SYNCHRONIZATION OF A MAGNETICALLY-COUPLED PISTON PUMP FOR THE RADON DISTILLATION COLUMN AT XENONnT

BACHELOR THESIS

Philipp Schulte

Westfälische Wilhelms-Universität Münster
Institut für Kernphysik
AG Weinheimer

Referent: Prof. Dr. C. Weinheimer
Koreferent: Prof. Dr. A. Andronic

August 21, 2019

Declaration of Academic Integrity

I hereby confirm that the present thesis “Synchronization of a magnetically-coupled piston pump for the radon distillation column at XENONnT” is the result of my own independent scholarly work, and that in all cases material from the work of others (in books, articles, essays, dissertations, and on the internet) is acknowledged, and quotations and paraphrases are clearly indicated. No material other than that listed has been used. I have read and understood the Institute’s regulations and procedures concerning plagiarism.

Münster, August 21, 2019 ______________________________

Philipp Schulte

Contents

1 Introduction

About 95 % of the energy density of the Universe is mostly unknown. Science research tries to find out more about this energy density with the help of astroparticle physics phenomena. Among others the phenomena include the existence of dark matter as an explanation for rotation curves of galaxies. One promising particle candidate for this dark matter is the Weakly Interacting Massive Particle, a particle beyond the standard-model.

Searching for such *invisible* particles a lot of scientists have joined forces shaping our knowledge of the Universe with new findings. One budding collaboration of scientists is the XENON Dark Matter Project using a dual-phase time projection chamber at "Laboratori Nazionali del Gran Sasso" to directly search for the interaction of Weakly Interactive Massive Particles with standard-model particles [1].

In order to detect these predicted, invisible particles, the collision with the noble gas xenon is used. Xenon has many advantages for this venture, for example its heavy nucleus and its high atomic number [2]. These properties result in a high self-shielding character as well as a high interaction probability with the WIMP. Furthermore xenon has the advantages of a high density, low chemical reactivity and few radioactive isotopes [2]. Radioactive contaminations in xenon like krypton and radon produce a contributing background and thus lower the sensitivity of the detector. Therefore a careful material selection of built-in components as well as removal systems purifying the xenon are of crucial importance. One method to reduce this intrinsic contaminations in xenon is cryogenic distillation [3]. The previous stage of the experiment, XENON1T, thereby reaches the lowest ever achieved krypton in xenon concentration for a dual-phase time projection chamber [3]. Thus, the background of the next generation detector, XENONnT, is dominated by radon [4]. Therefore, a new radon distillation column is developed for XENONnT. One component of this distillation column is a radon free high flux compressor. Considering that the compressor has to be radon-free a custom-made magnetically-coupled piston pump is used instead of a commercially available pump. A prototype of this piston pump was successfully developed and then used at XENON1T [5]. Due to the capability of in-house development the full control of all built-in components as well as the possibility for modifications resulting in a higher performance is given. This modified compressor for the radon removal system is required to feature a high and stable throughput and steady pressure built-up. In contrast to the prototype pump a one cylinder double-stroke pump can not be used by virtue of the cyclic movement. Rather a multi cylinder pump can smooth the throughput and ensure a higher flux. In this case a four cylinder magnetically-coupled piston pump will be used for the experiment. In order to reach the desired steady and high xenon flow and compression the four cylinders have to be operated in parallel with a shifted movement of the pistons.

To ensure the steady performance over long term, the shifted piston movement of the four cylinders have to be synchronized. This thesis aims to program a drive profile to each cylinder drive, synchronize the movement with a shift between all cylinders and develop an interface for a status readout of all drives.

At first, an introduction about dark matter and its detection techniques is given. Afterwards, the XENON Dark Matter Project will be presented in chapter 3. Subsequently the pressure curves of a single and a four cylinder magnetically-coupled piston pump will be discussed to motivate the needed shift. After that the implementation of the synchronization will be explained followed by the analysis of the first tests. Next a longtime operation of the synchronized movement ensuring the stability of the synchronization will be shown. Lastly a conclusion as well as an outlook will sum up the results and give an overview for future tasks.

2 Dark matter

The current state of research about the composition of the Universe predicts that the baryonic, visible matter is not the only existing form of matter. Only 4.9 % of the whole energy density of the Universe consists of this normal matter. The other part is made of 68.3 % so-called dark energy and of 26.8 % dark matter [6]. Dark energy is largely unknown today, while dark matter is postulated as an invisible form of matter interacting by gravitational forces and weak interaction on a small scale. The postulation is based on three main evidences: the cosmic microwave background (CMB) [7], gravitational lensing [8], as well as the rotation curves of spiral galaxies [9], where the last one is the most demonstrative evidence. Therefore, the rotation curves as an evidence for dark matter will be shown in detail in the following.

2.1 Evidence for dark matter: rotation curves

Rotation curves of spiral galaxies can be observed by measuring the velocity of stars in spiral galaxies. In order to measure the velocity, the Doppler effect, in particular the 21 cm hyperfine transition line of neutral hydrogen, is used. The observed speed of stars in a spiral galaxy as a function of the distance to the center of the galaxy differs from the theoretical consideration. With reference to Newton's laws the stars' movement can be described with

$$v \propto \sqrt{\frac{m(r)}{r}}, \tag{1}$$

where the velocity falls with increasing radius. Assuming that the central halo density ρ_0 of the galaxy is constant [10], its mass has a dependence on the radius r given by

$$m(r) = \rho_0 \frac{4}{3} \pi r^3. \tag{2}$$

In this case the velocity behavior is similar to a rigid body rotation

$$(v \propto r). \tag{3}$$

This expected linear increase until the mass density falls down can be seen in figure 1. This theoretical dependency of equation 3 could be observed. However, for large radii from the center of the galaxy the observation shown in figure 1 indicates that the velocity of stars stays constant in contrast to equation 1.

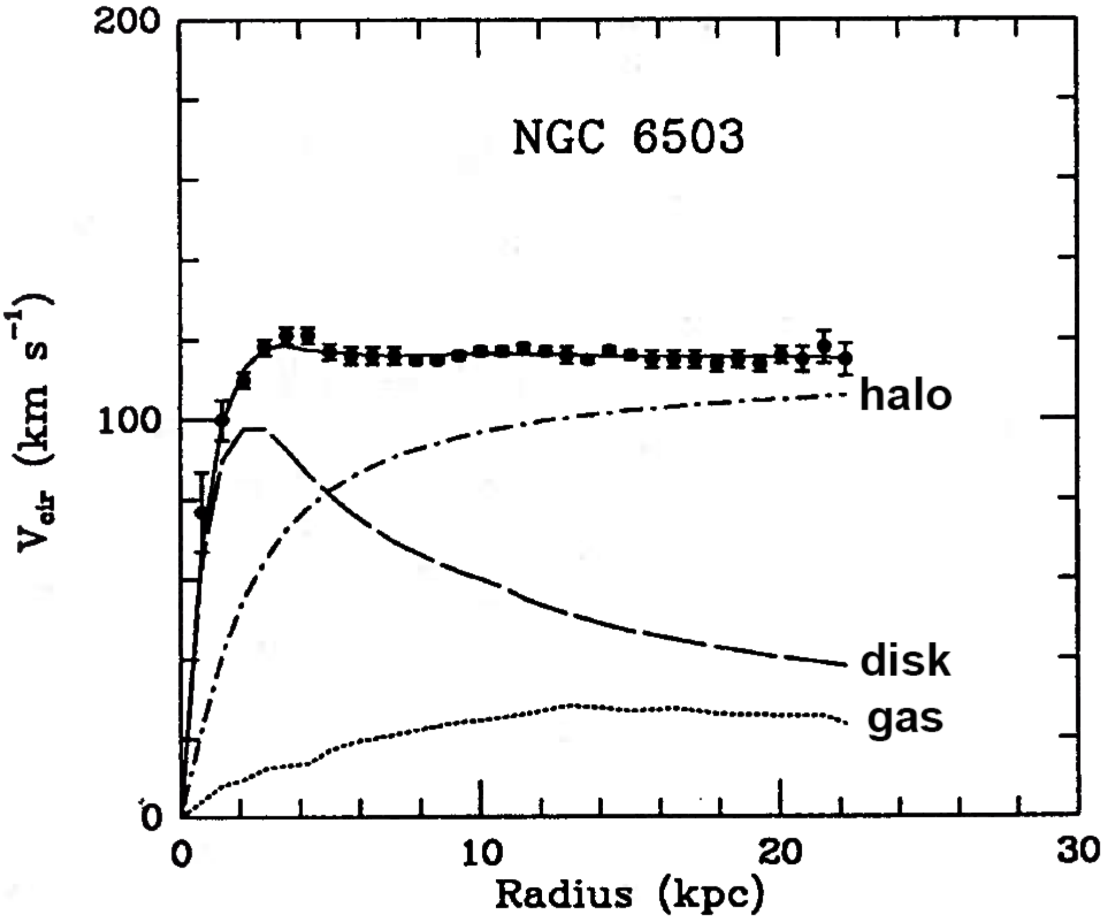


Figure 1: Measurement of the velocity from stars in the galaxy NGC 6503 as a function of the distance from the center [10]. Additionally, two theoretical curves are shown. One curve shows the velocity of the disk and the other of the gas. In addition a halo is modeled, which can explain the difference to the observed data.

An addition of a halo can explain the observed velocities at a large distance from the center of the galaxy. The postulation of a dark matter halo added to the ordinary matter compensates the discrepancy. As a result of this observation many theories about new hypothetical particles beyond the standard model were developed. One of these particles is the **W**eakly **I**nteracting **M**assive **P**article (WIMP). It is supposed to be massive, neutral and interacting on a scale similar to the weak interaction. The WIMP is a particle beyond the standard model [11]. Due to the fact that the particle interacts weakly, ambitious detection techniques have to be developed.

2.2 Dark matter detection techniques

The detection of WIMPs requires a coupling to baryonic matter. Figure 2 shows three strategies to detect the dark matter particles.

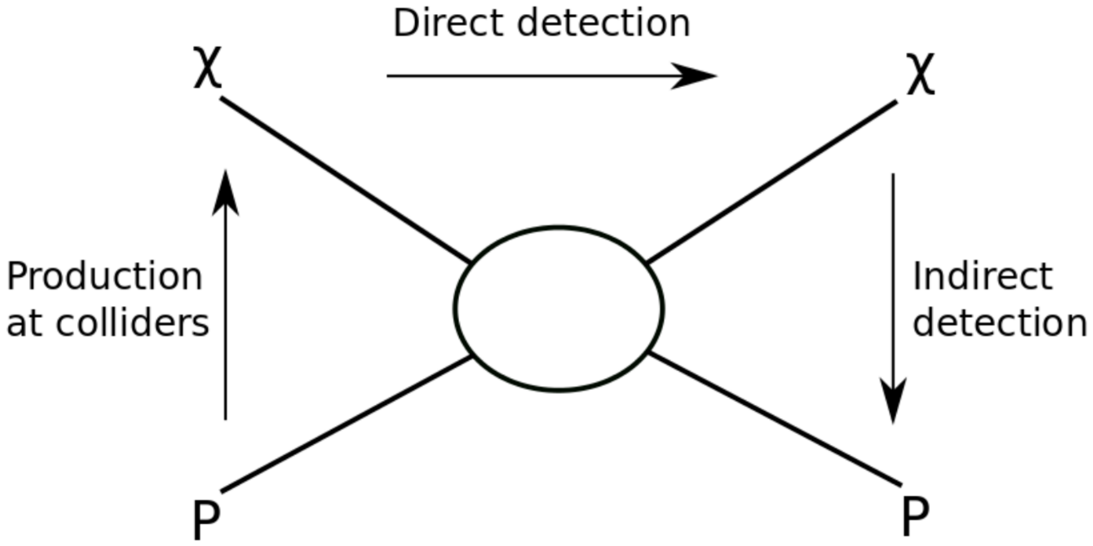


Figure 2: Schematic showing the possible dark matter detection channels [12].

On one hand WIMPs can be produced at colliders. For instance at the LHC at CERN, protons are accelerated and brought to a collision. This process is expected to pair-produce two WIMPs as depicted in the Feynman-diagram from bottom to top. The detection is based on missing transverse energy [13]. On the other hand WIMPs can be hypothetically detected by indirect detection. For example the IceCube experiment at the south pole could observe WIMP annihilations to neutrinos [14]. The third technique is the direct detection searching for a scattering of WIMP with nuclei shown from left to right in figure 2. Such collisions deposit energy in the detector material, which can be observed by low background detectors. This way of detection is used by the XENON Dark Matter Project.

3 The XENON Dark Matter Project

The XENON Dark Matter Project is a collaborative experiment of about 160 scientists from 27 institutions around the world and is installed in a mountain at "Laboratori Nazionali del Gran Sasso" (LNGS) in Italy [1]. To shield the detector from cosmic radiation, the experiment is built into a mountain under 1400 m of rock, which leads to a reduction by six orders of magnitude [15]. The low background detector is a **T**ime **P**rojection **C**hamber (TPC) filled with xenon because of its suitable characteristics.

3.1 Xenon as a detector medium

As a target medium, xenon benefits from its heavy nucleus ($Z = 54$). Xenon has a low chemical reactivity and belongs to the noble gases. Natural xenon is composed of isotopes with atomic numbers of $A = 124$ to $A = 136$. A high A of the target material is advantageous, since the probability of an interaction between a WIMP and a nucleus is proportional to A^2 [2]. Moreover, liquid xenon has a high density ($\sim 3\,\frac{\mathrm{g}}{\mathrm{cm}^2}$), which leads to a high stopping power that enables a self-shielding to protect the inner detector volume from background [2].

A critical background of such dark matter detectors is intrinsic radioactive contamination. Radioactive decays within the detector cause background that can not be shielded against. With regard to that, xenon is advantageous, considering that it consists of only a few radioactive isotopes. Even though commercially available xenon contains two unstable isotopes, that is 8.87 % ^{136}Xe and 0.09 % ^{124}Xe [2]. These isotopes constitute no disturbing background because of their rare decays. The radioactive isotope ^{136}Xe can decay via double β-decay with a half-life of $T_{1/2} = (2.165 \pm 0.016_{stat} \pm 0.059_{sys}) \times 10^{21}$ yr [16]. Secondly, the ^{124}Xe decays via double electron-capture with a half-life of $T_{1/2} = (1.8 \pm 0.5_{stat} \pm 0.1_{sys}) \times 10^{22}$ yr [17]. Furthermore, commercially available xenon contains other radioactive isotopes. Low concentrations of isotopes of krypton and radon also play a role for the background. These background sources will be discussed after an introduction to the construction and functionality of the TPC.

3.2 The time projection chamber

The XENON Dark Matter Project uses a dual-phase TPC filled with liquid xenon and a small gaseous phase above as depicted in figure 3.

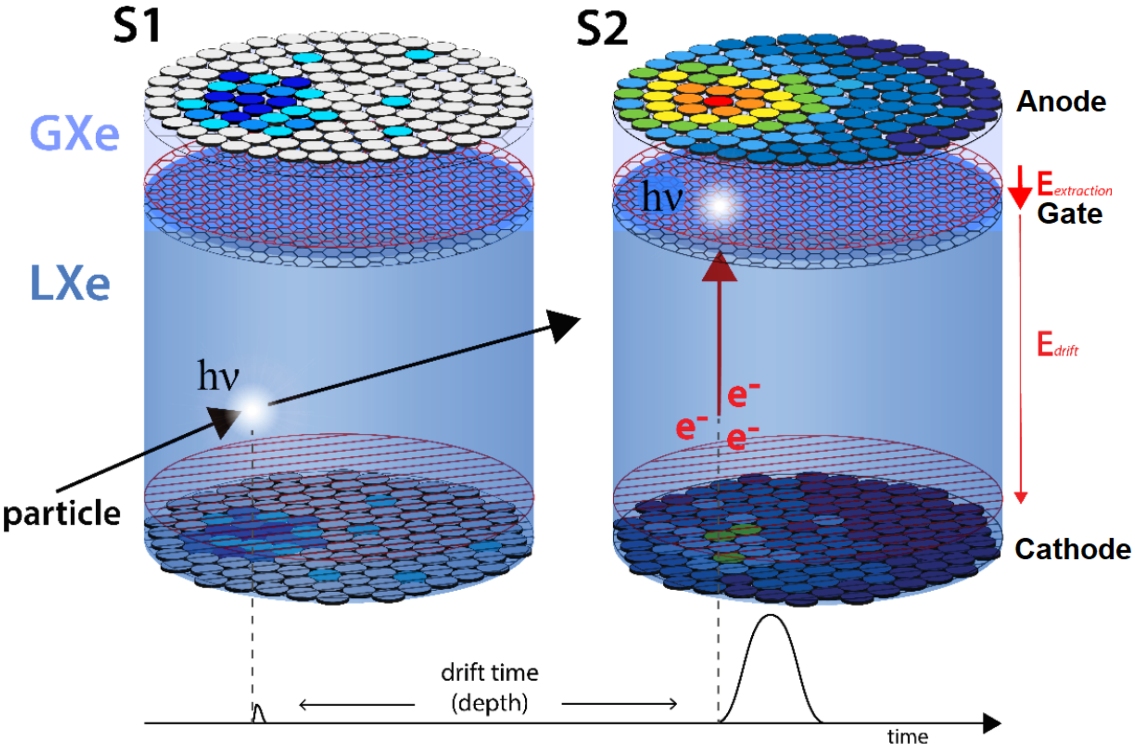


Figure 3: The principle of the TPC and the representation of the two signals $S1$ and $S2$.[18]

The detector is equipped with arrays of photomultipliertubes (PMTs) on both end caps. Furthermore, an electric field is generated by a cathode mesh on the bottom and a grounded gate in the middle. Moreover, an extraction field is generated by the grounded gate and an anode mesh on the top. If a particle scatters in the liquid target with a xenon nucleus, energy is deposited by a nuclear recoil. As a consequence of the recoil other xenon atoms will be excited and ionized as shown in the following [2]:

$$\begin{aligned} \chi + Xe &\rightarrow \chi + Xe^* \\ Xe^* + Xe + Xe &\rightarrow Xe_2^* + Xe \\ Xe_2^* &\rightarrow 2Xe + h\nu \end{aligned} \tag{4}$$

$$\begin{aligned} \chi + Xe &\rightarrow Xe^{+} + e^{-} + \chi \\ Xe^{+} + Xe &\rightarrow Xe_2^{+} \\ Xe_2^{+} + e^{-} &\rightarrow Xe^{**} + Xe \\ Xe^{**} &\rightarrow Xe^{*} + heat \\ Xe^{*} + Xe + Xe &\rightarrow Xe_2^{*} + Xe \\ Xe_2^{*} &\rightarrow 2Xe + h\nu \end{aligned} \tag{5}$$

An excited atom in the presence of other xenon atoms can lead to the formation of an excited dimer. Subsequently, the unstable dimer dissipates with emission of a photon (equation 4). The resulting 178 nm scintillation light generates a signal called $S1$ at the PMTs. Moreover, an ionized xenon nucleus can either recombine with the electron, which creates prompt 178 nm scintillation photon (equation 5) or the electrons can drift through the electric field to the top of the detector. Here, the electrons are extracted as a consequence of the stronger electric field between gate mesh and anode mesh. Thereby the electrons collide with xenon atoms and deposit their energy by electrolummesence. This generates a second signal called $S2$.

The time difference between these two signals, the $S1$ and $S2$, offers the possibility to reconstruct the scattering event three-dimensionally. In order to get a good reconstruction of each event, absorption of the scintillation light as well as electron capturing by impurities has to be avoided. Therefore, a purification system is required, which will be discussed in the next section.

3.3 Purification system

As described in the paragraphs above the target material contains two different types of impurities. On one hand the radioactive contaminations by krypton and radon and on the other hand electronegative impurities. Both of these have to be removed.

Electronegative impurities for example O_2 and H_2O attract and capture electrons drifting to the top of the vessel [5]. This leads to a reduced detection of the $S2$. In addition, the electronegative impurities absorb light in the wavelength range of xenon [5]. So these impurities must be separated from the xenon. This is done by a purification system as installed at the XENON1T experiment.

The detector material is continuously cleaned by circulating the gasous phase through hot metal getters, where electronegative substances are removed. The concentration of the electronegatives has to be reduced to a level such that the electron livetime τ_{e} within the detector is long enough for the electrons to reach the gaseous phase. Therefore, a system is required, which reduces these impurities to a lower concentration than 1 ppb[1] in xenon, such that the electrons can drift over lengths of 1 m and greater in the detector [5].

The other aspect of purification is related to radioactive contaminations. The two main contributing isotopes are ^{85}Kr and ^{222}Rn. ^{85}Kr is intrinsically abundant in air due to atmospheric nuclear tests and fuel reprocessing [19]. Commercially available xenon is gained from air and thus contains intrinsic contaminations of ^{85}Kr. Its decays are shown in figure 4. Inside the xenon ^{85}Kr β-decays into the ground state of ^{85}Rb can look like WIMP signals due to its Q value of 687 keV, which occur in 99 % of all decays.

