Development of a PDM-Simulator Board for JEM-EUSO

Diplomarbeit
von
Daniel Gottschall

Institut für Astronomie & Astrophysik
High Energy Astrophysics
Kepler Center for Astro and Particle Physics
Mathematisch-Naturwissenschaftliche Fakultät
Eberhard Karls Universität Tübingen

März 2013

Zusammenfassung

JEM-EUSO ist ein UV Teleskop, das von der ISS aus die Atmosphäre beobachtet, um kosmische Strahlung mit Energien über 5.5×10^{20} eV zu detektieren. Ein Teilchen, dass mit einer Energie aus diesem Bereich die Atmosphäre trifft, löst einen ausgedehnten atmosphärischen Teilchenschauer aus. Ein Teil dieses Schauers, der elektromagnetische Anteil, emittiert Licht im UV Bereich. Dieses Licht kommt zum einen von der Fluoreszenz angeregter Stickstoffatome durch Elektronen des Schauers und zum anderen durch Cherenkov Strahlung. Cherenkov Strahlung entsteht dann, wenn sich geladene Teilchen des Schauers mit einer Geschwindigkeit größer als die Phasengeschwindigkeit des Mediums bewegen. JEM-EUSO zeichnet dieses Licht mit Hilfe eines Teleskops, bestehend aus drei Linsen und Photomultipliern, auf, das Ergebnis ist eine Lichtspur in der Atmosphäre. Kapitel 1 beschreibt die Mission, den Aufbau des Teleskops und die astrophysikalischen Hintergründe.

Der Beitrag des Instituts für Astronomie und Astrophysik Tübingen IAAT heisst Cluster Control Board (CCB) und ist Teil der Elektronik der Fokalebene. Die Hauptaufgabe ist die Steuerung der Photo-Detector Module (PDM) und die Berechnung des Level 2 Triggeralgorthmus. Kapitel 2 beschreibt die Funktionsweise des CCB und des PDM. Durch die große Kollaboration ist es nur selten möglich, die komplette Hardware zu testen. Um trotzdem das CCB entwickeln und testen zu können, vor allem das Kalibrieren des Triggeralgorithmus, wurde ein Simulator of the Photo-Detector Module (PDM-Simulator) geplant und gebaut. Diese Arbeit beschäftigt sich mit der Entwicklung des Simulator Boards und ersten erfolgreichen Tests. Kapitel 3 beschreibt den PDM-Simulator und die Funktionsweise. Zum Einsatz kommt ein FPGA, das Verhalten wurde mit very-high-speed integrated circuits hardware description language (VHDL) entwickelt. Der PDM-Simulator ist in der Lage, das CCB mit simulierten Ereignissen zu versorgen. Dabei wird das Interface eins zu eins abgebildet, auch das Verhalten und Timing ähnelt dem echten PDM sehr. Im Verlauf der Arbeit kam als zusätzliche Aufgabe die Versorgung des CCB mit Clocks und einem externen Trigger hinzu.

Als Feldtest der Elektronik ist geplant, den Aufbau mit einem PDM nahe an einem Fluoreszenz Teleskop zu platzieren und dieses als externen Trigger zu verwenden. Vom November 2012 bis Januar 2013 fanden drei Integrationskampagnen statt, um die unterschiedlichen Komponenten aufeinander abzustimmen. Diese Kampagnen werden in Kapitel 4 beschrieben. Außerdem wurde in Tübingen ein kompletter Testaufbau, bestehend aus einem Computer mit SpaceWire Karte, dem PDM-Simulator und dem CCB gebaut und getestet. Dieser Testaufbau is ebenfalls in Kapi-

tel 4 beschrieben.
Insgesamt liefen die Integrationskampagnen erfolgreich und der Testaufbau in Tübingen bewies sich als wichtig und funktionstüchtig. Kapitel 5 beschreibt die wichtige Rolle des PDM-Simulators und fasst die Ergebnisse zusammen.

Abstract

The UV-telescope Extreme Universe Space Observatory on board the Japanese Experiment Module (JEM-EUSO) of the International Space Station (ISS) is a ultra violet (UV) telescope looking downwards to the Earth's atmosphere to detect Ultra High Energy Cosmic Rays (UHECRs) with energies above 5.5×10^{19} eV. A particle with these energies interacting with the atmosphere triggers Extensive Air Showers (EAS). A part of these showers, the so called electromagnetic component containing ultra-relativistic electrons and positrons generates light in the UV range due to Fluorescence and Cherenkov radiation. The EAS results in a track of light and JEM-EUSO records this light with multi anode photomultiplier tubes (MAPMTs), mirroring the track on the focal surface. Chapter 1 describes the mission, the components of the telescope and the theory of Ultra High Energy Cosmic Ray (UHECR) and EAS.

The main hardware contribution of the Institut für Astronomie und Astrophysik Tübingen (IAAT) to the mission, the so-called Cluster Control Board (CCB), is one of the essential elements of the focal surface electronics. The main objective of the CCB is to control the Photo-Detector Modules PDMs and to perform the Level 2 (L2) trigger algorithm. Chapter 2 describes the focal surface electronics from the multi anode photomultiplier tubes (MAPMT)s to the Mission Data Processing Unit (MDP). Since the complexity of the mission and given the large collaboration involved in JEM-EUSO, the entire hardware is usually (and will be) available in one place on very rare occasions in the development process. To be able to test the CCB, especially calibrating the L2 trigger, a Simulator of the Photo-Detector Module (PDM-Simulator) has been planned, designed, manufactured and tested. This work is about the development of the PDM-Simulator and present the first series of successful tests. Chapter 3 describes the hardware of the PDM-Simulator and the functions of the board. A Field Programmable Gate Array (FPGA) was used and the code was developed in very-high-speed integrated circuits hardware description language (VHDL). The interface, the behavior and the timing of the simulator perfectly match the ones of the real PDM. More specifically, the PDM-Simulator is capable of using simulated events transmitting them to the CCB. During the development, it was found necessary to implement an additional functionality: the PDM-Simulator is now able to provide the clocks and an external trigger to the CCB.

A planned field test for JEM-EUSO is to place a setup for one PDM next to the Telescope Array (TA) fluorescence telescope and use the telescope's trigger as an external trigger for the setup. During the end of 2012 and January 2013 three integration sessions took place for

prototype of the JEM-EUSO hardware on the site of Telescope Array (EUSO-TA). These integration sessions are described in Chapter 4. In addition, a laboratory test setup was built and tested in Tübingen. It is described in Chapter 4.
In conclusion the integration sessions were very successful and the test setup in Tübingen proved to be an essential and fully functional tool. Chapter 5 describes the important role of the PDM-Simulator and and gives a résumé of the results.

Contents

List of Figures

List of Tables

Acronyms

ADC	Analog Digital Converters
AGN	Active Galactic Nuclei
ASCII	American Standard Code for Information Interchange
ASIC	Application Specific Integrated Circuit
B2B	board to board
CAS	Column Access Strobe
CCB	Cluster Control Board
CERN	Conseil Européen pour la Recherche Nucléaire
CMB	Cosmic Microwave Background
CPU	Central Processing Unit
CR	Cosmic Rays
CRC	Cyclic Redundancy Check
CREAM	Cosmic Ray Energetics And Mass
CS	Chip Select
CYTOP	Amorphous Peffluoro Alkenylvinylether
DCM	Digital Clock Manager
DRAM	Dynamic Random Access Memory
EAS	Extensive Air Showers
EC	Elementary Cell
ESA	European Space Agency
ESAF	EUSO Simulation and Analysis Framework
EUSO	Extreme Universe Space Observatory

EUSO-Balloon	Extreme Universe Space Observatory Balloon experiment
EUSO-TA	prototype of the JEM-EUSO hardware on the site of Telescope Array
FIFO	First In, First Out
FoV	field of view
FPGA	Field Programmable Gate Array
FS	Focal Surface
FSM	Finite State Machine
GPS	Global Positioning System
GTU	gate time unit
GZK	Greisen-Zatsepin-Kuz'min
HTV	H-II Transfer Vehicle
IAAT	Institut für Astronomie und Astrophysik Tübingen
IC	Integrated Circuit
I^{2}C	Inter-Integrated Circuit
I/O	Input/Output
IR-camera	infrared camera
ISE	Integrated Software Environment
ISS	International Space Station
JAXA	Japan Aerospace Exploration Agency
JEM	Japanese Experiment Module
JEM-EUSO	Extreme Universe Space Observatory on board the Japanese Experiment Module
JTAG	Standard Test Access Port and Boundary-Scan Architecture
KI	special integration circuits
L1	Level 1
L2	Level 2
LAL	Laboratoire de l'Accélérateur Linéaire
LED	Light Emitting Diode
LHC	Large Hadron Collider

LEMO push/pull

LIDAR Light Detection and Ranging unit

LTT Linear Track Trigger

LVDS Low Voltage Differential Signaling

LVPS Low Voltage Power Supply

MAPMT multi anode photomultiplier tubes

MDP Mission Data Processing Unit

MISO master in/slave out

MOSI master out/slave in

NASA National Aeronautics and Space Administration

PAMELA Payload for Antimatter Matter Exploration and Light-nuclei

PC Personal Computer

PDM Photo-Detector Module

PDM-Simulator Simulator of the Photo-Detector Module

PMMA Poly-Methyl-Methacrylate

PROM Programmable Read-Only Memory

RAM Random Access Memory

RAS Row Access Strobe

RIKEN Japanese Institute of Physical and Chemical Research

SDRAM Synchronous Dynamic Random Access Memory

SRAM Static Random Access Memory

SHP Super-Heavy Particles

SPACIROC Spatial Photomultiplier Array Counting and Integrating Read Out Chip

SPI Serial Peripheral Interface

TA Telescope Array

TLE Transient Luminous Events

UHECR Ultra High Energy Cosmic Ray

USA United States of America

USB Universal Serial Bus

UTMI USB 2.0. Transceiver Macrocell Interface

UV ultra violet

VHDL very-high-speed integrated circuits hardware description language

WE Write Enable

WIMP weakly interacting massive particles

1 The JEM-EUSO Mission

1.1 Overview

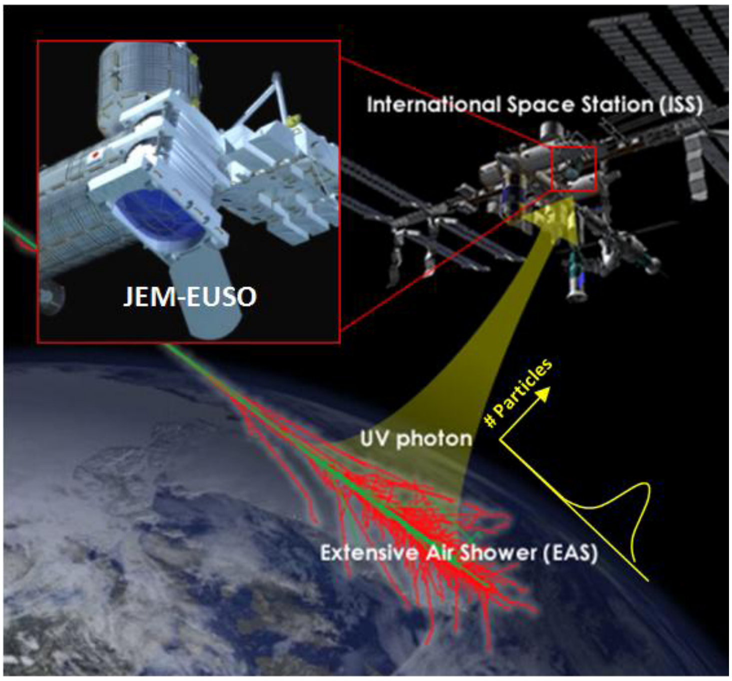

Figure 1.1: The observation principle of JEM-EUSO: Mounted on the ISS, the main instrument will monitor the UV-light produced by extensive air showers which are induced by Ultra High Energy Cosmic Rays UHECRs. (Image: JEM-EUSO Collaboration, 2010)

The UV-telescope *Extreme Universe Space Observatory on board the Japanese Experiment Module* JEM-EUSO of the International Space Station (ISS) is mainly designed to detect EAS generated in the atmosphere by Cosmic Rays (CR) with energies exceeding 5×10^{19} eV. This is the threshold of the so-called Greisen-Zatsepin-Kuz'min (GZK) effect, explained in detail in Section 1.2.3. In addition, it will be able to investigate extreme energy neutrinos, photons, Transient Luminous Events (TLE) and meteors. JEM-EUSO is developed by an international collaboration lead by the RIKEN and the Japan Aerospace Exploration Agency (JAXA). The European contribution is lead by the European Space Agency (ESA) and includes France, Italy, Germany, Spain, Bulgaria, Slovenia and Switzerland. Other participants are from the United

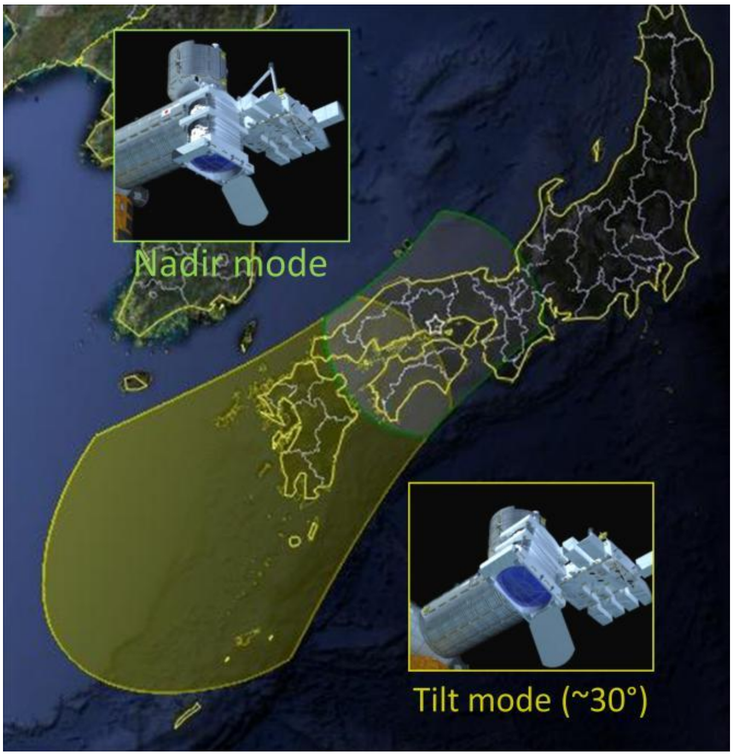

Figure 1.2: Artistic illustration of JEM-EUSO's viewing area: The nadir mode is shown in green and the tilted mode in yellow. A map of Japan is overlayed by the viewing areas. (Image: JEM-EUSO Collaboration, 2010)

States of America (USA), Mexico, Korea, and Russia.

The telescope will be looking downwards from the ISS and will use the atmosphere below it as a detector (see Figure 1.1). An Ultra High Energy Cosmic Ray (UHECR) entering the atmosphere generates a particle shower that can be detected by the UV telescope with a ±30° field of view (FoV), a timing resolution of 2.5 µs and an angular resolution of 1°, which corresponds to a pixel size of $\sim 750\,\mathrm{m}$ on ground. This allows the reconstruction of the energy and direction of the UHECR with good precision. For detailed information about the angular and energy resolution see Section 1.3. JEM-EUSO is expected to launch in 2017 and will operate for three years. Additional two years are possible in case of sufficient funding.

A first project called the Extreme Universe Space Observatory (EUSO) was selected by ESA for an accommodation study in March 2000, but due to financial problems the project was frozen after a successful feasibility conceptual (Phase A) study in 2004. In comparison to the original project of ESA, JEM-EUSO can operate in two modes, *nadir* and *tilted*. In *tilted mode* the telescope is turned 30° away from the zenith which leads to an effective area of $2.9 \times 10^5\,\mathrm{km}^2$ compared to the *nadir mode's* effective area of $1.2 \times 10^5\,\mathrm{km}^2$. The lower energy threshold is better in *nadir mode*, but in *tilted mode* the exposure above $1 \times 10^{20}\,\mathrm{eV}$ doubles and above $2 \times 10^{20}\,\mathrm{eV}$ it is almost five times better (see Figure 1.2). Figure 1.3 shows a comparison of

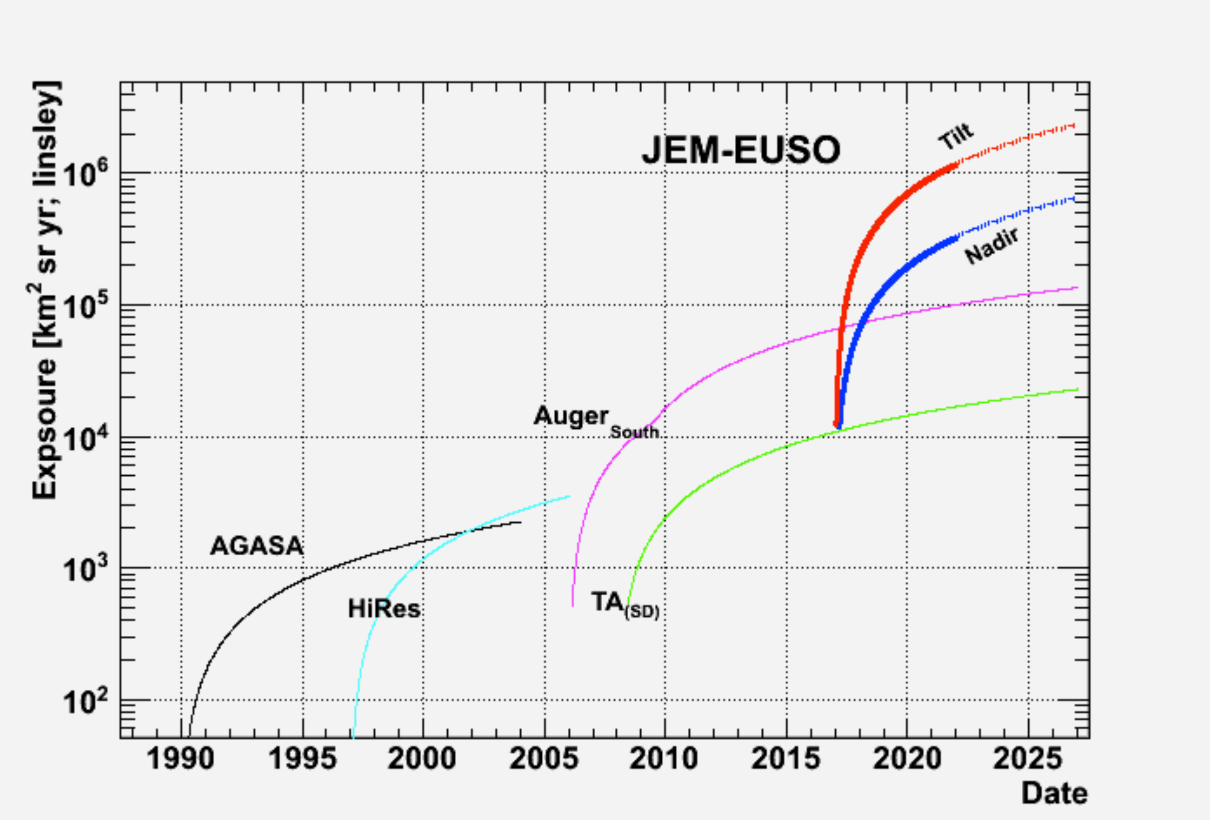

Figure 1.3: Expected cumulative exposure of JEM-EUSO in km^2 sr yr: The thick blue curve corresponds to pure nadir mode and the thick red curve to pure tilted mode; the actual exposure will depend on the final operating mode adopted and will lay between both curves. For comparison, the evolution of exposure by other retired and running EECR observatories is shown. (Image: JEM-EUSO Collaboration, 2010)

the different modes of JEM-EUSO and other UHECR-detectors. A detailed study about the performance of JEM-EUSO can be found in Adams Jr. et al. (2013).

