Data acquisition for an SiPM based muon detector

Von der Fakultät für Mathematik, Informatik und Naturwissenschaften der
RWTH Aachen University zur Erlangung des akademischen Grades eines
Doktors der Naturwissenschaften genehmigte Dissertation

vorgelegt von

Lukas Middendorf
M.Sc. RWTH

aus

Bocholt

Berichter: Univ.-Prof. Dr. rer. nat. Thomas Hebbeker
apl. Prof. Dr. rer. nat. Stefan Roth

Tag der mündlichen Prüfung: 29.05.2018

Abstract

To measure cosmic rays at the highest energies, ground based detectors sparsely covering an area of several thousand square kilometers to detect cosmic ray induced air showers are used. One of those experiments is the Pierre Auger Observatory, which is currently being upgraded for a better determination of the muonic component in the showers to allow a better identification of the type of primary particle. For this upgrade multiple different detector concepts and designs were proposed. One of those proposals is the Aachen Muon Detector (AMD), which is a tiled scintillator detector with silicon photomultipliers (SiPMs) as light sensors. To digitize the SiPM signals the EASIROC chip is used, which allows to read out 32 SiPMs and adjusting of their supply voltage with a low power consumption. The complete readout electronics, including the power supply unit as well as the firmware and software needed for operation, are described in this thesis. Furthermore, characterization measurements of the electronics and first measurements with the detector system are presented.

Zusammenfassung

Um die kosmische Strahlung bei den höchsten Energien zu messen, werden bodengestützte Detektoren verwendet. Sie decken Flächen von einigen tausend Quadratkilometern ab, um durch kosmische Strahlung induzierte Luftschauer zu detektieren. Eines dieser Experimente ist das Pierre Auger-Observatorium, das derzeit für eine bessere Bestimmung der myonischen Komponente in den Schauern erweitert wird, um eine verbesserte Identifizierung der Art des Primärteilchens zu ermöglichen. Für dieses Upgrade wurden mehrere verschiedene Detektorkonzepte und Designs vorgeschlagen und untersucht. Einer dieser Vorschläge ist der Aachen Muon Detector (AMD), ein kachelbasierter Szintillatordetektor mit Silizium-Photomultipliern (SiPMs) als Lichtsensoren. Zur Digitalisierung der SiPM-Signale wird der EASIROC-Chip verwendet, der es erlaubt, 32 SiPMs auszulesen und deren Versorgungsspannung mit geringem Stromverbrauch anzupassen. In dieser Arbeit wird die komplette Ausleseelektronik des AMD einschließlich der Spannungsversorgung und der für den Betrieb notwendigen Firmware und Software beschrieben. Weiterhin werden Charakterisierungsmessungen der Elektronik und erste Messungen mit dem Detektorsystem vorgestellt.

Contents

1. Introduction

The atmosphere of the earth is constantly bombarded by a multitude of cosmic ray particles. Victor Hess first discovered this fact during his now famous balloon flights in 1912. This flux of cosmic ray particles consists of various different types of particles, most prominently photons, electrons, positrons and nuclei including protons. When traversing the atmosphere, the particles initiate a cascade of secondary particles upon colliding with a nucleus of the air. These cascades of particles, called extensive air showers, can be observed by ground-based observatories. This is done either by detecting the secondary particles reaching the ground, or by observing their longitudinal development through fluorescence light, which is emitted when excited nitrogen molecules in the path of the shower deexcite to their ground state. The arrival directions and energies of the primary particles as well as their particle types can be deduced from the shower observables. The largest observatory for extensive air showers induced by ultra high energy cosmic rays (UHECRs) at energies above around $10^{16.5}$ eV is the Pierre Auger Observatory in the Pampa Amarilla in Argentina. Here both observation methods (Fluorescence Detector (FD) and Surface Detector (SD)) are combined for optimal precision and cross calibration.

Even though there are various models and theories of particle acceleration in astrophysical sources, no clear source of UHECRs has been identified until today. Most exotic top-down scenarios, where the decay of e.g. relic particles from the big bang might produce ultra-high energy cosmic rays, are by now severely constrained by the lack of an observation of photons at ultra-high energies. The recent observation of a dipole anisotropy in the arrival direction of cosmic rays at the highest energies by the Pierre Auger Observatory is a strong indication of an extragalactical source [31]. Due to their deflection in galactic and intergalactic magnetic fields, the arrival direction of charged cosmic rays does not directly point back to their point of production. At the highest energies of around 10^{20} eV however, at least for particles with only a small charge (mainly protons), the deflections are sufficiently small to attempt source identification.

To improve its separation between the different types of primary particles, the Surface Detector Pierre Auger Observatory is currently in the process of being upgraded. A scintillator module called Scintillating Surface Detector (SSD) is placed on top of each detector station as a second detector type to better distinguish between secondary electrons and muons. Different detector upgrades were proposed with the Aachen Muon Detector (AMD) being one of them.

The Aachen Muon Detector is a scintillator detector placed under the Auger SD stations and read out by silicon photomultipliers (SiPM). The digitization of the signals is performed by an application specific integrated circuit (ASIC) for SiPM readout called *EASIROC*, which is optimized for low power applications. To relay and buffer the digitized signals and to do triggering, a field programmable gate array (FPGA) is needed and has to be programmed.

Even though the AMD was not selected as the final upgrade for the Pierre Auger Observatory, work on the AMD prototype was continued to understand the behavior of the detector components and to apply the developed techniques to other detector designs. The performance of the detector parts is characterized and evaluated to confirm applicability for the foreseen application. One detector design that is based on the AMD components is the MiniAMD which uses an only slightly modified DAQ with a heavily modified mechanical design. It is a lightweight, flexible and modular muon detector which will be used for characterization measurements and quality control of detector modules for the SSD.

You are currently reading chapter 1. In chapter 2 a short overview over cosmic rays is given, including an introduction to extensive air showers in section 2.2. In chapter 3 the Pierre Auger Observatory and its upgrade *AugerPrime* is presented. This is followed by a short introduction to the device technologies used in this thesis for the construction of the AMD: SiPMs in chapter 5, FPGAs in chapter 6, and the EASIROC ASIC in chapter 7. Then the Aachen Muon Detector is introduced (chapter 8) with a special focus on its DAQ system and the FPGA firmware (chapter 9). Finally characterization measurements of the DAQ and first measurements with the complete AMD readout chain to validate the AMD performance are presented in chapter 10.

2. Cosmic rays

Cosmic ray particles constantly hit our atmosphere. This was first discovered by Victor Hess during his famous balloon flights in 1912 [44]. These particles have a broad energy range from a few MeV up to at least a few 10^{20} eV [21]. With rising energy, the flux of the particles decreases steeply and covers more than 20 orders of magnitude. The charged cosmic rays mainly consist of charged nuclei and electrons. The neutral part of the cosmic rays is dominated by photons, neutrinos and neutrons.

It is not possible to cover the complete span of energies and flux of particles with a single detector or experiment. Particles up to an energy of about 10^{15} eV, which have a high flux and therefore need only small detectors to reach a sufficient event rate, can be detected directly by satellites or balloon experiments. At higher energies, larger detectors on the ground with areas up to multiple thousand square kilometers, observing cosmic ray induced air showers, are used. While this allows a much greater effective detector area, the energy and particle type reconstruction becomes very challenging.

This chapter is only a very short introduction with a special focus on the highest energies. For a more complete introduction, see for example [25].

2.1. Nature and origin

2.1.1. Energy spectrum

The flux of cosmic rays decreases from $\sim 10^3\,\mathrm{m^{-2}s^{-1}}$ at a few GeV to $\sim 1\,\mathrm{km^{-2}}$ per century at 10^{20} eV [25]. The energy spectrum of cosmic rays mostly follows a broken power law

$$\frac{\mathrm{d}N}{\mathrm{d}E} \propto E^{-\gamma} \quad . \tag{2.1}$$

Broken means that the spectral index γ changes slightly at specific energies, which are given special names, breaking the spectrum into segments. The first break occurs at the *knee* at $E \approx 6 \cdot 10^{15}$ eV. Below, the spectral index is approximately $\gamma = 2.7$, at higher energies it is $\gamma \approx 3.1$. A slight further steepening occurs at the *second knee* at $E \approx 4 \cdot 10^{17}$ eV, while at the *ankle* at $E \approx 4 \cdot 10^{18}$ eV, the spectrum flattens again with a spectral index $\gamma \approx 2.7$. The energy of $E \approx 5 \cdot 10^{19}$ eV marks the onset of an even more rapid decline [82]. The spectrum is shown in figure 2.1, with the flux multiplied by $E^{2.6}$ to enhance visibility of the aforementioned features.

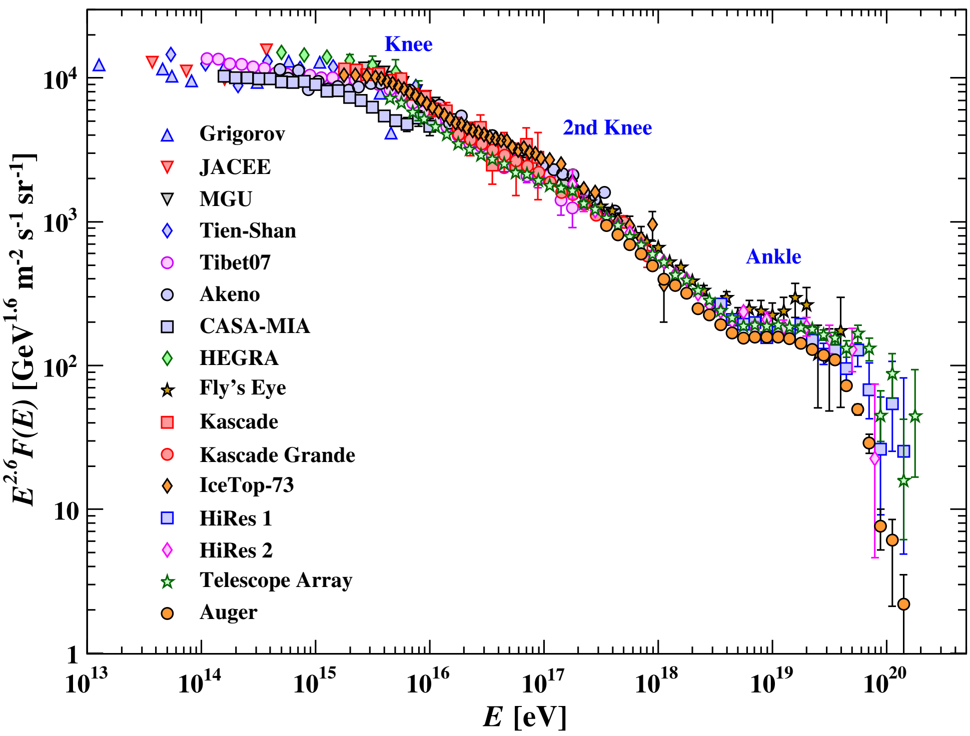


Figure 2.1.: The cosmic ray flux spectrum (all particle types) as a function of the particle energy from air shower measurements. Multiplied by $E^{2.6}$ to enhance visibility of features. Original and references in [63, chap 28].

There are different proposed explanations for the knee and the ankle, for example the transition from galactic to extragalactic cosmic ray sources, propagation effects or features in single sources. The two knees can be explained by the same effect occurring at different energies for different particles, e.g. at the same energy per nucleon for protons and iron nuclei. The final drop off coincides with the predicted GZK[1] effect (see section 2.1.4), but might also originate in an energy limit of the sources of cosmic rays.

2.1.2. Production mechanisms

Even though the production mechanisms and sources of ultra-high energy cosmic rays (UHECRs) are still unknown, there are many theories to explain their creation.

The currently favored models are acceleration models [53]. According to most of these models, acceleration of charged particles occurs in multiple steps. The so-

[1] Greisen-Zatsepin-Kuzmin

called *shock acceleration* occurs when particles are scattered in and around relativistic shock waves by turbulent magnetic fields. For each passing through the shock wave, the particles gain an energy of

$$\frac{\Delta E}{E} \propto \beta \quad , \tag{2.2}$$

where E is the current particle energy, ΔE is the energy gain and $\beta = v/c$ is the velocity of the shock wave. Multiple acceleration passes result in a high particle energy, which quite naturally follows a power law spectrum in accordance with the observed energy spectrum. Candidates for this form of acceleration are *Active Galactic Nuclei* (AGNs) and their jets and hot spots, *Gamma Ray Bursts* (GRBs) and shocks in the intergalactic medium (IGM). There are however still many uncertainties especially regarding the exact configuration of the magnetic fields, leading to a continuing evolution of acceleration models.

In other models rapidly changing magnetic fields, for example near young neutron stars (pulsars), can produce electric fields strong enough to accelerate charged particles to the highest energies in one shot.

Also “top-down” models, which explain the generation of UHECRs through the decay of ultra-heavy exotic particles produced in the big bang (e.g. [18], for a broad overview see [19]), were proposed. These models however also predict a high UHE photon flux, which has not been observed, and therefore are now strongly disfavored by the current limits on UHE photons [22].

2.1.3. Source candidates

Not all relativistic shock waves can accelerate particles to above 10^{20} eV. During acceleration, the magnetic fields in the source region have to keep the particles contained. The maximum possible energy E_{max} of a particle can be estimated [45]:

$$\frac{E_{\mathrm{max}}}{10^{15}\,\mathrm{eV}} \approx \frac{1}{2} \cdot \beta \cdot Z \cdot \frac{B}{\mu\mathrm{G}} \cdot \frac{R}{\mathrm{pc}} \tag{2.3}$$

Here Z is the charge of the particle, β is the velocity of the shock wave, B is the magnetic field strength and R the size of the source region.

This relation is visualized by the so-called *Hillas plot* [45], which is shown as an updated[2] version in figure 2.2. With it size and magnetic field of possible accelerators of cosmic rays are compared. Diagonal lines indicate constant possible E_{max}, which are obviously different for different particle types. A confirmation that the measured UHECRs originate in any of the proposed sources is still pending. The recent observation of a dipole anisotropy in the arrival direction of cosmic rays at the highest energies by the Pierre Auger Observatory is a strong indication of an extragalactical source [31].

[2] Knowledge about magnetic fields in the universe advanced in the last decades.

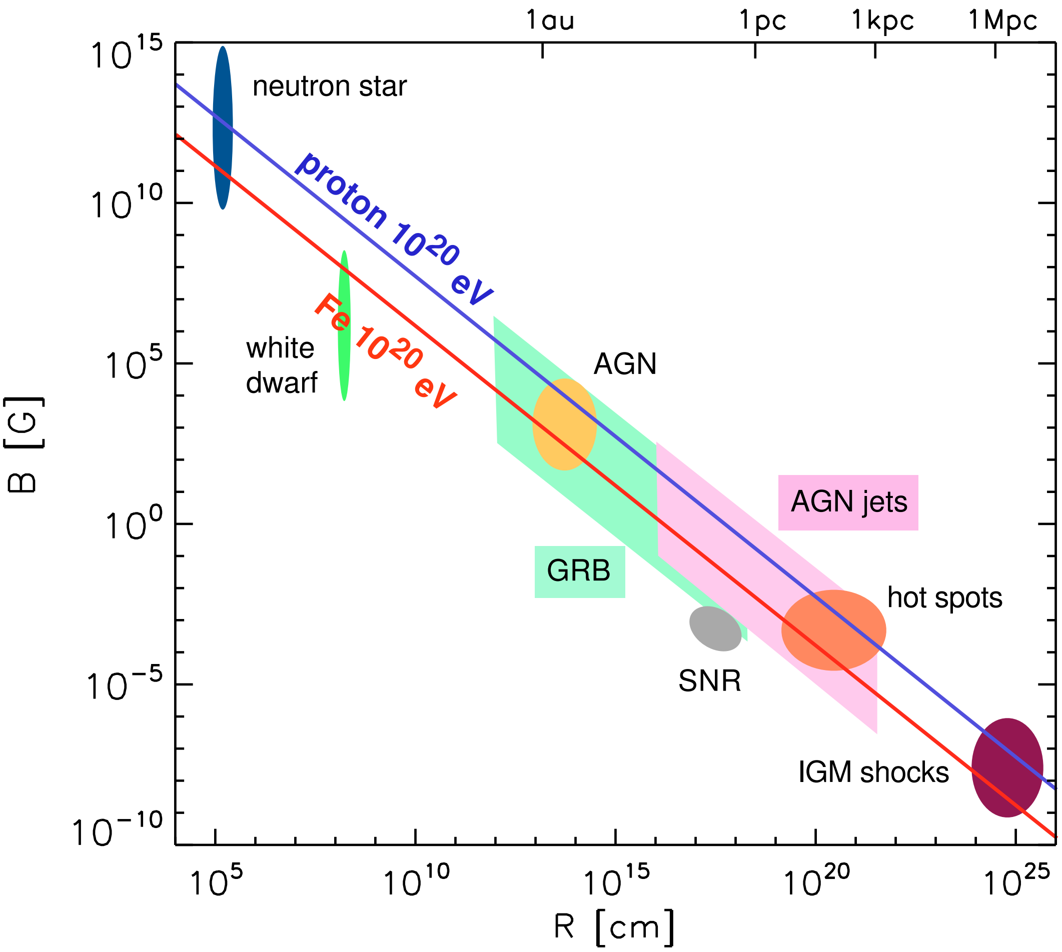


Figure 2.2.: An updated *Hillas plot* from [53]. Diameter R and magnetic field strength B of cosmic ray accelerator candidates are given with their uncertainties. Objects below the diagonal lines can not contain the specified particle during acceleration.

2.1.4. Propagation

It is not possible to directly infer the sources of charged cosmic rays from their arrival directions, since these particles are strongly deflected in the galactic and extragalactic magnetic fields. The gyroradius r of a charged particle being deflected in a magnetic field is proportional to E/Z, with the particle energy E and its charge Z. This leads to different deflections for different energies and particle types, and together with the non-homogeneity of the magnetic fields to an almost perfectly isotropic distribution of arrival directions below some 10^{19} eV. Only uncharged particles (photon, neutrino and neutrons), for which very stringent observational limits exist at energies above 10^{18} eV [22], are not affected by deflections in magnetic fields. For protons at energies above a few 10^{19} eV, it might be possible to observe a significant correlation between the sources of UHECRs and their arrival directions in the future [4].

In addition to deflection, also energy loss and a change of particle type can occur during propagation. Unstable particles like muons and neutrons can decay. The Hubble expansion of the universe causes an adiabatic energy loss. Due to interactions with background photons γ_B from star light or the cosmic microwave background (CMB) heavy nuclei can experience photodisintegration. Ultra high energy photons γ can undergo pair production:

$$\gamma + \gamma_B \rightarrow e^+ + e^- \tag{2.4}$$

For protons the GZK[3] effect ([40] and [85]) is the name for the resonant pion production with photons of the cosmic microwave background:

$$p + \gamma_{CMB} \rightarrow \Delta^+(1232) \rightarrow n + \pi^+ \tag{2.5}$$

$$\rightarrow p + \pi^0 \tag{2.6}$$

The resulting proton or neutron has a lower energy than the original proton. The so-called GZK cutoff is expected to occur at energies at which the production of $\Delta^+(1232)$ becomes possible. At lower energies energy loss due to direct e^+e^- pair production

$$p + \gamma_{CMB} \rightarrow p + e^+ + e^- \tag{2.7}$$

can occur [24].

The main energy loss effect for electrons is synchrotron radiation in magnetic fields, which at ultra high energies reduces their energy rapidly and also prevents their direct acceleration.

2.1.5. Composition

While at lower energies a variety of different particle types is observed, at ultra high energies only charged nuclei have been observed until now. Other particle types

[3] Greisen-Zatsepin-Kuzmin

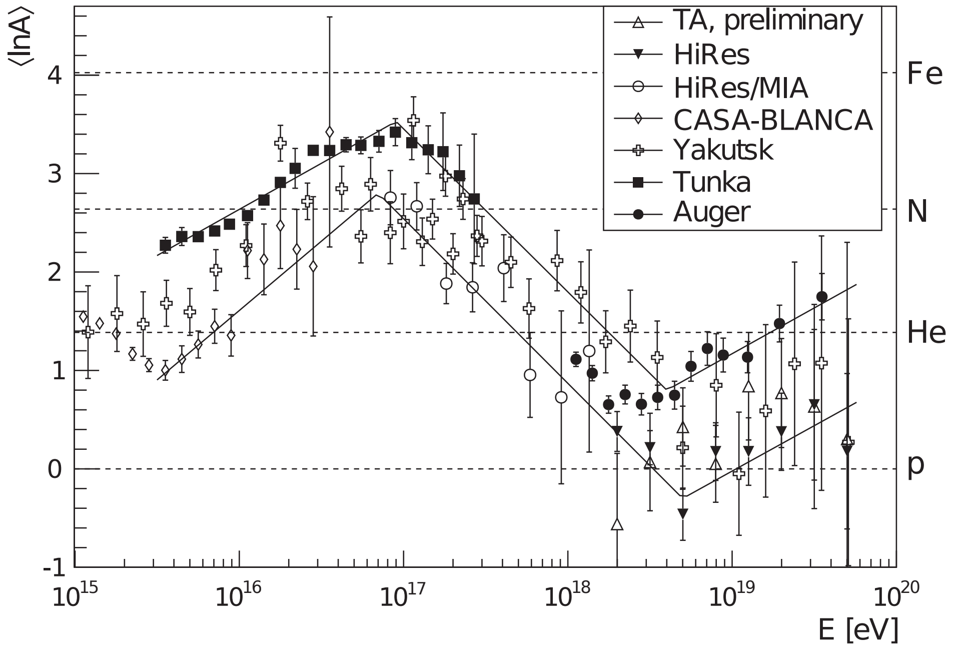


Figure 2.3.: Average logarithmic mass number of cosmic rays as a function of energy derived from measurements with fluorescence telescopes. Simulations using the hadronic interaction model QGSJetII have been used to convert the measured $X_{\max}$ values (see section 2.2.1) to atomic masses. Different interaction models change the absolute mass number scale but not the general trend. The lines are an estimate of the experimental systematic uncertainties. Taken from [48].

either can not reach the earth at those energies (electrons) or have very stringent limits on their flux (neutrons [72], photons and neutrinos [22]).

In figure 2.3 the evolution of the average mass number of the cosmic rays at energies above 10^{15} eV can be seen. Above the knee the composition becomes heavier, while above the second knee it becomes lighter only to become heavier again above the ankle. These measurements are compatible with a particle-dependent cutoff occurring for protons at a few 10^{15} eV and for iron at higher energies of about 10^{17} eV. At the energy cutoff at about 10^{20} eV the measurement uncertainties are still too large to draw a conclusion.

2.2. Cosmic ray induced air showers

When cosmic rays enter the atmosphere, the interactions result in *extensive air showers* (EAS). This fact was first discovered in 1938 by Pierre Auger through measurements of a high coincidence rate between particle detectors up to 300 meters apart, which can be explained by multiple particles from the same EAS triggering the different detectors [17].

During the first interaction, mostly with nuclei of nitrogen or oxygen, multiple secondary particles are produced which do further interactions themselves. The increasing number of particles from the secondary interactions form a cascade of particles, the extensive air shower. The secondary particles still travel roughly in the same direction as the primary particle at a speed close to the speed of light, which results in a slightly curved disc of particles with a thickness in the order of one meter and a lateral extension of a few kilometers when reaching the ground [8]. The exact dimensions of the shower, like its lateral distribution of particles and the curvature of the shower front, depend on the energy and type of the primary particle and undergo statistical fluctuations.

To characterize and describe the longitudinal development of extensive air showers, not a usual length or depth in the atmosphere is used but the slant depth X, which is the integral over the traversed air density, given in units of $\mathrm{g\,cm^{-2}}$:

$$X(x) = \int_x^\infty \rho(x')\,\mathrm{d}x' \tag{2.8}$$

where x is the (conventional) distance from the ground and ρ is the air density. It has to be taken into account that the integral follows the path of the shower, which in general is not vertical and leads to a slant depth which for inclined showers is bigger than for vertical showers at the same height above ground.

The secondary particles of a shower can be divided into hadronic, electromagnetic ($\mathrm{e}^\pm$ and γ) and muonic components (see figure 2.4) .

2.2.1. Hadronic component

Especially for hadronic primary particles, the interactions with air nuclei produce mainly charged and uncharged mesons (pions, kaons) and nucleons (protons, neutrons). The secondary hadronic particles can either do further scattering processes with nuclei or decay (e.g. $\pi^0 \rightarrow \gamma + \gamma$), which often leads to further contributions to other shower components.

The Gaisser-Hillas-Formula [39] is often used to phenomenologically parameterize the longitudinal evolution of the number of shower particles (for $X > X_1$):

$$N(X) = N_\mathrm{max} \left(\frac{X - X_1}{X_\mathrm{max} - X_1} \right)^{\frac{X_\mathrm{max} - X_1}{\lambda}} \exp\left(-\frac{X - X_\mathrm{max}}{\lambda} \right) \tag{2.9}$$

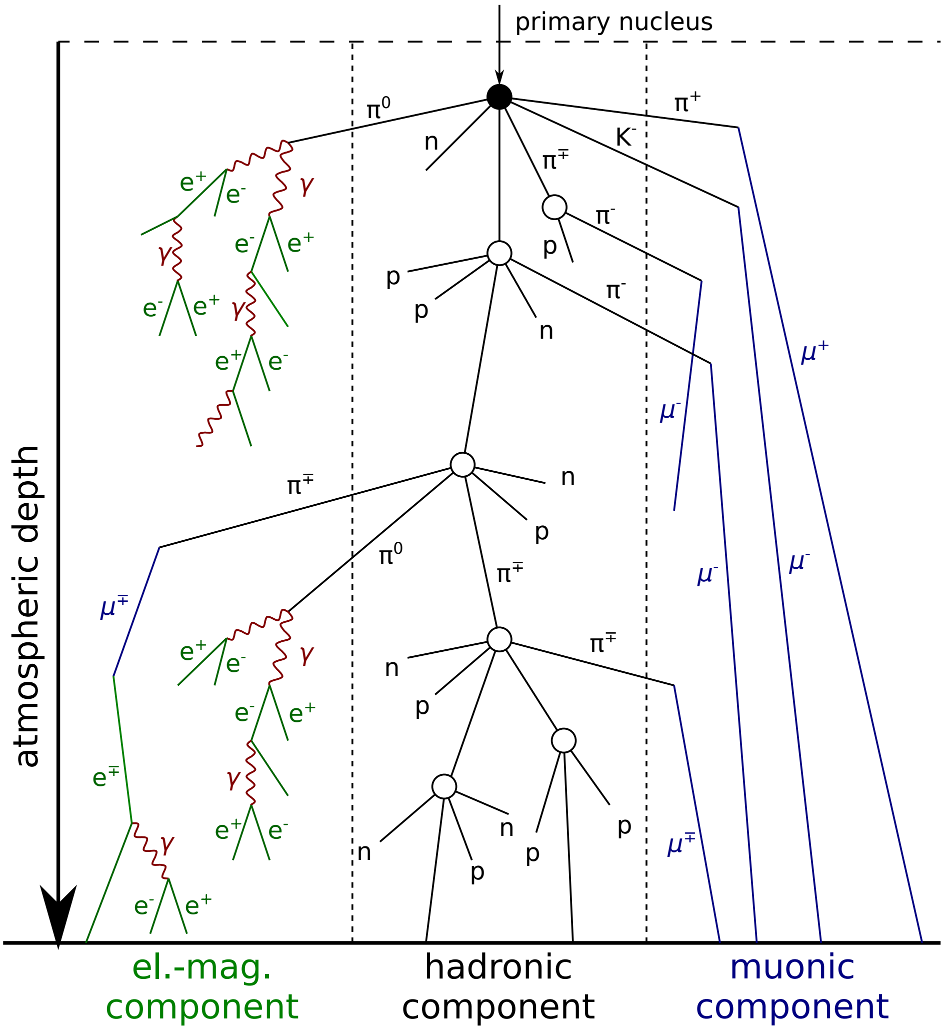


Figure 2.4.: Schematic view of the components in an extensive air shower. The spatial separation is just for illustrative purposes.

Here X_{max} is the slant depth of the shower maximum where N_{max} secondary particles exist, X_1 is the slant depth of the first interaction. λ appears as an effective free path length (slant depth), it is actually just a shape parameter without a clear physical equivalent. Up to X_{max} the energy of the particles is distributed onto an increasing number of particles, but at X_{max} the critical energy for the particles is reached, particles are being stopped and the shower dies out. X_{max} can be used as a mass-sensitive observable. Heavier primary nuclei on the one hand have a bigger cross section for the first interaction and therefore (on average) a smaller X_1. On the other hand they produce more particles (each with lower energy) in the first interaction, resulting in a shower development similar to multiple simultaneous lower energy proton-induced showers after the first interaction.

2.2.2. Electromagnetic component

The electromagnetic component of extensive air showers develops mainly from photons resulting from the decay of neutral pions, but to a lesser degree also electrons produced by decaying muons or particles scattering directly with the electrons of the air atoms. Through this the electromagnetic component is permanently renewed by and coupled to the hadronic component. The photons in the electromagnetic component mainly undergo pair production of $\mathrm{e}^+\mathrm{e}^-$ pairs, which then produce further photons due to bremsstrahlung. Also contributions to the hadronic component are possible with photonuclear interactions. Through an excitation of nitrogen molecules in the air, fluorescence light with an intensity proportional to the energy deposit and therefore to the number of $\mathrm{e}^{\pm}$ is produced [15].

2.2.3. Muonic component

Charged mesons $\mathrm{K}^{\pm}$ and $\pi^{\pm}$ mostly decay into muons and the corresponding neutrinos. This leads to a muonic component which is strongly coupled to the hadronic component. The muons undergo only limited scattering in the atmosphere and at their relativistic velocities also have a lifetime long enough to often reach the ground without decaying into electrons. Due to this, their trajectories often point back directly to their point of production and they are usually the first particles of the shower to arrive at the ground.

The total number of muons produced depends on the mass number A of the primary with energy E_0. Again, for higher mass numbers the showers can be seen as a superposition of A proton-induced showers, each with an energy of $E = E_0/A$.

This can be made plausible by assuming a simplified Heitler-like model, in which the hadronic cascade continues until the charged particles decay into muons when they reach the critical energy E_c. More cascade steps

$$N_{\mathrm{steps}}(E) = \log_N(E/E_c)$$

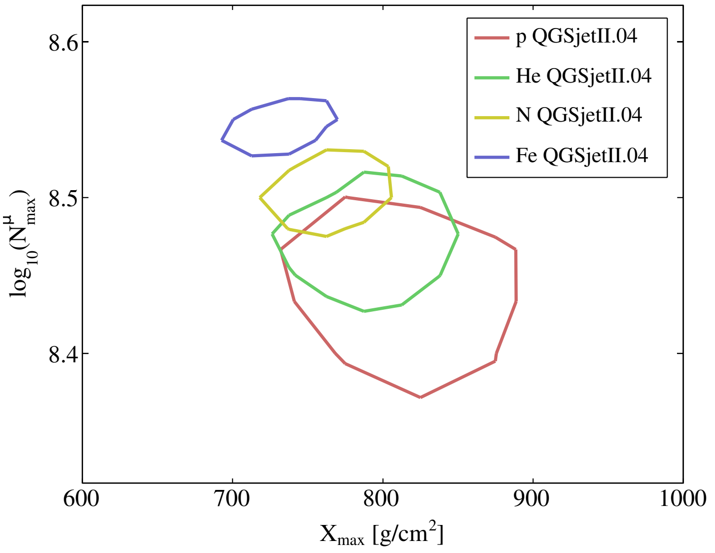


Figure 2.5.: Simulated 1σ contour lines of the distributions of $X_{\max}$ and $N^{\mu}_{\max}$, which is the number of muons at the maximum of the muonic shower development, for different primary particles. Air showers induced by primary particles with $E = 5 \times 10^{19}$ eV and zenith angle $\theta = 38°$. Taken from [4].

(where N is the number of particles produced in each step) happen for higher energies. For each interaction in the hadronic cascade, a fraction p_0 of the energy is carried by π^0, which decay into photons and are therefore removed from the hadronic cascade and can no longer produce muons, while only the fraction $p = 1 - p_0$ of the energy continues in the cascade. It can be deduced that

$$N_\mu \propto E_0 \cdot p^{N_{\text{steps}}(E)} \quad ,$$

which means that the number of muons N_μ increases slower than linear with the primary proton energy $E = E_0/A$ and therefore primaries with a higher mass number A produce a higher number of muons

$$N_\mu \propto E_0 \cdot p^{N_{\text{steps}}(E_0/A)} \quad .$$

As can be seen from the air shower simulations shown in figure 2.5, the number of muons can be used to discriminate different primary particle types even better than $X_{\max}$, with a combination of both observables preferable.

2.2.4. Neutrino component

During decay and production of muons also neutrinos are produced to balance the lepton number. Even though they can not normally be measured, the fact that they withdraw energy from the shower (*invisible energy*) has to be corrected for.

2.2.5. Detection principles

Extensive air showers can be detected using different methods. The most important principles are the detection of the fluorescence light produced by the electromagnetic component, through which the longitudinal shower development can be observed, and the measurement of the lateral distribution of particles at the ground level.

2.2.5.1. Fluorescence light detection

Fluorescence telescopes are used to observe the longitudinal shower development by measuring fluorescence light. The fluorescence light produced by nitrogen molecules in the air is strongly related to the energy deposit by the electromagnetic component. It therefore allows a calorimetric energy measurement if the fluorescence light yield and the amount of invisible energy" is known. The longitudinal measurement of the shower profile also allows a direct determination of X_{max}. A major inherent limitation of this detection principle is the fact that it only works in dark nights with decent weather[4].

A similar detection principle, used especially for lower energy showers (GeV up to about 100 TeV), is the detection of Cherenkov light emitted by the particles of the EAS while traveling faster than the speed of light in the atmosphere. As Cherenkov light is only emitted in forward direction, it can not be used to properly resolve the longitudinal development of a shower and only has a small effective area per detector station, but has a much smaller energy threshold compared to fluorescence light detection[5].

2.2.5.2. Particle detection at ground level

The easiest way to observe extensive air showers is to use particle detectors at ground. Due to the large areas that need to be covered (multiple thousand square kilometers for energies above 10^{19} eV) only a sampling with relatively small stations and distances up to a few km can be performed. Depending on the detector type and placement, different detection efficiencies and weights for the different shower components can be reached, which can lead to a separation of the shower components when multiple detector techniques are combined. An example for this is the combination of a scintillator detector at the surface to detect mainly the electromagnetic component with a buried scintillator detector to detect mainly the muonic component.

[4] no rain, fog or significant cloud coverage

[5] Cherenkov light is distributed only in a small solid angle range and results in a very short time-compressed light pulse.

2.2.5.3. Radio detection

An important additional technique to detect extensive air showers is the observation of radio waves. These waves are produced by the electrons ($e^{\pm}$) in the electromagnetic component and can be detected as short pulses by antenna stations near the shower axis. Two radio production mechanisms can be observed in air showers. With the so called geomagnetic effect the charge separation in the Earth's magnetic field causes a direction and position dependent signal [47]. A negative charge excess in the (moving) shower front caused by annihilation of positrons and knock-out of electrons from air molecules is the second mechanism [16]. From the signal strength and timing of the radio pulses shower observables like N_{max} and X_{max} can be reconstructed.

2.2.5.4. Hybrid approaches

A combination of multiple detection techniques on a single EAS can yield complementary information, enabling a better shower reconstruction. For example does a combination of fluorescence telescopes with a detector on the surface allow a better reconstruction of the arrival direction and subsequently the longitudinal profile than each of the systems alone. A combination of the longitudinal information about the electromagnetic component from fluorescence telescopes and the lateral information about the muonic component from a ground detector allows much higher precision for particle type determination and energy reconstruction than the single systems.

3. The Pierre Auger Observatory

The Pierre Auger Observatory is a detector complex for extensive air showers induced by ultra-high energy cosmic rays. It is the biggest cosmic ray detector ever built and is located near the city of Malargüe in western Argentina. It was originally designed to detect cosmic rays with energies between 10^{18} eV up to the highest energies beyond 10^{20} eV, but had its lower energy threshold extended down to 10^{17} eV with newer upgrades.

An extensive overview over the observatory is provided in [3].

The observatory is based on a hybrid design with two complementary detectors, which simultaneously measure the lateral shower profile on the ground and the longitudinal shower development. The part devoted to the lateral shower profile is the *Surface Detector* (SD), which consists of around 1600 detector stations on an area of roughly 3000 km^2. For the measurement of the longitudinal shower profile, the *Fluorescence Detector* (FD) is used. It consists of multiple fluorescence telescopes overlooking the area covered by the Surface Detector. Some additional detector enhancements, either extensions of the existing detectors or new detectors, have been added during the 15 years of operation and are also present in the area. An overview of the regular detectors is provided in figure 3.1.

The hybrid design of the detector allows to combine the advantages of both detection methods, while allowing the reconstruction of the properties of the primary cosmic rays with a precision not possible with a single detector type. While the FD has a much lower duty cycle compared to the SD, its systematic uncertainty of the energy determination is much lower, allowing for a good cross calibration. Even for analyses where mainly the FD is used, usually some information from the SD is added to improve the accuracy of the geometry reconstruction of the shower. Especially to determine the chemical composition of the primary cosmic rays, a combination of observables determined by both SD and FD leads to better separation between types of primary particles.

3.1. The Surface Detector

The Surface Detector consists of about 1600 detector stations using the Cherenkov effect in water (see figure 3.2). The stations are arranged on a hexagonal (triangular) grid with 1.5 km spacing in an area of about 3000 km^2. Each station is a tank with

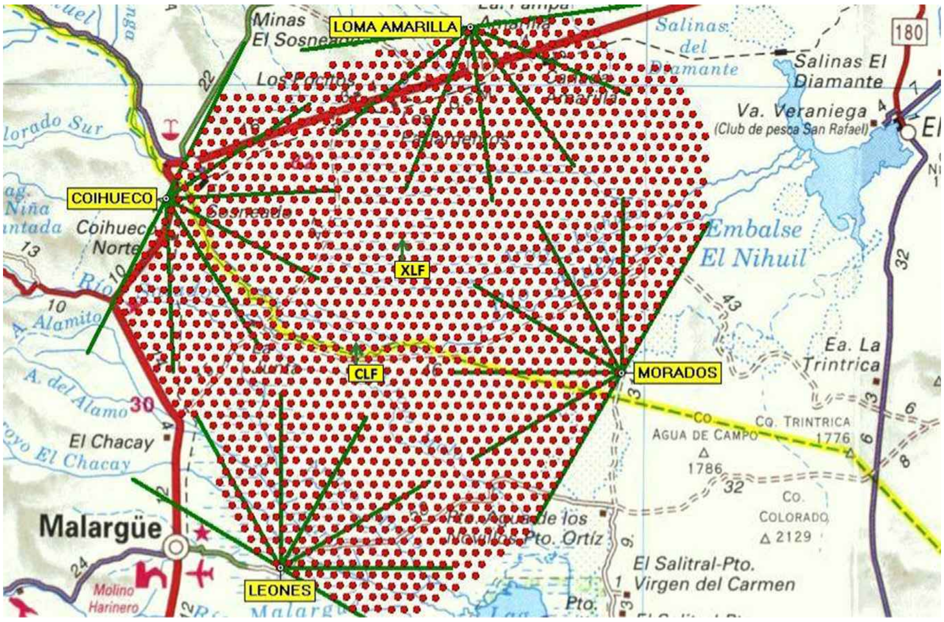


Figure 3.1.: The Pierre Auger Observatory. Stations of the Surface Detector are shown as red dots, while the field of view of the Fluorescence Detector is indicated by the green lines. Also two laser facilities (CLF and XLF) used for calibration are shown. Taken from [3].

about 12 tons of ultra-pure water, instrumented by three PMTs[1] to detect the light produced by Cherenkov radiation. The station is designed for autonomous operation with power being generated by a solar panel and stored in a battery for operation at night and during cloudy periods. Wireless communication is used to distribute triggers and facilitate data readout, while a GPS[2] module is employed to synchronize the internal clock of each station with the GPS atomic clocks.

The detector stations sample the lateral distribution of shower particles on the ground. A combination of muons and electromagnetic particles (electrons, positrons and photons) is measured, with the electromagnetic component mostly being stopped in the tank, while the muons as minimum ionizing particles mostly traverse the tank. Using timing information, the arrival direction of the shower and therefore the arrival direction of the primary cosmic ray can be determined. Evaluating the lateral shower profile, the energy of the primary cosmic ray particle can be estimated.

It is difficult to determine the type of the primary particle with the SD alone. Various approaches using the signal timing in single stations[3] or the overall shape of the lateral distribution are applied. These, however, are not able to achieve a clear separation between particle types for single showers, while additionally being strongly influenced by uncertainties on hadronic interaction models for air shower simulation.

[1] photomultiplier tubes

[2] Global Positioning System

[3] The arrival time distribution of muons is different from that of the electromagnetic component, with the muons arriving first.

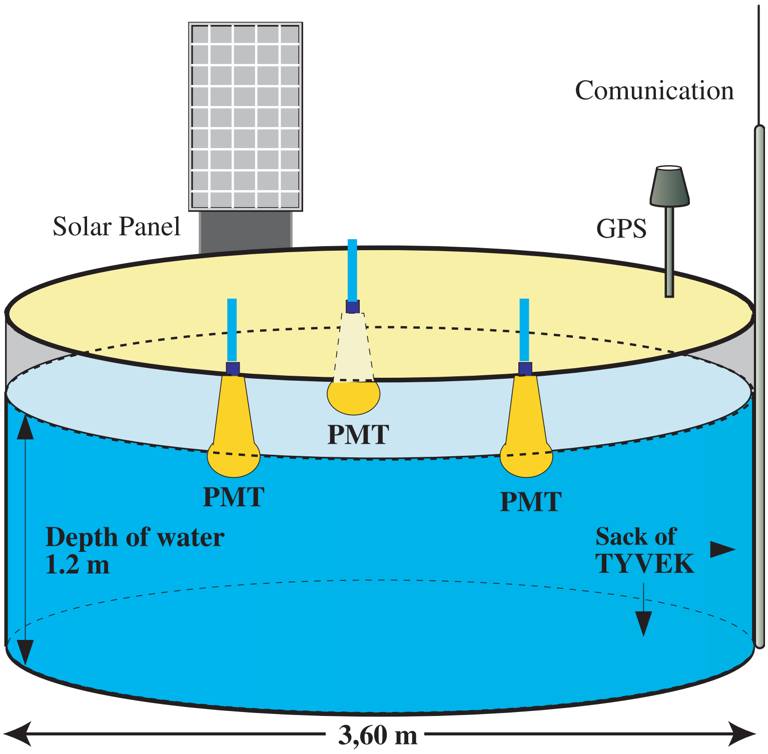


Figure 3.2.: Schematic view of an SD station [32]. The station is a tank with about 12 tons of ultra-pure water as a Cherenkov radiator, instrumented by three PMTs to detect the produced light. Power is generated by a solar panel and stored in a battery for operation at night and during cloudy periods. Wireless communication is used to distribute triggers and facilitate data readout, while a GPS module is employed to synchronize the internal clock to the atomic clocks in the GPS network.

3.2. The Fluorescence Detector

The Fluorescence Detector of the Pierre Auger Observatory [5] consists of 4 telescope sites or buildings[4] (see figure 3.3), housing 24 telescope *bays* (see figure 3.4) with Schmidt cameras. The segmented mirror of each telescope has an area of 10 m^2, focusing the light onto a camera equipped with 440 PMTs. To reduce the night sky background, a UV pass filter, which is mostly transparent for nitrogen fluorescence light, is installed at the aperture. A corrector ring is used as an approximation of a Schmidt plate. Each telescope has a field of view (FoV) of 30° × 30°. Each telescope site with six telescopes has a combined FoV of 180° × 30° (azimuth × altitude). All sites together overlook the complete SD array.

The Fluorescence Detector measures the fluorescence light, which is emitted by nitrogen molecules excited by the secondary particles of the air shower. Hereby mainly the electromagnetic part of the shower is visible with about 4 photons produced per electron per meter [71]. The angular resolution of about 1.5° per cam-

[4] Coihueco, Loma Amarilla, Los Morados and Los Leones

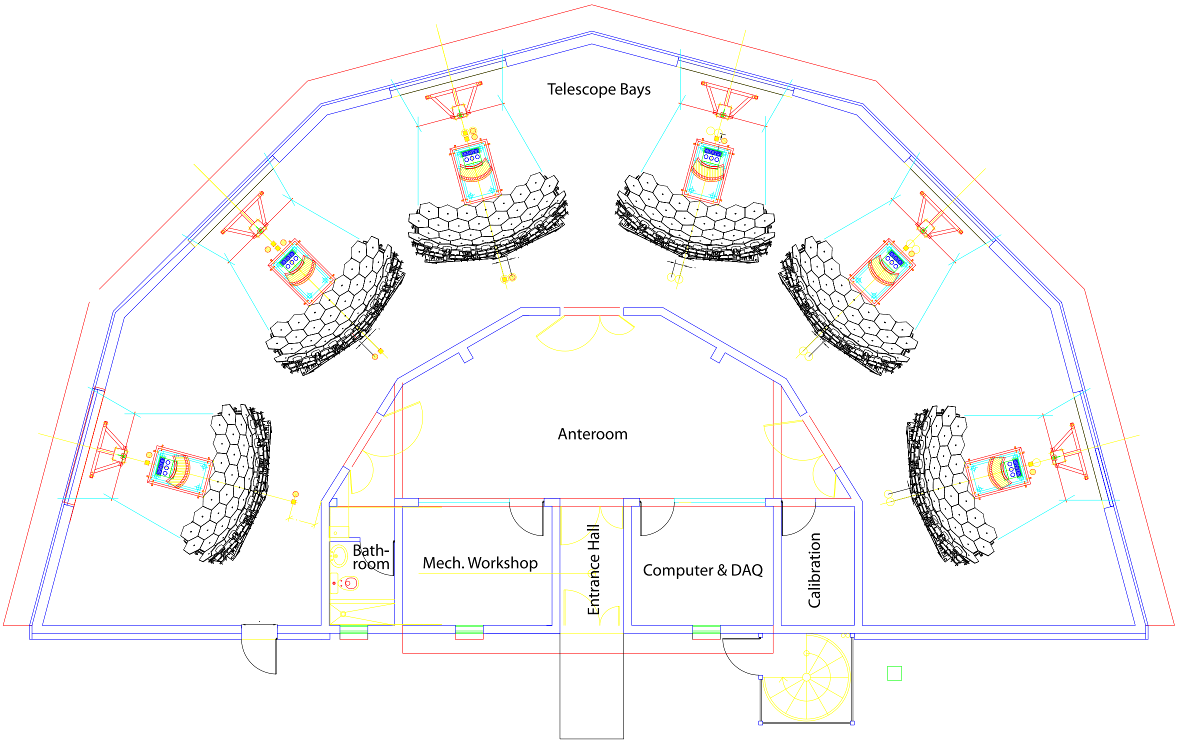


Figure 3.3.: Sketch of one Fluorescence Detector building [5]. Six fluorescence telescopes are located around some auxiliary rooms.

era pixel allows for a measurement of the longitudinal development with the atmospheric depth of the shower maximum $X_{\max}$ being the main composition sensitive observable of the FD. With the produced light being proportional to the energy deposit of the shower in the atmosphere, a direct calorimetric measurement of the shower energy is possible, based on an existing measurement of the *fluorescence yield*[5] as a calibration constant. Because the fluorescence yield as well as the light propagation from the shower to the telescopes depend on atmospheric parameters, these are continuously monitored, mainly using laser-based methods.

The FD can operate only during moonless nights with fine weather, which leads to a duty cycle of about 15 %.

[5] number of photons per deposited energy

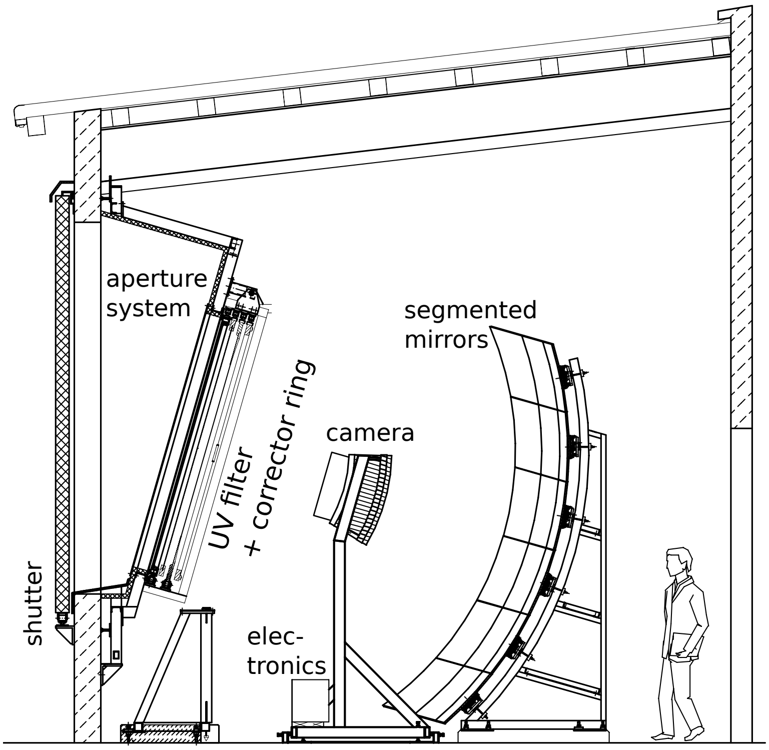


Figure 3.4.: Schematic view of one fluorescence telescope bay [5]. Fluorescence light enters through the shutter and aperture, passes a UV pass filter and is focused onto the PMT camera by a segmented mirror. Physicist for size comparison.

3.3. Existing enhancements and upgrades

In addition to the two main detector types, FD and SD, further detector components have been added since the start of the observatory.

3.3.1. AMIGA

AMIGA (*Auger Muons and Infill for the Ground Array*) [77] itself consists of two parts. One part of it is the infill, which consists of 61 additional SD stations between the original stations to reduce the grid spacing to 750 m in a small part of the array. The infill reduces the lower energy threshold of the surface detector to 10^{17} eV. With a steep increase in flux with lower energy, covering only a small part of the complete array is sufficient. The second part of AMIGA consists of scintillator muon detectors, which are buried near existing SD stations at a depth of 2.5 m. Here the pure muon signal at the SD stations is measured, due to the shielding provided by the earth, which only muons can pass. These muon detectors are a new component for the overall hybrid design of the observatory and allow for an improvement of the understanding of extensive air showers. Since the infill consists of normal SD stations it is nowadays often seen as part of the regular SD with *AMIGA* just referring to the muon detector part. AMIGA muon counters are currently installed at seven tanks in one hexagon in the infill area (*unitary cell*) with more planned.

3.3.2. HEAT

HEAT (*High-Elevation Auger Telescopes*) [56] is the name for three additional telescopes near the Coihueco FD site. These telescopes, which are optically identical to normal FD telescopes, can be tilted 30° upwards and a combined operation with Coihueco results in a total field of view up to 60° in elevation. This allows to observe showers at lower energies down to 10^{17} eV, which generally have an $X_{\max}$ above the standard FD field of view, as the visible faint showers are observable only very close to the telescopes while they also develop higher in the atmosphere. The HEAT telescopes overlook the area covered by the infill, so that both low energy extensions can be operated together in hybrid mode.

3.3.3. AERA

AERA (*Auger Engineering Radio Array*) [49] is the third enhancement of the Pierre Auger Observatory, covering an area of 6 km^2 in the infill area. It consists of 153 antennas sensitive between 30 MHz and 80 MHz. AERA is used to study the radio emission of extensive air showers in the VHF[6] band. With the radio emission in

[6] Very High Frequency

principle being calculable from basic electrodynamics, it could in the future be possible to improve the energy scale of the observatory using a cross calibration of the shower energy with AERA instead of the FD, thus eliminating the dependence on measurements of the fluorescence yield and atmospheric parameters. Also X_{max} can be determined with AERA, which allows a determination of this important shower parameter with 100 % duty cycle instead of the reduced FD duty cycle.

3.4. AugerPrime

After more than 15 years of data taking at the Pierre Auger Observatory, there are still open questions which can not be answered with more statistics alone. The solutions to the questions rely heavily on a more precise determination of the number of muons N_μ in the shower. To achieve this for the highest energies, the AMIGA unitary cell and the further planned AMIGA stations in the infill do not offer sufficient area coverage. Therefore a new detector component has to be added to the SD stations. The new detector part chosen for the upgrade is the *Scintillator Surface Detector (SSD)*, which consists of scintillator bars placed above each SD station. For better overall detector performance and to facilitate readout of the SSD, the main station electronics are upgraded. The complete upgrade is called *AugerPrime*, which is described in more detail in its preliminary design report (PDR) [4].

3.4.1. Upgrade of the SD electronics

For various reasons the electronics of the SD stations (SDE) is upgraded (SDEU). For this an Updated Unified Board (UUB) as the core of the SD tank electronics is developed.

The original design, which is more than 15 years old, only offers limited computational power for newer, more advanced triggers and data reduction for transfer. State-of-the-art FPGAs and processors can improve the capabilities.

To improve the possibilities of SD-tank based analyses using the exact shape and timing of the light pulses in the water Cherenkov detectors, the sampling rate of the ADCs used for the SD tank PMTs will be increased from the original 40 MHz to 120 MHz. With typical time differences between the tank PMTs in the order of 6 ns depending on particle direction and time bins of 8.3 ns, it becomes possible to extract some direction information for single stations. With the more precise measurement of the rising edges caused by single muons crossing the tank, for stations far away from the shower core counting of muons becomes possible with the WCD alone.

Very close to the shower core, the SD tank PMTs often saturate, which leads to the data of the station providing only a lower limit on the signal. Through the addition of a smaller PMT to the tank, which has to be read out and provided with bias voltage, the high-signal events can be recorded without clipping. This will be supported by the new electronics.

For further additions of detector components to the SD stations, digitization and communication has to be provided.

3.4.2. Proposals for a new muon detector

In order to determine the muon number with an accuracy higher than what is possible by a better timing resolution alone, an additional muon detector component will be added. Various approaches and concepts were proposed, each with different pros and cons. Two general approaches were possible:

- Add a muon detector either below the tank or buried close to the tank, which uses the tank or the earth as shielding to detect almost exclusively muons. The *Aachen Muon detector (AMD)*, which is described further in this thesis, belongs to this category.
- Add a detector with a different response to muons and electrons in the shower compared to the SD tank and calculate the muon and electron content using the inversion of the response matrix (*matrix method*).

While buried detectors like the ones used for AMIGA provide great accuracy, they are also very difficult to deploy, especially in parts of the array with high water levels[7]. While these will be deployed for part of the array to do a proper calibration of the matrix method (*high precision array*), the complete array will be outfitted with a scintillator detector mounted on top of the SD stations (*SSD*), which needs a much more complex analysis, but is vastly easier and cheaper to build and deploy.

3.4.3. SSD

The *Scintillator Surface Detector (SSD)* is the name for the surface detector extension. It consists of a scintillator panel with an area of $4\,\mathrm{m}^2$, made up of multiple plastic scintillator bars with wavelength shifting fibers guiding the light onto a single PMT. The new SD electronics board is used to digitize the single PMT signal per station. The SSD mainly measures electrons, while the SD tank measures more muons in comparison, enabling the matrix method for muon number determination. The planned SSD is shown in figure 3.5.

[7] Some SD stations are placed in a river.

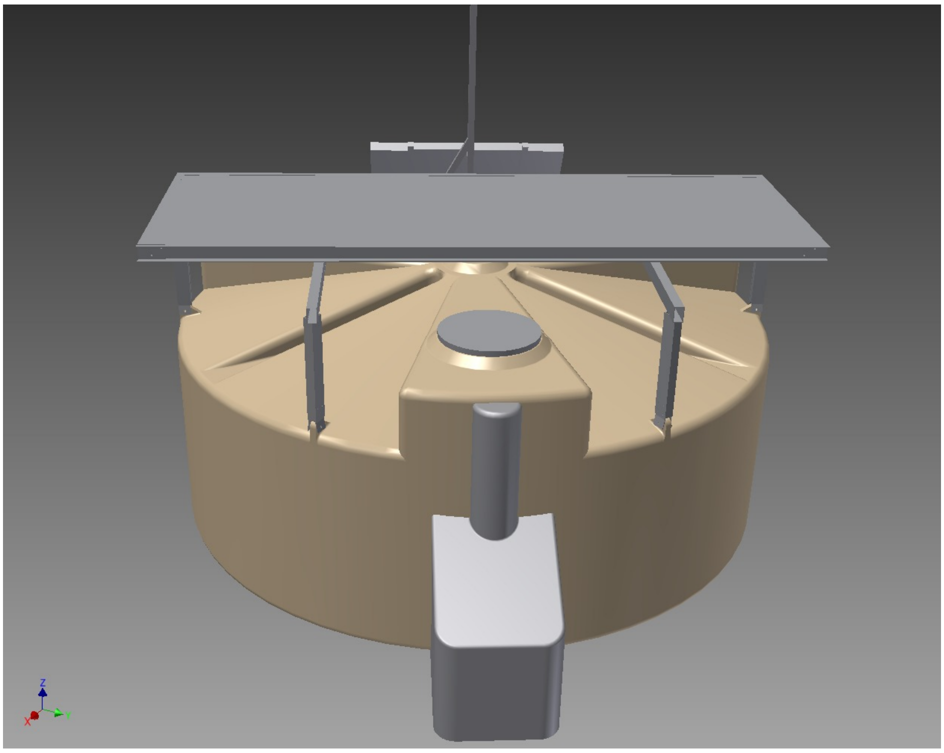


Figure 3.5.: The planned Scintillator Surface Detector (SSD) to be placed on top of the SD stations as part of AugerPrime [4].

4. Scintillators for particle detection

Scintillators are materials that emit light when they are crossed by ionizing particles. When read out by sensitive photon detectors (like photomultiplier tubes (PMTs) or silicon photomultipliers (SiPMs; see chapter 5)) this can be used to detect the particles. An almost linear relation exists between the number of photons emitted and the deposited energy. This can be used for the measurement of particle properties like energy or particle type. If the particle is stopped in the scintillator, the particle energy can be determined directly. For thin scintillators a measurement of dE/dx, the deposited energy per track length, together with its known dependence on the energy E, can be used to derive the energy. For some scintillator materials the time shape of the resulting light pulse contains information about the type of primary particle [55].