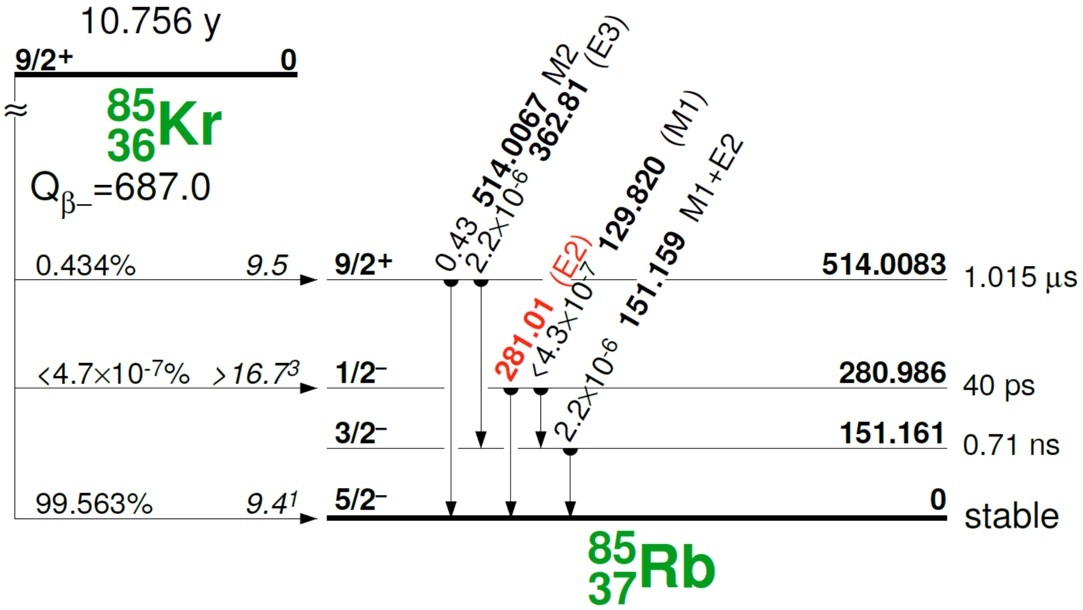
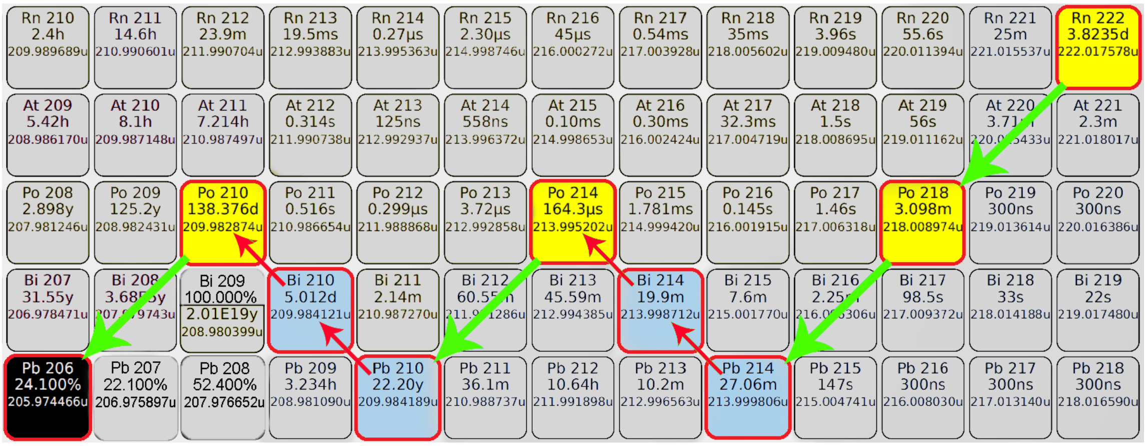


Figure 4: Decay scheme of ^{85}Kr (adapted from [20]).

Therefore, a distillation column was developed in Münster [3]. At XENON1T the cryogenic distillation of krypton contributes to the lowest ever achieved background in a dark matter experiment [3].

However the remaining background of the experiment is dominated by radon, especially ^{222}Rn [4]. The ^{222}Rn intrinsically is not a problem, because its α-decay has an energy in average of MeV instead of keV, which is expected for a recoil produced by a WIMP. The decay of ^{222}Rn is an α-decay with a half-live of $t_{\frac{1}{2}} = 3.8\,\mathrm{d}$ [21]. Therefore the radon is homogeneously distributed in the detector since it continuously emanates from detector components. The decay chain of ^{222}Rn is illustrated in figure 5.

[1] ppb ≙ parts per billion



Rn 210 2.4h 209.989689u	Rn 211 14.6h 210.990601u	Rn 212 23.9m 211.990704u	Rn 213 19.5ms 212.993883u	Rn 214 0.27µs 213.995363u	Rn 215 2.30µs 214.998746u	Rn 216 45µs 216.000272u	Rn 217 0.54ms 217.003928u	Rn 218 35ms 218.005602u	Rn 219 3.96s 219.009480u	Rn 220 55.6s 220.011394u	Rn 221 25m 221.015537u	Rn 222 3.8235d 222.017578u
At 209 5.42h 208.986170u	At 210 8.1h 209.987148u	At 211 7.214h 210.987497u	At 212 0.314s 211.990738u	At 213 125ns 212.992937u	At 214 558ns 213.996372u	At 215 0.10ms 214.998653u	At 216 0.30ms 216.002424u	At 217 32.3ms 217.004719u	At 218 1.5s 218.008695u	At 219 56s 219.011162u	At 220 3.71[illegible] 220.0[illegible]5433u	At 221 2.3m 221.018017u
Po 208 2.898y 207.981246u	Po 209 125.2y 208.982431u	Po 210 138.376d 209.982874u	Po 211 0.516s 210.986654u	Po 212 0.299µs 211.988868u	Po 213 3.72µs 212.992858u	Po 214 164.3µs 213.995202u	Po 215 1.781ms 214.999420u	Po 216 0.145s 216.001915u	Po 217 1.46s 217.006318u	Po 218 3.098m 218.008974u	Po 219 300ns 219.013614u	Po 220 300ns 220.016386u
Bi 207 31.55y 206.978471u	Bi 208 3.68[illegible]y 207.[illegible]9743u	Bi 209 100.000% 2.01E19y 208.980399u	Bi 210 5.012d 209.984121u	Bi 211 2.14m 210.987270u	Bi 212 60.55[illegible] 211.9[illegible]1286u	Bi 213 45.59m 212.994385u	Bi 214 19.9m 213.998712u	Bi 215 7.6m 215.001770u	Bi 216 2.25[illegible] 216.0[illegible]306u	Bi 217 98.5s 217.009372u	Bi 218 33s 218.014188u	Bi 219 22s 219.017480u
Pb 206 24.100% 205.974466u	Pb 207 22.100% 206.975897u	Pb 208 52.400% 207.976652u	Pb 209 3.234h 208.981090u	Pb 210 22.20y 209.984189u	Pb 211 36.1m 210.988737u	Pb 212 10.64h 211.991898u	Pb 213 10.2m 212.996563u	Pb 214 27.06m 213.999806u	Pb 215 147s 215.004741u	Pb 216 300ns 216.008030u	Pb 217 300ns 217.013140u	Pb 218 300ns 218.016590u

Figure 5: A section from the nuclide card, where the decay chain from ^{222}Rn is shown. The red arrows starting from blue nuclides represent β^--decays and the green ones, starting at yellow fields, are the α-decays. The black is a stable nucleus. [22]

Considering that the radon decay chain includes the β-emitter ^{214}Pb, the ^{222}Rn is an important background for the experiment. The ^{214}Pb decays with a probability of 9.3 % into the ground state of the daughter isotope as seen in figure 6.

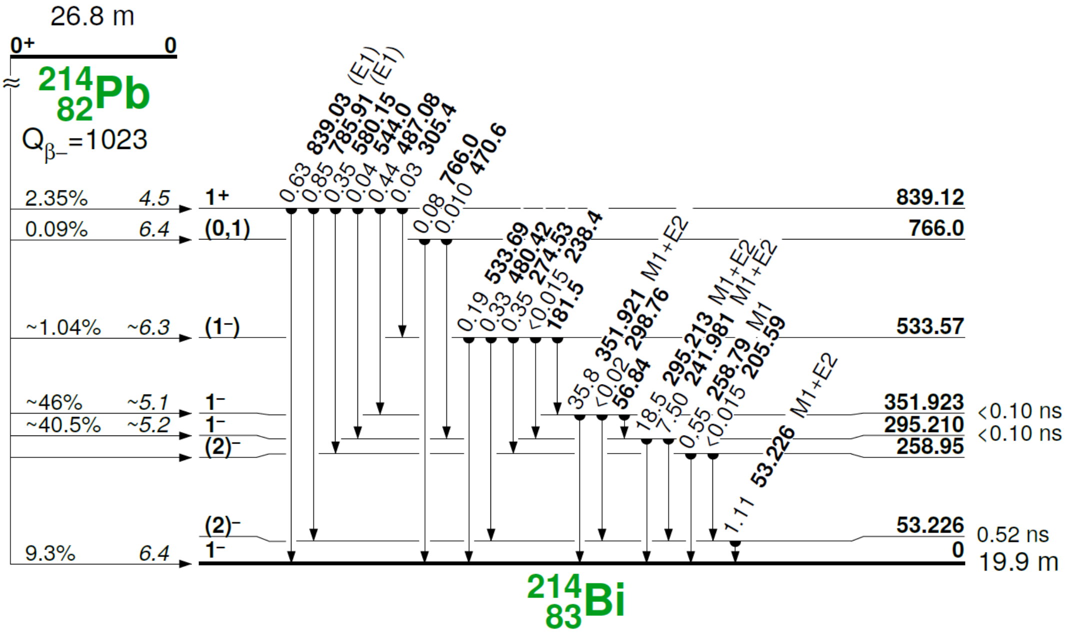


Figure 6: Decay scheme of ^{214}Pb (adapted from [20]).

However, the other decays of the chain can be better identified of the spectrum. The β-emitter ^{214}Bi, which also decays into the ground state of the daughter isotope, can be easily identified by looking at the time correlation of the daughters α-decay [23]. The other isotopes after ^{214}Po are trapped on surfaces within the detector and their generated signals can be separated [24].

Due to the fact that the radon emanates from detector components two different types of emanation sources have to be defined [4]: type one sources are for example components of the detector, which cannot be removed before entering the detector; type two sources on the other hand are sources, where the xenon can be purified, before the contamination can reach the detector. These can be removed by a removal system with a sufficient large reduction factor of the radon. For the type one sources, the concentration of the contamination further depends on the purification flow. Therefore, a new, high throughput distillation column is necessary.

3.4 Radon distillation column

The radon removal system is a distillation column, which lowers the concentration of radon in xenon. The column consists of two heat exchangers positioned at the top and at the bottom with package material in between. The basic idea of the whole column is shown in figure 7.

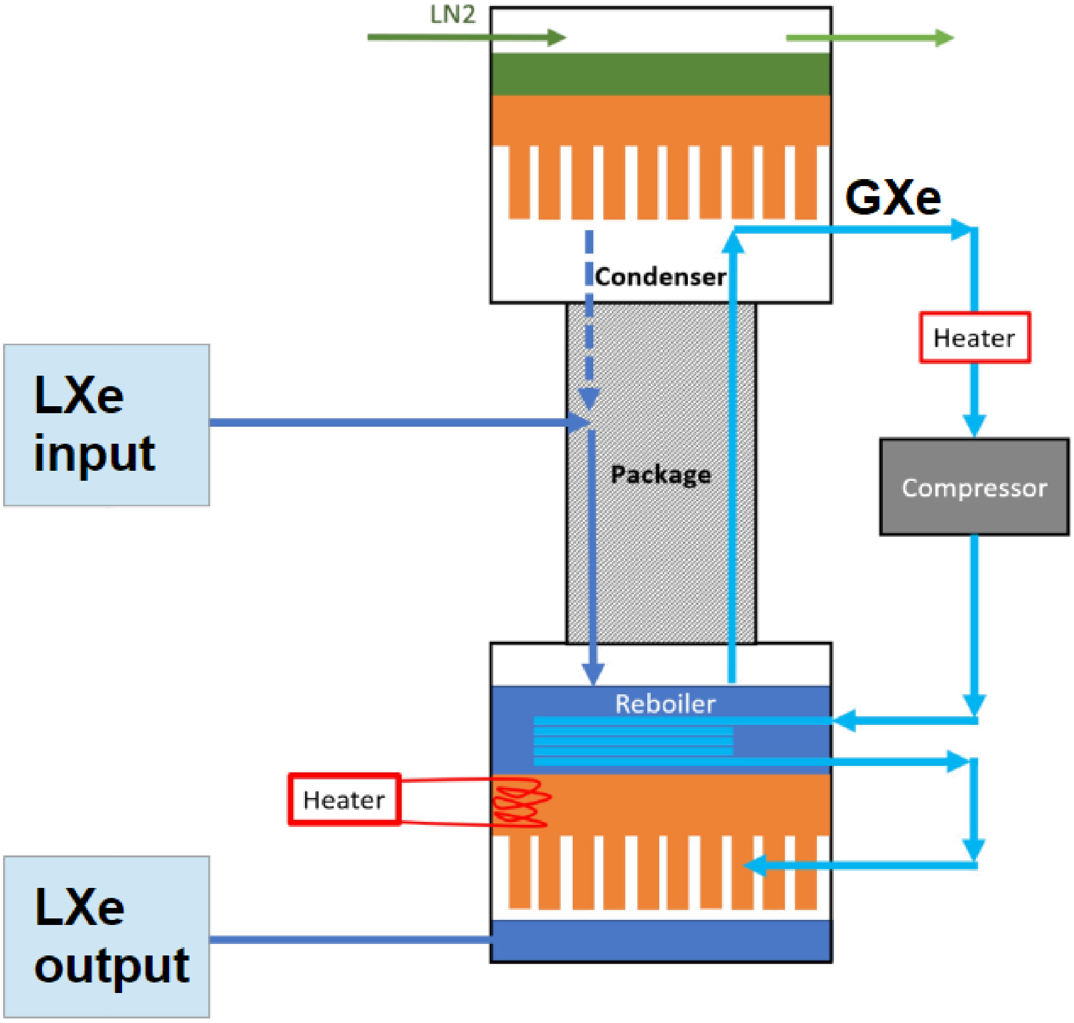


Figure 7: Scheme of the radon removal system [25].

The input for the liquid xenon from the liquid purification loop is in the middle of the package material. Here, the xenon drips to the reboiler, which is at the bottom of the column. In the reboiler heaters evaporate the xenon. Radon enriched xenon remains in the reboiler, since radon is less volatile than xenon [26] and the radon-depleted xenon goes through the package material into the top condenser. In order to reduce the probability of radon particles leaving the column, a part of the xenon will be condensed corresponding to a reflux. Liquid nitrogen cooling is used, transferring its cooling power via large surface oxygen free copper fins to the xenon. To regulate the cooling process, this heat exchanger also contains heaters. The remaining clean gaseous xenon, which is not condensed, leaves the condenser. With the intention to get the highly radon depleted xenon back to the liquid purification loop, the xenon has to be liquefied again. Therefore, the principle of a heat pump reducing the required cooling power is used. The clean xenon is forwarded to a compressor, which has to fullfil a throughput of 200 slpm and a compression of $\Delta P > 2\,\text{bar}$. The output of the compressor is further routed through a metal coil that is immersed in the liquid xenon in the reboiler. In a second step the xenon will be condensed at the bottom chamber of the reboiler on again large oxygen free copper surface fins before the liquid can be put back to the liquid purification loop.

This thesis will focus on the compressor, which requires special characteristics. Thus, its development is based on a custom-made magnetically-coupled piston pump shortly introduced in the following.

4 The magnetically-coupled piston pump for XENON1T

As mentioned in section 3.3, pumps are required to circulate the target material through the purification system. For experiments like XENON1T such pumps have to fulfill requirements in terms of gas containment, oil-freeness and low radon emanation. Therefore a pump with these special characteristics has been developed, because commercially available pumps do not fulfill these aspects. So the drive with oil-based lubrication is outside of the pump, isolated from the gas, and moves the pump piston over a magnetic coupling. In the following chapter, the design and special magnetic configuration is described.

4.1 Design

The body of the magnetically-coupled piston pump has a length of 520 mm and an inner diameter of 127 mm as shown in figure 8. Moreover the body has flapper valves located at the top and flapper valves located at the bottom. At the top as well as at the bottom flapper valves are for the input flow (left in figure 8) and for the output flow (right), which results in a double-stroke design. A ring containing three cylindrical neodymium magnet arrays with alternating polarity is attached around the body. This outer ring is driven by an electric cylinder drive. The piston inside the pump body has a length of 155 mm and a diameter of 125 mm. Furthermore the piston contains the inner ring of alternating magnet arrays. These magnets are magnetically-coupled to the three magnet arrays in the outer ring. Additionally, the piston rings and external rings have exactly opposite magnetic orientation resulting in a strong cross coupling, which boosts the performance of the pump.

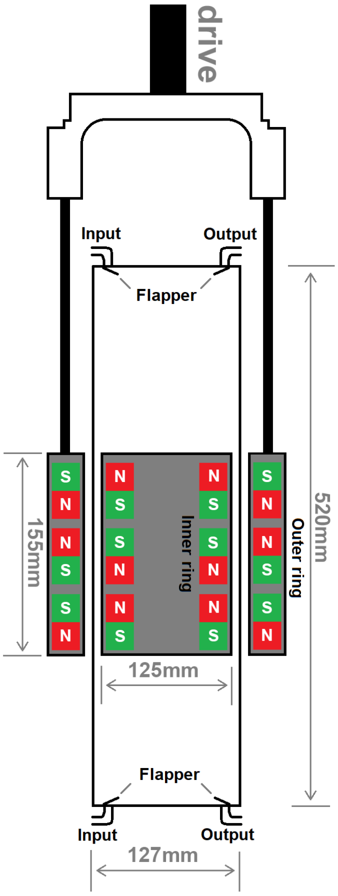


Figure 8: Sketch of the magnetically-coupled piston pump of XENON1T: The alternating polarity configuration of the magnets leads to a strong cross coupling.

Dependent on the inlet pressure this magnetically-coupled piston pump reaches an average xenon flow of about 100 slpm [27]. Due to the double-stroke engine combined with a linear movement of the drive fluctuations of flow and pressure can be observed [27]. Since such fluctuations can be problematic for a stable operation of a radon distillation column, the pressure built-up in- and outside of the pump will be discussed below.

4.2 Pressure curve

In [27] an insight to the theoretic pressure built-up inside the pump was given by a pV-diagram. Based on that, an estimation on the pressure behavior during operation can be done. For clarity, only one working volume, for example the volume above the piston, is shown. Secondly, the input pressure behavior is not considered for convenience. Assuming an infinite xenon reservoir and no heat transfer to the environment the following pressure behavior can be inferred.

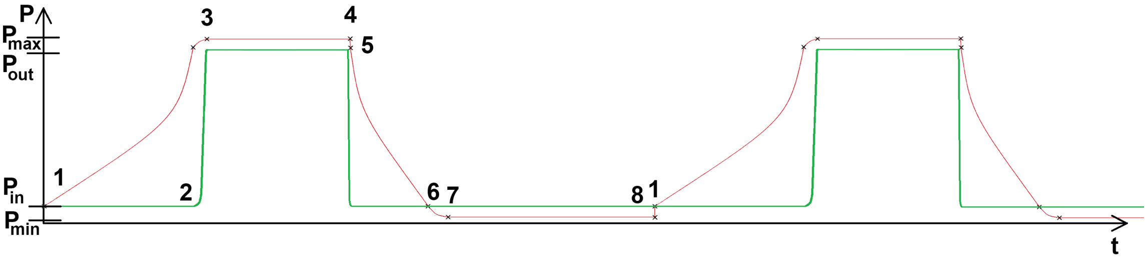


Figure 9: Theoretical pressure inside the pump on one side of the piston (red) and theoretical pressure at the outlet (green) with black points between the different changes of state.

Dependent on the pressure inside the pump, the pressure behind the output flappers is determined. So the different stages relating to the pressure are represented in the following:

- 1 → 2: The piston moves from its lowest position upwards: The pressure inside the pump rises from P_{in} by an adiabatic compression while all flappers are closed (red). Meanwhile the pressure behind the closed output is constant at P_{in} (green).

- 2 → 3 If the pressure inside the pump exceeds P_{out}: The output flappers open and the pressure inside the pump increases up to P_{max} due to the finite time, which is needed to fully open the flapper (red). At the output a prompt rise of the pressure to P_{out} is expected (green).

- 3 → 4 The piston moves still upwards while the output flappers are open: Due to the pressure difference between inside the pump (P_{max}; red) and outside the pump (P_{out}; green) a continuous gas flow through the flappers is expected.

- 4 → 5 The inner piston stops at its highest position for a change of the drives direction: A pressure equalization takes place and the output flapper close (red). The outlet pressure drops back to P_{in} (green).

- 5 → 6 The piston moves downwards: An adiabatic expansion of the gas inside the pump ensues (red). Meanwhile the pressure at the outlet is constant at P_{in} (green), because the flappers are still closed.

- 6 → 7 The pressure inside the pump falls until P_{in}. Then, the input flapper open such that gas can flow into the pump (red). This has no impact on the output pressure of the pump (green).

- 7 → 8 The inlet flappers are fully open and a steady gas flow enters the pump volume such that the pressure stays constant inside the pump (red). Furthermore no effect to the output pressure is expected (green).

- 8 → 1 The piston stops again at the lowest position for the change of the movement direction: A pressure equalization between the inlet and the working volume occurs and the cycle restarts with an up movement.

The cycle with the different changes of state shows that the pressure built-up inside the pump determines the output pressure. Thus the output pressure P_{out} of one single-stroke can be estimated by the adiabatic changes of state within the pump described by

$$P_{\text{out}} = \frac{P_{\text{min}} V_{\text{max}}^{\kappa}}{V_{\text{min}}^{\kappa}} \tag{6}$$

As a consequence for the double-stroke engine, the resulting compressions per time at the output are doubled compared to figure 9, which is shown in figure 10.