Section 1.2 gives a detailed description of the phenomena that are expected to be observed. The observation principles and requirements are listed in Section 1.3. Section 1.4 lays down the current baseline of the telescope design divided in optics and the focal surface with the detectors and readout electronics.
Chapter 2 gives a more detailed view on the electronics that are relevant for this work. The Section 2.4 is about the Cluster Control Board (CCB) developed in Tübingen. This is the device to which the Simulator of the Photo-Detector Module (PDM-Simulator) sends data. Section 2.3 shows the basic functions of the Photo-Detector Module (PDM).
The design of the PDM-Simulator and the simulation of the test data are described in Chapter 3 followed by an overview of test results (Chapter 4). Chapter 5 consists of conclusions and a short outline for further work in testing the hardware trigger of the CCB.

1.2 Ultra High Energy Cosmic Rays

1.2.1 Cosmic Rays (CR)

Figure 1.4: Picture of Victor Hess and his balloon. (Image by Victor-Franz-Hess-Gesellschaft)

CR have been discovered by Victor Hess in 1912 by using a balloon to measure the radiation in different altitudes (Unsoeld & Baschek, 2002). He concluded from the higher radiation in higher altitudes that the radiation must come from cosmic phenomena. To exclude the sun as possible source he repeated the experiment during an eclipse. Experiments regarding CR were source of important discoveries in particle physics since acceleration experiments were not available at this time. For example in 1932 the positron was discovered by Car David Anderson and in 1937 he discovered together with Seth Henry Neddermeyer the muon. Cecil Frank Powell discovered the pion in 1947.

The composition of CR have been measured by different experiments such as Payload for Antimatter Matter Exploration and Light-nuclei (PAMELA) (Casolino & The Pamela collaboration, 2009) and Cosmic Ray Energetics And Mass (CREAM) (Seo et al., 2004) for energies up to some 100 MeV. CR mainly consist of protons ($\sim 86\,\%$) and helium ($\sim 12\,\%$), but a lot of other particles are present as well, such as electrons ($\sim 2\,\%$), positrons and heavy ions up to actinides (Simpson, 1983). Compared to the solar abundance two things attract attention. There are less protons and Helium than expected. This could be because it is harder to accelerate them than heavier elements. The second feature is a richness of light elements like lithium, beryllium and boron. This can be explained by spallation of heavier nuclei in a collision with other particles in the interstellar medium.

1.2.2 Spectrum of Cosmic Rays (CR)

The energy spectrum of CR extends from 10×10^{8} eV to 10×10^{21} eV, over thirteen magnitudes, with a flux of $10 \times 10^{4}\,\mathrm{m^2sr}$ GeV/s to $10 \times 10^{-28}\,\mathrm{m^2sr}$ GeV/s (see Figure 1.5, 1.6 and Hanlon (2008)).

Between 10×10^{8} eV and 10×10^{10} eV the flux is constant, below 10×10^{8} eV the particles can not enter the planetary system because of the deflection of the interplanetary magnetic field. In this energy regime the flux is correlated to the strength of the solar wind. In case of high solar activity the flux is suppressed due to the magnetic deflection of the particles. During high solar activity the magnetic field of the sun is stronger and extends further.

Above 10×10^{10} eV the flux depends no longer on the activity of the sun and is constant in time. It can be described by a power law: $\frac{dN}{dE} \propto E^{-q}$. E is the energy of the particle and q is the power law index. From 10×10^{10} eV to 3×10^{15} eV the power law is $q = 2.7$ after that there is a change in the spectral index ($q \approx 3.1$), the depletion is stronger. This is the so called *knee* of the spectrum. One possible explanation is a shortage of objects that can accelerate particle to such high energies. At 1×10^{18} eV to 1×10^{19} eV the power law index decreases again ($q \approx 2.5$) because extra-galactic particles with these energies can reach earth. This feature of the spectrum is called *ankle*.

The spectrum extends up to 10×10^{21} eV but due to the low flux of only one event per $1000\,\mathrm{km}^2$ per year the number of measurements is limited and the statistic is low.

For energies below 10×10^{15} eV, supernova remnants are the most likely source of the CR. Shock fronts from a supernova explosion can accelerate particles up to this energy and the predicted spectrum resulting from this process follows a power law and fits the observed spectrum. Unfortunately due to the galactic magnetic field the sources of the particles can not be determinated (see 1.2.3). For energies between 10×10^{15} eV and 10×10^{18} eV the particles are thought to origin in the Milky Way as well, but the sources are unknown. Shock fronts from pulsar wind nebulae are under discussion as a possible source.

Particles with energies above 10×10^{18} eV are called Ultra High Energy Cosmic Ray (UHECR). These particles must origin outside the Milky Way because their gyro radii are larger then the diameter of the Milky Way.

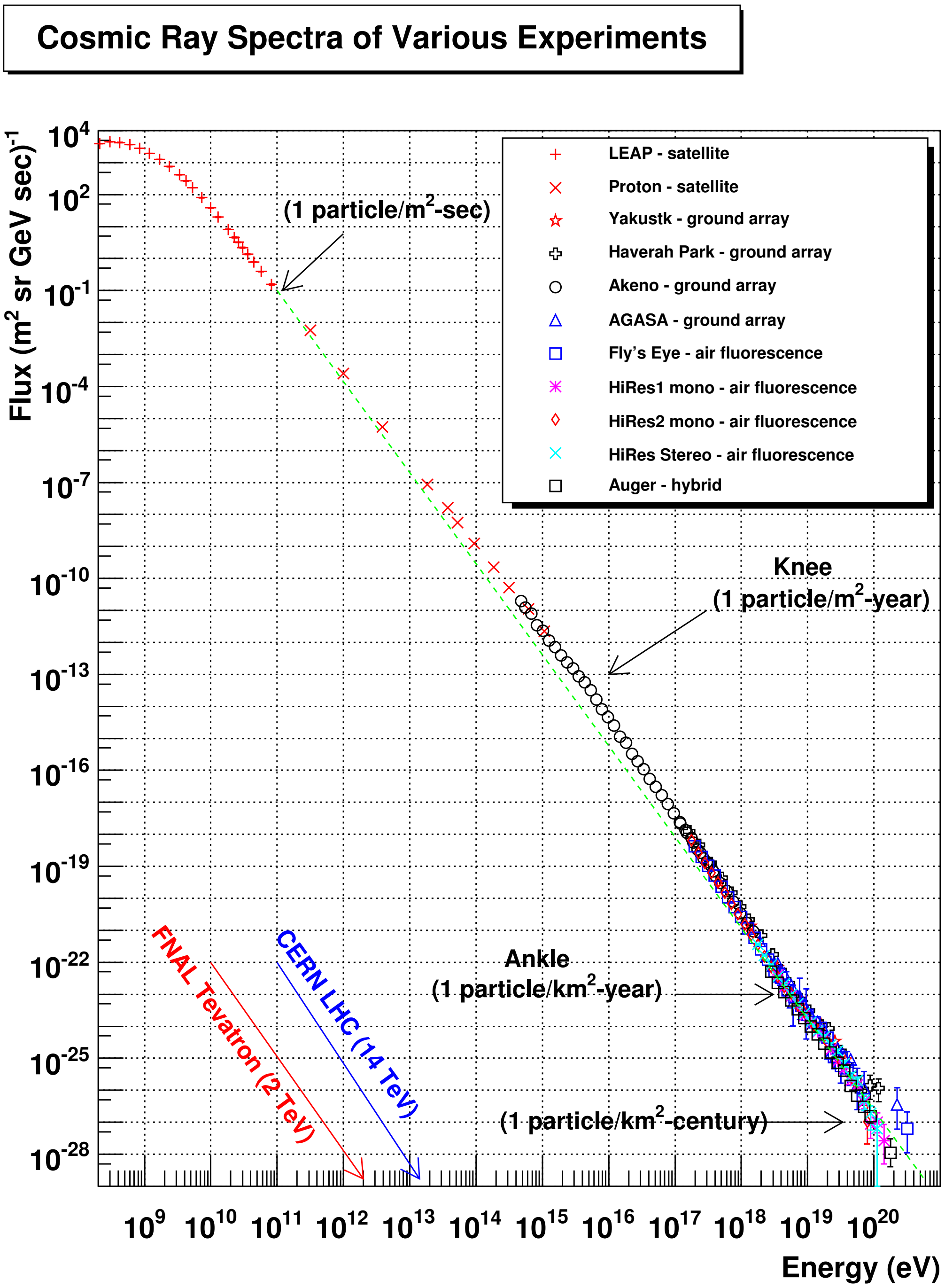

Figure 1.5: Spectrum of UHECR from 1×10^8 eV to 1×10^{21} eV (Image: Hanlon, 2008)

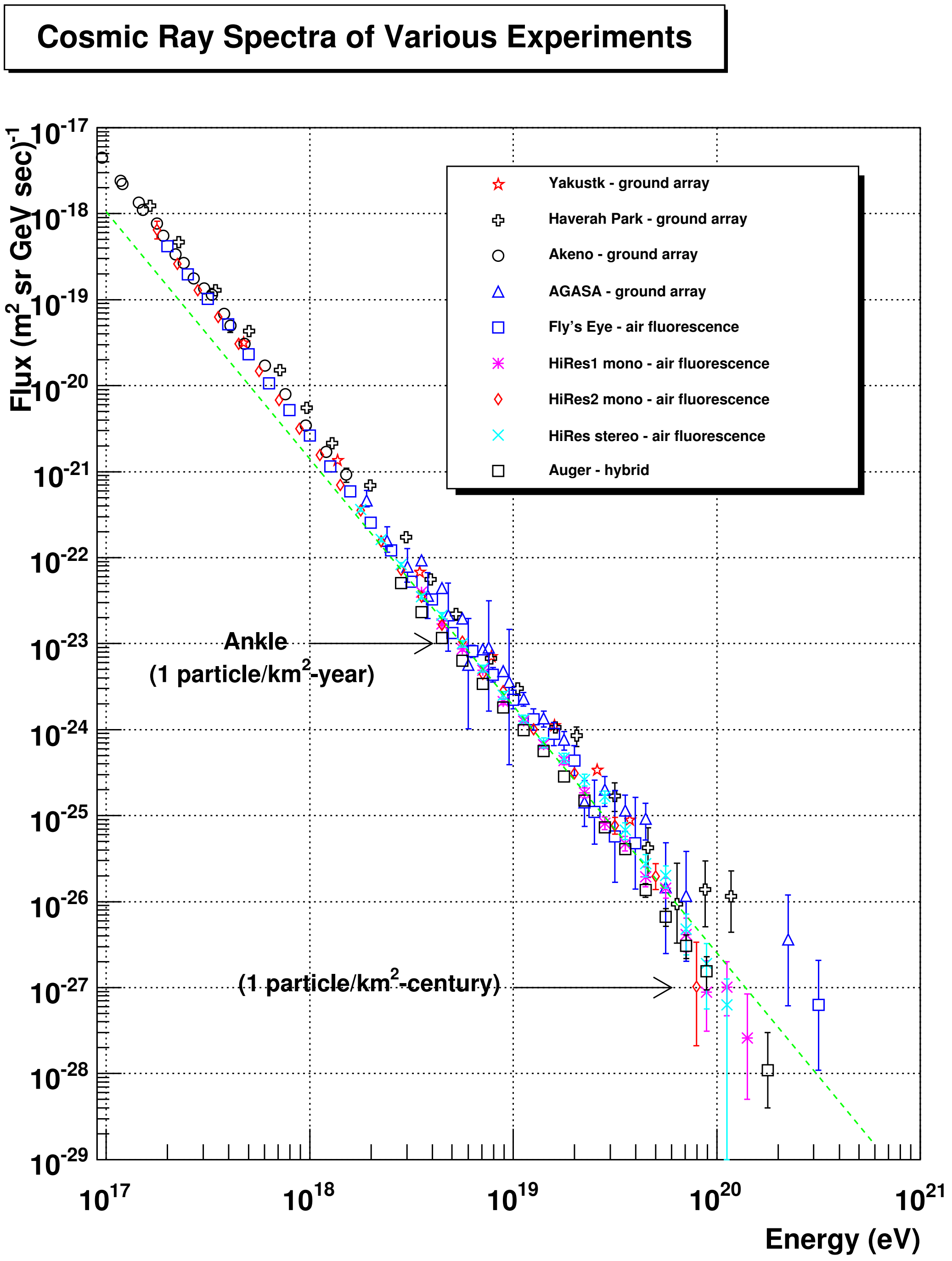

Figure 1.6: Spectrum of UHECR from 1×10^{17} eV to 1×10^{21} eV (Image: Hanlon, 2008)

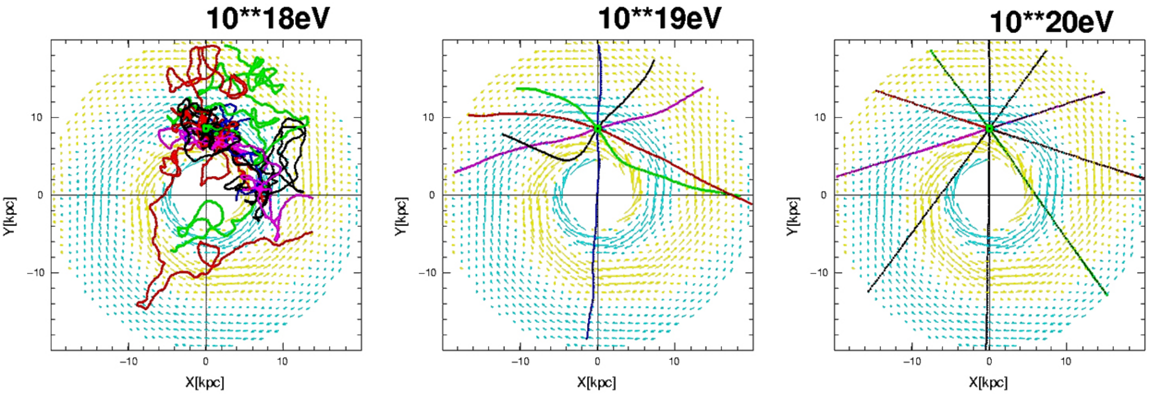

Figure 1.7: Simulated propagation of a proton injected into the galactic magnetic field: The first graphic shows that particles with an energy of 1×10^{18} eV are trapped in the Milky Way. The second graphic shows still a deflection for an energy of 1×10^{19} eV. For higher energies the tracks are almost straight (last graphic with an energy of 1×10^{20} eV). (Image courtesy of: Teshima, http://www.asi.riken.jp/en/laboratories/chieflabs/astro/images/fig2.jpg)

1.2.3 Propagation of UHECR

In the Subsection before it was stated that only particles with energies above 10×10^{18} eV can be considered as extra-galactic particles. Only for these particles the production site can be derived from their trajectories. This is due to the fact that magnetic fields deflect the path of the charged particles. If a charged particle travels through a magnetic field B, the path is altered by the Lorentz force. The resulting bending of its trajectory can be described by the gyro radius r_{Larmor}. This deflection is lower for higher energies E of the particle. The atomic number of the particle Z influences the path as well:

$$r_{\text{Larmor}} = \frac{E}{ZeB} \tag{1.1}$$

If the gyro radius of the particle is small compared to the dimension of a galaxy, it is trapped inside it. If the energy is higher, the gyro radius is larger. For example a particle with the energy of 5×10^{19} eV in an uniform magnetic field of 3 µG (Medina Tanco et al., 1998) has a gyro radius of 19 kpc. This length is comparable to the radius of the galactic magnetic field of 20 kpc (Medina Tanco et al., 1998) so the particle can leave the galaxy and the perturbation of the trajectory is small. As the lower energy threshold of JEM-EUSO is 5.5×10^{19} eV, a identification of the source should be possible. In Figure 1.7 a simulation of particles with different energies injected into a galactic magnetic field is shown.

Shortly after the discovery of the Cosmic Microwave Background (CMB), Greisen (1966), Zatsepin & Kuz'min (1966) published that protons with energies above 5×10^{19} eV could interact

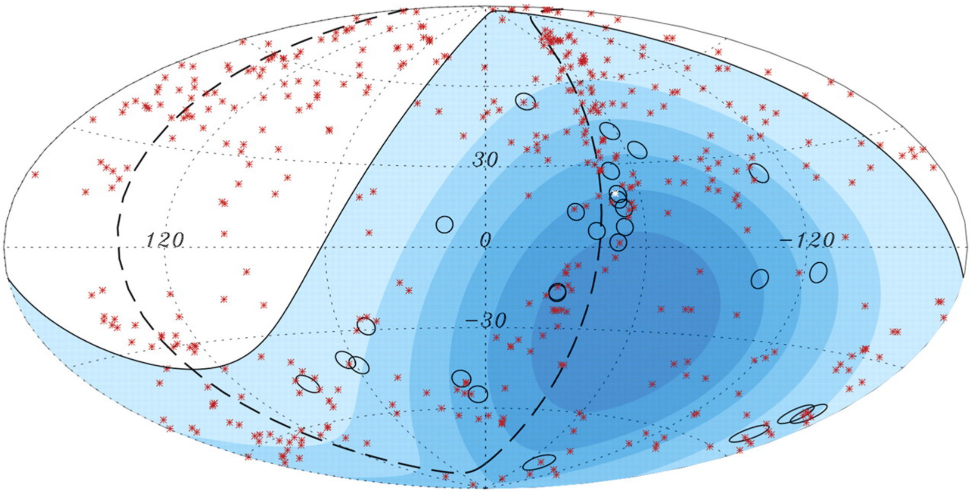

Figure 1.8: Map of events detected by Pierre Auger Observatory: The black circles show events detected by AUGER and the red stars are AGNs. The solid black line is the border of the FoV and the blue colors correlate to the relative exposure time. (Image: Abraham et al., 2008)

with photons of the CMB. The CMB is more or less isotropic with a temperature of 2.7 K. A proton p interacting with a photon γ produces a Delta-baryon Δ^+ that decays into a pion π and a proton p or a neutron n with reduced energy:

$$p + \gamma_{2.7K} = \Delta^+ = p + \pi^0 \tag{1.2}$$

$$p + \gamma_{2.7K} = \Delta^+ = n + \pi^+ \tag{1.3}$$

Sooner or later all particles with an energy above 5×10^{19} eV will lose energy until the energy is below the threshold of 5×10^{19} eV. The cross section of these processes would lead to an effective attenuation length of ~70 Mpc for protons with an energy of 5×10^{19} eV. The combination of the absence of objects that accelerate particle to energies above 1×10^{20} eV and the short attenuation length leads to a cutoff in the spectrum. So far the statistics for this energy range is very low. Nevertheless, the Pierre Auger Observatory (Abbasi et al., 2008) and the High Resolution Fly's Eye Cosmic Ray Detectors (HiRes) (Sokolsky for the HiRes Collaboration, 2010) found evidence of the existence of the GZK-cutoff. JEM-EUSO will improve this situation.