A special case of particles are muons, which over a broad range of momenta deposit a small, almost constant amount of energy per track length (see figure 4.1), and are therefore called *minimum ionizing particles (MIPs).* This on the one hand leads to their high penetration power and on the other hand makes them well suited for the calibration of particle detectors.

Other applications are time-of-flight measurements using thin scintillators and a readout with good timing resolution, as well as position measurements using segmented scintillators.

There are various materials used as scintillators, each having different characteristics. While inorganic scintillators like NaI have a good energy resolution due to high light output and their high density, they are difficult to handle because of their hygroscopy and brittleness and do not have a good timing resolution (e.g. decay constant $\tau = 230\,\text{ns}$ for NaI(Tl)). Organic scintillators offer a great timing resolution ($\tau \sim$ ns) at the expense of a rather mediocre energy resolution, caused by a lower light yield. Due to their lower density compared to inorganic scintillators, γ-rays are only detected through the Compton effect, which leads to a continuous Compton spectrum instead of a clear peak resulting from photoelectric conversion. Organic scintillators are often embedded into plastic material and are easy to handle and manipulate while being relatively robust to mechanical stress and moisture. Especially for big detector volumes, also liquid scintillators (organic scintillators in a liquid solvent) are used.

While the mean energy deposit for a MIP muon is a constant, the distribution of the

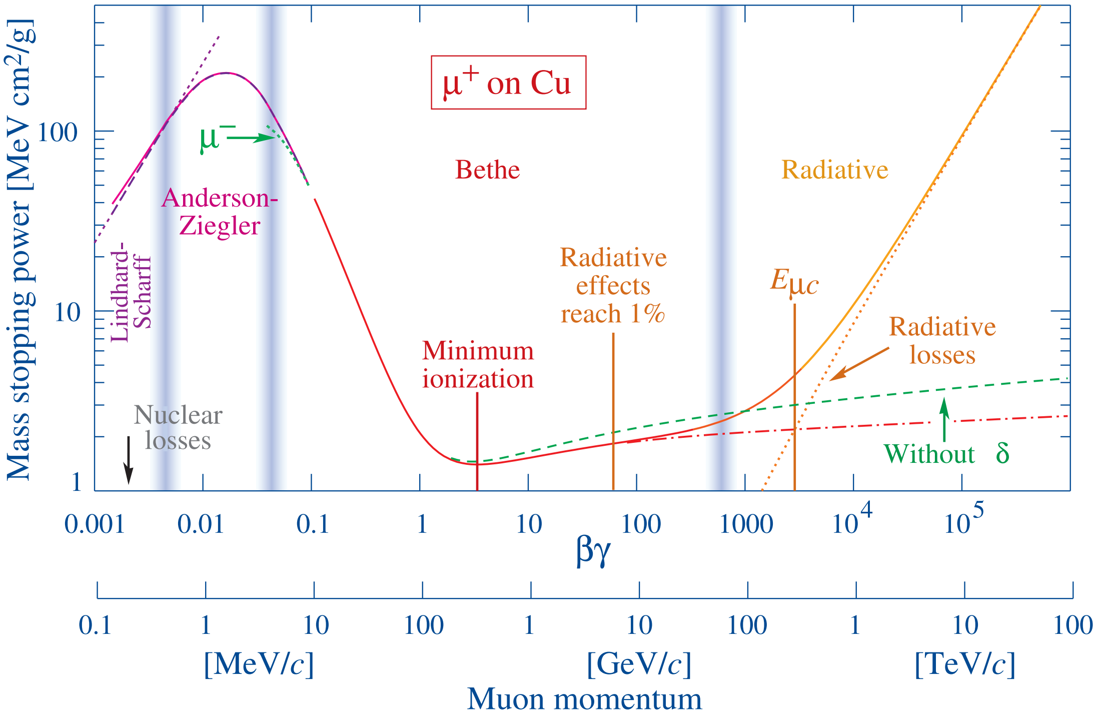


Figure 4.1.: Mass stopping power (= $\langle -\mathrm{d}E/\mathrm{d}x \rangle$) for positive muons in copper as a function of $\beta\gamma = p/Mc$. The region of minimum ionization for this material reaches from roughly 100 MeV to 100 GeV. Taken from [63, chap 32].

energy deposit when crossing a thin scintillator detector can be described roughly by a skewed Landau (or Landau-Vavilov) distribution [63, chap 32]. In figure 4.2 two different models are shown. The pure mathematical Landau distribution has a long tail, which causes the mean and variance of the distribution to be undefined. For this reason, to characterize the energy deposit and therefore the photon yield in a scintillator, usually the *most probable value (MPV)* is used rather than the mean, and the width at half maximum is utilized instead of the standard deviation. Of course a particle can at maximum deposit its total energy, therefore in physical reality there will be a suppression or cutoff of the tail.

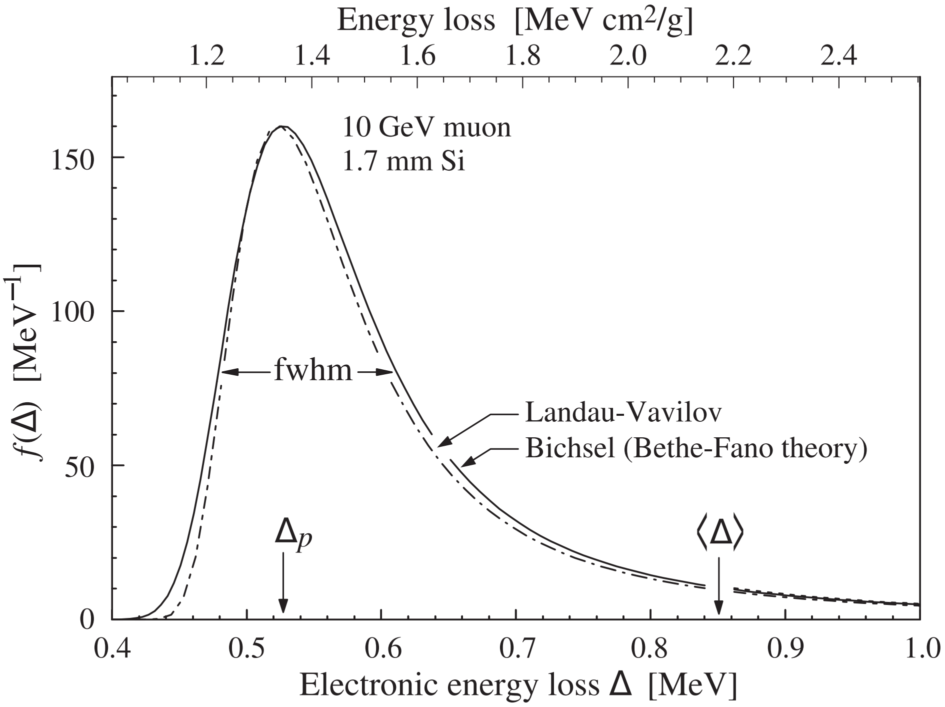


Figure 4.2.: Distribution of energy deposit for a 10 GeV muon traversing 1.7 mm of silicon (equivalent to about 0.3 cm of plastic scintillator) calculated using two different models. Δ_p is the most probable energy loss, and $\langle\Delta\rangle$ is the mean Bethe $\langle dE/dx\rangle$. Adapted from [63, chap 32].

4.1. Plastic scintillators

For the Aachen Muon Detector described in this thesis, plastic scintillator tiles are used. Plastic scintillators are the solution of an organic scintillator (fluorescent emitter, also called *fluor*) in a solid plastic solvent with a concentration of about 1 % [55]. The energy absorption from the passing particles mainly occurs in the base material. A part of the energy is then quickly transferred in a so-called *Förster resonance energy transfer* [36] to the actual scintillator emitting the detectable light. Often also a secondary solute in a low concentration is added as a wavelength shifter to reabsorb the primary photons (often in the UV range) and re-emit them in a different wavelength (often blue) which is easier to detect. The operation principle is sketched in figure 4.3.

During photon emission, a higher state often transits into a vibrationally excited lower state which then undergoes a radiationless deexcitation to the ground state. Due to this, good scintillators show a high transparency for their own light as the emitted photons do not have enough energy to directly excite the ground state to a higher energy level.

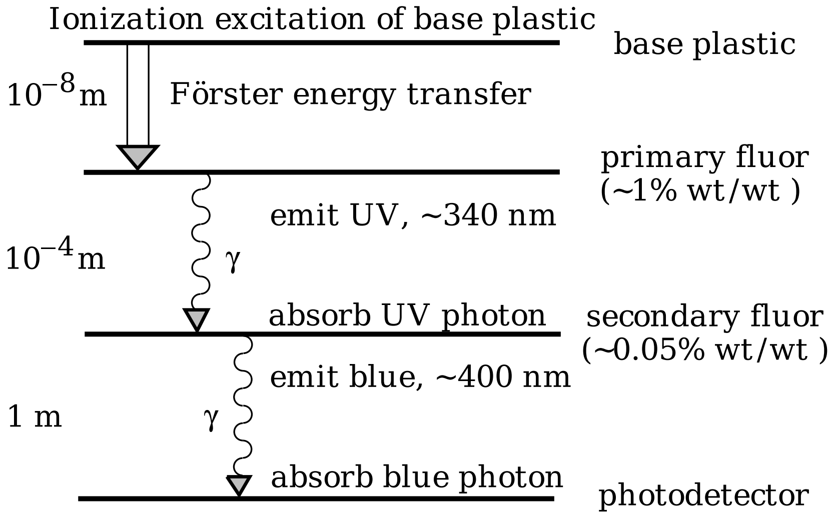


Figure 4.3.: Sketch of the operation principle of a plastic scintillator based detector with included wavelength shifting. The approximate length scales of the processes are shown on the left and the approximate fluor concentrations are shown on the right. Taken from [63, chap 33].

4.2. Photon collection and detection

To detect the light produced in a scintillator, often a PMT or an SiPM (see chapter 5) is used.

When the direct coverage of a sufficiently large surface area of the scintillator with a photon detector is not possible, there are two possibilities to increase the light output. An adiabatic light guide can be used to change the form of a surface without changing the area. This is for example often used to couple a thin but wide scintillator panel to a round PMT. By this the area covered is not changed but only its form, as Liouville's theorem has to be respected. The phase space (position spread and angle spread) covered by the ensemble of photons can not be reduced without losses. A reduction of the position spread at the expense of a larger angle spread is usually not possible as the angle spread already has its maximum possible value.

To overcome this limitation, wavelength shifting (WLS) fibers can be used. These are often glued or otherwise placed into plastic scintillators. When scintillation photons enter the fiber, they are absorbed and re-emitted with a shifted wavelength. This re-emission occurs isotropically, with about 5 % probability of the photon being contained in the fiber and being guided to its end [70]. In this application the shifting of the wavelength is just a side effect. The fiber usually has the absorption wavelength matched to the scintillation light, with the emission wavelength shifted such that the WLS fiber has an acceptable absorption length for its own light ($\gtrsim$2 m) and that the new photons can be detected with a high efficiency by the photon sensor. A clear fiber without the isotropic photon re-emission would not cause the same collection efficiency as all photons entering through the side do not fulfill the containment condition (total reflection) of the fiber and exit it again on the other side.

The WLS fibers cause a very big compression, which allows for the readout even of big scintillator tiles by SiPMs with their relatively small area. Also, the light yield of a larger scintillator tile becomes much more homogeneous through the use of WLS fibers compared to a coupling of the photo sensor directly to a small part of a side face [69].

4.3. Wrapping of scintillators

Bare scintillators lose light at the surfaces which are not covered by photo sensors. It is usually desired to (diffusely) reflect the photons back into the scintillator to allow part of them to be still detected. For this, scintillator detectors are often wrapped in a reflecting material like Teflon, Tyvek paper, aluminum foil or Titanium dioxide (as paint or co-extruded with a plastic scintillator). According to reflectivity measurements [46] the reflectance of all materials is similarly high, except for aluminum foil. Due to total reflection, which is better than any external reflection, occurring at polished scintillator surfaces, having an air gap between the scintillator and the reflector, is advantageous. Often the material that is easiest to apply, especially for a high quantity of detectors, is used.

5. Silicon Photomultipliers – SiPMs

Silicon Photomultipliers (SiPMs) are semiconductor photon detectors for low light levels with a good timing resolution. They are used as light sensors in the Aachen Muon Detector to detect the photons produced in the scintillator tiles and collected by the wavelength shifting fibers. SiPMs are also marketed under different names like MPPC (Multi Pixel Photon Counter) or SPM by some manufacturers. A photograph of an SiPM is shown in figure 5.1. A short but more complete introduction to SiPMs can be found in [73].

5.1. Structure

SiPMs mainly consist of an array of avalanche photodiodes (APDs), called cells, on a single silicon die, operated in Geiger mode. The general structure is shown in figure 5.2.

An APD is a diode which is operated with reverse bias, accelerating charge carriers (electrons and holes) in the depletion zone. When absorbing a photon in the silicon, an electron-hole-pair is created. The doping concentration of the silicon is often chosen such that a moderate drift field exists in an absorption zone, optimized in thickness for optimal photon absorption of a selected wavelength. Strong acceleration and amplification occurs in a small highly-doped acceleration layer, which the created electrons drift into.

When the energy gain in the electric field within the mean free path of the charge carrier is above the ionization energy of the silicon atoms, an avalanche of charge carriers is created, with the secondary charge carriers also being accelerated and causing further secondary charge carriers. This energy threshold is first reached for electrons, which leads to an avalanche developing only in the direction of the electron drift with the total number of charge carriers proportional to the number of primary electrons. This operation mode is called *proportional mode* and leads to a charge gain of $\mathcal{O}(10^2)$. When the energy threshold is reached for holes, also the holes can create secondary electron-hole pairs and contribute to the forming of the avalanche. This leads to a self-sustaining avalanche with a signal that is no longer proportional to the number of primary charge carrieres. This is called *Geiger mode* in analogy to the famous Geiger-Müller tube gaseous detector. The energy threshold is reached at the so called called *breakdown voltage* V_b, which is usually between 20 V and 100 V.

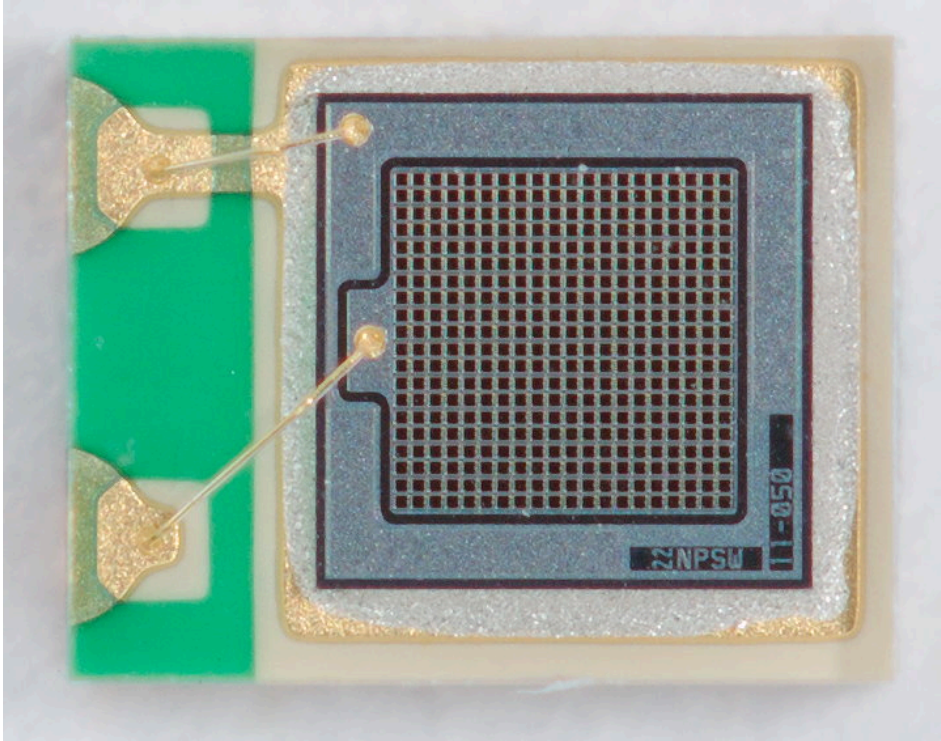


Figure 5.1.: Photo of a Hamamatsu S12571-50C SMD type SiPM. The active area is $1 \times 1\,\mathrm{mm}^2$, the distance between the 400 cell centers is 50 µm.

To end a cell breakdown, quenching is needed. For this, each cell has a quenching resistor, either made from polysilicon or metal, in series. The resistor causes the voltage applied to the cell to be reduced on cell breakdown below the breakdown voltage needed to supply the avalanche and therefore causes the avalanche to stop. After that, the cell, which can be modeled as a capacitor, slowly recharges (time constant $\tau \sim 50\,\mathrm{ns}$, strongly depending on SiPM type) and is ready again to break through and detect photons.

The cells being operated in Geiger mode means that each detected photon in one cell causes an identical signal of $\mathcal{O}(10^6)$ elementary charges, independent of the photon energy and also independent of the number of instantaneous photons hitting a cell. All cells are connected in parallel, which results in an output signal that is proportional to the number of simultaneously firing cells. The signal generated by one cell breakthrough is often named *photon equivalent* (p.e.).

A simplified SiPM circuit diagram is given in figure 5.3.

SiPMs are offered in different sizes (often $1 \times 1\,\mathrm{mm}^2$, $3 \times 3\,\mathrm{mm}^2$ or $6 \times 6\,\mathrm{mm}^2$), different cell sizes (usually between 10 µm and 100 µm cell pitch) and packages (through hole with conventional wire leads or surface mounted (SMD)).

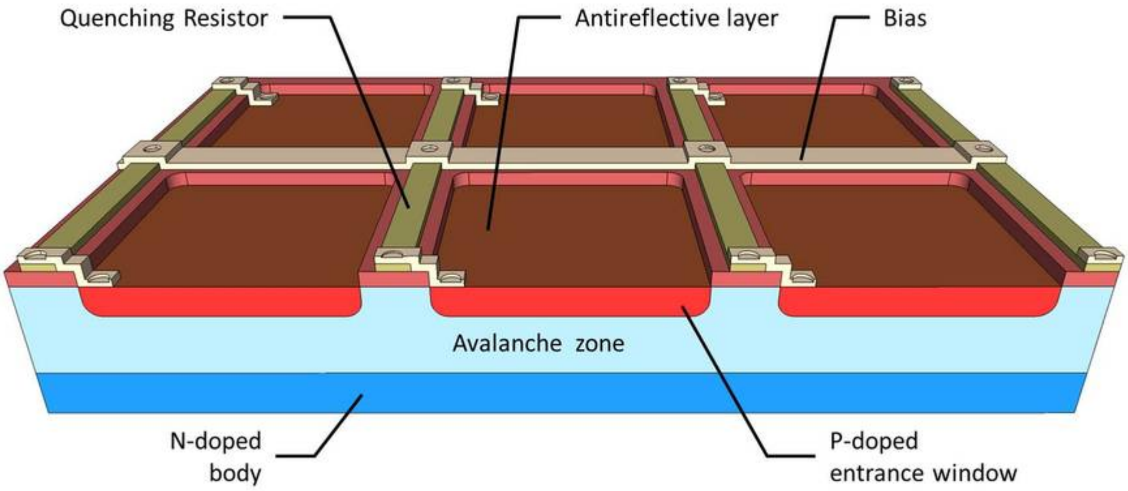


Figure 5.2.: General structure of an SiPM. Taken from [51]. Light is coming in from the top. The exact layer structure differs between manufacturers and device series. For the depicted doping a positive supply voltage has to be applied at the bottom.

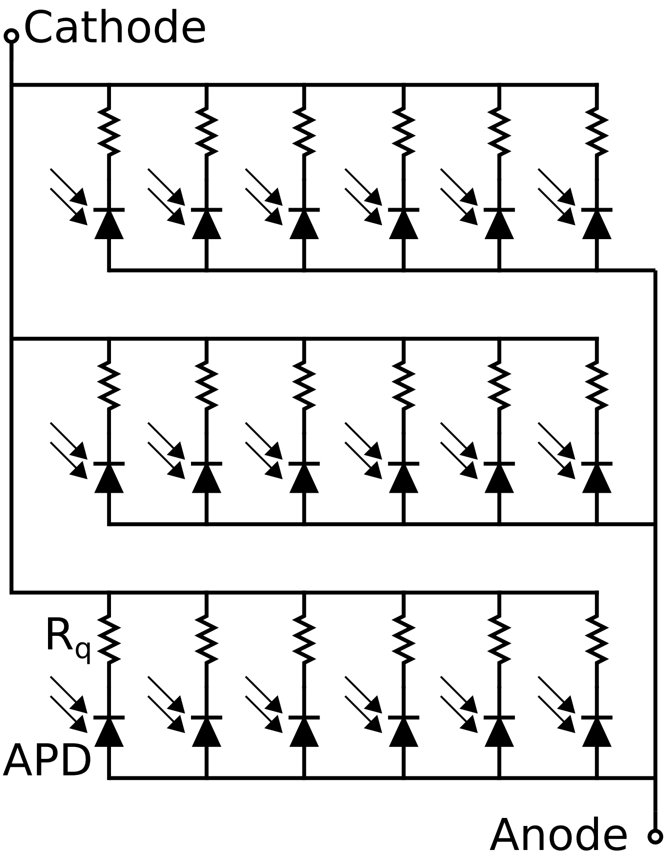
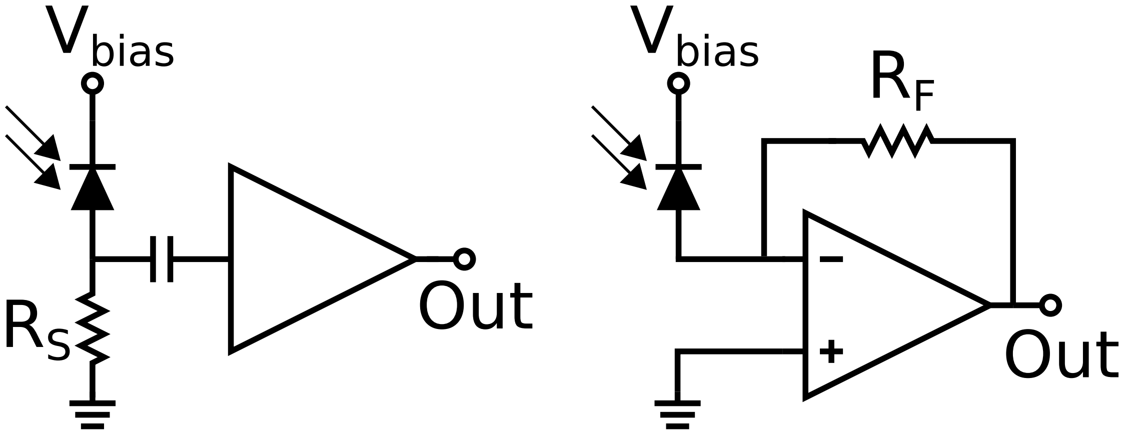


Figure 5.3.: Simplified SiPM circuit diagram. The order of diode and quenching resistor R_q can be inverted, depending on SiPM type.



(5.4.a) Passive current-voltage-conversion with optional AC coupling.

(5.4.b) Active current-voltage-conversion with transimpedance amplifier.

Figure 5.4.: Possible approaches for SiPM readout.

5.1.1. Readout

As SiPMs cause a current signal on breakdown[1], it usually has to be converted to a voltage signal that can be detected. Also further amplification beyond that achieved by the avalanche is often needed. To achieve this, either passive current-voltage-conversion using a shunt resistor R_S and successive voltage amplification can be done, or active current-voltage-conversion using a transimpedance amplifier can be implemented. Both approaches are shown in figure 5.4.

The passive approach is easy to implement and has few requirements on the amplifier (type, inverting/non-inverting, linear/integrating), which does not influence SiPM operation[2]. If an AC coupling capacitor is installed between the SiPM and the amplifier, the ground level of the SiPM can be different from the ground level of the amplifier if desired. The main downside is that the bias voltage of the SiPM is reduced by the voltage drop over the shunt resistor R_S, which reduces the bias voltage especially for big light fluxes and increases recharge times after a pulse. The complete transimpedance[3] is given by R_S times the gain of the amplifier.

With the active approach, the amplifier pulls its inverting input to ground (*virtual ground*), which leads to the SiPM always seeing the complete bias voltage applied without a voltage drop. To achieve this, a high speed current feedback amplifier should be used, which is often expensive, has relatively high power requirements and noise and can lead to an undesired oscillating behavior. This of course is no relevant disadvantage if a high speed amplifier is required anyway. As only an inverting amplifier can be realized, a second inverting amplifier in series is needed if positive polarity pulses are required. No AC coupling can be achieved at the input of the

[1] discharge and subsequent recharging of the diode capacity

[2] If its input impedance is high compared to R_S.

[3] Used instead of a gain for current-voltage conversion. Transimpedance has the dimension of a resistance.

amplifier, which leads to the ground levels of the SiPM and of the amplifier to be identical. The complete transimpedance is given by the feedback resistance R_F of the amplifier.

5.2. Characteristics

The main operation parameter which influences the SiPM characteristics is the overvoltage $V_{\text{over}} = V_{\text{bias}} - V_b$. It is the amount by which the supplied bias voltage V_{bias} succeeds the breakdown voltage V_b.

5.2.1. Pulse form and time behavior

Electrically an SiPM cell can be described as a capacitor with capacitance C_d charged to V_{bias}. It is internally discharged down to the breakdown voltage V_b during the avalanche process. The pulse form is influenced mainly by parasitic capacitances of the quenching resistor and the other cells, as well as parasitic inductances and capacitances of the connections of the SiPM. After discharge, the capacitor is recharged over the quenching resistor R_q which leads to a nearly exponential tail of the pulse. The length of the tail, which is at the same time the recovery time τ of the SiPM, is determined by $\tau = C_d \cdot R_q$. To a lesser degree it can also be influenced by an external shunt resistor, especially for large pulses with many simultaneously firing cells.

The recovery time of an SiPM is typically in the range of a few 10 ns to a few 100 ns, depending strongly on the cell sizes, with smaller cells (and therefore smaller capacitances) leading to faster recovery.

If a cell is hit by a second photon while it still recovers, during detection and breakthrough it behaves identical to a cell in an SiPM that is operated with the reduced overvoltage. The tail of the second pulse, which is caused by the recharge of the cell capacitor, is still determined by the full overvoltage.

5.2.2. Photon detection efficiency

One important feature of SiPMs is their Photon Detection Efficiency (PDE). It is the probability that a photon hitting the SiPM causes a cell breakthrough and is detected. For a given device it depends on the wavelength λ of the photon, the overvoltage V_{over} and the incident angle of the photon. It can be written as

$$PDE(\lambda, V_{\text{over}}, \theta) = \eta(\lambda) \cdot \varepsilon(V_{\text{over}}) \cdot F \cdot (1 - R(\theta)) \tag{5.1}$$

where $R(\theta)$ is the angle dependent reflectivity of the silicon surface and the applied coating of the SiPM, $\eta(\lambda)$ is the quantum efficiency of silicon, $\varepsilon(V)$ is the avalanche probability and F is the geometrical fill factor of the sensor. The fill factor is the ratio between the active cell surfaces and the complete device surface, which is reduced by dead space between cells needed for proper cell separation and for connections and quenching resistors. It is one important parameter to be optimized during device design. The avalanche probability gives the probability for the electron-hole pair to actually trigger a cell breakthrough and not recombine "harmlessly" before initiating an avalanche.

The PDE for SiPMs usually has a peak of between 20 % and 50 % for light with a wavelength of around 500 nm. This value varies strongly with cell sizes and device generations. The wavelength of the peak sensitivity can be shifted through changes in the device structure (layer thicknesses and doping).

5.2.3. Gain

The gain G of an SiPM, which is the ratio between the charge in a pulse and the charge of the electron that is created in the absorption of a photon, is given by the charge of the diode capacity divided by the elementary charge e:

$$G = \frac{C_\mathrm{d} \cdot V_\mathrm{over}}{e} \tag{5.2}$$

This means it is directly proportional to the overvoltage. For usual operation voltages and cell sizes, the gain is $\mathcal{O}(10^6)$.

5.2.4. Noise phenomena

Various SiPM noise phenomena exist, which are illustrated in figure 5.5.

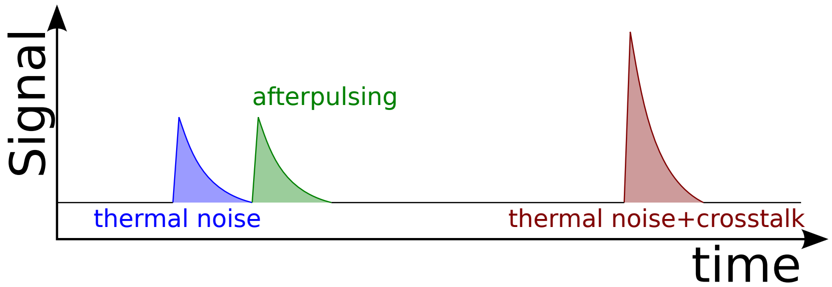


Figure 5.5.: Illustration of different SiPM noise phenomena.

5.2.4.1. Dark noise

The most noticeable noise phenomenon is *dark noise* or *thermal noise*. It arises when electron-hole pairs are created by thermal excitation of silicon atoms without the presence of impinging photons. A cell discharge can happen, which is indistinguishable from the ones of photon detection. Through being dependent of the avalanche probability ε, it also increases with the applied overvoltage.

For current SiPM devices and usual overvoltages at room temperature, the dark noise rate is around 100 kHz/mm^2.

5.2.4.2. Crosstalk

While dark noise is happening randomly, other noise effects can only occur during or after a cell breakthrough and are therefore called *correlated noise*.

The dominant effect is *optical crosstalk*. Just like photons can produce electron-hole pairs, electron-hole pairs can produce photons during recombination. This process, happening frequently during a cell discharge, can lead to photons escaping the cell and initiating a new avalanche in a different cell. This is especially evident for SiPMs otherwise displaying only dark noise, where pulses with a height corresponding to multiple simultaneous cell discharges can often be observed that would normally barely happen by chance.

The crosstalk probability P_{cross} is usually defined as the probability that a cell breakthrough causes at least one additional breakthrough. It can be calculated from measurements of the spectrum of dark noise pulse heights with N events where $N_{\geq 2\,\mathrm{p.e.}}$ is the number of events with a pulse height of at least 2 p.e.:

$$P_{\mathrm{cross}} = \frac{N_{\geq 2\,\mathrm{p.e.}}}{N}$$

The crosstalk probability is strongly dependent on the SiPM geometry and can be reduced by introduction of opaque metal-filled trenches between the cells. For conventional SiPMs at usual operation conditions, the crosstalk probability is around 30 %, while for newer trench-type SiPMs it can be reduced to below 5 %.

The crosstalk probability depends on the number of produced photons, which is proportional to the SiPM gain, and the avalanche probability. It therefore rises faster than linear with the overvoltage.

Crosstalk reduces the ability to do self-triggering on weak light pulses, as relatively high dark noise induced pulses can occur for a high dark noise rate. For a dark noise rate of $f_{\mathrm{dark}} = 100\,\mathrm{kHz}$ and a cross talk probability of $P_{\mathrm{cross}} = 0.3$ the rate of dark noise induced pulses with a height above 10 p.e. can be estimated to be

$$f_{10\,\mathrm{p.e.}} \approx f_{\mathrm{dark}} \cdot P_{\mathrm{cross}}^{9} \approx 2\,\mathrm{Hz} \quad .$$

On the other hand, crosstalk allows for an easy characterization of the SiPM gain using a dark noise spectrum, which is difficult if the crosstalk probability is too low.

5.2.4.3. Afterpulsing

A further type of correlated noise is *afterpulsing*. It appears when charge carriers produced during an avalanche are temporarily trapped by impurities in the silicon lattice and later initiate a new avalanche when being released. Afterpulsing occurs as pulses following normal cell breakthroughs with unusually short time distance, mostly with smaller amplitude[4].

While afterpulsing was a major problem with early devices, it is in the order of a few percent per cell breakthrough for current types and can often be neglected.

5.2.5. Dynamic range

For a low number of simultaneously arriving photons the number of triggered cells rises nearly linear with the light signal. For larger photon pulses single cells are hit by multiple photons and therefore non-linearities arise. The number of firing cells even in the weakly non-linear regime can usually be well described up to about 20 % of the number of SiPM cells [54]. For larger pulses it depends on the length of the light pulse because of possible partial recovery of the SiPM cells. For a known form (but unknown amplitude) of the light pulses, a good description can be achieved up to about the number of SiPM cells for short pulses and even beyond for longer pulses.

The dynamic range of an SiPM is commonly defined as the number of detected photons up to which the number of incident photons can be deduced with a given (low) relative uncertainty. For a given SiPM type with identical operating conditions, there are negligible device-to-device fluctuations in the SiPM response to light pulses. When employing advanced reconstruction methods based on SiPM simulation, even for unknown pulse forms of the light pulses, the dynamic range of SiPM based detectors can be increased by about an order of magnitude far beyond the linear response regime [27].

Additional limitations of the dynamic range of a system including readout are inflicted by amplifiers and the signal digitization.

5.2.6. Temperature dependence

Most SiPM properties like gain and PDE only depend on the overvoltage V_{over}. Due to the change of silicon lattice vibrations, the mean free path of charge carriers in the amplification zone changes with temperature and therefore also the breakdown voltage V_{b}. To reach a constant overvoltage, the applied bias voltage has to compensate for this temperature dependence. The temperature coefficient is usually between 20 mV/K and 60 mV/K, depending on device type.

Also the dark noise rate of the SiPM, which is caused by thermal excitation, is strongly influenced by temperature. As a rule of thumb the dark noise usually increases by a factor of 2 every 7 – 8K around room temperature, but the temperature dependence is not exactly exponential.

[4] They usually happen when the cell is not completely recharged.

An additional change of SiPM properties with temperature arises from a temperature dependence of the quenching resistor. This is not very much noticeable around room temperature but more at cryogenic temperatures, where a reduction of resistance for metal film quenching resistors can lead to insufficient quenching, while an increase for polysilicon resistors leads to extremely long recovery times.

5.3. Comparison to other photon detection methods

There are various other photon detection methods available. The detectors most comparable in terms of timing and pulse resolution are photomultiplier tubes (PMTs). Compared to these, SiPMs have the advantage of a much lower cost per channel, better single photon resolution, increased robustness to mechanical stress or high light levels, insensitivity to magnetic fields, a higher photon detection efficiency, much smaller device-to-device fluctuations with at the same time easier characterization, the need for a much smaller supply voltage and much smaller aging effects[5]. The downside of SiPMs are the higher cost per area compared to large PMTs, the higher thermal dark noise and the bigger temperature dependence, which has to be compensated for in the power supply. Therefore SiPMs are unsuited for applications needing very large instrumented areas (e.g. Super Kamiokande [38]) or which require low noise levels to achieve self-triggered counting of single photons. For other applications that can tolerate the relatively high noise levels and the small areas, SiPMs are the ideal and future-proof replacement for PMTs.

[5] No aging of SiPMs outside of high radiation environments has been observed, yet

6. FPGAs

Field Programmable Gate Arrays (FPGAs) are programmable logic devices that allow the creation of compact, fast and highly parallelized digital circuits. As the name already suggests, the configuration, which describes the logic functions implemented in the FPGA, can be changed *in the field* after it has been installed on a printed circuit board. It therefore is highly flexible and adaptable to changing requirements and allows for the creation of hardware[1] before the final digital logic has been designed.

The main readout board of the Aachen Muon Detector, which is described in detail in section 9.1.3, is equipped with one FPGA in addition to the EASIROC (see chapter 7) chip. The FPGA is used to steer the EASIROC, make trigger decisions, and readout and buffer data from the EASIROC to transfer them to a PC.

There are multiple manufacturers of FPGAs which use slightly different nomenclature for hardware features or parts of the configuration process. Throughout this thesis the nomenclature as employed by Altera is used.

6.1. Hardware

An FPGA contains a multitude of logic elements (LEs), each usually consisting of a look up table (LUT) and a D flip-flop (often called register). The LEs are connected via routing structures to transmit their signals. One example of an LE is sketched in figure 6.1.

One LUT features multiple inputs[2] and one output that can be configured to be an arbitrary combinational function[3] of the inputs. Often the LUT can be divided into two or more simpler[4] LUTs partially connected to neighboring LEs via a carry chain to allow the resource-effective calculation of sums[5].

A D flip-flop captures the value of its input D at each edge of an input clock and displays its state at the output Q. This means the register is a fast one-bit storage, which permanently provides its stored value at its output and has its stored value updated every clock cycle. Various additional control signals like asynchronous[6]

[1] or the usage of generic hardware

[2] often 4 or 6, depending on the device family, price segment and age

[3] e.g. simulating an OR gate

[4] e.g. only function as LUTs with fewer inputs or with limited logic functions

[5] ripple carry adder using only one LE per bit instead of 2 LEs per bit

[6] working immediately and not at the next clock edge

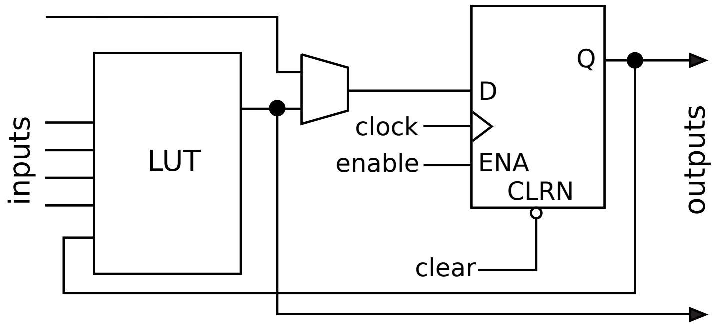


Figure 6.1.: Sketch of one logic element (LE) of an FPGA.

clear and set signals and a clock enable input (D input is only captured at a clock edge if the enable input is active) are usually present.

Also present in the LEs are various multiplexers, with their *select* inputs either driven by a signal or set during configuration, to facilitate the routing of signals into and out of the LEs and to enable optional features.

A small number of LEs[7] are grouped together in logic array blocks (LABs) which feature fast interconnects for signals between the included LEs and share control signals like clocks or asynchronous signals.

In modern and high-end FPGA families the LE as the basic unit is being replaced by more advanced units like Adaptive Logic Modules (ALM) that feature multiple LUTs and registers which can be used in a flexible way for a variety of purposes without the strict mapping of one register to one LUT [11]. Also special MLABs can be used as either a normal LAB with ALMs or as memory.

Throughout the whole FPGA a grid of column and row interconnects is used to distribute signals between distant LABs and to or from input-output (I/O) pins or special purpose blocks. Usually there are also lines for a small number[8] of special global signals (like clocks or asynchronous reset signals) that are distributed throughout the whole chip with the minimum possible time difference.

Usually at the edges of the FPGA, I/O banks[9] are located. These I/O banks are connected to external electrical contacts of the FPGA chip package. They can (depending on the device configuration) receive digital signals or output digital signals from the FPGA. Each I/O normally also contains its own registers (similar to those in the LEs) to buffer or synchronize input and output signals. The direction (input or output) can usually be selected either at configuration time or at runtime via an output enable signal to allow data transfer in both direction over the same line. The logic signal level (e.g. 3.3 V LVTTL, 1.5 V LVCMOS, LVDS[10]) normally can be selected in the FPGA configuration but the possibilities for a given I/O pin (especially for output) are usually limited by the voltages supplied externally to its I/O bank.

[7] usually around 10-16

[8] 4-10

[9] group of I/O pins supplied by the same operation voltage input

[10] using predetermined pairs of two adjacent I/O pins acting as a single I/O

To generate the clocks which drive the different registers in the FPGA, it features one or more phase lock loops (PLLs). PLLs use an externally connected clock signal (often from a crystal oscillator) to fine tune an internal oscillator while maintaining a fixed phase relationship between input and output clock. Through this, multiplication of clock frequencies by fractions with integer numerators and denominators becomes possible. Depending on the FPGA family there can be a different number of internal clock dividers in the PLLs to generate multiple synchronous clocks.

For configuration and debugging, an FPGA contains special circuitry (often compliant to the JTAG[11] standard [76]) to be connected to an external programming adapter. Through this, either the configuration can be written into the FPGA, or other data useful for debugging can be transferred between a PC and the FPGA (e.g. an internal logic analyzer).

FPGAs usually also contain a number of special purpose blocks. The most commonly present special purpose block is static random access memory (SRAM) that can be used from within the FPGA to form memory units of various sizes and word widths, FIFOs (first-in-first-out memory) or big shift registers that would otherwise use a huge amount of registers in normal logic elements. Other common special purpose blocks are DSP (digital signal processor) blocks, which allow for fast multiplication of numbers or floating point arithmetic, and high-speed transceivers for PCI, PCI express, DDR SDRAM or high-speed serial protocols.

The configurability of the FPGAs is normally achieved through many SRAM cells distributed throughout the chip. During configuration these SRAM cells are written. The voltage level of the memory cells then opens transistors to create connections between signals, drives multiplexers, etc. These SRAM cells are volatile and lose their configuration when power is lost. On power on, the configuration has to be provided either by a special configuration adapter from a PC or from a configuration device next to the FPGA, which contains the configuration in a flash EEPROM (electrically erasable programmable read-only memory) from which the FPGA can retrieve the configuration automatically after power on.

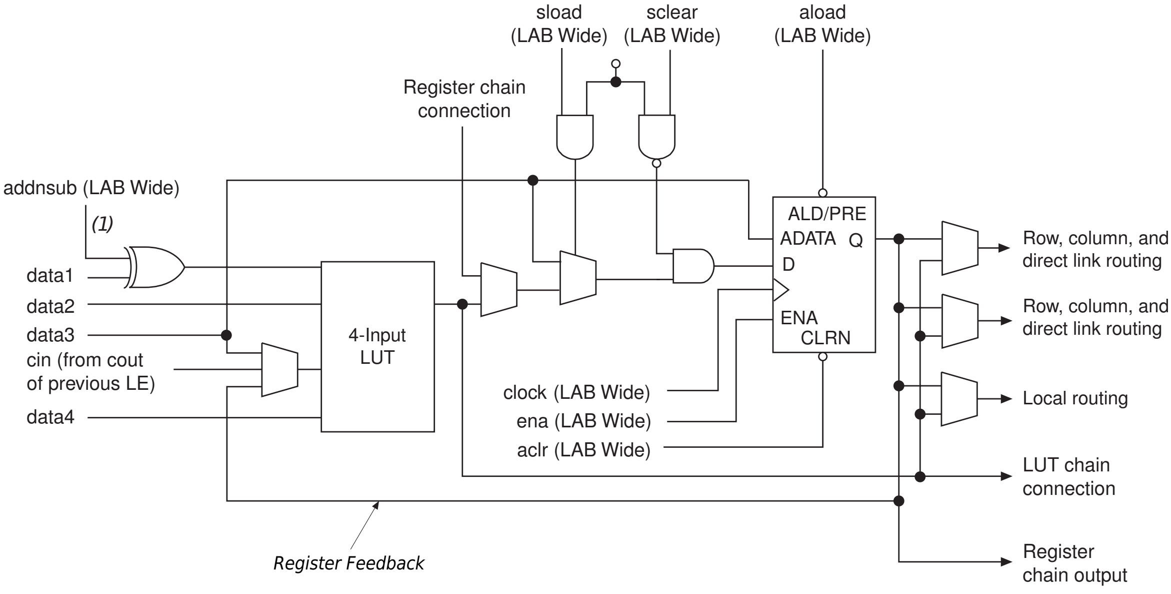
Many modern FPGA families also offer variants with integrated microprocessor cores, as these are more suited for some workloads and are easier to program. Highly timing-critical, high bandwidth or highly parallelized tasks are then implemented in the FPGA part. Complex tasks that have lower timing requirements are done on the processor cores, while using less hardware resources and easier code.

6.1.1. Altera Cyclone family

The main readout board of the AMD is equipped with an Altera Cyclone EP1C6 [9] FPGA. It belongs to the Cyclone[12] family of FPGAs which were introduced as low-

[11] Joint Test Action Group

[12] often called *Cyclone I* to distinguish it for its successors



(1) This signal is only allowed in normal mode if the LE is at the end of an adder/subtractor chain.

Figure 6.2.: Cyclone family LE in normal mode, suitable for general logic applications [9].

cost and low-power devices in 2002. For production a 0.13 μm process[13], resulting in a supply voltage of 1.5 V, is used. Due to its age and manufacturing process, all Cyclone FPGAs have a rather high power consumption and few logic resources by modern standards. The Cyclone family has since been superseded by the families Cyclone II, Cyclone III, Cyclone IV, Cyclone V and Cyclone 10.

The Cyclone LEs each contain one four-input LUT (figure 6.2). In arithmetic mode, which is used to implement adders, it can be divided into four two input LUTs (figure 6.3). Two of these two input LUTs generate the normal LUT output, selected by the carry input from the neighboring LE or a third input[14], the other two LUTs generate carry signals. Which of the two carry signals is used depends on a LAB-wide select signal, this allows easy and resource effective changes between additions and subtractions.

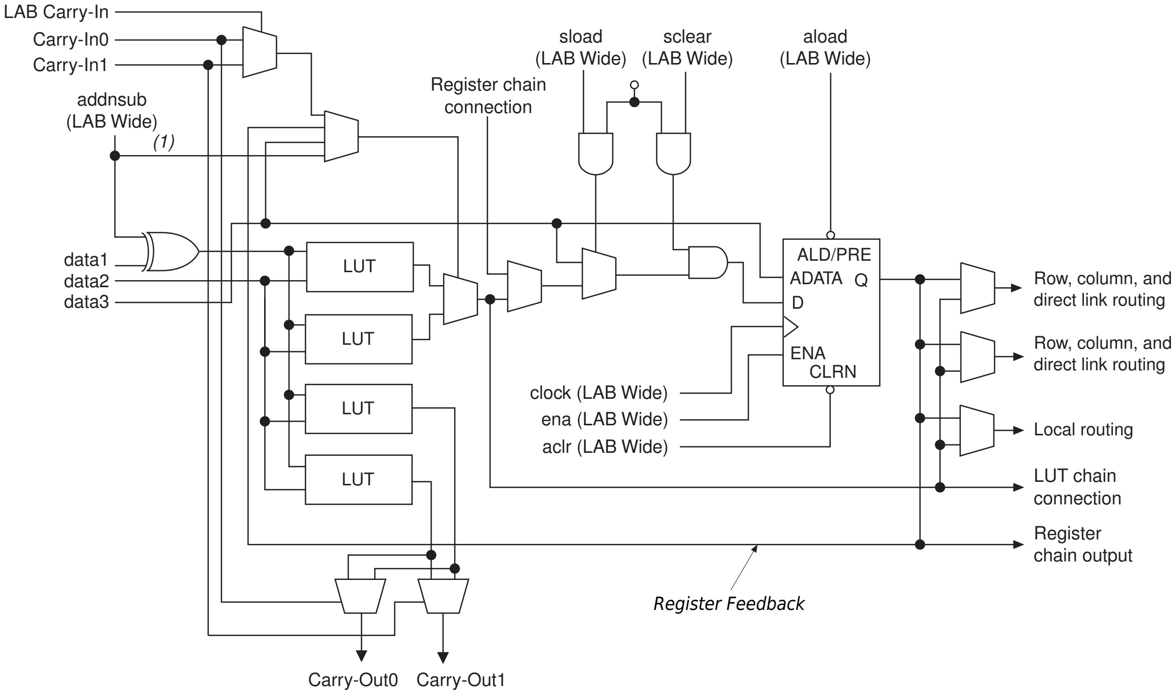
The PLLs in the Cyclone FPGAs contain two clock dividers for internal clocks[15] and can therefore generate a maximum of two different synchronous clocks from a single input clock. A schematic view is shown in figure 6.4.

The RAM blocks in the Cyclone family are called M4K and contain 4608 bits of memory. These can be used as 4096 × 1, 2048 × 2, 1024 × 4, 512 × 8, 256 × 16, 128 × 32 bits as well as 512 × 9, 256 × 18, 128 × 36 bits to enable error checking codes on the RAM content. If desired, the M4K blocks can be used as true dual port memory with two independent read/write ports, operating simultaneously on the same memory

[13] nominal minimal width of structures in the silicon

[14] These effectively form a three input LUT.

[15] one additional divider for an external output clock



(1) The addnsub signal is tied to the carry input for the first LE of a carry chain only.

Figure 6.3.: Cyclone family LE in arithmetic mode, especially suited for implementing adders [9].

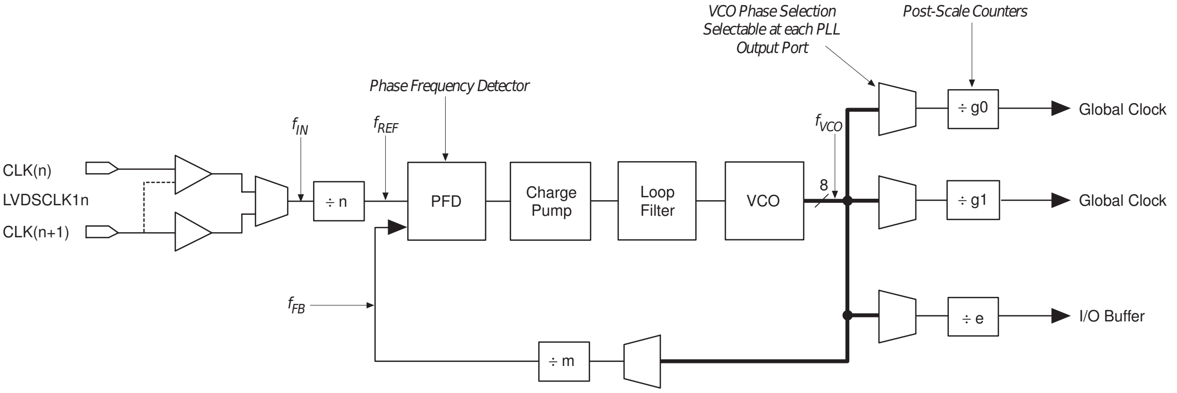


Figure 6.4.: Schematic of one PLL of the Cyclone family [9].

content[16].

The Cyclone family does not feature any DSP blocks or special transceivers.

The EP1C6 contains 5980 LEs, 92160 bits of memory in 20 M4K RAM blocks, 2 PLLs and 185 I/O pins.

6.2. Configuration

Bare FPGAs are without function. They only gain functionality after they are configured/programmed, i.e. their SRAM cells have been written.

The configuration file has to be generated from a description of the desired logic. This complex task is usually conducted by tools which are integrated into an integrated development environment (IDE) provided by the FPGA manufacturer. In addition to the compilation tools, a code editor, debug and visualization tools and a variety of GUIs and wizards to setup and configure projects and the compilation process are included in the IDE.

The logic description is usually written in either VHDL[17] [75] or Verilog [74]. These hardware description languages (HDL) break down the logic into individual modules and their interconnections. Those modules might be anything from basic units like adders, shift registers or FIFOs to modules handling communication with other devices or steering and storing values from an ADC (analog to digital converter). To utilize special device resources like memory blocks (as normal RAM or for FIFOs) or to efficiently do common tasks like counters, shift registers or multiplexers, the device manufacturers include various parameterized modules with their IDEs. For highly complex tasks like implementing an Ethernet interface, often third party intellectual property can be licensed for usage.

All generated logic works continuously in parallel. This is in contrast to e.g. microcontrollers which process a sequence of commands, one at a time. If a sequential task is required, usually a complex state machine has to be designed with rather complex timing interdependencies. State machines are defined by a description of the states and the state transitions. When doing state transitions or when being in a state, single steps of a bigger task can be performed, with the proper sequence of the states achieving a desired sequential task. Every state machine as well as every other function implemented in the FPGA continuously uses its assigned logic resources, even when waiting in an idle state and not doing any actual work.

Compilation from source to finished configuration usually takes multiple steps. In the first step the hardware description files are analyzed, their equivalent in logic gates[19] is deduced and the logic is mapped to a network of LUTs, registers and spe-

[16] Simultaneous write accesses to the same addresses have to be avoided.

[17] **V**HSIC[18] **H**ardware **D**escription **L**anguage

[18] **V**ery **H**igh **S**peed **I**ntegrated **C**ircuit

[19] primitive logical operations like OR, AND, XOR

cial blocks in the synthesis step. During this, multiple optimizations which modify the resulting logic to be more efficient (either resource usage or speed) without changing any outside-visible behavior are performed. In the next step, the fitter has to place the resources deduced during synthesis at concrete locations in the FPGA and route the needed interconnections through actual hardware interconnects. Also here optimizations are performed for example to reduce interconnect usage through placing corresponding logic close to each other or to more efficiently use the logic resources through combining two LEs where one utilizes only the LUT and the other uses only the register[20]. The fitting process is based on a number sequence from a pseudo random number generator (RNG) and therefore can yield vastly different results for different seed values for the RNG and for designs differing only very slightly in their HDL representation. Finally, from this configuration files can be generated.

Depending on the amount of logic, the logic resource utilization in the FPGA and the used processor and IDE version, a full compilation can take anything from half a minute to multiple hours.

6.2.1. Timing

One important verification step, after the fitter is finished, is the static timing analysis. Here the worst and best[21] possible signal delay from one register through all LUTs and interconnects to the next register is calculated for all signals. It is then made sure that the setup and hold requirements[22] as well as the removal and recovery requirements[23] of all registers are met. The time margin for the worst case signal path is called *slack*, while positive slack is good, designs with negative slack might show undefined and unintended behavior.

For the timing analysis to work properly, some *timing constraints* have to be defined. Here for example the characteristics of external clocks are given and exceptions for signals that are allowed to take multiple clock cycles, or that are even not timing critical at all, are defined. If specific relationships between input or output signals and external clocks are desired, they can also be given and are then checked during the timing analysis. Some timing constraints are also respected by the fitter with setting delay on input or output pins or allowing logic driven by a slow clock to be further apart[24] in the chip than logic driven by a fast clock.

Violations of hold requirements usually only happen if there is a sizeable skew between clocks[25] used for the *from* and *to* registers, while violations of setup require-

[20] logic cell packing

[21] depending on device variations and operating conditions

[22] The D input of the D flip-flops has to be stable for a specific time span before and after a clock edge.

[23] The clock enable and asynchronous control signals of registers have to be asserted or deasserted at a sufficient time distance from a clock edge.

[24] bigger signal delays

[25] e.g. through the usage of normal logic registers instead of PLL scalers as clock dividers (ripple clock)

ments is a common occurrence when too much combinational logic is placed between registers or the fitter fails to keep propagation delays low enough by ensuring short signal path lengths. To increase the setup slack, either the clock frequency can be reduced, or the amount of logic between registers can be reduced by introduction of more intermediate registers (*signal pipelining*)[26] or the shift of intermediate registers (*register retiming*, might be done automatically in the fitting or the synthesis step). If timing fails only very slightly, a new fitting approach with a different seed for the RNG might yield a sufficient improvement.

[26] For example splitting the sum of four numbers into two sums of two numbers, then a register and then the final sum of the two intermediate results.

7. The EASIROC ASIC for SiPM readout

The SiPM signals in the Aachen Muon Detector have to be digitized. For this the EASIROC chip is used. It is installed on a circuit board (*EASIROC board*) together with an FPGA and other support components (described in section 9.1.3).

The EASIROC [30] (*Extended Analogue Silicon pm Integrated Read Out Chip*) is an application-specific integrated circuit (ASIC) for SiPM readout. It was developed by the Ωmega (*Orsay MicroElectronics Group Associated*) at the LAL (*Laboratoire de l'Accélérateur Linéaire*) of the IN2P3 in Orsay, France. The EASIROC is one step in the series of ROC chips by Ω, being based on the SPIROC [26], while itself being developed further into the CITIROC [34]. Other ROC chips not specifically targeted for SiPM readout but optimized for other detector types also exist. As a front-end chip it enables the readout of 32 SiPMs while being optimized for low power consumption[1], allowing almost all functions, also for individual channels, to be powered down. The power consumption for the complete chip is about 155 mW.

The EASIROC serves these main purposes:

- Adjustment of the SiPM supply voltages
- Preamplification of the SiPM signals
- Threshold discriminators per channel for trigger purposes
- Recording of a charge integral per channel

The EASIROC allows an individual adjustment of the bias voltage V_b of each of the 32 connected SiPMs. While all SiPMs are supplied with a common external voltage V_ext, the virtual ground level of each SiPM is shifted by $V_\mathrm{DAC} = 0 \ldots 4.5\,\mathrm{V}$ using an 8-bit DAC (Digital to Analog Converter) in the EASIROC for each channel (internally called DAC8). This leads to the voltage $V_{\mathrm{b},i} = V_\mathrm{ext} - V_{\mathrm{DAC},i}$ (with i being the number of the SiPM or channel) being visible to each SiPM. To enable this, the 50 Ω shunt resistors on the EASIROC board, used to convert the current signals of the SiPMs into a voltage signals to be amplified in the EASIROC preamplifiers, are therefore not

[1] with drawbacks on speed and channel-to-channel fluctuations

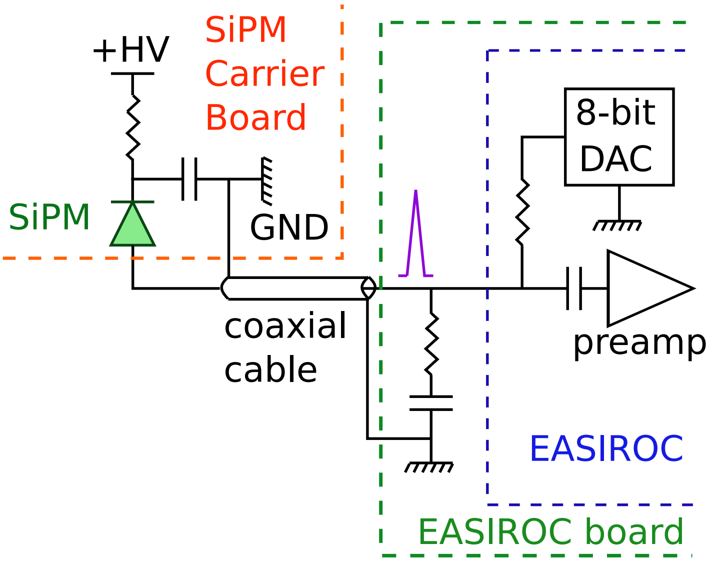


Figure 7.1.: Method for SiPM voltage fine adjustment as foreseen with the EASIROC. Adapted from [64].

connected to ground directly but only via additional capacitors[2]. During operation, the mean voltages of the capacitors are equal to the respective V_{DAC}. The SiPM signals themselves are then fed to the preamplifiers via AC coupling to remove the DC offsets applied by the DACs and to be able to operate the preamplifiers at a baseline different from ground, to forgo the need for a negative supply voltage.