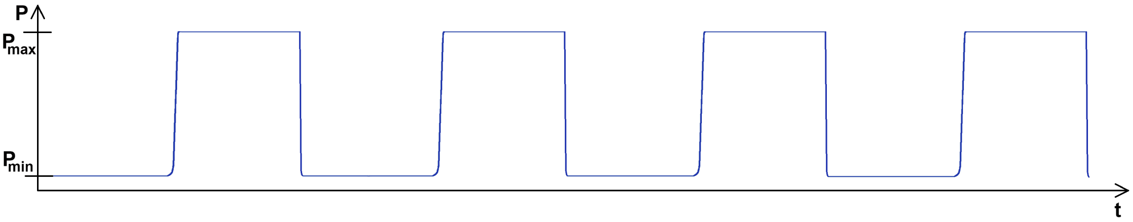


Figure 10: Sketch of the pressure behavior at the outlet of a double-stroke magnetically-coupled piston pump.

As mentioned in section 3.4 the future radon distillation column needs a 200 slpm xenon throughput with a steady flow and pressure output to guarantee a stable operation. Therefore a four-cylinder pump based on the magnetically-coupled piston pump for XENONnT is under development and thus will be explained in the following.

5 The four cylinder magnetically-coupled piston pump

The magnetically-coupled piston pump for XENON1T features a throughput in a range of 100 slpm of xenon. As explained in the previous section a single cylinder pump produces a fluctuating flow and compression at the output, which has to be avoided for the pump at a radon distillation column. In order to ensure this and reach the needed flow of 200 slpm a four-cylinder pump is planed, which is illustrated in figure 11.

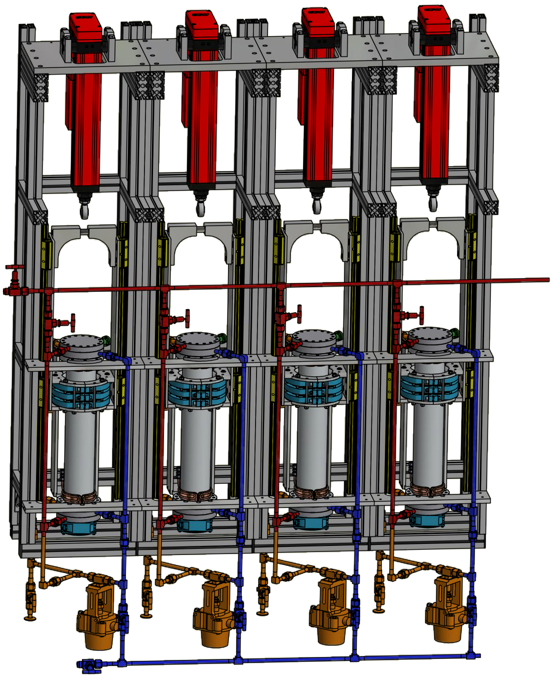


Figure 11: CAD-drawing of the four-cylinder pump for the radon distillation column. Drawing by C. Huhmann.

Reaching higher flows than the pump at XENON1T the four cylinders have to be operated in a parallel configuration. In such composition each pump has only to reach a flow of 50 slpm. Furthermore, the output pressure is the same pressure as the highest pressure of each pump. By adding a shift between the drives the fluctuations of flow and pressure can be reduced. In order to optimize this aspect the phase shift of the drives and its effect to the output pressure and flow will be discussed in more detail below.

5.1 Phase shift effect on the pressure and flow

The behavior of the flow Q and the pressure P behind the output flapper can be assumed to be similar when an output flapper opens. Based on the theoretic considerations of the last chapter, the flow of a one cylinder pump extends like shown in figure 12.

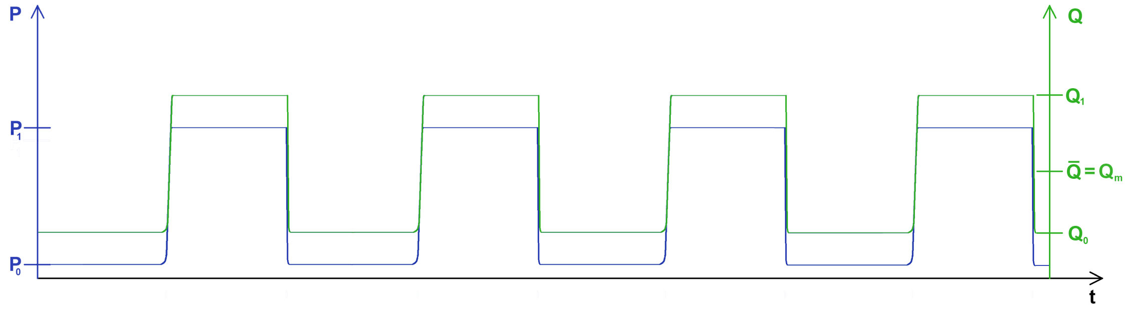


Figure 12: Fluctuations of output flow Q and output pressure P of a double-stroke engine pump as the magnetically-coupled piston pump.

Figure 12 shows the theoretically assumed flow behavior in green and pressure behavior in blue for a double-stroke pump behind the output flapper. At closed flappers a flow of $Q_0 = 0\,\mathrm{slpm}$ and a pressure behind the flappers of $P_0 = P_{\mathrm{system}}$ is expected. Whenever the output flappers open, a flow of Q_1 and pressure increase up to P_1 is reached. Moreover, an average flow of Q_{m} is reached over time.

Reducing these fluctuations, the low level zones have to be compensated such that $\bar{Q}$ comes closer to Q_1. The intuitive idea is to operate with two drives phase-shifted by 50 % of the single-stroke length. This way, the fluctuations can be reduced shown in figure 13.

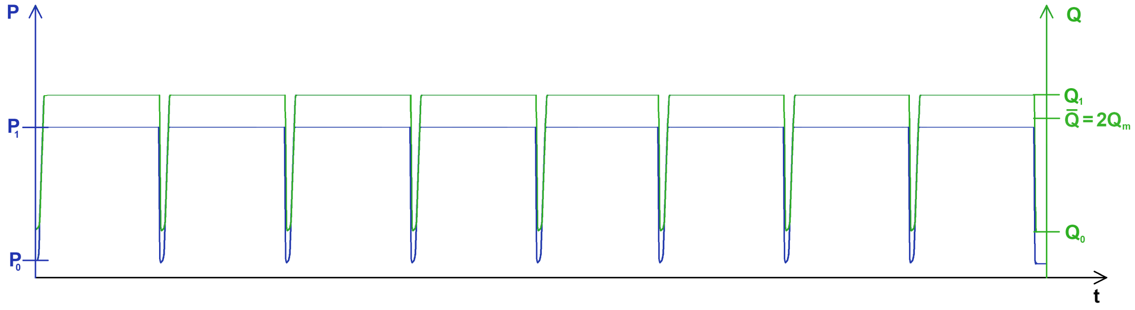


Figure 13: Output flow and output pressure by two pumps with a shift of 50 % of their single-stoke length. The oscillations could be flatten.

Additionally the average flow $\bar{Q}$ at the output is $2Q_{\mathrm{m}}$ since two times the gas volume is displaced at the same time. However, both the flow as well as the output pressure features still high oscillations, which results from opening and closing processes of the flappers. In order to further smooth fluctuations two more pumps will be used. This corresponds to four pumps with a 0 %, 25 %, 50 % and 75 % shift. This theoretical considerations should result in a completely constant pressure as shown in figure 14.

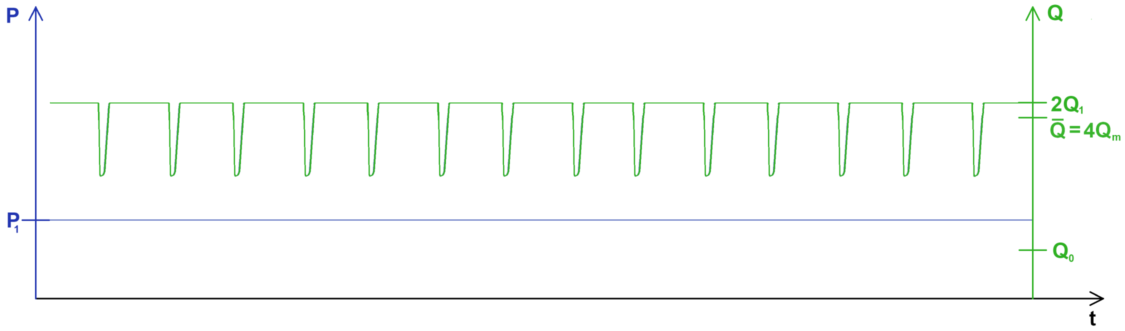


Figure 14: Output flow and output pressure of four pumps with a shift of 25 % of their single-stoke length.

Additional to the constant pressure the resulting flow will be doubled again to $4Q_\mathrm{m}$. Moreover the doubled flow will become even smoother since the optimal phase-shift will produce a continuous throughput. So to fulfill these aspects for a stable operation four cylinder drives of the new pump have to be synchronized to keep the optimal phase-shift constant during operation, which will be described in the next section.

6 Synchronization

As mentioned in the previous section the drives have to be synchronized with an offset of 25 % of the single-stroke length to each other. Therefore an interface is needed, which guarantees the shift between the four drives for the whole time of operation. Further requirements are having a full control of the speed and to be able of monitoring other parameters like a potential offset from the desired shift of each drive. At first the already used interface of the magnetically-coupled piston pump at XENON1T will be introduced. Afterwards the programming-based implementation of the drive-profiles and the slow control will be shown, which was developed together with Denny Schulte, who will also describe the synchronization in his upcoming PhD-thesis.

6.1 The interface

Similar to the magnetically-coupled piston pump at XENON1T commercially electric cylinder drives combined with frequency converters from the company SEW are used. As shown in detail in [27] the SEW drives are suitable for such applications due to their reliability for strong forces over longterm and the flexibility in programming of the movement and monitoring. A test setup for the synchronization of the drives has been built as shown in figure 15.

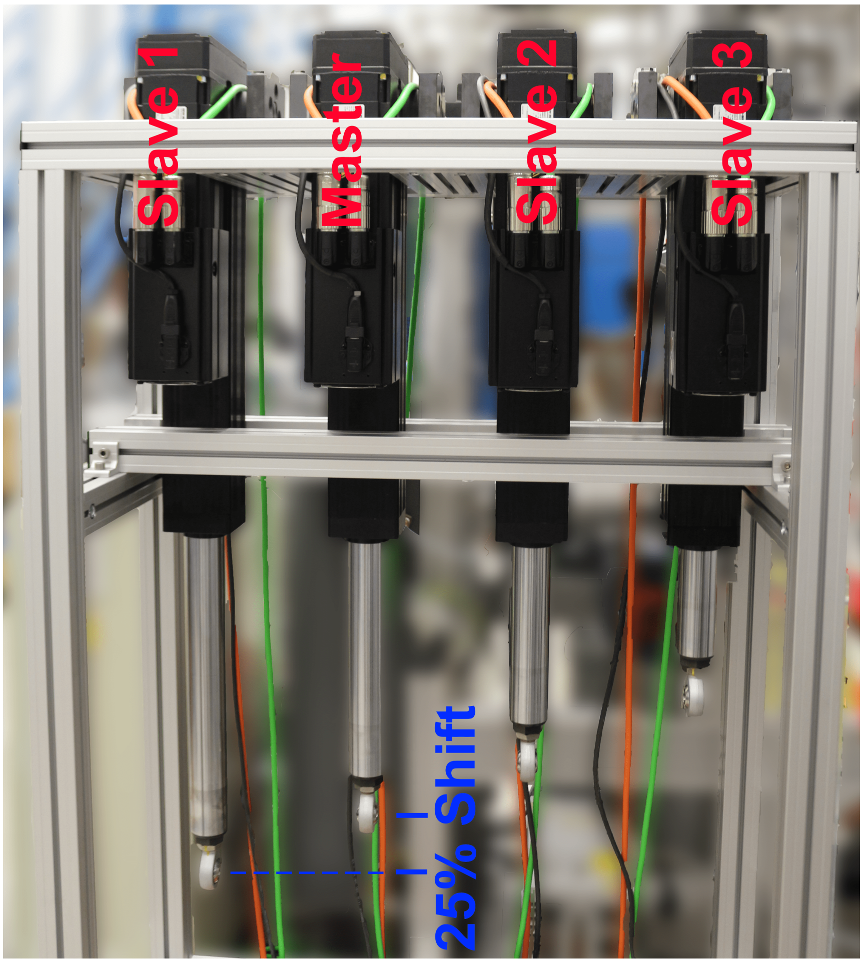


Figure 15: Synchronization test setup for the four electric cylinders drives.

Comparable to the magnetically-coupled piston pump of XENON1T the four drives will initially operated with the SEW engineering-software called “MOVITOOLS”. After uploading the programming code written in the C based SEW programming language IPOS to the frequency converters the drives are controlled and monitored via fieldbus on a slow control.

Here only one drive, which is called master in the following, is connected over fieldbus to the slow control. The other drives called slave communicate just with the master via SBus. Thus the implementation to the XENONnT slow control later on is similar to the integration of one cylinder drive, which was already done for XENON1T, and the control of all four cylinder drives can be kept simple for the operator. The master and slave allocation depends only on the uploaded programming code to the related converter and the fieldbus connection such that the drive can be changed in case of maintenance. A schematic of the default allocation and communication structure is shown in figure 16.

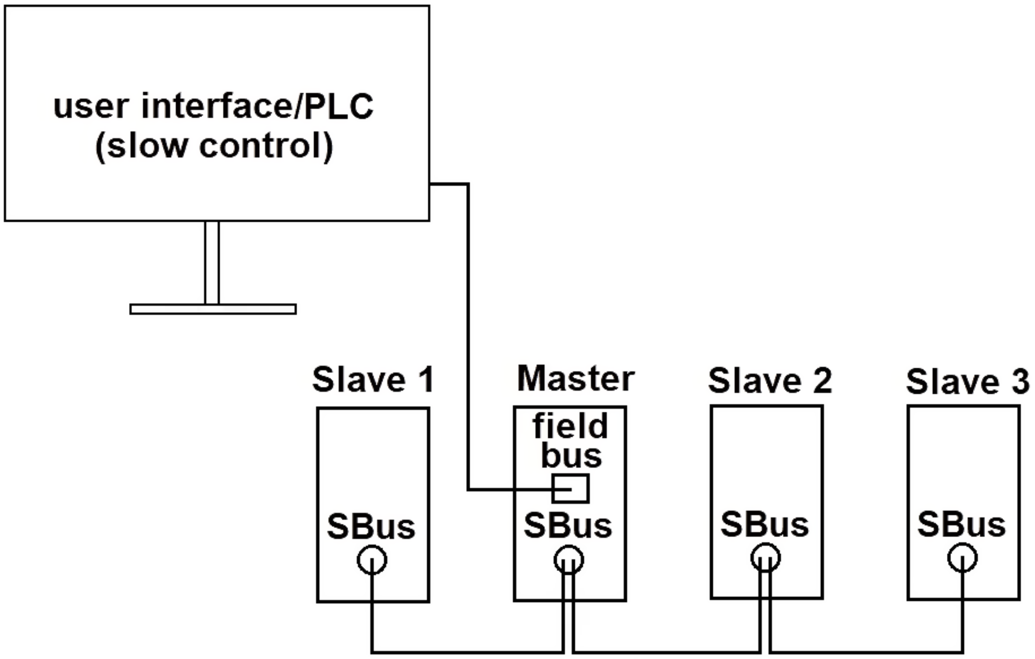


Figure 16: A schematic drawing of the communication between the drives and the slow control. The user interface communicates only with the master. Whereas the slaves get their instructions only from the master drive.

As a consequence of the SBus connection between the drives a synchronization of position and speed can be realized, which guarantees a long term operation with the optimal shift as mentioned in section 5.1. In the following the IPOS programming with the focus on the SBus communication channel will be discussed.

6.2 Programming of the movement

The intention of this section is programming the master to drive continuously up and down and to synchronize this movement with the slaves. Moreover the implementation of the required shift of the slaves will be discussed. Synchronizing the drives the SEW tool “electronic cam“ [28] was used at first, which assigns each master position to a slave position. Considering to the 25 % shift the drives reach their dead centers at different times and with a speed dependent length of stay. To change the direction of movement the drives have to decelerate before and accelerate after this dead centers. By virtue of this kind of synchronization the slave would accelerate and decelerate at the same time as the master does. Thus an own synchronized drive profile will be programmed with IPOS. Moreover, the realization of a status monitoring of each drive will be discussed. The following description of the code refers to the commented IPOS code in the appendix A.

6.2.1 Master

After initialization of the IPOS program (A.1.1) the used local and global variables are defined and initialized in figure A.1.1, A.1.2 and A.1.3. Subsequently a movement to the start position (figure A.1.5) and the up and down driving profile of the master was programmed similar to the code of the existing magnetically-coupled piston pump at XENON1T (figure A.1.9 and A.1.10). In this section of the code several state parameters are used to describe up or down movement. Additionally the states of acceleration and deceleration, which begin and end 8 % of the single-stroke length before and after the change of direction, get a characteristic state parameter. Such states are sent with the masters *current position* to the slaves for the synchronization. For this cyclic transfer of the parameters a SBus communication between the drives has to be developed. With reference to this, SEW offers the possibility to define a so called "SCOM structure" [28]. The initialization of the "SCOM structure" is given in figure A.1.4. This SBus communication can transfer the information over several channels defined in the "SCOM structure". Considering to the data transfer between the drives the master sends information over channel 1025 and receives the slave data over the channels 1060 to 1062. The received data is used to encode the status of the three slaves for the monitoring user interface (figure A.1.6 and A.1.7), which will be discussed in more detail later. Moreover in the state of acceleration and deceleration the master sends the *set speed* instead of the *current position* over this communication channels. On one hand a change of the *set speed* for the master has to be transferred to the slaves. On the other hand further synchronization of the master and slaves during acceleration and deceleration of the master would result in a slow down and a speed up of the slaves, while the slave do not have to accelerate and decelerate due to the phase-shift. After the slave changes the moving direction the synchronization to the master starts again. The deceleration and acceleration sections of the master amount to only 8 % of the whole stroke length, which allows a long enough synchronization time over the entire stroke length. The realization of this synchronization will be explained more in detail in the following section about the programming code of the slaves.

6.2.2 Slave

The before discussed programming code of the master sends the position and a state parameter of the master to the slave. With this data the slaves movement should drive with the desired shift. At first the IPOS programming code is initialized and variables are defined similar to the masters code (figure A.2.1 and A.2.2). In order to get the above mentioned parameters from the master, which are essential for the synchronization, the SBus communication also has to be configured (figure A.2.2). For safety reasons the slaves will not move until the communication with the master can be verified by waiting for received parameters from the master after setting them to default (figure A.2.3). Afterwards the slaves move with the first received parameters from the master to the start position (figure A.2.3 and A.2.4). The code segment in figure A.2.4 checks if the slave is moving to send this information to the master for monitoring. After positioning at start position the slaves IPOS code run into the main loop, which contains different cases shown in figure A.2.5 and A.2.6. The cases depend on the masters movement. Thus an up and down movement case is used. Each of this two cases are divided into two more cases: if the slave drives in the same direction compared to the master or not. This is important for the calculation of the desired slave position and to partly pause the synchronization due to deceleration and acceleration sections of the slaves. If the master and slave are moving in the same direction and both are more than 8 % of the stroke length away from the dead center, then they will be synchronized with regard to the desired shift. So the master sends its *position* to the slave which is used to calculate the desired target position of the slave (figure 17 a)). If the master enters the 8 % interval around the dead center, the synchronization is paused to avoid deceleration of the slave at the same time. This unsynchronized sections are used to synchronize the set speed of the master to the set speed of the slave. Thus a change of the speed by the user interface is transferred after one stroke to the slave (figure 17 b)). If the slave is within the 8 % interval around the dead center, the synchronization will also be partly paused and the slave uses the startposition and endposition (figure 17 c)).

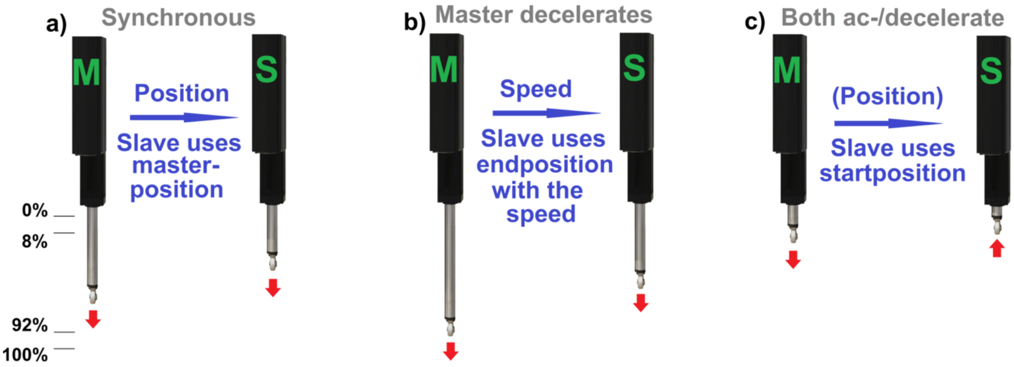


Figure 17: A schematic overview about sending parameters of the master and which parameters are used from the slave.