Another less effective reaction is pair production with an energy of about 7 × 10^{17} eV. The cross section for this process is low, in the spectrum a change of the power law index is not noticeable. In this case a proton p hitting a photon γ produces an electron e^-, a positron e^+ and a proton p with less energy:

$$p + \gamma_{2.7K} = p + e^- + e^+ \tag{1.4}$$

1.2.4 Sources of UHECR

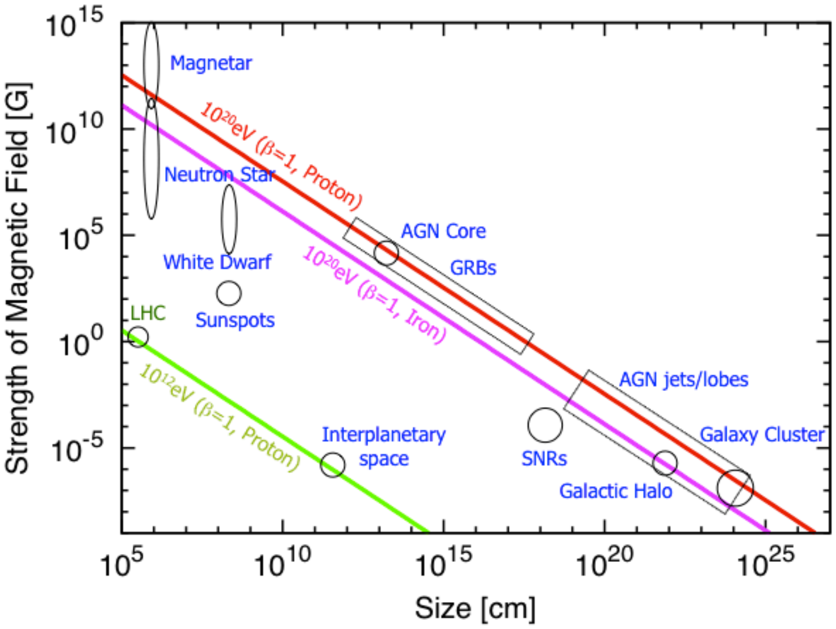

Figure 1.9: Hillas Plot: Possible sources of UHECR are plotted with their size over their magnetic field. The dimension and the strength of the magnetic field define up to which energies a particle is confined to the source due to its gyro radius. The red line is for a proton with the energy of 10^{20} eV, the magenta line for an iron ion and the green line for a proton with the energy of 10^{12} eV. Sources below the diagonal lines are not able to accelerate particles over the corresponding energies. β is the relative velocity of the particle. (Image: JEM-EUSO Collaboration (2010), based on: Hillas (1984))

UHECR are generated in the most extreme conditions known in astrophysics. There are two types of theories trying to explain such very energetic particles, but due to low statistics and poor angular resolution of the so far discovered events, it is hard to discriminate one from the other. The idea of the *bottom up* approach is that macroscopic effects like shock waves or turbulences transfer energy to a microscopic particle. The *top down* model assumes that UHECR are the decay products of Super-Heavy Particles (SHP).

Bottom up

In 1949 Fermi published a paper on the origin of CR. The theory is based on a statistical process of a charged particle encountering a moving cloud multiple times. Irregularities of the magnetic field associated to moving clouds act like magnetic mirrors to keep the particle in the region of the moving clouds. The particle gains energy until it can leave the acceleration region. This process is called *second order Fermi mechanism*, but it is not efficient enough to reach the observed energies of UHECR.

Four different groups postulated the *first order Fermi mechanism* by using shock fronts, irregular magnetic fields and relativistic movement of the particles (Axford et al. (1977), Krymskii (1977), Bell (1978) and Blandford & Ostriker (1978)). Fermi's statistical approach is still a important part of the theory.

Shock fronts are defined by particles moving faster then the speed of sound in a medium. They are present in many astrophysical phenomena like supernova explosions or AGNs. There are two regions present in shock fronts, the upstream region ahead of the shock front and the downstream region behind the shock front. The upstream region is described by the density ρ_1, the temperature T_1 and the impulse p_1, the downstream region by ρ_2, T_2 and p_2. There is also a irregular magnetic field present in both regions. When a particle passes through the shock front, the velocity distribution becomes isotropic. The particles have a gyro radius greater than the shock size.

By choosing the inertial system with the shock front at rest, a relation for the state parameters can be found (Longair, 2011). By using the equation of continuity, the continuity of the energy flux and the continuity of the momentum flux, a relation between v_1 and v_2 can be obtained:

$$v_2 = \frac{1}{4} v_1 \tag{1.5}$$

By changing the reference system to a particle in the upstream region being stationary, a particle in the upstream region will see the downstream region approaching with the velocity of:

$$v_1 - v_2 = U - \frac{1}{4}U = \frac{3}{4}U \tag{1.6}$$

with

$$U = |v_1| \tag{1.7}$$

The particle gains energy and is scattered by the turbulent magnetic field of the downstream region, so the velocity distribution is isotropic. By assuming the reference system is stationary for a particle in the downstream region, the result is the same. The upstream region is also approaching with the velocity of $U = \frac{3}{4}U$ for the particle in the downstream region. The same particle can gain energy again with a certain chance because the velocity distribution of the particles is isotropic. After the shock front it is scattered again by the magnetic field.

To calculate the energy the particle gains through this process, it is assumed that the shock front is non relativistic, but the particle is. The x-axis is perpendicular to the shock and the downstream region is approaching. From the energy of the particle and the particles x-component of the impulse

$$E = pc \tag{1.8}$$

$$p_x = \frac{E}{c} \cos\theta \tag{1.9}$$

the derivate of the energy is obtained:

$$E' = E + \frac{E}{c} \cos\theta \cdot v \tag{1.10}$$

The energy increase after one crossing of the shock front is:

$$\frac{\Delta E}{E} = \frac{v}{c}\theta \tag{1.11}$$

To integrate the equation to obtain the average energy, the probability of a particle crossing the shock within $[\theta, \theta + d\theta]$ has to be multiplied. The probability is $p = 2 \cdot \sin\theta \cdot \cos\theta d\theta$. After integrating from 0 to π over θ the result for the average energy is:

$$\left\langle \frac{\Delta E}{E} \right\rangle = \frac{2}{3}\frac{v}{c} \tag{1.12}$$

For passing through the shock twice the result is:

$$\left\langle \frac{\Delta E}{E} \right\rangle = \frac{4}{3}\frac{v}{c} \tag{1.13}$$

With

$$v = \frac{3}{4}U \tag{1.14}$$

the average energy increase for one cycle is:

$$\left\langle \frac{\Delta E}{E} \right\rangle = \frac{U}{c} \tag{1.15}$$

This process can go on until the energy of the particle is large enough to leave the accelerating region. The gyro radius increases with the energy until the particle is no longer confined to the accelerating region.
In the Hillas plot (Figure 1.9) the strength of the magnetic field is plotted over the size of dif-

ferent astrophysical objects. The lines in the plot represent protons for 10^{20} eV (red) and for 10^{12} eV (green). The third line is for an iron ion with the energy 10^{20} eV (magenta). Possible candidates from where UHECR originate are neutron stars, jets from AGN, gamma-ray bursts, radio galaxies and galactic clusters.

Top down

The second family of models called *top down models* predict annihilation products of SHP with masses between 1×10^{22} eV/c^2 and 1×10^{25} eV/c^2, the so called weakly interacting massive particles (WIMP)s (Aloisio et al., 2006). In theory WIMPs are a relict of the Big Bang. They are expected the be part of the cold dark matter, distributed over the whole Universe with some higher concentration in the center of the galaxy and clouds around galaxies. The UHECR from these sources should be isotropic with a peak near Sagittarius and some outer regions of the Milky Way. In addition, these particles could have energies higher than 1×10^{20} eV.

So far the Pierre Auger Observatory, an experiment to detect UHECR with similar energies as JEM-EUSO, found an anisotropy for UHECR (Abraham et al., 2008). This indicated that the *bottom up* theory might be correct. The collaboration published a sky map of the detected UHECR overlayed with the positions of about 200 AGNs. Although they found an anisotropy they do not claim that the UHECRs come from any of the AGNs (see Figure 1.8). A similar study with HiRes data, another UHECR observatory, found no anisotropic distribution (Abbasi et al., 2010). The statistics are low and the upper limit for the energy is not as high as for JEM-EUSO, so JEM-EUSO will provide a better understanding of this anisotropy and of the structure of the GZK cutoff.

1.2.5 Observation of UHECR

CR with energies below 10^{11} eV can be measured directly in space (e.g. PAMELA) or with a balloon (e.g. CREAM) using for example a scintillation counter. A CR entering a scintillator generates photons in the material that can be measured. Compared to particles with lower energies the flux of UHECR is much smaller. Measuring UHECR directly with a satellite or a balloon experiment is impossible. But a UHECR entering the atmosphere interacts with an air nuclei. The result are Extensive Air Showers (EAS) that can be observed.

The primary particle, which can be a high energy γ-photon, a neutrino, a proton or a heavier nucleon, interacts with an air nucleos and generates many instable hadrons. These particles again interact with other nuclei and produce more hadrons until their individual energy is below the threshold energy for the relevant process. This component of the shower is called hadronic component. The hadrons decay in muons, electrons and photons. The photons have such high energies that they produce electron positron pairs. The generated electrons ionize mainly nitrogen and generate photons in the UV-band (300 nm to 400 nm). This light is called fluorescence light. In the direction of flight of the particles, Cherenkov light is produced as well, because the charged particles move with a speed greater then the phase velocity of light for the atmosphere. Three different types of detectors observe the different components of the process. Cherenkov telescopes look for the Cherenkov light to detect primary photons with energies of 10 GeV to 10 TeV (Hinton, 2004). A prominent example is the HESS telescope in Namibia.

For higher energies particle detectors are used to detect the muonic component of EAS. The detectors consist of water tanks and photo-multipliers. The muons enter the water tanks and produce Cherenkov light proportional to their energy. These detectors have to be spread over a big area to compensate for the low flux. For example the Auger Observatory uses 1600 water tanks spread over 3000 km^2.

The last type of detectors are called Fluorescence Telescopes. This time the fluorescence light in the atmosphere is observed directly using telescopes with large mirrors. Again the Auger Observatory is the state of the art example of this kind of telescope. In total 27 telescopes observe at the same site as the particle detectors. This makes cross calibration possible (Matthiae, 2008). The disadvantage of these telescopes is that observations are only possible in clear and moon-less nights.

1.2.6 Observation of UHECR by JEM-EUSO

The approach of JEM-EUSO is to observe fluorescence light from space. An EAS appears to be a light spot moving through the atmosphere at the speed of light. The intensity of the shower increases until a maximum is reached and then decreases again. At the end of an event the Cherenkov light reflected from ground is visible in case of ideal weather and ground conditions (Figure 1.10).

From the height of the maximum of the shower an energy reconstruction is possible. Two techniques were developed by using simulated events. Either the height can be determinated by analyzing the shape of the shower or by using the difference between the Cherenkov peak and the maximum of the shower. The shape of a shower depends on the density of the atmosphere. To figure out the direction of the shower, the geometry of the shower has to be reconstructed from the recorded data. The geometry of the detector and the characteristic of the optics are used to reconstruct the shower. Several reconstruction algorithms are under investigation, using different approaches (Mernik, 2009).

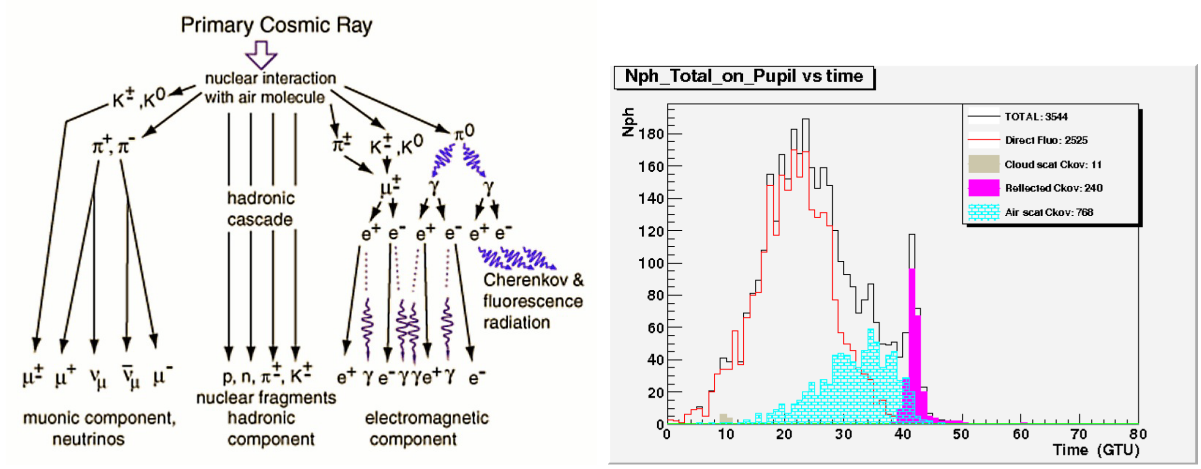

Figure 1.10: Left: Particle production in the atmosphere, fluorescence light in purple (Image: Tenzer & Santangelo, 2013)
Right: Shower profile with different components (Image: JEM-EUSO Collaboration, 2010)

1.2.7 Neutrinos

The cross section of neutrinos increase with the energy of the neutrino. Compared to solar neutrinos with energies of $\sim 7\,\mathrm{eV}$ the cross section is eight to nine magnitudes larger (Lemoine & Sigl, 2001). In the atmosphere neutrinos generate comparable EAS to protons. These EAS can not be distinguished from EAS produced by protons, so other evidences than the character of the tracks have to be used. The cross section of neutrinos is still smaller than for protons. It is much more likely that neutrinos interact deeper in the atmosphere or even after they passed through earth. Deeper showers or showers going upwards are an evidence for neutrinos. At the moment further studies about the angular reconstruction and trigger rate for these events are taking place at the IAAT. A diploma thesis on ultra high energy neutrinos was written in Tübingen by Bittermann (2010).

1.3 Requirements

The JEM-EUSO collaboration defined a list of requirements in the *report on the Phase A study* (JEM-EUSO Collaboration, 2010) to achieve the JEM-EUSO scientific goals. The science requirements are to observe UHECRs and to specify their arrival direction and energy, to determinate the trans-GZK structure in the cosmic ray spectrum, to discriminate among nuclei, gamma rays and neutrinos and to observe TLE. In the Observation requirements the performance necessary to achieve the Science requirements is defined. From the Observation requirements a number of requirements for the instruments are abstracted:

Science requirements:

SR1: Specification of UHECR origin by arrival direction analysis with determination accuracy better than a few degrees

SR2: Determination of trans-GZK structure in cosmic ray energy spectrum

SR3: UHECR primary identification capability: discriminating among nucleus, gamma ray and neutrino

SR4: Observation capability of TLEs

Observation requirements:

OR1: Observation area: $\geq 1.3 \times 10^5 \left(\frac{H_{orbit}}{400\,\mathrm{km}^2}\right)\,\mathrm{km}^2$

OR2: Arrival direction determination accuracy: $\leq 2.5°$ for E = 10^{20} eV and 60° zenith angle

OR3: Energy determination accuracy: $\leq$30% for E = 10^{20} eV and 60° zenith angle

OR4: X_{max} determination accuracy: $120\,\mathrm{g\,cm^{-2}}$ for E = 10^{20} eV and 60° zenith angle

OR5: Energy threshold: 5.5×10^{19} eV

OR6: Monitoring the average signal rate for all pixel every 3.5 s

OR7: Capability of observing TLE with time scales shorter than 1 s

Instrument requirements:

IR1: Observation duty cycle: $\geq$ 17%

IR2: Instrumental duty cycle: $\geq$ 30%

IR3: Lifetime: longer than 3 years. Two-year storage is also required before launch.

IR4: Altitude determination accuracy: $\leq 0.05°$ (this may be achieved by offline analysis)

IR5: Precision of absolute time: $\leq$ 1 µs

IR6: Wavelength band: covering major fluorescence lines in 330 nm to 400 nm

IR7: Synchronization precision among subsystems: $\leq$ 200 ns

IR8: Light shielding against stray light: less than GHz level

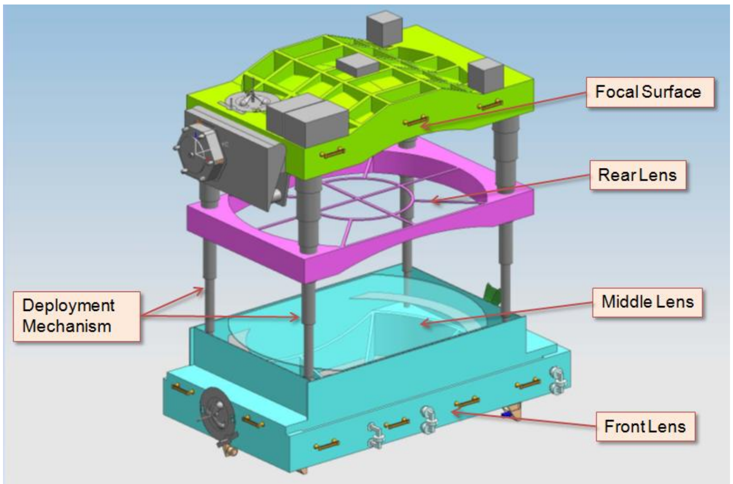

Figure 1.11: Scheme of the optical system including the extension mechanism (Image: JEM-EUSO Collaboration, 2010)

1.4 Overview of the Telescope

JEM-EUSO is a UV-telescope with a refractive optics composed of two Fresnel lenses and a diffracting precision lens with a FoV of ±30°. The focal surface is consists of 64 pixels MAPMT, for a total pixel count of 0.3 M. The result is a spatial resolution of 1°.

In addition, JEM-EUSO also has a Light Detection and Ranging unit (LIDAR) and an infrared camera (IR-camera) on board to evaluate the weather conditions, for example the altitude of clouds. This has a big impact on the profile of a shower since the reflected Cherenkov radiation and some parts of the fluorescence radiation can get scattered due to clouds.

The next two subsections will give a more detailed view on the optics and the focal surface. The focal surface electronics is presented in Chapter 2. The optical system is not circular, but cut at the sides. This is due to the limitations in weight and space available in the H-II Transfer Vehicle (HTV) which is planned to be used for transport. The telescope measures about 4 m in height compared to 2 m available in the HTV so an extension system is needed. This might lead to not properly aligned lenses, therefore a calibration system is required to reduce this problem.