The DACs have a high impedance for high frequencies to not disturb the SiPM signal present on the same line. To conserve power in the DACs, they can only sink current flowing through the SiPMs but not source current. This means they can only lower but not raise the V_{DAC} voltages themselves. For normal operation this is not relevant, but if the output voltage of the EASIROC DACs is to be measured, a multimeter with a high input impedance $\geq 1\,\mathrm{G\Omega}$ (normally $\approx 10\,\mathrm{M\Omega}$) has to be used. For the DACs either a precise external voltage reference on the EASIROC board ($\approx 4.5\,\mathrm{V}$ to achieve the maximum possible range of V_{DAC}) or an internal, not temperature stabilized, reference of 2.5 V can be used. By default the 256 steps of the 8-bit input DACs cover a range of about 5 V, which leads to a best-case resolution of $\approx 20\,\mathrm{mV/step}$. By using an external resistor the slope can be changed to allow for a smaller voltage range with a finer resolution. As shown in section 10.2.2 the DACs deviate from their ideal behavior and show strong variations with worst-case step sizes of about 100 mV for some channels. Therefore a characterization is necessary and unless the complete voltage range is needed, the slope should be reduced to proportionally lower the worst-case step size.

The input stage and bias voltage adjustment is shown in figure 7.1. The SiPMs are directly connected to the EASIROC without further external preamplifiers.

Each SiPM signal is amplified by two parallel preamplifiers, one low-gain and one

[2] 100 nF for the evaluation boards produced by Ωmega

high-gain. The gain of the preamplifiers is variable through selection of different feedback capacitors with a nominal gain of 1 to 15 for the low-gain and 10 to 150 for the high-gain preamplifiers. The gain can be selected with 4 bits for each gain path, with all channels sharing a common gain value.

In the trigger path, the high-gain preamplifier is then followed by a bipolar fast shaper with 15 ns nominal peaking time for each channel. The fast shaper output for each channel is compared to a common threshold (produced by a 10-bit DAC, called DAC10) by discriminators to generate digital trigger signals. After an optional latch circuit, all 32 channel triggers are individually fed to trigger output pins for external evaluation as well as to a wide OR-gate, which combines them to a single trigger output *OR32*. Using the latch circuit and a separate multiplexed digital output, the trigger state between two clears of the latches of all channels can be read sequentially with only one single output line.

In parallel to the trigger path, each channel also features two slow shapers, one after the high-gain preamplifier and one after the low-gain preamplifier. The nominal peaking time[3] can be selected between 25 ns and 175 ns in steps of 25 ns, separately for both gain paths (but not for each channel). The slow shaper outputs (characterized in section 10.2.4) are fed into an analog track and hold capacitor, which can be decoupled from the shaper through external driving of a hold pin of the EASIROC. This allows for one point of the slow shaper time trace to be stored. For optimal results, the hold signal should be driven such that it coincides with the peak of the slow shaper signal. The discriminator trigger outputs can be used together with an external delay adjusted for the chosen shaping time to achieve this. The content of the track and hold cells can be read out by external ADCs (*Analog to Digital Converter*) on the EASIROC board using two multiplexed analog outputs. A *read* shift register in the EASIROC is driven by an FPGA on the same board[4] to sequentially select which channel analog value is provided on the analog multiplexer output. The readout of a complete event takes about 1 µs, which is the dead time of the EASIROC for the recording of the next analog value.

For debug and calibration purposes, a common calibration input can be connected to the amplifier inputs of the channels, which is a feature not used in this thesis. Furthermore a probe output exists, that allows to monitor the signal at specific points in the signal chain for a single channel (after the preamplifiers or after the shapers). As the output has a very slow rise time[5] and also changes the signal being looked at, it is not suited for characterization purposes or during normal operation but just for debugging.

The configuration of the EASIROC functions (*slow control*) is performed using a shift register in the chip. It consists of multiple chained D flip-flops which are connected

[3] "Peaking time" is not defined properly in the datasheet, but can be assumed to be the time from the start of the rising edge to the maximum.

[4] Altera Cyclone EP1C6 (see section 6.1.1) for the EASIROC evaluation board (see section 9.1.3) as used for the AMD.

[5] low-power

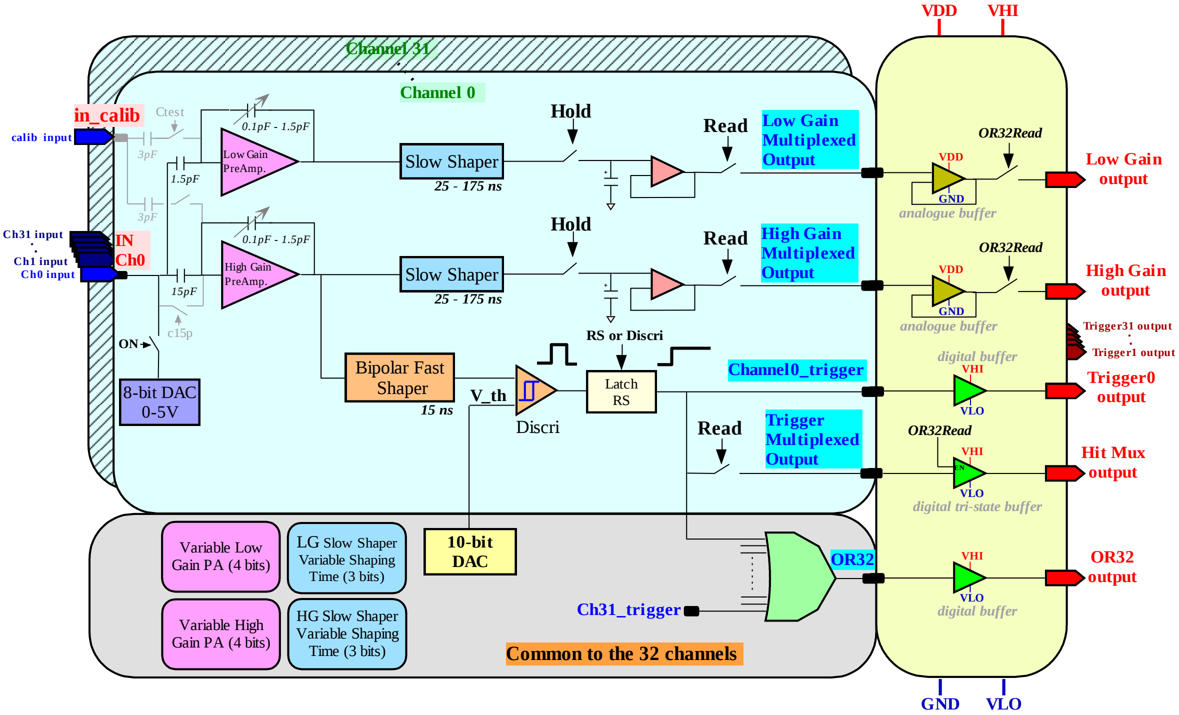


Figure 7.2.: Schematic view of the EASIROC with details of one channel [30].

to a common external clock input pin and the first D input is connected to a data input pin. Through this the configuration can be written one bit at a time. After the complete configuration is written, a load signal has to be asserted to cause the changed configuration to become effective[6].

To drive the control lines for the readout of the analog values using an ADC, to drive the hold line and to write the configuration bits to the EASIROC, an external FPGA on the EASIROC board is needed for proper operation. The EASIROC is prepared for high density applications, where all multiplexed outputs of multiple EASIROCs can be connected in parallel and only the desired real-time trigger outputs (the trigger outputs for the single channels and/or OR32 outputs) have to be connected to the FPGA and two external ADCs are sufficient to readout a big number of EASIROC chips[7]. The slow control shift registers and the shift registers for source selection of the analog multiplexers in the different chips can be daisy chained and therefore do not need multiple separate lines from the FPGA.

A schematic view of the EASIROC is shown in figure 7.2.

[6] There are some slow control bits that become effective immediately during write without loading first, which causes the controlled functions to change state multiple times during configuration as all configuration bits pass through the shift register.

[7] This increases the time needed for a complete readout.

8. The Aachen Muon Detector – AMD

8.1. Motivation

The *Aachen Muon Detector (AMD)* was started as a possible muon detector upgrade for the Surface Detector of the Pierre Auger Observatory. It was first described in an internal technical note [43] of the Pierre Auger Collaboration and later also presented in [66] and [50]. Although the SSD (see section 3.4.3) was chosen as the upgrade for AugerPrime (section 3.4) and the AMD is therefore no longer an option for AugerPrime, the project and the prototype tests were nevertheless continued to prove the concept for the application in future experiments. The experience gained, especially in scintillator handling and simulation, are now used for a new project called MiniAMD which reuses parts of the AMD technology.

8.2. Detector concept

The AMD is designed to be placed below the Auger SD stations to use the water of the SD tank as shielding against electromagnetic particles, and detects muons using scintillator tiles and SiPMs. Compared to being buried next to the tank this has the distinct advantages of not requiring digging work during deployment and being able to detect the same muons in the tank as well as in the muon detector. Compared to the finally chosen SSD concept, it has the advantage of producing a more pure muon signal. It therefore relies less on advanced analysis techniques to determine the actual muon number through an inversion of the response matrix of both detector components[1] to muons and electromagnetic particles. The main downside of the concept is that during deployment the water in the tank has to be removed so the station can be lifted onto the AMD. Furthermore the covered area is limited by the SD tank and can not be enlarged. The detector is placed into two steel housings to support the 12 t water tank. Two half detectors are used for easier deployment. Both detector halves are self-contained with the possibility to build a combined trigger.

A half detector consists of 32 scintillator tiles with a size of 30 cm × 30 cm × 0.5 cm each (see figure 8.1). The tiles are placed in trays of eight tiles with four trays per

[1] SD tank and muon detector upgrade

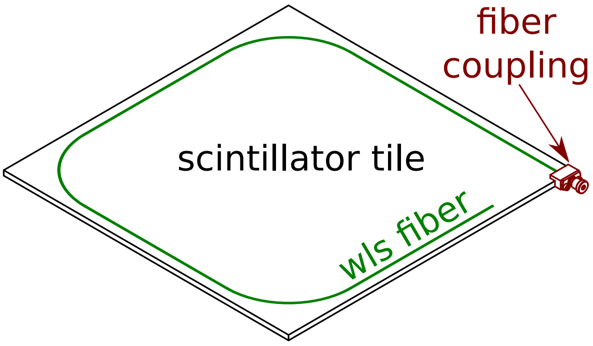


Figure 8.1.: Schematic view of an AMD scintillator tile with embedded wavelength shifting fiber and installed fiber coupling to connect a clear fiber.

steel housing. Embedded into sigma-shaped grooves in the tiles are wavelength shifting fibers to collect the scintillation light for detection. The light is then further guided onto SiPMs being placed at one end of the trays using clear polymer optical fibers to reduce the absorption losses between the tiles and SiPMs, compared to longer wavelength shifting fibers. This is especially important as the lengths of these fibers differ substantially between the tiles, but nevertheless a similar light output per muon has to be achieved. Placing all SiPMs together at one end of the detector at the one hand reduces the temperature variability and amount of temperature sensors needed to achieve proper temperature compensation of the SiPMs and at the other hand shortens and unifies the length of the signal cables transporting the relatively weak analog SiPM signals for digitization. Even though it turned out that longer cable lengths would be possible (see section 10.2.5), this was not yet known during detector design.

A schematic view of the Aachen Muon Detector can be seen in figure 8.2.

The SiPM signals of the tiles are digitized individually using two different simultaneous approaches:

- *Counting Method:* Counting the number of crossing muons using discriminators for each channel. This works for events where the muon flux in the respective detector station is low, i.e. not all tiles are hit and multiple muons per tile are a rare occurrence. As long as not all tiles are hit, it is possible to calculate a correction for double-hits.
- *Integration Method:* Generating and digitizing charge integrals for each tile for showers where the muon flux is high and many tiles are hit by multiple muons. Because of fluctuations of the light produced per muon, this method is actually less accurate than the counting method for low muon hit rates.

Due to constraints on power consumption and available data rate, it is not possible to install and run hardware to continuously digitize voltage traces of all SiPMs using FADCs. Only time traces of the discriminator status per tile and one integral value per channel and event can be recorded. A power budget of about 1 W per half detector was the original design goal.

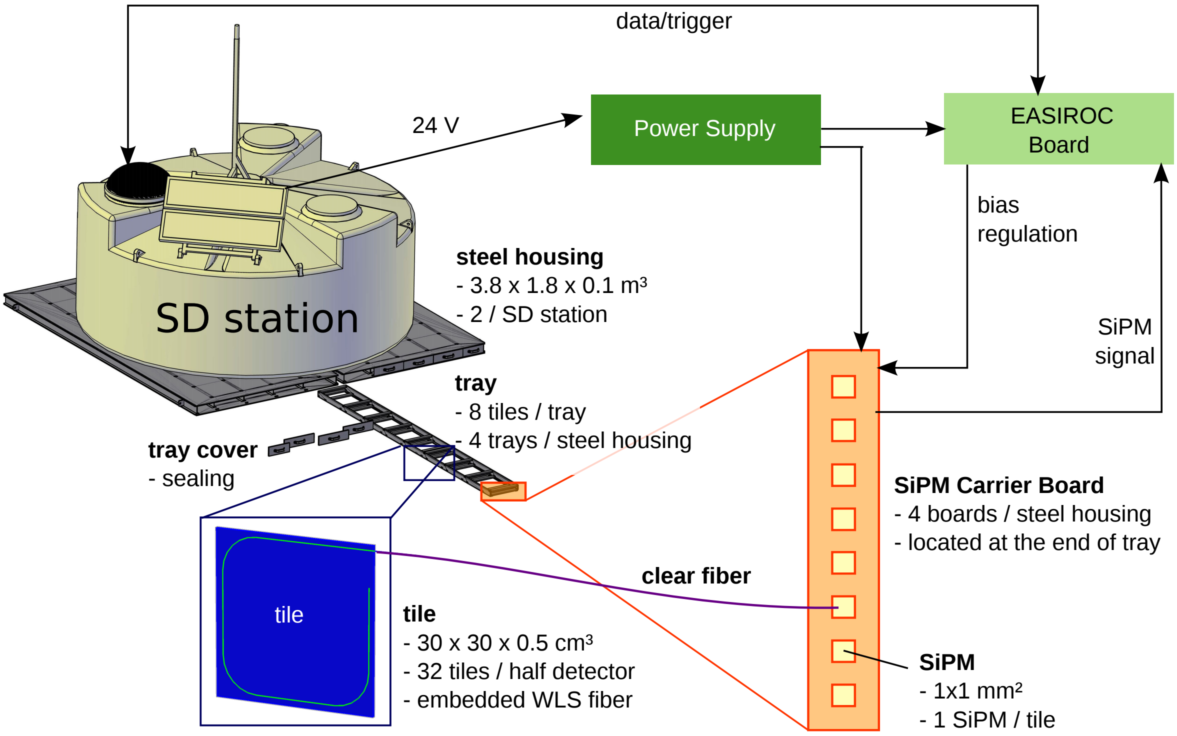


Figure 8.2.: Schematic overview of the AMD, together with the SD station. Adapted from [58], [66], [50].

8.3. Mechanical description

The steel housings of the AMD each cover an area of 3.8 m×1.8 m and have a thickness of 11 cm. Two housings next to each other can carry the 12 t SD station, which has a diameter of 3.6 m. The steel housing is a welded construction built of two steel sheets on the top and bottom held together with I-beams. Four crane splices are attached at the sides to lift the housing. The 10 cm high interior of the housing is divided into five compartments by the I-beams. Four compartments have a width of 37 cm to carry the trays with scintillating tiles and one has a width of 26 cm. The small compartment is placed at the outer edge of the detector and can be used to house DAQ electronics.

The ends of the steel housing can be sealed with aluminum panels using silicone. They have to provide light-tightness for SiPM operation as well as water-tightness. The enclosure has to be closed in the field during deployment and can be reopened in case maintenance is needed[2]. A space of about 6 cm exists between the inner I-beams and the cover panels to allow cable connections between the trays and the DAQ electronics.

The trays have a length of 3 m and a width of 35 cm, to house eight scintillator tiles and one SiPM carrier board. The carrier board is located at the front of the tray, with the SiPMs facing towards the scintillator tiles and the signal cables leading out. The

[2] By design the AMD does not require regular maintenance.

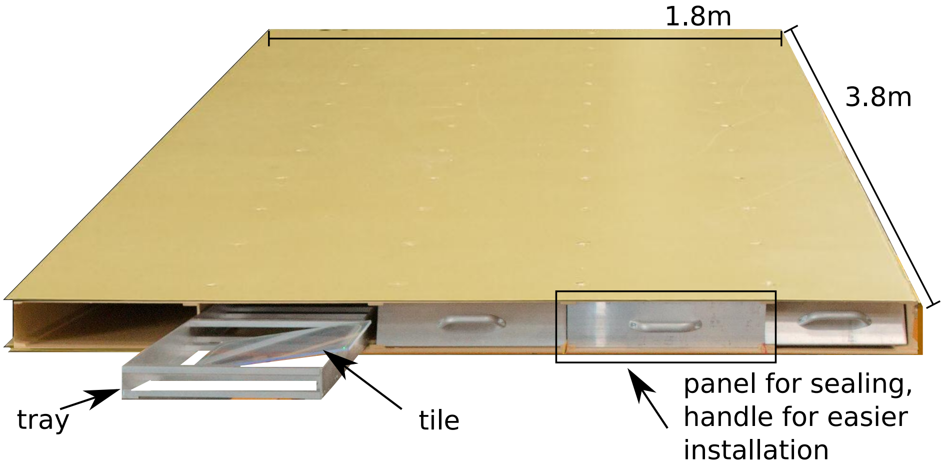


Figure 8.3.: Photo of the AMD prototype housing. One tray is partially inserted with one unwrapped tile for demonstration purposes. Three cover panels are also placed at their respective places (but not closed properly). Adapted from [58].

clear optical fibers, which have a length between 0.3 m and 2.9 m, are guided at the side of the tray and onto the SiPMs. The SiPM carrier board, housing the SiPMs with $1 \times 1\,\mathrm{mm}^2$ active area, is described in section 9.1.2. The tray is made of aluminum and can be inserted into and retrieved from the housing using an integrated handle.

A (cropped) photo of the prototype housing with some of the other components can be seen in figure 8.3.

The scintillator tiles with a size of 30 cm × 30 cm × 0.5 cm each (see figure 8.4) have a wavelength shifting fiber glued into a milled groove. The fiber has a diameter of 1 mm and follows a sigma-formed shape with a distance of 10 mm from the edges and a corner radius of 65 mm. It has been shown by simulation studies [84] that this fiber layout leads to a nearly optimal and homogeneous light output. The fiber protrudes from the tile by about 5 mm and is encased by a fiber coupling, affixed to the tile by two screws. The fiber coupling uses a collet to hold the clear optical fiber and precisely position it on the end of the wavelength shifting fiber. The quality of this coupling has been studied in [58] and [52].

The fiber end is reflective, to guide also the light traveling into the opposite direction in the fiber towards the SiPM, to achieve almost twice the light output compared to losing this light. It has been shown in [52], that the method used for the prototype tiles resulted in a non-reflective end. Therefore, the tiles used for the measurements in this thesis only have a much reduced light output compared to the original expectation.

For ideal performance and for reduced crosstalk between the tiles, they have to be wrapped. During this thesis multiple different methods and materials have been used and tested. The performance differences between the tested materials are negligible, with sheets of Tyvek paper currently being the easiest-to-apply method.

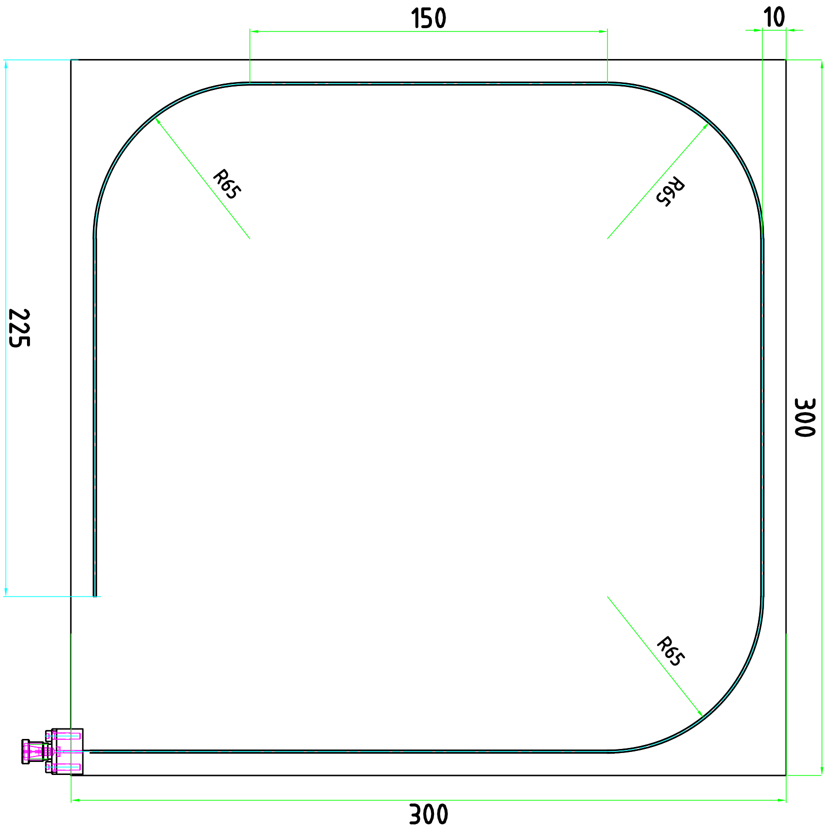


Figure 8.4.: Technical drawing of an AMD tile with the fiber coupling installed. All dimensions given in mm.

8.4. DAQ

The DAQ is based on the EASIROC ASIC (described in chapter 7), which allows to achieve both mentioned digitization approaches for 32 SiPMs with low power consumption. The counting method can be performed time-resolved for long time periods, only limited by the available data buffer memory[3]. For the integration method only one single sample per event and channel can be measured from a shaped signal, due to the long dead-time of the readout. The maximum nominal peaking time of the shaper of 175 ns is also the approximate time after event start when the shaper output has to be sampled[4]. Due to the slow rise of the shaper output, muons arriving at different times after the start of the event contribute to the sampled value with a variable weight (see also section 10.2.4). Only muons arriving in the first $\approx$ 100 ns have a significant contribution. The exact influence of the shaping on the muon number determination and the optimal time of sampling (hold delay) will be discussed in [65]. The DAQ, except for the SiPMs and temperature sensors near the SiPMs in the trays, is located in the smaller fifth electronics compartment of the housing. The complete DAQ is in detail described in chapter 9.

8.5. (Expected) performance

Extensive simulations of the AMD have been performed for the response of tiles to single muons, as well as for the response of the complete detector to extensive air showers. The simulation of the AMD or the single tiles is performed by Geant4 [7], while the SiPM response is simulated by G4SiPM [62], [61]. First simulations were presented in [43], additional simulation studies were performed in [61] (for a different SiPM type) and [84]. Further studies will be presented extensively in the doctoral thesis of Christine Peters [65]. None of the simulations shown here have been performed by the author of this thesis.

According to simulations during design, a single crossing muon leads to a most probable value of between 30 and 40 photons detected by the SiPM (including crosstalk). As expected, the distribution of the number of photons mainly follows a Landau distribution. The simulated distribution of the number of detected photons is shown in figure 8.5. A trigger threshold of about 10.5 p.e., which leads to about 10 Hz of random noise triggers, was simulated to have a trigger efficiency of better than 99 %. Since an irreducible background of about 15 Hz from random muons from low energy showers[5] exists, it is not sensible to apply a higher trigger threshold at the expense of a lower muon trigger efficiency to reduce the dark noise trigger rate significantly below 10 Hz.

[3] For the EASIROC evaluation board used by the AMD, about 1000 sampling points can be stored for each channel.

[4] The *hold* signal of the EASIROC is driven

[5] Estimated from a muon flux of $1\,\mathrm{cm}^{-2}\,\mathrm{min}^{-1}$ for a horizontal detector [63, chap 28] and an area of $30\,\mathrm{cm} \times 30\,\mathrm{cm}$.

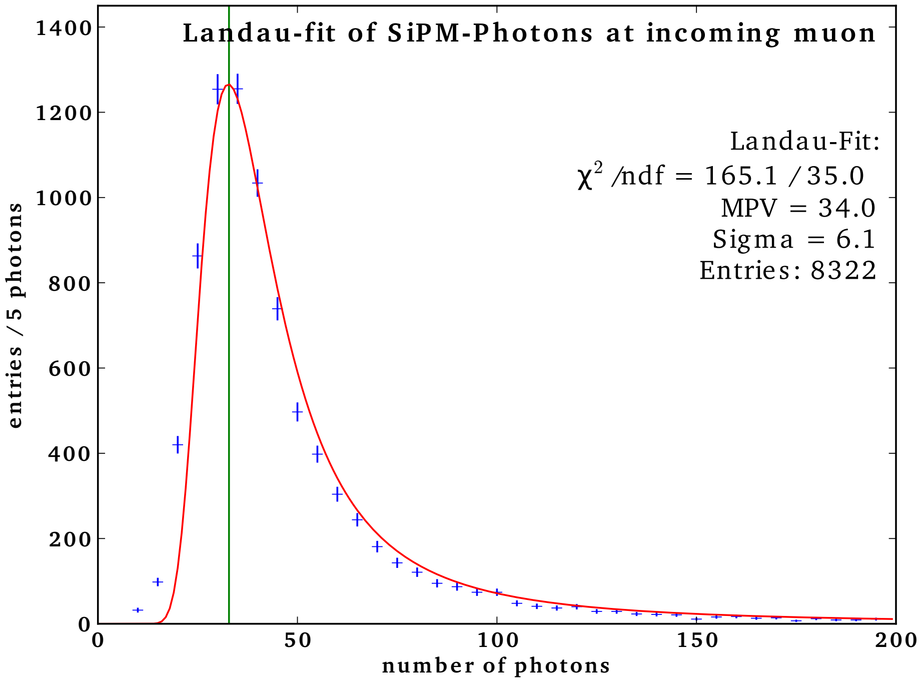


Figure 8.5.: Simulated distribution of the number of detected photons for a single muon crossing a tile. A fit of a Landau distribution is shown to describe the data, though not perfectly. The mismatch is likely caused by SiPM crosstalk, photon losses modifying the distribution and the distribution itself being only an approximation to the original distribution of produced photons. Taken from [84].

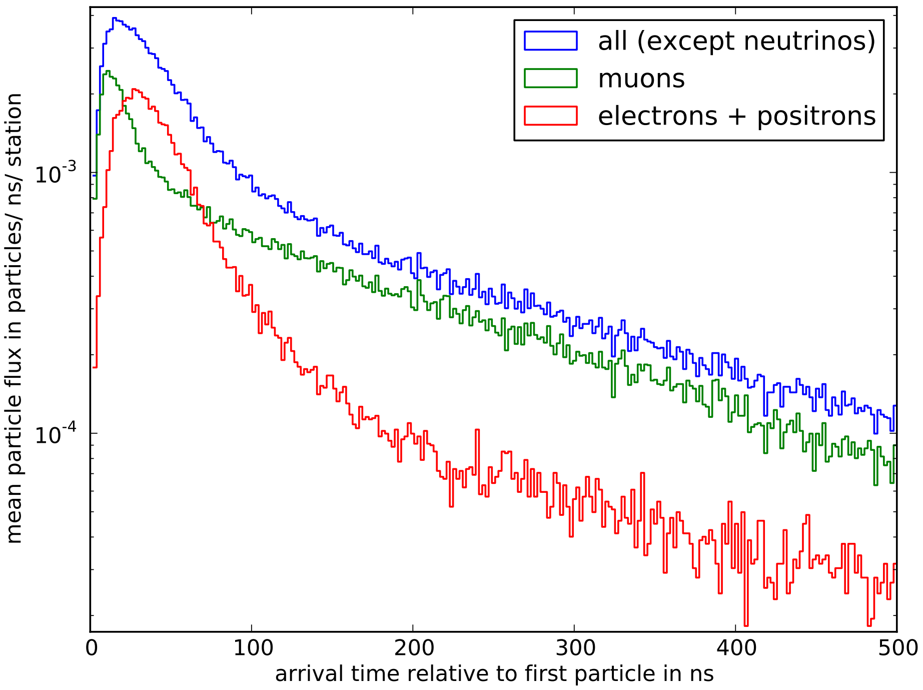


Figure 8.6.: Simulated arrival times of shower particles with energies above 10 MeV at the AMD below an SD tank. The first particle hitting the detector defines the time $t = 0$. Shower simulation for 10 proton primaries with energy of $10^{19.5}$ eV and inclination 40°. The highest muon rate only happens in the first ≈ 60 ns. First shown in [43].

According to simulations, the highest muon density at a detector station for air showers occurs in the first 60 ns after the first shower particle hits the station, with a peak at about 25 ns (see figure 8.6). This is important as the integration method works only in the first part of the shower. As most muons arrive during that time and after 100 ns only relatively few particles arrive, the counting method should be sufficient for the later parts of the shower.

8.6. Applications of AMD technology

There are various variations of the AMD detector technology already in use. These have been either built for easier characterization of the AMD or are used for purposes outside of the scope of the AMD.

8.6.1. The AMD tile container

To allow the characterization or usage of a single tile in a high-light environment, a special light-tight container for a single tile with SiPM has been constructed. Here one of the usual wrapped tiles is connected to a single SiPM similar to the ones used

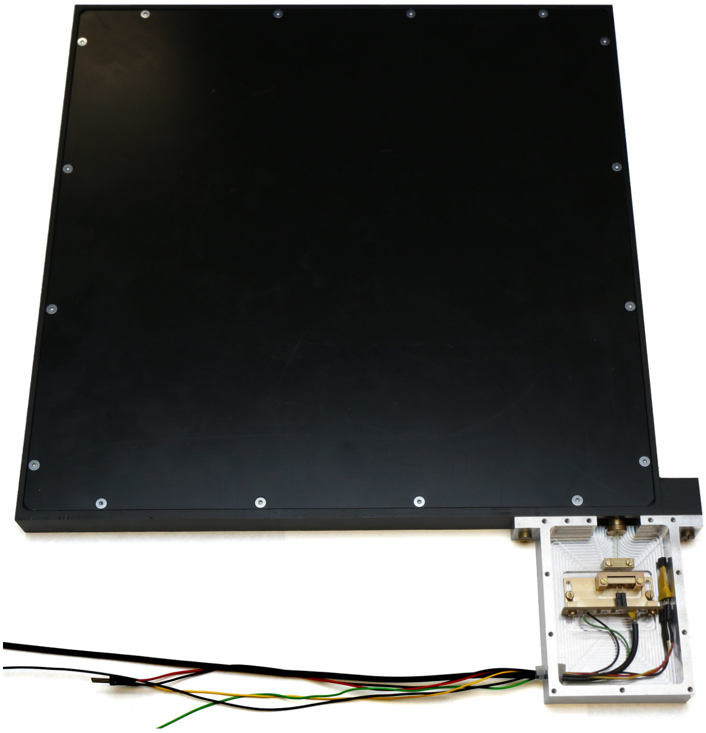


Figure 8.7.: The AMD tile container with opened SiPM compartment. One AMD scintillator tile is enclosed in the flat light-tight black box. One SiPM and temperature sensor is placed in the attached aluminum box (also light-tight during normal operation).

in the AMD with a short optical fiber. The container is shown in figure 8.7. The SiPM compartment also contains a temperature sensor. The tile container has electrical connections similar to the ones of the normal AMD to be connected directly to the AMD DAQ. The SiPM is of a through-hole type[6] instead of the surface mounted devices in the normal AMD and easily exchangeable, to allow testing of different types of SiPMs. Placing the tile in a different detector providing a trigger and position resolution, the homogeneity of the tile can be measured. Multiple tiles together could be used as a trigger for different experiments.

8.6.2. The AMD shelf

Mainly to test the AMD system and to characterize the tiles, the *AMD shelf* has been built. In it up to eight tiles are vertically arranged together with one AMD SiPM carrier board with eight SiPMs. Also included is one of the EASIROC boards and a power supply board (see section 9.1.1). It has a closed top and bottom, which are connected with vertical bars at the four corners and the center of the long edges. The enclosed space can be covered at the sides to provide the low light-levels needed for operation. A photo of the shelf can be seen in figure 8.8.

[6] with conventional wire leads

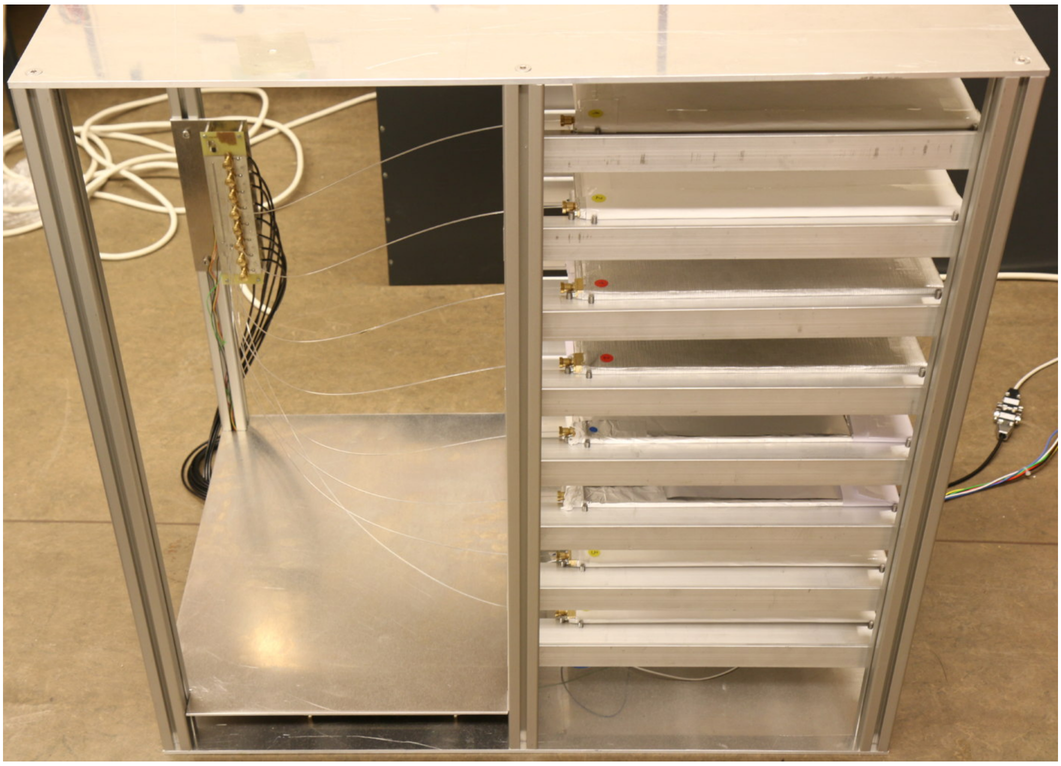


Figure 8.8.: The AMD shelf with a vertical tile arrangement used for calibration and test measurements. Eight tiles are installed and connected to the SiPMs on the carrier board via 0.5 m optical fibers. The DAQ electronics is hidden below a protective aluminum board at the bottom.

8.6.3. AMD tiles as muon trigger

To do characterization measurements of different scintillator detectors using cosmic muons, often an external trigger is needed. This trigger can be provided by AMD tiles arranged above and below the detector to be characterized. The AMD DAQ features trigger outputs that can be used, as well as special trigger modes for exactly this purpose, which can be enabled in the firmware. These specially implemented triggers take a coincidence of one of the scintillator tiles above and one of the scintillator tiles below the detector to be characterized, with the trigger threshold for the tiles to be determined by the needed purity and trigger rate. It is also possible to use only a single layer of tiles for a greater area coverage with a given number of tiles at the expense of a less pure muon trigger. Like for the normal AMD, light-tightness of the setup has to be ensured. The AMD tiles and DAQ have been used successfully in this operation mode for extensive characterization measurements of parts for the SSD (section 3.4.3).

8.6.4. MiniAMD

An offspring of the AMD project is MiniAMD. It is an idea for a calibration detector with position resolution, that can be deployed as two modules around the SSD detector. 2×4 tiles each are placed in a thin light-tight housing, with one module being placed above and one below the SSD detector. A drawing of one MiniAMD module is shown in figure 8.9. The free space in the housing between the tiles is filled with extruded polystyrene foam, and together with the thin aluminum walls of the housings, each module has an estimated weight of below 30 kg. In contrast to the original AMD concept, the SiPMs with temperature sensors are placed directly at the tiles, forgoing the clear optical fibers and their couplings. This reduces costs for the fiber couplings, simplifies the design and results in a higher light yield due to reduced photon losses. Signal cables of around 2.5 m lengths, whose operation has been verified to be possible (see section 10.2.5), connect the SiPMs to the rest of the DAQ electronics including the EASIROC which is located in a separate electronics box. For completely autonomous data taking, also an ARM single board computer is included, which leads to a total power consumption of the MiniAMD of around 6 W. For better performance and more tolerances of the SiPM and fiber placement, newer SiPMs with $1.3 \times 1.3\,\mathrm{mm}^2$ active area and lower crosstalk are used.

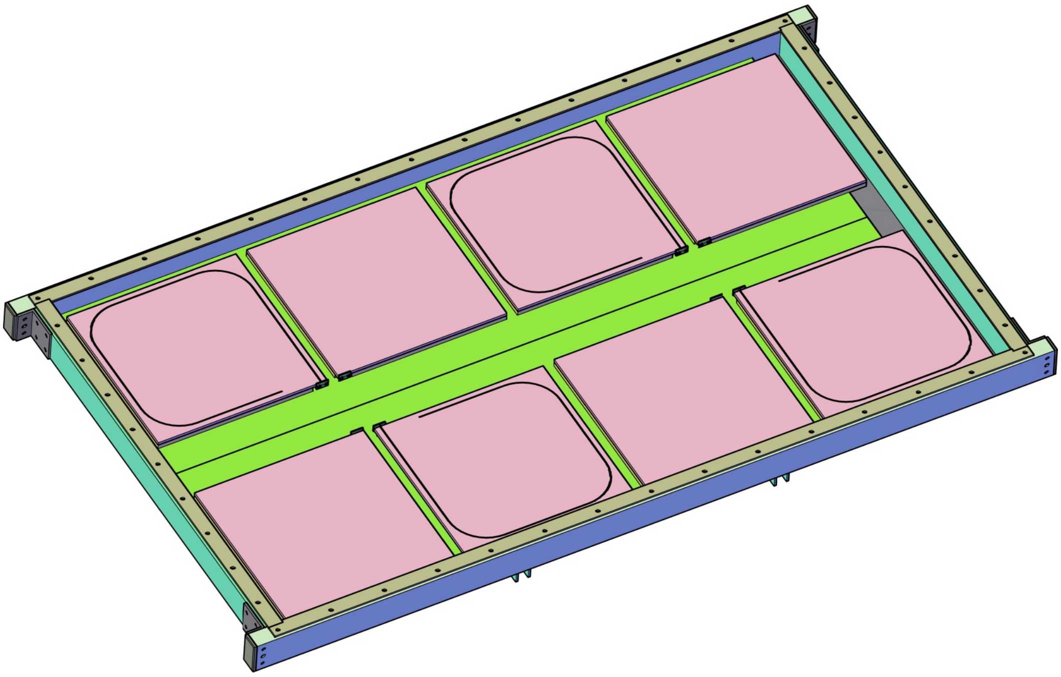


Figure 8.9.: Schematic image of one of the MiniAMD module with the top cover removed. Eight scintillator tiles with one SiPM located directly at each tile are placed in a light-tight and light-weight housing. The SiPM signal is fed to an EASIROC placed in a separate box. Taken from [50].

9. The DAQ of the AMD

The DAQ system of the Aachen Muon Detector consists of the hardware based on the EASIROC ASIC, firmware running on an FPGA on the EASIROC board and software running on a PC or the processor in the main SD station electronics. A sketch of the system is shown in figure 9.1.

9.1. Hardware

The electronics of the AMD consist of three parts:

- The Power Supply Unit (PSU) (described in section 9.1.1) that converts the battery-provided 24 V into 6.5 V usable by the EASIROC board and ≈ 70 V for the SiPMs.
- SiPM carrier boards (described in section 9.1.2) that house the SiPMs, bias voltage filtering and temperature sensors.
- The EASIROC board (described in section 9.1.3) that houses the EASIROC ASIC, an FPGA and some peripherals to perform triggering and digitization of the SiPM signals.

Also needed is an external PC or microcontroller to perform additional slow control calculations, do steering and store or relay measurement data. The software is described in section 9.3. The firmware running on the FPGA on the EASIROC board is described in section 9.2.

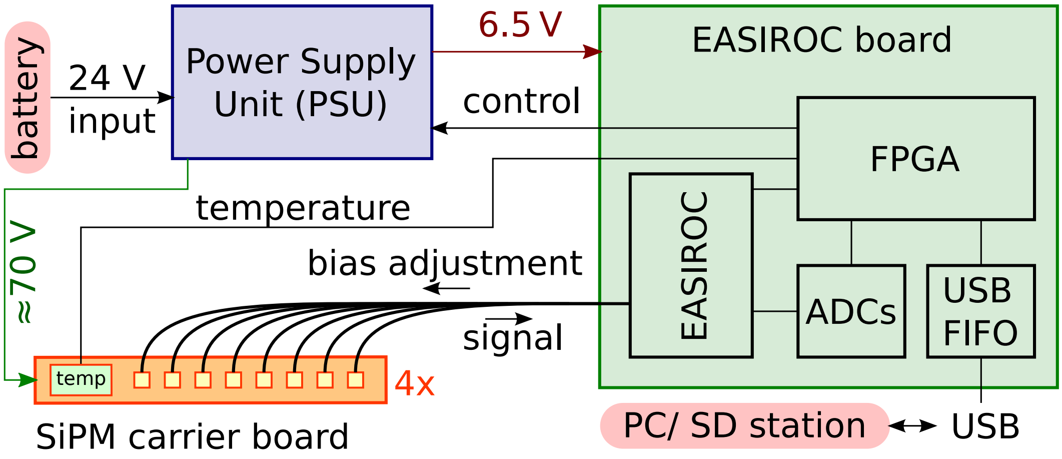


Figure 9.1.: Schematic structure of the AMD DAQ.

The described components comprise one half of a complete AMD. Both detector halves are connected to synchronize and exchange triggers, but otherwise operate independently of each other.

9.1.1. The Power Supply Unit – PSU

The core AMD DAQ is powered by a single 24 V supply. This is the voltage provided by the Auger SD batteries, but for the prototype tests it is produced by a laboratory power supply, since an external PC with high power requirements (instead of the SD electronics) is used anyway to perform the external calculations, event data storage and transmission.

The Power Supply Unit converts the 24 V into 6.5 V usable by the EASIROC board and ≈70 V for the SiPMs. 6.5 V is generated by a *TRACO POWER TSR 1-2465* [81] step-down converter. The resulting voltage is further reduced to 3.3 V by a *TRACO POWER TSR 1-2433* [81] step-down converter to power the other ICs on the PSU board.

The SiPM supply voltage is generated in two steps. First, a self-made step-up converter based on an *MSP430 F2013* [78] microcontroller is used to generate a voltage of about 75 V with low accuracy and high ripple. Then this voltage is reduced further to the desired voltage by a transistor and an op-amp based regulation circuit. This results in a stable voltage nearly free of ripple. This high voltage output is supposed to provide about 100 µA, which is more than enough to operate 32 SiPMs while also providing the 10 µA needed to measure the voltage with a 10 MΩ multimeter. Each SiPM draws less than 0.1 µA current in dark conditions at room temperature. The additional current caused by the average light flux from the scintillators can be neglected compared to the dark noise.

The PSU board is also prepared to provide about 5 V to a separate low-power data logger board for environment measurements including a charge controller for a dedicated NiMH battery for this circuit. This data logger is not part of the AMD design but has been foreseen for the prototype phase to evaluate the temperature and humidity in the AMD housing.

The board has been designed and produced in the electronics workshop of the Physics Institute III A at the RWTH Aachen University.

9.1.1.1. Version 1

The first iteration of the PSU board (shown in figure 9.2) allows for an adjustment of the SiPM bias voltage in a range between roughly 65 V and 73 V using a potentiometer. Due to the lack of a dedicated high precision reference voltage supply and due to temperature dependencies in the used parts (resistors, potentiometer), it features a temperature dependence of the output voltage of between 11 mV/K and 16 mV/K as shown in figure 9.3. This temperature dependence is not exactly linear but changes

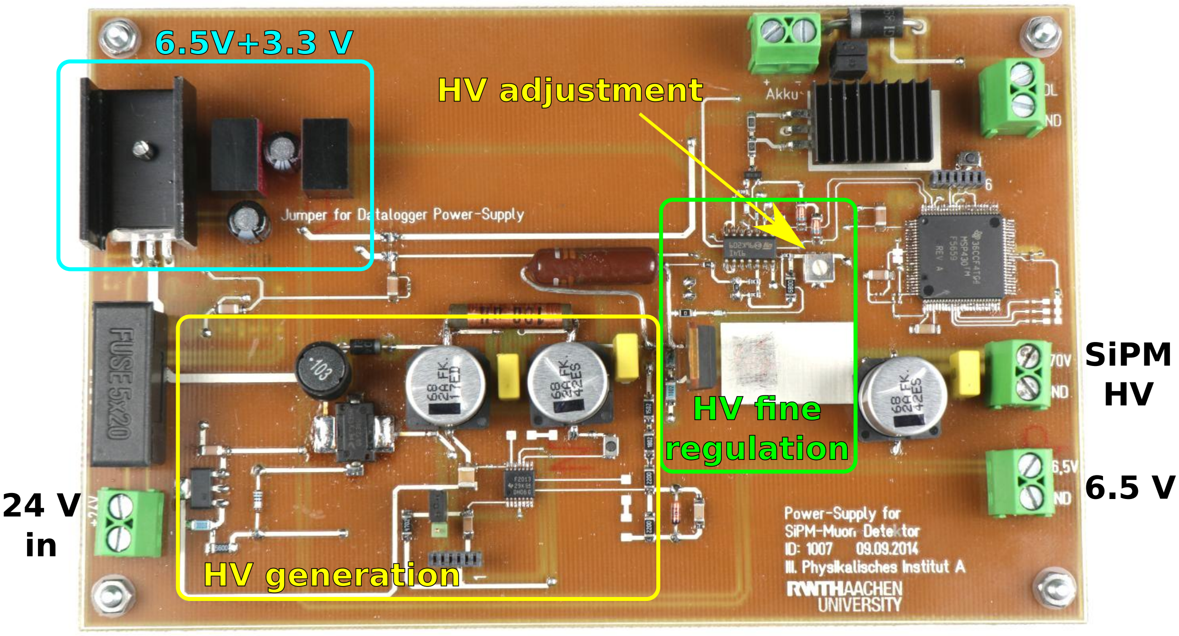


Figure 9.2.: PSU board version 1. Voltage input for +24 V is on the left, outputs for +6.5 V (bottom) and +70 V (center) are on the right. The top right part can supply 5 V to a stand-alone data logger. The potentiometer can be used to adjust the output voltage for the SiPM.

in slope at about 15 °C for unknown reasons. Some early EASIROC characterization measurements using SiPMs (see section 10.2) have been performed with this board together with a Fluke 8845A [35] digital desktop multimeter to monitor the exact SiPM supply voltage. To achieve a slow voltage ramp up and ramp down and to allow disabling of the voltage without powering down the complete setup, an external board with manually switchable RC circuits is used during characterization measurements.

9.1.1.2. Version 2

The second iteration of the PSU board (shown in figure 9.4) features a greatly reduced temperature dependence and the possibility to adjust and shut down the SiPM supply voltage remotely. To achieve this, an extra *MSP430 F5659* [79] microcontroller with integrated DAC is installed to provide the reference voltage for the regulation circuit. The desired value can be set between 65 V and 72.5 V in steps of 0.5 V using a jumper on a set of pins. A second jumper has to be used on a separate pair of pins to enable the voltage output.

Through communication lines connected with the FPGA on the EASIROC board, the desired voltage can be selected remotely. Four pins are used to select the 16 possible output voltages. The enable pin can be driven by another line to the FPGA instead of using the jumper. The communication lines are active low. To set a bit or enable

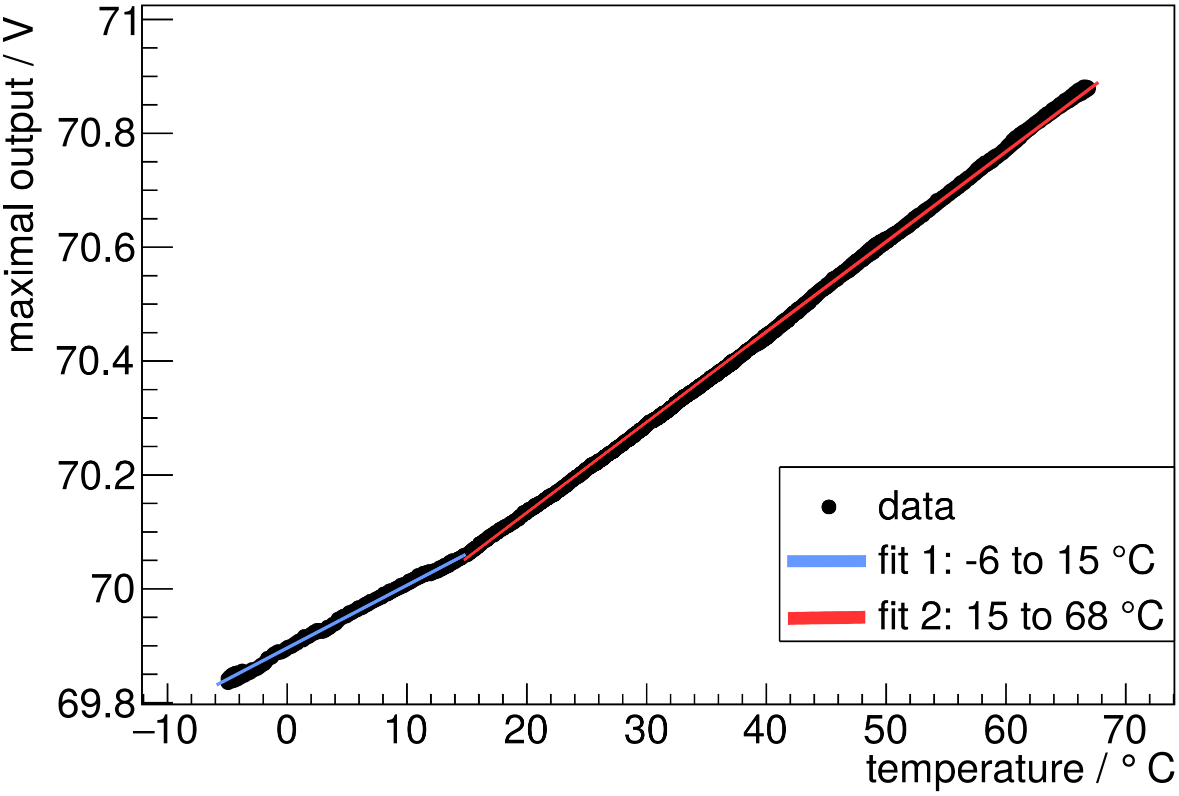


Figure 9.3.: Temperature dependence of the output voltage of the PSU board version 1. Taken from [58].

the voltage, the pin has to be pulled down (from 3.3 V). The FPGA interface pins are only used when no jumper is installed for voltage selection.

The output voltage ramps up and down automatically with about 3 V/s to prevent damage to the EASIROC input DACs or preamplifiers.

Shown in table 9.1 are calibration measurements for the different nominal voltages of the PSU board version 2, measured at 30 °C. The voltages have been measured using a FLUKE 8845A multimeter (accuracy for the given range about 4 mV) at maximum internal averaging and read manually from the front display to be the center of the observed interval of about 2 mV width. The average output voltage is about 10 – 20 mV above the nominal voltage. Ripple with about 20 mV amplitude and about 1 kHz frequency has been observed with an oscilloscope.

Over a range of −20 °C to 60 °C, the output voltage changes by a maximum of 25 mV, which is less than the random voltage ripple and corresponds to an SiPM temperature uncertainty of less than 0.5 °C, an SiPM gain uncertainty of about 1 % and is similar to the accuracy of the EASIROC DACs.

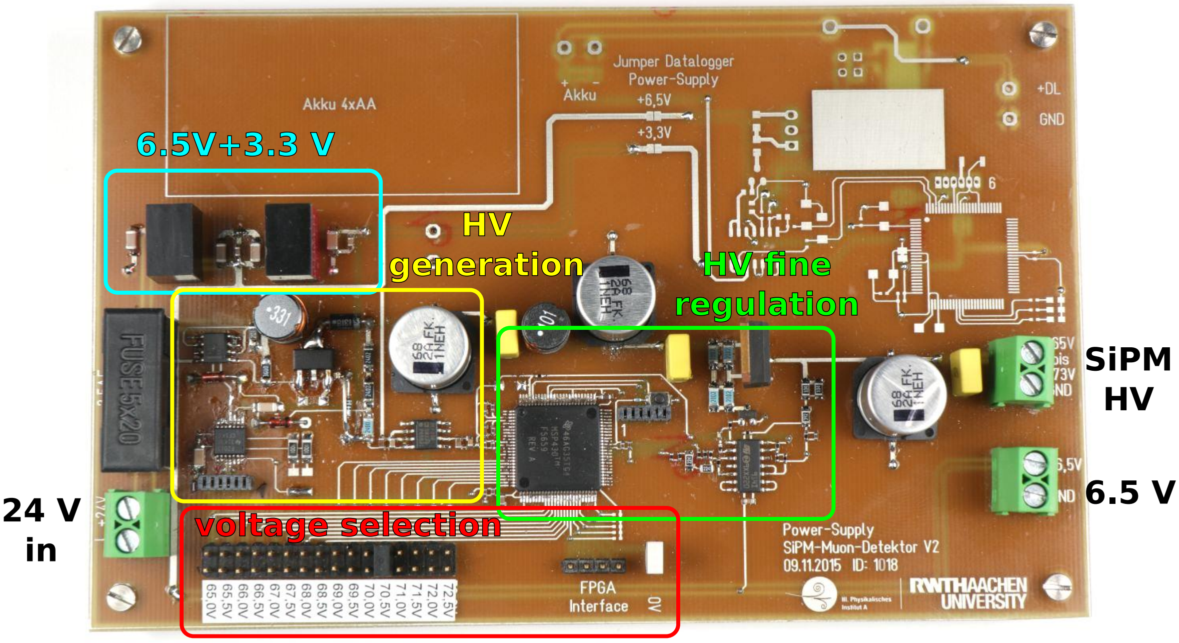


Figure 9.4.: PSU board version 2. Voltage input for +24 V is on the left, outputs for +6.5 V (bottom) and the SiPM supply voltage (center) are on the right. The unpopulated part can be used to supply a stand-alone data logger. The pin headers at the bottom allow to select the SiPM supply voltage.

nominal	measured in V	nominal	measured in V
65.0 V	65.0155	69.0 V	69.0116
65.5 V	65.5120	69.5 V	69.5126
66.0 V	66.0126	70.0 V	70.0138
66.5 V	66.5122	70.5 V	70.5146
67.0 V	67.0126	71.0 V	71.0164
67.5 V	67.5126	71.5 V	71.5172
68.0 V	68.0136	72.0 V	72.0178
68.5 V	68.5149	72.5 V	72.5180

Table 9.1.: Voltage calibration of the PSU version 2 at 30 °C

The consumption of the PSU alone (no EASIROC board connected) with disabled SiPM bias voltage output is 30 mA, which corresponds to 720 mW. With the SiPM bias voltage set to 70.5 V (nothing connected), a supply current of 50 mA (1.2 W) can be measured.

With a connected EASIROC board, the complete power consumption is about 3.9 W.

The version 2 board has been used for most of the EASIROC characterization measurements in chapter 10.

To increase tolerance to a short circuit of the high voltage output, which could destroy the regulation circuit and cause the complete voltage of the step-up converter to be provided at the output, an additional resistor has been included before the voltage fine regulation circuit. This resistor also limits the maximum current at which the full output voltage can be provided due to reducing the voltage at the input of the fine regulation circuit, but it does not achieve a hard current limitation. Such a resistor has been added as a modification to the version 1 and version 2 boards.

9.1.1.3. Version 3

The third iteration features further improvements and enhanced functionality. Due to reliability problems[1] and relatively high noise and ripple produced by the MSP430-based step-up converter, as well as its high idle power consumption, it has been replaced by the commercial step-up converter AdvancedEnergy/UltraVolt 0.1XS5-P0.1 [6]. With this new step-up converter and optimizations of the linear fine regulation circuit[2], the power consumption of the PSU itself could be reduced significantly to 130 mW with no output and 165 mW with 70.5 V output voltage. The influence of the output current (≤100 µA) on the power consumption is small. The total power consumption is dominated by the EASIROC board. More power consumption values are given in table 9.2.

load at 6.5 V	HV	input voltage	input current	power consumption
no	0 V	12 V	10.9 mA	130 mW
no	0 V	24 V	7.3 mA	175 mW
yes	0 V	12 V	210 mA	2.5 W
yes	0 V	24 V	110 mA	2.6 W
no	50 V, 5 µA	12 V	12.5 mA	150 mW
no	50 V, 5 µA	24 V	8.4 mA	200 mW
no	50 V, 104 µA	12 V	13.2 mA	159 mW
no	50 V, 104 µA	24 V	8.8 mA	210 mW
no	70.5 V, 7 µA	12 V	13.8 mA	165 mW
no	70.5 V, 7 µA	24 V	9.1 mA	218 mW
no	70.5 V, 78 µA	12 V	14.5 mA	174 mW
no	70.5 V, 78 µA	24 V	9.5 mA	230 mW

Table 9.2.: Power consumption of the PSU V3 at different operating points

Also included into the new design is a wider output voltage range[3] to facilitate the operation of SiPMs with a lower operation voltage. With minor firmware adjustments the selectable output voltage range could be changed further as needed, even

[1] The microcontroller of the step-up converter does not always start operation on power-on, especially at low temperatures.

