Emergency shut down
TimeFinish	-	TOF	Delay for switching off compressors
TimePumpOn	-	TOF	Delay for switching off pump 22
TimeStartCompress	-	TON	Delay before start of the compressors
TimeStartRegulation	-	TON	Delay before start regulation on

A.3 Function Blocks

A.3.1 regulknick

```
FUNCTION regulknick : UINT
VAR_INPUT
        SOLL: DINT;
        IST: DINT;
        DEV: DINT;
END_VAR
VAR
END_VAR
```

```
IF ( (ABS ( IST - SOLL ) ) < DEV) THEN
        regulknick := DINT_TO_UINT( 2048 + ( ( ( ( ( 2048 * (IST -
           SOLL ) ) / DEV ) * ( IST - SOLL ) ) / DEV ) * ( IST -
           SOLL ) ) / DEV) ;
        ELSE
        IF IST < SOLL THEN
                regulknick := 0 ;
                ELSE
                regulknick := 4095 ;
        END_IF
END_IF
```

A.3.2 regulator

```
FUNCTION regulator : UINT
VAR_INPUT
        SOLL: DINT;
        IST: DINT;
        DEV: DINT;
END_VAR
VAR
END_VAR
```

```
IF ( ( ABS ( IST - SOLL ) ) < DEV) THEN
        regulator := DINT_TO_UINT( 2048 + ( ( 2048 * (IST - SOLL )
           ) / DEV ) ) ;
        ELSE
        IF IST < SOLL THEN
                regulator := 0 ;
                ELSE
                regulator := 4095 ;
        END_IF
END_IF
```

Appendix B

How to operate the PLC

This chapter explains how a Beckhoff PLC can be started, controlled and switched off. The host computer has to be connected to the PLC either directly or with a network switch.

Start:

1. Start the TwinCAT software:

   Right-click on the TwinCAT symbol in the notification area → System → Start

2. Start PLC Control:

   Right-click on the TwinCAT symbol in the notification area → PLC Control

   If necessary: Open the project file [.pro] (Datei → Öffnen...; Shortcut: Control+O)

3. Select destination system:

   Online → Auswahl des Zielsystems...

   Select by double click:

   Laufzeitsystem 1 (Port 801)

   directly under

   CX_061D2E (5.6.29.46.1.1)

   (Not the Upper one!)

4. Log in:

   Online → Einloggen (Shortcut: F11)

   If you are asked: "Das Programm wurde geändert! Soll das neue Programm geladen werden?" Click: Ja

5. Run the PLC

   Online → Start (Shortcut: F5)

The Program is now running on the PLC.

Set variables:

1. Go to the window "Globale_Variablen" within the TwinCAT PLC control window.
2. Double click on the variable you want to set. A window will pop up.

3. Enter the desired value in the lower line and click OK. The window will disapear, but the variable is not yet set. The desired value is diplayed turquoise next to the present value. You can correct it by double clicking on the variable again.
4. In order to finally set it:

   Online → Werte schreiben (Shortcut: Control + F7)

<u>Switch off:</u>

1. Stop the PLC:

   Online → Stop (Shortcut: Shift + F8)
2. Log out:

   Online → Ausloggen (Shortcut: F12)

You may now switch off the PLC power supply and turn off the computer.

Bibliography

[1] S. Leupold, V. Metag and U. Mosel, “Hadrons in strongly interacting matter.” Int. J. Mod. Phys. E 19, 147 (2010).

[2] G. Agakishiev et al., “Dielectron production in Ar+KCl collisions at 1.76A GeV.” Phys. Rev. C 84, 014902 (2011).

[3] G. Agakishiev et al., “The High-Acceptance Dielectron Spectrometer HADES,” *Eur. Phys. J.*, vol. A41, pp. 243–277, 2009.

[4] K. Zeitelhack et al., “The HADES RICH Detector.” Nucl. Instr. Meth. A433 (1999) 201.

[5] G. Agakishiev et al., “Study of dielectron production in C+C collisions at 1 AGeV,” *Phys. Lett.*, vol. B663, pp. 43–48, 2008.

[6] G. Agakishiev et al., “Dielectron production in C-12 + C-12 collisions at 2-AGeV with HADES,” *Phys. Rev. Lett.*, vol. 98, p. 052302, 2007.

[7] M. Weber, “Simulationen zum e^+e^- Nachweis mit dem HADES RICH für Projektilenergien bis 15 AGeV,” Diploma thesis, Technische Universität München, October 2007.

[8] J. Friese and R. Gernhäuser, *Purification Extended GAs SUpervising System for the HADES-RICH detector*, September 2010. Manual and User’s Guide.

[9] “Einführung in IEC-1131-3 Programmierung,” November 1998. Beckhoff TwinCAT.

Danksagung

Zunächst möchte ich mich bei Herrn Prof. Dr. Reiner Krücken bedanken, dass er es mir ermöglicht hat, diese Arbeit zu schreiben.

Mein Dank gilt weiterhin Jürgen Friese, der mich nicht nur physikalisch weiterbildete, sondern auch physisch ein Ansporn war (und es noch ist).

Ebenfalls bedanken möchte ich mich bei Ludwig Meier und den übrigen E 12 Mitarbeitern, die mir wertvolle Tipps für diese Arbeit gegeben haben.

Besonderer Dank gilt natürlichen meiner Familie und meinen Freunden, die mich besonders in den letzten Wochen zusätzlich unterstützt haben.

Schließlich möchte ich mich noch ganz herzlich bei den Leuten bedanken, die dafür gesorgt haben, dass ich überhaupt so weit gekommen bin: Thurid Mannel, Ann-Kathrin Straub, Maximilian Totzauer und Jonas Probst. Ich bin sehr froh, dass ich mit euch studieren konnte.