1.4.1 Optics

The current baseline is using three lenses composed of Poly-Methyl-Methacrylate (PMMA). This design is called *PPP 2010*. The material is widely used in space missions and has a known behavior under extreme conditions. In addition, another design is under investigation. It is called *CPP 2010* In this case the front lens would be composed of Amorphous Peffluoro

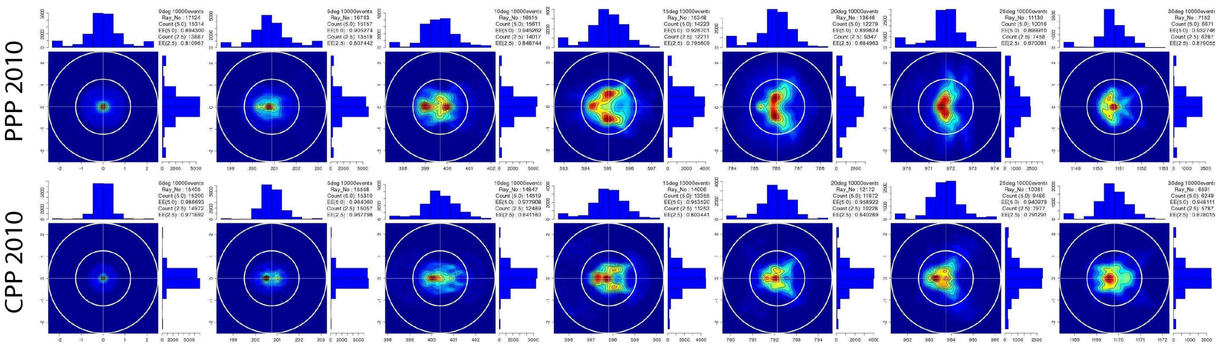

Figure 1.12: Spot size of the *PPP 2010* design (top) compared to the *CPP 2010* design (bottom (Image: JEM-EUSO Collaboration, 2010)

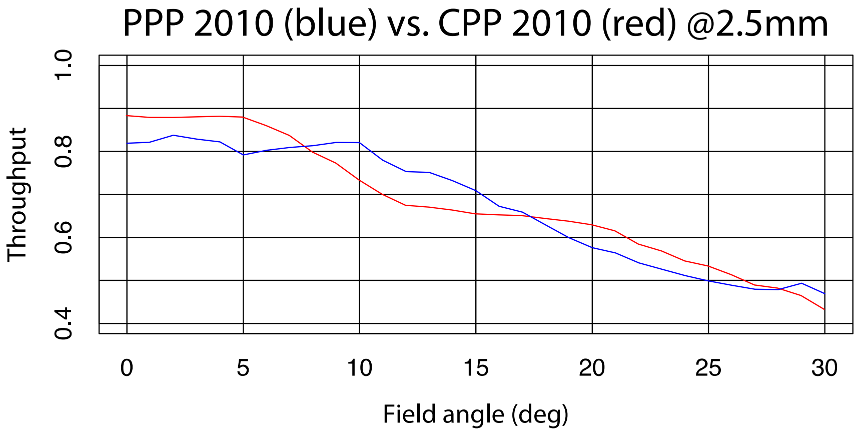

Figure 1.13: Throughput of the different designs: *PPP 2010* design (blue) compared to the *CPP 2010* design (red) (Image: JEM-EUSO Collaboration, 2010)

Alkenylvinylether (CYTOP) and the other two lenses would be composed of PMMA. The performance of this design is better, but there is no operating experience in space. At the moment, studies regarding the aging of the material under space conditions are ongoing. A major disadvantage is that CYTOP is twice as heavy as PMMA. This leads to higher weight of the advanced design. A comparison between the two designs is shown in Figures 1.12 and 1.13. The PMMA design is more likely to become the final design since the performance is sufficient and there are less uncertainties.

The main requirement on the optics is a small spot size and a high throughput of UV light. In addition, the optics have to keep the same behavior over the entire lifetime of the telescope.

1.4.2 Focal Surface

The focal surface is a section of a sphere with a radius of 2505 mm, consisting of 4932 MAPMTs with 64 pixels each and a total pixel number of 315 648. Four MAPMTs together are called Elementary Cell (EC), each with 256 pixel controlled by one Application Specific Integrated

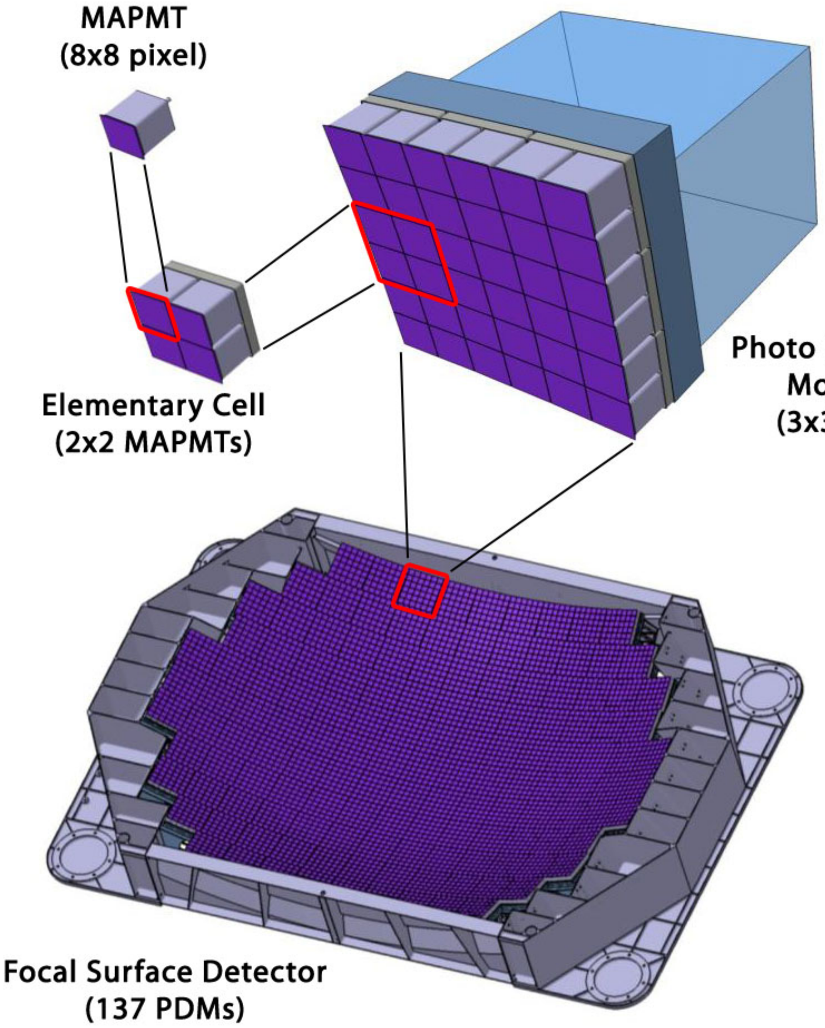

Figure 1.14: Scheme of the focal surface: On the bottom the entire focal surface is shown with 137 PDMs. One PDM is enlarged showing the structure of the nine ECs. One EC consists of four MAPMTs with 64 pixels each. (Image: JEM-EUSO Collaboration, 2010)

Circuit (ASIC). A combination of three ASIC-boards with three ASICs each is called Photo-Detector Module (PDM). In total 137 PDMs are necessary for the focal surface. Each PDM has a FPGA-board to control the ASICs, do a first analysis of the data and to provide a trigger algorithm. Figure 1.14 is a scheme of the focal surface. Chapter 2 will give a more detailed view on the functionality of the PDM board and the CCB. The CCB handles up to 8 PDM boards and carries a second level trigger.
The requirements for the focal surface detector are (JEM-EUSO Collaboration, 2010):

FSR1: Shape of focal surface: spherical

FSR2: Photon detection efficiency: ≥ 0.12

FSR3: Average pixel size: $\leq 4.5\,\mathrm{mm}$

FSR4: Focal surface area: $\geq 3.6\,\mathrm{m}^2$

FSR5: Trigger efficiency: ≥ 0.95 ($E_0 = 10^{20}\,\mathrm{eV}$, zenith angle 60°; FOV center)

FSR6: Gate time: $\leq 2.5\,\mu\mathrm{s}$

FSR7: Dynamic range: 200 photons within a gate time (TBC)

FSR8: Dead time: $\leq 3\,\%$ (including trigger and readout)

FSR9: Dark current noise: $\leq 50\,\mathrm{GHz}$

FSR10: Acquisition rate of slow mode data: 3.5 s

FSR11: Attenuation of gain for high photon intensity mode <0.001

FSR12: Slow data acquisition mode with 1 ms gate time (TBC)

The focal surface detectors have to be fast and need single photon detection capability because of the light spot moving nearly with the speed of light and the low number of generated photons. Also a slow data acquisition is very important to detect slower events like TLEs or comets. For this, an intelligent hardware is needed to switch between the modes of data acquisition. The amount of data collected is very high, a continuous readout would result in more then 100 GB/s, so the electronics must provide a trigger algorithm to reduce the data significantly. However, due to the small number of events none of them should get lost, so the trigger efficiency has to be better then 95 %.

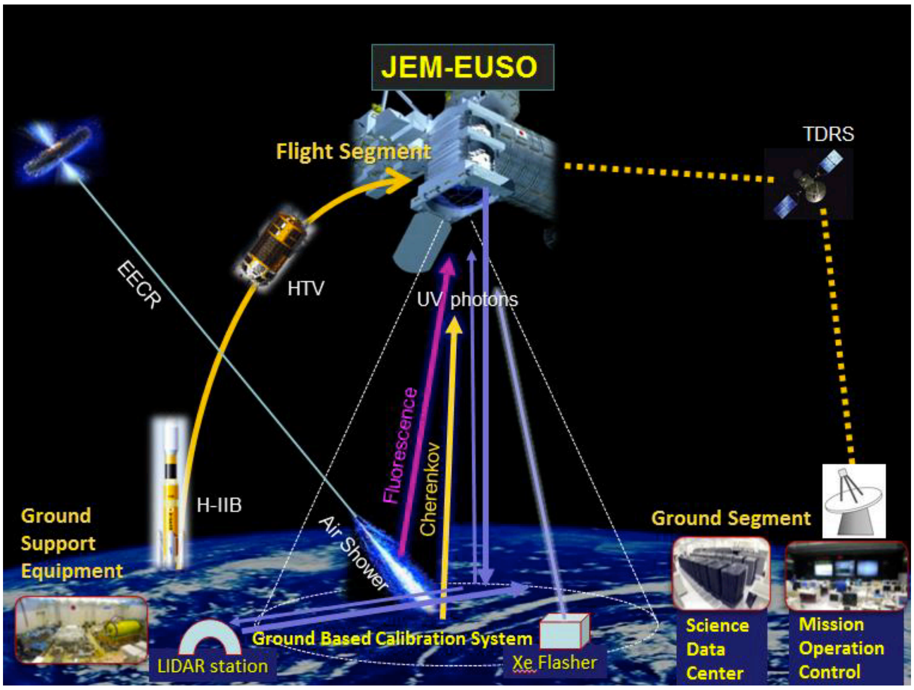

Figure 1.15: Summary of the JEM-EUSO mission components: JEM-EUSO is transported to the ISS using the HTV. For communication the TDRS is used. Several ground bases calibration systems can send laser flashes to the telescope. (JEM-EUSO Collaboration, 2010)

1.5 Status of JEM-EUSO and field tests

JEM-EUSO is currently in Phase B. Prototypes of the hardware have been produced and are tested at the moment. A prototype of the CCB is currently tested at the IAAT using the PDM-Simulator described in Chapter 3. In addition, a second prototype with a smaller FPGA for only one PDM has been manufactured for two field test scheduled for the next two years. In April 2013 a test setup is placed on the site of Telescope Array (TA), the successor of HiRes. For 2014 a balloon campaign is scheduled in Canada.

The start of JEM-EUSO is scheduled for 2017 using an H-IIB rocket of the Japan Aerospace Exploration Agency (JAXA). The telescope will be transported to the ISS with an H2 transfer vehicle and will be moved to the Japanese Experiment Module (JEM) Remote Manipulator System using the Space Station Remote Manipulator System. From there JEM-EUSO is deployed to the Exposed Facility Unit Port number two of the Japanese Experiment Module. The telescope will operate for three years, two additional years are possible if the scientific goals are reached and if additional funding is possible.

After sunset the cover of the telescope is opened and operation begins until sunrise. To secure the telescope the cover is closed again. Several ground based calibration systems will send laser flashes to the telescope to support the onboard calibration system. During daylight the telescope

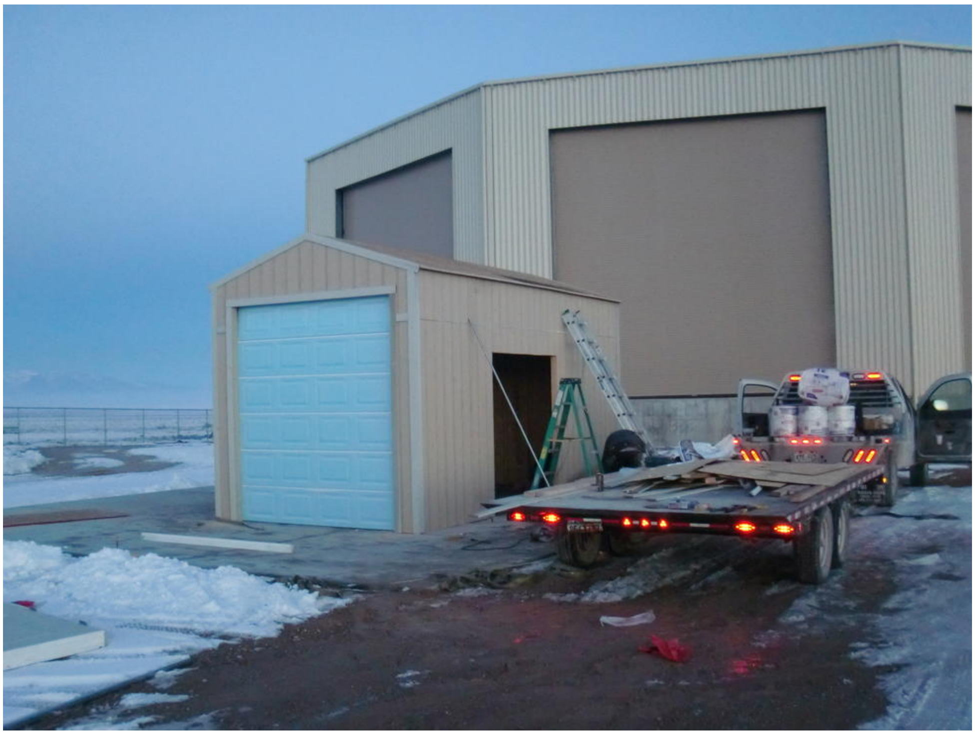

Figure 1.16: Picture of the EUSO-TA site: In the background the building of one fluorescence telescope can be seen. The hut in front of the picture will be the home of the EUSO-TA setup.

is calibrated and the stored data is send down to Earth using the Tracking and Data Relay Satellite System of the National Aeronautics and Space Administration (NASA). During the observations the atmospheric monitoring system records the current weather conditions for the energy and angular reconstruction.

1.5.1 prototype of the JEM-EUSO hardware on the site of Telescope Array (EUSO-TA)

For EUSO-TA a setup including one PDM, one CCB, the MDP and a two lens optic is planned to be deployed on the site of TA. From November 2012 to January 2013 three integration sessions took place to test and integrate the hardware components. TA features a calibration laser that sends a UV laser beam through the field of view. It is planned to use the laser for tests of the hardware and for calibrations. The fluorescence telescopes of TA have a trigger for EAS. The mean trigger rate is about 3 Hz (Tameda et al., 2009). To record EAS with the EUSO-TA setup the TA trigger is used as an external trigger for the setup. Currently the integrations are completed and the campaign is scheduled for April 2013 (see Figure 1.16).

1.5.2 Extreme Universe Space Observatory Balloon experiment (EUSO-Balloon)

For the EUSO-Balloon the same setup as for EUSO-TA is deployed in a balloon gondola. It is a helium balloon with an operation height of $\sim 40\,\text{km}$. The setup is facing downwards, observing the atmosphere below the balloon. The internal trigger algorithm are going to be tested during this flight. The flight is scheduled to be taking place in Canada in 2014. The project reached Phase C/D in January 2013, the production of the hardware started recently. During 2014 the integration of the hardware is taking place.

2 Focal surface electronics

2.1 Overview

This chapter will present the Focal Surface (FS) electronics of JEM-EUSO, the trigger algorithm and the interfaces between the CCB and the PDM boards. Figure 2.1 shows the structure of the electronics used for JEM-EUSO. The design follows a hierarchical approach, the amount of data is reduced by the levels of the electronic by using different trigger algorithms. The trigger algorithms are described in Section 2.3 and 2.4.

The data produced by the multi anode photomultiplier tubess (MAPMTs) have to be reduced from about 130 Gb/s to the available downstream to Earth. According to *Overview of the JEM-EUSO Instruments* of the JEM-EUSO Collaboration (2012), only 297 kb/s are available. However, as the telescope can not acquire data during daylight or full moon, by storing the events during observation and sending them to Earth during these down times, a virtual downstream of about 900 kb/s can be achieved using an on board mass storage device.

The data reduction will be achieved by using different trigger algorithms. Table 2.1 shows the mean trigger rate for the different electronic levels defined in the Phase A report. These trigger rates might change during the development. The data rate is reduced step by step until only one event every 10 s ($\sim$ 500 kb/s) for the entire FS is recorded and sent to Earth.

Table 2.1: Estimated mean trigger rates for each trigger level compared to the expected rate of CR events. (Bayer (2011a) and JEM-EUSO Collaboration (2010), modified)

Level		trigger rate at PDM level in Hz	trigger rate at FS level in Hz
L1	Photon trigger	9.2×10^{8}	1.4×10^{11}
	Counting trigger	7.1×10^{5}	1.1×10^{8}
	Persistency trigger	7.0	1.0×10^{3}
L2	Linear Track Trigger	6.7×10^{-4}	0.1
Expected rate of CR events		6.7×10^{-6}	1.0×10^{-3}

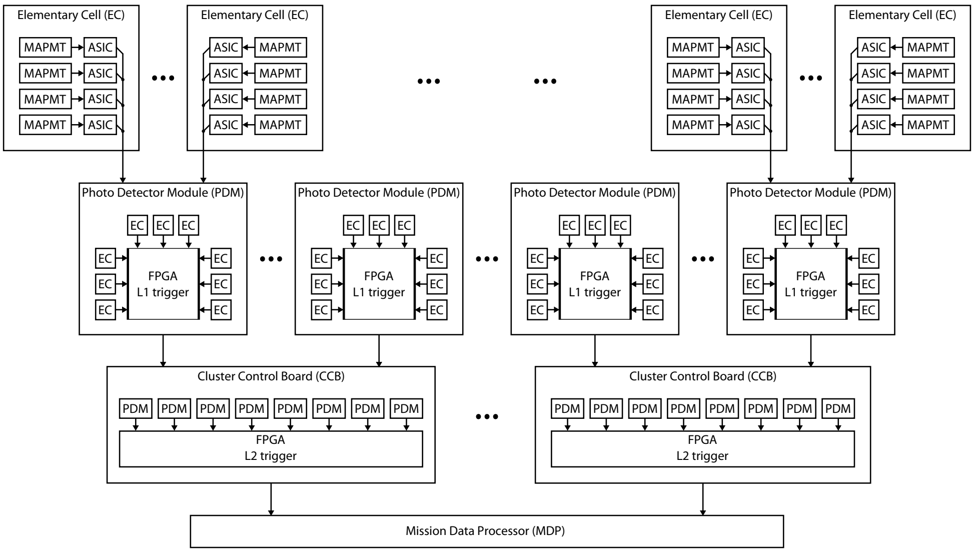
Elementary Cell (EC)
MAPMT
ASIC
Photo Detector Module (PDM)
EC
FPGA
L1 trigger
Cluster Control Board (CCB)
PDM
FPGA
L2 trigger
Mission Data Processor (MDP)

Figure 2.1: Overview of the trigger and read-out scheme: The focal surface electronics are arranged hierarchically. Every level reduces the data by using filtering trigger algorithms. The L1 trigger is implemented at ASIC and PDM level and the L2 trigger is implemented on the CCB. For more details see Bayer (2011a).