[2] Usage of a FET instead of a bipolar transistor

[3] 50 V-73 V by jumper or 10 V-73 V through FPGA communication interface, each in steps of 0.5 V

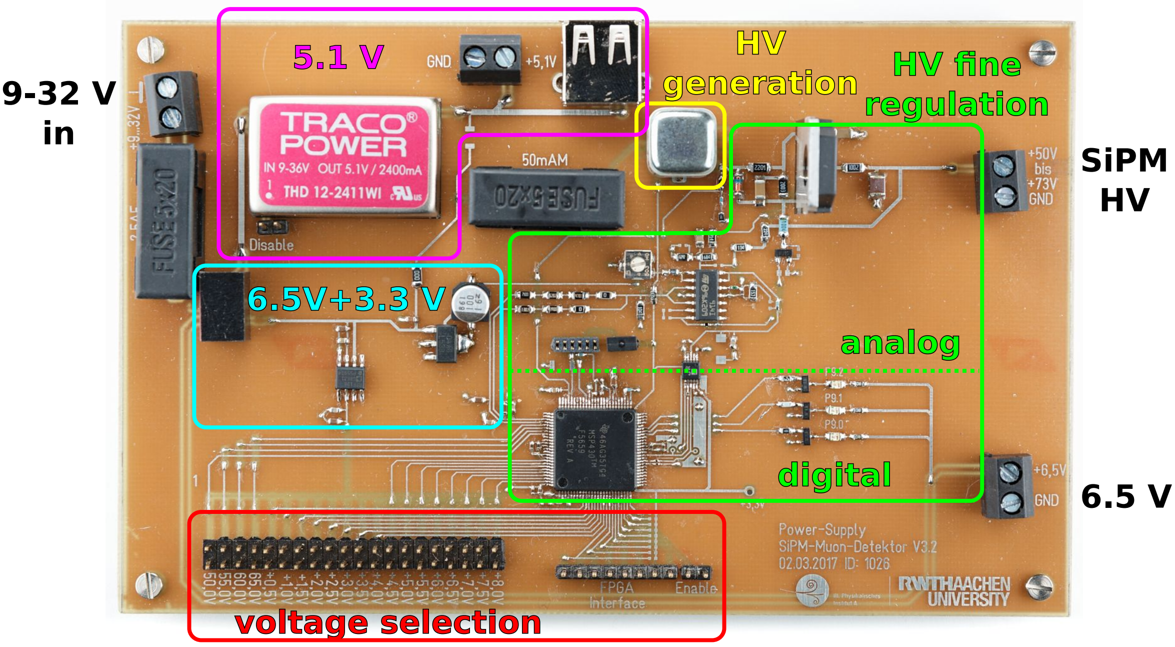


Figure 9.5.: PSU board version 3. Voltage input for +24 V is on the left, outputs for +6.5 V (bottom) and the SiPM supply voltage (top) are on the right. The USB connector and screw terminal on the top provide 5.1 V (2 A) for a single board computer. The pin headers at the bottom allows to select the SiPM supply voltage.

though changes of some resistor values and recalibration are needed for output voltages above 73 V (up to a maximum of about 95 V). The communication interface to the FPGA on the EASIROC board is extended to eight lines to accommodate settings in the increased voltage range and allow for an indication when a stable output voltage has been reached. As for the version 2 board, the FPGA interface lines are active low with weak pull-up resistors. Lines 0-6 are inputs which are interpreted as a 7 bit integer n with 0 representing a disabled output voltage and all other values representing an output voltage $U = (n \cdot 0.5 + 9.5)$ V. Line 7 is an output that is active (low) when the output voltage is stable.

Voltage regulation is achieved in two stages. The target voltage for the improved analog voltage regulation circuit is provided by the DACs in the *MSP430 F5659* micro controller. The output voltage after filtering is digitized by a *AD7685 16-bit ADC* [12] to tune the target voltage set by the DACs. To reduce losses caused by the voltage drop in the fine regulation circuit, the steerable output voltage of the 0.1XS5-P0.1 is dynamically set to be only a few volts above the board output voltage.

The new board also features an increased input voltage range of 9 V to 32 V, which allows for a direct supply with 12 V and 24 V lead-acid batteries at various states of charge. A hard limitation of the SiPM supply current to about 500 µA has been implemented and provides short circuit tolerance on the output.

As a preparation to power the single board computer that is to be included in the

MiniAMD (section 8.6.4), a screw terminal and a USB-A socket provide up to 2.4 A at 5.1 V. This voltage is generated by a Traco Power THD 12-2411WI [80] step down converter, which can be shut down using a jumper if the output is not needed. The battery charge controller for the data logger that was present but unused in the earlier versions of the PSU has been removed completely.

Performance Characterization measurements show the output voltage to be at its nominal value better than about 10 mV at 25°C after calibration at this temperature. Calibration constants in the PSU firmware have been set in an iterative process of adjustments and measurements. The mean deviation over all voltages has been set to zero ($< 1\,\mathrm{mV}$) and the mean slope of the output voltage over the nominal voltage has been set to one. The slope of the output voltage over temperature is less than 1 mV/K for temperatures between −7°C and 60°C and all possible nominal voltages. At voltages above 30 V (lower voltages are not used with any current SiPM type) the ratio of the output voltage V_{out} over its nominal voltage V_{nom} only depends on temperature and not on the nominal voltage:

$$\frac{V_{\mathrm{out}}}{V_{\mathrm{nom}}} = 1 - \frac{1.113 \cdot 10^{-5}}{^{\circ}\mathrm{C}} \cdot \left(T - 27.6^{\circ}\mathrm{C}\right) \tag{9.1}$$

This weak temperature dependence can easily be included in the temperature compensation of the complete setup with only moderate temperature measurement accuracy requirements (SiPM temperature dependence is 60 mV/K). Even the worst case deviation of a single voltage measurement from its nominal value (linearly temperature corrected) of about 15 mV is comparable to a SiPM temperature measurement inaccuracy of 0.2 K and comparable to the accuracy of the EASIROC DACs for bias voltage fine adjustment.

More details on the measurements are shown is section 10.1.

9.1.2. SiPM carrier board

One half AMD uses four SiPM carrier boards. Each board carries eight SiPMs of type Hamamatsu S12571-050P [42] as surface mounted devices, each with $1 \times 1\,\mathrm{mm}^2$ active area. All SiPMs are provided with the same externally generated high voltage (produced by the PSU board) at their cathode, with the anode connected to the EASIROC via ≈1 m RG-174/U 50 Ω coaxial cable. The cable is soldered directly to the board with the shield connected to ground. While the first boards were equipped with cables of 1 m length[4], cables of about 1.5 m length will be needed for the AMD, to reach from the outer tray to the electronics compartment on the other side of the housing. No significant effect is expected to arise from this moderate cable length increase (see also section 10.2.5).

[4] All characterization measurements were performed with this cable length.

Each SiPM is connected to the supply voltage via two consecutive RC low pass filters (100 nF+100 Ω, 470 nF+1 kΩ) and all SiPMs have an additional RC low pass filter (4.7 µF+10 kΩ) in common[5]. With about 0.3 µA current needed for all eight SiPMs together at room temperature, this leads to about 3 mV voltage drop over the filters which can be neglected (with a temperature coefficient of 60 mV/K of the SiPMs this corresponds to a temperature offset of 0.05 K). If a better accuracy is desired in the future, this offset can of course be corrected by using one-time measurements of the temperature-dependent dark current together with the SiPM temperature to calculate the voltage drop. As a positive side effect, the resistors also protect the EASIROC against high currents through the SiPMs in case of high light fluxes or excessive SiPM supply voltages.

Each carrier board also carries one DS18B20 [57] 1-Wire temperature sensor. This sensor measures the temperature of the board which should not differ much from the temperature of the SiPMs, as the SiPMs do not receive great light levels and therefore are not expected to produce much heat (~ µW). The ground for the temperature sensor is kept separate from the ground of the SiPM part to reduce interference and ground loops.

Each carrier board has a size of $190 \times 45\,\mathrm{mm}^2$ with four mounting holes at the edges and two additional mounting holes near each SiPM to install fiber couplings. Most of the design and layout work for the board was done by Rebecca Meißner during her Master thesis [58]. The PCB[6] itself was produced at the electronics workshop of the Physics Institute III A at RWTH Aachen University. The board is shown in figure 9.6.

Due to tolerances—mainly in the mounting holes for the fiber couplings—the fiber couplings on the SiPM carrier board have to be adjusted to perfectly align with the SiPMs. This is a challenging and time consuming task, that will hopefully be unnecessary for further boards due to reduced tolerances. During this procedure, a wavelength shifting fiber is inserted into the coupling for better visibility. Due to the limited space between the SiPM and the bottom end of the fiber coupling, it is hardly possible to see the SiPM surface as well as the fiber. When viewed from the side[7], the light cone emitted from the fiber can be seen relative to the SiPM active area. The ideal alignment is reached, when the amount of light falling onto the side of the SiPM is equal on all sides. An image taken during the process can be seen in figure 9.7.

9.1.2.1. SiPM carrier board for the MiniAMD

For the changed MiniAMD design, where one Hamamatsu S13360-1350PE [41] SiPM is placed at each tile, also a changed SiPM carrier board with a size of 12 mm×32 mm

[5] The resulting ≈50 ms time constant is significantly larger than the ≈1 ms time constant of the measured ripple of the PSU board version 2.

[6] Printed Circuit Board

[7] Preferably with a digital camera and a macro lens in live view mode.

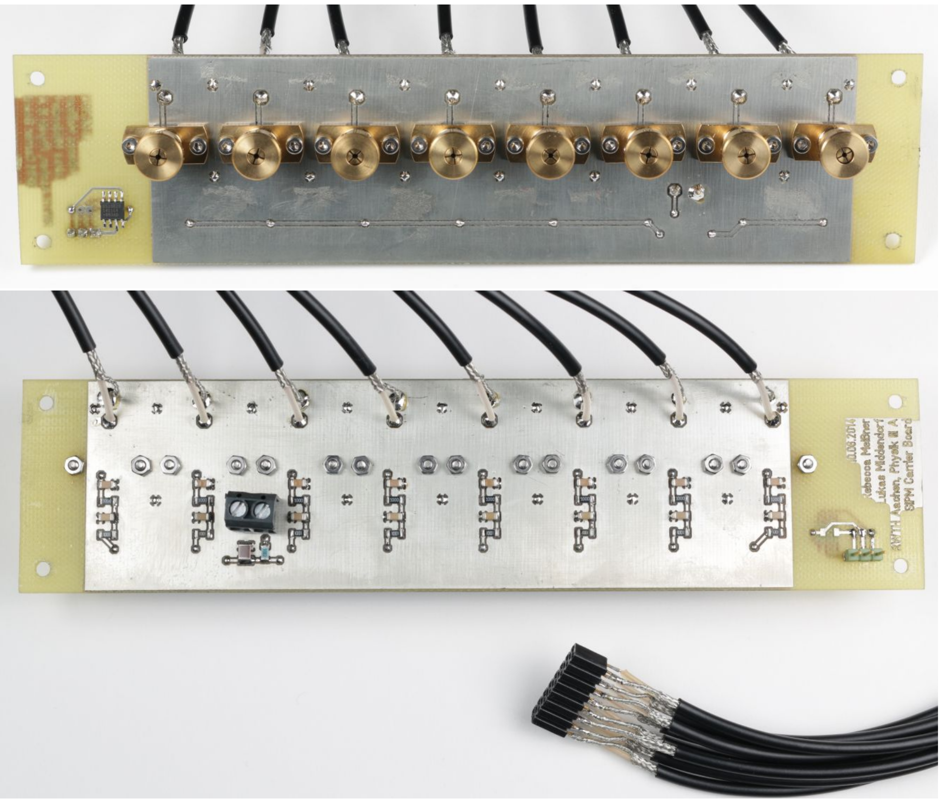


Figure 9.6.: The SiPM carrier board (front+back) for the AMD with soldered signal cables. The SiPMs (not visible in the photo) are located under the installed fiber couplings. Screw terminals are installed to provide the SiPM bias voltage. The pin header can be used to connect to the DS18B20 temperature sensor.

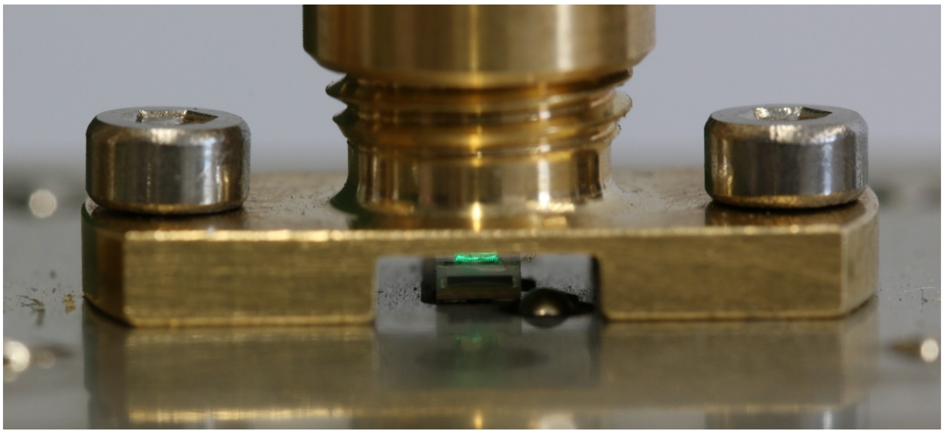


Figure 9.7.: Close up of the brass fiber coupling on the SiPM carrier board during adjustment. A short wavelength shifting fiber is inserted from the top and emits green light onto the SiPM. The coupling can be shifted slightly to align the fiber with the SiPM active area.

is needed. A different SiPM footprint is foreseen on the PCB due to the changed SiPM type with a slightly bigger active area. Two mounting holes near the SiPM are no longer used to attach a fiber coupling but to attach the board directly to the tile. The mounting holes are shifted with regard to the SiPM center to match the mounting holes in the tile center and fiber position at the edge of the tile. In addition to the SiPM each board also contains an RC low pass filter for the bias voltage (10 nF+1 kΩ) and one DS18B20 temperature sensor on the back side. Connection to the temperature sensor is achieved using a pin header, the coaxial cable for the SiPM signal and the single wire for the bias voltage are soldered to the board directly.

Due to the removal of the fiber coupling which allows position adjustment of the fiber relative to the SiPM, position tolerances of the board and the tile have to be small enough to achieve a good position match with a simple mounting procedure. The edge length of 1.3 mm of the SiPM active area compared to the 1.0 mm diameter of the fiber allows for a deviation of 0.15 mm with the fiber still contained in the active area. But even slightly larger deviations only have a minor effect, with only a small part of the fiber area extending beyond the SiPM active area. Tests have shown that tolerances of between 0.05 mm and 0.1 mm of the SiPM position relative to the mounting holes can be achieved.

9.1.3. EASIROC board

The EASIROC board is the heart of the AMD DAQ system. It contains the EASIROC ASIC (see chapter 7), an FPGA and some periphery needed for proper functioning, like DC-DC converters, USB communication, oscillators and ADCs.

For the AMD prototype, EASIROC evaluation boards (see figure 9.8) from Ωmega are used, while for later stages of the project a self-designed PCB with optimized form factor and power consumption is foreseen.

The EASIROC evaluation board (in the following just called "EASIROC board") features a single EASIROC ASIC, which is connected to a 2 × 32 pin header for connection to the SiPMs. One row is connected to ground (but can in principle also carry supply voltage to the SiPMs), while the other row is connected to the signal inputs of the EASIROC. The signal lines are also connected to ground via a 50 Ω resistor and a capacitor, each being charged to the individual offset value for the SiPM supply voltage provided by the EASIROC (see also figure 7.1).

The digital lines of the EASIROC are connected to an Altera Cyclone EP1C6 [9] FPGA. The function of its logic is further described in section 9.2. The FPGA features 5980 logic elements and 92160 bits of memory in 20 memory blocks. To store the firmware for the FPGA, an Altera EPC4 flash configuration device is installed. For clock generation a 40 MHz quartz crystal oscillator is connected to the FPGA.

To digitize the analog charge integral from the EASIROC slow shapers, two AD9220 [13] 12-bit 10 MSPS A/D converters (for low- and high-gain) are installed, with their digital data and control lines connected to the FPGA.

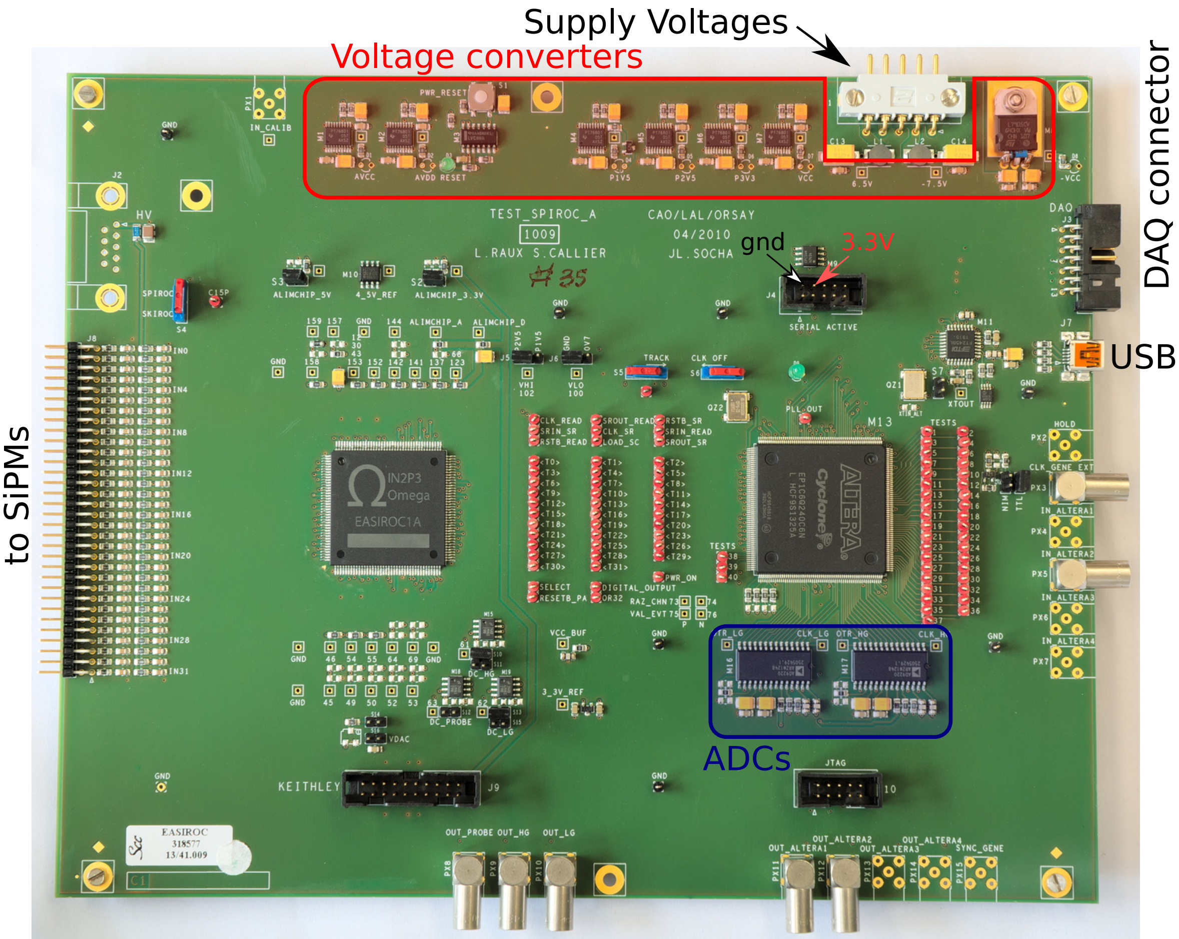


Figure 9.8.: Photo of the EASIROC evaluation board used for the AMD prototype. EASIROC ASIC on the center left, Altera Cyclone FPGA on the center right. Voltage converters are located at the top, next to the connector for the power supply. ADCs for digitization of the analog shaper peak values are located below the FPGA. SiPMs can be connected on the left. Multiple test points are located between EASIROC and FPGA (connecting to signals between them) and on the right of the FPGA (connecting to unused ports of the FPGA).

For connection with the outside world via USB, a FT245B [37] parallel USB FIFO from FTDI is connected to the FPGA.

The EASIROC board uses 6.5 V as supply voltage and generates all other voltages (1.5 V, 2.5 V, 3.3 V, 5 V for the digital part; 5 V and 3.3 V for the EASIROC) using linear voltage regulators. To provide a highly accurate 4.5 V reference for the DACs in the EASIROC, a REF194 [14] chip is used. To power three analog output buffers that can output debug signals from the EASIROC, an additional negative supply voltage of -7.5 V is needed. Since these buffers are not used during normal detector operation, this negative supply voltage is not provided in that case. Nevertheless the buffers are still connected to their positive supply voltage which causes their (normally) negative supply -VCC (−5 V) to drift to positive voltages. To prevent damage to the installed polarized tantalum capacitors, the board is modified from its original state and an additional diode is installed on the board between the -VCC line and ground to limit the maximum positive voltage to a safe level (< 0.5 V). The supply voltages are supplied via the connector with five contacts at the top edge of the board (figure 9.8). Ground is connected to the center pin 3, 6.5 V is connected to either (or both) outer pins (1 and 5) and −7.5 V is provided via either of the other pins (2 and 4).

The power consumption of the EASIROC board[8] from its 6.5 V input can be measured to be 2.4 W, with the majority being lost in the linear voltage regulators. With step-down converters and more modern and power efficient FPGAs and ADCs on a future custom designed board, a power consumption of below 1 W is achievable.

One DS18B20 1-Wire temperature sensor, like the ones used on the SiPM carrier boards, is glued to the EASIROC to allow temperature monitoring and compensation for temperature effects in the EASIROC. It is connected to a free FPGA pin on the DAQ connector of the EASIROC board in parallel with the other temperature sensors. The 3.3 V produced on the EASIROC board is used to power the sensors.

Besides the protection diode for the -VCC voltage, two additional modifications (not counting the soldering of missing LEMO 00 sockets and the removal of some LEDs) of the EASIROC board compared to its default state have been performed by the author of this thesis. All capacitors for high voltage filters on the EASIROC board have been short-circuited to have ground on the lower row of the SiPM connection pin header and subsequently on the shield of the coaxial cables. A 330 kΩ resistor has been installed as R30 between the *iref_dac_5V* pin and ground to reduce the voltage range of the EASIROC input DACs to about 2.5 V (from 4.5 V) to reach a higher possible precision of voltage selection. With the further decreasing breakdown voltage differences between specimens of current SiPMs and the coarse voltage selection for temperature compensation possible with the PSU board versions 2 and 3, an even further reduction of the range could be possible. This further modification has not been performed, as it would require a new calibration of the DACs. If this modification is performed in the future, it might be advisable to also replace the installed 4.5 V voltage reference with a lower voltage model.

[8] After removal of the by-default installed LEDs indicating the availability of the supply voltages, each contributing about 10 mA to the needed supply current.

9.2. Firmware

The firmware, which is described in this section, is running on the FPGA on the EASIROC board. It was completely written from scratch by the author of this thesis. The firmware was written in the Verilog hardware description language [74] using the Altera Quartus II version 11.0 sp1 [10] software[9].

For the EASIROC evaluation board a default firmware is provided by Ωmega. It is specifically tailored for characterization measurements and uses a Windows communication driver library and a low-performance and low-flexibility LabVIEW [60] vi as a counter part on the PC. Instead of understanding the provided VHDL based stock firmware and adjusting it for measurements with the AMD, which would have been a major task (and also might have led to a copyright problem), it was not used at all. Instead the firmware was based on an early firmware for a custom board based on the similar MAROC3 [23] ASIC, also written by the author of this thesis.

All parts of the firmware have been written by the author of this thesis, unless generated by the Quartus-II software.

A much simplified schematic view of the firmware is shown in figure 9.9, with a more detailed view shown in figure 9.10. It is described in detail in this section. Some details are not relevant for normal users of the system but only if modifications of the firmware are to be performed and therefore are provided in appendix C.

9.2.1. General structure

The firmware is divided into multiple modules with some tasks implemented directly in the main module. The tasks present in the main module are trigger generation, steering of hold and readout, and some signal multiplexers.

Parts that require lots of logic (source code), which can be easily reused in multiple parts or which form more or less self-contained units are done in separate modules. One example is the `timeor` module, which at the output provides the `OR` connection of one input over multiple clock cycles and therefore lengthens the input to a selectable minimum time duration.

Many things like widths of counters, widths of some signals, sizes of buffers, etc. are parameterized and can be easily adjusted with a simple change at the top of the main module. This can be utilized to accommodate changing requirements and make different compromises in the allocation of the limited device resources.

Wire and register declarations in the Verilog code are usually not done in one single place at the beginning of modules but are mostly declared right before the start of the `always`-block where they are generated or near the module where they serve as output or control signal.

[9] Even though the 2001 revision for Verilog is selected in Quartus, some features from the 2005 revision (especially `$clog2()`) are supported and used.

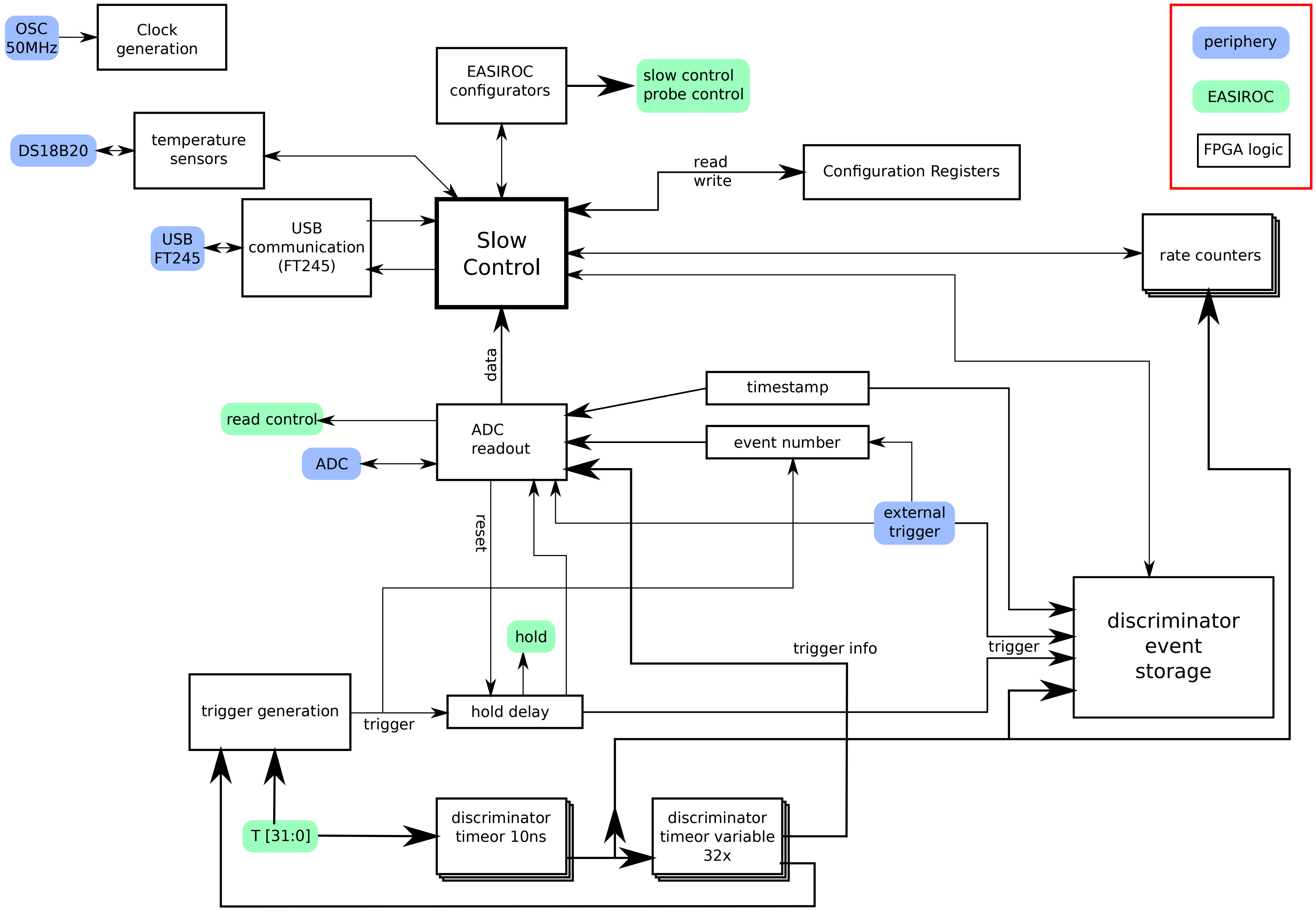


Figure 9.9.: Simplified schematic view of the functionality of the AMD firmware.

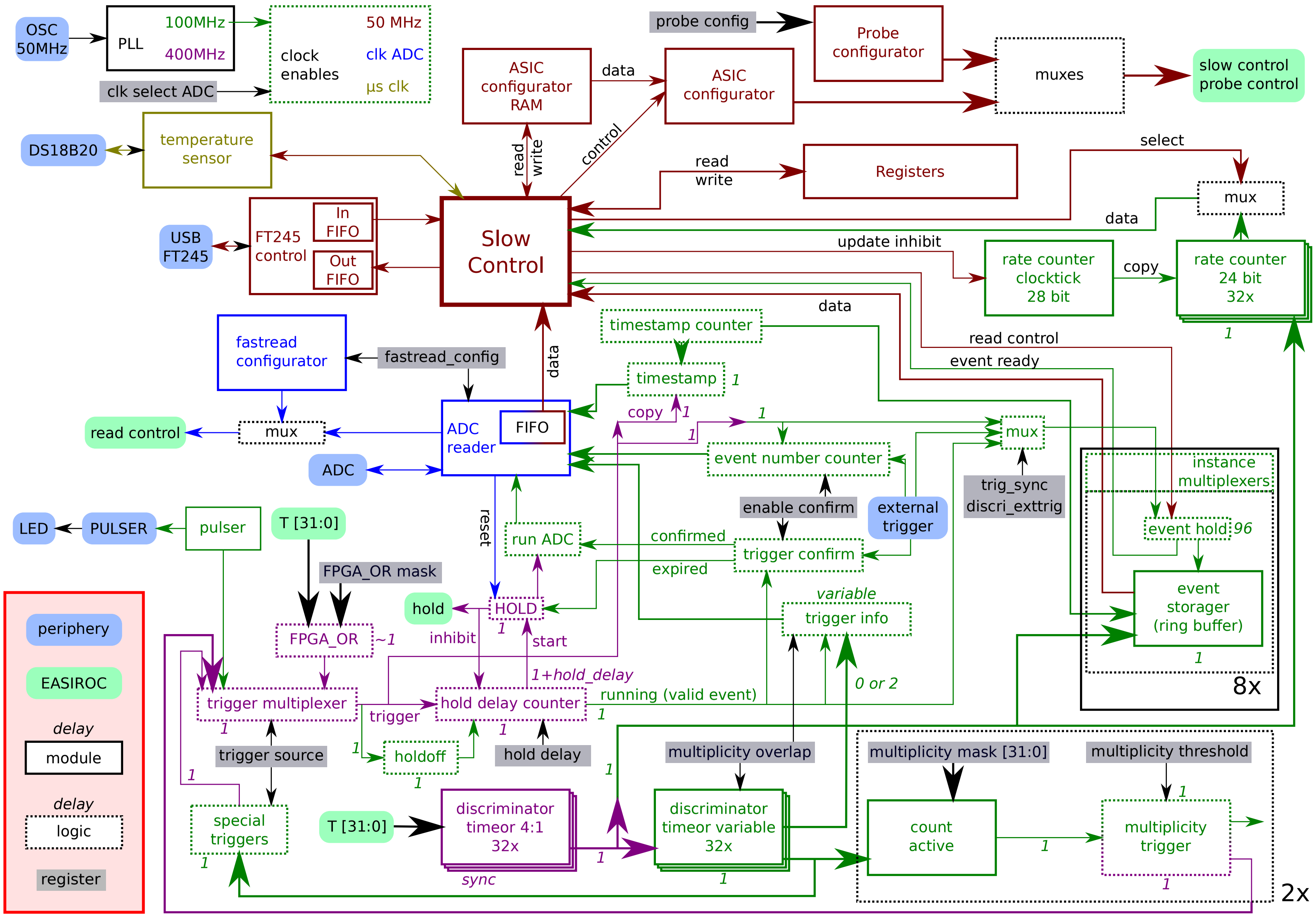


Figure 9.10.: Schematic view of the main layer of the AMD firmware. Italic numbers denote the number of registers in the signal path, i.e. the signal delay in clock cycles. The color of the blocks and numbers represents the used clock.

In various modules formal state machines are used. During synthesis these state machines are handled specially by the Quartus software and are optimized differently than normal logic. Especially the encoding of the state is often altered from the user-provided encoding of the definition. Usually a one-hot encoding[10] is selected, which, while using more registers to encode the current state (compared to a simple state numeration), needs much less combinational logic for state transitions and the recognition of the current state. Every work done in the state machine is done synchronously (at state exit or repeated every clock cycle while in a state) in the same `always`-block and `case` construct in which the state transitions are described.

A simplified example illustrating the coding style is provided below:

```
reg control_line1, control_line2, busy;
parameter
    Sidle=2'd0,
    Sdo_something=2'd1,
    Sdo_more=2'd2,
    Sfinish=2'd3;
reg [1:0] state=Sidle;
always @(posedge CLK) begin
    if(reset) begin
        state<=Sidle;
    end else begin
        case(state)
            Sidle: begin //wait for start to be asserted
                busy<=1'b0;
                control_line1<=1'b0;
                control_line2<=1'b0;
                if(start) begin
                    state<=Sdo_something;
                end
            end
            Sdo_something: begin
                busy<=1'b1;
                control_line1<=1'b1;
                if(some_external_signal)
                    state<=Sdo_more;
            end
            Sdo_more: begin
                control_line1<=1'b0;
                control_line2<=some_other_external_signal;
                state<=Sfinish;
            end
            Sfinish: begin //wait for start to be deasserted
                control_line2<=1'b0;
                if(!start) begin
                    state<=Sidle;
                end
            end
        endcase
    end
```

[10] One register per state is used, with only one register active at a time to indicate the current state.

```
end
```

The register `state` stores the actual state, with the correlation of names to numbers done by Verilog parameters[11] (all prefixed by `S`). The registers `control_line1` and `control_line2` are driven by the example state machine to achieve something. The wires `some_external_signal` and `some_other_external_signal` control what the state machine is doing.

In the example state machine like in most state machines in the firmware, the externally provided wire `start`[12] starts the procedure. The start signal is supposed to be disabled after `busy` is enabled by the state machine and must not be newly enabled when `busy` is already enabled. In the `Sfinish` state, the state machine waits for `start` to be disabled again[13] and `busy` is disabled in the idle state. Through this handshaking system, the external logic invoking the example state machine can operate in a different clock domain than the state machine itself[14]. There are no timing requirements for `busy` active to `start` inactive and there shall be no external timing requirements for `start` active to `busy` active and for `start` inactive to `busy` inactive.

In some state machines in the firmware, a different handshaking logic is used. Here a `run` or `do` signal starts the state machine, but it has to stay enabled until the state machine enables a `done` signal in the final state, where it stays until `run` is disabled. When `run` is disabled, the state machine goes to the idle state, regardless of the current state. This control scheme is especially useful for cases where some external multiplexers are used to route the signals produced by the state machine, with the `run` signal acting as a select for the multiplexers and where the state machine is supposed to abort its current process when `run` is deasserted[15].

9.2.2. Clock system

A single external 40 MHz crystal oscillator provides a clock signal for the FPGA. This clock is used to drive one of the two phase lock loops (PLLs) in the Cyclone FPGA (schematic view in figure 6.4). The PLL allows for two different synchronized clocks with different frequencies to be generated. The fastest internal clock speed usable with the Cyclone FPGA is 400 MHz, which is 2.5 ns per clock cycle. To allow a fine granularity of the hold delay for the EASIROC after a trigger and fast trigger processing, this maximum of 400 MHz is used for these tasks (`CLK_400`). A second clock of 100 MHz (10 ns clock cycle) is used for all other tasks in the firmware (`CLK_100`). Unfortunately no further (slower) clocks can be generated, to be used for tasks that need a slower clock for external communication, better timing margin or to con-

[11] The actual numbers are irrelevant but have to be unique.

[12] usually driven by a different state machine

[13] or just goes to `Sidle` again if it already is deasserted

[14] If clocks are unrelated, the `start` signal has to be synchronized before usage in the state machine.

[15] e.g. external timeout

serve power. For these tasks clock enable signals are used instead to create virtual slower clocks. Registers with a clock enable signal only act on clock edges if the clock enable signal is active.

A clock enable signal with 50 MHz (called `clk_enable_50`) is generated for general use for the USB communication and everything not timing critical like the configuration registers. A clock enable signal with a variable frequency—either 10 MHz or 2.5 MHz—is generated for the ADC readout. The frequency can be selected at runtime to match the drive strength of the EASIROC output buffer, which can be reduced for lower power consumption while simultaneously reducing the maximum possible readout frequency. This clock enable is simultaneous to the `clk_enable_50`. A clock enable signal with 1 MHz (called `us_clock_enable`) for the readout of the 1-Wire temperature sensors is also generated. This clock enable is not (necessarily) simultaneous to any of the other clock enable signals, as a different counter is used for the generation. It is not possible to use clock enable signals with the 400 MHz clock (e.g. to generate a 200 MHz clock) without violating removal and recovery timing requirements.

9.2.3. Communication via USB

The core of the firmware, which takes a sizable amount of about 1400 LEs, is the communication via USB. It is divided into two parts, the module to transfer data via the FT245 parallel USB FIFO (about 200 LEs), and a module which handles the high level communication called `slow_control_command_interpreter`[16].

The module `FT245_async_rxtx` contains two dual clock FIFOs to store the received and to be transmitted data, which are filled and drained by the slow control. The module runs with a 50 MHz clock (using a clock enable), which is critical for correct timing of the communication with the FT245. It is connected to the inputs and outputs (with the data lines being tri-stated bidirectional pins) of the FT245. The completely asynchronous signals *TXE* and *RXF*, which indicate the state of the internal FIFOs in the FT245 are synchronized to the clock before usage. The control signals *RDn* and *WRn* to the FT245 are initialized into the high state and are kept there during idle state, even though according to the data sheet they are supposed to be driven high only shortly before a read or write cycle. This is to prevent a read and write cycle to be initialized after every FPGA reconfiguration, as for an unconfigured FPGA the lines are weakly pulled up to the active state. As the FT245 is only half-duplex (it can either receive or transmit at a given time), a prioritization has to be made. If the transmit FIFO contains data, it is always sent and otherwise any available data is read.

[16] Even though it is named *slow control*, it also performs the rather fast event data transfer.

9.2.3.1. The high level communication protocol

Data can be transferred through the FT245 byte-wise. Each byte can either be used directly as one byte of data or as one ASCII character, consuming two characters per data byte, but allowing for more advanced and human-readable communication protocols which e.g. allow for dedicated and non-ambiguous start and stop characters for messages.

As raw binary data transfer allows for a bigger throughput and is easier to handle in the FPGA, such a protocol is implemented for the AMD DAQ.

The communication follows a strict master slave protocol, in which the FPGA is only allowed to send data when requested by the PC. This way each reply can be non-ambiguously matched with the corresponding request as the order of the data in both[17] data streams is preserved.

To ensure data integrity a *Cyclic Redundancy Check (CRC)* [68] check sum is used. For this, the CRC-8 variant of Dallas/Maxim that is also used for the 1-Wire bus is employed. This choice was not based on any special characteristics of this particular CRC implementation but just on the fact that it was already implemented in the FPGA and the PC driver to use 1-Wire temperature sensors. As no error is expected to occur during normal operation, its purpose is not to reliably detect any error but to detect implementation problems that produce many errors and where the 1/256 chance of randomly producing a correct check sum is negligible. As the original implementation using 0x00 as initialization value is incapable of detecting an error in the number of zero-bytes at the beginning of a message, an initialization value of 0xFF is used instead, leading to a detectability of a wrong number of leading zeros. The checking and sending of CRC values can be disabled in the firmware using parameters to save logic resources. If no CRC is to be sent by the FPGA, the respective CRC byte is set to zero.

There are multiple commands in the communication protocol, which can be divided into two classes:

- Short commands that just consist of one command byte without the need for an additional data payload. For these the most significant bit of the command byte is *zero*.
- Long commands which transmit a data payload to the FPGA and therefore consist of multiple bytes. Here the most significant bit of the command byte is *one*. The command byte is followed by one byte giving the number of following bytes (including one CRC8 byte), then by the payload bytes and finally by one CRC8 byte generated from the complete message including command byte and count byte. No terminator byte is employed at the end of the message. The content of the payload is defined by the individual commands.

  The payload size is limited to 254 bytes with the long command. If a future ap-

[17] receive and transmit

plication makes even longer payloads for commands necessary, an extended command with two or more bytes to give the payload byte count can be defined.

The command form has to be strictly followed, otherwise the system might enter an undefined state which only an FPGA reset can resolve.

This approach is limited to 256 different commands (128 short and 128 long commands), of which the used ones are outlined here and later described in detail.

name	code	description
`ping`	0x00	return success code
`error`	0x01	return error code (like ping, but always fails)
`write_asicconfig`	0x02	write slow control from FPGA to EASIROC
`write_probeconfig`	0x03	write probe config from FPGA to EASIROC
`transfer_ADC`	0x04	Transfer one ADC event (32 × 32 bits)
`transfer_ADC_single`	0x05	Transfer single line of ADC event (32 bits)
`convert_temp`	0x06	convert 1-Wire sensor temperature
`read_temp`	0x07	read and return temperature of 1-Wire sensor
`transfer_rates`	0x08	return current rate counters
`transfer_event`	0x09	return discriminator event
`read_asicconfig`	0x0a	transfer configuration from FPGA to PC
`long_ping`	0x80	ping command checking input data with CRC
`long_ping2`	0x81	identical to `long_ping`
`change_asicconfig`	0x82	transfer configuration from PC to FPGA
	0x83	reserved (don't reuse)
`write_register`	0x84	write an FPGA configuration register
`read_register`	0x85	read an FPGA configuration register
`convert_temp_match`	0x86	convert temperature of single 1-Wire sensor
`read_temp_match`	0x87	return temperature of one specific 1-Wire sensor

Each command mandates an answer from the FPGA:

name	code	description
`ok`	0x10	success
`ok_long`	0x90	success, data is returned
`ok_extended`	0xd0	success, much data is returned
`error`	0x20	an error happened
`error_timeout`	0x21	timeout
`error_crc`	0x22	CRC check failed
`error_unknown`	0x23	unknown command
`no_data`	0x30	no data available for transfer command

As can be seen, the high 4 bits specify the class of the return (see also table 9.3), with the four low bits being available for further specification (only used for error codes at the moment). The no_data return is a special case of being an error and a success at the same time, indicating some expected failure. It is in principle possible to construct long and extended versions of the error codes, but they are not realized in the current firmware.

bit	description
0x10	success
0x20	error
0x80	long answer
0x40	extended answer

Table 9.3.: Function of the high bits of the return code.

All returns that are not long or extended (i.e. highest bit is zero) consist of exactly three bytes. The second byte is the command that actually caused the answer and the last byte is the CRC of the complete command message (some value for short commands, zero[18] for the long commands). This had its main use during development of the communication logic, but is now maintained for backwards compatibility.

The answer byte of the long returns is followed by one byte giving the number of following payload bytes, for the extended returns two bytes give the number of payload bytes (up to 65535). The payload usually contains a CRC value at the end, but this is command specific.

The extended answer is used for transfer_event, the long answer is used for all other transfer and read commands.

ping, error, long_ping, long_ping2: These commands are mainly to test communication and check that an actual EASIROC board with the custom firmware is installed and can be communicated with. The error command always returns the error code even though no actual error did occur. The long ping versions read and discard the complete command message and return either a success or CRC error.

write_asicconfig, write_probeconfig: These commands trigger the sending of the configuration of the slow control or the configuration of the analog probe output for the EASIROC from the FPGA to the EASIROC. The actual configuration is not transferred to the FPGA in this step but with the change_asicconfig and write_register commands.

convert_temp, convert_temp_match: These commands trigger a temperature conversion of the connected DS18B20 1-Wire temperature sensors. The match version takes the 7 bytes (56 bits) sensor ID as a payload, and triggers only one sensor,

[18] The calculated CRC of the command message cancels out with the transmitted CRC at the end of the command message.

the non-match version takes no payload and triggers a convert for all sensors. The command does not wait for the conversion to finish and does not do error checking.

`read_temp`, `read_temp_match`: These commands trigger a temperature readout of the connected DS18B20 1-Wire temperature sensors. The match version takes the 7 bytes (56 bits) sensor ID as a payload, and triggers only one sensor, the non-match version takes no payload and reads out any connected sensor, which leads to a valid result only if just one sensor is connected. The payload of the transfer is the temperature in two bytes (sign bit properly expanded for two's complement) with the least significant byte (LSB) being sent first, followed by a CRC byte calculated from both temperature bytes. A simple error is returned when the actual read on the 1-Wire bus fails (no answer or CRC error).

`write_register`, `read_register`: The read command takes the one byte configuration register address as a payload, while the write command takes one address byte and one data byte as payload. The read command returns as payload one data byte followed by a CRC byte calculated over just this data byte. A usage of invalid addresses results in undefined behavior but not in an error.

`transfer_ADC`, `transfer_ADC_single`: 32 × 4 bytes or 4 bytes (single) of data from the ADC event FIFO are returned as payload, followed by a CRC calculated from the data. The structure of the data is further described in section 9.2.12. The `no_data` return code is sent when no event is available.

`transfer_rates`: The current single channel trigger rates for the 32 channels are returned with 3 bytes per channel (least significant byte first, channel 0 first) followed by a CRC over all data bytes. This command blocks until a new rate measurement (since the last readout or since a reset) is performed.

`transfer_event`: The data payload for this command contains 6 timestamp bytes (LSB first) followed by multiple discriminator status values of 32 bits each (LSB first) and a CRC over the payload data bytes. The `no_data` return code is sent when no event is available.

`read_asicconfig`, `change_asicconfig`: The `read_asicconfig` command returns as data payload the slow control bits of the EASIROC (LSB first) that are stored in a memory buffer in the FPGA. It is followed by a CRC over the data bits. The `change_asicconfig` command changes the internal FPGA memory of the EASIROC slow control bits. An arbitrary range of bits can be changed. Its command payload is two bytes of bit start address (LSB first) and two bytes of data length (LSB first), followed by the data bits (filled to whole bytes with don't care bits).

9.2.3.2. The slow control module

The module slow_control_command_interpreter handles the high level communication using the protocol described in section 9.2.3.1. It is connected to the internal FIFOs in the FT245_async_rxtx module, which forwards the data from/to the FT245 USB FIFO. To perform all the tasks foreseen in the communication protocol, it is connected to about 40 other signals and buses, like FIFOs for the different events, the memory for the EASIROC configuration, the internal FPGA configuration registers used to store runtime configuration for the FPGA functions (mainly trigger), and the module for the temperature sensor handling.

The core part of the slow_control_command_interpreter is a big state machine which handles the main part of reading the command, sends most status code replies and also handles simple commands. More complex commands which have a long or extended reply or a complex command message[19] are handled in separate modules each with their own state machine.

The CRC of a message is automatically checked in the CRC_autocalculator module while it is being received (through a monitoring of the FIFO read request and of the data signals) to be used in the main state machine and the other modules if these do the reading of the command payload themselves.

The state machines in the submodules are started with do signals, which at the same time are also used as select signals in multiplexers allowing the active submodule to drive the control lines of the USB transmit and receive FIFOs if necessary.

9.2.4. Configuration of the firmware functions

Runtime-configuration of the FPGA functions is done via FPGA configuration registers, which are implemented in the registers module. The module acts as a memory to be interfaced by the slow control module to read and write configuration register content, while simultaneously the configuration register contents are constantly available to other logic through appropriately named signal outputs from the module.

The configuration registers have a width of eight bits each and have an address of up to eight bits, of which currently only the lower five bits are used. Read and write requests to non-existent registers are undefined and might operate on other registers, as only the lowest five bits are checked. If not all bits of a configuration register byte are used, the unused bits are undefined during read[20], and any write has no effect.

[19] change_asicconfig

[20] currently implemented mostly to return zero

address	name	description
0x00	probe config	configuration for probe output
0x01	hold delay	delay between trigger and hold signal
0x02	fastread config	read ADC only for this channel
0x03	configA	various functions on single bits
0x04	OR_maskA	FPGA OR mask bits 0..7
0x05	OR_maskB	FPGA OR mask bits 8..15
0x06	OR_maskC	FPGA OR mask bits 16..23
0x07	OR_maskD	FPGA OR mask bits 24..31
0x08	multiplicity threshold	threshold for multiplicity trigger
0x09	multi_maskA	multiplicity mask bits 0..7
0x0a	multi_maskB	multiplicity mask bits 8..15
0x0b	multi_maskC	multiplicity mask bits 16..23
0x0c	multi_maskD	multiplicity mask bits 24..31
0x0d	DAQ outputs	tri-stated DAQ lines to actively pull to GND
0x0e	DAQ inputs	tri-stated DAQ lines that are low
0x0f	firmware revision	read only firmware revision
0x10	multiplicity overlap	four bits of multiplicity time overlap
0x11	configB	enable bits for special triggers
0x12	configC	various functions on single bits
0x13	trigger holdoff	trigger holdoff time in 10 ns
0x14	stack threshold	threshold for stack multiplicity
0x15	stack mask	mask for stack multiplicity
0x17	reset	write-only register to trigger reset
0x18	multiplicity threshold 2	threshold for multiplicity trigger 2
0x19	multi_mask2A	multiplicity 2 mask bits 0..7
0x1a	multi_mask2B	multiplicity 2 mask bits 8..15
0x1b	multi_mask2C	multiplicity 2 mask bits 16..23
0x1c	multi_mask2D	multiplicity 2 mask bits 24..31
0x1f	scratch	read/write register for test purposes

A detailed configuration register description can be found in appendix A.

9.2.5. Configuration of the EASIROC

The configuration for the EASIROC is stored in a memory in the FPGA to be then transferred to the ASIC. The memory is initialized to some sensible default values close to the default values of the EASIROC itself during each FPGA configuration. The memory is an instance of `altsyncram` instantiated in the main firmware module. The memory can be read in total using the `read_asicconfig` command to get the default or previously written values.

The only way to change the memory content[21] is to execute the change_asicconfig command. The changed configuration bits are stored into a separate FIFO while the complete USB command is received and only copied to the memory after a successful CRC check. In this way the memory content is unchanged in case of a CRC mismatch.

The configuration is transferred from the FPGA memory to the EASIROC by an instance of config_to_asic called asic_configurator in the main module. Its operation is started by the logic that handles the write_asicconfig command in the main slow control module. As no timing requirements are given in the EASIROC data sheet for writing to the slow control, the shift register is driven using a moderate clock frequency of 12.5 MHz.

The second configurable part in the EASIROC is the analog probe output. The configuration is stored in the probe_configuration FPGA configuration register (see section 9.2.4), holding the (single) probe configuration bit (0..159) to be activated. Any number bigger than 159 causes no bit to be activated and therefore a disabled probe output.

The probe configuration is transferred to the EASIROC by an instance of the module probe_to_asic (called probe_configurator) in the main module. Its operation is started by the logic that handles the write_probeconfig command in the main slow control module. The shift register is always shifted 160 times, but only when a probe config number smaller than 160 is selected, the shift register input is enabled once.

The data lines to the EASIROC which are shared by both the asic_configurator and the probe_configurator are multiplexed by logic implemented directly in the main module.

9.2.6. Temperature sensors

All temperature sensors used in the AMD are DS18B20 1-Wire sensors [57]. They use the 1-Wire bus which is a master/slave bus that allows a big number of devices to communicate using only one data line (and ground). The logic in the FPGA acts as a 1-Wire master, while all the sensors are slaves. Most 1-Wire devices support a *parasitic power* mode which also provides the supply voltage via the data line. This feature, which is not used in the AMD, requires special line pull-up handling in the master and is not supported in the current firmware. Each 1-Wire sensor has a unique 48-bit[22] identifier embedded and can through this be addressed to trigger a measurement and perform readout.

The raw temperature values are transferred to the PC using the USB communication. No interpretation of the temperature values or calculations based on them (e.g. to

[21] except for a complete FPGA reconfiguration

[22] +8-bit device type + 8-bit CRC

perform SiPM bias voltage corrections) are done in the firmware but have to be performed externally.

The actual 1-Wire communication (reading and writing of a data byte, doing reset) is implemented in the `onewire_interface` module. It is connected to a tri-stated input-output pin with an external pull-up resistor, which is only doing active pull-downs during communication.

The interface module is used by the `DS18B20` module, which does the high level communication. It can trigger a temperature conversion and read out the scratchpad (which contains the converted temperature) of the attached sensors. The temperature values are extracted from the scratchpad and provided as a module output. Communication with the sensors can either be done using the *Skip Rom* command, which causes any attached sensor to perform the desired action, or the *Match Rom* command, which addresses only one specific device ID. Also implemented in the `DS18B20` module but disabled with a module parameter is the *Read Rom* command, which allows to read out the ID of a single connected sensor. Device enumeration using a *Search Rom* cycle to find out the IDs of connected sensors, which is possible in the 1-Wire protocol, is not implemented.

The `onewire_interface` and `DS18B20` modules are working on a 1 MHz clock[23], which is ideally suited to fulfill the timing requirements of the 1-Wire bus protocol.

The `DS18B20` module is instantiated in the main firmware module and its operation is initiated by the `slow_control_command_interpreter` for temperature conversion and its submodule `transfer_temp` for the temperature readout. No logic to handle the output of a possible *Read Rom* command is implemented. To allow completely asynchronous operation between the slow control module and the actual hardware interface, all data[24] is transferred via wide parallel buses using a `start`/`busy` handshake system (see section 9.2.1).

9.2.7. Discriminator input handling

The asynchronous trigger outputs of the EASIROC have to be synchronized to the FPGA clocks before usage. For the 400 MHz clock a direct synchronization using sampling at the clock edges is performed.

For parts using the 100 MHz clock, a logical OR of the discriminator status over four consecutive cycles of the 400 MHz clock is generated. This prevents discriminator switching from being missed, if the discriminator is active for less than 10 ns between two consecutive clock edges of the 100 MHz clock. This is done using the `timeor` module running with the 400 MHz clock, with the output `ssdiscri_100_sync400` then synchronized to the 100 MHz clock to create the 32 bits wide `ssdiscri_100` signal.

[23] implemented as a clock enable on the 100 MHz clock

[24] temperature and device ID

For some applications the state of the discriminators over time intervals longer than 10 ns are important. The module timeor_counter_dual uses an input signal of parametrized width to produce two OR outputs[25] over two variable (changeable at runtime using select inputs) number of consecutive clock intervals. This is done using counters (instances of lpm_counter) which is reset when the corresponding input bit is active. The respective bits of the output become active when the input arrives and inactive when the counter reaches a selectable number. The two paths of the module using individual select inputs use the same counter and therefore save resources compared to two instances of a similar module with only one path. The resource usage for the module is only linear in the width of the select inputs, which is in contrast to the timeor module which achieves a similar task (for only one output and a fixed overlap) using a wide OR over a shift register instead of a counter and therefore has a resource usage linear to the length of the shift register.

The timeor_counter_dual module is used to generate the discri_timeor and trigger_channel_info signals from the ssdiscri_100_sync400 signal. The first signal uses the multiplicity_overlap configuration register as the select input for later generation of the multiplicity trigger, while the second signal uses the doubled overlap.

It is ensured that all different derivatives of the discriminator input using the 100 MHz clock work on data from the same four consecutive clock cycles of the 400 MHz clock.

9.2.8. Single channel trigger rates measurement

Using the firmware it is possible to do trigger rate measurements for all tiles. For this, 32 counters with 24 bits each are implemented. Each counter increments by one when the discriminator for the corresponding channel becomes active[26]. A clock tick counter (implemented in the main AMD module) with 28 bits counts 10 ns clock cycles and on overflow after 2^{28} cycles (≈2.68 s) causes all rate counters to freeze until they are read out. After read out all counters including the clock tick counter are reset and count triggers again for 2^{28} clock cycles.

A firmware parameter exists to enable buffering of the counters. In this operation mode the rate counters are not frozen after 2^{28} clock cycles but copied to a second set of storage registers and reset to continue with a new rate measurement. This way the rate measurement is never older than 2^{28} clock cycles at the expense of substantially more logic usage (about 900 Logic Elements).

The actual counters and also the eventual storage for the copies are implemented in the rate_counter module, which is instantiated 32 times.

For readout, all counter outputs are multiplexed to a single rate_muxed wire using rate_channel_select as the select signal. The select signal is driven in the

[25] each with same width as input

[26] is active and was not active in previous clock cycle

`transfer_rates` module, handling the `transfer_rates` USB command and transferring the counter value of all channels in a successive order. Transfer only occurs when a new rate counter value is available, so no duplicate or invalid values can be read.

The individual rate counters as well as the clock tick counter are reset (and held in a reset state) when the EASIROC configuration is changed. To allow the complete setup to stabilize and to get a current measurement in the non-buffered mode, the first rate measurement value after changing the EASIROC configuration should be discarded.

9.2.9. Event trigger handling

Event trigger generation is one very important part of the firmware, mostly implemented directly in the main module. A trigger causes a hold signal for the EASIROC, an ADC readout of the stored analog values and the generation of a discriminator status event. Trigger generation can be configured by various FPGA configuration registers.