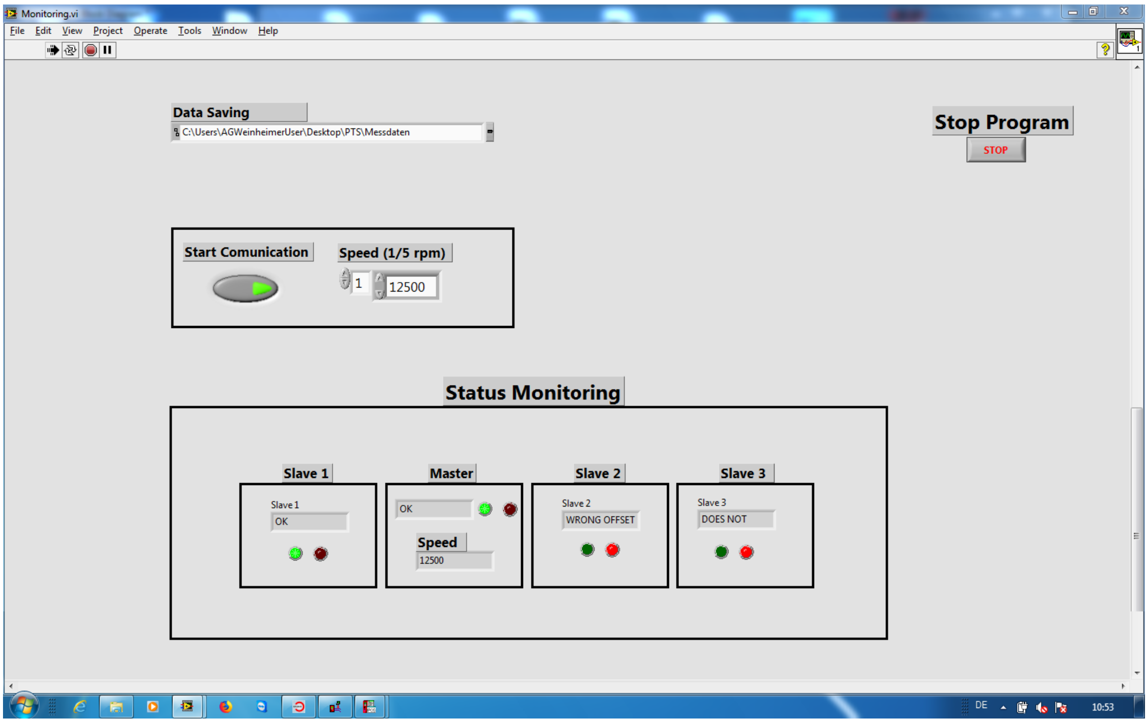
This way the synchronization of the master and the slaves with its optimal phase-shift could be realized, where the movement of the slaves regarding the phase-shift to the master is different. The slave 1 is a special case, because the drive has not enough time to synchronize to the master position with a 75 % shift behind. As mentioned before, the synchronization is realized at the section of the drive profile where the synchronized drives are moving in the same direction. With a 75 % shift of the stroke length this section is only 25 % of the stroke length minus 8 % by acceleration and 8 % by deceleration sections. First tests show that the remaining 9 % of the stroke length does not suffice at speeds higher 1400 rpm to successfully synchronize to the desired target position. To increase the synchronization interval of this drive the drive profile was changed to a 25 % shift of the stroke length ahead the master position (figure A.2.5a). Further slave 2 and 3 drive 25 % and 50 % behind the master (figure A.2.5b). In the following the data monitored with the slow control during operation will discussed more in detail.

6.2.3 Monitoring of the drives

As already mentioned in the previous chapters the monitoring have to be realized via SBus communication from the slaves to the master and via fieldbus from the master to the slow control. Therefore several parameters of the slaves, which display its status, have to be sent to the master. As mentioned in [27] the frequency converters of the electric cylinder drives support a data transfer to external devices of three process data variables via modbus. Similar to the drive at XENON1T two of the parameters are used for the *status word* and the *speed* of the master. Hence the status of all slaves has to be send over one parameter. Therefore the status of each slave is encoded by the master to a number code, which is subsequently decoded by the slow control. Thus the status code of each slave contains the information:

- if the communication is fine
- if the slave is moving
- when moving, if the shift is correct.

To encode this information the slaves send their position and a control variable, which signaled that the communication is fine (figure A.2.4), to the master. The master encodes with the two variables of each of the slaves the possible status to a single digit code (figure A.1.6 and A.1.7). Thus, the current status of each drive will be combined to a collective parameter (figure A.1.8). This collective parameter can be send via the third process data variable and decoded by the slow control. In Münster a LabVIEW-based slow control combined with a real-time micro-controller for a secure control and monitoring is used. So a status monitor four the four cylinder pump has been developed in LabVIEW shown in figure 18. In the following the shift of the four drives with this synchronization code will be analyzed.



Data Saving: C:\Users\AGWeinheimerUser\Desktop\PTS\Messdaten

Speed (1/5 rpm): 1 12500

Slave 1: OK

Master: OK

Speed: 12500

Slave 2: WRONG OFFSET

Slave 3: DOES NOT

Figure 18: LabVIEW-based status monitor.

6.3 The analysis of the drives shift

The IPOS code explained before, was now downloaded to the converters for a first test. For the analysis of the shift between the drives the position of them has to be sent and stored on a computer. As mentioned before the frequency converters supports only three process data variables via modbus. Thus a LabVIEW-based user interface was written to save the position of two drives, the master and one of the slaves. The LabVIEW program generates a textfile containing position values, which can only be in the interval of $[0, 65536]$ by virtue of the limited digit system ($16^4 = 65536 \mathrel{\hat{=}} 4$ bit hexadecimal number system). Nevertheless for the pump a stroke of an interval of $[0, 240000]$ increments is needed. To solve this problem a java program was written, which corrects the data by analyzing the overflow of the LabVIEW output.

Plenty of measurements are essential to investigate the velocity dependency of the synchronized movement. Additionally, the average of the shift between the drives has to be of appropriate significance. Therefore, a measurement was made deciding on a convenient measuring time with both a low working effort as well as meaningful number of strokes. The measurement was done with a speed of 1400 rpm[2], which is about the half-range, and a measuring interval of one and five minutes. Figure 19 displays a cutout of the position of master and one slave in percent of the stroke length as a function of time during operation.

[2] rpm $\mathrel{\hat{=}}$ revolutions per minute

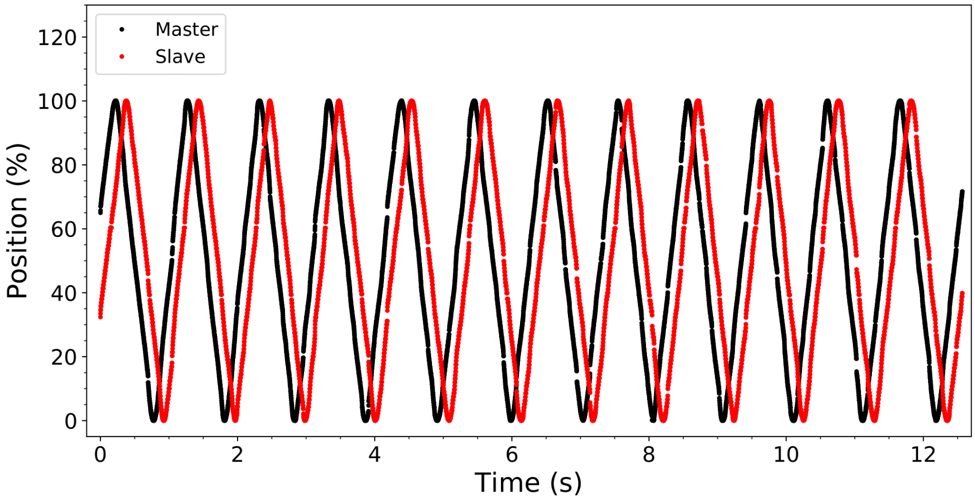


Figure 19: Master and slave 2 position during operation in percent of the single-stroke length (240 000 increments) for a drive speed of 1400 rpm. A cutout of a one minute measurement.

As expected the slave has a similar movement as the master, in which the slave is desired to be postponed by 25 % of the stroke length. However, in this case the slave is postponed by 32 % of the stroke length, which will be explained later in chapters 6.3.1 to 6.3.3. Nonetheless, the difference between both positions seems constant, but when one drive is at a change of direction, the difference will be changed due to acceleration and deceleration sections. Therefore the difference between both positions as a function of time is depicted in the following.

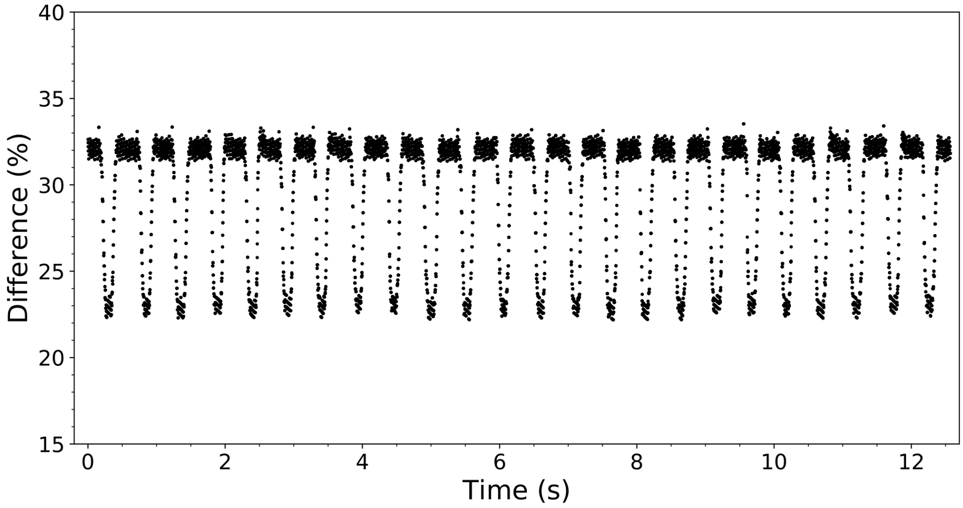


Figure 20: Difference of master and slave 2 position during operation in percent of the single-stroke length (240 000 increments) for a drive speed of 1400 rpm. A cutout of a one minute measurement.

Figure 20 shows a curve with periodic decreases and flat maximums at around 32 %. In this connection the decreases do not fall off to zero, because the difference describes the distance the slave has to drive to get to the masters position instead of a simple position subtraction. The decreases are created by the acceleration and deceleration parts of both drives where they become closer. The flat maximums arises from sections, where the drives move with the same speed and thus will be synchronized with the desired shift in between. Figure 21 illustrates the histogram of the one minute measurement. In order to motivate an expedient binning size for the spectrum three important factors play a role. The LabVIEW-based interface readouts and saves the data every 10 ms. This can be called sampling rate. Another limiting factor is the transfer rate. The here used transfer rate via modbus has a transferring frequency of $f_{\text{trans}} = 100\,\text{MHz}$ [29]. This rate is faster than the sampling rate and thus does not have to be considered. The last factor is the resolution of the absolute encoder (AK1H) of the electric cylinder drive given in *mm* [30]. The conversion of *mm* to *increments* [28] leads to a resolution of 143 increments. This interval of increments is greater than the interval of the sampling rate by a speed of 1400 rpm. Therefore, the accuracy of the measured position values is dominated by the absolute encoder and not limited by the sampling and transfer rate. Under this assumption the binning is set to 143 increments.

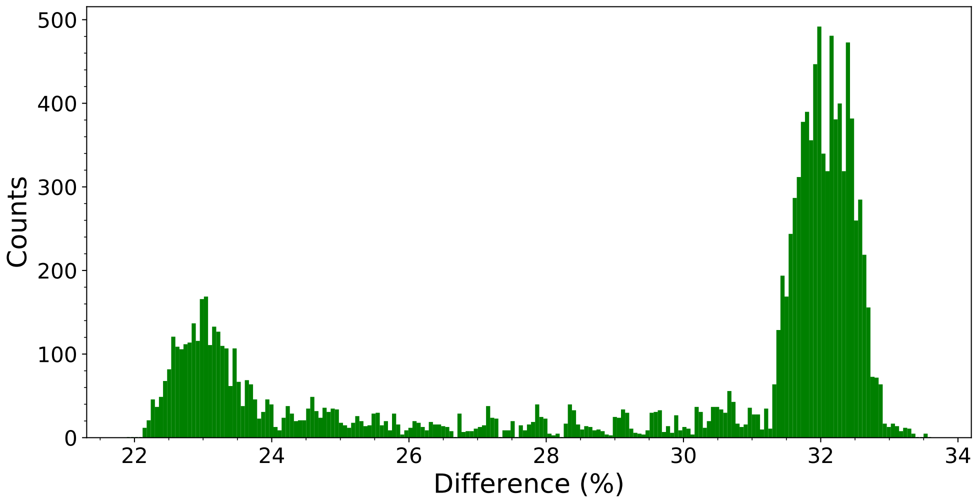


Figure 21: The difference of master and slave 2 position for a drive speed of 1400 rpm measured over one minute shown in a histogram (143 increments bins).

The histogram shows two peaks, where the one at 32 % indicates the difference between the master and slave positions at the synchronized sections. Moreover, the one at 23 % arises from the changes of the direction at a dead center of one drive whereas the other drive is still moving. Since the synchronization is not given in exactly these situations, only the large peak on the right is used to evaluate its working. When the unsynchronized parts, which means 8 % of the stroke length before and after the dead centers, are removed from the spektrum, figure 20 will be change to figure 22. In this one can see that the difference is constant without the decreases from the unsynchronous sections of the drive profile.

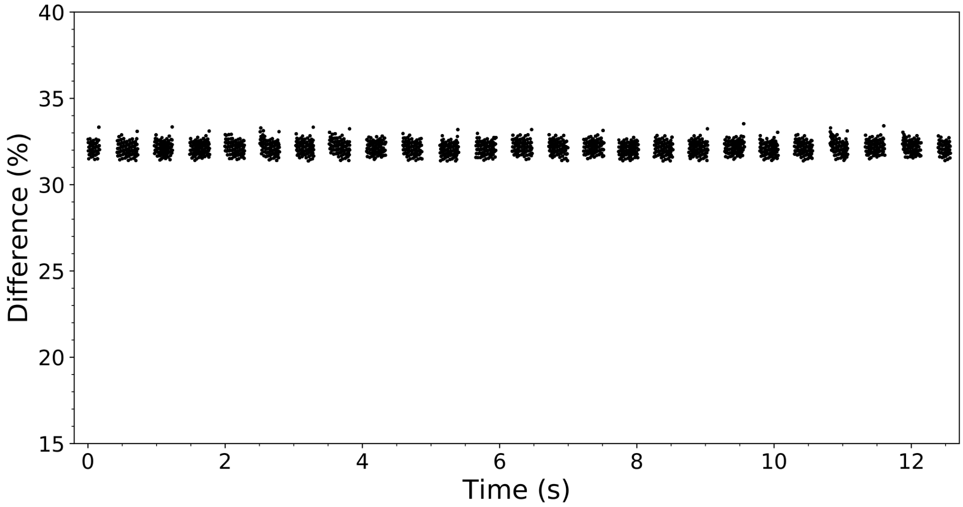


Figure 22: The difference of master and slave 2 position in percent of the single-stroke length for a drive speed of 1400 rpm without acceleration and deceleration sections of the drive profile. A cutout of an one minute measurement.

Based on this the average value of all measured differences between master and slave is calculated. To decide if the measurement of one minute fulfills the conditions of providing meaningful results this measurement is compared to a five minutes measurement. For the five minutes measurement the same analysis has been done. Table 1 shows a comparison of the average value and the maximum deviated values.

Table 1: Compared parameters of the two time measurements in percent of a whole stroke length.

	average difference	minimal difference	maximal difference
one minute	$(32.143 \pm 0.371)\,\%$	$31.37\,\%$	$33.53\,\%$
five minutes	$(32.142 \pm 0.365)\,\%$	$31.34\,\%$	$33.57\,\%$

The comparison shows that the difference of the parameters from a one minute measurement is negligible compared to the five minute measurement. As a result the measuring time of one minute was used for the analysis of the tested IPOS code at different speeds and phase-shifts. Nevertheless to ensure no phase-shift at all a long term measurement over weeks will be shown later on.

The analysis of the different slaves with its specific shift at different speeds shows that the shift does not increase over time but a speed dependent offset from the desired shift was found. For example as perceived before the desired shift of 25 % features at a speed of 1400 rpm an additional offset of $(7.1 \pm 0.4)\,\%$ since a total difference of $(32.1 \pm 0.4)\,\%$ was measured between master and slave position (q.v. table 1). In the following the analysis for the three desired shifts at several speeds is shown and a correction function for each slave is developed to compensate the speed dependent offset. As mentioned before the slave 1 is a special case, which thus at first the two drives moving behind the master will be analyzed in the following.

6.3.1 Slave 2 with 25% shift

The slave 2 drives behind the master with an offset of 25 %. Since the electric cylinder drives feature a maximum speed of 3000 rpm and a slave needs a higher speed to compensate offsets from the target position (in this case 100 rpm) 29 measurements from 100 to 2900 rpm master speed with 100 rpm increment were performed. After data collection the shift between the master and slave 2 has to be analyzed. Additionally a java program offering the possibility to load the LabVIEW generated datafile was written to handle this amount of data. The program calculates the offset from the desired shift between the drives while subtracting the position of the master with the position of the slave minus the shift. This results in a velocity dependent offset and thus has to be corrected. Figure 23 shows the mean value of this offset as a function of the velocity. The uncertainty of each value is determined by the standard deviation:

$$u(\bar{x}) = \sqrt{\frac{\sum_{n=1}^{N} (x - \bar{x})^2}{N - 1}}. \tag{7}$$

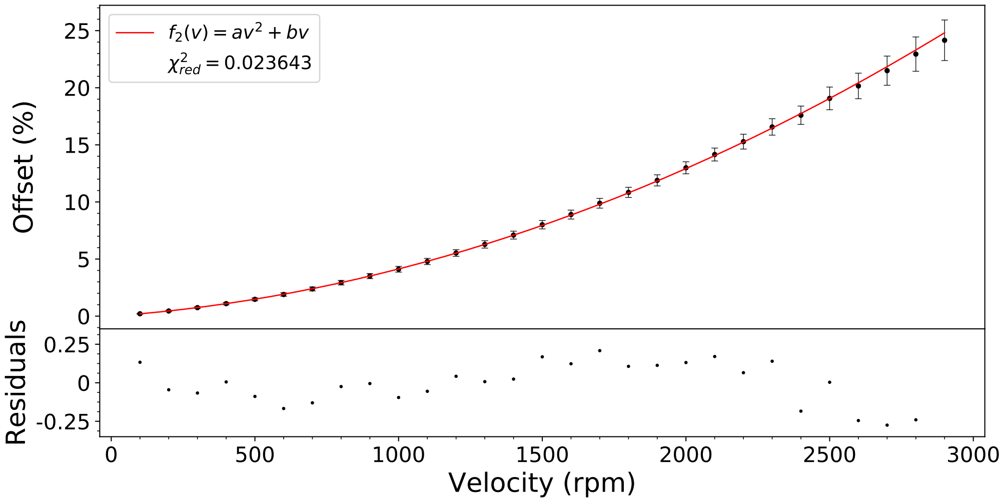


Figure 23: Offset from the desired shift of slave 2 for several cylinder speeds. A velocity dependent increase of the offset to the optimal shift can be seen and fitted with a square function.

Figure 23 demonstrates that the offset of the slave from the desired shift differs at slow velocities by only a few percent from the stroke length. However, the difference between master and slave increases by rising speeds. At the speed maximum of 2900 rpm the offset is about 25 %. This corresponds to position difference between both drives of about 50 % at the synchronous sections of the drive profile instead of the desired 25 %.

In order to compensate this speed dependent offset from the shift a correction function will be determined, which displaces the target position of the slave such that the desired shift can be kept constant.

A square fit in form of $f(v) = av^2 + bv$ was used to describe the offset by a correction function. For slave 2 the correction function

$$f_2(v) = (5.34 \pm 0.05) \cdot 10^{-3} v^2 + (4.8 \pm 0.1)v \tag{8}$$

was found, where the parameters are transferred from percent of the single-stroke length in increments such that the IPOS code can handle. The found χ^2_{red} is clearly below 1. This can be explained by the high uncertainties produced by the large distribution of the measuring points. To find a function, which describes the points even better, some fits with a higher polynomial degree were used and tested on the converters as well. With this functions the drive moves not fluently, because the converters reach the computing power limit. In the following the correction function for slave 3 and slave 1 will be shown and afterwards the working functionality of the implemented correction functions will be checked.

6.3.2 Slave 3 with 50% shift

Likewise as slave 2 slave 3 drives behind the master. The data analysis for slave 3 was similar done compared to slave 2. Figure 24 shows the mean offset from the desired 50 % shift of slave 3 dependent on the velocity. The uncertainties are also determined by the standard deviation (equation 7).

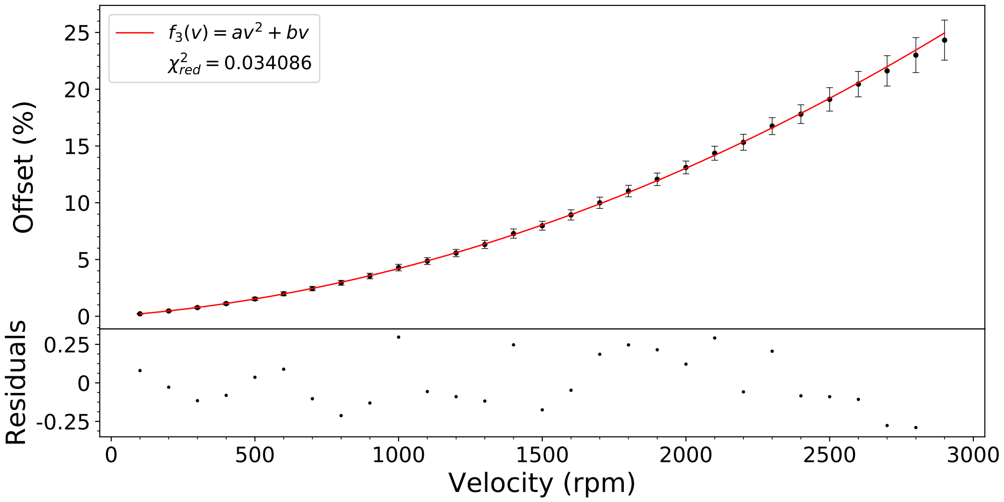


Figure 24: Offset from the desired shift of slave 3 for several cylinder speeds. A velocity dependent increase of the offset to the optimal shift can be seen.