2.2 Elementary Cell

2.2.1 Overview

The Elementary Cell (EC) consists of four MAPMTs and four SPACIROC ASICs. This is the first level of focal surface electronics. The purpose is to detect the photons with the MAPMTs and to convert the analog signals into digital signals with the SPACIROC. Subsection 2.2.2 describes the functionality of the MAPMTs and Subsection 2.2.3 the ASICs. The EC interface to the PDM-board is a parallel interface for data for every ASIC and the PDM-board can write to shift registers of the SPACIROC for configuration.

2.2.2 Multi anode photomultiplier tubes

The observation technique and the small number of photons reaching the focal surface[1] requires a detector with a high quantum efficiency. Most commonly used to match this requirement are MAPMTs. Only light in the near UV band from 330 nm to 400 nm is a signal from fluorescence or Cherenkov light from the EAS. All other photons are background and are filtered by a UV-glass entrance filter. The remaining photons hit the photo-cathode specifically designed to emit electrons when a photon hits the surface. These electrons are accelerated to a series of metal

[1] ∼ 5000 photons reach the focal surface for a standard shower with 1×10^{20} eV and 60° (see *The ESAF Simulation Framework for the JEM-EUSO Mission* JEM-EUSO Collaboration (2012)).

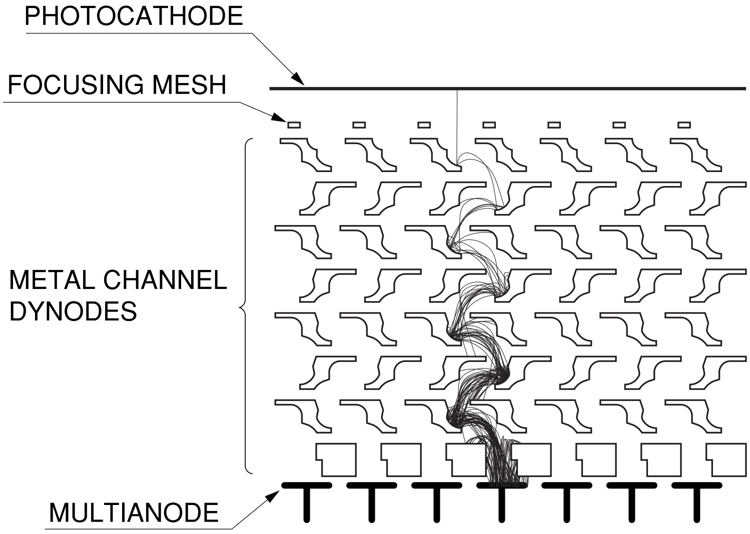

Figure 2.2: Schematic of a MAPMT: Seven channels and seven dynodes of a MAPMT shown. Coming from the top a photon hits the photo-cathode and generates an electron. The electron is accelerated to the first dynode and a cascade of secondary electrons is generated. The dynode's surface focuses the electrons to the next dynode to avoid cross talk between different channels. (Hamamatsu Photonics, 2006)

channel dynodes stacked in a vacuum case. Every MAPMT has 8 × 8 channels, 64 in total. The channels consist of twelve dynodes with an increasing voltage down to the anode where the charge is collected. Every dynode produces more secondary electrons. The increasing voltage is achieved by using tapered voltage dividers. Figure 2.2 shows a schematic of a MAPMT with seven channels and seven dynodes. To reduce cross talking between the different channels the dynode's surface focuses the electrons to the next dynode. Otherwise, if an electron hits a dynode of another channel, a cascade of electrons is produced as well and leads to false signals. The MAPMTs in the current baseline have been developed by RIKEN and Hamamatsu Photonics and have a cross-talking of about 1 %. For a high voltage power supply of 0.9 kV the gain is of the order 10^6.

Compared to the original ESA EUSO design the quantum efficiency of the MAPMTs is increased by a factor of 1.75 and the sensitive area is 1.9 times larger. The quantum efficiency lies between 35 % and 40 %. The sensitive area is 23.04 mm × 23.04 mm compared to the physical dimension of 26.2 mm × 26.2 mm (see Figure 2.4). For the entire set of specifications see JEM-EUSO Collaboration (2010).

Figure 2.6 shows the PDM board on top of the structure for the MAPMT. The MAPMTs are mounted at the bottom of the structure, in this case facing the table. The three horizontal frames in the space between the PDM board and the structure for the MAPMT are the casing for the EC boards. The ASICs are connected directly to the MAPMT on the same board (see Figure 2.5).

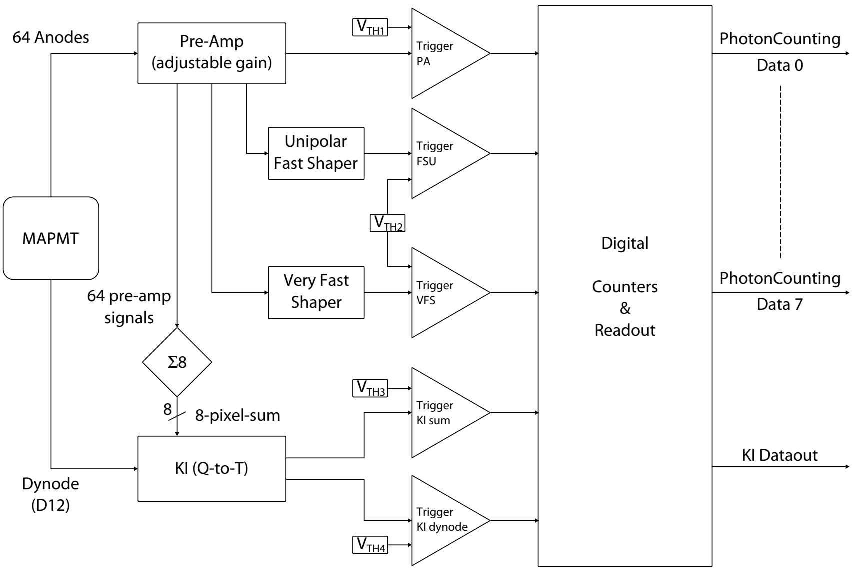

Figure 2.3: Block diagram of the SPACIROC. After a preamplifier for every analog signal a digital signal is generated by using different trigger units. The analog signal is compared to a threshold voltage. If the signal exceeds the threshold voltage a digital signal is generated that increases the photon counters. The KI-dataout is a charge over time value generated in case of too much light for single photon counting. (Ahmad et al., 2010)

2.2.3 Spatial Photomultiplier Array Counting and Integrating Read Out Chip

The ASIC used for JEM-EUSO is called Spatial Photomultiplier Array Counting and Integrating Read Out Chip (SPACIROC) developed at the Laboratoire de l'Accélérateur Linéaire (LAL) in France.

The signals generated by the 64 anodes of the MAPMTs are preamplified and triggered by three different trigger components. The first trigger is comparing the signal to a threshold voltage and increases the photon counter if the signal exceeds the threshold. In addition two shaper units and trigger units are used. The shapers add additional gain to the preamplified signals before they are discriminated by the trigger units. Currently all three can be used, but in the end, after measuring the performance and power consumption for the trigger paths, only one will be in the final design.

In addition, a charge to time conversion by measuring the signal duration above a fixed threshold is taking place. After the preamplification eight channels are summed up. This is done using special integration circuits (KI) and leads to eight channels of KI-data. A ninth channel is generated by using directly the signals of the 12th dynode of the MAPMT. The KI data is an indicator of strong light for the MAPMT. It is used to protect the electronics by switching to the *slow mode* and lower gains (see Figure 2.3).

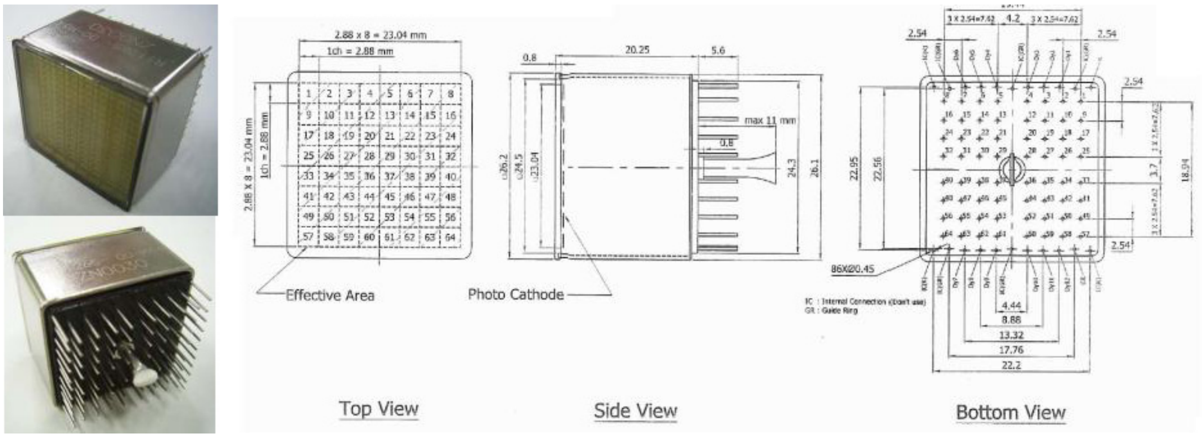

Figure 2.4: Schematic of the MAPMT: The upper picture on the left shows the sensitive area with the filter to constrain the wavelength of the approaching photons to the UV band. In the bottom left picture the connectors to the ASICs are shown. The schematics on the right hand side from left to right show the top view with the pixel map, the side view and the bottom view with the connectors. (JEM-EUSO Collaboration, 2010)

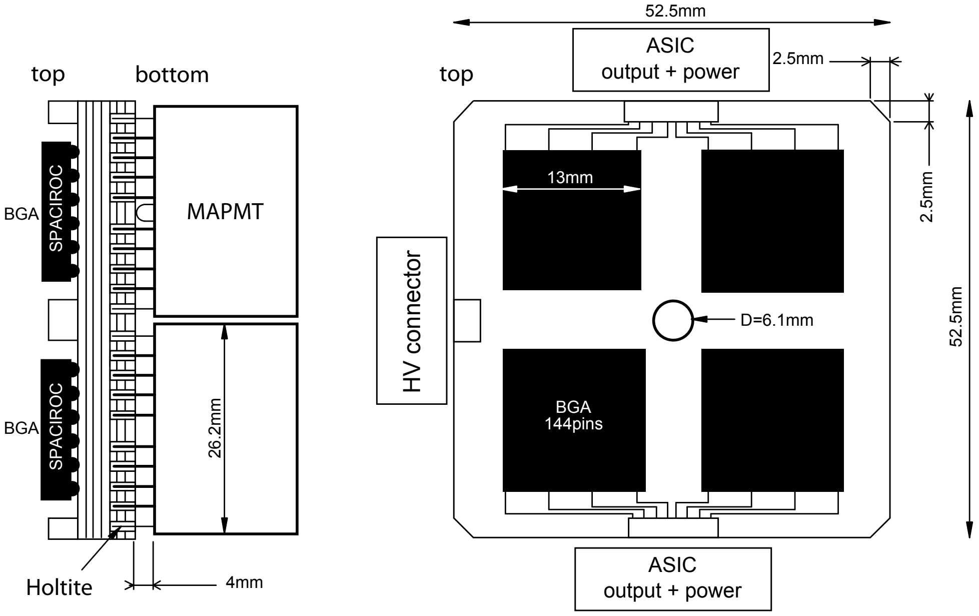

Figure 2.5: Connection between MAPMTs and SPACIROC: From the bottom side the MAPMTs are plugged in the board, from top the SPACIROC are soldered to the board using ball grids. On the right, a scheme of the dimensions and the positions of the connectors are shown. (JEM-EUSO Collaboration, 2010)

2.3 Photo Detector Module

2.3.1 Overview

The Photo-Detector Module (PDM) is divided in two parts, an FPGA board called PDM-board and the Elementary Cell (EC). The EC is described in Section 2.2. The PDM-board is a contribution of the Korean Ewha Womans University. The current prototype uses a Virtex 6 FX240 manufactured by Xilinx. It has three connectors to the three EC-boards and a connector to the CCB (see Figure 2.6). The main task is to provide a Level 1 (L1) trigger, to read out and control the SPACIROC and to decide when to switch from the *fast mode* for EAS to a *slow mode* for TLE comets and other slow atmospheric events. The major task of this work was to develop a simulator board with comparable functions but using simulated data transmitted using an Universal Serial Bus (USB).

Subsection 2.3.2 gives a description of the current baseline for the L1 trigger. An L1 trigger signal is also provided by the PDM-Simulator (see Chapter 3). It is not an actual implementation on the board, but by adding a command to the data sent by the Personal Computer (PC) the PDM-Simulator issues a L1 trigger with trigger seed information at the right time.

The PDM-board gathers data from 36 MAPMTs and controls 36 SPACIROC ASICs. In total, JEM-EUSO needs 137 PDM-boards for the entire focal surface to control all 315 648 channels separately. Every PDM-board controls 2304 channels. For the entire focal surface the PDM-board reduces the trigger rate from 1.4×10^{11} Hz to 1.0×10^{3} Hz. This is the first step to reduce the recorded data to the available down stream to Earth of 297 kb/s.

Subsection 2.3.3 describes the interface between the CCB and the PDM-board. This interface is reproduced by the PDM-Simulator. The PDM-Simulator behave exactly in the same way as the PDM-board for the current baseline. In Subsection 3.6.4 the implementation of the interface for the PDM-Simulator is shown.

2.3.2 First level Trigger

The trigger algorithms are designed to perform a general reduction of the recorded data through e.g. pattern recognition in order to exclude background or events that are not part considered to by EASs or TLEs in the atmosphere, for example cities moving through the field of view.

The first level begins with the single photon counting and checks if a signal exceeds a threshold defined for the background for a predefined time. The L1 trigger is reducing the trigger rate from 1.4×10^{11} counted photons per second for the focal surface to 1.0×10^{3} persistent events per second for further analyses. According to the Phase A study (JEM-EUSO Collaboration, 2010) the L1 trigger is divided into three sub levels.

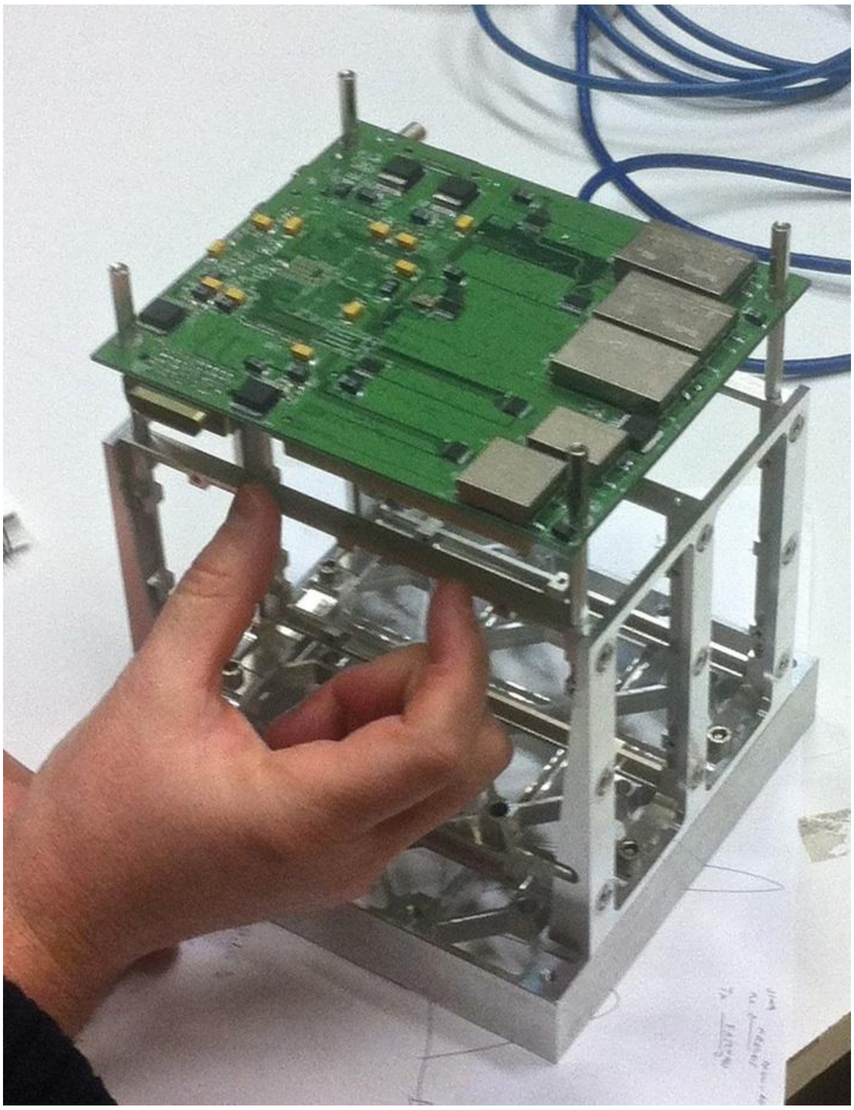

Figure 2.6: Picture of the PDM-board in Orsay during the integration session for TA-EUSO. The PDM board is mounted on top of the structure, on the bottom the MAPMTs will be placed facing the table. Between these two planes the chassis for the EC boards are shown. (image by Chris Tenzer)

First, on the EC board, the ASIC detects a photon signal over the electronic noise (see Subsection 2.2.3). The number of photons is integrated over 2.5 µs the gate time unit (GTU), and transmitted to the PDM-board). This is the so-called *photon trigger*.
On the PDM board the *counting trigger* is implemented: If a photon counter exceeds the average background in one GTU, the MAPMT is marked as active.
The third and most complex sublevel is called *persistency trigger*. The main idea is to check if the second sub level trigger is active for a defined number of consecutive GTUs. All photon counters are therefore stored in a ring buffer. If a *persistency trigger* is detected, the PDM waits for an exposure time to record the entire event. After that the ring buffer is frozen and the PDM signals the CCB that an event is ready.
In addition, some values are calculated to characterize the L1 trigger. Most important is a total photon count over the entire event and a trigger seed for the L2 trigger. The trigger seed includes information about the number of counts and the position of the maximum of the event. The *persistency trigger* is still under development.

2.3.3 Interface between the Photo-Detector Module and the Cluster Control Board

The interface has to be robust and fast to avoid corruption of the data and the dead time. For the data a parallel interface was chosen secured by a Cyclic Redundancy Check (CRC). For the command interface a Serial Peripheral Interface (SPI) with a command width of 32 bit is implemented. To avoid delays by using interface protocols, single signals for the L1 trigger and the broadcast command are used. To provide uniform timestamps for all boards a gate time unit clock is distributed to all PDM-boards and CCBs from the clock board. The PDM-Simulator uses the GTU clock and the broadcast signal as well to provide timestamps and to send data after a broadcast signal. To replace the clock board for test setups in the laboratory the PDM-Simulator is capable to generate and provide the GTU clock and broadcast signal as a secondary functionality.
In total the interface from the PDM-board to the CCB uses 50 Low Voltage Differential Signaling (LVDS) lines in the current design. It is divided into a parallel interface with 18 lines, a Serial Peripheral Interface (SPI) with eight lines, 16 auxiliary lines and eight single signal lines. Every LVDS signal consists of two lines. One line for the positive signal, the other for the same signal but negative. The receiver subtracts the two signals to a single ended signal. The idea is that a disturbance affects both lines in the same way and does not have an effect on the difference.
The parallel interface has a data width of 8 bit. It is used to transfer the recorded data after a L1 trigger or a broadcast signal from the PDM to the CCB. With the SPI commands to the PDM and the status of the PDM are transmitted. The eight auxiliary lines have no functionality and are only a backup. If no problems occur in the field tests of the hardware, the auxiliary lines will not be part of the final design. The four remaining signals are the system clock, the GTU

clock, the L1 trigger and the broadcast signal to freeze the data acquisition and send the data to the CCB.