There are multiple selectable trigger sources which are OR-connected:

- A masked `FPGA_OR` of the EASIROC trigger outputs
- An internal pulser trigger firing in fixed intervals for characterization measurements[27]
- Two multiplicity triggers with individually configurable multiplicity thresholds, individual channel masks and a common overlap interval
- Both multiplicity triggers can be AND-connected
- One stack multiplicity trigger of matching tiles in two MiniAMD modules with configurable multiplicity threshold and channel mask
- Special hard coded triggers
  - *special trigger 0*: Any fixed pair of channels (0+1; 2+3; 4+5; 6+7)
  - *special trigger 1*: One of four channels (0,2,4,6) and one of four channels (1,3,5,7)
  - *special trigger 2*: Any fixed pair of channels (0+8; 1+9; 2+10; 3+11; 4+12; 5+13; 6+14; 7+15) (matching tiles of two different MiniAMD modules)
  - *special trigger 3*: One of eight channels (1-7) and one of eight channels (8-15) (one tile each of two MiniAMD modules)

The trigger multiplexer to select the trigger source[28] and the `FPGA_OR` work with the 400 MHz clock. The output of the trigger multiplexer is called `trigger_raw`, while a version synchronized to the 100 MHz clock (oversampled with an instance of the

[27] The pulser output can be sent to an output pin to trigger for example the pulsing of an LED.

[28] multiple simultaneous trigger sources possible

timeor module) is called trigger_raw_100. This leads to an accuracy of 2.5 ns of the desired trigger reaction relative to the actual trigger condition becoming true.

The multiplicity triggers and the special triggers work with the 100 MHz clock. This is mainly done to save limited logic resources[29], and also to avoid a complex manual signal pipelining in the counting of the active channels. This also greatly reduces the number of signals propagating with the fast 400 MHz clock, which are very timing critical and make timing closure hard (see section 9.2.14), especially for an FPGA with a high fraction of used resources. Usually there also is an intrinsic variation in the trigger times between the channels in the same order of 10 ns, and the shaping time and hold delay is chosen such that only a small uncertainty on the signal amplitude arises from this 10 ns timing jitter.

The multiplicity triggers use the count_active module to count the number of active bits (trigger channels) in the discri_timeor signal (section 9.2.7), using the multiplicity_mask configuration register to select which channels contribute to the count. The count is then compared to the content of the configuration register multiplicity_threshold value to produce a trigger when the number of a active channels is greater than the selected threshold. The multiplicity_threshold configuration register value is offset by 1 compared to the naive multiplicity (e.g. has to be set to “3“ to achieve a 4-fold coincidence). This ensures that a minimum of one channel has to be active (threshold = 0) and a 32-fold coincidence can be selected using a 5 bit value (threshold = 31). The multiplicity trigger is synchronized to the 400 MHz clock before entering the trigger source multiplexer.

The complex trigger generation (as well as the trigger delays associated with the different parts) is visualized in figure 9.10.

A counter trigger_holdoff_counter is implemented, which is reset when a new raw trigger (trigger_raw_100) arrives and increments until the selected hold-off (trigger_holdoff_time configuration register) is reached. This is indicated by the wire trigger_holdoff_expired. The signal trigger_holdoff_expired becomes inactive one clock cycle after a raw trigger becomes active. The trigger holdoff is retriggerable, which means the holdoff time is reset by further incoming triggers. While the trigger_holdoff_expired signal is inactive, new triggers do not lead to new events.

For obvious reasons, no triggers are processed while the event buffers are completely filled or while an event (or its storage) is still ongoing. Using the configuration bit trig_sync_discri_adc, triggers can be synchronized between ADC events (ADC values for each channel) and discriminator events (time trace of the status of the discriminators). When this bit is enabled, triggers are only processed when an event of both event types can be generated, which causes matching events of both types to be produced. Otherwise events of one type are still generated while the buffers for the other type are full.

When the enable_confirm configuration bit is enabled, the *IN_ALTERA1* input of

[29] Less registers needed for signal pipelining and less registers needed to achieve the same delays.

the EASIROC board is used as an external trigger input. The external trigger can happen with a big time jitter and delay relative to the SiPM signal. As the timing between the SiPM signal and assertion of the hold signal for the EASIROC is critical, it is not possible to use the external trigger signal directly to produce ADC events. Instead the external trigger is used as a trigger confirm, meaning an internal trigger is used to cause the hold signal to be asserted, but the ADC readout and event creation only occurs when an external trigger confirms the event. When an external trigger happens, the `ext_trigger_happened` signal is generated and a counter (`ext_trigger_counter_lpm`) with 3 bits[30]—running with the 100 MHz clock—is started. The `ext_trigger_happened` signal is active from the start of the external trigger input signal until the counter is full and the external trigger signal becomes inactive. This ensures a duration of the `ext_trigger_happened` signal of at least 80 ns, which prevents missed events in case the external trigger arrives before the internal trigger. When the hold delay for the ADC event (see section 9.2.12 and appendix C.1) is started and no confirm arrived yet, also a counter `ext_trigger_counter_post_lpm` with 6 bits[31] is started. When no external trigger arrives in time the counter overflows (640 ns after start), the hold signal is released and the event is canceled.

9.2.10. Event number

Event number generation is implemented in the main firmware module using the 100 MHz clock. For inclusion in the ADC events, an event number counter (`eventnumber_counter`) is implemented with 32 bits[32]. Depending on the state of bits in the FPGA configuration registers, different events cause the event number to be increased:

- if `enable_confirm` is active: All (and only) external triggers are counted.
- if `evno_count_all` is active: Any trigger of enabled trigger sources is counted, when the trigger has been inactive for a runtime-configurable trigger holdoff time. These trigger do not necessarily lead to an actual event if the previous readout is still ongoing or the event buffer is full.
- else: Every trigger that leads to the start of a hold delay is counted (and because `enable_confirm` is disabled, this also leads to a new ADC event).

9.2.11. Timestamp

Also implemented in the main module is a counter named `timestamp_counter` with 48 bits[33], which increments every clock cycle of the 100 MHz clock. This leads

[30] parameter `ext_trigger_prewaitbits`

[31] parameter `trigger_postwaitbits`

[32] parameter `event_number_bits`

[33] parameter `timestamp_bits`

to a timing resolution of 10 ns for usage as a timestamp in events and a time until overflow of around 32.6 days. The precision compared to other free running external clocks is defined by the crystal oscillator on the EASIROC board[34]. There currently is no means to synchronize the counter with an external clock providing a signal in fixed intervals. The easiest way to implement a synchronization of the internal clock is the external triggering of events with a precisely known (external) time for a determination and correction of the clock offset during data analysis[35].

9.2.12. ADC readout

ADC readout is an important part of the firmware. Data of the events is stored in a FIFO for subsequent data transfer, which is performed by the `transfer_ADC` submodule of the slow control.

On a trigger, after a selectable hold delay (configured by the `hold_delay` configuration register in 10 ns) the hold signal for the EASIROC is asserted. The timestamp of the trigger, the incremented event number and the mask of channels with active discriminators (`trigger_channel_info`, see section 9.2.7) are stored for inclusion in the event data. Afterwards the ADC readout is started and the ADC values of the 32 channels (high-gain and low-gain) together with the auxiliary data are stored in the event FIFO. If *fastread* has been enabled, only a single channel is read out and the data of only this channel is stored in the event FIFO.

More implementation details are described in appendix C.1.

9.2.12.1. ADC event structure

For each channel, each ADC (high-gain and low-gain) provides 12 bits of data and one over range (OTR) bit. As the output voltage of the EASIROC is in the ADC input voltage range in any case, an active OTR bit indicates some serious (hardware) problem and should not occur during normal operation. As each channel ADC data consists of 26 bits, six additional bits of the 32 bits in the FIFO line remain free. The remaining bits are used to transfer additional data. The triggerinfo (information which discriminators were active at the time of the trigger) and event number, which are provided externally, are added to the data stream one bit per channel. The timestamp which encompasses 48 bits is padded to 64 bits (with zero) and added to the data stream as two (consecutive) bits per channel. Two further bits per channel are unused. The lowest valued bits of the additional data is transferred with the data corresponding to the first channel (channel 0).

The exact bit assignment is shown in table 9.4.

[34] The exact precision specifications are unknown.

[35] This needs the implementation of a separate external trigger and the introduction of an additional information bit in the event structure.

[31:30]	29	28	[27:16]	[15:14]	13	12	[11:0]
xx	info	OTR_LG	adc_LG	time	evno	OTR_HG	adc_HG

Table 9.4.: Bit assignments of one line of ADC event data. The ADC values for the high-gain (HG) and low-gain (LG) channels are stored together with their over range (OTR) bits. *info*, *time* and *evno* (event number) are single bits of values distributed over multiple channels.

For fastread, besides the conserved time to drive the EASIROC read configuration lines, only one ADC value has to be measured and only one FIFO line is written. As the main bottleneck for the achievable event rate is the data transfer via USB, this leads to a 32 time increase in the achievable event rate. A downside is that due to the limited number of available data bits, no additional information like event number, timestamp or trigger information are transferred (all special bits are undefined).

9.2.13. Time trace of discriminator status

To facilitate the *counting method* of the AMD operation, a time trace of the status of the discriminators of the EASIROC is needed. The fastest sampling time achievable with the Cyclone FPGA is 100 MHz (limited by the maximum clock frequency of the embedded M4K memory blocks), which is sufficiently fast considering the 15 ns nominal peaking time of the fast shaper. For a reduced data volume, an even slower sampling might be desirable. The discriminator event is built using the `ssdiscri_100` signal, which is effectively the status of the discriminator during previous 10 ns.

Each event, which can be transferred via USB, consists of a timestamp of 6 bytes (least significant byte first) followed by 128 × 4 bytes of discriminator status.

The event building and trigger handling is encapsulated in the `discrievents` module. It receives different triggers depending on the state of some configuration registers. If the `trig_sync_discri_adc` configuration register bit is set, the signal `hold_delay_running_100`[36] is used. If the `discrievent_exttrig` configuration register bit is set, the external trigger input is used as a trigger signal. If both bits are unset, the same trigger as for the ADC events is received. This results in a slightly different time offset between ADC and discriminator status events depending on the setting. The module indicates whether an event is ready to be transferred via USB and whether it can receive a trigger to facilitate the synchronization with the ADC events. It also receives the timestamp counter for inclusion in events.

The external interface of the `discrievents` module is a FIFO from which the event data can be read byte-wise and which indicates whether a complete event has been read. The FIFO readout is operated with a different clock (50 MHz) than the sampling clock.

[36] This indicates that an ADC event will be generated.

Actual event storage is done using multiple instances of ring buffers, which continuously record the discriminator status. When a trigger is received, a predetermined (fixed) number of post-trigger samples is recorded until a ring buffer is frozen and the corresponding timestamp is saved, while the other ring buffers (if not frozen themselves previously) continue to be updated for the next trigger. On read, the earliest recorded event is transferred and afterwards the ring buffer is unfrozen and ready for the next trigger (after it has updated all required pre-trigger samples).

A detailed description of the implementation, which might be interesting when trying to modify the firmware, is provided in appendix C.2.

9.2.14. Timing constraints

Like for all projects in the Quartus software, timing constraints for the TimeQuest timing analyzer in form of an *sdc* file have to be made. Basic clock definitions including automatic determination of PLL clocks and uncertainties are made.

With false path exceptions a signal can be excluded from being analyzed for timing in cases where the designer knows that it is not timing critical (e.g. changes only when the logic using it is in a reset state), but this fact can not be derived automatically.

For the AMD firmware, false path exceptions are set from the configuration registers to everything running with `CLK_400`, as configuration register content is static during normal operation where `CLK_400` drives the trigger functions. Furthermore a false path exception is set for paths from `ADC_event_timestamp`, which is copied with `CLK_400` during trigger and hold generation and therefore has very little calculated timing margin, but is used only many clock cycles later.

Finally proper multi cycle path exceptions are set for all registers which use a common clock enable. This tells the timing analyzer that a signal is allowed to not only take one but a specific number of clock cycles without a timing violation. For this a new procedure is defined. It takes the register names of the clock enable signal (a list of multiple signals if multiple different but synchronized clock enable signals are present) as the first parameter and the implemented clock divider as the second parameter.

```
#define function to easily set multicycle path exceptions for registers with common clock enable
proc clk_enable_multicycle {regname setupvalue} {set_multicycle_path $setupvalue -to [get_fanouts $regname -through [get_pins -hierarchical *|ena]] -from [get_fanouts $regname -through [get_pins -hierarchical *|ena]] -end -setup; set_multicycle_path [expr $setupvalue-1] -to [get_fanouts $regname -through [get_pins -hierarchical *|ena]] -from [get_fanouts $regname -through [get_pins -hierarchical *|ena]] -end -hold}

#set multicycle path exceptions for registers with clock enable
clk_enable_multicycle {clk_enable_50 ADC_clock_enable} 2
clk_enable_multicycle ADC_clock_enable 10
clk_enable_multicycle us_clock_enable 100
```

With the clock of 400 MHz, the timing closure is very tight, so that usually multiple fitting attempts with different seeds are needed, before in the best cases around 200 ps positive slack can be achieved[37].

9.2.15. Reset logic

A complete reset of the FPGA occurs when it is reconfigured, i.e. the firmware is newly loaded from flash or via JTAG. If a reconfiguration is triggered via JTAG during normal operation, the outcome is indistinguishable from a complete power cycle of the FPGA. Apart from that, there are two reset states in the firmware.

The first one is a global reset, which resets all FPGA configuration registers, clears all FIFOs and buffers, resets event number and timestamp counters, and terminates any ongoing processes in state machines[38] like ADC readout or USB communication. This reset state is entered when an external reset signal to the FPGA is enabled or the PLL for the clock generation looses its lock. If PLL lock is lost, the reset signal stays active for some additional clock cycles after lock is reacquired, to allow a proper reset of the parts of the logic that use a synchronous reset and therefore require a proper clock signal for the reset to be performed. The external reset signal can be generated either by a push button on the PCB or it is automatically generated for a duration of about 50 ms when board power is lost and restored.

The memory for the EASIROC slow control configuration can not be reset to its default state during a simple reset without a complete reload of the FPGA configuration.

The second type of reset is triggered when the EASIROC configuration is written to the EASIROC or an FPGA configuration register is written. Every currently ongoing readout is aborted and all data buffers (for ADC events, discriminator events and trigger rates) are cleared and no new triggers are accepted until the EASIROC is properly configured again. Timestamp counter and event numbers are not reset.

To reset the timestamp counter and event number counter, a write to the `reset` configuration register can be performed. Each counter can be reset individually by setting the corresponding bit at register write. The write-only register is volative and does not permanently store what is written to it.

The probe configuration in the EASIROC is reset during FPGA firmware reset, but the normal *slow control* configuration in the EASIROC is unchanged and therefore does no longer necessarily reflect the state in the FPGA configuration memory.

[37] often only a few 10 ps

[38] go back to idle states

9.2.16. Other functionality

The FPGA-configurable LED on the board is blinking with a frequency which is proportional to the (selectable) ADC readout clock. It can be disabled using a configuration register bit to turn the board dark.

There are four *out_altera* outputs available on the board. These outputs can be enabled through enabling of the `trigger_outputs_enable` configuration register bit. Disabling is advisable while changing the EASIROC slow control, to provide only well defined trigger outputs. By default *out_altera_1* outputs `run_read_ADC` signal, which indicates a new ADC event is digitized. *out_altera_2* and *out_altera_3* output the multiplexed raw trigger. *out_altera_4* outputs the special trigger 2 (any one MiniAMD stack). If the `trigger_outputs_karlsruhe` configuration register bit is enabled, these outputs are replaced by outputs of the (synchronized) discriminator inputs. Each output is fed by four OR-connected inputs, which can individually be masked by the `OR_mask` configuration register: (0+1+8+9)→1, (2+3+10+11)→2, (4+5+12+13)→3, (6+7+14+15)→4. This second set of output assignments has been implemented to perform measurements in Karlsruhe[39], where an external oscilloscope was to digitize the trigger outputs to measure position-dependent trigger efficiencies of the tiles.

9.3. Software

For the AMD DAQ to work, not only the hardware and the firmware in the FPGA is needed, but also an external computer program. For the prototype, communication occurs via the FT245BW chip [37] on the EASIROC board, which is connected to the PC via USB and appears as a serial port. Communication can in principle be adjusted to use any serial byte-wise form of digital communication with the FPGA. The main tasks to be performed by the programs are:

- Run-time configuration of FPGA operation using the FPGA configuration registers.
- Readout of the temperature sensors using the FPGA and any other possible externally connected "sensors"[40].
- The generation of the configuration of the EASIROC to be written via the FPGA, including calculation of the SiPM bias voltage temperature correction.
- Initialization of data transfer (single channel trigger rates, ADC events (amplitude of the shaper output), discriminator events) and subsequent data storage.

After initial configuration, a permanently running program is needed for SiPM bias voltage correction, even when only hardware trigger outputs are used and no digitized data is desired.

[39] The actual measurements were never performed.

[40] e.g. external voltages using multimeters

9.3.1. The LibLAB

The LibLAB is a laboratory software framework designed at the Physics Institute III at the RWTH Aachen University. It is a library of hardware drivers mainly for DAQ hardware but also for power supplies, stepper motors and other auxiliary hardware in the form of C++ classes. The LibLAB is developed for Linux, but most parts can also be compiled for Windows using the MinGW [2] compiler.

The LibLAB classes consist of *interfaces* and *modules*, organized in *buses*. Each bus has an abstract interface class with one or more implementations. Each of the modules, which are mainly the actual devices but might also again be an interface implementation of a different bus, uses just the abstract interface class and can therefore in principle be used with multiple interface implementations without any change to the actual module class. Many classes are designed and tested to work on little endian (x86) and big endian (ppc) systems. For most hardware registers of devices, bitfields are implemented to allow easily readable register manipulation using bit names without the usage of obscure bit shifts and bit masks. These bit fields are usually part of a union which allows easy access to the complete `raw` register value in additional to its components without performing type casts. According to the official C++ standard, reading from the not most-recently written to member of a union is undefined, but using the gcc compiler it gives reliable results, comparable to what a `reinterpret_cast` produces[41]. Even though it is not yet implemented everywhere, LibLAB classes are supposed to throw exceptions derived from `llexception: public std::exception` in case of errors instead of using return values for error checking. This usually does not include cases of *no data available*, which can be a normal occurrence and should be signaled using return values.

The most important buses are `vme` and `rs232`, with the second bus containing not only real rs232 devices but also devices using other implementations of a character based asynchronous communication. The currently implemented VME interfaces are for the Wiener VM-USB [83], the MVME3100 [33] and the logically compatible CAEN CONET2 connection via optical fiber using the CAENComm library [29]. VME module classes exist for various CAEN VME devices, iSeg high voltage modules and some self-built devices. Interface implementations for the rs232 bus are currently available for tty devices under Linux for usage with everything that emulates a standard serial port (`linux_rs232`), for FTDI USB to serial adapters using the libFTDI (`libftdi_rs232`), for the COM port on windows (`windows_rs232`), for a communication using the telnet protocol (`telnet_rs232`) and for using a network socket (`ether_rs232`). Module classes for various devices (e.g. multimeters and oscilloscopes) are implemented.

Implementations of the `rs232_interface` in the LibLAB are used for communication between the AMD firmware and measurement software. Also used for this thesis is the driver for the FLUKE 8845A multimeter, which can be used with the

[41] The exact behavior differs by endianness, therefore two implementations of the unions (and especially the bit fields) are necessary for big and little endian systems.

`linux/windows_rs232` as well as with the `telnet_rs232` interface implementations.

The LibLAB is not yet publicly available, but there are plans to release the LibLAB under the LGPL license.

9.3.2. Driver for communication with the firmware

A C++ driver class for communication with the FPGA uses the `rs232_interface` from the LibLAB. It encapsulates the communication and provides high level functions for configuration and data readout. It is described in detail in appendix D.

9.3.3. Measurement programs

All measurement programs for the AMD prototype use the driver class mentioned in section 9.3.2 and described in appendix D to perform their task. Besides some test programs that have been used during firmware development or the *test1wire* program that performs a readout of the temperature sensors using the FPGA firmware and can also be used for general testing, two main measurement programs have been written. The *trigger rate scan* program varies the trigger threshold of the EASIROC while using the FPGA to count the trigger rates for all channels and then writes the trigger rates in dependence of the trigger threshold to an output file. The *shelf* program was mainly written to operate the AMD shelf (see section 8.6.2), but has since then been extended to facilitate a variety of measurements with the AMD, involving the readout and storage of ADC and discriminator events. Both measurement programs also include the readout of the temperature sensors and setting of the proper DAC values for SiPM bias voltage regulation.

9.3.3.1. The shelf program

The `shelf` program has its primary purpose in reading and storing ADC and discriminator events measured in the AMD shelf or whole detector. A second possible application is to initialize and run the EASIROC to provide trigger signals for other experiments, even when no events are read out. During operation, it measures temperatures (SiPM, EASIROC and PSU) and optionally the external SiPM supply voltage of the PSU using a multimeter, and it does a bias voltage stabilization using the input DACs in the EASIROC. It is highly configurable and most of its operation can be adjusted at run time by using a configuration file or command line arguments. Some default values are set in the source code, but almost anything that changes between different variations of the setup or different modes of operation is configurable without recompilation of the source code. The properties of the connected SiPMs like their operation voltage and corresponding temperature sensor IDs can easily be configured. The program can display histograms of the currently running measurement using *root* [28], but this has to be enabled through a configuration option and can also be completely disabled by removing a preprocessor `#define` to remove any overhead (also the inclusion of the root header files is omitted and subsequently the root libraries don't have to be linked).

To achieve the configurability, the *boost_program_options* package [1] is used. It allows an easy parsing of command line options as well as configuration files using the same keywords. In the command line, one or multiple configuration files can be specified (either with `--options-file`, with `-o` or by just giving the file name). It is also possible to use the `options-file` keyword in a configuration file to include further configuration files. Parameters specified in the command line take precedence over configuration files, with configuration files specified earlier taking precedence over later configuration files (configuration files are parsed completely before any further inclusions are handled). The lines of the configuration files have the form `key=value`, with comment lines starting with `#`. A list of all available configuration options can be printed using the `--help` switch.

Signal handlers for SIGINT (ctrl+c) and SIGTERM are installed to cause a proper termination of eventual telnet processes used for multimeter readout. It also causes the measurements to stop after the current measurement interval on a single press of ctrl+c, while keeping the histogram windows open, if these have been activated. A second press to ctrl+c causes an immediate termination of the program.

The PSU versions 2 and 3 can be controlled to output a selectable SiPM supply voltage and optionally also ramp down the voltage again after the end of the measurement.

Any trigger source possible in the FPGA firmware (see section 9.2.9) can be enabled. The trigger threshold for the discriminators in the EASIROC can be chosen. The masks and multiplicity thresholds of the multiplicity triggers, as well as for the stack multiplicity trigger can be configured. Special trigger options like the synchronization of discriminator and ADC events and the external trigger confirm input can be selected. The hold delay in the FPGA is configurable as well as the shaping times and feedback (gain) for the EASIROC.

For a nomal event readout of all 32 channels a maximum readout rate of about 5 kHz can be achieved. To perform an almost complete readout of SiPM dark noise for 1 mm^2 SiPMs with more than 100 kHz, the fastread feature of the FPGA firmware can be enabled, which allows readout and data storage of just one channel, increasing the achievable event rate by a factor of 32 at the expense of not having event time stamps, event IDs and trigger discriminator information (which of the 32 discriminator channels have been active at the time of the trigger).

The storage of event data can be disabled, for example when only the plotted histograms are used as a quick functionality check, or if only the trigger outputs of the EASIROC board are to be used. Storage of data happens into a new sub-directory named `run_n`, with `n` being the lowest unused number in a directory that can be specified in the configuration or command line (default "shelfout", relative to the current working directory). File formats and its names have mainly been chosen at some point in the past for earlier program versions and are since then unchanged or only slightly modified to preserve compatibility with analysis programs. For this reason, some choices of the output formats appear non-optimal in retrospect. Most files are recreated every (by default, but configurable) 60 minutes with an incrementing number postfix, to limit file size and to be able to limit the measurement time to be used for analysis in post, for example when at some point during the measurement external light starts to disturb the measurement. Four types of files can be created:

- Log data including the measured temperatures, voltages and selected DAC values for bias voltage regulation. New data is added at the back of a single file called `mip_peak.log`.
- Histograms of the ADC events. Files called `histos_n.out` are overwritten every time the histogram is stored and preserve the last content when `n` changes.
- Full data for each ADC event including all ADC values, event number, time stamp and trigger information. Files are called `singles_n.out`, with new data (about 600 bytes for one event with all 32 channels) added at the back of the current file.
- Discriminator events. Files called `discri_n.out` have new data added at the back of the current file.

The log data and histograms are always stored when storage is not completely disabled, the files containing the single events have to be enabled individually. All files are tab-separated ASCII files, with the logfile and histogram files containing data in decimal notation unless specified otherwise, while the data files with the single events contain data in hexadecimal notation.

The log file contains in its first line the selected trigger mask (hexadecimal), the trigger multiplicity, the selected value of the threshold DAC (DAC10), the time difference between new files in seconds, the feedback for the high- and low-gain channels, the shaping parameter for the high- and low-gain channels and the selected hold delay. Some older output files might contain less information in the first line, if reading a file, additional and missing information should be expected. Every subsequent line contains the time in seconds since the start of the run, the total number of recorded events, the temperatures of the PSU, and the EASIROC, the measured or assumed external voltage and the temperature of the SiPM. Following on the same line is for each channel the (calculated) set voltage offset, the selected DAC8[42] value and the resulting SiPM bias voltage.

The histogram files contain the runtime in seconds and the total event count in their first line. It is followed by 4096 lines, each containing the bin number, followed by the entries of the high-gain and low-gain histograms for each channel.

The “singlesfile” contains all numbers in hexadecimal notation. One line per event contains the timestamp, the event number, the triggerinfo, the number of following

[42] input DACs to shift the effective SiPM ground level

read channels, and for each channel the high-gain and the low-gain ADC data. When fastread is enabled, the timestamp and event number are zero and the triggerinfo always has only the bit corresponding to the selected channel enabled.

The "discrifile" also contains all numbers in hexadecimal notation. Each line contains a single event, which consists of the time stamp, the number of data points and finally all the data points (a 32 bit number each).

9.3.3.2. Trigger rate scan

The program to measure the trigger rates of the individual EASIROC channels in dependence of the discriminator threshold is called `threshold_rates`. Its operation and the EASIROC settings are configurable using configuration files and command line options similar to the `shelf` program. SiPM operation voltages, EASIROC preamplifier gain and the range of discriminator thresholds to measure are among the configurable parameters.

The program starts at a selected start value of the threshold DAC and decreases it by a selected step size down to a selected end value, which leads to a rising threshold voltage during the measurements. At each threshold a selectable number of event count measurements is taken and read out. The first rate measurement always is discarded to make sure that the EASIROC operating conditions (especially the threshold and the SiPM bias voltage) are stable and at their selected values.

Between the different attempts to read the rate data, the file descriptor of the serial port can be closed. This stops the USB communication between the PC and the FT245 chip on the EASIROC board, which has been observed to cause noise triggers, especially at low trigger thresholds. Communication is only resumed for short time periods after waiting for a sufficient time interval to ensure the next rate measurement is available.

Before the first threshold setting and after a selectable time, temperatures of EASIROC and SiPM and the external voltage of the PSU (if a multimeter is used for this task) are measured. Afterwards, the input DACs of the EASIROC are set to values suitable to achieve the correct temperature compensated SiPM bias voltage. The `easiroc_dacs` class is used to calculate the DAC values using previously measured calibration constants.

A simple tab-separated ASCII file with numerical values (decimal notation) is used to store the measured trigger rates. A string called `specifier` can be chosen, and the file is named `data_threshold_specifier_n.out`, where `n` is the first free number (starting at 1) which results in a file name not already used. Each line contains the threshold, the temperature of the EASIROC, the temperature of the SiPMs, the time interval over which the event count was accumulated and the external PSU high voltage. In the same line this is followed by one SiPM bias voltage data set per activated channel (can vary). Each data set contains the channel number, the required voltage difference, the resulting DAC value and the (calculated) resulting bias voltage (taking differences between the desired and achieved DAC voltage into account). Finally the line is completed by the actual event counts for all 32 channels. For each channel an overflow flag (“1” if the rate counter experienced an overflow, “0” otherwise) is followed by the corresponding event count. If no voltage and temperature measurement and DAC adjustment has been performed between two different threshold settings, the corresponding values are unchanged compared to the previous threshold setting.

9.4. Further plans

Communication with external detectors, especially with the Updated Unified Board (UUB, see section 3.4.1) of the Auger SD stations, has not been implemented in the FPGA firmware yet, except for a simple trigger confirm input and some simple trigger outputs. Communication paths to be implemented are (depending on the availability of an USB port on the UUB) on the one hand the data readout and control as well as the receiving of trigger signals using a different hardware interface, and on the other hand a synchronization of clocks and event numbers. This communication has not yet been implemented, because it was not clear that the AMD will ever be operated in Argentina together with an SD station and the design of the protocol and implementation of the counterpart in the SD electronics firmware would have involved SD electronics experts outside of the AMD group, which would very likely not spend an effort without a clear benefit.

In case of a large-scale deployment of the AMD detector concept, a better and custom designed EASIROC board has to be developed. Besides a slightly bigger and newer FPGA with more resources to use and better means of digital communication with the SD tank electronics, a reduction of power consumption is the main goal. This goal can be achieved by more efficient step down voltage converters for power supply[43], a more modern FPGA architecture[44] and the omission of some powered components not useful for AMD operation. Different connectors for the SiPM signals, which are better shielded and impedance-matched than the pin headers currently used, can hopefully improve noise performance.

Also still needed for a full scale AMD detector is the synchronization between the two detector halves in case a combined trigger is necessary. If both detector halves are synchronized to the SD tank electronics, no explicit clock synchronization between both detector halves has to be done. For cost reasons (water tight connectors with many signals) and due to timing issues caused by the relatively long signal delays, it is not realistic to build a combined multiplicity trigger of both detector halves, but both will have their own multiplicity trigger. In cases where only one detector half causes a multiplicity trigger on its own for a shower, the number of

[43] The EASIROC evaluation board uses very inefficient linear voltage converters.

[44] A significant power consumption reduction can be expected due to technological advances in chip production in the last 15 years.

muons (and therefore the chance of double-hits) is low enough that the generation of a discriminator event to use the counting method in the other detector half is likely sufficient. If AMD events are only stored and transferred if also a trigger of the normal SD station is generated, a direct link between both detector halves could be forgone altogether with the SD station sending a trigger to the AMD. If still desired, it would be possible to use the trigger signal from the other detector half to drive the hold signal for a discriminator event, but signal delays have to be properly accounted for in this case.

10. First characterization measurements with the AMD

With the AMD DAQ electronics, firmware and software ready, the system has to be calibrated and characterized. A sufficient performance of the individual detector components has to be confirmed to allow successful measurements with the complete system.

The SiPM gain should be stable better than 2%, which corresponds to a bias voltage accuracy of 50 mV at 2.5 V overvoltage. SiPM gain differences up to 4% (≙100 mV) can be tolerated if the actual voltage difference is known to better than the original 50 mV, to correct the difference during data analysis. To achieve this voltage accuracy, the input DACs in the EASIROC, which define the ground level for the individual SiPMs, and the Power Supply Unit are characterized and calibrated in section 10.1. The used DS18B20 temperature sensors are specified to have an accuracy of 0.5 °C, which corresponds to a voltage accuracy of 30 mV. It is safe to assume that with a temperature resolution of 0.0625 °C for a given sensor the relative precision is significantly better.

To facilitate a proper analysis of AMD data and to enable an electronics simulation to select proper operating parameters, the behavior of the slow shapers in the EASIROC is characterized in section 10.2.4. In particular the cross calibration between the high-gain and low-gain path in the EASIROC is proven using real light produced in the AMD scintillator tiles in section 10.2.4.2.

The most important design goal of the AMD system is a 99 % trigger efficiency for crossing muons in a single tile at a trigger threshold corresponding to 10 Hz dark noise triggers per channel. This dark noise rate is slightly less than the rate of random crossing muons in a tile, which is an irreducible background. This measurements using stacked tiles are presented in section 10.3.2.

EASIROC characterization measurements of the input DACs and trigger threshold scans have been performed together with Rebecca Meißner and are published in her master thesis [58]. Some of the measurements presented here have already been published in a very short form in [66], [67] and [50], but unless stated otherwise the measurements and analyses have been performed by the author of this thesis.

10.1. Characterization of the Power Supply

The Power Supply Unit version 3 (see section 9.1.1.3) has been characterized using a Fluke 8846A digital multimeter, which has an absolute accuracy of about 4 mV [35].

Two sets of measurements have been performed:

- Output voltage depending on nominal voltage for three different temperatures
- Output voltage depending on temperature for four different nominal voltages

For the first set of measurements, temperatures of −3 °C, 25 °C and 61.6 °C have been chosen and kept stable to better than 0.1 °C.

For each nominal output voltage (10 V to 73 V in steps of 0.5 V), 20 voltage measurements have been taken and their average stored together with their standard deviation. For the measurements at −3 °C and 61.6 °C two runs[1] have been recorded. At 25 °C 21 runs have been recorded. The different measurement runs are averaged and plotted. The error bars show an estimate for the complete standard deviation, indicating the deviations shown with the single voltage measurements (not the error on the calculated mean). As all single voltage measurements have not been stored to compute the complete variance, only an estimate can be calculated. For each temperature the variances (square of the standard deviations) of the individual points from the different measurement runs are added together with the variance of the mean values of the different measurement runs. For most voltages the variance of the different mean values is significantly larger than the individual variances of the measurements and dominates the resulting variance estimate. The measurements are shown in figures 10.1 and 10.2 as a difference and as a ratio to the nominal voltage.

At 25 °C only a small difference (< 10 mV) between measured and nominal voltage is visible over the whole voltage range. At higher or lower temperatures a difference of up to 30 mV is apparent. Minor structures are visible in the measurement which is identical at all temperatures. This likely originates from slight non-linearities in the ADC used in the PSU for adjustment. The calibration constants in the PSU have been chosen such that at 25 °C the mean deviation from the nominal voltage is zero. At nominal voltages below 25 V (range not used for AMD operation) a significant deviation from the linear behavior becomes apparent. In case a new PSU is built or the calibration constants in the existing PSU are changed, this range should be excluded to achieve an even smaller deviation in the relevant voltage range above 30 V.

It can be seen that the ratio of the output voltage V_{out} to the nominal voltage V_{nom} is stable for voltages above 30 V, which is well below the expected operation voltage for the used SiPMs. Using a linear regression through the three constants, a simple description of the voltage of

[1] measurement of every voltage, so 127 voltage steps per run

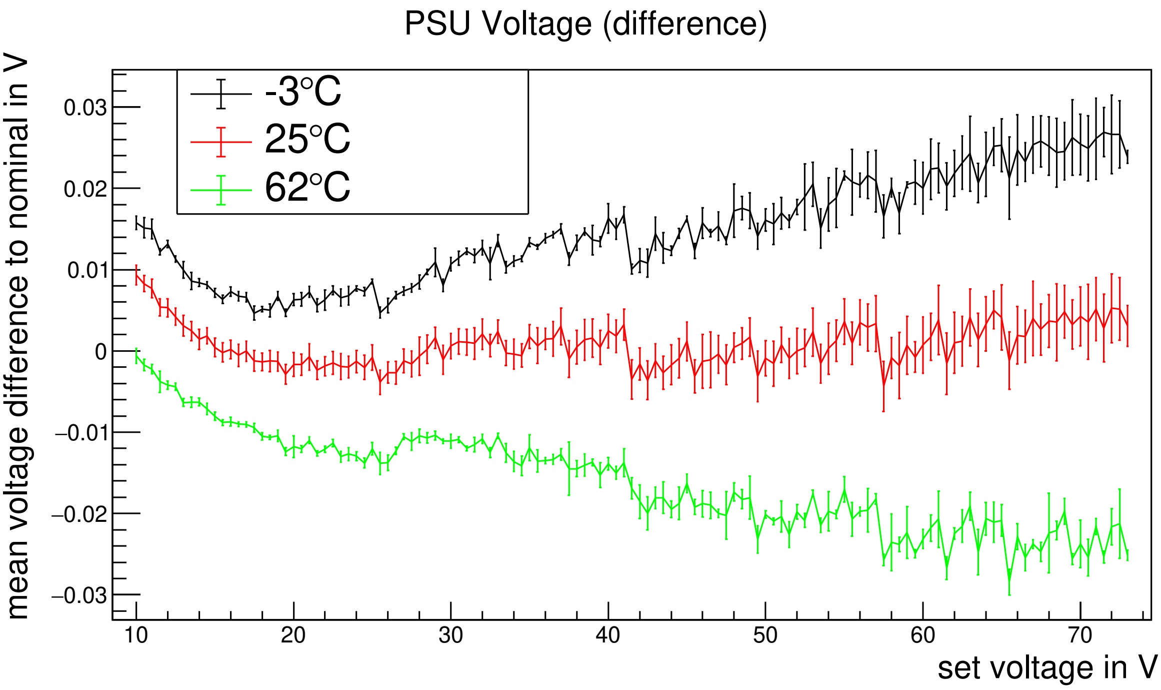


Figure 10.1.: Difference between PSU SiPM output voltage and selected voltage for the PSU V3 for three different temperatures. The estimated standard deviation over all voltage readings at each point is shown as error bars.

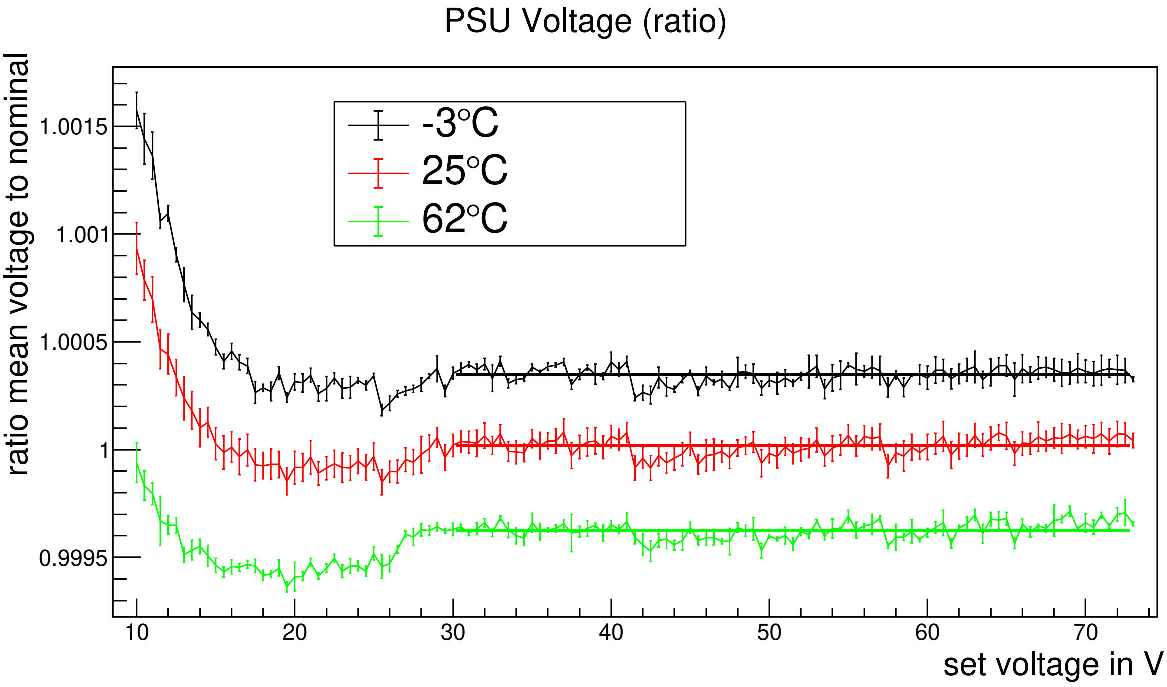


Figure 10.2.: Ratio of PSU SiPM output voltage to selected voltage for the PSU V3 for three different temperatures. The estimated standard deviation over all voltage readings at each point is shown as error bars. A constant line is shown for each temperature, describing the curve reasonably well for selected voltages above 30 V.

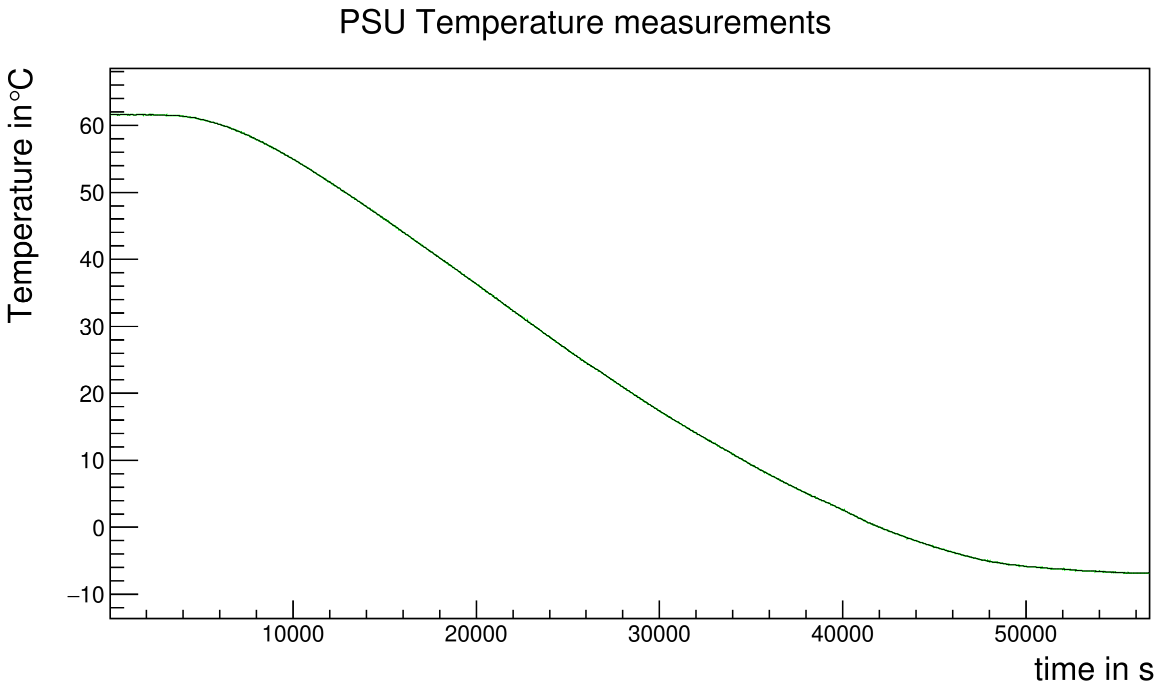


Figure 10.3.: Temperature evolution during the measurements of the temperature dependence of the PSU V3.

$$V_{\text{out}} = V_{\text{nom}} \cdot \left(1 - \frac{1.113 \cdot 10^{-5}}{°\text{C}} \cdot (T - 27.6°\text{C})\right) \tag{10.1}$$

can be derived. The temperature dependence of the ratio of output voltage over nominal voltage is then identical for all voltages:

$$\frac{V_{\text{out}}}{V_{\text{nom}}} = 1 - \frac{1.113 \cdot 10^{-5}}{°\text{C}} \cdot (T - 27.6°\text{C}) \tag{10.2}$$

For the second set of measurements the output voltage has been measured with a slowly changing temperature between 61.5 °C and −7 °C (see figure 10.3). During the single temperature ramp, nominal voltages of 15 V, 40 V, 59 V and 70 V have been cycled through continuously. For each voltage 20 individual voltage readings have been taken before switching to the next voltage. The temperature can be assumed to have been stable over the 20 voltage readings that needed only a few seconds to take. Most of the measurement time was spent waiting for the new voltage to settle.

In figure 10.4 the measurements are shown with the spread (difference between maximum and minimum values) in the 20 individual readings shown as error bars. Linear fits to the data are depicted as solid lines, showing good agreement with the data. Also equation 10.2, shown as the dotted magenta line, can reproduce the measurements at voltages of 40 V, 59 V and 70 V reasonably well.

For the absolute data shown in figure 10.5, equation 10.1 is plotted as dotted lines for the temperatures above 30 V. It shows that the parametrization in equation 10.1 de-

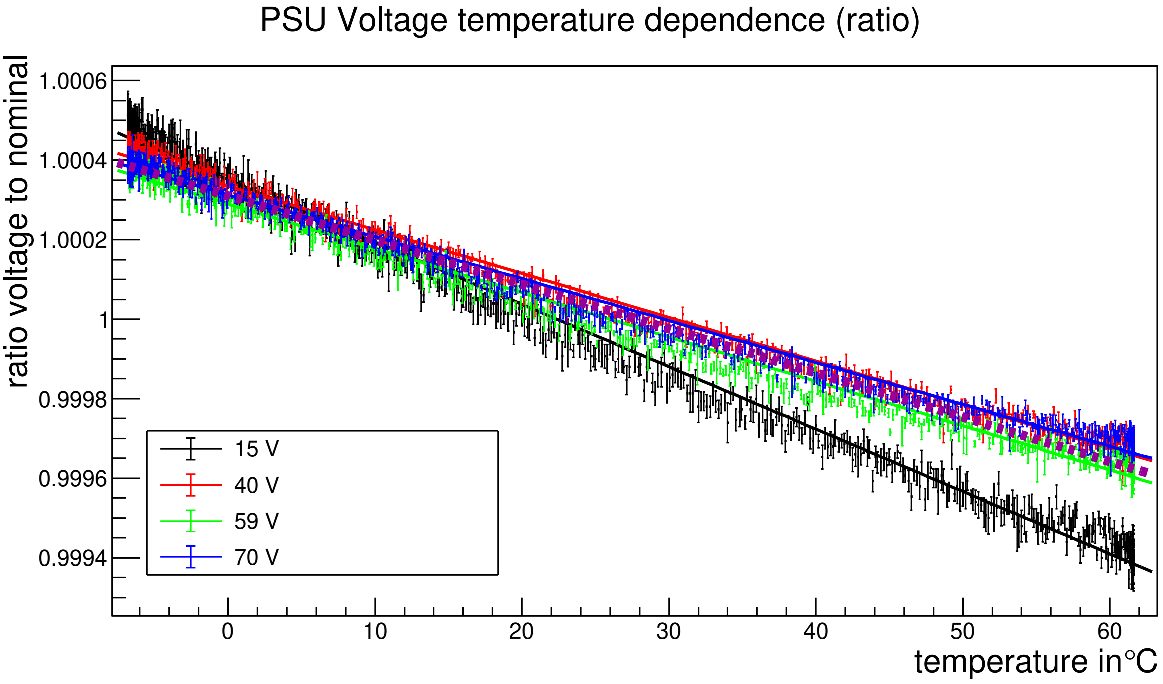


Figure 10.4.: Temperature dependence of the ratio of PSU SiPM output voltage to selected voltage for the PSU V3. The spread (difference between maximum and minimum values) of the 20 voltage readings at each point is shown as error bars. The used temperature curve is shown in figure 10.3. The solid lines show linear fits to the data points, the magenta dotted line depicts equation 10.2 (same for all voltages).

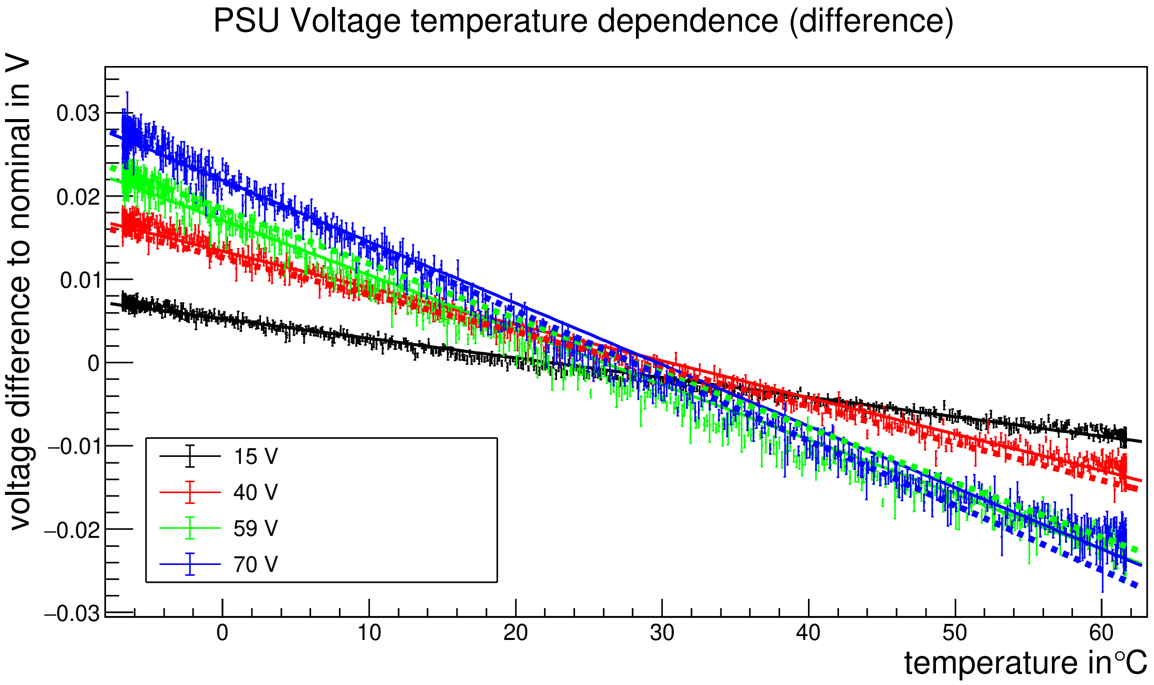


Figure 10.5.: Temperature dependence of the difference between SiPM output voltage and selected voltage for the PSU V3. The spread (difference between maximum and minimum values) of the 20 voltage readings at each point is shown as error bars. The solid lines show linear fits to the data points, the dotted lines depict the expectation from equation 10.1 for temperatures above 30 °C.

scribes the measured PSU output voltages[2] with an accuracy better than 10 mV (corresponding to about 0.5 % difference in SiPM gain at 2 V overvoltage). It is therefore sufficient to use this simple formula to compensate for the PSU temperature dependence during normal operation.

10.2. Characterization of the EASIROC ASIC

10.2.1. Characterization of DACs

DACs are employed in two places in the EASIROC. For each channel there is one 8-bit input DAC which can do a fine regulation of the ground level seen by each SiPM to facilitate a regulation of the operation voltage. For all channels together there is one 10-bit DAC providing the threshold value for the discriminators in the chip.

DACs usually are desired to be linear and temperature-independent. Under the assumption of the former, the relation between the selected DAC count and the resulting voltage can be described by a simple linear function with two parameters[3]. A possible temperature dependence for the DAC can then be expressed as a temperature dependence of these two parameters.

The DACs in the EASIROC work by starting at a reference voltage V_{start}, either generated internally or provided externally, and then subtracting a voltage depending on the selected DAC value. For each active bit n in the DAC value, a specific voltage V_n is subtracted:

$$U_i = V_{start} - \sum_{n=0}^{N-1} b(i,n) \cdot V_n \tag{10.3}$$

V_n is the voltage value of bit n, with $n = 0$ being the lowest valued (least significant) bit.
$b(i,n)$ is 1 if bit n is active in the number i, 0 otherwise[4].

For the case of ideal linearity, the voltage value of each bit is exactly twice the voltage value of the preceding bit:

$$V_n = V_0 \cdot 2^n \tag{10.4}$$

Due to production tolerances of the EASIROC, however, the voltage values of the bits can deviate from their ideal values. This leads to non-linearities visible as steps that are greater or smaller than the nominal step width of V_0, including possible negative steps, breaking the monotony of the DAC-value to voltage relation.

[2] Shown for nominal voltages above 40 °C, but the parametrization is expected to work for temperatures above 30 °C as seen in figure 10.2.

[3] slope and offset

[4] in C source code: `(i&(1<<n))?1:0;`

To parametrize such a DAC, it no longer takes only two temperature-dependent parameters but one parameter per bit plus one for the start value.

The value of a bit can be calculated from the measured voltages U_i through an averaging over the voltage difference of all pairs of DAC values in which only the studied bit differs. That is, averaging over all combinations of the other bits, calculating the sole effect of switching on the studied bit. For a DAC with N bits, this can be written using a sum over the 2^{N-1} combinations of the other bits:

$$V_n = \frac{1}{2^{N-1}} \sum_{i=0}^{2^{N-1}-1} \left(U_{bit_n==off;bits_{\overline{n}}==i} - U_{bit_n==on;bits_{\overline{n}}==i} \right) \tag{10.5}$$

$$= \frac{1}{2^{N-1}} \sum_{i=0}^{2^N-1} (-1)^{b(i,n)} \cdot U_i \tag{10.6}$$

In cases where some voltage values U_i can not be used, for example when the DAC goes into saturation, the affected voltage pair has to be removed from the averaging.

Any uncorrelated and identical statistical uncertainty $\sigma(U)$ on U_i can be propagated to $\sigma(V_n)$ with Gaussian error propagation:

$$\sigma(V_n) = \sqrt{2^N} \frac{\sigma(U)}{2^{N-1}} = \frac{\sigma(U)}{2^{\frac{N-2}{2}}} \tag{10.7}$$

Any systematic constant offset cancels out, any systematic scale error on U directly transfers to V.

The start value V_{start} can be directly set to U_0, the DAC value where all bits are switched off. The start value V_{start} can also be calculated from any U_i by adding the enabled bits (inversion of equation 10.3) and subsequently as the mean $\overline{V}_{\text{start}}$ over all values:

$$\overline{V}_{\text{start}} = \frac{\sum_{i=0}^{2^N-1} \left(U_i + \sum_{n=0}^{N-1} b(i,n) \cdot V_n \right)}{2^N} \tag{10.8}$$

$$= \frac{\sum_{i=0}^{2^N-1} U_i + \sum_{n=0}^{N-1} \left(\sum_{i=0}^{2^N-1} b(i,n) \right) \cdot V_n}{2^N} \tag{10.9}$$

$$= \frac{\sum_{i=0}^{2^N-1} U_i + \sum_{n=0}^{N-1} \frac{2^N}{2} V_n}{2^N} \tag{10.10}$$

This last step uses the fact that each bit i is enabled in half of all possible numbers n.

It can be deduced further:

$$\begin{aligned}
\overline{V}_{\text{start}} &= \frac{1}{2^N}\sum_{i=0}^{2^N-1} U_i + \frac{1}{2}\sum_{n=0}^{N-1} V_n && (10.11)\\
&= \frac{1}{2^N}\sum_{i=0}^{2^N-1} U_i + \frac{1}{2}\sum_{n=0}^{N-1} \frac{1}{2^{N-1}}\sum_{i=0}^{2^N-1} (-1)^{b(i,n)} \cdot U_i && (10.12)\\
&= \frac{1}{2^N}\sum_{i=0}^{2^N-1} U_i + \frac{1}{2}\cdot\frac{1}{2^{N-1}}\sum_{i=0}^{2^N-1} U_i \sum_{n=0}^{N-1}(-1)^{b(i,n)} && (10.13)\\
&= \frac{1}{2^N}\sum_{i=0}^{2^N-1} U_i + \frac{1}{2^N}\sum_{i=0}^{2^N-1} U_i \cdot (N - 2\cdot \text{bits}(i)) && (10.14)\\
&= \frac{1}{2^N}\sum_{i=0}^{2^N-1} U_i \cdot (1 + N - 2\cdot \text{bits}(i)) && (10.15)
\end{aligned}$$

Where $\text{bits}(i)$ is the number of enabled bits in i.

The statistical uncertainty $\sigma(\overline{V}_{\text{start}})$ for uncorrelated and equal errors $\sigma(U)$ on all U_i can be estimated using error propagation:

$$\sigma(\overline{V}_{\text{start}})^2 = \frac{1}{2^{2N}}\sigma(U)^2 \sum_{m=0}^{N} \binom{N}{m} \cdot (1 + N - 2\cdot m)^2 \tag{10.16}$$

Any systematic offset of the measured values as well as any systematic linear scale error transfers directly onto V_{start}. Other systematic effects (like for example a slight non-linearity in the first points) are much reduced compared to when using only U_0.

For the special case of $N = 8$ this results in:

$$\begin{aligned}
\sigma(\overline{V}_{\text{start}})^2 &= \frac{1}{2^{16}}\sigma(U)^2 && (10.17)\\
&\left(1\cdot 9^2 + 8\cdot 7^2 + 28\cdot 5^2 + 56\cdot 3^2 + 70\cdot 1^2 + 56\cdot 1^2 + 28\cdot 3^2 + 8\cdot 5^2 + 1\cdot 7^2\right) && (10.18)\\
&= \frac{1}{2^{16}}\sigma(U)^2\left(81 + 8\cdot 49 + 28\cdot 25 + 56\cdot 9 + 70 + 56 + 28\cdot 9 + 8\cdot 25 + 49\right) && (10.19)\\
&= \frac{2304}{2^{16}}\sigma(U)^2 && (10.20)\\
\sigma(\overline{V}_{\text{start}}) &= \frac{48}{2^8}\sigma(U) = \frac{3}{16}\sigma(U) = 0.1875\,\sigma(U) && (10.21)
\end{aligned}$$

This statistical uncertainty is much lower than when using $V_{\text{start}} = U_0$.

10.2.2. Input DACs

The 32 8-bit input DACs of the EASIROC are used for fine-regulation of the SiPM supply voltage through an adjustment of the effective ground level for the respec-

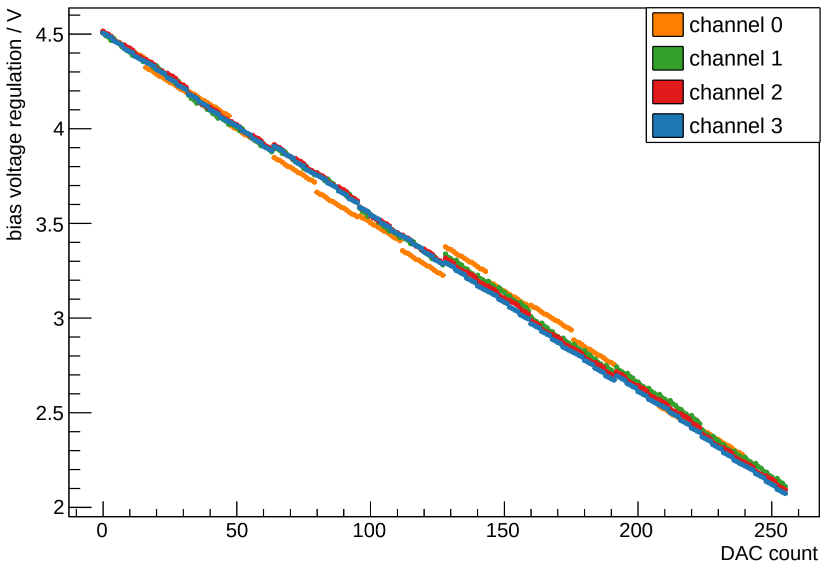


Figure 10.6.: Output voltage for the 8-bit input DACs of four EASIROC channels. Big steps and channel-to-channel fluctuations make an individual calibration for the channels necessary. Taken from [58].

tive SiPM. As the SiPM properties depend directly on the applied overvoltage, it is important to keep it stable and at a known value. For the SiPM gain for example, using the recommended overvoltage of 2.6 V, a 1 % gain difference is caused by a voltage difference of 26 mV, which also corresponds to about 0.5 °C difference in temperature. This means for the goal of 2 % gain stability, the operation voltage has to be kept stable to better than 50 mV.