This figure discloses a similar pass like slave 2 with a desired shift of 25 %. At slow speeds the offset differs not much and at higher speeds the offset differs by about 25 %. The velocity dependence can be described by a square function. The square fit in increments results in the correction function

$$f_3(v) = (5.31 \pm 0.06) \cdot 10^{-3} v^2 + (5.0 \pm 0.1) v. \tag{9}$$

6.3.3 Slave 1 with 25% shift

Finally, the data analysis of slave 1 is shown. As mentioned before, slave 1 will move 25 % of the stroke length before the master in contrast to slave 2 and 3 moving 25 % and 50 % behind the master. Considering that the 25 % shift with regard to the single-stroke length has a negative sign in contrast to the shifts of slave 2 and slave 3 similar measurements and analysis were performed. Figure 25 shows the mean of the offset from the optimal shift as a function of the velocity of slave 1.

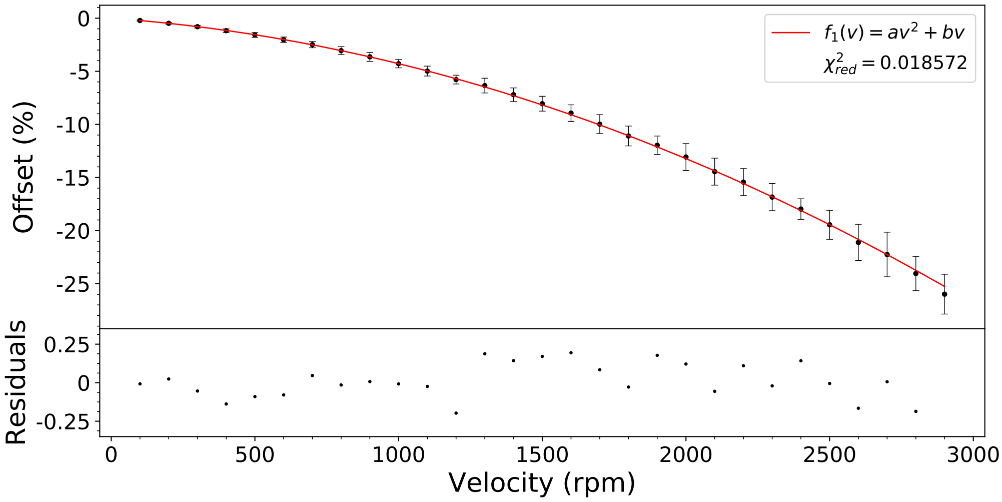


Figure 25: Offset from the desired of slave 1 for several cylinder speeds. A velocity dependent increase of the offset to the optimal shift can be seen.

Through the driving ahead of slave 1 and due to the fact that the movement hang back from the IPOS code, this slave comes closer to the master. Thus figure 25 shows an approaching of master and slave 1. At slow speeds the slave still drives 25 % ahead and at higher velocities the slave moves with a lower distance to the master. This can be described by a square function:

$$f_1(v) = -(5.83 \pm 0.06) \cdot 10^{-3} v^2 - (4.2 \pm 0.1) v. \tag{10}$$

The correction functions f_1, f_2 and f_3 can now be implemented to the IPOS code from each slave to synchronize the related slave to its optimal phase shift from the master. In the following section the implementation and test of this correction functions will be discussed in more detail.

6.4 Corrected IPOS programming code

As mentioned before the correction functions will displace the target position of the slaves such that the desired shift can be kept constant. Therefore the parameter in the IPOS code, which contains the information about the desired shift, must be readjusted (figures A.3.1 to A.3.3).

Afterwards the previous measurements were redone verifying the functionality of the shift correction. With reference to this, the new recorded data will be analyzed in the next subsections.

6.4.1 Corrected 25% shift of slave 2

The analysis of the data for slave 2 synchronized with the correction function is shown in figure 26.

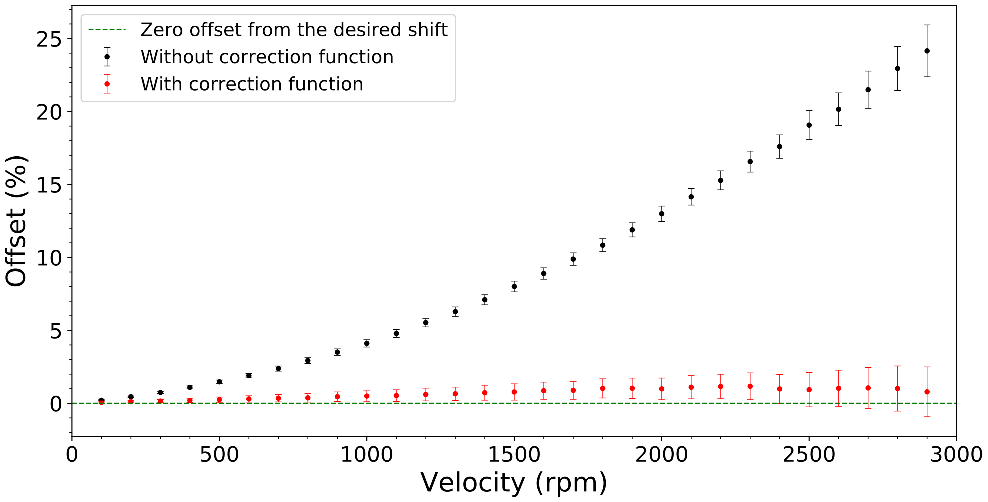


Figure 26: Offset from the optimal shift of slave as a function of the velocity of the electric cylinder drive (red). The implementation of the correction function can lower the offset from the optimal shift.

The figure shows the measurement of the offset between master and slave 2 with correction function (red) and without correction function (black) for several speeds. The correction function can reduce the offset from the desired shift. The offset with implemented correction function differs 1 % of the stroke-length in maximum from the optimal shift. At a stroke length of 35 cm, 3.5 mm offset from the desired shift is not expected to be visible in the performance of the pump, because the desired shift results only in a continuous, steady flow and pressure throughput. Thus the observed offset from the desired shift with the correction function is in an acceptable range for the needed pump as long as it will not increase over time. The long term measurement will be discussed in more detail after the following analysis verifying the correction functions of the other slaves.

6.4.2 Corrected 50% shift of slave 3

The results of slave 3 are shown in figure 27.

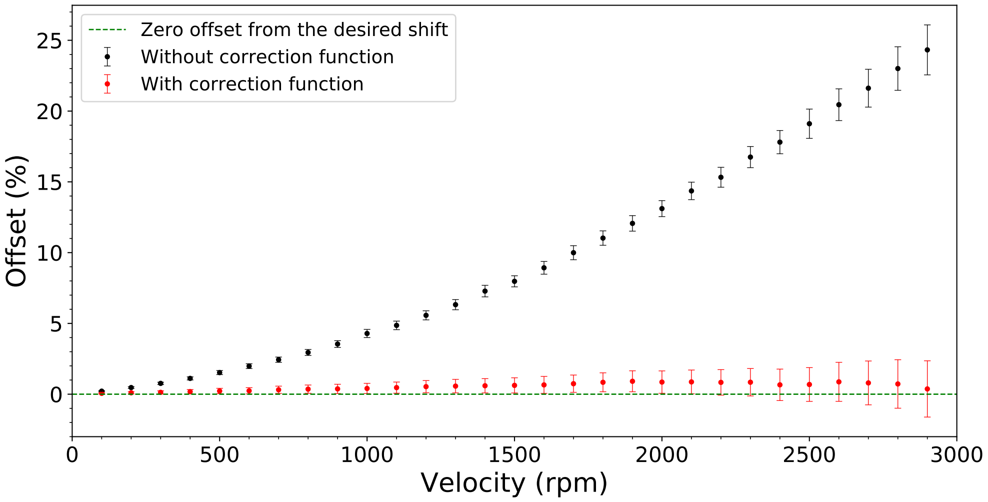


Figure 27: Offset from the optimal shift of slave 3 as a function of the velocity of the electric cylinder drive (red). The implementation of the correction function can lower the offset from the optimal shift.

The figure displays again the measurements of the offset between master and slave, in this case slave 3, with correction function (red) and without correction function (black). As shown slave 3 behaves similar to slave 2. The measuring points also differ up to about 1 % with reference to the stroke length from the zero offset. Thus also slave 3 operates as desired. Finally the correction function of slave 1 is tested and analyzed below.

6.4.3 Corrected 25% shift of slave 1

Comparison of measurements with slave 1 and the master for different speeds with implemented correction function and without correction function.

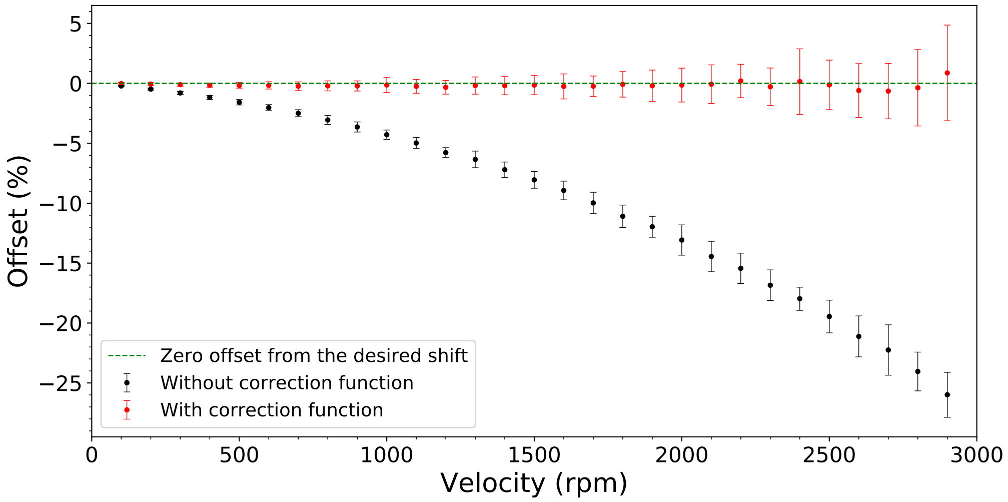


Figure 28: Offset from the optimal shift of slave 1 as a function of the velocity the electric cylinder drive (red). The implementation of the correction function can lower the offset from the optimal shift.

In figure 28 the measuring points with correction function (red) and without (black) are given. The result of the analysis from slave 1 shows that the correction function $f_1(v)$ can compensate the offset from the desired shift similar to the correction functions at the other slaves. The maximum offset from the desired shift again is about 1 % with regard to the stroke length. However the uncertainty at speeds higher than 2300 rpm is larger than at the other slaves. This can be explained by the sections where the synchronization starts again. The slaves driving after have to accelerate in this sections to be synchronous in contrast to slave 1 driving ahead, which has to decelerate until the optimal shift is reached. The slave 1 has to decelerate by virtue of the reverse definition of its programming, which results in a slight overcompensation of the correction function. After slave 1 reaches the optimal shift the slave has to accelerate to drive with the same speed of the master. This small fluctuation leads to a higher scattering of deviation. This effect occurs especially at high velocity by virtue of the higher speed difference. The maximum uncertainty deviates by around 4 % of the stroke length, which also can still guarantee the reliability of a continuous throughput as long as its stable over time.

This way the synchronization of the slaves with their optimal shift to the masters movement could be implemented. On a short timescale the stability keeping the optimal shift constant can be guaranteed. However, similar stability is required over long term to ensure the optimal phase shift during 24 hours per day and 7 days per week operating of the column minimizing the oscillations of the flow and pressure through the system. Therefore a continuous operation of more than one week will be analyzed in the following.

6.5 Constant operation of the pump

The before mentioned longtime measurement was made with slave 3 over nine days, since the deviations are expected to be higher due to the larger 50 % shift to the master position. The measurement was done with a speed of 2500 rpm. On one hand this is the most likely operation speed expected for the operation of the distillation column at XENONnT. On the other hand higher deviations are expected for higher speeds. The analysis of the data will be performed in several steps. At first a comparison of each day will be done to check if the shift is constant. Therefore the average offset from the desired shift is calculated and the maximum offsets around the desired shift will be determined. Afterwards the shift over the measured time of day 4 to 9 will be investigated. The days 1 to 3 are excepted, because a bad sampling rate in LabVIEW was used: If the sampling rate is faster than the SBus communication between master and slave, the slave positions are wrongly allocated to the master position. Finally a comparison of the first ten minutes of day 4 to the last ten minutes of day 9 will be done to ensure no shift over time. The results of the analysis are presented in table 2.

Table 2: The analyzed longtime measurement.

	average offset	max offset before	max offset after
day 1	$(0.9308 \pm 1.2717)\,\%$	$-2.0867\,\%$	$4.0696\,\%$
day 2	$(0.8983 \pm 1.2638)\,\%$	$-3.6521\,\%$	$4.0604\,\%$
day 3	$(0.6750 \pm 1.1704)\,\%$	$-3.6525\,\%$	$3.1642\,\%$
day 4	$(0.6767 \pm 1.1642)\,\%$	$-2.5925\,\%$	$3.1650\,\%$
day 5	$(0.6758 \pm 1.1654)\,\%$	$-2.5925\,\%$	$3.1654\,\%$
day 6	$(0.6767 \pm 1.1683)\,\%$	$-2.5917\,\%$	$3.1642\,\%$
day 7	$(0.6750 \pm 1.1729)\,\%$	$-2.5917\,\%$	$3.1625\,\%$
day 8	$(0.6758 \pm 1.1671)\,\%$	$-2.5921\,\%$	$3.1650\,\%$
day 9	$(0.6763 \pm 1.1704)\,\%$	$-2.5921\,\%$	$3.1646\,\%$
day 4-9	$(0.6763 \pm 1.1679)\,\%$	$-2.5917\,\%$	$3.1654\,\%$
day 4 first 10 min	$(0.7158 \pm 1.1642)\,\%$	$-2.5913\,\%$	$3.0950\,\%$
day 9 last 10 min	$(0.6583 \pm 1.1429)\,\%$	$-2.5900\,\%$	$2.9563\,\%$

The analysis shows that the first three days differ from the other days. The data of day 1, day 2 and day 3 feature a larger average offset compared to day 4 to day 9. Additionally the maximum offset around the desired shift differs from the other days. This can be explained by the bad sampling rate of LabVIEW, whereby at some point the slave position was wrongly assigned to the master position. Furthermore, the data presents that no change of the shift can be determined in the last six days. As well the average offset from the desired shift of day 4 to day 9 is with around 0.68 % in the same range as the maximum deviated day interval of it (day 6 with $(0.6767 \pm 1.1683)\,\%$). Moreover, the long term measurement displays negligible small maximum offsets from the desired shift. About $-2.6\,\%$ and $3.2\,\%$ of the stroke length as maximum differences from the desired shift corresponds to only a few millimeters of the stroke length and thus will not have a visible effect on the pump performance. Furthermore, the comparison of the first ten minutes of day 4 and the last ten minutes of day 9 demonstrates no differences of the offset parameters. Finally a shift within the long term measurement can be excluded.

7 Conclusion

The aim of this bachelor thesis was to synchronize the four cylinders of a newly developed magnetically-coupled piston pump for the radon distillation column of XENONnT. In order to achieve the related requirements, the cylinders have to be operated in parallel with a movement shift regarding a single stroke cycle. This includes the programming of the driving profiles of each electric cylinder with data transfer between the drives, testing of the programming code with the focus on the desired shift between the cylinders as well as verifying a longtime reliability of the synchronization.

Firstly, the desired shift was motivated. Therefore several considerations for the flow and pressure built-up were made starting with one cylinder going on to two and four cylinders in parallel. This theoretical foundations result for four cylinders operated in parallel in a 25 % shift with regard to the stroke length. So a first driving profile based on the position transfer between the drives was developed for each cylinder. However, the desired shift which was programmed into the profiles, could not be kept constant for several speeds of the movement. The first test of this version of IPOS code shows a velocity dependent offset from the desired shift. To compensate this offset a correction function was fitted on the recorded data to readjust the related target position within the programming code. Additionally the found offset of the test shows that the drive with 75 % shift behind the master (slave 1) has not enough time to synchronize to the master. To increase the synchronization time of this drive the drive profile was changed to a 25 % shift of the stroke length ahead the master position. After the appropriate adjustment of the programming code the driving profiles were tested by several speeds with the result that the desired shift can be kept constant. To ensure this operation over longtime a nine days measurement was made. The comparison of the first ten minutes of day 4 and the last ten minutes of day 9 suggests no shift during steady operation. Furthermore the measurement features an average offset of (0.68 ± 1.17) % of the stroke length with the a maximal deviation of -2.6 % and 3.2 % of the stroke length. Due to the fact that the average is almost zero as well as constant over long term and that such maximum deviations are not expected to have an influence to the performance of the pump, the synchronization was successfully realized.

Furthermore a status monitor has been developed to ensure the secure operation of the four piston pump. This monitoring gives information about the individual status of the slaves, which contain the information if the slave is driving, when driving if the slave has the desired shift or if an error is triggered by transferring information. That allows fast acting in case of an unexpected mistake of the system.

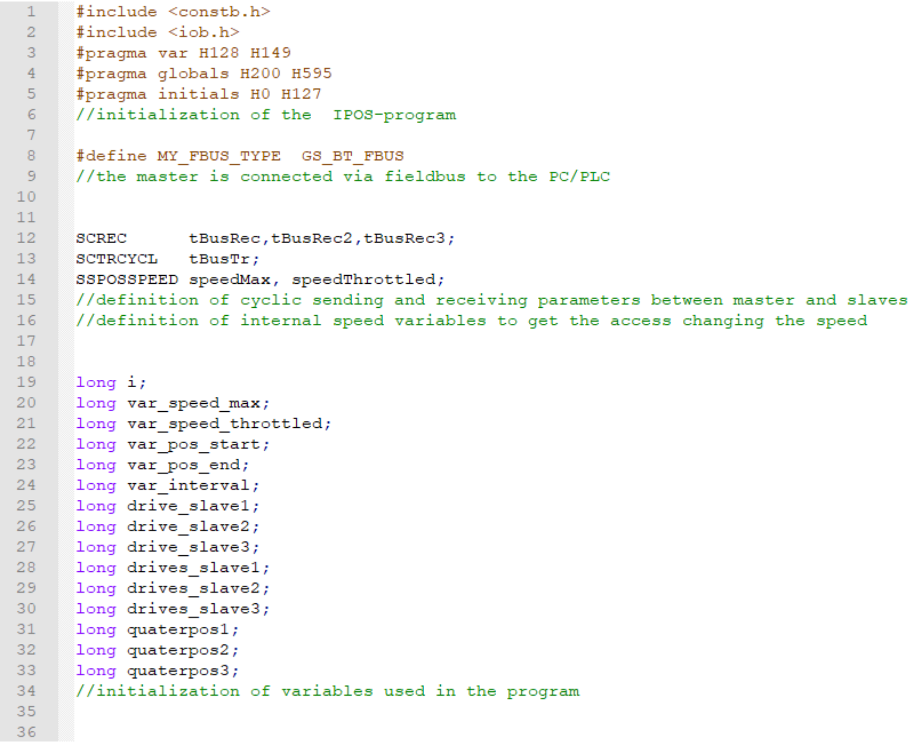
8 Outlook

The newly developed four cylinder magnetically-coupled piston pump is planned to be operated continuously over years. This requires a long term reliability not only from the software-based point of view by synchronization of the drive profiles, but also from the mechanical components. Mechanically moved systems have to be necessarily maintained at some point. To prevent downtime of the whole radon removal system of XENONnT due to maintenance work on one cylinder of the pump, the four cylinders should feature the possibility to decouple one cylinder. In order to operate the remaining three cylinders during such service work in an optimal way, the shifts have to be changed. For this option new correction functions have to be calculated based on recorded data of the configuration with a 33 % shift.

Besides, the monitoring slow control has to be adjusted for the operation at XENONnT. Due to the fact that LabVIEW is used for many applications in Münster by default the first version of the monitoring slow control was also developed in LabVIEW. Thus the next step is to convert the status monitoring from a LabVIEW based slow control into a General Electric based slow control. This conversion is necessary since the status of the slaves has to be monitored over the main slow control of XENONnT, which was developed in General Electric.