The bus width of the parallel interface is 8 bit, the ninth line is used for a clock: On the falling edge of this clock the data is valid on the data bus. The data clock is 40 MHz fast, that leads to a transfer rate of 40 MB/s. This is an easy to use and robust technique, but it uses a lot more lines then a fast serial interface as suggested by Bayer (2011a) for the CCB interface. For this interface no serialization or deserialization is necessary. The data eye width, where the data is valid, is more then 20 ns wide for the parallel interface compared to 0.5 ns to 2.5 ns for a serial interface depending on the speed (Bayer, 2011a).

Important to note is that the parallel interface is secured by a CRC. In this case a *IBM-CRC-16* is implemented to secure the data. The idea is to use a polynomial known to both devices, in this case it is a polynomial of degree 16.

$$x^{16} + x^{15} + x^2 + 1 \tag{2.1}$$

The real data is interpreted as a polynomial as well. With the known polynomial, a polynomial division takes place. The remainder is used as the result of the calculation and gets added to the data stream. On the CCB the receiver does the same calculation, but this time including the remainder calculated by the PDM board. If there is no error in the transmission, the result will be zero. This check is sufficient for random errors. A detailed study of error detection is provided by Koopman (2004). The entire data stream of about 300 kB is interrupted by a CRC every GTU frame which results in 2592 B to be secured by 16 bit CRC. Because LVDS signals are used and the interface is not very fast, a maximum of two errors per 16 bit is expected. In this case the error check is more than sufficient (Koopman, 2004).

The SPI consists of four signals, a SPI-clock, a chip select, a master in/slave out (MISO) line for data from the PDM and a master out/slave in (MOSI) line for data to the PDM. SPI is a master/slave protocol where only the master can initiate a transmission, but it works in a full duplex mode, so master and slave can send data at the same time. The interface runs with 2 MHz which leads to a data rate of 400 kB/s in one direction. While the chip select is active and the clock is running, both devices can send bit by bit on their output signals and can receive signals on the input. The purpose of this interface is to send commands to the PDM, for example *get data* or configurations for the ASICs or the PDM. Every data packet gets echoed one transmission later as an acknowledge. In addition, if the interface is idle, the CCB asks for the status of the PDM by sending a *request status*. In these cases the PDM answers with a status vector. This leads to a regular status update. The status indicates if a data packet is ready to transmit or, while the parallel interface is used, whether the PDM-board is still providing data for the parallel interface. Every packet is 32 bit wide. Although the data rate of the SPI is only 400 kB/s compared to other solutions like SpaceWire with 25 MB/s, it meets the requirements. At first a SpaceWire interface was discussed for the command interface, but since SpaceWire is much more complex (see Subsection 2.4.2), it needs a lot more resources in the FPGA. Also it is too slow for the data interface (25 MB/s compared to 40 MB/s necessary) so a parallel interface or a faster serial interface in addition would have been necessary. SPI just needs a shift register and some buffers on both sides. For 2 MHz the problem with the short data eye width does not occur. This is the reason why it was decided to use SPI. In A.1 the entire command list of the PDM is shown.

2.4 Cluster Control Board

2.4.1 Overview

The Cluster Control Board (CCB) is the main hardware contribution by the IAAT to JEM-EUSO. Its purpose is to be a command relay for up to PDM-boards and to perform the second level trigger. The L2 trigger is a pattern recognition to identify tracks on the focal surface. These tracks indicate a EAS event. The trigger rate is reduced from 1.0×10^3 Hz to 0.1 Hz for the focal surface (Bayer, 2011b).

The current baseline is to use a Virtex 4 FX140 FPGA manufactured by Xilinx. It is available in a radiation tolerant version. This is sufficient for JEM-EUSO because the altitude of the orbit of the ISS is about 400 km. Most of the background radiation that could damage the chip is blocked by the magnetic field of the Earth. This board has interfaces to eight PDM-boards. Every interface has a dedicated Random Access Memory (RAM) for data storage while the L2 trigger calculation takes place (see Figure 2.8).

2.4.2 Interfaces between the CCB and the CPU

The interface between the PDM and the CCB has been described in Subsection 2.3.3. For the data and command structure see A.1.

For the connection to the CPU, a SpaceWire connection is used. This interface is error tolerant, every byte is secured by a parity bit, so if there is a bit flip in the data, the calculation of the parity bit fails and the byte is sent again. In addition, every SpaceWire message is secured by a CRC-32 check. A normal command or status message is 13 bytes long (see Appendix A.1). A data packet is about 300 kbyte, the total number is still to be defined. The parity check in addition with the check sum leads to a secure data transmission.

SpaceWire is a standard for spacecraft communication networks, whose development is coordinated by the European Space Agency (ESA). SpaceWire is a serial interface with low latency,

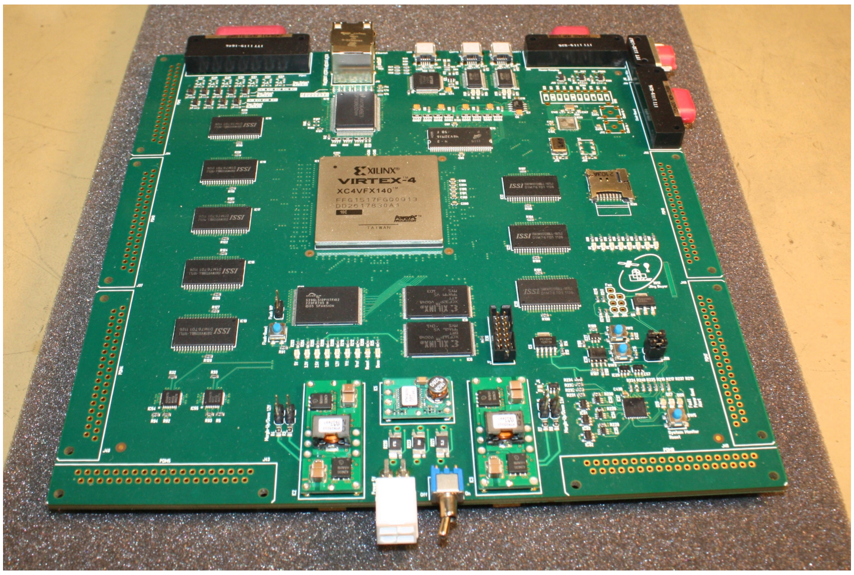

Figure 2.7: Picture of the Cluster Control Board prototype: The FPGA (metallic) is placed in the middle of the board surrounded by the eight SRAM chips (ISSI) and two PROMs (Xilinx). On the top left one connector for a PDM-board is soldered, seven casings for additional connector are placed on the edge of the board. SpaceWire, housekeeping and clock board connectors are soldered on the top right. The power supply is shown on the bottom and on top additional components are placed for debugging and other applications. So far only a USB connection to the housekeeping chip is used.

full duplex and point to point connection. The entire standard is defined by the European Cooperation for Space Standardization (2008). The protocol and hardware is optimized for use in space, the protocol is error tolerant and the proposed connectors and cables are space qualified. For the CCB the *SpaceWire Light* core by *Joris van Rantwijk* from *OPENCORES* is used.

2.4.3 Housekeeping

The temperature, voltages and currents on the CCB are monitored by an Integrated Circuit (IC) with Analog Digital Converters (ADC) to sample the analog signals. The chip is connected via an Inter-Integrated Circuit (I^2C) interface to the FPGA.

I^2C is a serial interface using two lines. It only consists of a clock and a data line. It is a master/slave system like SPI but not full duplex, the master sends addresses of registers of the IC and can change or read out the registers. The chip is also capable of comparing the values with predefined thresholds and issues an alarm if one ore more values are out of operation range. The alarms are fed into an alarm vector in a register of the FPGA and all values are stored as well.

The housekeeping board has an SPI connection and reads out all individual values including

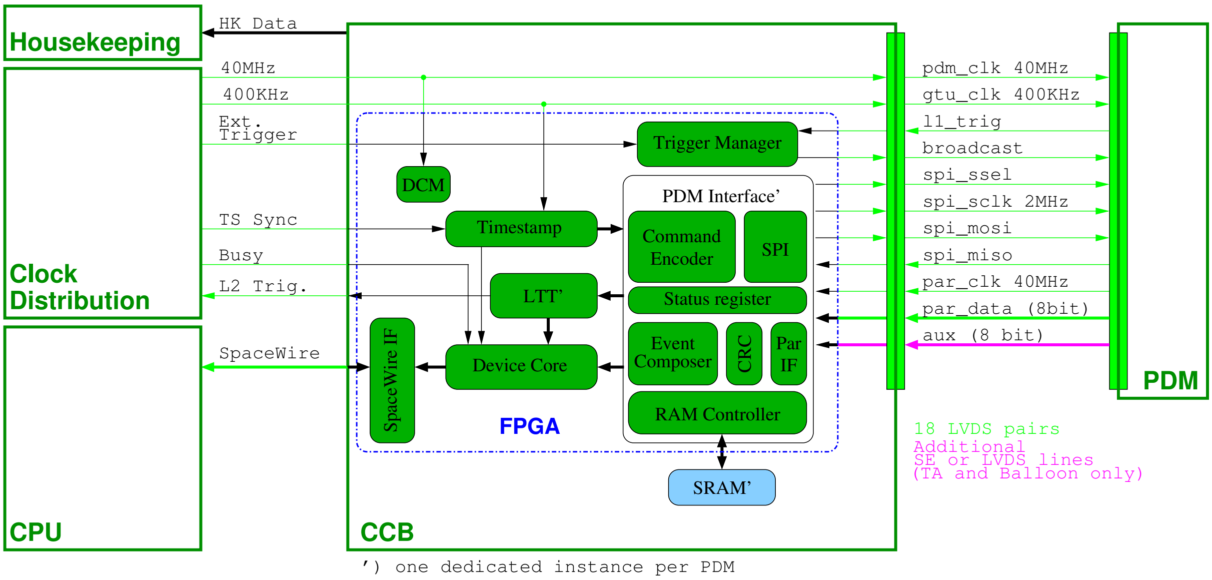

Figure 2.8: Block diagram of the CCB and the connected boards: On the right the interface to the PDM is plotted, on the left to the data processing, clock distribution and the housekeeping. For every PDM a PDM interface instance including a RAM controller for data storage is used. It includes the SPI and parallel interface, CRC, command encoder, status register and an event composer. The DCM provides the system clock and SpaceWire clock generated from the 40 MHz clock. A time stamp is generated for all messages to the CPU, the LTT issues a L2 trigger and the device core composes the packets for the SpaceWire core. Auxiliary signals are drawn in violet. (based on: Distratis, 2012, modified)

the alarm vector. The housekeeping board can also issue a soft reset for the firmware or a hard reset. In case of a hard reset the entire firmware is reloaded from the PROM.

2.4.4 L2 Trigger

The L2 trigger is also called Linear Track Trigger. Its objective is to identify tracks in the data provided by the PDM and compare the tracks with a set of directions stored on the CCB in order to perform a rough track reconstruction method for triggering the events. An EAS appears to be a moving spot of light on the focal surface (see Subsection 1.2.6). By integrating the data, a track of light is visible.

To achieve an identification of these tracks, the algorithm starts with the position of the first appearance of the L1 trigger which is part of the trigger information from the PDM board, and puts an integration box around it. This trigger seed is part of the track since the L1 trigger waits if a light spot is persistent for several GTUs. Currently, 15 GTU time frames, seven in each direction of time, are used for the integration. Boxes are added for one angular direction for the 14 additional GTUs. In the end, all counts along a virtual track of 15 boxes are integrated. This is done for an entire set of directions to cover 360°. This integration leads to a total photon count for every direction.

By identifying the global maximum, a statement for the direction of the track can be done. If the integrated counts for this direction exceeds a predefined threshold and the difference between the global maximum and the second maximum is significant, an L2 trigger is issued.

Bayer (2011a) has done an analysis comparing the performance of 324 and 67 directions. 324 directions will be the offline analysis done with the ESAF for angular reconstruction. Since computing time is limited on board and the dead time will be larger for these so called *baseline* set of directions, a reduced set of 67 directions was tested as well. This reduced set of directions shows a difference of ±200 counts, which is less than the background per pixel and GTU, for nearly all events. In Figure 2.9 a comparison of these two analyses is shown. The global maximum is in almost the same position and with almost the same number of counts. The x-axes are not given in degrees, but show the index of direction. The smaller maxima occur if parts of the track are inside the integration boxes, but not the entire track.

One of the main objectives of the PDM-Simulator is to test this trigger algorithm with a set of simulated ESAF data. The objectives are to test the trigger efficiency, possible flaws in the algorithm as well as the timing of the data transmission and the calculation. This will be done to evaluate the dead time in a comparable set up. After this test, a decision about the final number of directions, the threshold of the integrated counts and the number of used GTU frames can be taken.

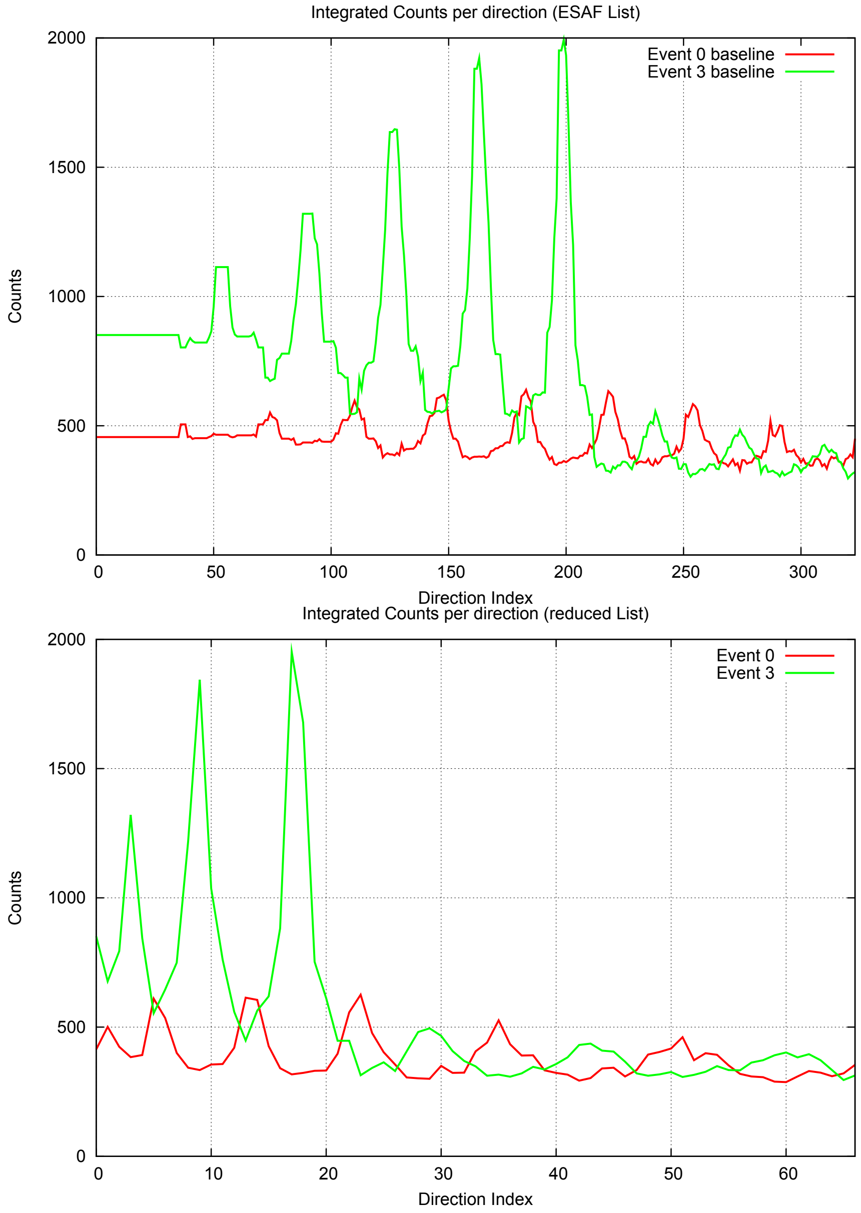

Figure 2.9: Comparison between the baseline and the reduced set of directions: The x-axis shows the index of the directions, the y-axis the integrated counts. The two colors are different data sets with different energies. Event 0 has an energy of 1.39×10^{20} eV, Event 3 of 3.92×10^{20} eV. The top graph is the result using the 324 ESAF directions, the lower one shows the reduced 67 directions. The global maximum corresponds to the direction of the track, the other maxima occur because, for some directions, parts of the track are inside the box but not the entire track. The reduced set of directions shows a difference of ±200 counts, which is less than the background per pixel and GTU, for nearly all events. (Bayer, 2011a)

3 Simulator of the Photo-Detection Module

3.1 Overview

The Cluster Control Board (CCB) is currently being developed at the Institut für Astronomie und Astrophysik Tübingen (IAAT). The hardware interfacing the CCB is developed in Italy (MDP and Clock Board), in Mexico (housekeeping) and in Korea (PDM-board). Only on rare occasions the hardware is in the same place for tests. To test the interfaces and the behavior of the CCB replacements for the components are used. The MDP is replaced by a PC with a SpaceWire card and a near real time analyses framework. The housekeeping is substituted by a SPI to USB converter with a readout software and a visualization tool.
This work is about the replacement of the PDM-board and the Clock Board, the PDM-Simulator. The PDM-Simulator should feature the same interface functions and the overall behavior as the real PDM. For tests of the interface, the PDM-Simulator should provide dummy data in the same format defined for the PDM and for tests of the LTT it should provide simulated data from the EUSO Simulation and Analysis Framework (ESAF). To replace the Clock Board the PDM-Simulator must generate the system clock, the GTU clock and the broadcast signal for the CCB. This Chapter will give a detailed description of the components used for the PDM-Simulator and about the functionality.

3.2 EUSO Simulation and Analysis Framework

Simulated data generated by the EUSO Simulation and Analysis Framework (ESAF) is used to test the interface between the PDM-Simulator and the CCB and will also be used for tests of the LTT on the CCB. ESAF is a framework developed for the former EUSO project. It is written in C++ and FORTRAN and provides functions to simulate the entire chain of events for a space based observation of UHECR. After injecting a primary particle, it calculates the particle production in the atmosphere including the fluorescence and Cherenkov photons and their propagation to the detector. The properties of the optics and the MAPMTs are simulated as well and in the end the detector signals are simulated. This can be done for several events with different energies and angles to evaluate for example the angular and energy reconstruction and to test the trigger algorithm. ESAF uses ROOT, a software developed by the Conseil Européen pour la Recherche Nucléaire (CERN) to analyze huge amounts of data from the Large Hadron Collider (LHC) experiments. The Framework is highly modular, every step is done by a different module. In case the interface between the modules is not changed they can be replaced without adjusting the entire framework. For example the detector part was exchanged when moving from EUSO to JEM-EUSO or for simulating events for the EUSO-Balloon, a field test planned before the space mission (see Chapter 4). For the simulations different Monte Carlo and hadronic interaction models can be used. Mernik (2009) prepared his diploma thesis on this subject and provides a detailed description of the techniques applied in ESAF.