The characterization measurements of the input DACs have been performed by Rebecca Meißner and written down in her master thesis [58], where all the details and results can be found. Some details have also been published in [66].

The measurements in figure 10.6 show big steps with a high channel-to-channel fluctuation, caused by a deviation of the value of the individual bits from their nominal values. While the nominal step size of an 8 bit DAC with a range of 2.5 V is 10 mV, most channels have step sizes up to about 25 mV. Worst-case step sizes of up to 75 mV for single channels[5] can be observed. The method described in section 10.2.1 can successfully be applied to completely describe the output voltage of the input DACs, including all steps. The temperature dependence of the DACs can be described as a simple linear temperature dependence of the bit values, which is shared between all channels and chips. The difference between the applied voltage and the maximum output of 4.5 V changes by a factor of about $3 \cdot 10^{-3}/°\mathrm{C}$ ($\equiv 7.5\,\mathrm{mV}/°\mathrm{C}$ at the maximum possible difference of 2.5 V).

Due to the high temperature dependence and deviation of the DAC behavior from

[5] Up to 150 mV when the output range of the DACs is not reduced to 2.5 V as described in section 9.1.3.

the ideal linear case, corrections according to these calibration measurements have to be applied during detector operation. With the Power Supply Unit Version 3, which allows for a change of the external voltage applied to the SiPMs, the operation region of the DACs can be shifted. It can be chosen such that the achieved operation voltage deviates much less than the goal of 50 mV from its desired value. In any case the applied voltage can be known to better than 15 mV for offline corrections during data analysis (including temperature uncertainties).

For more details see [58].

10.2.3. Threshold DAC and discriminators

One 10-bit DAC in the EASIROC is used to provide the common threshold voltage for all 32 channel trigger discriminators. Its value can be selected using the slow control configuration of the EASIROC ASIC. It is not affected by the load signal but is fed from the configuration shift register directly, leading to arbitrary voltages being generated during the EASIROC configuration. After changing the slow control configuration bits in the EASIROC, it takes about 1 ms until the threshold voltage is stable again. The threshold voltage is provided at a pin of the EASIROC to allow characterization.

For a characterization of the 10-bit threshold DAC, the voltage at each DAC setting has been measured for four temperatures between −6 °C and 67 °C using a Fluke 8845A digital multimeter with an accuracy of about 5 mV [35]. The measurement results are shown in figure 10.7.

While the start value at DAC count 0 is relatively unchanged with temperature, the slope is different, leading to a shift of about 50 DAC counts at the highest DAC counts (lowest threshold) over the 73 °C temperature range of the measurement.

The calculated DAC bit values normalized to bit 9 and scaled by their ideal value of 2^{bit} are plotted in figure 10.8. The bits values deviate by up to 8 % from their ideal values.

The simple bit model described in section 10.2.1 with linear temperature dependence can describe the measured values with only minimal deviations in the relevant range, as can be seen in figure 10.9.

The spike at a DAC count of around 915 coincides with the extremely fast discriminator switching due to baseline noise. The position of this spike can be used to determine the position of the baseline. The asymmetry of the spike can be understood through the fact that as soon as the effect onsets, the threshold is increased, causing it to either increase the effect (right edge) or decrease the effect (left edge). The highest point of the spike on the right edge shift from DAC counts 929 to 914 for increasing temperatures in the measurement range. This is much less than the 50 DAC counts shift of the actual threshold voltage, leading to the conclusion that also the baseline of the signal after the fast shaper in the EASIROC shifts, counteracting

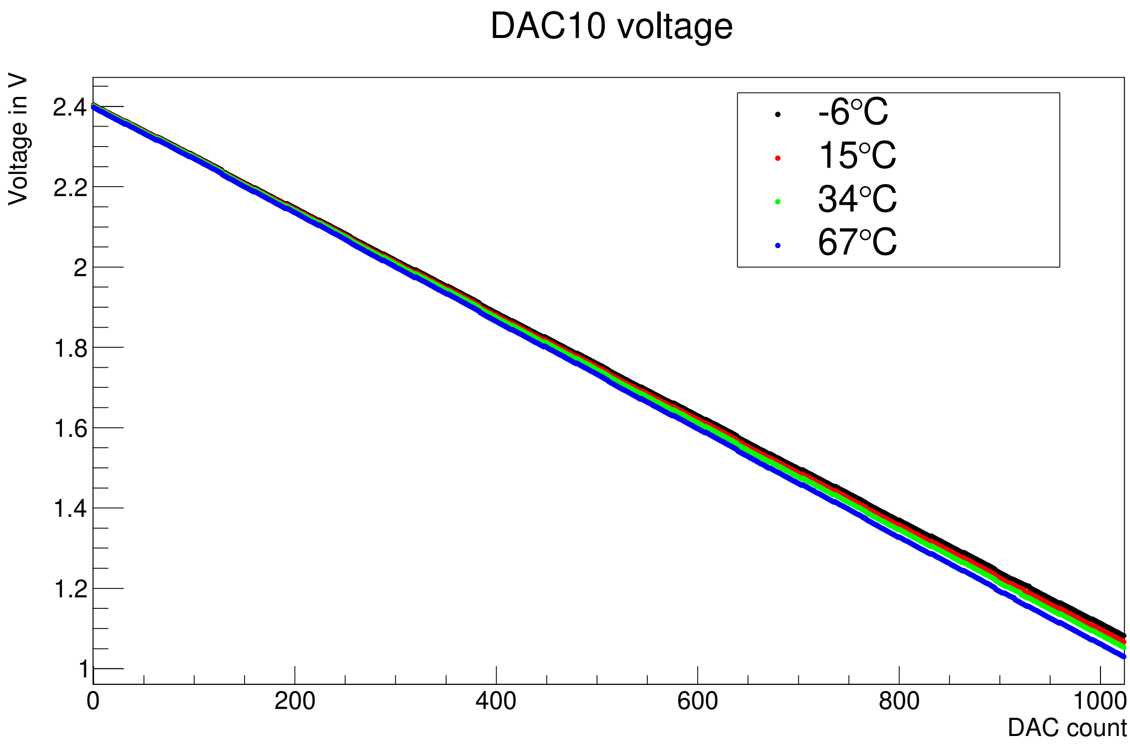


Figure 10.7.: Measurement of the voltage output of the 10-bit DAC for the trigger threshold of one EASIROC at different temperatures.

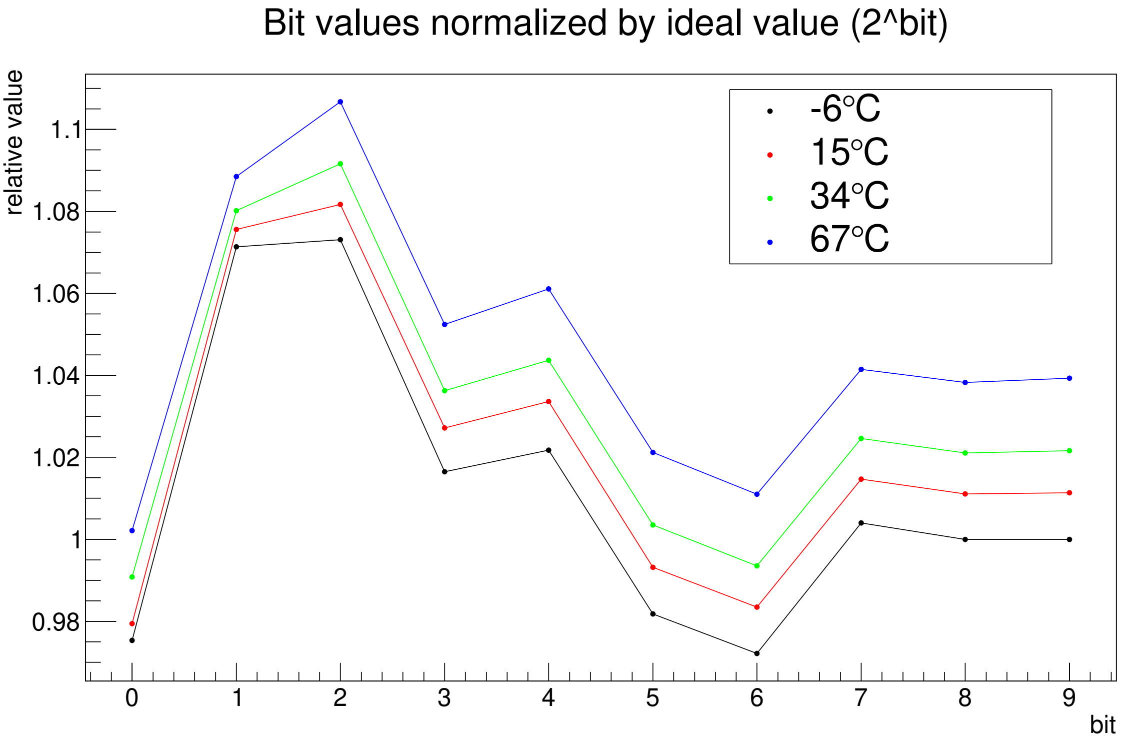


Figure 10.8.: Calculated bit values for each of the measurement series. Normalized to bit 9 of the first measurement and scaled by 2^{bit} (the ideal value). In the ideal case flat lines are expected.

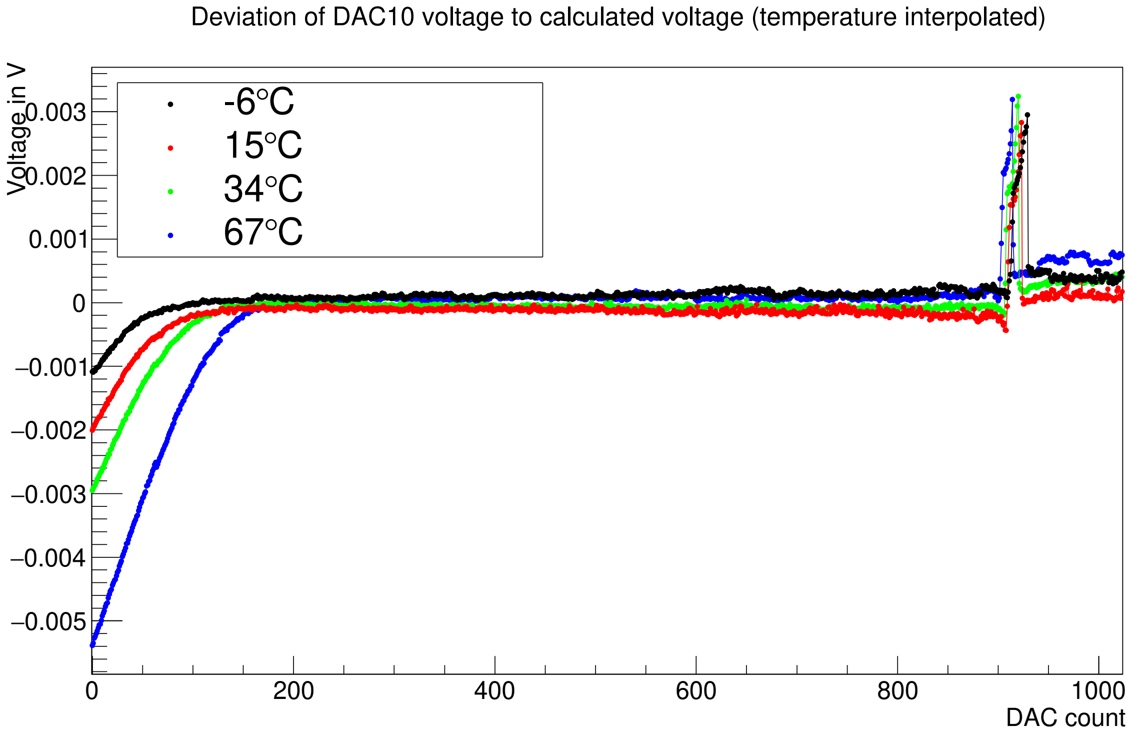


Figure 10.9.: Deviation of the measured DAC voltage from the voltages calculated from the measured DAC bit values. A global description of the bit values including the temperature dependence is used instead of the bit values calculated for each individual measurement series.

most of the threshold shift.

The shift at the highest DAC counts (lowest threshold) is not understood, but is only very small (less than 1 DAC count) and the range is not used during normal operation anyway.

The non-linearity at DAC values below 200 (highest thresholds) indicates a small shift in the DAC slope. At DAC values below 200, also a saturation of the SiPM signal in the trigger path of the EASIROC can be observed, which is much greater than the non-linearity of the threshold DAC. For this reason this effect does not cause a problem.

For the analysis of the DAC bit values, only DAC values between 200 and 890 have been used. This range can almost perfectly be described by the simple bit model (see figure 10.9). The excluded ranges are either in or below the baseline (values above 890) or above the linear range of the preamplifiers and shapers. Even for the excluded ranges the deviations from the ideal values are less than five DAC counts.

The measured voltage values can be used to convert the abstract DAC counts, which have an inverted direction with lower values corresponding to higher voltages, into voltages for an easier understanding. The position of the spikes in figure 10.9 can be used to determine the effective voltage above the threshold. To compensate for temperature effects in setting the threshold voltages, so that the threshold is kept stable, the measurements are of limited use due to likely gain differences in the amplifiers and the apparent shift of the baseline of the fast shaper. Furthermore, the ideal threshold in p.e. likely changes with temperature due to changes in the SiPM dark noise allowing lower thresholds with a higher trigger efficiency for muons while keeping the dark noise trigger rate constant.

More useful but also more difficult is the identification of p.e. steps in measurements of the EASIROC trigger rate in dependence of the selected threshold at different temperatures with connected SiPMs. That way the complete signal chain is calibrated.

Such measurements have been performed by Rebecca Meißner for her master thesis [58]. It could be shown that there is a significant temperature dependence in the relation of p.e. to DAC counts. A second important result is that the amplitude of the fast shaper output shows only a small dependence on the selected preamplifier gain/feedback. While there is a nominal factor of 15 between highest and lowest gain setting and the slow shaper peak value shows a factor of 8 to 10 (depending on shaping constant), the fast shaper output only shows roughly a factor of 2. It is therefore not possible to increase the trigger threshold in p.e. by a big factor. On the other hand it is possible to set the gain as needed for the slow shaper measurements without giving a big consideration to its influence on triggering.

10.2.4. Slow Shapers

The performance and characteristics of the slow shapers influence the overall performance of the AMD to determine the number of muons in high-flux scenarios[6], where tiles are simultaneously hit by multiple particles and simple hit/no-hit determination is not sufficient. The slow shaper characteristics determine which time window of muon arrival times can be observed and how muons arriving in this time window contribute with different weights. The precise influence will be studied in [65].

Even though the EASIROC has an analog probe output which can be used to monitor the signal of one of the two slow shapers for any one of the 32 channels, this can not be used to properly characterize the slow shapers. Due to power consumption constraints, the slew rate and bandwidth of the output is limited so it does not perfectly resemble the actual on-chip signal. Similar constraints apply to the analog multiplexer outputs which can be setup to permanently follow the output of a single slow shaper through a constant selection of the specific channel for readout[7] and the not-assertion of the hold signal for the track and hold cells.

Due to these constraints, the measurement of the slow shaper pulse form has to occur using the normal readout including the track and hold cells and the external ADCs. This leads to the possibility to sequentially sample one point of the slow shaper trace per event through a variation of the hold delay after a signal (step size of 2.5 ns). Using this possibility, each measurement point originates from a different input pulse. It entails either the necessity of a big number of measurements per point to get a good average over the different pulse heights, or pulses very similar in form and height. Because this measurement method only works for the slow shaper but not for the fast shaper, it is not possible to easily characterize the fast shaper of the EASIROC.

To characterize the slow shapers, an external pulser is used as an input signal for the EASIROC. Pulses are generated using digital general purpose I/O-pins of an external FPGA[8], connected to the EASIROC analog inputs via a 10 kΩ resistor each. Using a 400 MHz clock in the FPGA, roughly trapezoidal pulses with a lengths of a few nanoseconds (about 10 % of the shortest peaking times) can be achieved. As shown later, the pulses are short enough that their exact form does not matter. The separate *in_calib* input of the EASIROC, which could in principle also be used for this measurement, is not used, because its exact influence on the result is unknown. The pulser is triggered by a digital signal generated in the FPGA on the EASIROC board. Only one input channel is pulsed at a time to prevent excessive noise from the simultaneous switching of multiple channels. The first time sample is chosen about 50 ns before the trigger signal for the pulse generator is sent to get a measurement of the baseline before the pulse. All measurement series shown here are shifted to have this baseline at zero.

[6] stations close to the shower core and for high primary particle energies

[7] Usually outside of event readout the multiplexer outputs are kept in high-Z mode.

[8] a Cyclone 4 EP4CE22 on a DE0-nano development board

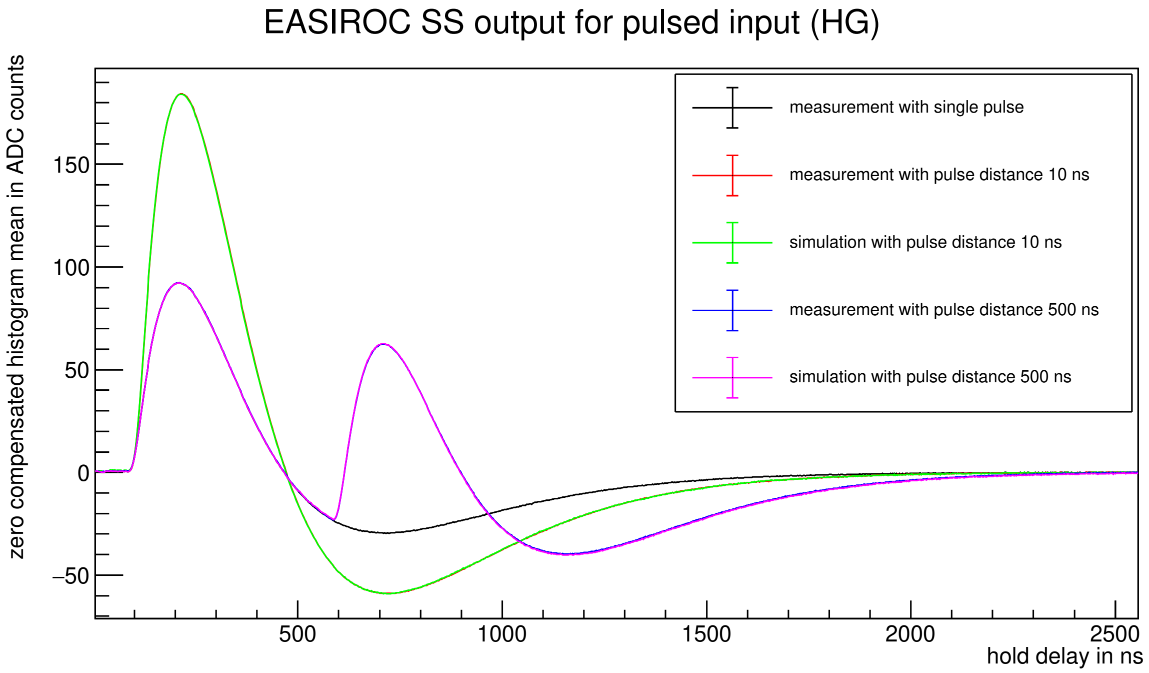


Figure 10.10.: Measurement of the high-gain slow shaper output with 125 ns nominal peaking time and feedback setting 3. Two consecutive pulses of 5 ns length each are generated by the FPGA with various start time differences. The shown simulations are the additions of two shifted single pulse measurements. The simulations agree so well with the double pulse measurements that the measurements are completely hidden. The first part of the signal until the onset of the second pulse is identical for all measurements.

Through a separate serial communication interface between the PC and the pulser FPGA, the pulse length and the enabled channels can be selected. Also a second pulse following with a selectable time delay after the first pulse can be enabled. With the usage of two separate consecutive pulses it can be tested if the inputs and shapers are linear, which means in this case that the resulting signal is equal to the sum of two shifted single input pulse signals. A measurement using two consecutive pulses is shown in figure 10.10, confirming a good linearity. The calculated sum of the two shifted single pulses is identical to the measurement of two consecutive pulses within the statistical fluctuation of the measurement points. In the plot they appear as only a single line.

Especially for the low-gain path, which only shows a weak signal, the noise influence of switching in the digital trigger path of the EASIROC can be observed as dips in the measured shaper output (see figure 10.11). Especially the *OR32* signal[9] is clearly visible. Therefore, for the final characterization measurements the discriminators of the EASIROC and all trigger outputs have been disabled.

The form of the shaper output does not change significantly with a doubling of the

[9] A fast OR combination of all activated trigger channels generated in the EASIROC chip.

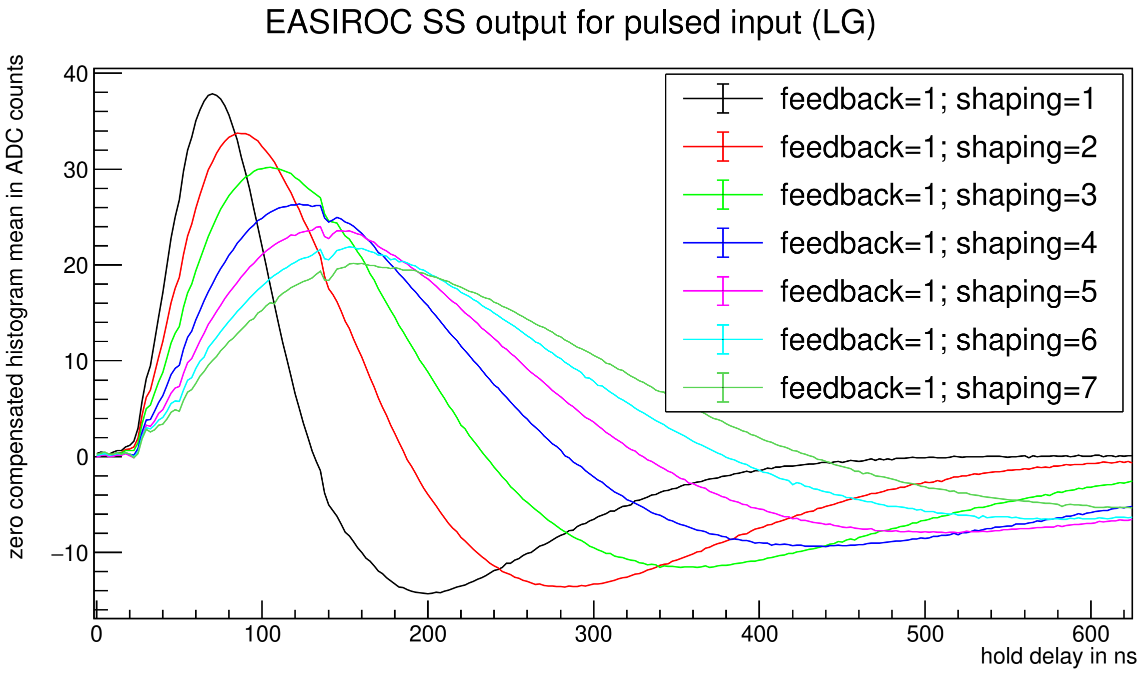


Figure 10.11.: Measurement of the slow shaper output with highest gain setting for different shaping settings in the low-gain channel. The crosstalk from the digital trigger path is clearly visible as dips in the traces.

input pulse width, which leads to the conclusion that the used pulses are sufficiently short so that their exact form does not influence the result.

For different EASIROC channels the pulse form is virtually identical for the same shaping and gain settings, but there is a small deviation in the pulse height. It has not been further investigated whether this result is caused by variations in the EASIROC itself or caused by deviations in the different digital FPGA I/O pins used to generate pulses for the individual channels. Any real differences in the channels can be easily calibrated out together with the SiPM and tile variations using single muon measurement data in the final detector.

Except for gain differences and therefore different noise levels in the measurements, the pulse forms for the high and the low-gain channels are identical for matching gain and shaping settings.

The height of the maximum of the peak above the baseline level before the pulse is extracted from the measurements. Due to the usage of input pulses of an arbitrary height the absolute height of the output pulses is also arbitrary. It is normalized to the value for shaping and feedback settings of 1 and given in table 10.1. The second value extracted from the measurements is the position of the peak maximum, called *peaking time*. The choice of the zero position is arbitrary and has been chosen by hand to be the first sample point where a rise above the baseline becomes visible for any shaper and feedback setting. The same zero position relative to the trigger signal to the pulser has been chosen for all measurements. The uncertainty on the manual

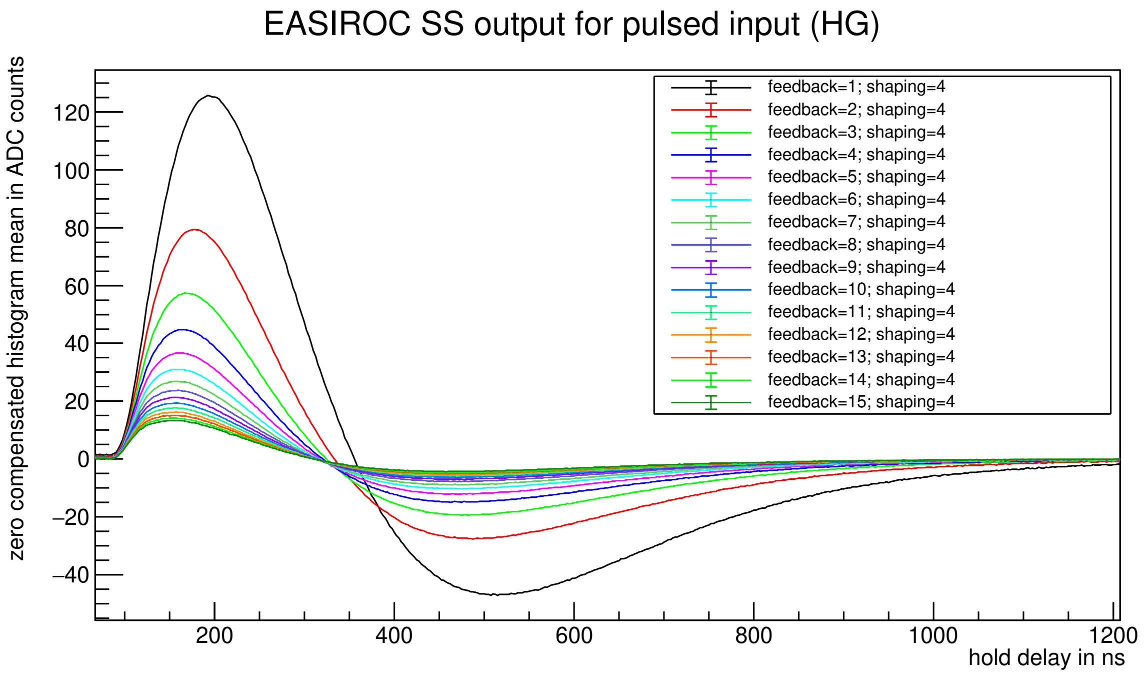


Figure 10.12.: Measurement of the high-gain slow shaper output of one channel with different feedbacks (nominal gain ∝ 1/feedback) for one shaping setting. The peaking time becomes slower for higher gain.

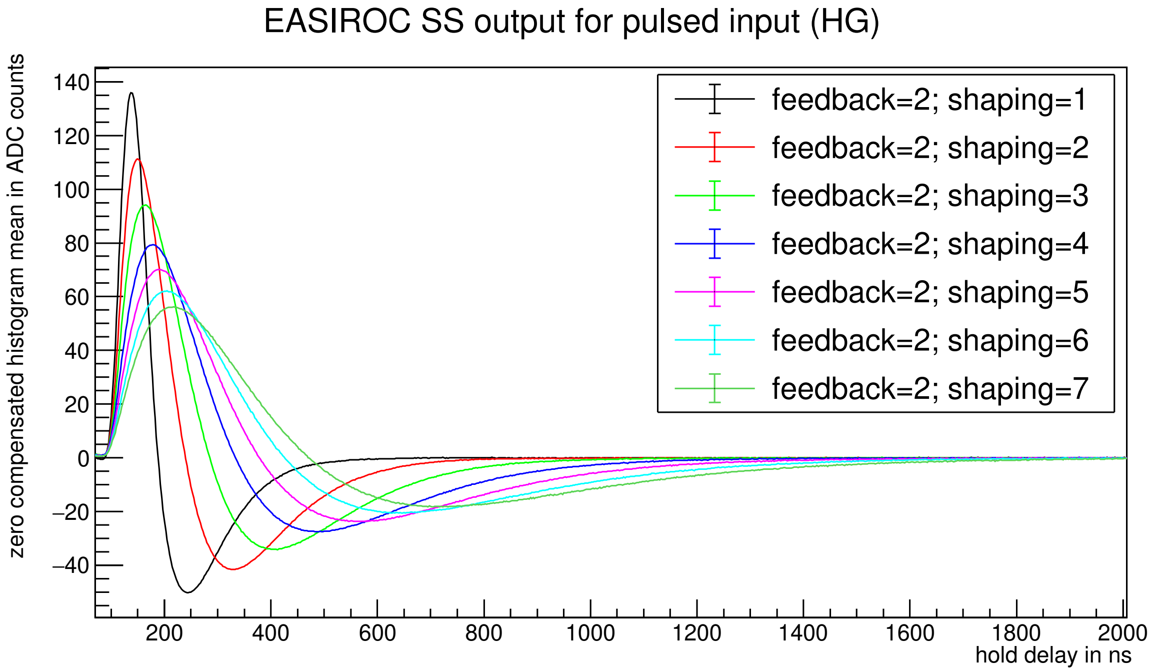


Figure 10.13.: Measurement of the high-gain slow shaper output of one channel with different shaping settings for one gain/feedback setting. The height of the peak becomes significantly lower and the peak broadens for longer peaking times.

shaping feedback	1 25 ns	2 50 ns	3 75 ns	4 100 ns	5 125 ns	6 150 ns	7 175 ns
1	1	0.879	0.786	0.686	0.618	0.559	0.511
2	0.741	0.606	0.513	0.433	0.382	0.339	0.307
3	0.589	0.463	0.379	0.313	0.274	0.24	0.217
4	0.487	0.373	0.298	0.244	0.213	0.187	0.167
5	0.414	0.313	0.247	0.2	0.174	0.152	0.137
6	0.359	0.269	0.21	0.169	0.147	0.128	0.116
7	0.317	0.235	0.183	0.146	0.127	0.112	0.1
8	0.283	0.209	0.161	0.129	0.112	0.098	0.088
9	0.255	0.188	0.144	0.116	0.1	0.088	0.079
10	0.233	0.171	0.131	0.105	0.091	0.08	0.071
11	0.215	0.157	0.12	0.097	0.083	0.073	0.065
12	0.199	0.146	0.111	0.089	0.077	0.067	0.06
13	0.185	0.136	0.103	0.082	0.072	0.063	0.056
14	0.174	0.127	0.096	0.077	0.068	0.059	0.052
15	0.164	0.119	0.091	0.073	0.063	0.054	0.049

Table 10.1.: Relative height of the peak maximum of the slow shaper for different shaping and feedback settings for one channel for a short arbitrary input pulse. The pulse height for shaping 1 and feedback 1 is defined as the reference value.

determination of the zero position can be estimated to be about 5 ns and affects all measured peaking times as a common offset.

It can be seen from figures 10.12 and 10.13 and tables 10.1 and 10.2 that the position of the maximum and the height of the maximum of the shaper output each depends on the settings for shaping and feedback/gain. This means that for changing the gain while keeping the real peaking time constant, the shaping setting also has to be changed to compensate for the influence the gain setting has on the peaking time.

There is a significant deviation of the measured peaking times from the nominal values. For a nominal peaking time[10] of 25 ns at a shaping setting of 1, the real peaking time is longer by about a factor of two (depending on the gain setting). For a nominal peaking time of 175 ns the real peaking time is shorter by a factor of up to 1.6. Furthermore, there is a significant deviation of the real gain values from the nominal values, with a nominal factor of 15 between the highest and lowest gain.

[10] "peaking time" is not properly defined in the data sheet but can be assumed to be the time from start of the pulse to maximum of the peak.

shaping	**1**	**2**	**3**	**4**	**5**	**6**	**7**
feedback	**25 ns**	**50 ns**	**75 ns**	**100 ns**	**125 ns**	**150 ns**	**175 ns**
1	55	72.5	90	105	125	137.5	152.5
2	50	62.5	77.5	90	102.5	117.5	130
3	47.5	55	67.5	80	92.5	112.5	125
4	45	52.5	62.5	77.5	85	97.5	115
5	42.5	50	60	75	87.5	102.5	117.5
6	42.5	47.5	57.5	72.5	87.5	97.5	110
7	40	47.5	55	67.5	80	95	110
8	40	45	55	72.5	85	92.5	105
9	40	45	52.5	70	80	97.5	105
10	40	42.5	52.5	67.5	80	92.5	105
11	37.5	42.5	47.5	67.5	80	95	105
12	37.5	42.5	47.5	70	80	92.5	112.5
13	37.5	40	52.5	70	80	87.5	105
14	37.5	42.5	47.5	67.5	80	87.5	105
15	37.5	42.5	47.5	72.5	85	87.5	105

Table 10.2.: Position of the peak maximum in ns (bin size of 2.5 ns) of the slow shaper for different shaping and feedback settings for one channel for a short arbitrary input pulse. The zero position (identical for all settings) has been chosen by hand at the point where the pulses start to deviate from the baseline. For the lower gain (higher feedback setting) measurements there are some fluctuations of the position of the maximum bin entry due to noise.

10.2.4.1. Shaper electronics simulation model

The characterization of the slow shapers can be used to develop a simple phenomenological simulation model. The measurements reflect the response of a shaper to a short input pulse. It has been confirmed that the response to two short input pulses is the direct sum of the responses, which leads to the reasonable expectation that for arbitrary inputs (except when saturation occurs) the output of the slow shaper is a simple convolution of the input pulse and the measured slow shaper response up to a constant proportionality factor. For the simulation the measured slow shaper response S (with the corresponding values for shaping and feedback) is used from just before the start of the leading edge of the pulse to where the baseline is (almost) reached again after the undershoot. As the form of the shaper output is identical for the low-gain and high-gain paths, the measurement of the high-gain path, which shows a much lower noise level, can be used for the simulation of both gain paths.

For the simulation, the SiPM signal input pulse form I_i ($i \in [0, m)$) with discrete sampling points is converted to the output pulse form O_i ($i \in [0, m - l)$) with the same sampling distance. Hereby m is the step count of the input pulse form. The measured shaper response S_i ($i \in [0, l)$) needs to have the same discrete sampling distance, with the l points being calculated from the actual measured points using a linear interpolation where needed.

$$O_n = \sum_{i=0}^{l-1} I_{n+l-1-i} \cdot S_i \tag{10.22}$$

This means that every output point depends on l previous input points (each weighted by the time-inverted shaper response), therefore the usable output has a length which is reduced by l points compared to the input, because the first l input points do not have sufficient preceding points.

The model does not include saturation, especially not the fact that saturation can occur not only in the slow shaper itself but also in the preamplifier, which means that two different pulse forms with the same slow shaper peak value can have different slow shaper peak values when their pulse heights are increased proportionally. Saturation is expected only for high signals, especially when taking into account the possibility of combining high-gain and low-gain signal paths (see section 10.2.4.2).

The absolute pulse height of the shaper output is only simulated up to a constant proportionality factor. Also not simulated in this model is any propagation delay, so that the simulation can not be used to determine timing relative to the internal trigger or other (external) detectors.

As no equivalent characterization measurement for the fast shaper can be done, one simple approximation for the fast shaper simulation is to use the measured slow shaper pulse form also for simulation of the fast shaper. For this simulation the measured curve has to be compressed to account for the shorter peaking time of the fast shaper. This can be done by a variation of the assumed time steps used for measuring the curve. To achieve the nominal 15 ns peaking time of the fast shaper, a conversion factor of 4 has to be used on the measured curve with a shaping setting of 1 and a feedback setting of 1. This simulation is then only a rough estimate of the fast shaper function.

10.2.4.2. High-gain - Low-gain cross calibration

The high-gain and low-gain channels, consisting of separate preamplifiers and slow shapers in the EASIROC, provide an extension of the dynamic range for measurements. The high-gain path, of which the preamplifiers also provide the input of the fast shapers for the trigger discriminators, can achieve a resolution of single photon equivalents, while going into saturation at pulse heights of as low as 40 p.e. (at highest gain setting). The low-gain path, which nominally has a gain reduced by a factor of ten compared to the high-gain path, can not provide single p.e. resolution, but goes into saturation only at about 400 p.e. pulses.

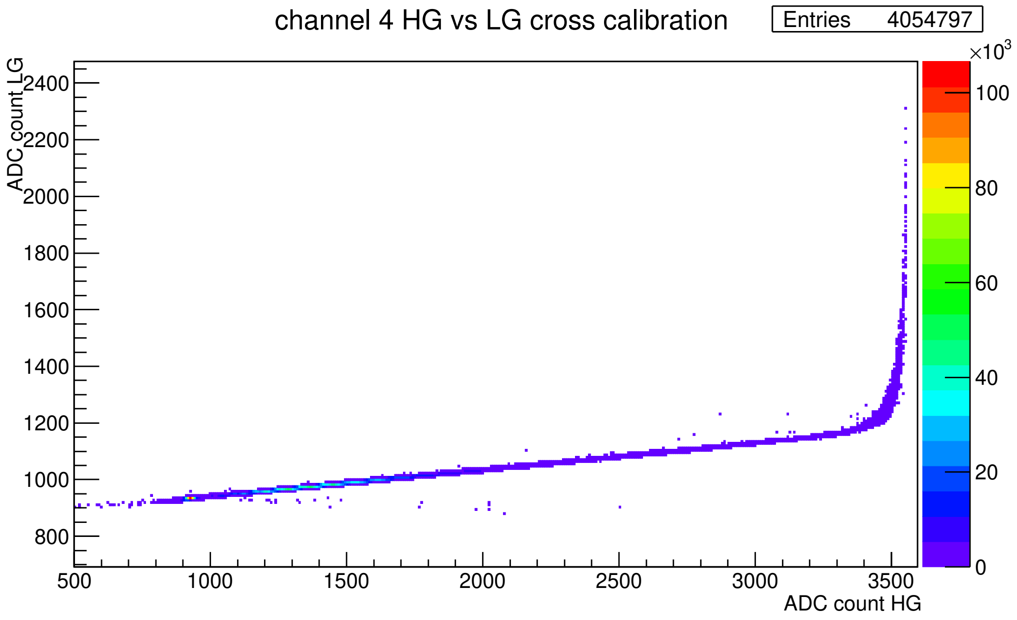


Figure 10.14.: Correlation of pulse height of the high-gain (HG) and low-gain (LG) paths of the slow shapers. Events of a central tile (channel 4) with four of eight tiles in the stack seeing at least 2.5 p.e are analyzed. Highest gain (feedback setting 1) and 125 ns nominal peaking time (shaping setting 5) are used. An almost perfect linear correlation shows for ADC counts up to 3000, with serious saturation starting at 3350.

During normal applications the signals from both signal paths can be combined, with the high-gain path providing the value for signals below its saturation threshold and the low-gain path providing the value when the high-gain path is in saturation[11].

It becomes clearly visible in figure 10.14, that – besides some outliers – for the non-saturated region the relation between high-gain and low-gain values are almost perfectly linear, so a simple linear conversion of the values becomes possible. This exact correlation is only valid for identical shaping characteristics for both high-gain and low-gain paths. This has been confirmed during the slow shaper characterization for identical settings of shaping time and preamplifier gain. As there is only a single hold signal which has to occur in the peak of the shaped signal for both gain paths, this condition has to be fulfilled anyway and is no significant additional constraint.

This cross-calibration can be easily performed with the data set the calibration is to be applied to, assuming the pairs of values for low- and high-gain and not only histograms are stored. The exact values are expected to differ per channel and per feedback/shaping setting because of manufacturing tolerances of the EASIROC and are in this thesis only given for one exemplary case.

[11] For the 12 bit ADCs on the EASIROC evaluation board, saturation starts at an ADC count of about 3000 (from total of 4096), while the baseline is at around 950.

channel	a	b
0	0.0951	848.6
1	0.0944	848.2
2	0.0941	845.0
3	0.0941	846.5
4	0.0932	848.3
5	0.0936	845.1
6	0.0930	843.8
7	0.0941	846.7

Table 10.3.: Exemplary high-gain vs low-gain cross-calibration parameters for equation 10.23.

Performing an analytic linear fit of the function

$$V_{LG} = a \cdot V_{HG} + b \tag{10.23}$$

with free parameters a and b to the pairs (V_{HG}, V_{LG}) (high-gain and low-gain ADC values) of all events with $1000 \leq V_{HG} \leq 3000$ and $900 \leq V_{LG} \leq 3000$ yields the parameters in table 10.3. The values agree between the different channels with a deviation of about 2 % and also agree within 8 % with the nominal value of 0.1.

As can be seen in figure 10.15, over the whole range of $1000 \leq V_{HG} \leq 3000$ any systematic deviation of the linear curve from data is much smaller than the achieved measurement resolution of the data points. This good linear correlation also indicates a good linearity of the high-gain signal path itself up to $V_{HG} \sim 3000$, because it is unexpected for both paths to show exactly the same non-linearity in different relative regions of their operation. Due to the fact that for different input pulse forms or EASIROC settings, saturation effects might start at slightly different V_{HG}, this value should be confirmed and adjusted if necessary for the specific application.

A simple combination of the data can be performed by using the V_{HG} value calculated from V_{LG} by an inversion of equation 10.23 when the originally measured value of V_{HG} is above a decision boundary of e.g. 3000. In this case the boundary becomes clearly visible in the resulting histogram (figure 10.16). This is caused first by the fact that for the low-gain region only about every tenth bin is filled, while for the high-gain region the same data is distributed over ten bins. To overcome this, during conversion the low-gain value can be randomly smeared by a uniform distribution in a range of ±0.5 bins. Through this still a reduction of resolution occurs at the transition, but it is much less apparent. A second cause for a visible boundary comes down to the decision whether the high-gain value or the converted low-gain value is used. In the ideal case this is decided using the “true” value, which leads to a smooth transition, but is impossible to achieve because the “true” value is not known. The problem occurs when the high-gain value and low-gain value are fluctuating in a different direction. Assuming the higher resolution high-gain value is used to decide

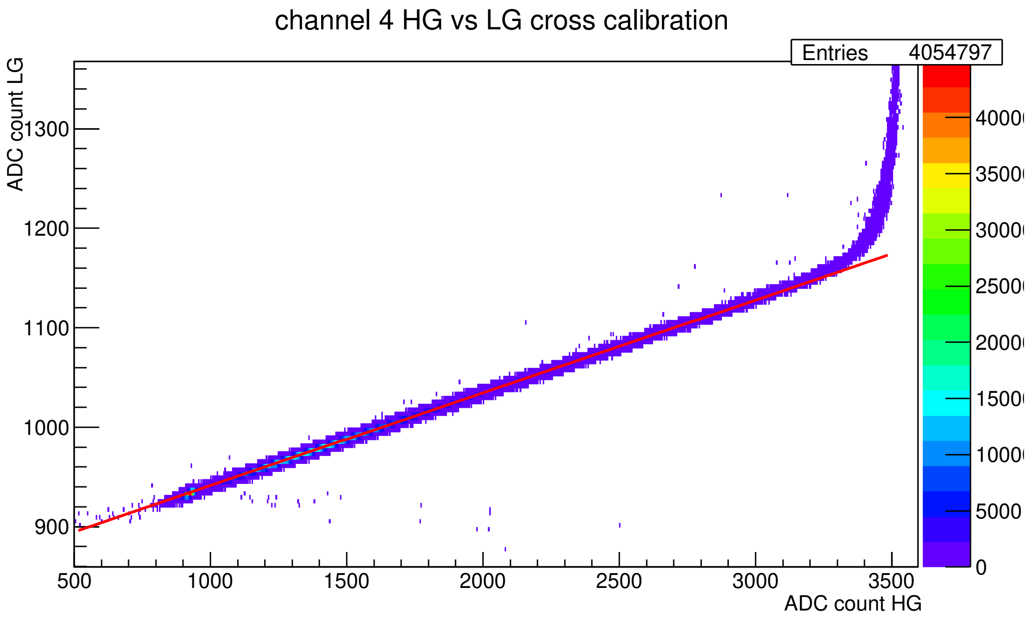


Figure 10.15.: Correlation of pulse height of the high-gain and low-gain paths of the slow shapers. Events of a central tile (channel 4) with four of eight tiles in the stack seeing at least 2.5 p.e. Highest gain (feedback setting 1) and 125 ns nominal peaking time (shaping setting 5) are used. Good agreement with fitted linear calibration function with parameters as in table 10.3. Same data as in figure 10.14, but with fit included.

which value to use, near the decision border in both cases[12] the value fluctuating to a lower value is used. This leads to a higher event count below the decision border and a lower event count directly above the border.

The best decision has been found to be made using the arithmetic mean of the high-gain value and the converted low-gain value. This way counteracting fluctuations cancel each other out. This is shown in figure 10.17. The remaining bump is most likely caused by a slight deviation of the real conversion relation from the used linear function. Restricting the conversion function fit range tighter around the decision value of 2000 ($1500 \leq V_{HG} \leq 2500$), and therefore slightly improving the conversion relation for the used decision value, can remove the bump.

10.2.5. Study of different signal cable lengths

The AMD contains RG-174/U 50 Ω coaxial signal cables between the SiPMs and the EASIROC of about 1 m length. For the rest of the way to the tiles clear optical fibers are used, to transmit the photons directly instead of the weak electrical signals. For other applications (e.g. mini AMD section 8.6.4) longer cable lengths might be desirable.

[12] High-gain value fluctuating below border ⇒ use low high-gain value. High-gain value fluctuating above border ⇒ use low low-gain value.

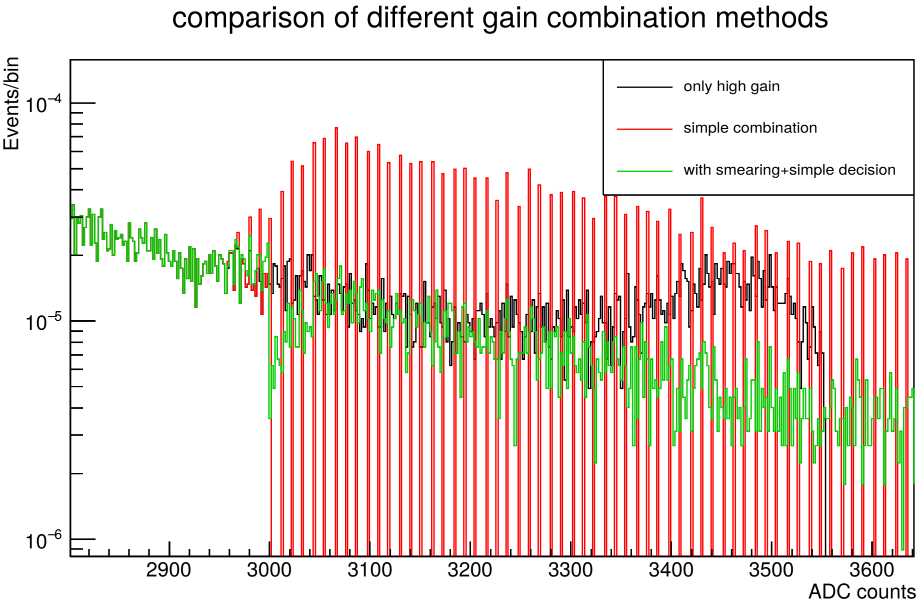


Figure 10.16.: Different methods to combine low-gain and high-gain data. Events of a central tile (channel 4) with four of eight tiles in the stack seeing at least 2.5 p.e. Linear conversion function with parameters as in table 10.3. Decision boundary at 3000 (high-gain) ADC counts. The unconverted high-gain values (black) show saturation, the simple converted low-gain values (red) show discrete values, application of smearing by ±0.5 (low-gain) ADC counts (green) solves this issue but shows a clear dip at the decision boundary.

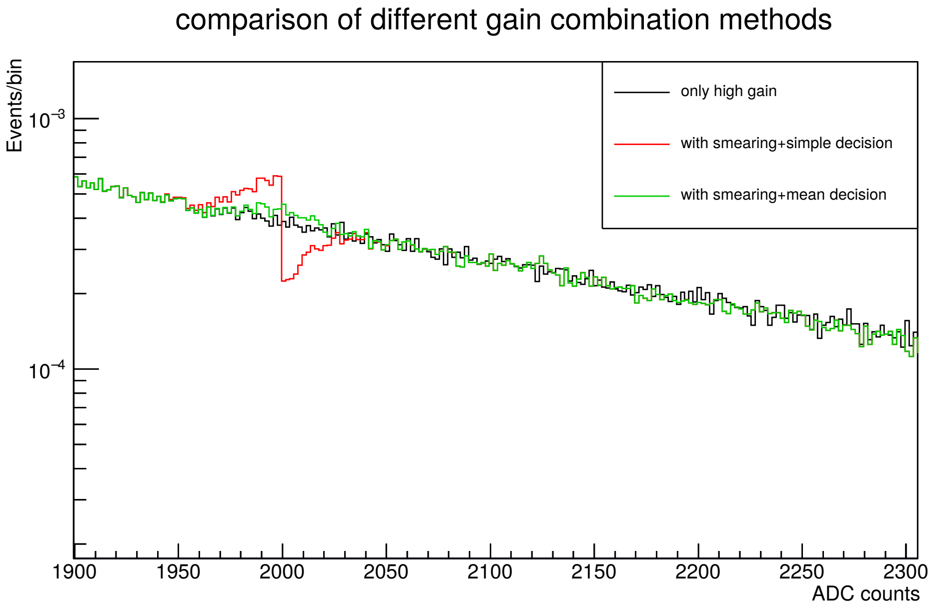


Figure 10.17.: Different methods to combine low-gain and high-gain data. Linear conversion function with parameters as in table 10.3.
Decision boundary at 2000 (high-gain) ADC counts to have more statistics at the decision boundary for demonstration purposes. Unconverted high-gain values (black) as comparison. The smeared converted values (red) shows the bipolar dip due to miss-decision using the high-gain value alone. Using the mean of high-gain and converted low-gain value (green) for the decision only produces a slight bump

Longer cables can cause two main problems. The first is general signal degradation due to attenuation, dispersion or a change in the interplay between the SiPM and the amplifier or the shunt resistor[13] for readout (see section 5.1.1) caused by a delay in the cable. The second possible problem is more external noise being picked up on the longer cables.

Due to the operation principle of the EASIROC with its internal bias voltage regulation and matching of the dynamic range to unamplified SiPM signals, as well as due to power consumption constraints, no external preamplifiers can be installed to avoid or at least reduce the problems.

To test the viability of different cable lengths, measurements of SiPM dark noise have been performed using cable lengths of 1 m, 2 m, 3 m, 4 m and 6 m, as well as with a 1.5 m cable into which a LEMO-LEMO coupling and a cable extension of 5 m was inserted. The same Hamamatsu S12571-050C SiPM[14] and the same EASIROC channel was used for all measurements. The temperature was not explicitly stabilized, but differed less than 1.5 °C between the different measurement series. The SiPM bias voltage was adjusted for the temperature changes, but a slight change in dark noise rates remains. The influence of the SiPM temperature T on the rates can be almost perfectly be compensated by scaling it with a factor of $2^{(25\,^\circ\mathrm{C}-T)/7\,^\circ\mathrm{C}}$, which corresponds to a doubling of the dark noise rate every 7 °C.

During all measurements with all cable lengths a strong sensitivity to noise produced either by the digital part of the EASIROC board (e.g. USB communication) or the voltage converters on the PSU board was observed. Due to this, it is important to place all signal cables as far away from these noise sources as possible or to install additional metal housings for the electronics boards to provide additional shielding.

Measurements of the trigger rate using the program described in section 9.3.3.2 were performed for 13.4 s for each threshold value between 1023 and 100. The results can be seen in figure 10.18, which shows only very little change between the different cable lengths. The measurement at 2 m shows an additional noise contribution, especially at thresholds corresponding to less than about 0.5 p.e. and also some slightly increased gain, resulting in a left-shift of the steps. The noise contribution can be explained by less than optimal cable routing in the setup for this cable length which was successively improved during the multi-day measurement campaign. The gain difference can not be completely explained, but might in some way be caused by the noise, might at least to some degree be caused by some temperature effect in the EASIROC[15], or might be a real effect[16] caused by the cable length. For the other cable lengths a small decrease of the gain with cable length is visible, while there is no apparent change of the form of the steps, especially the width or slope of the

[13] The shunt resistor is placed at the other end of the cable as seen by the SiPM.

[14] The same type as the AMD SiPMs but in a wired instead of in a SMD package

[15] Unfortunately the temperature sensor measuring the EASIROC temperature came loose during the measurements and therefore measured a too low temperature, which might also have influenced the SiPM bias voltage.

[16] some kind of resonance

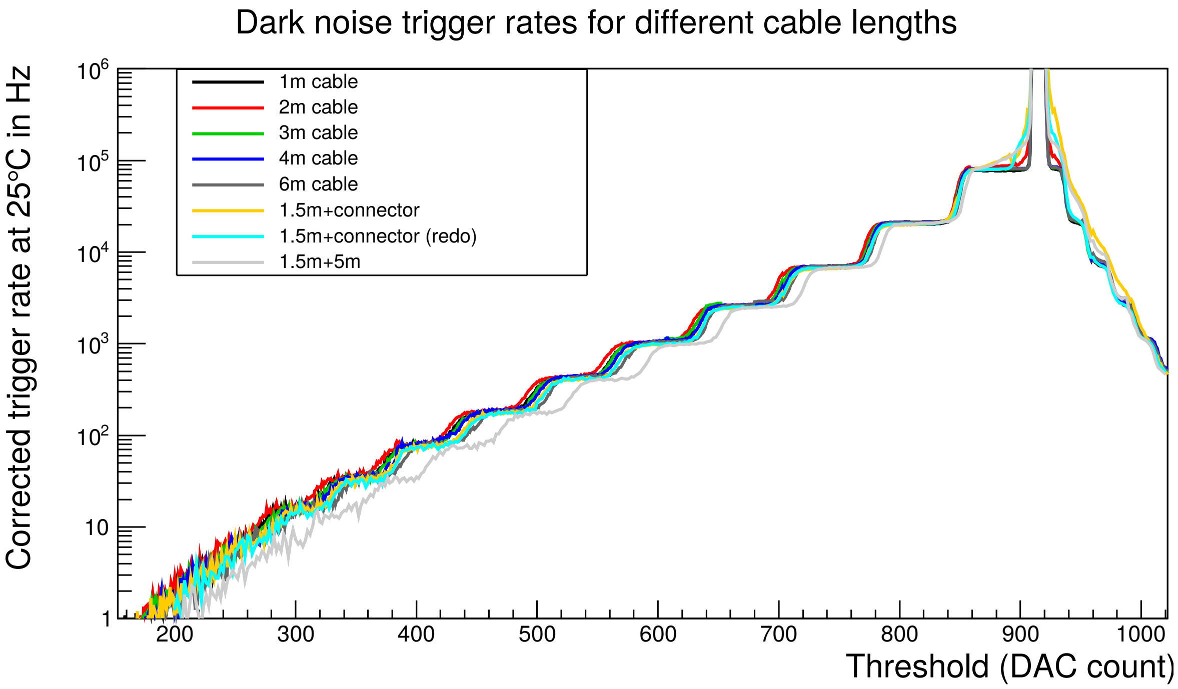


Figure 10.18.: Trigger rate scan of SiPM dark noise for different cable lengths. Rates corrected for temperature variations. S12571-050C SiPM connected with different cables to the same EASIROC channel. Lower DAC counts correspond to a higher threshold voltage. For thresholds above 915 the undershooting part of the pulses after the fast shaper causes the triggers.

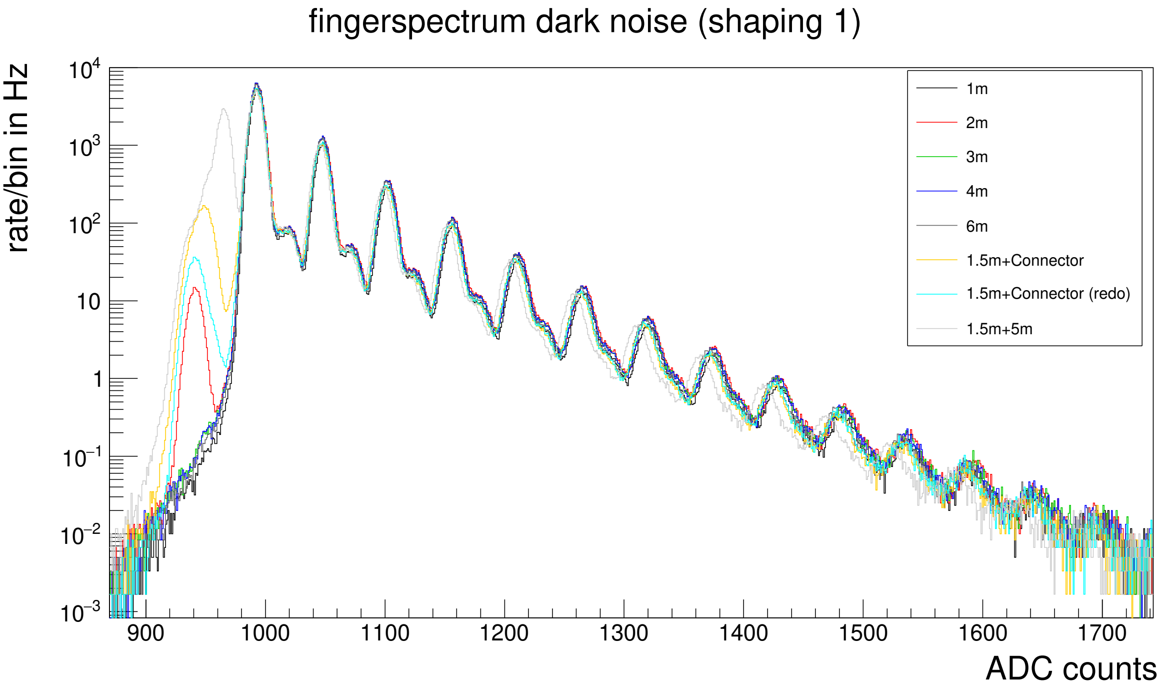


Figure 10.19.: Finger spectrum of SiPM dark noise for different cable lengths using a shaping time setting of 1 (25 ns nominal peaking time).

steps. The measurements employing the LEMO-LEMO coupling and especially the measurement with the 5 m cable extension show a decreased gain and a significant noise contribution at thresholds of 1 p.e. or lower. Provided proper cable routing and no great EM noise, cable lengths of up to 6 m and cable couplings do not seem to pose a problem, especially for the high thresholds (low DAC values) which are needed for an efficient suppression of SiPM dark noise.