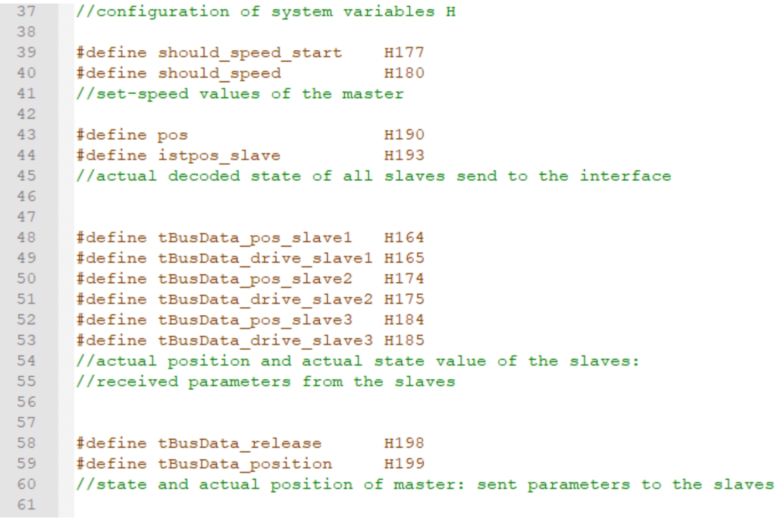
A Appendix

A.1 Program code of master



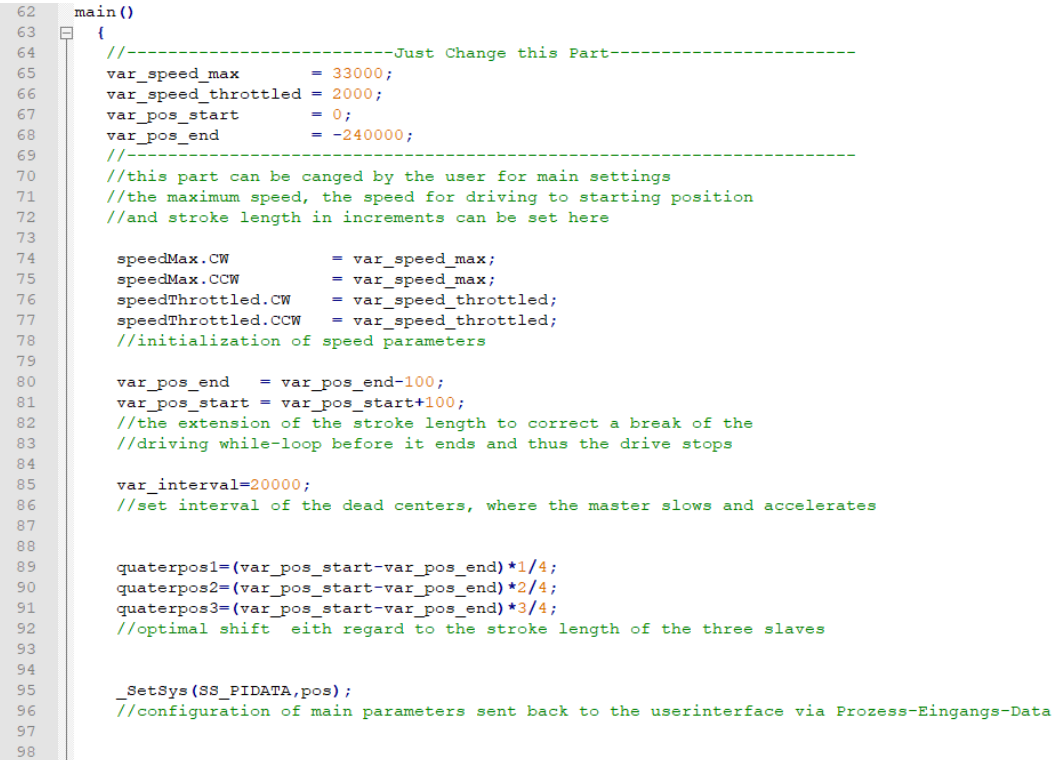
```
#include <constb.h>
#include <iob.h>
#pragma var H128 H149
#pragma globals H200 H595
#pragma initials H0 H127
//initialization of the  IPOS-program

#define MY_FBUS_TYPE  GS_BT_FBUS
//the master is connected via fieldbus to the PC/PLC


SCREC      tBusRec,tBusRec2,tBusRec3;
SCTRCYCL   tBusTr;
SSPOSSPEED speedMax, speedThrottled;
//definition of cyclic sending and receiving parameters between master and slaves
//definition of internal speed variables to get the access changing the speed


long i;
long var_speed_max;
long var_speed_throttled;
long var_pos_start;
long var_pos_end;
long var_interval;
long drive_slave1;
long drive_slave2;
long drive_slave3;
long drives_slave1;
long drives_slave2;
long drives_slave3;
long quaterpos1;
long quaterpos2;
long quaterpos3;
//initialization of variables used in the program


```

Figure A.1.1: General initialization of each IPOS program with definition of all variables.



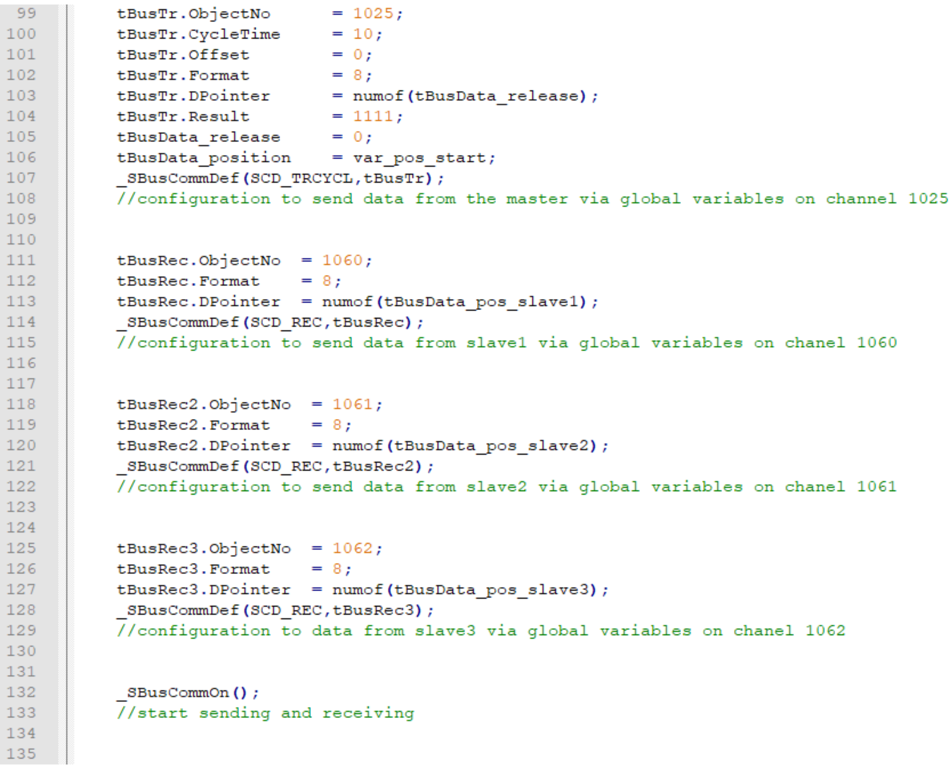
```
//configuration of system variables H

#define should_speed_start    H177
#define should_speed          H180
//set-speed values of the master

#define pos                   H190
#define istpos_slave          H193
//actual decoded state of all slaves send to the interface


#define tBusData_pos_slave1   H164
#define tBusData_drive_slave1 H165
#define tBusData_pos_slave2   H174
#define tBusData_drive_slave2 H175
#define tBusData_pos_slave3   H184
#define tBusData_drive_slave3 H185
//actual position and actual state value of the slaves:
//received parameters from the slaves


#define tBusData_release      H198
#define tBusData_position     H199
//state and actual position of master: sent parameters to the slaves

```

Figure A.1.2: Definition of the global variables.



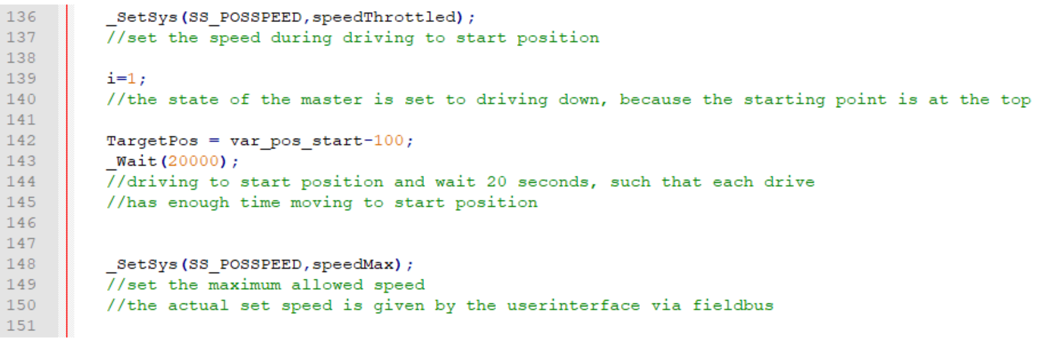
```
main()
  {
   //--------------------------Just Change this Part------------------------
   var_speed_max       = 33000;
   var_speed_throttled = 2000;
   var_pos_start       = 0;
   var_pos_end         = -240000;
   //-----------------------------------------------------------------------
   //this part can be canged by the user for main settings
   //the maximum speed, the speed for driving to starting position
   //and stroke length in increments can be set here

    speedMax.CW          = var_speed_max;
    speedMax.CCW         = var_speed_max;
    speedThrottled.CW    = var_speed_throttled;
    speedThrottled.CCW   = var_speed_throttled;
    //initialization of speed parameters

    var_pos_end   = var_pos_end-100;
    var_pos_start = var_pos_start+100;
    //the extension of the stroke length to correct a break of the
    //driving while-loop before it ends and thus the drive stops

    var_interval=20000;
    //set interval of the dead centers, where the master slows and accelerates


    quaterpos1=(var_pos_start-var_pos_end)*1/4;
    quaterpos2=(var_pos_start-var_pos_end)*2/4;
    quaterpos3=(var_pos_start-var_pos_end)*3/4;
    //optimal shift  eith regard to the stroke length of the three slaves


    _SetSys(SS_PIDATA,pos);
    //configuration of main parameters sent back to the userinterface via Prozess-Eingangs-Data

```

Figure A.1.3: Initialization of used variables.



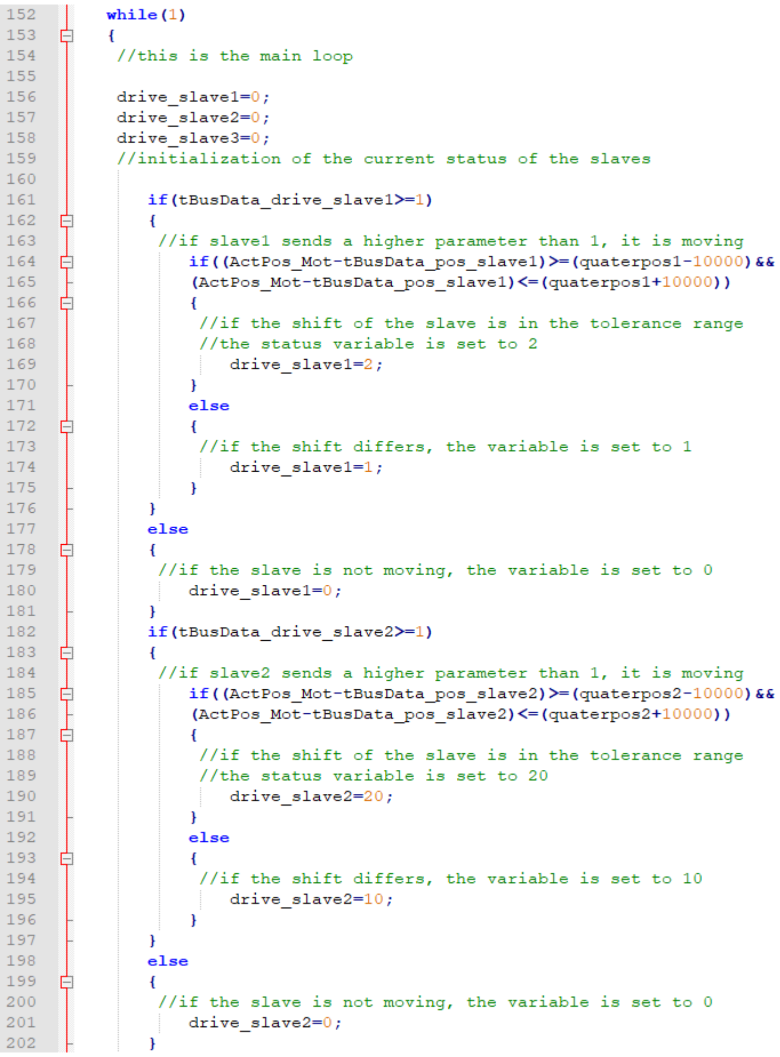
```
    tBusTr.ObjectNo      = 1025;
    tBusTr.CycleTime     = 10;
    tBusTr.Offset        = 0;
    tBusTr.Format        = 8;
    tBusTr.DPointer      = numof(tBusData_release);
    tBusTr.Result        = 1111;
    tBusData_release     = 0;
    tBusData_position    = var_pos_start;
    _SBusCommDef(SCD_TRCYCL,tBusTr);
    //configuration to send data from the master via global variables on channel 1025


    tBusRec.ObjectNo   = 1060;
    tBusRec.Format     = 8;
    tBusRec.DPointer   = numof(tBusData_pos_slave1);
    _SBusCommDef(SCD_REC,tBusRec);
    //configuration to send data from slave1 via global variables on chanel 1060


    tBusRec2.ObjectNo   = 1061;
    tBusRec2.Format     = 8;
    tBusRec2.DPointer   = numof(tBusData_pos_slave2);
    _SBusCommDef(SCD_REC,tBusRec2);
    //configuration to send data from slave2 via global variables on chanel 1061


    tBusRec3.ObjectNo   = 1062;
    tBusRec3.Format     = 8;
    tBusRec3.DPointer   = numof(tBusData_pos_slave3);
    _SBusCommDef(SCD_REC,tBusRec3);
    //configuration to data from slave3 via global variables on chanel 1062


    _SBusCommOn();
    //start sending and receiving

```

Figure A.1.4: Configuration of the SBus communication channel between master and the slaves.



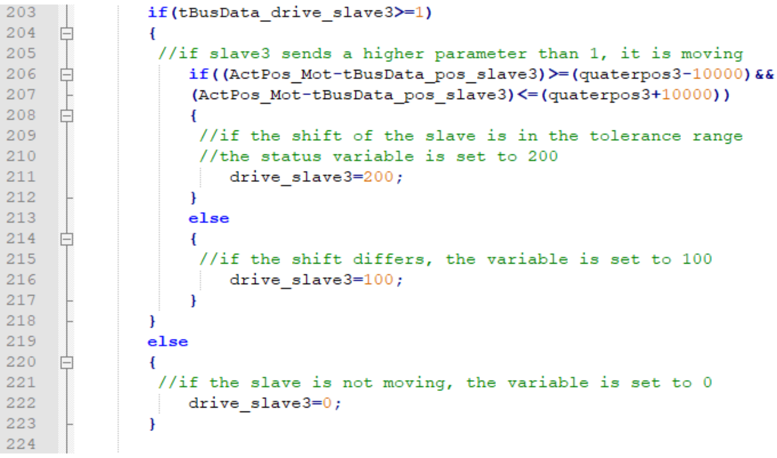
```
_SetSys(SS_POSSPEED,speedThrottled);
//set the speed during driving to start position

i=1;
//the state of the master is set to driving down, because the starting point is at the top

TargetPos = var_pos_start-100;
_Wait(20000);
//driving to start position and wait 20 seconds, such that each drive
//has enough time moving to start position


_SetSys(SS_POSSPEED,speedMax);
//set the maximum allowed speed
//the actual set speed is given by the userinterface via fieldbus

```

Figure A.1.5: The starting process where the master moves to the starting position at the top.



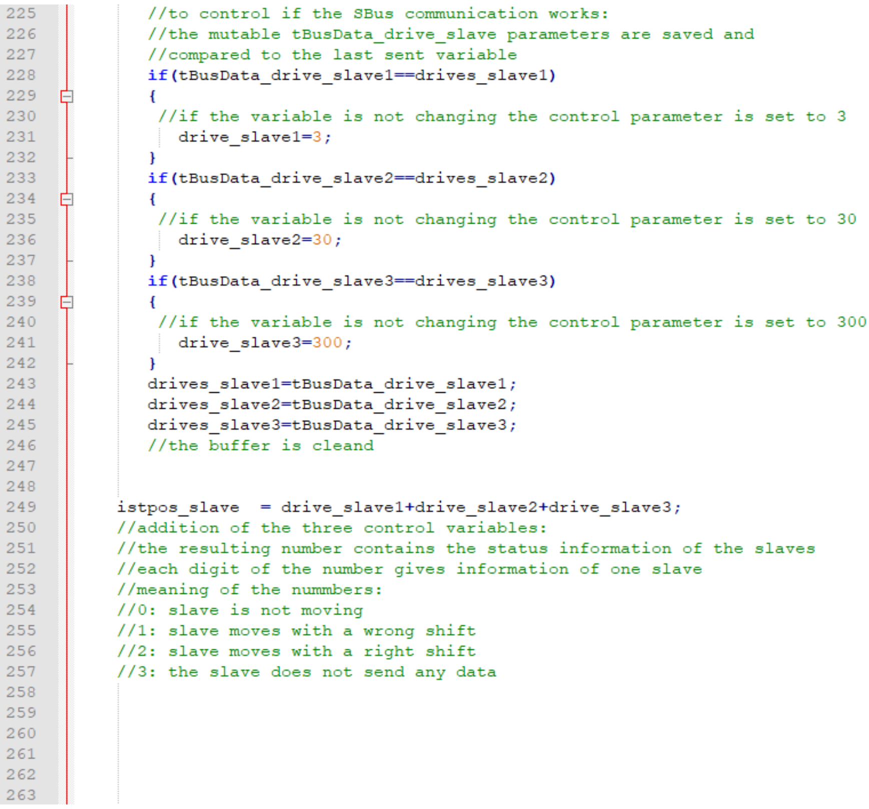
```
while(1)
{
 //this is the main loop

 drive_slave1=0;
 drive_slave2=0;
 drive_slave3=0;
 //initialization of the current status of the slaves

    if(tBusData_drive_slave1>=1)
    {
     //if slave1 sends a higher parameter than 1, it is moving
        if((ActPos_Mot-tBusData_pos_slave1)>=(quaterpos1-10000)&&
        (ActPos_Mot-tBusData_pos_slave1)<=(quaterpos1+10000))
        {
         //if the shift of the slave is in the tolerance range
         //the status variable is set to 2
            drive_slave1=2;
        }
        else
        {
         //if the shift differs, the variable is set to 1
            drive_slave1=1;
        }
    }
    else
    {
     //if the slave is not moving, the variable is set to 0
        drive_slave1=0;
    }
    if(tBusData_drive_slave2>=1)
    {
     //if slave2 sends a higher parameter than 1, it is moving
        if((ActPos_Mot-tBusData_pos_slave2)>=(quaterpos2-10000)&&
        (ActPos_Mot-tBusData_pos_slave2)<=(quaterpos2+10000))
        {
         //if the shift of the slave is in the tolerance range
         //the status variable is set to 20
            drive_slave2=20;
        }
        else
        {
         //if the shift differs, the variable is set to 10
            drive_slave2=10;
        }
    }
    else
    {
     //if the slave is not moving, the variable is set to 0
        drive_slave2=0;
    }
```

Figure A.1.6: Status monitoring of slave1 and slave2.



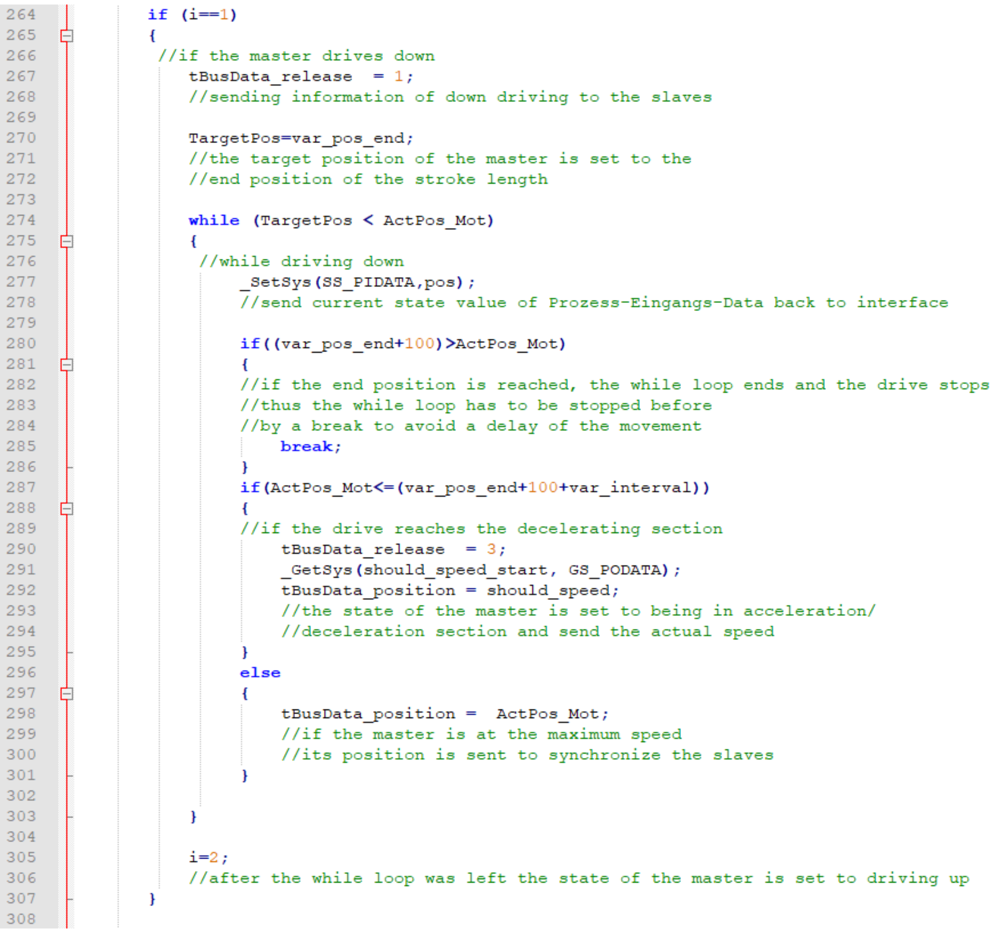
```
    if(tBusData_drive_slave3>=1)
    {
     //if slave3 sends a higher parameter than 1, it is moving
        if((ActPos_Mot-tBusData_pos_slave3)>=(quaterpos3-10000)&&
        (ActPos_Mot-tBusData_pos_slave3)<=(quaterpos3+10000))
        {
         //if the shift of the slave is in the tolerance range
         //the status variable is set to 200
            drive_slave3=200;
        }
        else
        {
         //if the shift differs, the variable is set to 100
            drive_slave3=100;
        }
    }
    else
    {
     //if the slave is not moving, the variable is set to 0
        drive_slave3=0;
    }
```

Figure A.1.7: Status monitoring of slave3.