3.2.1 Simulated Test data from ESAF for the PDM-Simulator

The main subject of this work is to test the interfaces between PDM, CCB and CPU. Most of the time a dummy data packet was used. The data consists of the current number of bytes counting from 0 to 255 and start over. This data was generated with a BASH script using the *printf* function. The output is saved to a data file. To test a visualization tool on the German PC, simulated data from ESAF was used as well. This is a list of a set of events for EUSO-Balloon setup with and without background used during the development of the PDM-Simulator[2]:

Table 3.1: Simulated events for EUSO-Balloon

Energy in eV	θ	ϕ
1×10^{19}	30	130
1×10^{19}	20	135
1×10^{18}	30	130

On the top Figure 3.1 shows the second simulated event of table 3.2 for the EUSO-Balloon setup without background. On the bottom a histogram with the GTU as x-axis and the number of counts as y-axis for a comparable event is shown. This figure shows how many photons are generated by an event and what fraction of them is actually detected.

The data is converted from the *ROOTS* data format to a raw data file using extended American Standard Code for Information Interchange (ASCII) encoding. In this case the pixel map of the PDM starts with the pixel on top left and it is counted from left to right. In addition to the actual data, headers are added to the data stream before it is sent via the USB to the PDM-Simulator to fit the structure of the real data packet by using another BASH script. For a detailed overview of the data structure see Appendix A.1.

[2] I wish to thank my colleague Thomas Mernik for providing me the simulated data.

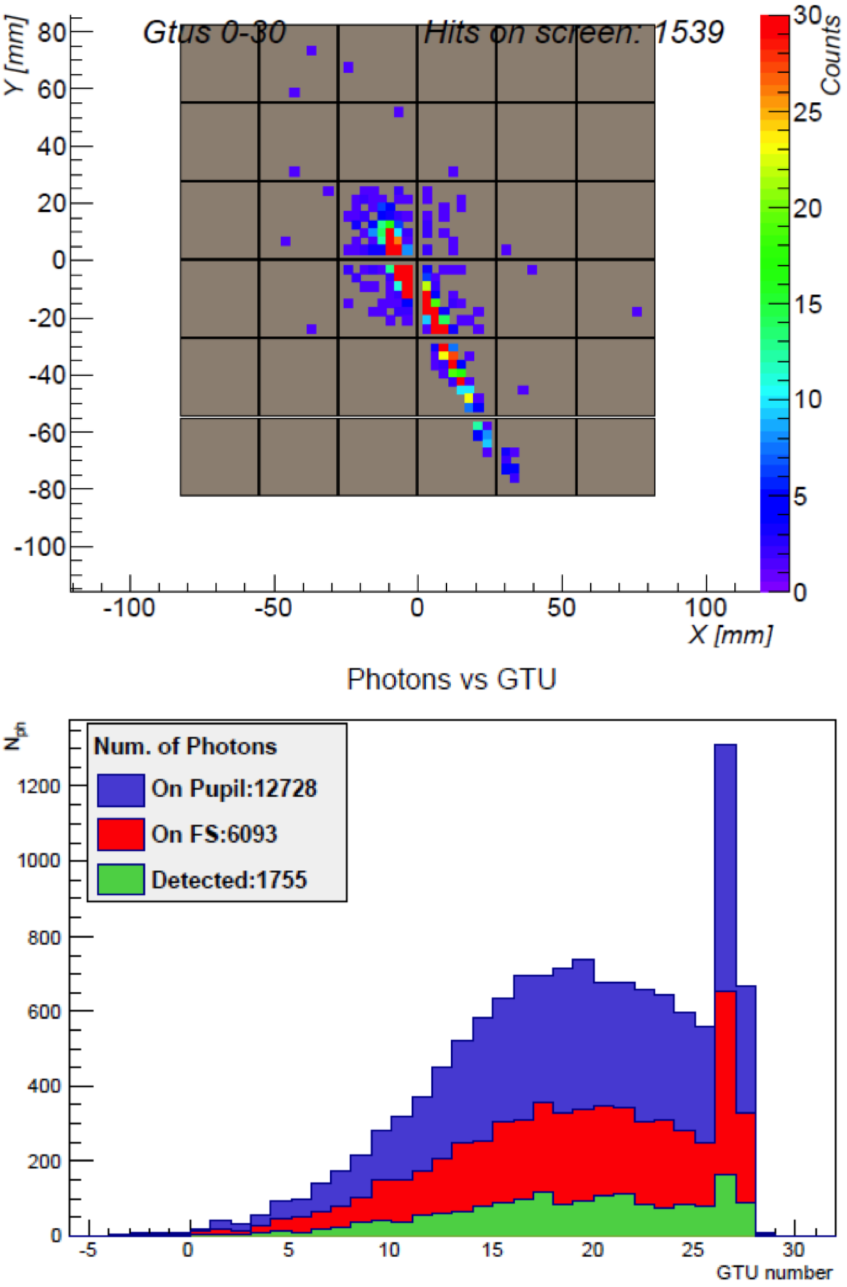

Figure 3.1: Simulated EAS: The image on the top shows a simulated event for EUSO-Balloon with an energy of 1×10^{19}, $\theta = 20°$ and $\phi = 135°$. The focal surface consists of one PDM, the black squares are one MAPMT with eight times eight pixels. On the bottom, a histogram of a comparable event is plotted with the GTU on the x-axis and number of counts on the y-axis. In blue the number of photons reaching the pupil is plotted, in red the photons reaching the focal surface and in green the detected photons. (graphics by Thomas Mernik)

3.3 Requirements of the PDM-Simulator

The main goal of this work was to develop a board to replace the PDM for testing in the laboratory. Another project in Tübingen is a computer with a SpaceWire card to replace the CPU in the JEM-EUSO set up. In a big collaboration it is not possible to have the entire hardware available at all times which makes a test setup necessary. In the end an entire chain with the CCB in the middle was established to test the interfaces and test the LTT.

The PDM-Simulator should have a connection to a computer to send ESAF events to it. This interface has either to be fast enough to establish a constant data stream from the computer to the CCB or it must have sufficient storage on board.

The interface to the CCB should work in the same way as the real interface, so a great number of Input/Output (I/O)s are necessary to reach this goal. The PDM-Simulator also needs computing power to calculate the CRC check sum. In addition, the board should issue a L1 trigger if needed and supply the L2 trigger with a trigger seed.

During the development additional requirements were defined. The PDM-Simulator should also be able to supply the system clock, GTU clock and the external trigger to the CCB. The CCB has no onboard oscillator to generate a clock so it has to be supplied from an external source. At first a function generator was used to generate the 40 MHz clock and an external LVDS driver to feed it to the CCB, but due to shifts of the clock this was not stable. This lead to malfunctions of the CCB from time to time. By using the clock provided by the PDM-Simulator and a faster LVDS driver onboard the PDM-Simulator this problem was solved.

3.4 Trenz Micro-module: TE0146

The idea was to use an FPGA as the main component to have the flexibility to develop all functionality in a continuous process. Additional functions can be added by reprogramming the device. In addition an FPGA provides many I/Os (TE0146: 55 I/Os), parallel processing of different tasks and a fast access to memory on the chip. A Spartan 3 manufactured by Xilinx was chosen because it is a cheap solution and it is used in many setups at the IAAT. Trenz Electronics manufactures small development boards with a Spartan 3 and a USB chip. With this chip first tests were made, but it quick showed some disadvantages.

The USB connection has a maximum transfer rate of about 10 MB/s. Compared to the transfer rate of the parallel interface between CCB and PDM of 40 MB/s this is not sufficient to establish a constant realistic data stream. The internal memory of the Spartan 3 is too small to store an entire event.

One solution that was suggested was to store only pixels with data above the background and add a generated background before sending the data to the CCB on the FPGA. But because of limited resources and the wish to use the simulated ESAF background, this plan was discarded.

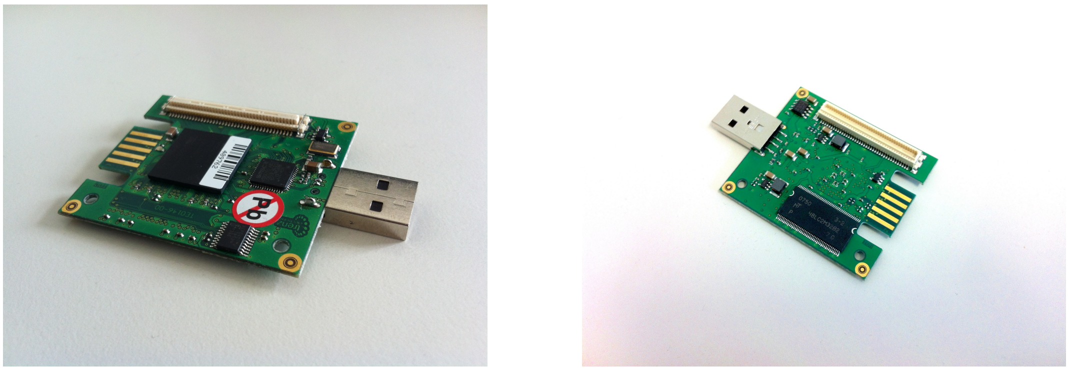

Figure 3.2: Trenz micro module *TE0146*: The picture on the left shows the top view of the micro module. In the center the Spartan 3 is soldered. Right next to it the UTMI chip is placed followed by the USB connector. The golden pins on the far left is the JTAG interface. The white connector is called B2B connector and provides all available I/Os, power supply, USB and JTAG interface. The picture on the rigth is the bottom view with the SDRAM on the bottom.

The background generated by the FPGA would have been less random then the simulated ESAF background. Another problem with this board is the limited number of LVDS I/Os.

Another product of Trenz Electronics, the *TE0146*, features a SDRAM onboard. Figure 3.2 shows two pictures of the module. On the left, the top view with the FPGA and the USB chip is displayed; on the right, the bottom view with the SDRAM. This memory is fast enough to feed the interface with data, but the micro-module has less I/Os than the board that was considered at first. The conclusion was to design a base board for the *TE0146* and use external LVDS drivers and receivers.

The next subsection gives a detailed description about the FPGA that is used, Subsection 3.4.2 will give an idea how the USB and SDRAM works from a hardware point of view. Section 3.5 is about the layout of the baseboard, the LVDS ICs and the power supply.

3.4.1 The Spartan 3 FPGA

The Spartan 3 is an FPGA manufactured by Xilinx. It is a low cost device with fast switching characteristics better then 5 ns so data clocks up to 200 MHz are possible. The version on the *TE0146* board is called *XCS1000* and features 17 280 logic slices and 552 kb internal memory. At the IAAT the Spartan 3 is a widely used device with several base boards for developing. The firmware is written in the VHDL using a development kit from Xilinx called ISE. For the final version *ISE 14.3* was used for synthesizing the code. The code is translated in a bit stream for the FPGA. The Spartan 3 uses SRAM technology, the bit stream configures the logic slices and connects them in a way that the behavior described in VHDL can be performed by the FPGA. FPGAs are capable of executing highly parallel simple operations very fast.

3.4.2 Other Components on the TE0146

The *TE0146* main component is the Spartan 3 FPGA, the I/Os not used by the other components onboard are routed to a B2B connector with 80 pins in total. With this B2B connector, the programming interface and the power supply can be connected. This leaves the user with 55 I/Os, 20 of which can also be paired to 10 LVDS pairs. A status Light Emitting Diode (LED) placed on the micro board is available.

The other connectors are a USB connector and a JTAG interface on the other side of the board. The USB connector can be used to power the device from the computer using the 5 V power supply defined for a standard USB port.

On the micro module a PROM for the VHDL design is placed on the right and a UTMI chip in the middle. On the other side of the board a SDRAM with 64 Mb is soldered.

This module features all components necessary to fulfill the requirements, but the number of LVDS I/Os is too small and the signal standard does not match the signal standard of the CCB. The *TE0146* uses 3.3 V as a reference voltage for the I/O although the CCB uses 2.5 V. To solve these problems it was decided to design a dedicated base board for the PDM-Simulator and not to use the development base board provided by Trenz Electronics.

3.4.2.1 SDRAM

One memory cell of of a Dynamic Random Access Memory (DRAM) consists of a capacitor and a transistor. The content is terminated by the charge of the capacitor. Compared to a SRAM memory cell, that stores the content by using a flip-flop consisting of at least four transistors, this design provides more storage on the same region of a chip because it needs less resources (see Figure 3.3).

But the content of a SRAM is stable as long as it is supplied with power, compared to a DRAM because the charge of the capacitors dissipate over time. From time to time the content of the DRAM memory cells has to be refreshed by rewriting it. In the case of the PDM-Simulator synchronous DRAM is used. All commands and data are synchronous to the system clock.

This SDRAM chip has a capacity of 64 Mb, with a 32 bit wide data interface and it can be used with a clock up to 143 MHz. In the case of the PDM-Simulator, the 60 MHz system clock is used for the SDRAM as well. Without duty cycles the chip has a data rate in this configuration of 240 MB/s. Excluding the refresh cycles every 3.9 µs and the commands to precharge and activate the rows, it is safe to say that the SDRAM using the 60 MHz clock exceeds the requirement for the interface between the PDM-Simulator and the CCB (40 MB/s) easily. The actual performance was not measured so far.

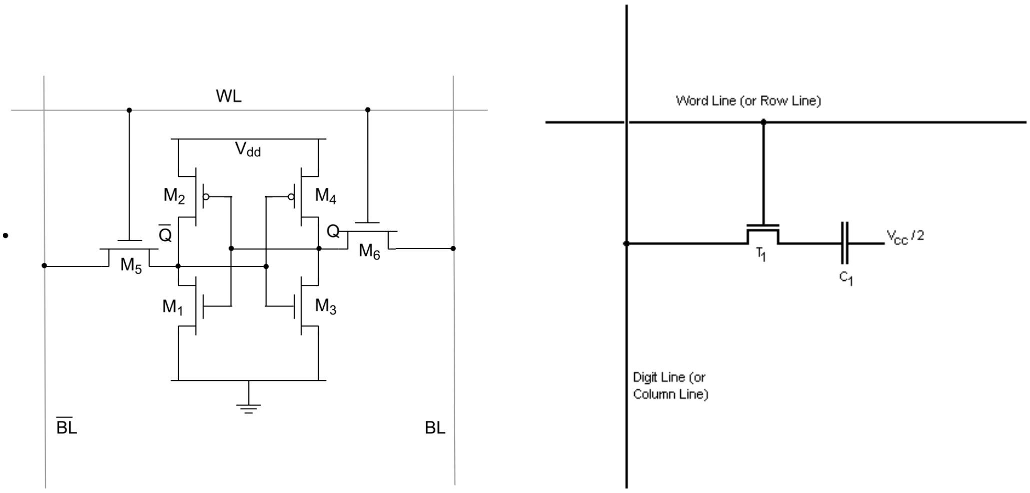

Figure 3.3: Comparison of a SRAM and a SRAM memory cell: On the left a SRAM memory cell is shown. In this case a it consists of six transistor to store the data in a flip-flop (http://web.sfc.keio.ac.jp/~rdv/keio/sfc/teaching/architecture/computer-architecture-2012/lec08-vm.html). The right plot is showing a DRAM memory cell with only one transistor and one capacitor. (http://www.cse.scu.edu/ tschwarz/coen180/LN/DRAM.html)

3.4.2.2 USB

The UTMI chip used on the *TE0146* is capable of USB 2.0 480 Mb/s *high speed* data rate. It is widely used at the IAAT, the necessary program for the computer is written in C provided by Henry Gebhardt, who used this chip in his diploma thesis and also provided the VHDL component. For more information see Gebhardt (2009) and USB Implementors Forum (2000). For the amount of data used in this design it was necessary to split the data transmission in several parts.

3.5 Base board for PDM-Simulator

The main objective of the base board is to provide a sufficient number of LVDS I/Os, a USB and a programming connector. As a reference the Trenz Electronics development board was used but with changes concerning the JTAG signals that are located in a different position. For designing the board *Target 3001* developed by *Ing.-Büro Friedrich* was used. Figure 3.4 displays the circuit diagram of the base board.

The board consists of four copper layers, the top level on ground and the bottom level on 3.3 V. The two layers in between are used for routing signals and power connections if it is not ground or 3.3 V. There is no ground connection to the screws, the shielding of the board or the shielding of the cable. One requirement for the hardware is that all components work without a ground

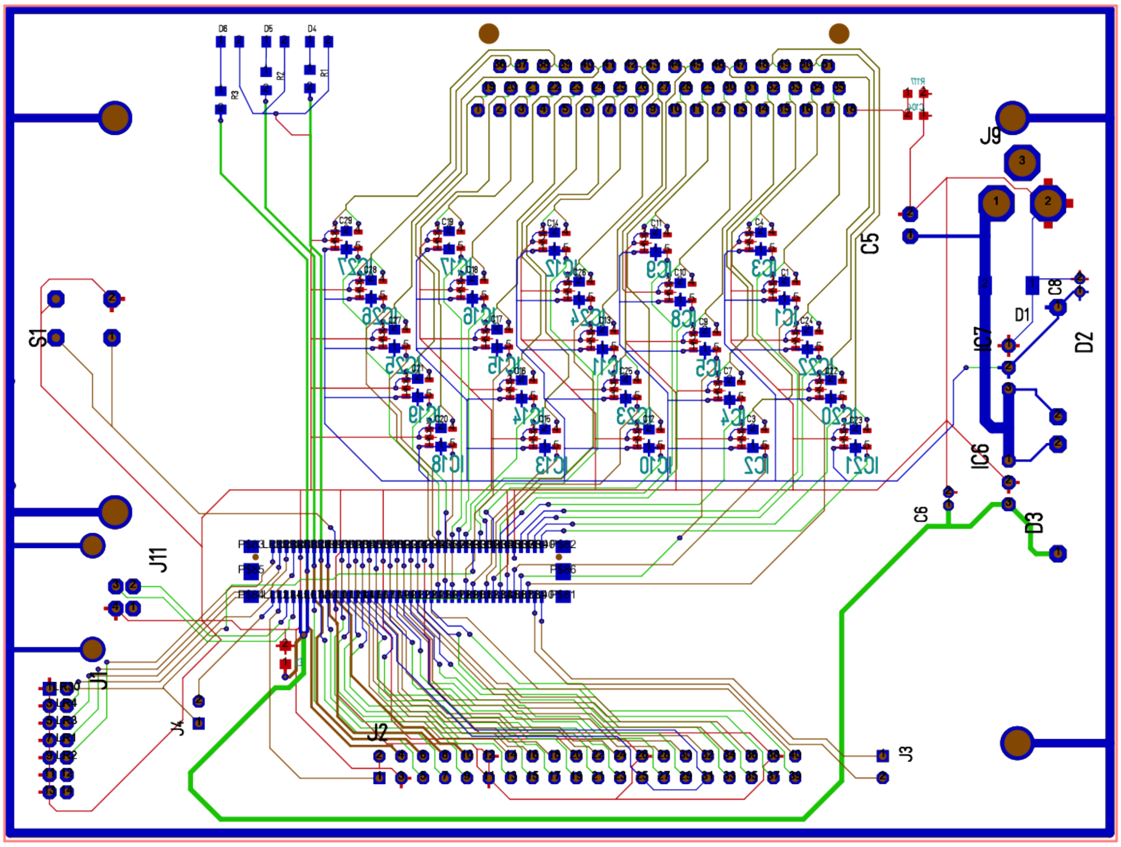

Figure 3.4: Circuit diagram of the base board for the PDM-Simulator: The base board consists of four copper layers. The red layer is the ground layer. Although the layer is except for a few signals completely on ground, the ground connections are visible as single signals. The same applies to the blue layer, but for 3.3 V. In a last step these layers are converted into a copper plain to provide as much copper as possible for the connections with the highest current. The inner two layers in green and brown are the signals. On the right the structure of the power supply is shown. On the far left the USB connector and the JTAG connector are soldered. On the bottom the spare I/Os and power supplies are place as a pin band. On top the PDM to CCB connector is shown. Left of it three LEDs can be connected and the 25 ICs are the LVDS drivers and receivers. Next to it the B2B connector for the Trenz micro module is placed.