As a second measurement, self-triggered finger spectra were taken using the slow shaper and the ADCs on the EASIROC board. With the `shelf` program described in section 9.3.3.1 and the *fastread mode* of ADC readout as described in section 9.2.12, an almost complete readout of the ~75 kHz dark noise was possible, which resulted in about 45M events during each of the 600 s measurement intervals. The trigger threshold was set to 880 for these measurements (about 0.5 p.e. as can be seen in figure 10.18) for all cable lengths. For triggering on the single activated channel, the multiplicity trigger path in the FPGA was used, but with a requirement of only one active channel with only the used channel 0 contributing. For each cable configuration, one spectrum using the lowest shaping time setting of 1 (25 ns nominal) with a hold delay setting of 11 was taken, as well as one spectrum using a shaping time setting of 5 (125 ns nominal) with a hold delay setting of 46[17].

The results of the measurements with the low shaping time are shown in figure 10.19. The highest peak at 995 ADC counts is the 1 p.e. peak, with peaks at higher ADC values corresponding to crosstalk events with higher multiplicity. The peaks at lower ADC counts visible in some measurements are caused by noise triggers without an

[17] The same as the settings used for most other measurements in this thesis

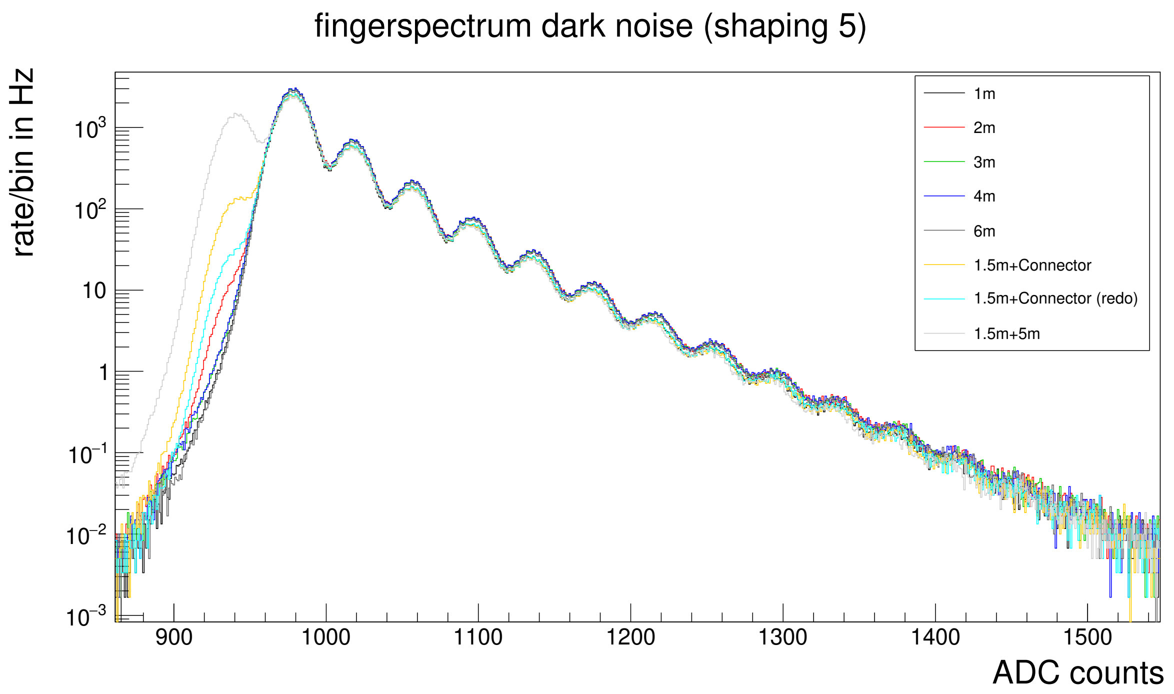


Figure 10.20.: Finger spectrum of SiPM dark noise for different cable lengths using a shaping time setting of 5 (125 ns nominal peaking time).

actual signal in the slow shaper. There is a slight 0 p.e. contribution for the 2 m cable length and the cables employing cable couplings which is also visible in the trigger rate scan. There is no general trend of a decrease in resolution, visible as a broadening of the fingers, or a decreased gain with increasing cable length. Only a small influence of the cable on the gain (position of the fingers) is visible, with a slight decrease of the gain for the measurement with the 5 m cable extension (like for the trigger rate measurements).

All spectra (even the best ones) show smaller peaks shifted to the right by about 0.25 p.e. next to each normal peak. These double peaks are caused by SiPM afterpulsing and also appear in simulations. The relatively discrete nature of the afterpulsing peak heights is likely caused by the adverse effects of higher amplitudes for longer recharge times canceling out with the attenuation due to the slow shaper[18] being stronger for later pulses.

The results of the measurements with the high shaping time are shown in figure 10.20. The highest peak at 980 ADC counts is the 1 p.e. peak, with peaks to the right corresponding to crosstalk events with higher multiplicity and eventual peaks to the left being caused by noise triggers. The differences between the different cable lengths are almost identical to those observed for the shorter shaping time.

Figure 10.21 illustrates the great influence the cable routing has onto the result, especially onto the noise performance. For the first measurements the routing had to

[18] The response of the shaper to the afterpulsing has not yet reached its maximum when the hold signal is asserted.

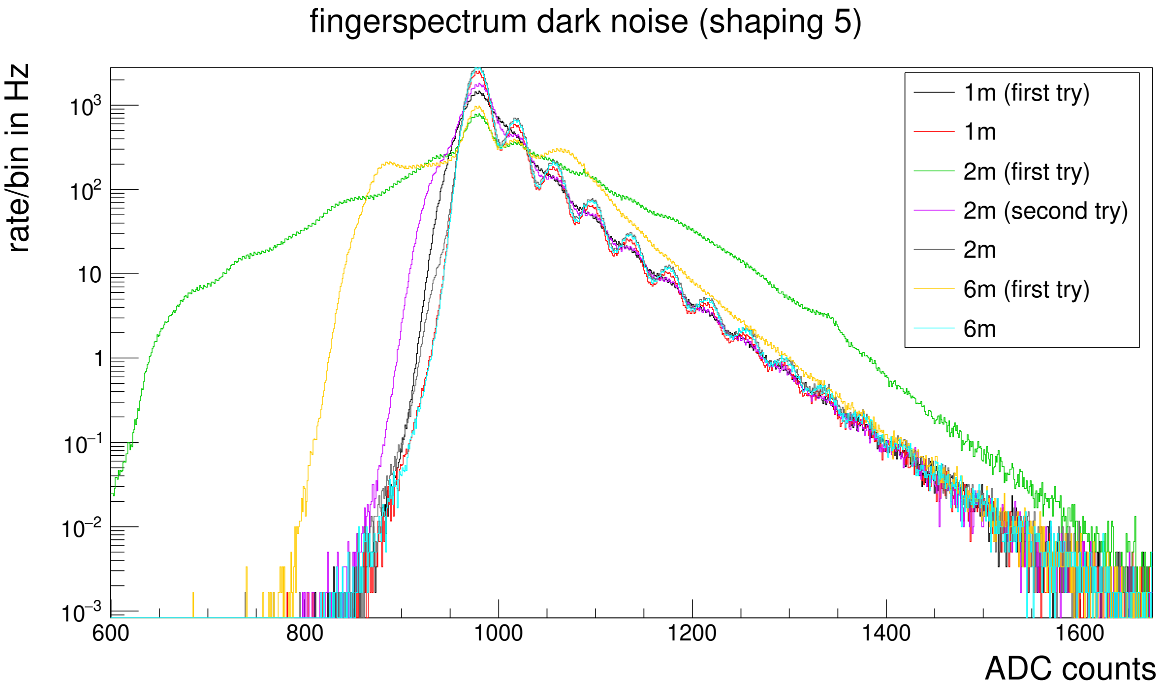


Figure 10.21.: Finger spectrum of SiPM dark noise for different cable lengths. Comparing multiple attempts of cable routing and noise shielding which either suffer from significant electronics pickup noise (first try) or have much reduced electronics noise.

be redone due to visible excessive noise which shows as a general decrease in resolution of the spectrum for the 1 m case or even as a completely distorted spectrum for the 2 m and 6 m cases. Even for the second measurement with 2 m cable length the routing was apparently non-optimal. Between the measurements also the power supply unit was moved to a better shielded position. For both 1 m measurements the trigger rate spectrum is almost identical[19], even though the finger spectrum differs significantly.

A proper operation at cable lengths of up to 4 m could be verified. Using longer cables is possible but might show decreased noise performance and resolution. In any case, care has to be taken to reduce electronic noise influences on the cable through cable routing away from noise sources and proper grounding of the cable shield.

10.2.6. SiPM dark noise spectrum

Besides the simple qualitative comparison of dark noise spectra in section 10.2.5, the dark noise spectra can also be quantitatively described. Any finger spectrum (not taking into account the extreme broadening through electronic noise seen in some of the previous spectra) can in principle be described as a sum of multiple shifted

[19] plot not shown here

normal distributions.

The modeling can be performed in two steps:

1. Calculate the distribution of number of cell breakthroughs (p.e.) due to crosstalk.
2. Conversion of SiPM p.e. to ADC counts taking into account the combined gain of SiPM and EASIROC and applying Gaussian smearing due to electronic noise and SiPM cell gain non-uniformity.

An additional modification of the finger spectrum can be caused by the used trigger, which might add some 0 p.e. contribution[20] caused by electronic noise in the trigger path and reduces the ≥ 1 p.e. peaks by the relative trigger efficiency. These modifications are currently not modeled. This model also does not include effects of afterpulsing (see section 5.2.4.3), which are only a small contribution ($\lesssim 1\%$) for current SiPM types and should lead only to a small broadening of the Gaussian smearing of the peaks at a shaping setting of 5.

The modeling of the distribution of SiPM crosstalk has been described by the FACT collaboration in [20, App A]. In this paper various distribution functions are fitted to the measured dark noise spectrum to phenomenologically describe the distribution. It has been shown that a distribution similar to an Erlang distribution with an additional modification parameter ν well describes the dark noise pulse heights of the SiPMs used for the FACT telescope. The probability $P(i)$ that a single breakthrough results in a total of i breakthroughs can be written as

$$P(i) = c \cdot \frac{(q \cdot i)^{i-1}}{[(i-1)!]^{\nu}} \quad \text{with} \quad q = p \cdot e^{-p} \quad . \tag{10.24}$$

c is a normalization parameter such that

$$\sum_{i=1}^{\infty} P(i) = 1 \tag{10.25}$$

p is a probability parameter but does not directly correspond to the crosstalk probability which is usually defined as the probability to cause any number of additional cell breakthroughs:

$$P_{\text{cross}} = 1 - P(1) \tag{10.26}$$

A second interesting quantity to characterize crosstalk is the mean number $\overline{N}$ of cell breakthroughs due to crosstalk:

$$\overline{N} = \sum_{i=1}^{\infty} i \cdot P(i) \tag{10.27}$$

The actually measured dark noise spectrum can then be described as a sum of normal distributions at positions

$$gain \cdot i + shift \tag{10.28}$$

[20] often called pedestal

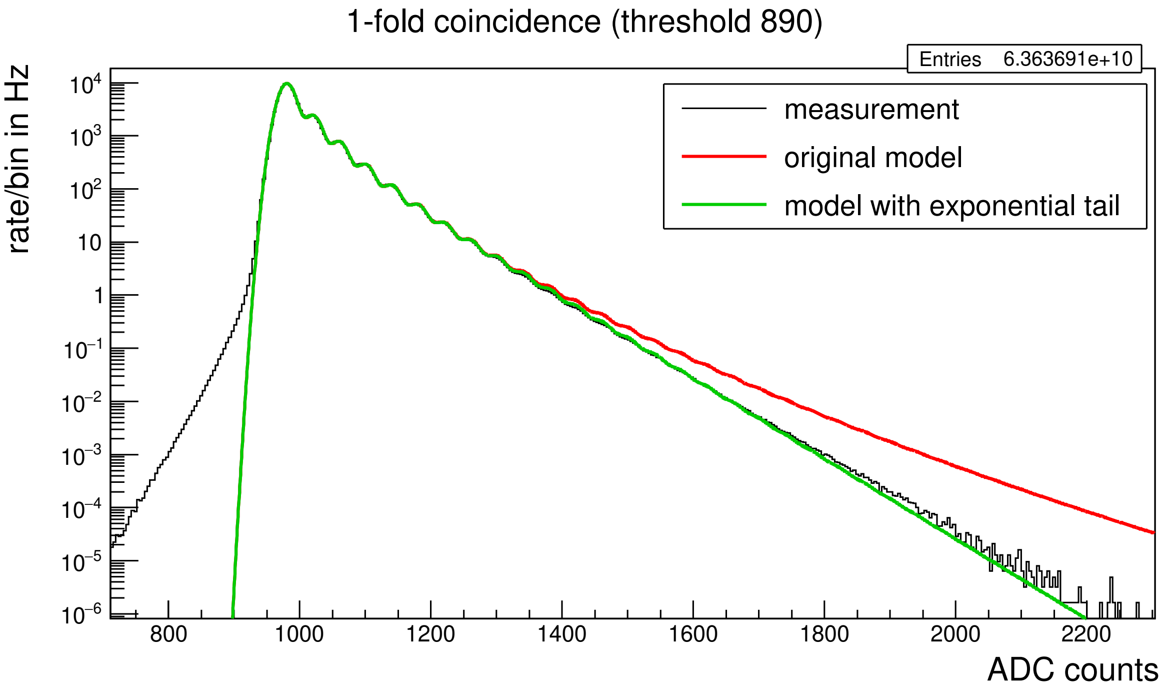


Figure 10.22.: Measurement of the dark noise spectrum of one AMD SiPM using the EASIROC at 125 ns nominal peaking time (shaping 5) and highest possible gain (feedback 1). The two fits of phenomenological models describe the distribution well up to around 9 p.e. The simple original model (Erlang-like distribution) fails for higher crosstalk peaks, while a modified model with an exponential distribution in the tail region can describe it well. The left tail of the distribution below 900 ADC counts is caused by the rare conditions where an event occurs in the undershooting part of the EASIROC slow shaper output of a previous untriggered pulse.

with widths

$$\sigma^2 = \sigma_e^2 + i \cdot \sigma_1^2 \tag{10.29}$$

and heights

$$N_{\text{dark}} \cdot P(i) \quad . \tag{10.30}$$

Hereby σ_1 is the relative fluctuation of the charge of one cell breakthrough and σ_e an additional readout noise. N_{dark} is the total number of events in the distribution.

During seven days the vast majority of dark noise events of one AMD SiPM was recorded with a trigger rate of around 100 kHz. The temperature was not explicitly stabilized, but differed by less than 2 °C during the measurement and the bias voltage was adjusted to compensate for any temperature change. The measurement is shown in figure 10.22.

The highest peak at around 980 ADC counts is the 1 p.e. peak. Crosstalk events up to 30 p.e. can be seen while up to around 13 p.e. individual peaks are visible.

The smaller tail at the left edge of the peak is not expected. As it is below the baseline, it has to be caused by an event being located in the undershooting part of the slow shaper output (see section 10.2.4) of a previous event, which is not triggered due to detector dead-time during readout. It has been confirmed in later measurements that the tail can be removed with a trigger holdoff, which ensures a minimum time distance from a preceding pulse.

Fits to the distribution were performed using the described model. The red fit is the model based on the Erlang-like distribution with ν parameter in equation 10.24, which does not describe the distribution well for crosstalk above 10 p.e.. For the green fit $P(i)$ was replaced by a simple exponential distribution for $i > 9$ (continuing smoothly). Even though the χ^2 value of the fit does not indicate a good fit due to the extremely small statistical uncertainty caused by the huge event count, the fits describe the measured distribution reasonably well for such a simple model.

Both fits result in a fitted probability parameter $p \approx 0.150$ and a ν parameter of 0.81 (red) and 0.80 (green), which converts to a crosstalk probability of $P_{\text{cross}} = 28.7\,\%$ and a mean breakthrough number of $\overline{N} = 1.46$. This is in agreement with the crosstalk probability of around 28 % given by Hamamatsu for S12571-050 SiPMs at their default operation point at 2.6 V overvoltage (see [42]). The fit result is clearly dominated by the first few p.e. peaks which have a low relative statistical Poisson uncertainty compared to the later peaks. To utilize the position of all peaks in the histogram, a gain value of 39.8 was chosen by hand to describe all peaks well. If left free, the fit chooses a gain value around 39.2 which does not properly describe the peak positions above 10 p.e. The SiPM crosstalk distribution determined this way can be used when analyzing the distribution (*MIP-peak*) of the signal caused by crossing muons in the scintillator tiles (see section 10.3.1).

10.3. Measurement of atmospheric muons with an AMD test configuration

The following characterization measurements were performed in the AMD shelf (see section 8.6.2) with the `shelf` program (section 9.3.3.1).

All measurements have been performed with the EASIROC board #36, the first SiPM carrier board and the PSU board version 2. The external SiPM supply voltage was monitored with a FLUKE 8845A multimeter. The SiPM bias voltages were adjusted with the DACs in the EASIROC, running on the external 4.5 V reference voltage, which is more stable than the internal 2.5 V reference voltage (see also section 7).

For triggering, the multiplicity trigger has been enabled to perform a four-fold coincidence (unless stated otherwise) (see also section 9.2.9).

The self-trigger of the AMD shelf itself imposes some challenges on the analysis. If a fixed subset of tiles is used as a trigger, all events that do not cause a trigger in these tiles are lost. This leads to a cut of the low signal events in the histograms of ADC values and prevents the determination of the trigger efficiency of these tiles

(100 % trigger efficiency for the recorded events by definition). The self-trigger also influences the normalization of all histograms. The number of events in each histogram bin is reduced by the product of all trigger efficiencies. The number of events in the bins above the cut value are, however, **not** changed by the trigger efficiency of the respective tile. To normalize all histograms to the same integral in the complete (uncut) histogram[21], each histogram of the trigger tiles has to be scaled by the trigger efficiency of the tile. Because this trigger probability is unknown and can not be deduced from the measurement, the normalization of the histogram of the trigger tiles can not be determined correctly. This especially prevents an ordering of different tiles by their photon yield using the tails of the distribution alone.

To circumvent this problem and to allow an investigation of all installed tiles, a multiplicity trigger n of m (m is usually the number of installed tiles) with $n < m$ can be enabled in the FPGA on the EASIROC board and an event selection can be performed offline during analysis. For the event selection, the trigger information of the ADC events that contains the information which tiles have surpassed the discriminator threshold around the time of the trigger is necessary. The events for the analysis of each tile are selected separately, with events not necessarily being usable for all tiles. This results in different event counts being available for each tile. Various offline cut criteria can be imposed, each explicitly not taking into account the information from the respective tile:

- n' other tiles included in the original trigger have triggered ($n' \geq n$ to ensure the original trigger decision did not depend on the respective tile).
- At least one tile above and one tile below the respective tile has triggered (muon has crossed the tile).
- SiPMs without a tile (if not all are connected) have not seen a signal (to suppress any common electronics noise).

With the first criterion it can be ensured that the event has been (or would have been) recorded regardless of the trigger decision of the tile under investigation. The selected event sample is therefore identical to the event sample that would have been recorded if the tile had not been included in the original multiplicity trigger in the FPGA. It is possible to set the cut number threshold n' to a value greater than the original FPGA trigger multiplicity n, to achieve a better purity of the muon triggers than the original trigger settings.

Especially when we are investigating the trigger probability of single tiles for crossing muons, the second criterion is important. If it is not imposed, a substantial amount of background events (muon misses the tile) are added, especially for the outer tiles. These background events may make the normalization of the histograms difficult. When this criterion is enforced, no events for the top-most and bottom-most tiles can be selected.

[21] The histogram is such scaled that the bins above the cut have the values they would have if the tile was not used as a trigger.

10.3.1. Photon yield of the AMD tiles

For the measurement of the photon yield of a set of AMD tiles, eight tiles have been put into the AMD shelf. The SiPM signal from nearly vertical muons crossing multiple tiles has been recorded with the EASIROC board using the `shelf` program. The measurement took about four million events during its run time of nearly six days. The trigger threshold has been set to 720 DAC counts, which corresponds to an effective threshold of 0.26 V above the baseline and to about 2.5 p.e.. The highest gain (feedback 1) and a nominal peaking time of 125 ns (shaping 5) has been configured in the EASIROC. A four-fold coincidence of all tiles has been used as a hardware trigger.

To remove the effect of self-triggering from the histograms, during analysis an event selection has been performed individually for each tile. A four-fold coincidence of the other tiles is required. To get less pedestal events where a muon did not cross a specific tile, one tile above and one tile below the respective tile have to have seen something. This last requirement obviously can never be fulfilled for the top-most and bottom-most tiles and is therefore not applied for these tiles.

The analysis has been performed on a combination of data from the high-gain and low-gain ADC values as shown in section 10.2.4.2. The measurement is shown in figure 10.23. As after event selection each histogram contains a different event count, each histogram is normalized to the same integral value.

As has already been introduced in chapter 4, the energy deposit of a particle crossing a thin scintillator detector can be approximated by a skewed Landau distribution. Assuming the light output is proportional to the energy deposit, the Landau distribution can be used as a starting point to model the distribution of signal measured for AMD tiles. All light losses occurring between photon production and detection in the scintillator, in the fibers and also including the PDE of the SiPM can be included by folding the original Landau distribution with a binomial distribution. The resulting distribution then represents the number of photons detected by the SiPM before the inclusion of any noise effects:

$$N_{\text{detected real photons}} \propto \text{Landau} * \text{binomial} \tag{10.31}$$

The dark noise adds an additional contribution in the 1 p.e. bin. The resulting sum of the distributions of real detected photons and dark noise (weighted with individual weights a and b) then has to be folded with the crosstalk distribution (derived in section 10.2.6). The resulting distribution describes the number of detected photon equivalents of the events. An additional pedestal contribution of pedestal events in which no SiPM cell breakdown has occurred can be added with weight c:

$$N_{\text{p.e.}} = (a \cdot \text{Landau} * \text{binomial} + b \cdot \text{darknoise}) * \text{crosstalk} + c \cdot \text{pedestal} \tag{10.32}$$

The original crosstalk distribution $P(n)$ derived in section 10.2.6 is defined for a single initial cell breakthrough. An iterative process handling each successive initial cell

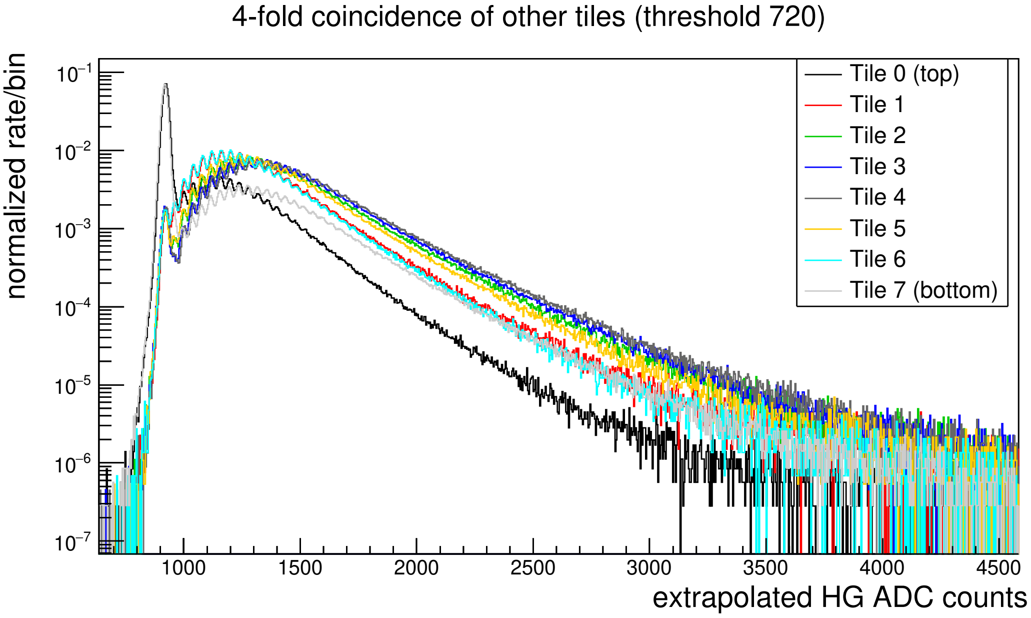


Figure 10.23.: MIP peak measured for vertical crossing muons in different AMD tiles. The highest gain (feedback 1) and a nominal peaking time of 125 ns (shaping 5) has been selected. For events to be included for a channel, four other tiles have to see a signal. One tile above and one tile below the respective tile have to see a signal for the central tiles (not top-most and bottom-most). The histogram is normalized to the same value for all channels, even though different event counts for different channels result from the event selection.

breakthrough individually can be applied to calculate the probability $P_n(m)$ to get m cell breakdowns after crosstalk from n initial (true) breakdowns:

$$P_n(m) = \sum_{i=1}^{m-n+1} P_{n-1}(m-i) \cdot P_1(i) \quad \text{with } P_1(m) = P(m) \quad \text{and } m \geq n \tag{10.33}$$

These crosstalk distributions can then be used in the numerical folding calculation.

This simple model does not include SiPM saturation effects, where cell breakdowns are prevented by a previous or simultaneous cell breakdown in the same cell, except where it is already included in the crosstalk distribution.

Identically to what has been performed in section 10.2.6 with the crosstalk distribution alone, the discrete distribution $N_{\text{p.e.}}$ of detected photon equivalents can be converted to the distribution of measured ADC values in the AMD. The individual $N_{\text{p.e.}}(i)$ can be used as weights for a sum of normal distributions at positions

$$\mathit{gain} \cdot i + \mathit{shift} \tag{10.34}$$

with widths

$$\sigma_i^2 = \sigma_{\text{e}}^2 + i \cdot \sigma_1^2 \quad . \tag{10.35}$$

Hereby σ_1 is the relative fluctuation of the charge of one cell breakthrough and σ_{e} an additional readout noise.

To perform a fit of the model to the measured data, the folding in equation 10.32 has been done numerically while discarding any normalization factors of the mathematical distributions themselves. All normalization factors are absorbed in the parameters a, b and c. Starting directly from the Landau distribution, for which the implementation in `root::TMath` [28] is used, the numerical calculation of $N_{\text{p.e.}}$ in the fit routine is performed for integer values of the photon count.

In the fit function there are multiple free parameters. The approximate peak position μ and scaling parameter σ of the approximate Landau routine as implemented in `root::TMath`, the *pde* (photon detection efficiency of the whole setup including light losses in the fibers and scintillator and the actual SiPM PDE), the scaling factors a, b and c, and the finger spectrum parameters *gain*, *shift*, σ_1 and σ_{e}. The parameters of the crosstalk distribution are fixed to the values determined in section 10.2.6.

The parameter μ is only the approximate peak position of the Landau distribution. From the parameters μ and σ of the Landau routine, the real peak position *mpv* can be derived as the position of the maximum. The uncertainty on the $\mathit{mpv}(\mu, \sigma)$ can be estimated by varying μ and σ by their respective statistical uncertainties.

The parameters μ and *pde* are strongly correlated. Over a broad range of values, results for a constant product of $\mathit{mpv} \cdot \mathit{pde}$ give very similar fit results. Due to the degeneracy, the fit is very unstable with best fit values of the μ parameter anywhere between 20 and 2000, varying with parameter start values. It is not expected that the *mpv* (or μ) differs much between tiles but that instead any differences in the light yield can be described by light losses, which are described by the *pde*. Therefore, the

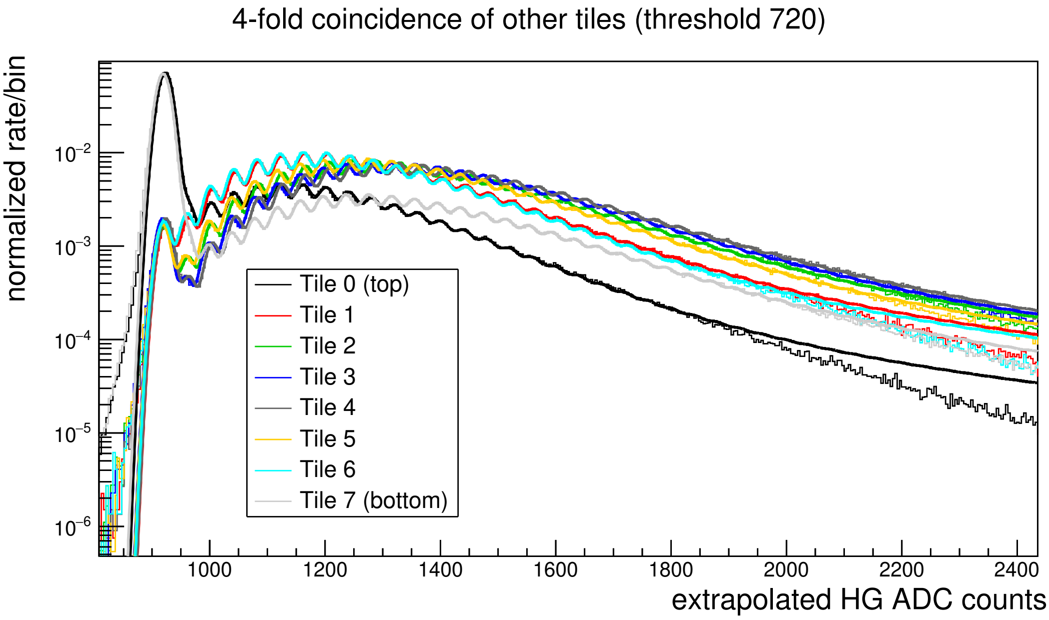


Figure 10.24.: MIP peak measured for vertical crossing muons in different AMD tiles. The distributions are fitted with the model in equation 10.32 in the range of 900 to 2000 ADC counts. The fit describes the data well in the important peak region but not in the tails, as can be expected from the simple model. Same data as in 10.23.

μ parameter was fixed to 500 for the final fits for all tiles. The σ parameter of the Landau distribution was left free.

The result of the fits together with the measurement data can be seen in figure 10.24. The fit was performed in the range of 900 to 2000 ADC counts, which includes peaks from the pedestal up to around 27 p.e.. Even though the χ^2 per degree of freedom is higher than one for all fits, a good agreement between model and data can be observed up to about 20 p.e.. The tails of the distributions are not described well. In the tails the fitted functions are significantly higher than the measured distributions. This is most likely caused by the used Landau distribution being only an approximation of the true energy deposit. The Landau distribution allows infinitely high energy deposit which is unphysical. As the goal of this measurement is the determination of the position of the maximum of the MIP peak, this deviation in the tails can be ignored here.

The position of the measured MIP peak maximum can be calculated from the fit values as

$$N_{\text{max,ct}} = mpv \cdot pde \cdot \overline{N} \quad , \tag{10.36}$$

where $\overline{N} = 1.46$ is the mean number of cell breakdowns resulting from one original breakdown due to crosstalk (see section 10.2.6). The parameters of the fit result and the calculated positions of the MIP peak including the statistical uncertainty from the fit are given in table 10.4. Systematic uncertainties on the MIP peak position arising from the arbitrary choice of μ are below 0.2 p.e. for all tiles. The uncertainties on

ch.	μ	σ	mpv	pde (%)	MIP	χ^2/NDF
0	500	46.9 ± 0.3	489.56 ± 0.06	0.886 ± 0.001	6.348 ± 0.006	11.5
1	500	51.3 ± 0.3	488.58 ± 0.06	1.015 ± 0.001	7.259 ± 0.005	2.1
2	500	50.1 ± 0.2	488.83 ± 0.05	1.268 ± 0.001	9.073 ± 0.005	2.6
3	500	47 ± 0.2	489.53 ± 0.05	1.408 ± 0.001	10.09 ± 0.005	3.7
4	500	48.6 ± 0.2	489.18 ± 0.05	1.408 ± 0.001	10.086 ± 0.005	3.9
5	500	46.6 ± 0.2	489.62 ± 0.05	1.251 ± 0.001	8.966 ± 0.005	2.8
6	500	48.7 ± 0.2	489.16 ± 0.05	0.992 ± 0.001	7.100 ± 0.005	2.1
7	500	52.1 ± 0.3	488.39 ± 0.06	1.281 ± 0.001	9.159 ± 0.006	4

Table 10.4.: Fit results and MIP peak position of some AMD tiles. The μ of the Landau distribution is fixed for the fit, σ and *pde* are left free. Different *mpv* values can be calculated due to different values for σ, even for identical μ. All uncertainties are statistical uncertainties from the fit.

the determination of the MIP peak position from a measurement can be neglected when they are compared to the difference between tiles or different optical fibers.

The measured light output of the scintillator tiles differs strongly between individual tiles. The biggest part of the differences originates from a variances in the quality of the wrapping of the scintillator tiles and smaller differences might also arise from differing quality of the polished fiber ends. A significant degradation ($\lesssim$ 20 %) of the quality of the clean fibers that were repeatedly inserted to and removed from the fiber couplings could be observed. Between different measurement series where tiles and fibers were removed and reinstalled, differences in the light yield in the order of 10 % could be observed.

The position of the MIP peak is only at about 10 p.e. for the best tiles, which is much less than the 40 p.e. previously simulated [61]. The chosen method to apply a reflective end in the tile has been shown to be ineffective [52]. The biggest contribution to the light loss, which simulations do not predict properly, originates from the fiber coupling between the wavelength shifting fiber in the tile and the clear optical fiber guiding the light onto the SiPM. Together with other improvements in the simulations, the measurements can be brought into agreement with the simulations taking these points into account [65].

10.3.2. Trigger efficiency for single muons

To determine the trigger efficiency of an AMD tile for a crossing muon, muon events have to be triggered using other tiles (or even an external trigger), while recording whether the tile under investigation has itself caused a trigger. It is particularly important to ensure that the muon really crosses the tile, which can be ensured with the requirement of at least one tile above and below the tile under investigation having triggered.

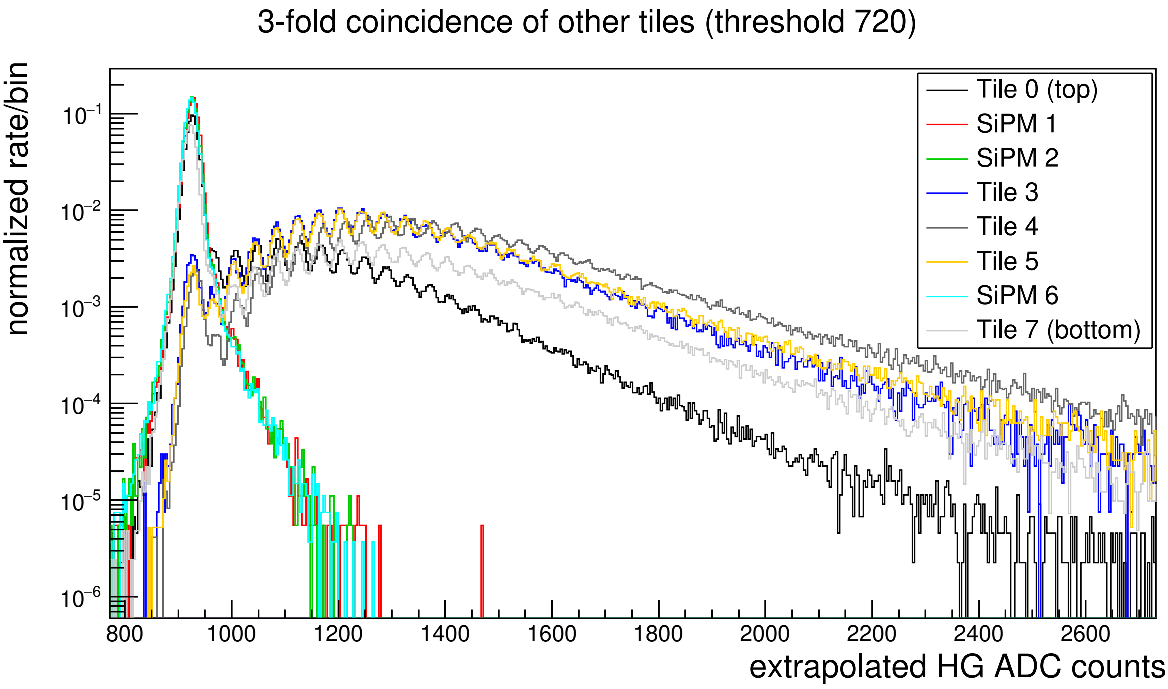


Figure 10.25.: Measured MIP pulse height spectrum for the tile configuration of the trigger efficiency measurement. Data recorded at a trigger threshold of 720 DAC counts (≈2.5 p.e.; 265 mV). A three-fold coincidence of the other tiles and at least one hit in a tile above and one in a tile below the respective tile (not for top and bottom) is required for event selection.

The trigger efficiency measurements had to be performed utilizing five tiles only, because the other tiles were in use as trigger tiles for SSD measurements (see section 8.6.3). This means that with the top-most and bottom-most tiles not being available for analysis, only three tiles have meaningful data. Tiles are connected to channels 0, 3, 4, 5 and 7 on the EASIROC board and are placed in the AMD shelf (section 8.6.2) in this order. These channels are included in the multiplicity trigger mask, as well as the not-connected (bare SiPM) channel 2, to check whether inclusion in the trigger has any effect for a not-connected channel (channels 1 and 6 serve as comparison). All tiles and SiPMs are numbered according to their respective EASIROC channel number.

A distribution of the signal output in ADC counts is presented in figure 10.25. Comparing the light output of the tiles in use, tile 4 has an above average light output (MIP peak at 10.1 p.e.), tiles 3, 5 and 7 have a comparable and average output (8.3 p.e., 8.5 p.e., 8.5 p.e.) and tile 0 is much below average (5.8 p.e.). The MIP peak position was determined as in section 10.3.1.

Measurements were performed during three hours for each threshold setting, starting from a threshold of 900 DAC counts (≡ 30 mV), down to 140 DAC counts (≡ 1.03 V) in steps of 20 DAC counts. A second run from 890 down to 150 in steps of 20 DAC counts was performed to result in a combined step size of 10 DAC counts. A trigger-

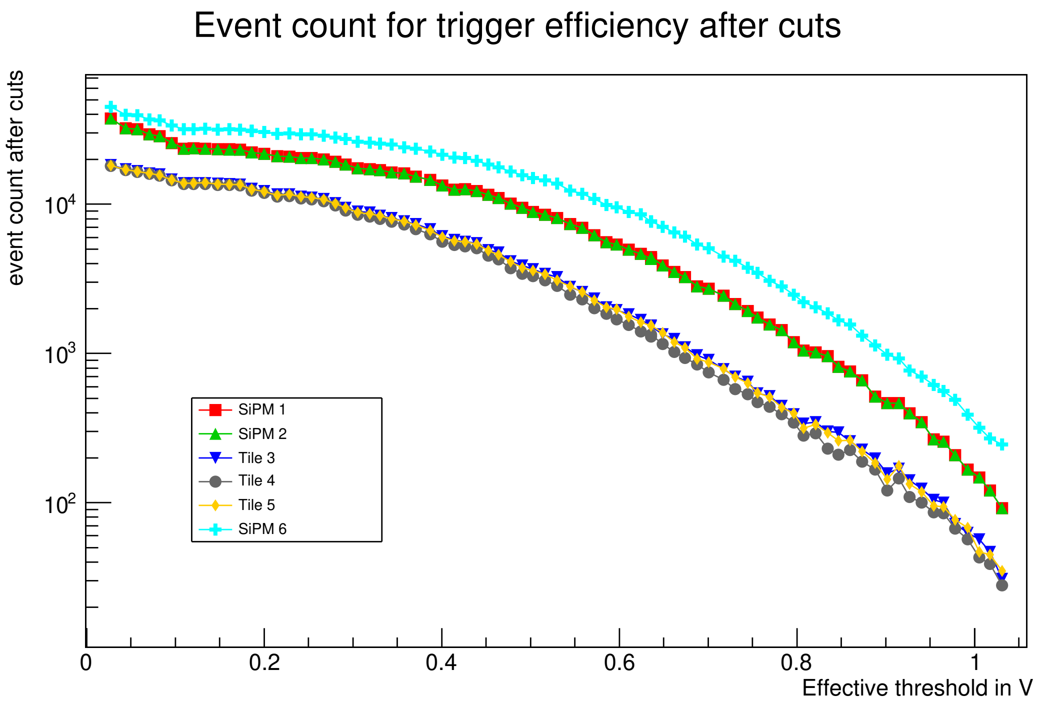


Figure 10.26.: Event counts for the different channels after a four-fold coincidence of the other tiles is used for event selection. The channels without tiles have a higher event count due to one additional tile available which can fulfill the cut condition. The difference between SiPM 6 and SiPMs 1+2 is caused by a much lower light output of tile 0 (has to be active for the events for channel 1+2) compared to tile 7 (has to be active for events for channel 6).

multiplicity setting of "3"[22] was used for thresholds between 900 and 780 (≡ 185 mV), a setting of "2" between 780 and 440 (≡ 635 mV) and of "1" between 440 and 140, with the borders being done twice to study the influence of this trigger setting. A high-gain preamplifier setting of "1" (maximum gain) has been selected. The measurements were performed at room temperature, varying between 27 °C and 28 °C for the SiPMs. All effective threshold voltages given in this section have been calculated from the DAC values using the characterization measurement presented in section 10.2.3 and are given relative to the respective baseline as determined from the DAC characterization measurement. The event counts available for each channel after event selection requiring a four-fold coincidence of the other tiles are shown in figure 10.26.

Due to the "one tile above and one tile below" selection criterion for events, the top-most and bottom-most tiles can not be included in the analysis.

For each event sample of a given channel and threshold, the number of all events N and the number k of events in which the triggerinfo bit for the respective channel is active (channel discriminator was active at a time close to the trigger from the other

[22] more than 3 tiles = at least 4 tiles

tiles) is counted. The best estimate of the trigger efficiency ε is then

$$\hat{\varepsilon} = \frac{k}{N} \quad .$$

The number k of events can be assumed to follow a binomial distribution, with the expectation value $\langle k \rangle = \varepsilon N$ and the standard deviation $\sigma_k = \sqrt{\varepsilon(1-\varepsilon)N}$. With the usual error propagation it can then be estimated that

$$\sigma_\varepsilon = \frac{\sigma_k}{N} = \sqrt{\varepsilon(1-\varepsilon)/N} \quad .$$

Substituting the observed efficiency $\hat{\varepsilon}$ for the real efficiency ε is an often performed approximation, which yields sufficient results for moderately large numbers of N while $k \neq 0$ and $k \neq N$. To get an improved upper-bound estimate of the uncertainty (instead of $\sigma_\varepsilon = 0$) for $k = 0$ and $k = N$, in these cases $k = 1$ or $k = N - 1$ can be used for the uncertainty estimations.

A Bayesian approach to the problem could yield another estimate for the uncertainty of ε. These solutions are computationally difficult and involve various (arbitrary) choices of prior and how to define the uncertainty intervals. Due to the fact that the observed trigger efficiencies for the measurements presented here are influenced to a non-quantifiable degree by other effects, the simple estimate of the statistical uncertainty as presented above is deemed sufficient for this application. Systematic uncertainties arise from the event selection, including the distribution of zenith angles and energies of the muon sample, uncorrected temperature effects and the EM noise level in the laboratory.

The trigger efficiencies of the different tiles in dependence of the trigger threshold are plotted in figure 10.27. The results derived in section 10.2.3 are used to convert the threshold DAC values to an effective threshold voltage. A multiplicity condition of $n' = 4$ (the maximum possible with the setup) is applied for the event selection for all thresholds. No significant difference between the measurements performed twice with originally different FPGA trigger multiplicity thresholds can be seen if identical stronger cuts are applied during analysis.

For higher thresholds the trigger efficiency first decreases as expected, but then increases again significantly for the highest thresholds. As it is not physically meaningful that the real trigger efficiency for crossing muons increases with increasing threshold, this has to be caused by an event selection effect. Events which produce an above-average energy deposit in all tiles have an above-average overall trigger efficiency, which causes them to be overrepresented in the selected event sample. The overall trigger efficiency is proportional to the product of all individual trigger efficiencies, which at the highest thresholds are low for all tiles due to the requirement to set the same threshold for all tiles. Therefore, at the highest trigger thresholds the event rate is very low. Events with above-average energy deposit might be caused by non-minimum-ionizing muons with very low or high energy or multiple simultaneous muons from the same air shower.

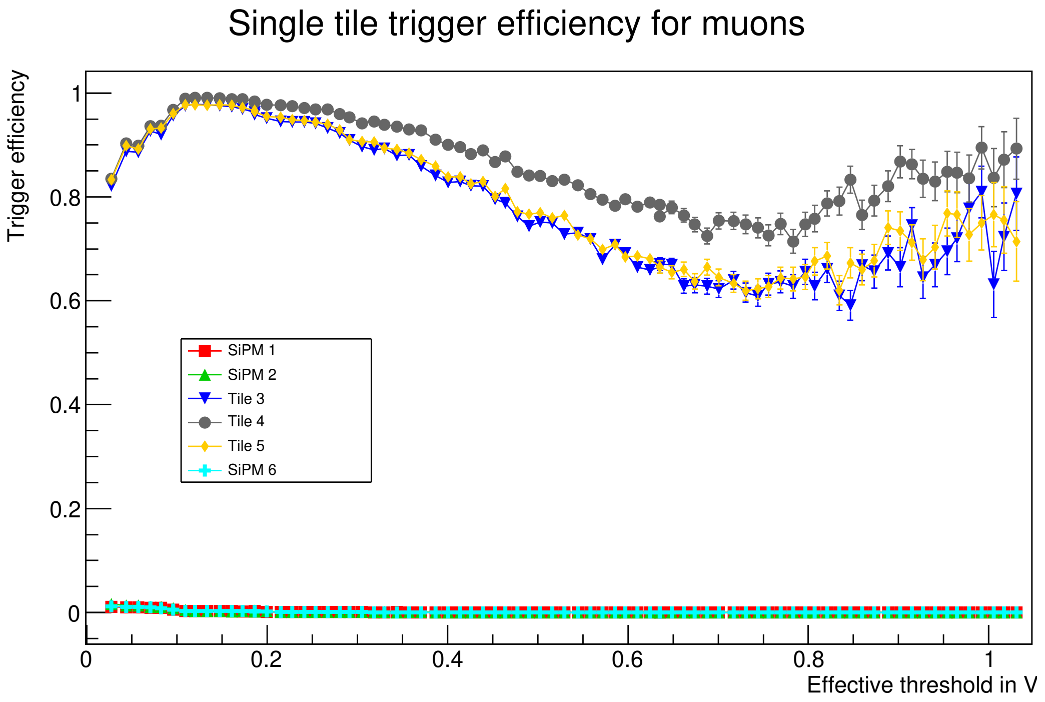


Figure 10.27.: Measured trigger efficiency of some AMD tiles. A four-fold coincidence of the other tiles has been used for event selection. The drop at low thresholds can be explained by random noise triggers. The increase at high thresholds is an indication of an enrichment of events with an increased energy deposit in all tiles in the event samples at the highest thresholds (in the EASIROC a single threshold is used for all channels). The total event counts after cuts leading to the efficiency calculation is shown in figure 10.26.

One possibility to reduce the effect of a changing muon sample is to relax the multiplicity selection threshold n' for higher trigger thresholds. The lower multiplicity condition during analysis is possible due to the low multiplicity value used for the hardware trigger. Due to a lower dark noise rate at higher thresholds, no significant contribution of dark noise triggers due to random coincidences arises even at lower multiplicity values. The resulting single tile trigger efficiencies should be independent of the exact trigger condition of the other tiles. If the multiplicity requirement is relaxed for a too low threshold, noise triggers are included (reducing the apparent efficiency). If the multiplicity is changed for a too high threshold, the change of the composition of the event sample, which is to be prevented, has already occurred. Also in this case a change of the apparent trigger efficiency to a lower value will be observed. It, however, has to be noted that through a change of the multiplicity requirement also the distribution of muon zenith angles is changed slightly[23], which also influences the muon trigger efficiency. For the final analysis, the multiplicity requirement was set to $n' = 3$ for DAC values below 520 DAC counts (530 mV) and to $n' = 2$ for DAC values below 370 DAC counts (730 mV). As required, no changes of the trigger efficiencies at the point of change of the multiplicity condition is visible. The event counts available for each channel after the new event selection are shown in figure 10.28. The results are shown in figure 10.29 for effective threshold voltages calculated from the DAC values.

For low threshold values, a significant fraction (≈ 17 % for tile 4) of events is caused by random noise triggers, which lead to a reduction of the apparent trigger efficiency. The steps visible in the measurements for these lowest thresholds can be explained by a slight difference in noise level for the two interleaved measurement series due to temperature differences. At a trigger threshold of about 135 mV (DAC value 820; ≈1.5 p.e.) a maximum of the trigger efficiency is reached. The maximum trigger efficiency (including statistical uncertainty as introduced above) is 0.991 ± 0.001 for tile 4 and 0.977 ± 0.001 for tiles 3 and 5. The systematic uncertainty due to dark noise triggers can be estimated to be about 0.005. For tiles with significantly different light outputs, the differences are expected to be below 0.05.

At a realistic threshold (see below) of about 925 mV (DAC 220), the measured trigger efficiencies for a vertical crossing muon in a single tile (including statistical uncertainty) are 0.374 ± 0.010 for tile 3, 0.549 ± 0.005 for tile 4 and 0.396 ± 0.006 for tile 5. The systematic uncertainty due to the used trigger condition for the other tiles can be assumed to be less than 0.05. For tiles with significantly different light outputs, at these thresholds drastic trigger efficiency differences of up to 0.3 arise.

The trigger efficiencies for tile 4 being slightly higher than for tiles 3 and 5 is in concordance with the observed pulse height spectrum of the tiles, shown in figure 10.25.

In figure 10.30 a zoom on the SiPM channels without tiles is provided. SiPM 2, which is included in the primary trigger generation, has a slightly higher chance of a random coincidence of 0.0163 ± 0.0007 than the other SiPMs (0.0110 ± 0.0005

[23] The muons do not necessarily have to cross the outer-most tiles any longer.

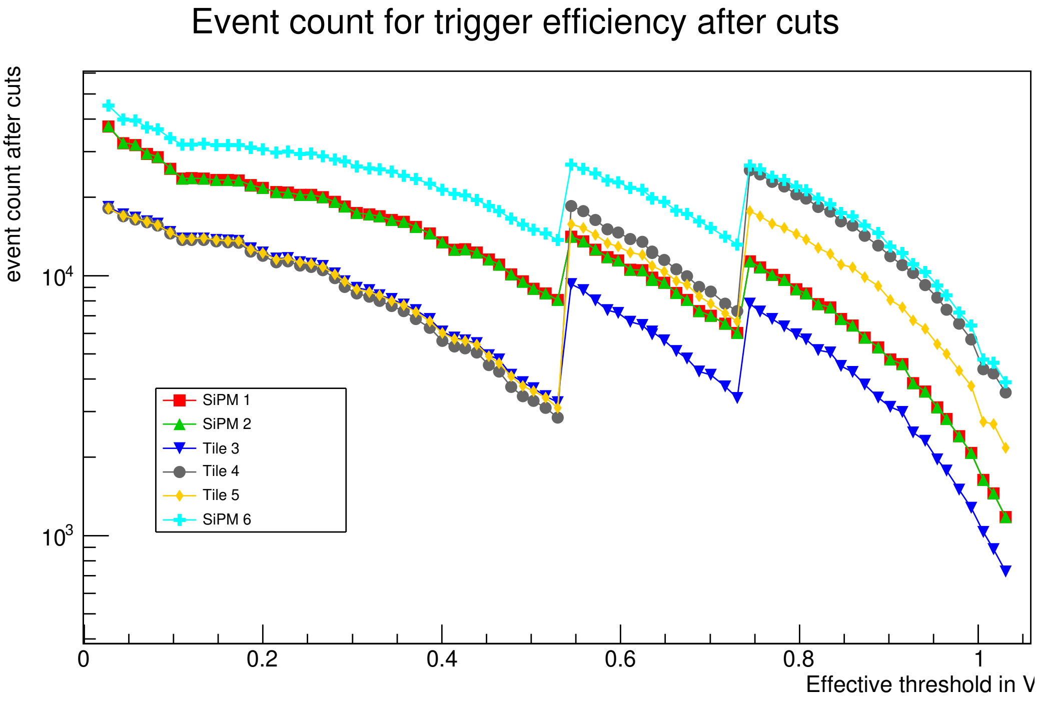


Figure 10.28.: Event counts for the different channels after event selection. A four-fold coincidence of the other tiles has been used for event selection for low threshold values (<530 mV), a three-fold coincidence for medium thresholds (<730 mV) and a two-fold coincidence for high thresholds. The thresholds at which the multiplicity condition is relaxed is clearly visible as a rise in the event count.

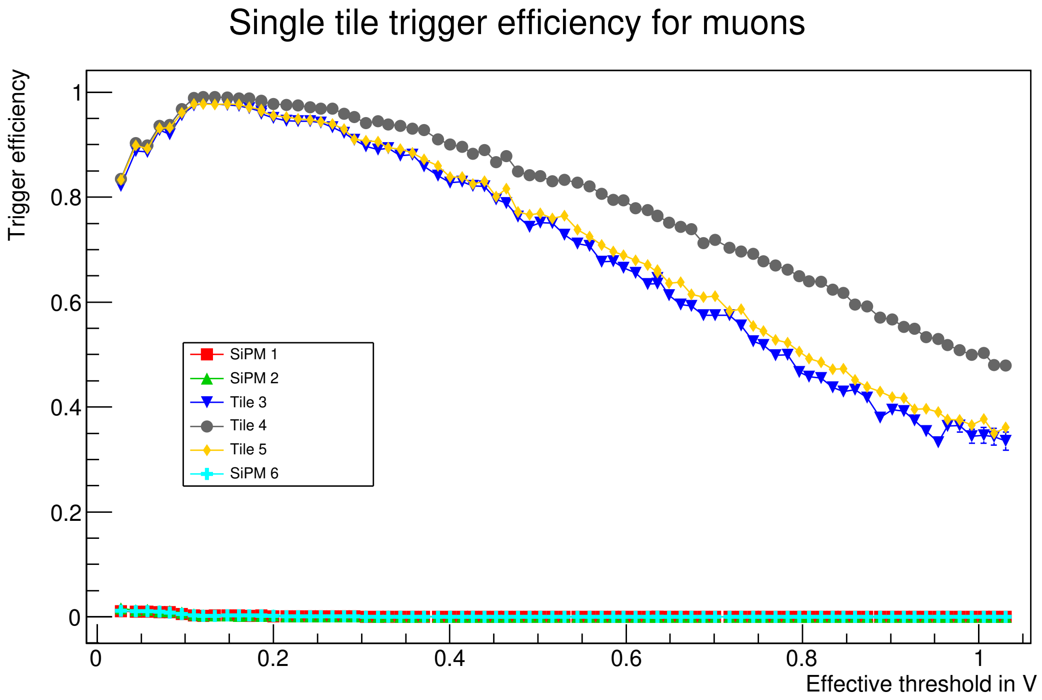


Figure 10.29.: Measured trigger efficiency of some AMD tiles. The drop at low thresholds can be explained by random noise triggers. A four-fold coincidence of the other tiles has been used for event selection for low threshold values (<530 mV), a three-fold coincidence for medium thresholds (<730 mV) and a two-fold coincidence for high thresholds. The total event counts after cuts leading to the efficiency calculation is shown in figure 10.28.

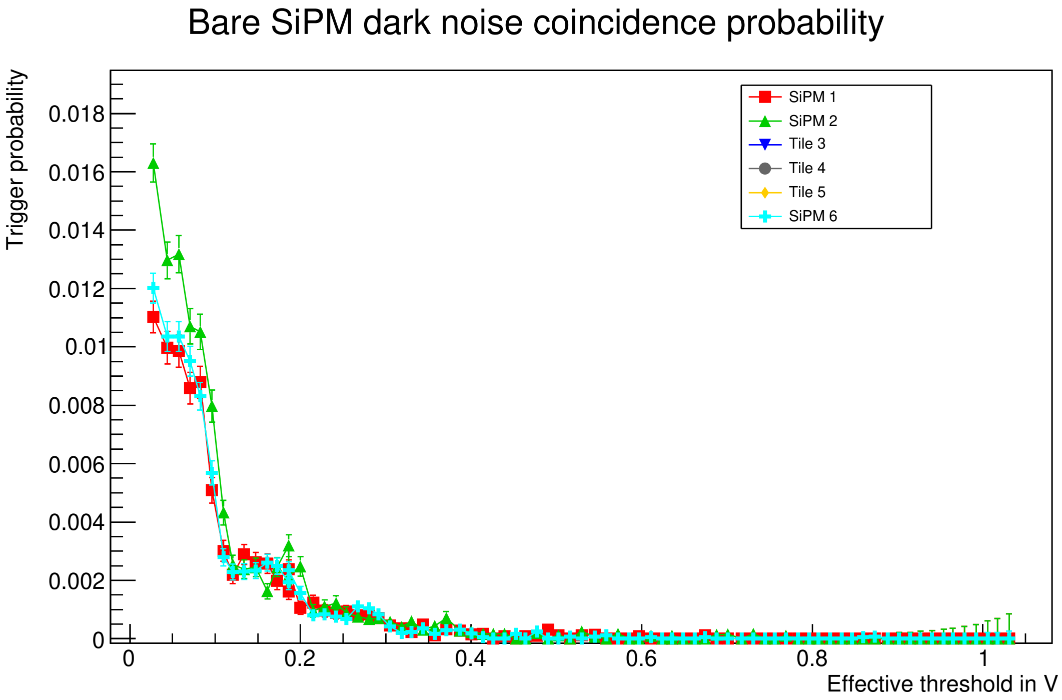


Figure 10.30.: Measured apparent trigger efficiency (chance of random coincidences due to dark noise) of the SiPM channels without connected tiles. Zoom of figure 10.29.

and 0.0120 ± 0.0005) at the lowest measured threshold of 30 mV (DAC value 900). But at the threshold of 135 mV (DAC value of 820), at which fake triggers due to random coincidences do no longer play an important role for the tiles, the difference has disappeared and all random coincidence chances are at 0.0025 ± 0.0003. This value is consistent with a 2 p.e. dark noise rate of 35 kHz (see figure 10.31) and a coincidence window of 70 ns (50 ns overlap for the triggerinfo generation in the FPGA and 20 ns discriminator activity). For thresholds above 425 mV (DAC value 600) random coincidences are completely negligible.