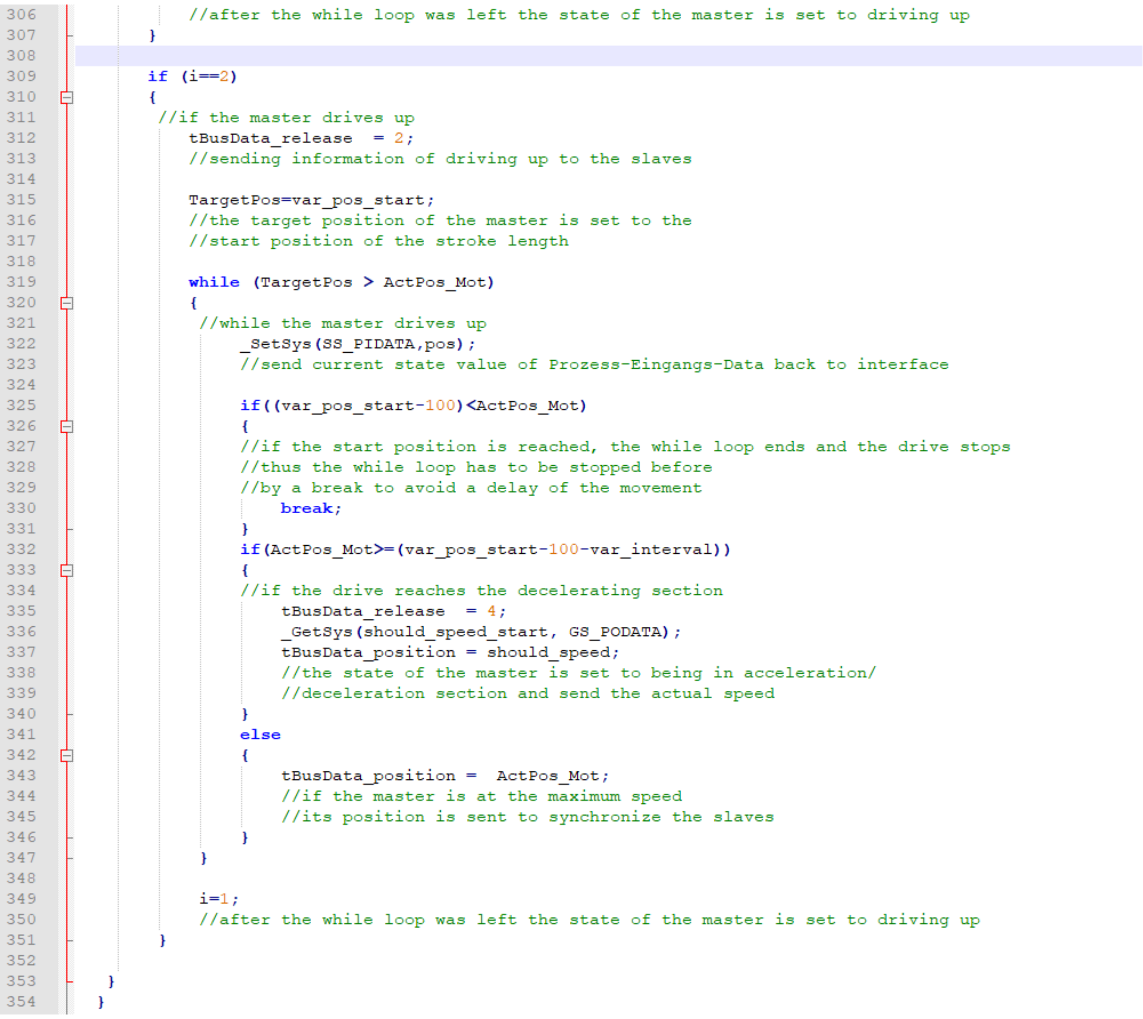
```
   //to control if the SBus communication works:
   //the mutable tBusData_drive_slave parameters are saved and
   //compared to the last sent variable
   if(tBusData_drive_slave1==drives_slave1)
   {
    //if the variable is not changing the control parameter is set to 3
      drive_slave1=3;
   }
   if(tBusData_drive_slave2==drives_slave2)
   {
    //if the variable is not changing the control parameter is set to 30
      drive_slave2=30;
   }
   if(tBusData_drive_slave3==drives_slave3)
   {
    //if the variable is not changing the control parameter is set to 300
      drive_slave3=300;
   }
   drives_slave1=tBusData_drive_slave1;
   drives_slave2=tBusData_drive_slave2;
   drives_slave3=tBusData_drive_slave3;
   //the buffer is cleand


istpos_slave  = drive_slave1+drive_slave2+drive_slave3;
//addition of the three control variables:
//the resulting number contains the status information of the slaves
//each digit of the number gives information of one slave
//meaning of the nummbers:
//0: slave is not moving
//1: slave moves with a wrong shift
//2: slave moves with a right shift
//3: the slave does not send any data
```

Figure A.1.8: Monitoring if the SBus communication works and the slaves are still sending data.



```
if (i==1)
{
 //if the master drives down
    tBusData_release  = 1;
    //sending information of down driving to the slaves

    TargetPos=var_pos_end;
    //the target position of the master is set to the
    //end position of the stroke length

    while (TargetPos < ActPos_Mot)
    {
     //while driving down
        _SetSys(SS_PIDATA,pos);
        //send current state value of Prozess-Eingangs-Data back to interface

        if((var_pos_end+100)>ActPos_Mot)
        {
        //if the end position is reached, the while loop ends and the drive stops
        //thus the while loop has to be stopped before
        //by a break to avoid a delay of the movement
            break;
        }
        if(ActPos_Mot<=(var_pos_end+100+var_interval))
        {
        //if the drive reaches the decelerating section
            tBusData_release  = 3;
            _GetSys(should_speed_start, GS_PODATA);
            tBusData_position = should_speed;
            //the state of the master is set to being in acceleration/
            //deceleration section and send the actual speed
        }
        else
        {
            tBusData_position =  ActPos_Mot;
            //if the master is at the maximum speed
            //its position is sent to synchronize the slaves
        }

    }

    i=2;
    //after the while loop was left the state of the master is set to driving up
}
```

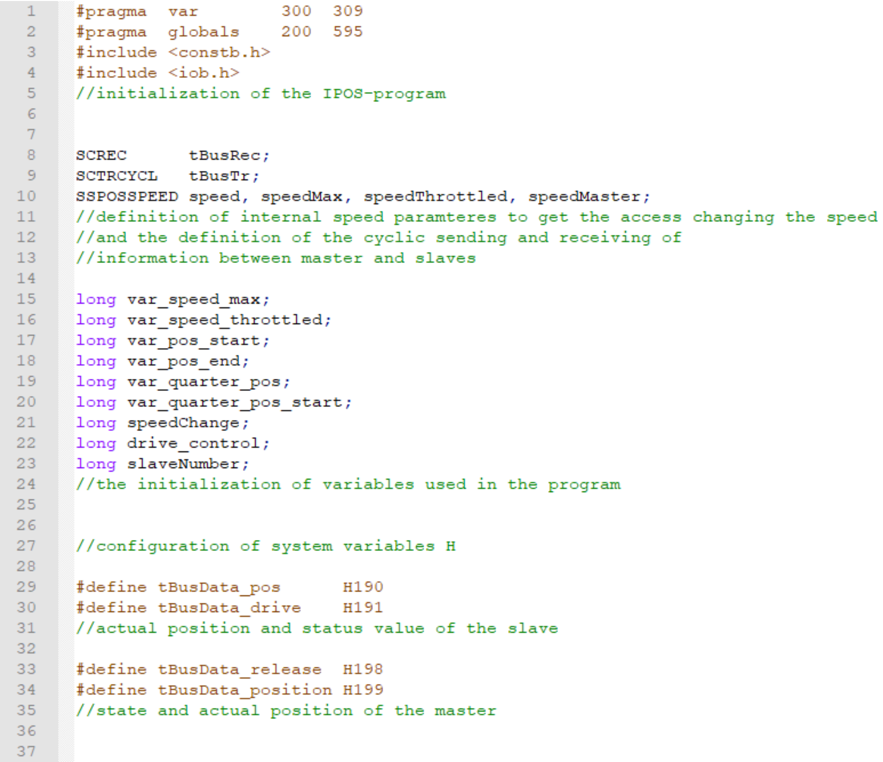
Figure A.1.9: The programming code while the master drives down.



```
         //after the while loop was left the state of the master is set to driving up
     }

     if (i==2)
     {
      //if the master drives up
         tBusData_release  = 2;
         //sending information of driving up to the slaves

         TargetPos=var_pos_start;
         //the target position of the master is set to the
         //start position of the stroke length

         while (TargetPos > ActPos_Mot)
         {
          //while the master drives up
              _SetSys(SS_PIDATA,pos);
              //send current state value of Prozess-Eingangs-Data back to interface

              if((var_pos_start-100)<ActPos_Mot)
              {
              //if the start position is reached, the while loop ends and the drive stops
              //thus the while loop has to be stopped before
              //by a break to avoid a delay of the movement
                  break;
              }
              if(ActPos_Mot>=(var_pos_start-100-var_interval))
              {
              //if the drive reaches the decelerating section
                  tBusData_release  = 4;
                  _GetSys(should_speed_start, GS_PODATA);
                  tBusData_position = should_speed;
                  //the state of the master is set to being in acceleration/
                  //deceleration section and send the actual speed
              }
              else
              {
                  tBusData_position =  ActPos_Mot;
                  //if the master is at the maximum speed
                  //its position is sent to synchronize the slaves
              }
          }

          i=1;
          //after the while loop was left the state of the master is set to driving up
      }

 }
}
```

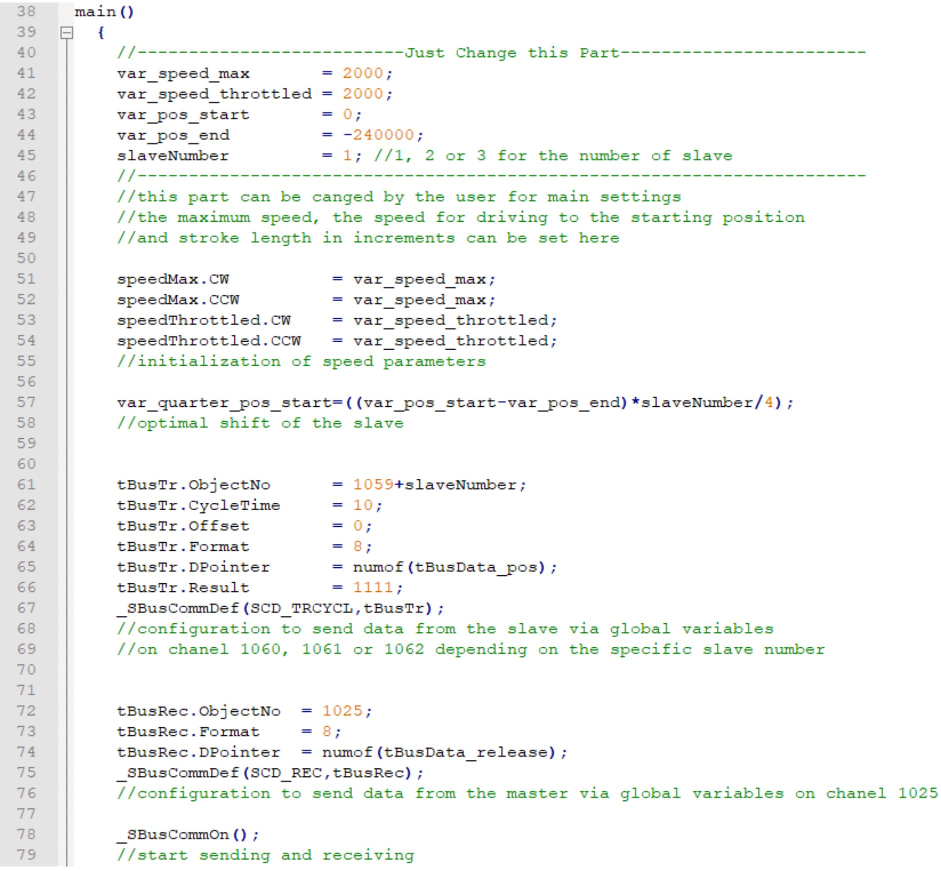
Figure A.1.10: The program code while the master drives up.

A.2 Program code of slave



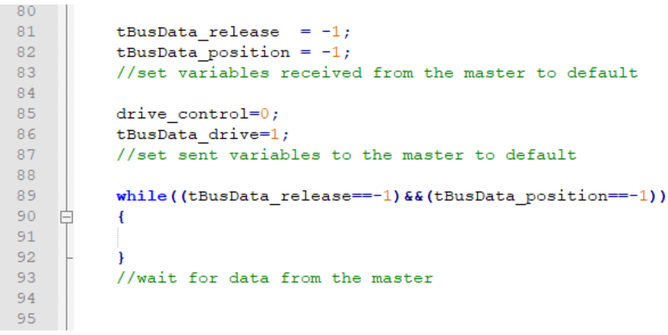
```
#pragma  var        300  309
#pragma  globals    200  595
#include <constb.h>
#include <iob.h>
//initialization of the IPOS-program


SCREC      tBusRec;
SCTRCYCL   tBusTr;
SSPOSSPEED speed, speedMax, speedThrottled, speedMaster;
//definition of internal speed paramteres to get the access changing the speed
//and the definition of the cyclic sending and receiving of
//information between master and slaves

long var_speed_max;
long var_speed_throttled;
long var_pos_start;
long var_pos_end;
long var_quarter_pos;
long var_quarter_pos_start;
long speedChange;
long drive_control;
long slaveNumber;
//the initialization of variables used in the program


//configuration of system variables H

#define tBusData_pos      H190
#define tBusData_drive    H191
//actual position and status value of the slave

#define tBusData_release  H198
#define tBusData_position H199
//state and actual position of the master


```

Figure A.2.1: General initialization of each IPOS program with definition of local and global variables.



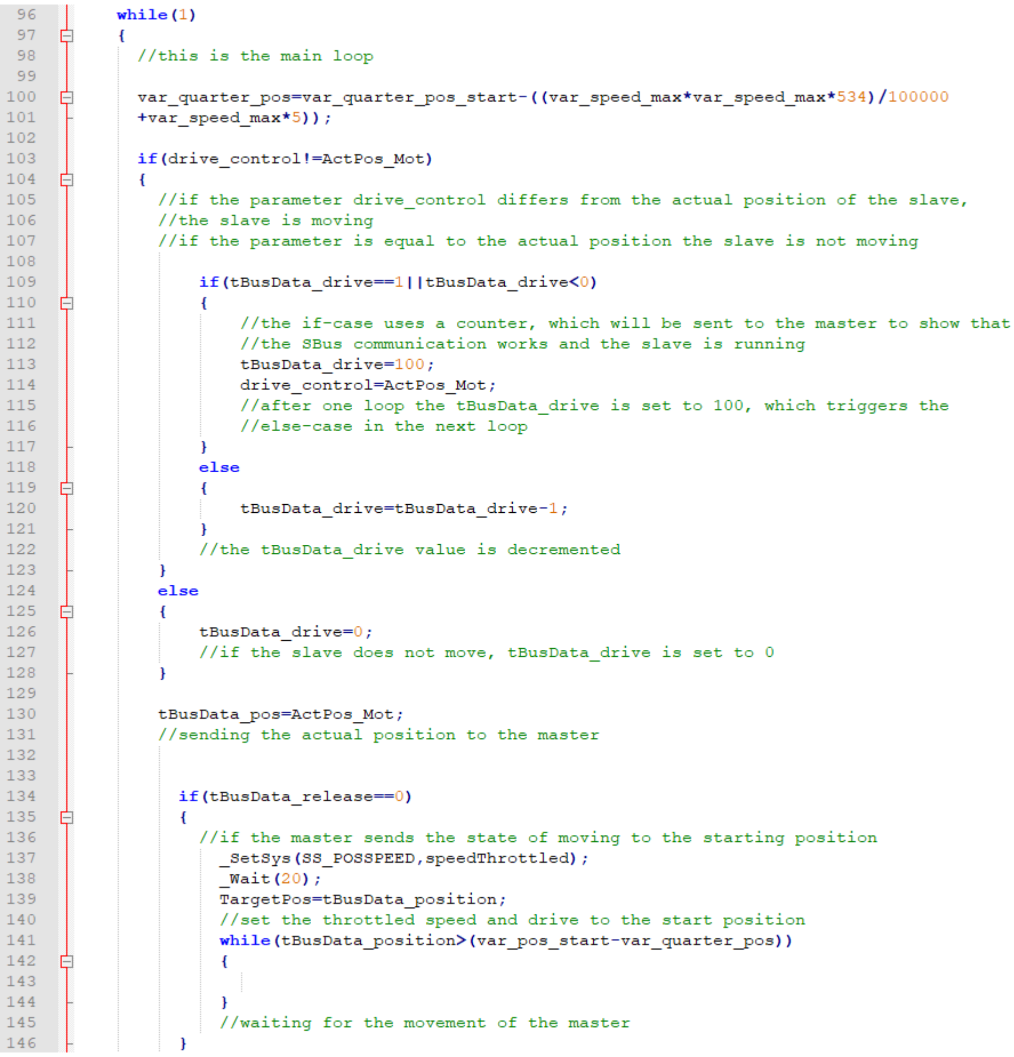
```
main()
  {
    //--------------------------Just Change this Part------------------------
    var_speed_max       = 2000;
    var_speed_throttled = 2000;
    var_pos_start       = 0;
    var_pos_end         = -240000;
    slaveNumber         = 1; //1, 2 or 3 for the number of slave
    //----------------------------------------------------------------------
    //this part can be canged by the user for main settings
    //the maximum speed, the speed for driving to the starting position
    //and stroke length in increments can be set here

    speedMax.CW          = var_speed_max;
    speedMax.CCW         = var_speed_max;
    speedThrottled.CW    = var_speed_throttled;
    speedThrottled.CCW   = var_speed_throttled;
    //initialization of speed parameters

    var_quarter_pos_start=((var_pos_start-var_pos_end)*slaveNumber/4);
    //optimal shift of the slave


    tBusTr.ObjectNo      = 1059+slaveNumber;
    tBusTr.CycleTime     = 10;
    tBusTr.Offset        = 0;
    tBusTr.Format        = 8;
    tBusTr.DPointer      = numof(tBusData_pos);
    tBusTr.Result        = 1111;
    _SBusCommDef(SCD_TRCYCL,tBusTr);
    //configuration to send data from the slave via global variables
    //on chanel 1060, 1061 or 1062 depending on the specific slave number


    tBusRec.ObjectNo  = 1025;
    tBusRec.Format    = 8;
    tBusRec.DPointer  = numof(tBusData_release);
    _SBusCommDef(SCD_REC,tBusRec);
    //configuration to send data from the master via global variables on chanel 1025

    _SBusCommOn();
    //start sending and receiving
```

Figure A.2.2: Initialization of used variables and configuration of the SBus communication channel between master and the slaves.



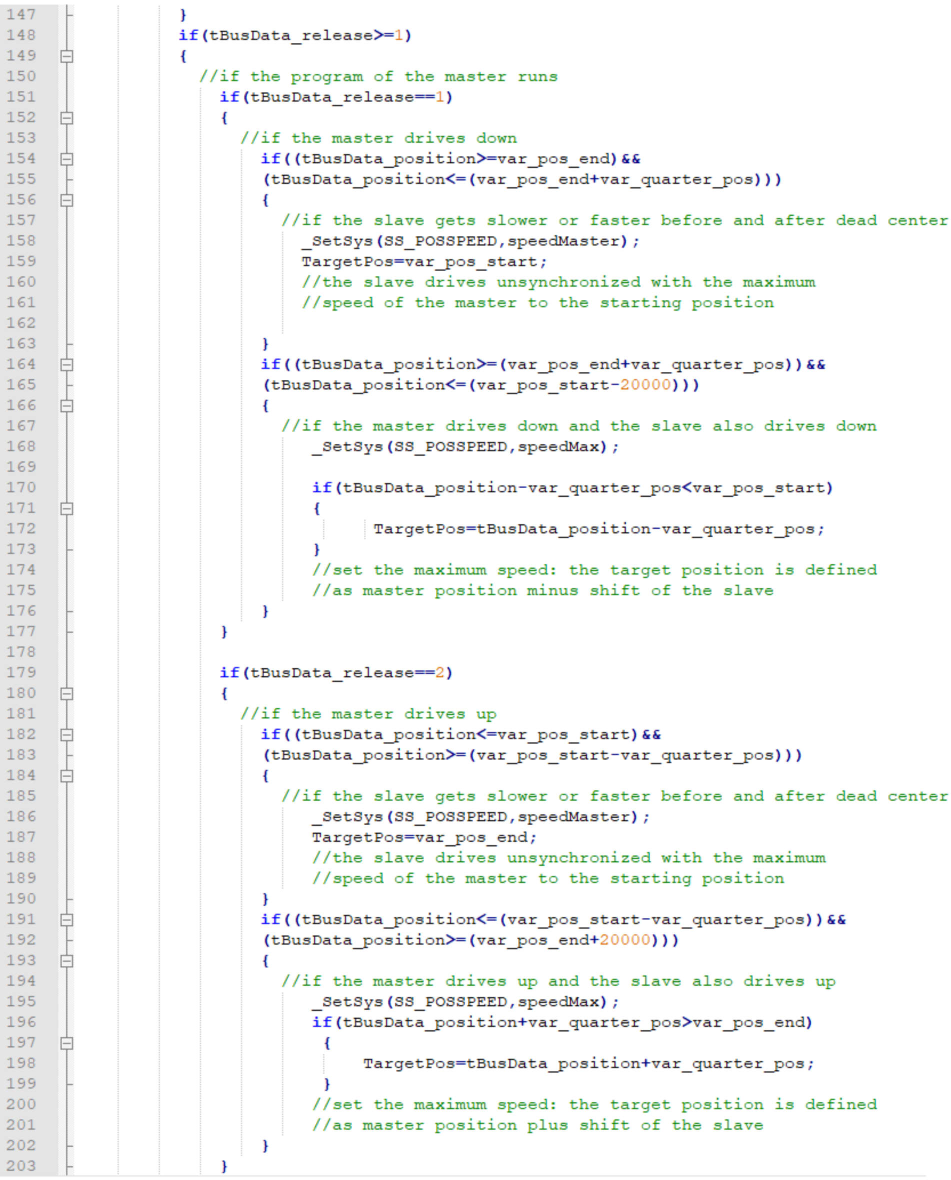
```

    tBusData_release  = -1;
    tBusData_position = -1;
    //set variables received from the master to default

    drive_control=0;
    tBusData_drive=1;
    //set sent variables to the master to default

    while((tBusData_release==-1)&&(tBusData_position==-1))
    {

    }
    //wait for data from the master


```

Figure A.2.3: Setting of all SBus communication variables to default and waiting until receiving data from the master.



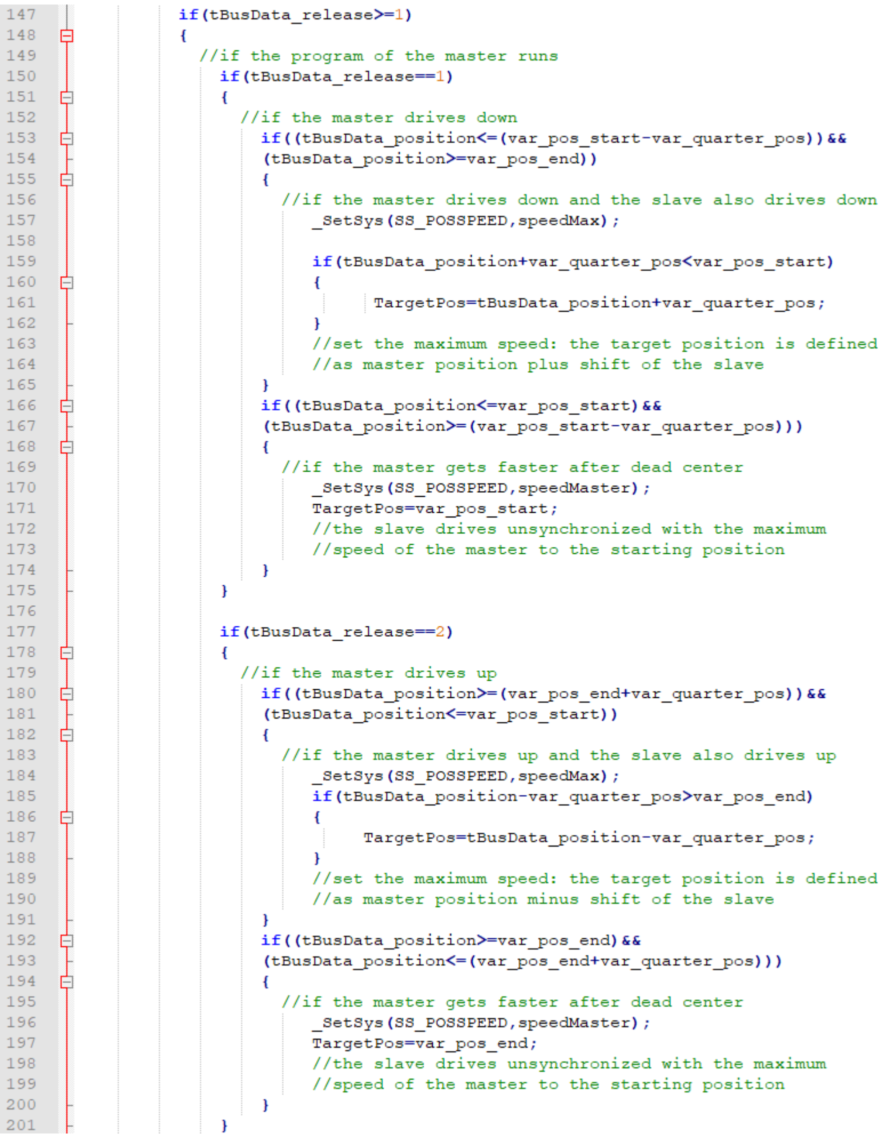
```
while(1)
{
  //this is the main loop

  var_quarter_pos=var_quarter_pos_start-((var_speed_max*var_speed_max*534)/100000
  +var_speed_max*5));

  if(drive_control!=ActPos_Mot)
  {
    //if the parameter drive_control differs from the actual position of the slave,
    //the slave is moving
    //if the parameter is equal to the actual position the slave is not moving

        if(tBusData_drive==1||tBusData_drive<0)
        {
            //the if-case uses a counter, which will be sent to the master to show that
            //the SBus communication works and the slave is running
            tBusData_drive=100;
            drive_control=ActPos_Mot;
            //after one loop the tBusData_drive is set to 100, which triggers the
            //else-case in the next loop
        }
        else
        {
            tBusData_drive=tBusData_drive-1;
        }
        //the tBusData_drive value is decremented
    }
    else
    {
        tBusData_drive=0;
        //if the slave does not move, tBusData_drive is set to 0
    }

    tBusData_pos=ActPos_Mot;
    //sending the actual position to the master


      if(tBusData_release==0)
      {
        //if the master sends the state of moving to the starting position
          _SetSys(SS_POSSPEED,speedThrottled);
          _Wait(20);
          TargetPos=tBusData_position;
          //set the throttled speed and drive to the start position
          while(tBusData_position>(var_pos_start-var_quarter_pos))
          {

          }
          //waiting for the movement of the master
      }
```

Figure A.2.4: The slave information is sent to the master verifying a working SBus connection and a moving slave. The slave moves to the starting position similar to the master.



```
}
if(tBusData_release>=1)
{
  //if the program of the master runs
    if(tBusData_release==1)
    {
      //if the master drives down
        if((tBusData_position>=var_pos_end)&&
        (tBusData_position<=(var_pos_end+var_quarter_pos)))
        {
          //if the slave gets slower or faster before and after dead center
            _SetSys(SS_POSSPEED,speedMaster);
            TargetPos=var_pos_start;
            //the slave drives unsynchronized with the maximum
            //speed of the master to the starting position

        }
        if((tBusData_position>=(var_pos_end+var_quarter_pos))&&
        (tBusData_position<=(var_pos_start-20000)))
        {
          //if the master drives down and the slave also drives down
             _SetSys(SS_POSSPEED,speedMax);

             if(tBusData_position-var_quarter_pos<var_pos_start)
             {
                   TargetPos=tBusData_position-var_quarter_pos;
             }
             //set the maximum speed: the target position is defined
             //as master position minus shift of the slave
        }
    }

    if(tBusData_release==2)
    {
      //if the master drives up
        if((tBusData_position<=var_pos_start)&&
        (tBusData_position>=(var_pos_start-var_quarter_pos)))
        {
          //if the slave gets slower or faster before and after dead center
             _SetSys(SS_POSSPEED,speedMaster);
             TargetPos=var_pos_end;
             //the slave drives unsynchronized with the maximum
             //speed of the master to the starting position
        }
        if((tBusData_position<=(var_pos_start-var_quarter_pos))&&
        (tBusData_position>=(var_pos_end+20000)))
        {
          //if the master drives up and the slave also drives up
             _SetSys(SS_POSSPEED,speedMax);
             if(tBusData_position+var_quarter_pos>var_pos_end)
              {
                  TargetPos=tBusData_position+var_quarter_pos;
              }
             //set the maximum speed: the target position is defined
             //as master position plus shift of the slave
        }
    }
```

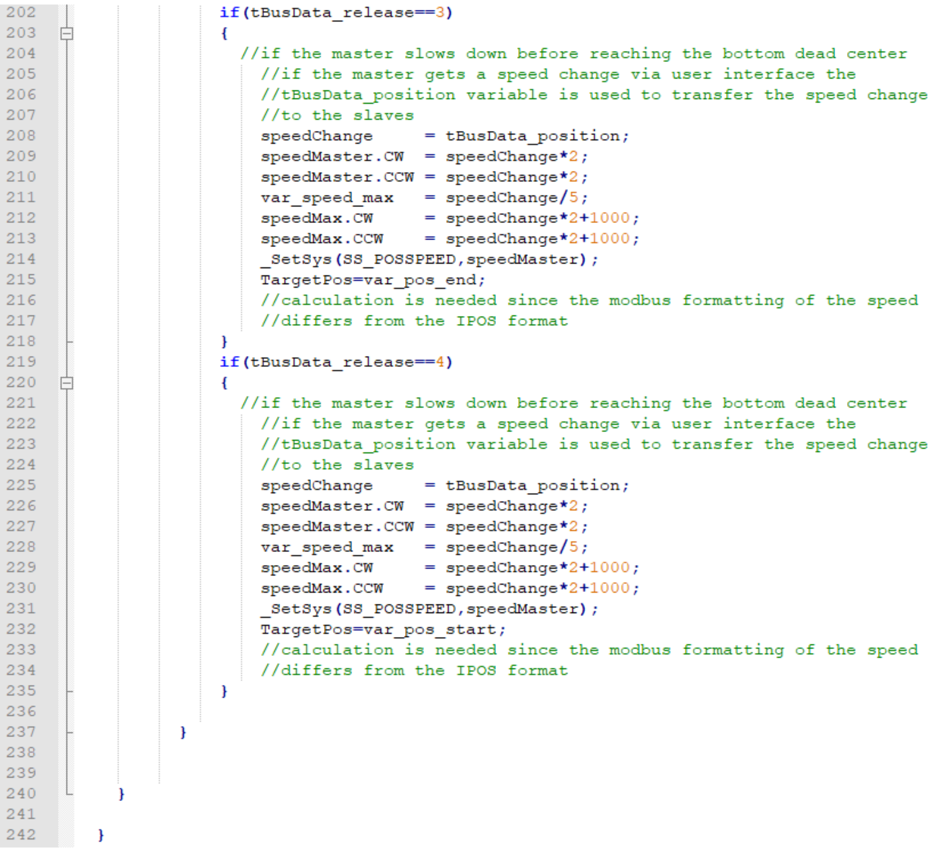
(a) Slave 1.



```
if(tBusData_release>=1)
{
  //if the program of the master runs
    if(tBusData_release==1)
    {
      //if the master drives down
        if((tBusData_position<=(var_pos_start-var_quarter_pos))&&
        (tBusData_position>=var_pos_end))
        {
          //if the master drives down and the slave also drives down
             _SetSys(SS_POSSPEED,speedMax);

             if(tBusData_position+var_quarter_pos<var_pos_start)
             {
                   TargetPos=tBusData_position+var_quarter_pos;
             }
             //set the maximum speed: the target position is defined
             //as master position plus shift of the slave
        }
        if((tBusData_position<=var_pos_start)&&
        (tBusData_position>=(var_pos_start-var_quarter_pos)))
        {
          //if the master gets faster after dead center
             _SetSys(SS_POSSPEED,speedMaster);
             TargetPos=var_pos_start;
             //the slave drives unsynchronized with the maximum
             //speed of the master to the starting position
        }
    }

    if(tBusData_release==2)
    {
      //if the master drives up
        if((tBusData_position>=(var_pos_end+var_quarter_pos))&&
        (tBusData_position<=var_pos_start))
        {
          //if the master drives up and the slave also drives up
             _SetSys(SS_POSSPEED,speedMax);
             if(tBusData_position-var_quarter_pos>var_pos_end)
             {
                  TargetPos=tBusData_position-var_quarter_pos;
             }
             //set the maximum speed: the target position is defined
             //as master position minus shift of the slave
        }
        if((tBusData_position>=var_pos_end)&&
        (tBusData_position<=(var_pos_end+var_quarter_pos)))
        {
          //if the master gets faster after dead center
             _SetSys(SS_POSSPEED,speedMaster);
             TargetPos=var_pos_end;
             //the slave drives unsynchronized with the maximum
             //speed of the master to the starting position
        }
    }
```

(b) Slave 2 and Slave 3.

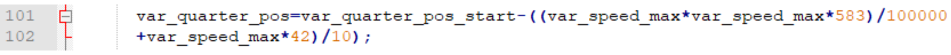
Figure A.2.5: Up and down movement of the slaves: the drive profile contains sections with unsynchronized parts and parts, which are synchronized to the masters position.



```
            if(tBusData_release==3)
            {
              //if the master slows down before reaching the bottom dead center
                //if the master gets a speed change via user interface the
                //tBusData_position variable is used to transfer the speed change
                //to the slaves
                speedChange     = tBusData_position;
                speedMaster.CW  = speedChange*2;
                speedMaster.CCW = speedChange*2;
                var_speed_max   = speedChange/5;
                speedMax.CW     = speedChange*2+1000;
                speedMax.CCW    = speedChange*2+1000;
                _SetSys(SS_POSSPEED,speedMaster);
                TargetPos=var_pos_end;
                //calculation is needed since the modbus formatting of the speed
                //differs from the IPOS format
            }
            if(tBusData_release==4)
            {
              //if the master slows down before reaching the bottom dead center
                //if the master gets a speed change via user interface the
                //tBusData_position variable is used to transfer the speed change
                //to the slaves
                speedChange     = tBusData_position;
                speedMaster.CW  = speedChange*2;
                speedMaster.CCW = speedChange*2;
                var_speed_max   = speedChange/5;
                speedMax.CW     = speedChange*2+1000;
                speedMax.CCW    = speedChange*2+1000;
                _SetSys(SS_POSSPEED,speedMaster);
                TargetPos=var_pos_start;
                //calculation is needed since the modbus formatting of the speed
                //differs from the IPOS format
            }

        }


  }

}
```

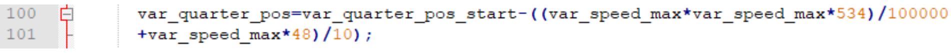
Figure A.2.6: Transfering of speed changes via user interface to the slaves.

A.3 Implemented correction functions



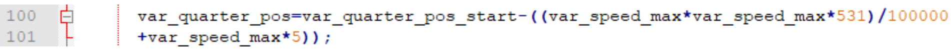
```
var_quarter_pos=var_quarter_pos_start-((var_speed_max*var_speed_max*583)/100000
+var_speed_max*42)/10);
```

Figure A.3.1: Implemented correction function of Slave 1.



```
var_quarter_pos=var_quarter_pos_start-((var_speed_max*var_speed_max*534)/100000
+var_speed_max*48)/10);
```

Figure A.3.2: Implemented correction function of Slave 2.



```
var_quarter_pos=var_quarter_pos_start-((var_speed_max*var_speed_max*531)/100000
+var_speed_max*5));
```

Figure A.3.3: Implemented correction function of Slave 3.

B References

[1] T. X. experiment, “The xenon detection principle.” `http://www.xenon1t.org/`, Jan. 2014.

[2] E. Aprile and T. Doke, “Liquid Xenon Detectors for Particle Physics and Astrophysics,” *Rev. Mod. Phys.*, vol. 82, pp. 2ff., 11f., 27, 2010.

[3] E. Aprile *et al.*, “Removing krypton from xenon by cryogenic distillation to the ppq level,” *Eur. Phys. J.*, vol. C77, no. 5, p. 275ff., 2017.

[4] E. Aprile *et al.*, “Online ^{222}Rn removal by cryogenic distillation in the XENON100 experiment,” *Eur. Phys. J.*, vol. C77, no. 6, p. 358, 2017.

[5] E. Brown *et al.*, “Magnetically-coupled piston pump for high-purity gas applications,” *Eur. Phys. J.*, vol. C78, no. 7, p. 1, 2018.

[6] ESA, “Planck reveals an almost perfect universe.” `http://www.esa.int/Our_Activities/Space_Science/Planck/Planck_reveals_an_almost_perfect_Universe`, Mar 2013.

[7] G. F. Smoot, “Nobel lecture: Cosmic microwave background radiation anisotropies: Their discovery and utilization,” *Rev. Mod. Phys.*, vol. 79, p. 1349, Nov. 2007.

[8] D. Clowe, M. Bradac, A. H. Gonzalez, M. Markevitch, S. W. Randall, C. Jones, and D. Zaritsky, “A direct empirical proof of the existence of dark matter,” *Astrophys. J.*, vol. 648, p. 1, 2006.

[9] F. Zwicky, “On the Masses of Nebulae and of Clusters of Nebulae,” *Astrophysical Journal*, vol. 86, p. 218, Oct. 1937.

[10] K. G. Begeman, A. H. Broeils, and R. H. Sanders, “Extended rotation curves of spiral galaxies - Dark haloes and modified dynamics,” *Monthly Notices of the RAS*, vol. 249, p. 526f., Apr. 1991.

[11] L. Roszkowski, E. M. Sessolo, and S. Trojanowski, “WIMP dark matter candidates and searches—current status and future prospects,” *Rept. Prog. Phys.*, vol. 81, no. 6, p. 6f., 2018.

[12] T. Marrodán Undagoitia and L. Rauch, “Dark matter direct-detection experiments,” *J. Phys.*, vol. G43, no. 1, p. 8, 2016.

[13] A. Boveia and C. Doglioni, “Dark Matter Searches at Colliders,” *Ann. Rev. Nucl. Part. Sci.*, vol. 68, p. 1, 2018.

[14] M. G. Aartsen *et al.*, “Search for Neutrinos from Dark Matter Self-Annihilations in the center of the Milky Way with 3 years of IceCube/DeepCore,” *Eur. Phys. J.*, vol. C77, no. 9, p. 4, 2017.

[15] L. overview, "Gran sasso national laboratory." https://www.lngs.infn.it/en/lngs-overview.

[16] J. B. Albert *et al.*, "Improved measurement of the $2\nu\beta\beta$ half-life of ^{136}Xe with the EXO-200 detector," *Phys. Rev.*, vol. C89, no. 1, p. 1, 2014.

[17] E. Aprile *et al.*, "Observation of two-neutrino double electron capture in ^{124}Xe with XENON1T," *Nature*, vol. 568, no. 7753, p. 1, 2019.

[18] L. Althüser, "Light collection efficiency simulations of the xenon1t experiment and comparison to data," Master's thesis, Westfälische Wilhelms-Universität Münster. Institut für Kernphysik, https://www.uni-muenster.de/Physik.KP/AGWeinheimer/arbeitsgruppe/arbeiten.html, 2017.

[19] R. Harrison and H. ApSimon, "Krypton-85 pollution and atmospheric electricity," *Atmospheric Environment*, vol. 28, no. 4, p. 637, 1994.

[20] C. M. B. Richard B. Firestone and S. Chu, *Table of isotopes*, vol. 4. Wiley-VCH, 8 ed., Aug 1999.

[21] N. Rupp, "Radon background in liquid xenon detectors," *JINST*, vol. 13, no. 02, p. 1, 2018.

[22] C. Hoffmann, "Hoffmanns nuklidkarte." https://www.physiklehrer.net/Nuklidkarte/nuklidkarte.pdf, Mar. 2016.

[23] E. Aprile *et al.*, "Physics reach of the XENON1T dark matter experiment," *JCAP*, vol. 1604, no. 04, p. 10f, 2016.

[24] E. Aprile *et al.*, "Dark Matter Search Results from a One Ton-Year Exposure of XENON1T," *Phys. Rev. Lett.*, vol. 121, no. 11, p. 6, 2018.

[25] D. Schulte, "Status of the radon distillation column for xenonnt." XENON nternal presentation, May. 2019.

[26] E. Aprile *et al.*, "Online ^{222}Rn removal by cryogenic distillation in the XENON100 experiment," *Eur. Phys. J.*, vol. C77, no. 6, p. 358, 2017.

[27] D. Schulte, "Upgrading and characterization of a magnetically-coupled piston pump for ultra-clean noble gas applications," Master's thesis, Westfälische Wilhelms-Universität Münster. Institut für Kernphysik, https://www.uni-muenster.de/Physik.KP/AGWeinheimer/arbeitsgruppe/arbeiten.html, 2018.

[28] SEW, https://download.sew-eurodrive.com/download/pdf/11320400.pdf, *Positionierung und Ablaufsteuerung IPOS*$^{\text{plus}}$®, Nov. 2004.

[29] SEW, https://download.sew-eurodrive.com/download/pdf/11483407.pdf, *MOVIDRIVE® MDX60B/61B*, Sep. 2006.

[30] SEW, `https://download.sew-eurodrive.com/download/pdf/22138595.pdf`, *Betriebsanleitung CMS..50 – 71*, May. 2015.