As a comparison, the single-SiPM trigger rates for all channels are shown in figure 10.31. For the highest thresholds above about 900 mV (DAC counts below 240), the connected SiPMs show a higher trigger rate than the unconnected SiPMs. At a trigger threshold of 925 mV (DAC 220) for the channels connected to tiles and of 890 mV (DAC 250) for bare SiPMs, the single channel noise trigger rates falls below 20 Hz (dark noise+muons). This means that at this point the dark noise rate is about the same as the expected random muon hit rate of around 10 Hz, which is the desired noise rate to achieve with the selected threshold during normal operation.

It is visible, that the SiPM 2 (green) shows a slightly increased gain (about 1 %), while the SiPM 7 shows a slightly decreased gain (about 3 %) and also shows a slightly decreased crosstalk probability (deduced from the ratio of the 1 p.e. rate to the 7 p.e. rate) compared to the other channels. This can be corrected for by adjusting the SiPM bias voltages by −20 mV and 40 mV, respectively. For all measurements in this

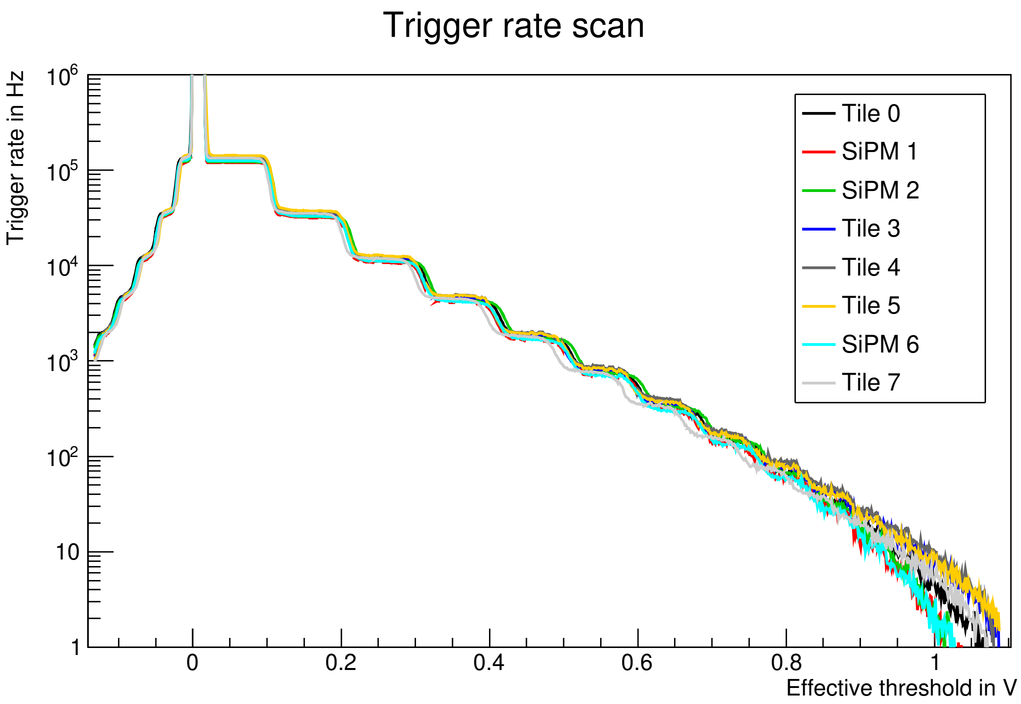


Figure 10.31.: Scan of the single-channel trigger rates for the trigger efficiency setup with a high-gain preamplifier setting of “1”.

thesis, however, the bias voltages[24] given by Hamamatsu for the individual SiPMs are used.

It becomes apparent that the achieved muon trigger efficiency at an acceptable single channel noise rate is far below the design goal of 99 %. To achieve this goal, the light output of the tiles has to be significantly increased[25]. The usage of low-crosstalk SiPMs, which lower the threshold at which acceptable noise rates are achieved, can also be advantageous and is likely necessary for this goal.

Only slight hints of a step-like development of the trigger efficiency, which corresponds to the different p.e.-steps, can be seen in figure 10.29 for thresholds of about 200 mV (DAC 770; 2 p.e.), 300 mV (DAC 690; 3 p.e.) and 400 mV (DAC 620; 4 p.e.). This can be understood by the fact that the time span of photon arrival on the SiPMs is not instantaneous in contrast to dark noise with crosstalk, which is shown in figure 10.31. The application of the fast shaper for triggering with a 15 ns nominal peaking time leads to a smearing of triggered pulse heights for non-instantaneous photon signals. Due to this, application of a trigger threshold at a precise p.e. value is not possible and also not sensible to try for normal AMD operation. The steps can however be seen for the unconnected SiPMs (figure 10.30), which experience dark noise only.

[24] adjusted for temperature

[25] Better reflective fiber end and/or less losses at the fiber coupling. See [65].

11. Summary and Outlook

The Aachen Muon Detector (AMD) has been presented as one proposed upgrade of the Pierre Auger Observatory to improve the measurement of the muonic air shower component to allow in particular for a better determination of the mass of the primary cosmic ray particles. The hardware and firmware of the data acquisition (DAQ) system has been described in detail in this thesis.

The DAQ system for the Aachen Muon Detector has been built and commissioned successfully. The FPGA firmware and the measurement software perform as desired during various measurements.

The important parts of the EASIROC and the Power Supply Unit (PSU) have been characterized. The EASIROC performance has been found to be sufficient for the envisioned application. The PSU performs within expectations. The SiPM bias voltage can be kept at the desired value using the PSU and the input DACs in the EASIROC with a deviation of less than 50 mV, which leads to a gain stability of better than 2 %.

Some techniques for data analysis or simulation of EASIROC-based detectors have been developed and presented. The cross calibration between high-gain and low-gain channels allows an increased dynamic range for measurements with the slow shapers and ADCs. A simulation model of the EASIROC slow shapers has been developed. A model describing the ADC measurements of the slow shaper output for SiPM dark noise and single muons in a scintillator tile has been used to determine SiPM properties and the light output of the scintillator tiles.

First measurements with this system, including measurements of the light output and the trigger probability of the AMD scintillator tiles have been performed. The light output of the scintillator tiles has been found to be only around 10 p.e. for the best tiles, which is much smaller than previously simulated. The biggest losses are likely to occur at the coupling of the optical fibers. The light output of the tiles differs by more than 40 % due to variances during tile production and hand-made wrapping. Due to this low light output, the design goal of 99 % trigger efficiency for crossing muons in a tile can not be reached. The trigger efficiency is less than 50 % for most tiles.

To achieve the design goal of 99 % single tile trigger efficiency for muons, the original AMD detector concept has to be changed. With lower crosstalk SiPMs the trigger threshold could be reduced, leading to a higher trigger efficiency, but still below 99 %. The long clear optical fibers between the tiles and the SiPMs, which are the main cause for the low light output of the tiles, have been introduced to avoid long signal cables to the EASIROC. In this thesis it has however been shown that while leading to a slightly increased noise level, long signal cables of up to 6 m between the SiPMs and the EASIROC are viable. An improved AMD detector with SiPMs located directly at the ends of the wavelength shifting fibers in the tiles and long signal cables instead of optical fibers to bridge the gap to the central EASIROC board can likely fulfill the requirements.

11.1. Outlook

The original concept of the Aachen Muon Detector will not be pursued further. However, a prototype of the modified MiniAMD detector concept is currently under test. The removal of the clear optical fiber from the design shows much reduced light losses and therefore a higher light output. The reduced crosstalk in the newer SiPM type reduces the dark noise trigger rate to negligible levels even at thresholds of about 3.5 p.e.. Both changes together lead to a much increased trigger efficiency for muons.

The data acquisition described and characterized in this thesis will be used for the measurement of the muons of air showers with the MiniAMD detector. Measurements together with the small air shower detector array used for lab courses at the RWTH Aachen University will be performed. A detailed study of the spatial resolution and homogeneity of the MiniAMD modules using a calibration detector in Karlsruhe, which can provide exact position coordinates of crossing muons, is also foreseen. It is planned to employ two MiniAMD modules as trigger detectors in the quality control process of SSD modules built at the RWTH Aachen University for the AugerPrime upgrade of the Pierre Auger Observatory [4].

The work on MiniAMD will be published in [65].

Bibliography

[1] boost–c++ libraries. `http://www.boost.org/`, visited 2018-02-22.

[2] MinGW–Minimalist GNU for Windows. `http://www.mingw.org/`, visited 2018-02-22.

[3] A. Aab et al. The Pierre Auger Cosmic Ray Observatory. *Nucl. Instrum. Meth.*, A798:172–213, 2015. arXiv:1502.01323 [astro-ph.IM].

[4] Alexander Aab et al. The Pierre Auger Observatory Upgrade - Preliminary Design Report. 2015. arXiv:1604.03637 [astro-ph.IM].

[5] J. Abraham et al. The Fluorescence Detector of the Pierre Auger Observatory. *Nucl.Instrum.Meth.*, A620:227–251, August 2010. arXiv:0907.4282v1 [astro-ph.IM].

[6] Advanced Energy. *XS Series Extra-small High Voltage Biasing Supply – Datasheet.* `www.advanced-energy.com/uv_docs/XSSeries.pdf`, visited 2018-02-22.

[7] S. Agostinelli et al. Geant4 – a simulation toolkit. *Nuclear Instruments and Methods in Physics Research A*, 506(3):250 – 303, 2003.

[8] O. C. Allkofer. *Introduction to Cosmic Radiation.* Verlag Carl Thiemig, 1975. ISBN 3-521-06098-5.

[9] ALTERA. *Cyclone Device Handbook, Volume 1*, May 2008. `https://www.altera.com/content/dam/altera-www/global/en_US/pdfs/literature/hb/cyc/cyc_c5v1.pdf`, visited 2018-02-22.

[10] ALTERA. *Quartus II Handbook Version 11.0*, May 2011. `https://www.altera.com/content/dam/altera-www/global/en_US/pdfs/literature/hb/qts/archives/quartusii_handbook_archive_11.0.pdf`, visited 2018-02-22.

[11] ALTERA. *Cyclone V Device Handbook, Volume 1*, December 2017. `https://www.altera.com/en_US/pdfs/literature/hb/cyclone-v/cv_5v2.pdf`, visited 2018-02-22.

[12] Analog Devices. *AD7685, 16-Bit, 250 kSPS ADC – Data Sheet.* `http://www.analog.com/media/en/technical-documentation/data-sheets/AD7685.pdf` , visited 2018-02-22.

[13] Analog Devices. *AD9220 Complete 12-Bit, 10.0 MSPS Monolithic A/D Converter – Data Sheet*, 2003. `http://www.analog.com/media/en/technical-documentation/data-sheets/AD9221_9223_9220.pdf`, visited 2018-02-22.

[14] Analog Devices. *REF19x Series Precision Micropower, Low Dropout Voltage References – Data Sheet*, 2011. `http://www.analog.com/media/en/technical-documentation/data-sheets/REF19xSeries.pdf`, visited 2018-02-22.

[15] F. Arqueros, J. R. Hörandel, and B. Keilhauer. Air fluorescence relevant for cosmic-ray detection—Summary of the 5th fluorescence workshop, El Escorial 2007. *Nuclear Instruments and Methods in Physics Research A*, 597:1–22, November 2008. arXiv:0807.3760 [astro-ph].

[16] G. A. Askaryan. Excess negative charge of an electron-photon shower and its coherent radio emission. *Sov. Phys. JETP*, 14(2):441–443, 1962. [Zh. Eksp. Teor. Fiz.41,616(1961)].

[17] P. Auger, P. Ehrenfest, R. Maze, J. Daudin, J. Robley, and A. Fréon. Extensive cosmic-ray showers. *Rev. Mod. Phys.*, 11:288–291, July 1939.

[18] V. Berezinsky, M. Kachelriess, and A. Vilenkin. Ultrahigh-energy cosmic rays without GZK cutoff. *Phys. Rev. Lett.*, 79:4302–4305, 1997. arXiv:astro-ph/9708217.

[19] P. Bhattacharjee and G. Sigl. Origin and propagation of extremely high-energy cosmic rays. *Phys. Rept.*, 327:109–247, 2000. arXiv:astro-ph/9811011.

[20] A. Biland et al. Calibration and performance of the photon sensor response of FACT — the first G-APD Cherenkov telescope. *Journal of Instrumentation*, 9(10):P10012, 2014. arXiv:1403.5747 [astro-ph.IM].

[21] D. J. Bird, S. C. Corbato, H. Y. Dai, J. W. Elbert, K. D. Green, M. A. Huang, D. B. Kieda, S. Ko, C. G. Larsen, E. C. Loh, M. Z. Luo, M. H. Salamon, J. D. Smith, P. Sokolsky, P. Sommers, J. K. K. Tang, and S. B. Thomas. Detection of a cosmic ray with measured energy well beyond the expected spectral cutoff due to cosmic microwave radiation. *The Astrophysical Journal*, 441:144–150, March 1995. arXiv:astro-ph/9410067.

[22] C. Bleve et al. Update of the neutrino and photon limits from the Pierre Auger Observatory. *Proceedings of Science*, ICRC2015:1103, 2015. `https://pos.sissa.it/236/1103`.

[23] S. Blin, P. Barrillon, and C. de La Taille. MAROC, a generic photomultiplier readout chip. *Journal of Instrumentation*, 5(12):C12007, 2010.

[24] G. R. Blumenthal. Energy Loss of High-Energy Cosmic Rays in Pair-Producing Collisions with Ambient Photons. *Phys. Rev. D*, 1:1596–1602, March 1970.

[25] J. Blümer, R. Engel, and J. R. Hörandel. Cosmic rays from the knee to the highest energies. *Progress in Particle and Nuclear Physics*, 63:293–338, October 2009. arXiv:0904.0725 [astro-ph.HE].

[26] M. Bouchel, F. Dulucq, J. Fleury, C. de La Taille, G. Martin-Chassard, and L. Raux. SPIROC (SiPM Integrated Read-Out Chip): Dedicated very front-end electronics for an ILC prototype hadronic calorimeter with SiPM read-out. In *Nuclear Science Symposium Conference Record, 2007. NSS '07. IEEE*, volume 3, pages 1857–1860, Oct 2007.

[27] T. Bretz et al. SiPMs – A revolution for high dynamic range applications. *Proceedings of Science*, ICRC2017:472, 2017. `https://pos.sissa.it/301/472/`.

[28] R. Brun and F. Rademakers. ROOT — An object oriented data analysis framework. *Nuclear Instruments and Methods in Physics Research A*, 389(1):81–86, 1997. `http://root.cern.ch`, visited 2018-02-22.

[29] CAEN. Caencomm library – interface library for caen data acquistion modules. `http://www.caen.it/csite/CaenProd.jsp?parent=38&idmod=684`, visited 2018-02-22.

[30] S. Callier, C. de La Taille, G. Martin-Chassard, and L. Raux. EASIROC, an Easy & Versatile ReadOut Device for SiPM. *Physics Procedia*, 37:1569 – 1576, 2012. Proceedings of the 2nd International Conference on Technology and Instrumentation in Particle Physics (TIPP 2011).

[31] Pierre Auger Collaboration. Observation of a Large-scale Anisotropy in the Arrival Directions of Cosmic Rays above 8×10^{18} eV. *Science*, 357(6537):1266–1270, 2017. arXiv:1709.07321 [astro-ph.HE].

[32] H. P. Dembinski. *Measurement of the flux of ultra high energy cosmic rays using data from very inclined air showers at the Pierre Auger Observatory.* PhD thesis, RWTH Aachen, 2009.

[33] Emerson Network Power. *MVME3100 Series–VMEbus Single-Board Computer.* `http://www.elma.com/~/media/product-files/region-americas/systems-solutions/embedded-boards/datasheets-021213/mvme3100_ds.ashx`, visited 2018-02-22.

[34] J. Fleury, S. Callier, C. de La Taille, N. Seguin, D. Thienpont, F. Dulucq, S. Ahmad, and G. Martin. Petiroc and Citiroc: front-end ASICs for SiPM read-out and ToF applications. *Journal of Instrumentation*, 9(01):C01049, 2014.

[35] FLUKE. *Fluke 8845A/8846A Extended Specifications.* `http://media.fluke.com/documents/2748044_6200_ENG_B_W.PDF`, visited 2018-02-22.

[36] T. Förster. Zwischenmolekulare Energiewanderung und Fluoreszenz. *Annalen der Physik*, 437:55–75, 1948.

[37] FTDI Chip. *FT245BL Data Sheet*, 2005. `http://www.ftdichip.com/Support/Documents/DataSheets/ICs/DS_FT245BL.pdf`, visited 2018-02-22.

[38] S. Fukuda et al. The super-kamiokande detector. *Nuclear Instruments and Methods in Physics Research A*, 501(2–3):418 – 462, 2003.

[39] T. K. Gaisser and A. M. Hillas. Reliability of the method of constant intensity cuts for reconstructing the average development of vertical showers. *Proc. 15th ICRC*, 8:353–357, 1977.

[40] K. Greisen. End to the Cosmic-Ray Spectrum? *Physical Review Letters*, 16:748–750, April 1966.

[41] Hamamatsu Photonics. *MPPC (Multi-Pixel Photon Counter) S13360 series – Datasheet.* `http://www.hamamatsu.com/resources/pdf/ssd/s13360_series_kapd1052e.pdf`, visited 2018-02-22.

[42] Hamamatsu Photonics. *MPPC S12571-025, -050, -100C/P – Datasheet.* `http://www.hamamatsu.com.cn/UserFiles/DownFile/Product/20140117210531642.pdf`, visited 2018-02-22.

[43] T. Hebbeker, M. Lauscher, L. Middendorf, T. Niggemann, and C. Peters. Muon measurement using scintillator tiles with SiPM readout. Auger internal note, GAP-2013-057, 2013.

[44] V. F. Hess. Über Beobachtungen der durchdringenden Strahlung bei sieben Freiballonfahrten. *Physikalische Zeitschrift*, 13:1084–1091, 1912.

[45] A. M. Hillas. The Origin of Ultra-High-Energy Cosmic Rays. *Annual review of astronomy and astrophysics*, 22:425–444, 1984.

[46] M. Janecek and W. W. Moses. Optical Reflectance Measurements for Commonly Used Reflectors. *IEEE Transactions on Nuclear Science*, 55:2432–2437, August 2008.

[47] F.D. Kahn and I. Lerche. Radiation from cosmic ray air showers. *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 289(1417):206–213, 1966.

[48] K.-H. Kampert and M. Unger. Measurements of the cosmic ray composition with air shower experiments . *Astroparticle Physics*, 35(10):660 – 678, 2012. `http://dx.doi.org/10.1016/j.astropartphys.2012.02.004`.

[49] J. L. Kelley, J. Abraham, et al. AERA: the Auger Engineering Radio Array. *Proc. 32nd Int. Cosmic Ray Conf.*, 2011. arXiv:1107.4807v1 [astro-ph.IM].

[50] J. Kemp et al. The application of SiPMs in the fluorescence telescope FAMOUS and the Aachen Muon Detector. *Proceedings of Science*, ICRC2017:466, 2017. `https://pos.sissa.it/301/466/`.

[51] Ketek. Sipm technology. `https://www.ketek.net/sipm/technology/microcell-construction/`, visited 2018-02-22.

[52] D. Keßler. Coupling of optical fibers for the Aachen Muon Detector, 2016. bachelor thesis.

[53] K. Kotera and A. V. Olinto. The Astrophysics of Ultrahigh-Energy Cosmic Rays. *Annual Review of Astronomy and Astrophysics*, 49:119–153, September 2011. arXiv:1101.4256 [astro-ph.HE].

[54] M. Lauscher, S. Navas, et al. Dynamic range measurement and calibration of SiPMs. *Journal of Instrumentation*, 11(03):P03009, 2016.

[55] W.R. Leo. *Techniques for Nuclear and Particle Physics Experiments.* Springer, 2 edition, 1994. ISBN 978-0-3875-7280-2.

[56] T. H.-J. Mathes. The HEAT Telescopes of the Pierre Auger Observatory: Status and First Data. *Proc. 32nd Int. Cosmic Ray Conf.*, 2011. arXiv:1107.4807v1 [astro-ph.IM].

[57] Maxim Integrated. *DS18B20 Programmable Resolution 1-Wire Digital Thermometer – Data Sheet*, 2015. `https://datasheets.maximintegrated.com/en/ds/DS18B20.pdf`, visited 2018-02-22.

[58] R. Meißner. Development and Characterisation of a Scintillator Based Muon Detector with SiPM Readout for Air Shower Experiments. Master's thesis, RWTH Aachen University, 2015.

[59] L. Middendorf. Search for Ultra-High Energy Photons from Centaurus A with the Pierre Auger Observatory. Master's thesis, RWTH Aachen University, 2012.

[60] National Instruments. *LabVIEW System Design Software.* `http://www.ni.com/labview`, visited 2018-02-22.

[61] T. Niggemann. *The silicon photomultiplier telescope FAMOUS for the detection of fluorescence light.* PhD thesis, RWTH Aachen University, 2016.

[62] T. Niggemann et al. G4SiPM: A novel silicon photomultiplier simulation package for Geant4. *Nuclear Instruments and Methods in Physics Research A*, 787:344 – 347, 2015. New Developments in Photodetection NDIP14.

[63] K. A. Olive et al. Review of Particle Physics. *Chin. Phys.*, C38:090001, 2014. with 2015 update.

[64] Ω micro. *EASIROC datasheet*, 2011.

[65] C. Peters. *Measurement of the muonic component in air showers with the Pierre Auger Observatory.* PhD thesis, RWTH Aachen University, to be submitted in 2018.

[66] C. Peters et al. The muon detector prototype AMD for the determination of the muon content in UHECRs. *Proceedings of Science*, ICRC2015:596, 2016. https://pos.sissa.it/236/596/.

[67] C. Peters et al. Prospects of silicon photomultipliers for ground-based cosmic ray experiments. *JPS Conference Proceedings*, 19 (UHECR2016):011030, 2018. https://doi.org/10.7566/JPSCP.19.011030.

[68] W. W. Peterson and D. T. Brown. Cyclic codes for error detection. *Proceedings of the IRE*, 49(1):228–235, 1961.

[69] O. Pooth, S. Weingarten, and L. Weinstock. Efficiency and timing resolution of scintillator tiles read out with silicon photomultipliers. *Journal of Instrumentation*, 11(01):T01002, 2016.

[70] Saint-Gobain Crystals. *Plastic Scintillating Fibers.* product broschure https://www.crystals.saint-gobain.com/sites/imdf.crystals.com/files/documents/fiber_product_sheet.pdf, visited 2018-02-22.

[71] N. Sakaki, Y. Watanabe, M. Nagano, and K. Kobayakawa. Fluorescence in air excited by electrons from a ^{90}Sr source. *Nuclear Instruments and Methods in Physics Research A*, 597:88–93, November 2008.

[72] F. Salesa Greus. Searches for galactic neutron sources with the Pierre Auger Observatory. In *Proceedings, 33rd International Cosmic Ray Conference (ICRC2013): Rio de Janeiro, Brazil, July 2-9, 2013*, page 1125.

[73] SensL sense light. *An Introduction to the Silicon Photomultiplier*, 2011. http://www.sensl.com/downloads/ds/TN%20-%20Intro%20to%20SPM%20Tech.pdf, visited 2018-02-22.

[74] IEEE Computer Society. 1364-2005 – IEEE Standard for Verilog Hardware Description Language, 2006. ISBN 978-0-7381-4851-9.

[75] IEEE Computer Society. 1076-2008 – IEEE Standard VHDL Language Reference Manual, 2009. ISBN 978-0-7381-6853-1.

[76] IEEE Computer Society. 1149.1-2013 - IEEE Standard for Test Access Port and Boundary-Scan Architecture, 2013. ISBN 978-1-5044-1497-5.

[77] F. Sánchez, J. Abraham, et al. The AMIGA detector of the Pierre Auger Observatory: an overview. *Proc. 32nd Int. Cosmic Ray Conf.*, 2011. arXiv:1107.4807v1 [astro-ph.IM].

[78] Texas Instruments. *MSP430F20x3 Mixed Signal Microcontroller – Datasheet.* `http://www.ti.com/lit/gpn/msp430f2013`, visited 2018-02-22.

[79] Texas Instruments. *MSP430F565x Mixed Signal Microcontrollers – Datasheet.* `http://www.ti.com/lit/gpn/msp430f5659`, visited 2018-02-22.

[80] Traco Power. *DC/DC Converter, THD 12WI Series, 12 Watt – Datasheet.* `www.tracopower.com/products/thd12wi.pdf`, visited 2018-02-22.

[81] Traco Power. *DC/DC Step-Down Converter, TSR 1 Series, 1 Amp – Datasheet.* `www.tracopower.com/products/tsr1.pdf`, visited 2018-02-22.

[82] I. Valino et al. The flux of ultra-high energy cosmic rays after ten years of operation of the Pierre Auger Observatory. *Proceedings of Science*, ICRC2015:271, 2015. `https://pos.sissa.it/236/271`.

[83] Wiener Plein & Baus Elektronik. *VM-USB VME Controller with USB-2 interface.* `http://www.wiener-d.com/sc/modules/vme--modules/vm-usb.html`, visited 2018-02-22.

[84] M. Wirtz. Simulation studies of a novel scintillator detector with SiPM readout for muons of cosmic ray air showers, 2014. bachelor thesis.

[85] G. T. Zatsepin and V. A. Kuz'min. Upper Limit of the Spectrum of Cosmic Rays. *Journal of Experimental and Theoretical Physics Letters*, 4:78, 1966.

Declaration of pre-released results

Some measurements have been performed together with Rebecca Meißner and are published in her master thesis:

[58] R. Meißner. Development and Characterisation of a Scintillator Based Muon Detector with SiPM Readout for Air Shower Experiments. Master's thesis, RWTH Aachen University, 2015.

A much simplified description of the detector, the data acquisition and the measurement procedures developed for this thesis is included in that work. No text has been shared with that work, images taken from that work are cited where applicable. The measurements described in that work are only paraphrased and cited in this thesis. The contributions of Rebecca Meißner to the hardware development (SiPM carrier board; section 9.1.2) has been declared at the appropriate place. The firmware and software described in and used for this thesis has been written by the author of this thesis. The measurement programs described in this thesis have partially been used also for that work.

Some of the measurements performed for this thesis have already been published in a very short form in the following conference proceedings:

[66] C. Peters et al. The muon detector prototype AMD for the determination of the muon content in UHECRs. *Proceedings of Science*, ICRC2015:596, 2016. https://pos.sissa.it/236/596/

[67] C. Peters et al. Prospects of silicon photomultipliers for ground-based cosmic ray experiments. *JPS Conference Proceedings*, 19 (UHECR2016):011030, 2018. https://doi.org/10.7566/JPSCP.19.011030

[50] J. Kemp et al. The application of SiPMs in the fluorescence telescope FAMOUS and the Aachen Muon Detector. *Proceedings of Science*, ICRC2017:466, 2017. https://pos.sissa.it/301/466/

The author of this thesis is coauthor but not the primary author of those works. No text has been shared between those works and this thesis. Any graphics have been cited to those works where applicable. Unless stated otherwise the measurements and analyses in this thesis have been performed by the author of this thesis.

Appendix A.

Firmware register description

A write to any register causes a reset of all event FIFOs.

A.1. `0x00`: probe config

Configuration for the analog probe output of the EASIROC. Directly gives the number of the enabled signal (or disabled if ≥ 160). The value is converted to the corresponding shift register bit in the EASIROC by the `probe_to_asic` module triggered by the `write_probeconfig` command.

Beginning from zero, the first 64 numbers select the signal of the preamplifiers for the different channels in pairs of two (even numbers for high-gain, odd numbers for low-gain). Beginning from 64, the next 64 numbers select the signal after the slow shapers for the different channels in pairs of two (even numbers for high-gain, odd numbers for low-gain). Beginning from 128, the next 32 numbers select the fast shaper signal of the different channels. Any number greater or equal to 160 disables the probe output.

Default: 0xff

A.2. `0x01`: hold delay

Directly gives the 8 bit value (0..255) of the hold delay in steps of 2.5 ns.

Default: 0x18

A.3. `0x02`: fastread config

Configuration of the fastread function to perform ADC readout only for the selected channel instead of all 32 channels.

bit	name	function
[4:0]	channel	channel number
5	enable	enable fastread
[7:6]		unused

Default: 0x00

A.4. 0x03: config A

bit	name	function
0	clk_select_ADC	select clock for ADC readout (0: 10 MHz, 1: 2.5 MHz)
1	pulser_enable	enable internal pulser counter for calibration measurements
2	trig_src_pulser	use internal pulser counter as trigger source for events
3	trig_src_FPGA_OR	use FPGA_OR as trigger source for events
4	trig_src_multiplicity	use multiplicity trigger as trigger source for events
5	trigger_outputs_enable	enable trigger outputs (default: enabled)
6	enable_confirm	use external trigger confirm input
7	LED_dark	turn off blinking of the LED on the EASIROC board

Default: 0x20

A.5. 0x04: OR_maskA

FPGA OR mask bits 0..7

Default: 0xff

A.6. 0x05: OR_maskB

FPGA OR mask bits 8..15

Default: 0xff

A.7. 0x06: OR_maskC

FPGA OR mask bits 16..23

Default: 0xff

A.8. 0x07: OR_maskD

FPGA OR mask bits 24..31

Default: 0xff

A.9. 0x08: multiplicity threshold

Threshold for multiplicity trigger. Number of active channels has to be **greater** than the given threshold to cause a trigger.

bit	name	function
[5:0]	`multiplicity_threshold`	multiplicity threshold
[7:6]		unused

Default: 0x02

A.10. 0x09: multi_maskA

multiplicity mask bits 0..7

Default: 0xff

A.11. 0x0a: multi_maskB

multiplicity mask bits 8..15

Default: 0xff

A.12. 0x0b: multi_maskC

multiplicity mask bits 16..23

Default: 0xff

A.13. 0x0c: multi_maskD

multiplicity mask bits 24..31

Default: 0xff

A.14. 0x0d: DAQ outputs

tri-stated DAQ lines to actively pull to GND

DAQ signals DAQ5, DAQ6, DAQ7, DAQ8, DAQ9, DAQ10, DAQ11, DAQ12. Corresponding to pins 7 to 14 on the DAQ connector.

Default: 0x00

A.15. 0x0e: DAQ inputs

tri-stated DAQ lines that are low

DAQ signals DAQ5, DAQ6, DAQ7, DAQ8, DAQ9, DAQ10, DAQ11, DAQ12. Corresponding to pins 7 to 14 on the DAQ connector.

read-only

A.16. 0x0f: firmware revision

Read only firmware revision (8 bits).

A.17. 0x10: multiplicity overlap

The time window used for the generation of the multiplicity trigger. Each trigger input is stretched by the given number of 100 MHz clock cycles.

0: Multiplicity threshold has to be reached in a single clock cycle of the 100 MHz clock.

4: Multiplicity threshold has to be reached by channels active at any time in five consecutive clock cycles of the 100 MHz clock.

Depending on when the triggers arrive relative to the 100 MHz clock, the actual window width differs between $\mathit{overlap} \times 10\,\mathrm{ns}$ and $(\mathit{overlap} + 1) \times 10\,\mathrm{ns}$.

bit	name	function
[3:0]	`multiplicity_overlap`	multiplicity overlap window
[7:4]		unused

Default: 0x04

A.18. `0x11`: configB

Enable bits for special triggers.

bit	name	function
0	`trig_src_special[0]`	enable special trigger 0 as trigger source Any fixed pair of channels (0+1; 2+3; 4+5; 6+7)
1	`trig_src_special[1]`	enable special trigger 1 as trigger source One of four channels (0,2,4,6) and one of four channels (1,3,5,7)
2	`trig_src_special[2]`	enable special trigger 2 as trigger source Any fixed pair of channels (0+8; 1+9; 2+10; 3+11; 4+12; 5+13; 6+14; 7+15)
3	`trig_src_special[3]`	enable special trigger 3 as trigger source One of eight channels (1-7) and one of eight channels (8-15)
[7:4]		unused

Default: 0x00

A.19. 0x12: configC

bit	name	function
0	trig_sync_discri_adc	enforce simultaneous recording of ADC events and discriminator events
1	evno_count_all	count all triggers in event number not just those leading to event storage (default: enabled)
2	trigger_outputs_karlsruhe	switch to trigger output assignment for Karlsruhe measurements
3	enable_veto	enable veto defined by FPGA-OR mask
4	discrievent_exttrig	trigger discriminator events by external trigger input
5	trig_src_stacks	enable stack multiplicity as trigger
6	trig_src_multiplicity2	enable second multiplicity as trigger
7	trig_src_multi_and	enable AND connection of both multiplicities as trigger

Default: 0x02

A.20. 0x13: trigger holdoff

trigger holdoff time in 10 ns

The trigger condition must not have been fulfilled for this time period before a new trigger can be generated.

Default: 0x03

A.21. 0x14: stack threshold

Threshold for stack multiplicity trigger. Number of active stacks (channels 0+8; 1+9; 2+10; 3+11; 4+12; 5+13; 6+14; 7+15) has to be **greater** than the given threshold to cause a trigger.

bit	name	function
[3:0]	stack_threshold	stack threshold
[7:4]		unused

Default: 0x00

A.22. 0x15: stack mask

stack mask: Stacks to be included in stack multiplicity trigger.

Default: 0xff

A.23. 0x17: reset

Write-only volatile reset register. Write with respective bit enabled to cause specific reset (bit clears automatically).

bit	name	function
0	reset_timestamp	reset timestamp counter to zero
1	reset_eventnumber	reset event number counter to zero
[7:2]		unused

A.24. 0x18: multiplicity threshold 2

Threshold for multiplicity trigger 2. Number of active channels has to be **greater** than the given threshold to cause a trigger.

bit	name	function
[5:0]	multiplicity_threshold2	multiplicity threshold 2
[7:6]		unused

Default: 0x02

A.25. 0x19: multi_mask2A

multiplicity mask 2 bits 0..7

Default: 0xff

A.26. 0x1a: multi_mask2B

multiplicity mask 2 bits 8..15

Default: 0xff

A.27. `0x1b`: multi_mask2C

multiplicity mask 2 bits 16..23

Default: 0xff

A.28. `0x1c`: multi_mask2D

multiplicity mask 2 bits 24..31

Default: 0xff

A.29. `0x1f`: scratch

Register that can be read and written for test purposes. Might be used to control non-permanently implemented test-functionality in the firmware.

Default: 0x00

Appendix B.

Cabling

Various cable connections between the different boards have to be established. These are described here mainly as reference for future users of the system.

The common female connector for the eight coaxial cables of one SiPMs carrier board (see figure 9.6) has to be connected to the J8 connector on the EASIROC board (left edge in figure 9.8). The orientation is such that the upper row connects to the inner conductors, the bottom row to the shielding. Four carrier boards can be connected next to each other.

The power supply cable for the EASIROC board (see left image in figure B.1) can be connected in any orientation. The outer red cable is connected to the screw terminal labeled 6.5V on the PSU board, the black center ground cable is connected to the adjacent GND terminal on the PSU. The blue cable can be left free or connected to ground, too. If operation of the analog debug outputs of the EASIROC board is required, the blue cable has to be connected to some source of −7.5 V (relative to ground). To avoid excessive differences of the ground levels of the EASIROC board and the PSU, which transfers onto the SiPM bias voltage, the black ground cable has to be as short and thick as possible.

The ribbon cable shown in the right image of figure B.1 connects the EASIROC board to the control input of the PSU V2 and to the temperature sensor. It is connected to the DAQ connector of the EASIROC board (top of right edge in figure 9.8) on the one end and the PSU on the other end. The four jointed wires (connected to pins 7-10 of the DAQ connector; named DAQ5-DAQ8 in the board schematics) are connected to the *FPGA Interface* of the PSU V2 (cable facing towards the board edge), the single cable (connected to pin 11 of the DAQ connector; named DAQ9 in the board schematics) is connected to the top pin (figure 9.4) of the enable pin header (labeled *0V* in PSU version 2). The short free wire connected to pin 1 of the DAQ connector is used to connect to the yellow data line of the temperature sensors. For the PSU version 3 a similar ribbon cable with eight connected pins on the PSU end is foreseen to connect the PSU to the pins 7-14 of the DAQ connector (named DAQ5-DAQ12 in the board schematics).

The temperature sensors are currently connected using a big Belgium-colored cable assembly (individual wires) and a three-pin connector at the sensor ends. Black is

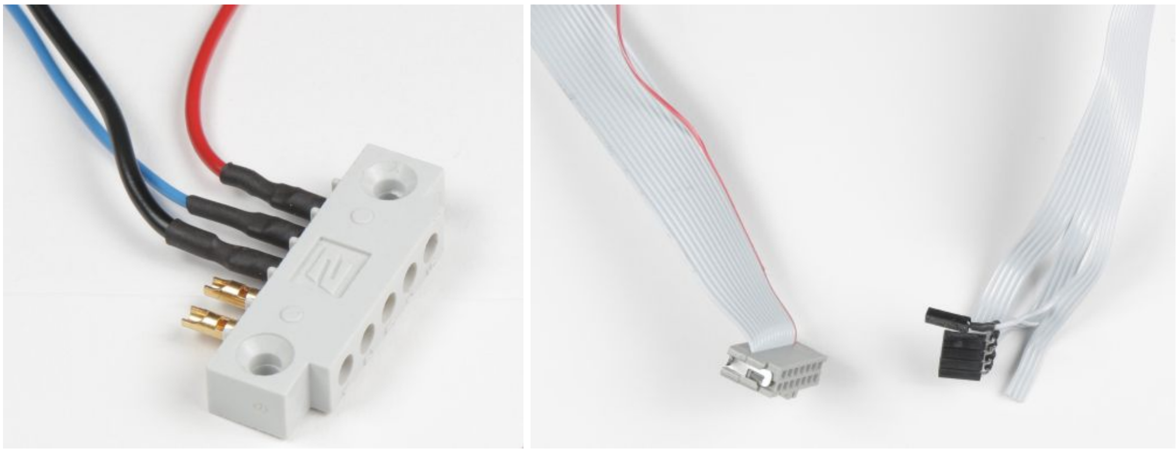


Figure B.1.: Power cable for the EASIROC board (left) and control cable for the PSU board (right).

ground, red is VCC (3.3 V) and yellow is the data line. At one point of the cable, the data line has to be connected to the VCC line using a 5 kΩ pull up resistor[1]. A ribbon cable or a twisted pair cable will likely be used for the final setup. The black ground cable is connected to pin 2 (GND) of the *serial active* shrouded header J4 on the EASIROC board (above the FPGA in figure 9.8), while the red cable is connected to pin 4 (3.3 V) on the same header. The yellow data line is connected to pin 1 of the DAQ connector on the EASIROC board using the connection provided on the PSU control cable.

The bias voltage output of the PSU board has to be connected to the bias voltage input of the SiPM carrier board. As ground is provided via the shields of the signal cables, no separate ground connection to the SiPM carrier board is installed to reduce ground loops. The ground connection of the SiPM carrier board is needed just for shielding and the low pass filters, the supply current of the EASIROC flows back through the signal cable into the EASIROC.

The 24 V supply voltage has to be connected to the supply input of the PSU board. A colored cable bundle is used for the test setup, where the black and red cables are to connect the negative and positive pole of the 24 V voltage supply. The blue cable can be connected to the blue cable of the EASIROC board power connector to transfer −7.5 V from an additional external power supply if desired. The yellow, white and green cables can be used to measure the SiPM bias voltage and the current flowing, and to produce the ramp-up and ramp-down of the supply voltage with the version 1 PSU board. In this case, no direct connection between the PSU and the SiPM carrier board is installed. The yellow cable is connected to a GND terminal on the PSU board and the white cable is connected to the SiPM bias voltage terminal on the PSU board. The voltage between the yellow and the white cable can be measured externally with a multimeter. A separate ground line is needed for the measurement due to the voltage drop over the black cable. The green line can connect the SiPM

[1] The exact value has to be adjusted when building the complete prototype using more sensors and longer cables.

bias voltage to the SiPM carrier board. The white and the green cable have to be externally connected, and the connection point can be used to measure the SiPM current and to connect and disconnect the SiPM voltage.

Appendix C.

Detailed description of firmware modules

C.1. Detailed description of the ADC readout

Data of the events is stored in a FIFO for subsequent data transfer, which is performed by the transfer_ADC submodule of the slow control.

Some time after a trigger, the *track* signal (inverted *hold*) for the EASIROC (output name holdb_alt) has to be disabled. This output is internally driven by the signal track_intern in the main firmware module, which is effectively an inverted hold signal. The hold delay and hold signal are driven with the 400 MHz clock. When trigger_raw becomes active and a new event can be recorded (not already holding, and in case the configuration bit trig_sync_discri_adc is active when also the discriminator event can receive a trigger) the hold_delay_running register is enabled. The 8 bits hold_delay_counter counts up while hold_delay_running is true (and is reset to 0 otherwise). When the hold_delay_counter reaches the selected hold_delay configuration register value, the hold_start signal is enabled for one clock cycle. This causes hold_delay_running as well as track_intern to be disabled in the next clock cycle. The hold_delay_running signal (as well as derived signals hold_delay_running_100 and new_hold_delay_100) is used in some other parts of the module to recognize that a new event trigger has arrived and is being acted upon.

When the raw trigger arrives and no previous event is still being handled, the current timestamp is copied into the register ADC_event_timestamp for inclusion in the event data. The event number is also copied for inclusion in the event data after the event number has been incremented, either directly after the hold delay starts or after the external confirm trigger occurs. The event triggerinfo is copied into the register ADC_triggerinfo. For this, trigger_channel_info (section 9.2.7), which is a mask indicating the trigger channels that have been active in the last 2×multiplicity_overlap clock cycles, is copied after the proper delay to record an interval symmetrical around the trigger time. This ensures that regardless of the time when the multiplicity condition is fulfilled in the multiplicity overlap window, the complete window is always included in the triggerinfo.

Once track_intern is disabled, if the ADC readout is not already in progress and the external confirm trigger happened (only when enable_confirm is enabled), the run_read_ADC signal is enabled with the 100 MHz clock. This causes the read_ADC module to start the actual readout with the ADC.

The control signals of the EASIROC, which steer the analog multiplexer of the track and hold cells, are driven by two different modules in the firmware (read_ADC and fastread_configurator) with a multiplexer deciding which module to give the control. The latter has the control when fastread is enabled or being enabled/disabled, and the former in all other cases.

C.1.1. Configuration of fastread

The fastread_configurator module controls the enabling of the fast readout of the ADC. While during normal ADC readout all stored channel voltages are read out channel-by-channel using the analog multiplexers in the EASIROC, for fastread the multiplexers are set to permanently output one channel. On setup of fastread, the read control lines to the EASIROC are driven like for a normal readout but stopped when the multiplexer outputs are set to the selected channel. The read control lines are no longer modified until a different channel number is selected or fastread is disabled. The interrupted "readout" can be resumed when a higher channel number is selected, otherwise a reset has to be performed first.

The *fastread config* configuration register contains the number of the selected channel (5 bits) and one enable bit. The fastread_configurator permanently monitors this register and acts on a change. The actual configuration is done by a state machine in the fastread_configurator module. Its status is shown with the two output signals busy, which indicates fastread configuration (or reset) is ongoing, and fastread_configured, which indicates that fastread has been setup and its configuration can be used for the actual ADC readout.

C.1.2. The read_ADC module

The read_ADC module handles the actual readout of the slow shaper data of the EASIROC using the external ADC on the EASIROC board and stores the data into an internal FIFO. The FIFO is an instance of dcfifo with 32 bits line width and a capacity multiple of 32 lines (currently 32 × 32 = 1024 lines[1]). The hardware resources needed for the FIFO are one M4K per 128 lines (four events).

For its operation the module uses a clock with variable frequency (or rather a fixed clock with a variable clock enable) to enable different speeds of ADC readout. The different clock frequencies are provided to accommodate different drive strengths of

[1] parameter ADC_eventcount_bits

the EASIROC analog output, which besides power consumption also influences the time needed for the voltage output of a channel to be stable after selection of this channel in the analog multiplexer.

Inputs and outputs of the module are the control and data lines to and from the ADCs and the control lines to the EASIROC. Further input/output signals are the read port of the internal FIFO which is connected to the slow control for data transfer. To be embedded into the ADC event, the module receives the event timestamp, the event number and the event triggerinfo. Finally, the control lines `run`, `busy`, `reset` and `fastread` are controlling and signaling the operation of the module.

The actual work of the readout is performed by a state machine that is started by the `run` control signal. Operation only commences when there is enough free space in the FIFO for one complete event. For the case that fastread is enabled, one event contains only one line, otherwise 32 FIFO lines are required. First the read register of the EASIROC is reset and afterwards the first channel is selected for readout. Four clock cycles after an EASIROC channel is selected, the clock edge causing the ADCs to sample the voltage is generated. After sampling with the ADCs, the next channel is selected in the EASIROC and the cycle is repeated for a total of 32 times. The state machine handles the case of fastread by skipping the states that change the EASIROC read configuration and ending the read after the single ADC data set has been written into the FIFO.

The used ADCs are pipelining ADCs, which means the digital data appears on the output only three clock cycles after the actual sampling happens. When reading channel 3, the write request line of the internal FIFO is enabled (and disabled again in the next clock cycle) to cause the data to be saved. After the last channel data has been sampled, the ADC clock is cycled three additional times to flush all data from the ADCs and store it into the internal FIFO. As the internal channel counter of the module is already at channel 3 at the time the data of channel 0 appears at the ADC output and is written to the FIFO, any additional data to be injected into the channel data has to be internally reordered to account for this.

C.2. Detailed description of the `discrievents` firmware module

The external interface of the `discrievents` module is a FIFO from which the event data can be read byte-wise and which indicates whether a complete event has been read. The FIFO readout is operated with a different clock (50 MHz) than the sampling clock.

Actual event storage is done using multiple instances of ring buffers, which continuously record the discriminator status. When a trigger is received, a predetermined (fixed) number of post-trigger samples is recorded until a ring buffer is frozen and the corresponding timestamp is saved, while the other ring buffers (if not frozen themselves previously) continue to be updated for the next trigger. On read, the earliest recorded event is transferred and afterwards the ring buffer is unfrozen and ready for the next trigger (after it has updated all required pre-trigger samples).

The number of available ring buffers is set with the parameter `discrievent_number`. The parameter can be zero, in which case attempts to read are ignored, and it is always indicated that no event is ready and that no trigger can be received.

Two shift registers of `discrievent_number` bits length, named `current_read` and `next_trigger`, are used to track which ring buffer is to receive the next trigger and which one is to be read from next. They are both initialized to select the first ring buffer (lowest bit active). If a read succeeds, `current_read` is left-shifted one position in a ring with the highest bit being shifted into the lowest bit, which selects the next ring buffer to read. If the currently active ring buffer is becoming frozen and the trigger input is disabled again, then `next_trigger` is left-shifted one position in a ring. The condition that the buffer has to *become* frozen (i.e. is frozen and has not been frozen the previous clock cycle) ensures that `next_trigger` does not simply continues to be advanced to the next buffer if it is set to an already frozen buffer. In the case that `discrievent_number` equals one, the single bits of both shift registers just stay active permanently[2]. There is no explicit relation between `current_read` and `next_trigger`, but as long as reads are only performed if the `discrievent_ready` output of the `discrievents` module is active, `current_read` always selects the oldest frozen buffer (or the buffer pointed to by `next_trigger` if no buffer is frozen) and `next_trigger` always selects the next free buffer (or the buffer to be freed in the next read access if all buffers are frozen).

A reductive OR (|) used on a vector of signals from the individual ring buffers masked by `current_read` or `next_trigger` respectively can perform as a multiplexer to get the signal from the selected ring buffer. Any control signals to the ring buffers have to be masked with the respective bit from `current_read` or `next_trigger` to affect only the currently selected buffer.

One ring buffer—including the storage of the timestamp and FIFO logic for readout—is implemented in the module `discrievent_storage`, which is instantiated multiple times in a *generate for* loop in `discrievents`. Also included in the same loop is the trigger handling and counters that ensure the post-trigger samples are recorded before the respective ring buffer is frozen using a `storages_hold_multiplicity` signal. `storages_hold_multiplicity` of a ring buffer is released when the signal `storages_rd_hold` is set (and the buffer is selected with `current_read`), and therefore is unfrozen as soon as `storages_rd_hold` is released.

C.2.1. The `discrievent_storage` module

The `discrievent_storage` module has an internal instance of `altsyncram` with a 32 bits wide write port and a 8 bits wide read port for data storage. One M4K block

[2] Both are then optimized away and do not actually use resources.

per 128 samples is needed.

The discriminator status, coming in as the `discri` input, is used as data input for the write port of the memory. In the `wraddr` register the current address being written to is stored (used as input to the write address port of the memory) and incremented every clock cycle as long as the `hold` input is inactive.

The module also contains a 48 bits[3] wide register to store the timestamp of the last recorded sample, which is updated continuously when the memory is written.

The outputs `event_ready` and `pretrig_collected` indicate that the complete event buffer or the pre-trigger samples of an event have been written after a `hold` is released.

The register `rdaddr` is used as address input to the read port of the memory. It is set to zero when `hold` becomes active. For reading (when `hold` is enabled), the `rd_inc` input of the module can be used to get the next data byte (increment `rdaddr` by one), which is provided at the `data` output. First the bytes of the timestamp are provided at the `data` output (until `rdaddr` is equal to the number of bytes in the timestamp minus one), then `rdaddr` is set to the address of the oldest sample in the memory (`wraddr` with the lowest two bits filled with zero) and the data output of the memory is provided at the `data` output.

When `rdaddr+1` is equal to the address it was originally set to, the `read_done` output is enabled to indicate the complete event has been transferred.

[3] parameter `timestamp_bits`

Appendix D.

Driver for communication with the EASIROC board firmware

The driver class for communication with the FPGA uses the `rs232_interface` from the LibLAB (see section 9.3.1. It has been tested to work with the `linux_rs232` and `libftdi_rs232` interface implementations. It could in principle be included in the LibLAB, but has not yet been mainlined due to the limited usability for other experiments, caused by being tailored especially to the custom AMD firmware.

The main functionality is contained in the class `easiroc`, which is defined in the header file `easiroc.h` and implemented in `easiroc.cpp`. The header file also contains the exception class `llbad_easiroc`, which is thrown in case of errors.

The first part of the class is the public definition of some constants (number of bits for the EASIROC slow control configuration and for the timestamps and the size of events) and an `enum reg_addr` giving the addresses for the FPGA configuration registers as described in section 9.2.4. It is followed by unions (containing bit fields) for the FPGA configuration registers which can be decomposed in single bits[1].

`union ASICconfig` is the biggest union in the class, containing a description of the slow control of the EASIROC. As many larger values in the configuration (e.g. the discriminator mask of 32 bits) are not properly aligned to 32 bits boundaries or have a size different to 2^n (the configuration values for the 32 input DACs have a size of 9 bits each, 8 bits for the value and one enable bit) it is in many cases not possible to easily define one large data member or an easily addressable array for these configuration values. For this reason the easiroc class also contains `setConfigBit` and `getConfigBit` methods, together with `get` and `set` methods for various configuration values[2]. The `set/getFeedbackCap` methods take into account the inverted bit order of the high-gain path configuration and hide this fact from the user. The `set/getSSTimeConstant` methods take into account that all bits are negated and hide this fact from the user.

The state of the configuration is stored in the instance of the `easiroc` class as the

[1] Opposed to configuration registers containing only one value in the complete byte.

[2] `setFeedbackCap, getFeedbackCap, setSSTimeConstant, getSSTimeConstant, getDAC8, setDAC8, getDAC8enabled, setDAC8enabled, setDAC10, getDAC10, setDiscriMask, getDiscriMask`

public data member config, which can be synchronized with the content of the RAM in the FPGA using the method readAsicconfig and can be transferred to the FPGA using the method changeAsicconfig. The transferAsicconfig can then be used to trigger the transfer from the FPGA RAM to the EASIROC. Using the method printConfig, a nicely formatted output of the complete or a part of the configuration can be produced. This method also uses the method getBitname, which returns the name of a numbered bit of the configuration. The definitions of the names are stored in the file register_names_array.inc in a form which can be included as an initializer list for an array of strings.

To write or read the FPGA configuration registers, the methods writeRegister and readRegister are implemented. These methods are also called in some convenience functions that provide an easier (and named) access to some configuration registers[3].

After setting the EASIROC probe configuration with changeProbeconfig (the probe bit number corresponding to a specific channel-path combination can be calculated with calcProbenum), the method transferProbe can be used to trigger the actual configuration of the EASIROC.

The methods convertTemp and readTemp are implemented (supporting operation on a single device ID and on all connected devices) to read out the temperature sensors connected to the board. The method tempToDouble is provided to convert the raw binary value to a human-readable temperature. For operation on a single device ID, the ID can either be given as an array of uint8_t, or as a string of a hexadecimal notation of the ID with the different bytes separated by a colon ("xx:xx:..."). To facilitate the second form, the static method strToID takes the ID string and converts it into a vector of uint8_t.

When using the PSU versions 2 and 3 (see section 9.1.1), the DAQ outputs of the EASIROC boards control the PSU output voltage. The methods setVext, getVext, setVextEnabled and getVextEnabled can be used to easily set the corresponding FPGA configuration register, while the methods voltageToVext and VextToVoltage convert the four bit voltage code to the actual voltage and vice versa.

To perform all communication, some private methods are implement the communication protocol described in section 9.2.3.1. The methods sendShort and sendLong dispatch short one-byte commands and long multi-byte commands. The method read is used to read command return messages. It can properly recognize and handle short three-byte return messages and long multi-byte (including extended) return messages, but does not by itself perform the CRC check. The methods executeShort and executeLong dispatch short or long commands (using sendShort and sendLong) and check for a short success return message. Any method using a command which returns a long message has to perform command dispatching using sendShort or

[3] changeProbeconfig, changeHoldDelay, getFirmwareRevision, setFPGA_ORmask, getFPGA_ORmask, setMulti_mask, getMulti_mask, setVext, getVext, setVextEnabled, getVextEnabled

sendLong itself and afterwards process the message returned from read.

Further members of the easiroc class are the definitions of the structs ADC_event and Discri_event, which serve as containers for the methods performing the event data read out. These structs also contain fill methods (and a constructor using it), which take the raw command return messages of the corresponding read commands and parse the data into the data members. For the ADC and discriminator events, there exist two different types of readout methods, old methods reading into raw arrays and new methods reading into std::deques of ADC_event and Discri_event. The old methods can be considered deprecated and are just maintained for compatibility with existing programs, while new programs should use the new methods.

The transferADC methods can be used to read and parse ADC events into data arrays, the ADC_event struct is not used. Two overloaded versions exist, one for single events and one for a bigger number of events. A *valid* Boolean parameter for each event or the return value for the single event version indicates whether data has been read or only a *no data* code was returned. The version for multiple events sends all read commands together instead of waiting for the return message before sending the next read command. This is done for performance reasons, to limit dead time caused by the USB communication with the FT245 and by a direction change of the half-duplex communication of the FT245 FIFO with the FPGA. With 200 events read at the same time, an event readout rate of around 5 kHz can be achieved.

The transferADCsingle methods are equivalent to the transferADC methods, but only read one single 32-bit line of event data, for usage when *fastread* is configured in the FPGA, to get events containing only data from one EASIROC channel. As these single channel data read commands have more overhead per amount of data than the full event (32 channels) read commands, for best performance it is advisable to use the transferADC also for single channel events, which then yields "events" containing not data of 32 different channels but of 32 different real events. Using this approach, the special data bits like timestamp, event number or discriminator status will still be undefined and readout is only possible if 32 events have been accumulated. This then leads to an event readout rate of about 150 kHz, which is enough to read the complete dark noise for 1 mm^2 SiPMs.

To read discriminator events, the method readDiscriEvent is implemented. It reads one discriminator event into an array, not making use of the Discri_event struct.

The methods readDiscriEvents, readADCEvents and readMixedEvents read discriminator events, ADC events or both event types in an alternating succession into a std::deque of the corresponding event structs. The methods send out all read commands before processing any reply messages. New events are added at the back of the double-ended queue, which is passed to the methods by reference and can be reused between multiple calls to the methods even while still containing some previous events (for example when still waiting for the corresponding event of the other event type). The actual event parsing is done by the fill methods of the event structs. Even though `readMixedEvents` always places equal numbers of reads of both event types, a non-matching number of reads might return *no data*. When triggers for both events are synchronized in the FPGA firmware, it is not expected for the number of already read out events to differ by more than one, or for the order of the events to be destroyed.

The method `readRates` fetches the last trigger rate measurements for the 32 channels of the EASIROC. As the raw communication command blocks until new data becomes available and the `read` method of the driver also blocks until an answer message is received, this method blocks until new rate data is available and always succeeds (unless the write itself fails).

D.1. Additional classes

One important class for usage together with the `easiroc` class is `easiroc_dacs`, which is defined and implemented in its own files, named accordingly. It is used to calculate the DAC8 values to be set in the EASIROC to achieve a desired output voltage for SiPM bias voltage regulation and to calculate the voltage corresponding to a given DAC value. It uses the calibration values read from a file produced during the characterization measurements described in section 10.2.2 and takes the currently measured EASIROC temperature into account.