connection. The grounding is the last step before deploying the hardware to have full control of all connections between the boards to avoid ground loops.
In the middle the *TE0146* connector is placed. The board is powered by two 5 V connections. Spare I/Os and the output of the power converter of the *TE0146* are routed to a pin band on the bottom of the board. On the left the JTAG and USB signals are routed to the connectors. In addition a *PROG* pin is connected to ground separated with a switch. By pulling this pin to low the design is reloaded from the PROM.
On top of the circuit diagram pads to solder three LEDs are placed. The 3.3 V of the FPGA is transformed into 1.2 V for the LED by using resistors. Two of them are used as status indicators for the PDM-Simulator. The green one indicates if the board is operational. After a reset this light will go off for one second. The red LED indicates a data packet in the SDRAM ready for transmission.
The 25 signals for the PDM interface are routed to 25 ICs, either a LVDS driver or receiver. From these ICs the signal pairs are routed to the pin holes for the PDM connector. Important is to route signal pairs close to each other and with the same length to avoid lagging of some signals. In this case the signals are routed right above each other, such that a electro-magnetic disturbance will most certainly affect both lines. The ICs need a power supply matching the reference voltage of the single ended signals, in this case 3.3 V and a ground connection. To avoid spikes in the power supply, capacitors are placed between the 3.3 V and ground for every IC. For the CCB a *MDM-51* connector is used.
For the board two low voltage power supplies are necessary. The *TE0146* needs a constant voltage of 5 V and the LVDS ICs need 3.3 V to generate the signal pairs or the single ended output. The power supply is found on the left hand side of the circuit diagram. The board can handle an input of 5.5 V to 12 V, but high voltage can lead to thermal problems. This was never tested since the voltage regulators generate stable output for lower voltage. A commercial AC converter with an out voltage of 7.5 V is used. The converters onboard give the freedom to use any AC converter with an output in the specified range. This increases the flexibility since no accurate laboratory equipment is needed to power the board. Figure 3.5 is a picture of the board with all components but the *TE0146* in place. Figure 3.6 shows the connectors of the PDM-Simulator.

3.5.1 LVDS Driver and Receiver

The LVDS driver and receiver used are commercial products from Texas Industries called *SN65LVDS2* (receiver) and *SN65LVDS1* (driver). They have a power supply and ground pin in addition to the three pins for signal conversion. According to the data-sheet every receiver consumes up to 60 mW and every driver 25 mW. The five receiver and fourteen driver used on the board consume 650 mW together. To relieve the power consumption of the *TE0146* board, the ICs are connected to the main power supply.

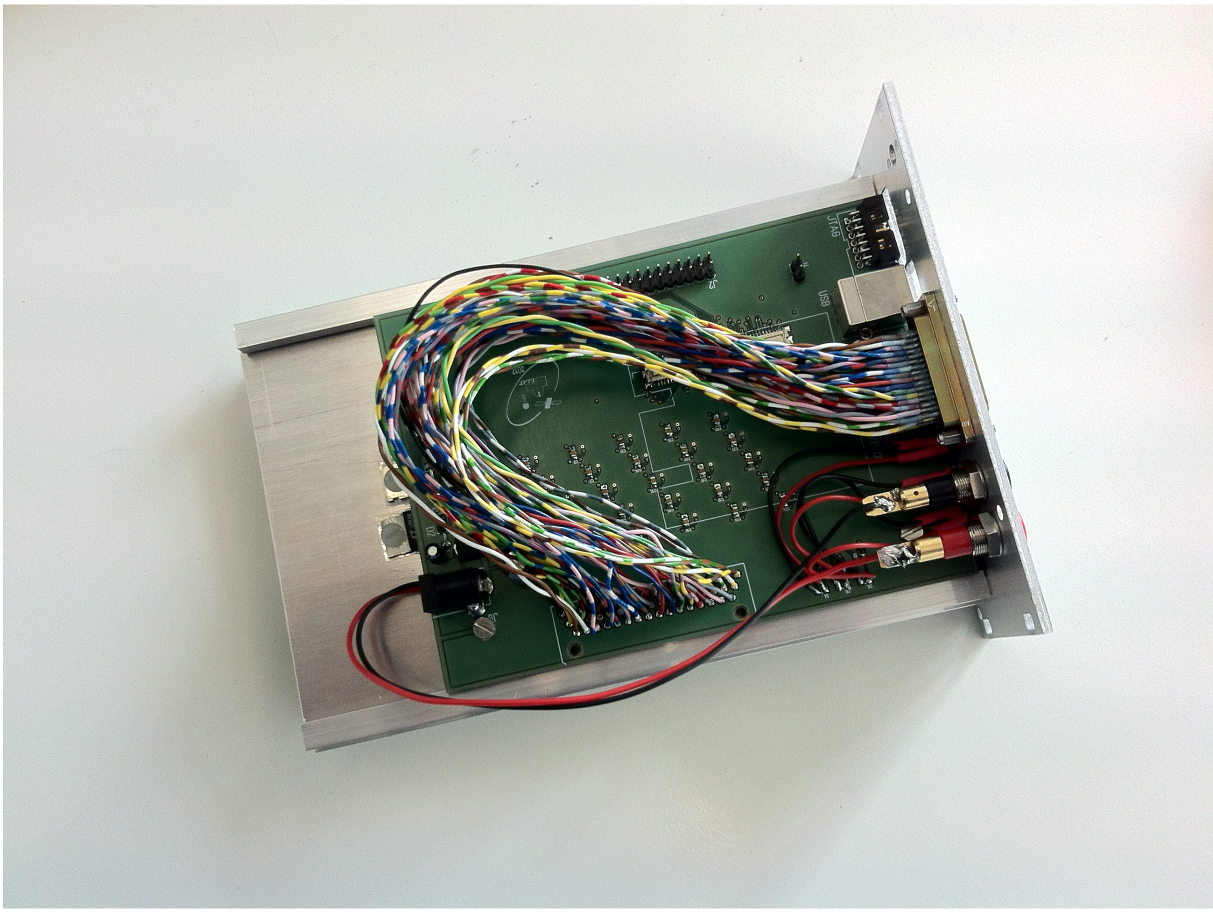

Figure 3.5: Base board for the PDM-Simulator: On this picture all components are in place except for the *TE0146*. On the left the power supply is connected to the front pane and the power converters are connected to the casing to dissipate the heat produced by the converters. The pin band and the B2B connector are partly hidden by the cable for the CCB connector on top of the picture. In the middle the capacitors for the LVDS driver and receivers are soldered. On the right the front plane with the connectors and the status LEDs can be seen edge-on.

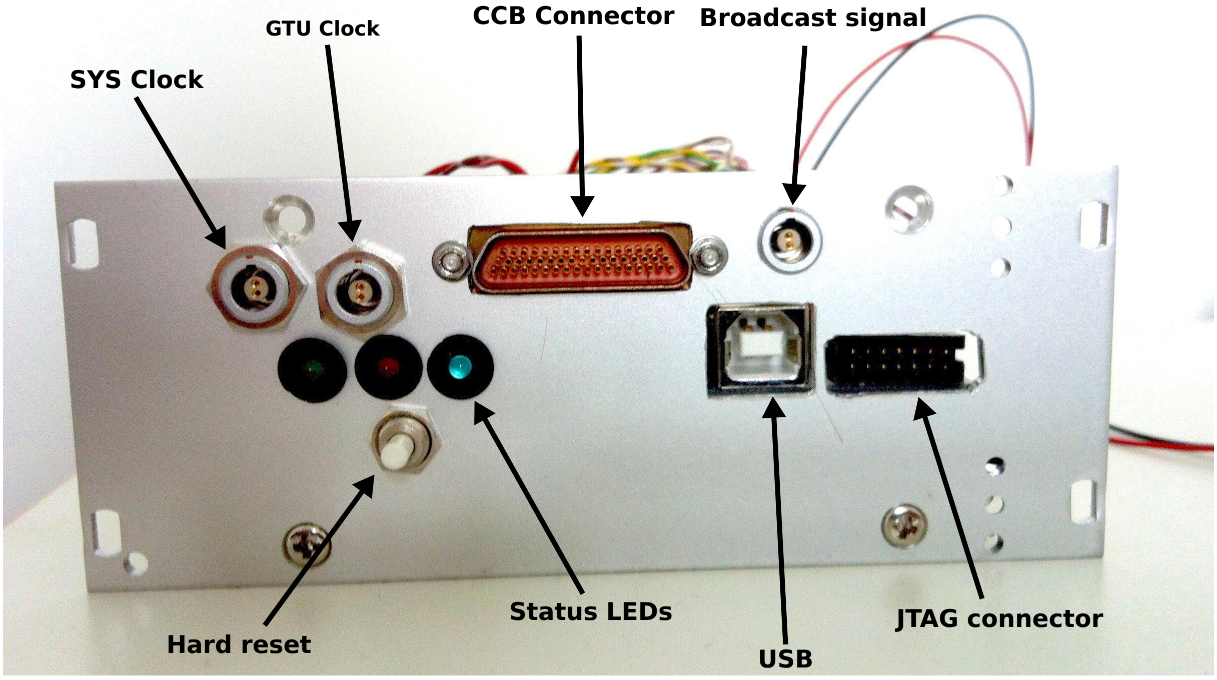

Figure 3.6: Front of the PDM-Simulator: The connectors for the power supply on the left are replaced by the LEMO connectors for the clocks. The power is connected from the back of the board. The third LEMO connector provides the broadcast signal. Between the LEMO connectors the MDM51 connector for the CCB is placed. In the second row from left to right are the status LEDs, the USB connector and the JTAG connector. The white button issues a hard reset.

The transmitter injects a constant current of 3.5 mA in one of the signal pairs depending on the logic level of the input to the driver. The receiver terminates the cable with a 100 Ω resistor to avoid reflections. After passing through the resistor the current returns to the transmitter. The result is a voltage difference of ±350 mV between the to lines, positive for high and negative for low. The difference is compared to thresholds to derivate a digital signal.

The result of the two signal pairs is a common mode voltage of 1.2 V, a maximum voltage of about 1.4 V and a minimum of 1.0 V. A close electromagnetic coupling of the two pairs reduces the generation of noise on the common mode voltage and the difference of the two signals. Most of the fields are canceled due to the opposite direction of the current. Noise generated by external fields does not affect the signal in a way it would in case of a single ended signal because the effect is the same for both lines. Thus, the difference does not change. LVDS was first used to transmit data to displays and expanded fast to other data transmission applications.

3.5.2 Power Supply

The power supply input of 5 V to 12 V is routed to two power converters on the board. To protect the electronics from an inverted voltage a Schottky diode is used. In addition, to smooth the input and output of the power converter, capacitors are placed between ground and the power rails. The power converters are specified for up to 12 V input. The output of one converter is 5 V, only used for the *TE0146* board. The second converter's output of 3.3 V is connected to the first copper layer of the board to supply the LVDS ICs. In addition, the 3.3 V are routed to the I/O bank on the bottom of the layout in case additional components are connected to the board.

The power converters produce most of the heat. To avoid high temperatures, the cases of the converters are coupled with the plate on which the board is mounted. An isolation layer is placed between the converters and the plate.

3.6 VHDL-Design

3.6.1 Overview

The firmware for the FPGA is written in VHDL using the *XILINX* development tool Integrated Software Environment (ISE). For debugging, the simulation tool *ISIM* and the internal logic analyzer *Chipscope Pro* were used.

The main task was to implement the data handling of the data received from the computer via USB using different buffers to separate the interfaces. The VHDL design also includes the generation and distribution of a 40 MHz system clock, a 400 kHz GTU clock and an external trigger. The external trigger is generated by using a button attached to the board. The signal coming from the button is converted to a signal identical to in the clock board design.

3.6.2 USB

The data from the computer is received by the USB component. Only the receiving part of the component is used because there is no case planned in which the PDM-Simulator sends data to the computer. All the status changes are sent to the CCB and from there to the CPU via SpaceWire. The data is provided parallel and synchronous to 60 MHz with a bus width of 8 bit. Due to the fact that the USB protocol does not achieve a data rate of 60 MB/s and the FPGA has the possibility to delay the data stream the following two handshakes are implemented.

If a byte is ready in the USB component, it issues a *RXVAL* signal to indicate that there is valid data on the *RXDAT* port. With a *RXRDY* signal the receiving of the data is acknowledged. The moment there is valid data, it is stored in a First In, First Out (FIFO) buffer. The FIFO is generated by the *XILINX Core Generator*. It is a buffer where the data is stored in the order of the writing. As soon as the data is read the memory cells are free again and can be used for the next data packet. The FIFO is realized by using the block rams on board of the FPGA. In this case they are fast static RAM memory cells grouped together to 16 kb blocks. The addresses are generated by using pointers. To avoid over or underflows the FIFOs feature *FULL* and *EMPTY* signals. The user can also introduce programmable thresholds for additional indicators of the content of the buffer.

The FIFO is used for several reasons. The output of the USB component is 8 bit wide, but the input of the SDRAM is 32 bit. This ratio can be changed by using a FIFO and simply waiting for 4 bytes before providing a 32 bit output. This could also be achieved by using shift registers. However, to avoid slowing down the USB connection, a bigger buffer than just a shift register is advisable. Since another reason is that the SDRAM is not always ready to store data. The SDRAM runs in a burst mode of eight consecutive read and write cycles. To complete one entire write command, 32 Byte need to be written.

3.6.3 SDRAM

The SDRAM component consists of only one Finite State Machine (FSM). After powering up the FSM starts with the *initial* state. In this state a counter is increased up to a little over 100 µs. In this state all the operations necessary for the start up take place.

The SDRAM is controlled by four signals, Chip Select (CS), Row Access Strobe (RAS), Column Access Strobe (CAS) and Write Enable (WE). The signals are active low. Figure 3.2 gives a command list for the SDRAM. This is a reduced command list, only the commands used in the design are mentioned. To simplify the process, the commands has been defined as constants in the code and a command vector was defined.

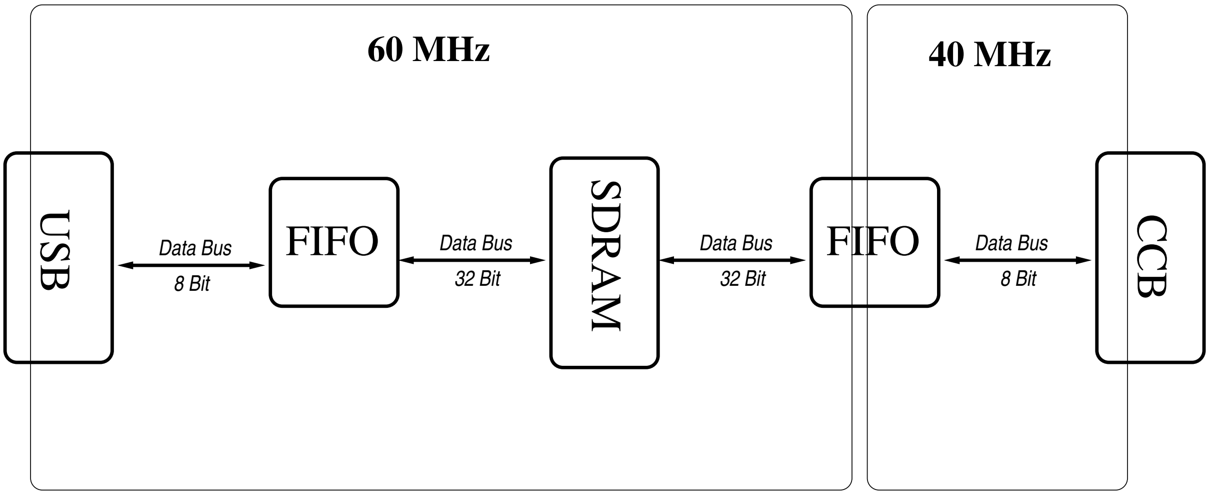

Figure 3.7: Schematic of the data handling onboard the PDM-Simulator: From left to right the data path onboard the PDM-Simulator is shown. After receiving the 8 bit wide data stream the data is stored in a FIFO. If at least 32 B are stored the data is transmitted to the SDRAM with a data width of 32 bit. The data is requested by the PDM to CCB interface and read to another FIFO and read by the interface with 8 bit data width. In the process the clock domain is changed by using the 40 MHz clock as a read clock.

During the 100 µs the command is changed from *COMMAND INHIBIT* to *NO OPERATION* the CS is pulled to low to activate the SDRAM. After this period all cells have to be precharged. At least two refresh cycles are necessary before the mode register can be programmed. A *PRECHARGE ALL* and two *AUTO REFRESH* commands are issued to prepare the chip for programming the mode register. Before sending a *LOAD MODE REGISTER* command the desired mode register is loaded into the address bus. The mode is set to *programmed burst length* of eight, sequential output with a latency of three clock cycles. For a latency of three clock cycles the SDRAM access time is 5.5 ns compared to a system clock duration of 16.667 ns. After that the SDRAM is ready for operation.

Without duty cycles data rates up to 240 MB/s are possible, the actual data rate was never tested but should be above 200 MB/s and exceeds the data rate of the USB ($\sim$ 10 MB/s) and of the PDM to CCB interface (40 MB/s) by far.

The bus width of the SDRAM is 32 bit compared to 8 bit for the USB component. To overcome this ratio and to have 32 Byte on hand for the burst length of eight a FIFO is used as a buffer between these to components. If the FIFO stored enough data, the programmable empty changes to low and the SDRAM controller reads one burst and saves it to the SDRAM. By using this threshold, buffer underflows are impossible. The addresses are generated by using a counter. If this counter reaches the predefined packet length plus an overhead of 128 Byte, the FIFO and USB controller are reseted to be ready for the next data. The overhead is used for additional information, for example the L1 trigger and the trigger seed and, in case the data is

Table 3.2: Reduced command list for the SDRAM

Current State	CS	RAS	CAS	WE	Command
any	H	X	X	X	COMMAND INHIBIT
	L	H	H	H	NO OPERATION
idle	L	L	H	H	ACTIVE (select and activate row)
	L	L	L	H	AUTO REFRESH
	L	L	L	L	LOAD MODE REGISTER
	L	L	H	L	PRECHARGE
row active	L	H	L	H	READ
	L	H	L	L	WRITE
	L	L	H	L	PRECHARGE (deactivate row in bank)

not a multiple of 32 Byte, to fill up the data.

To simplify the operations and to save logic on the FPGA a write process will be continued until it is completed or no further data is received via USB. In this time the SDRAM controller does not react to any other commands. When the data is complete, the SDRAM controller issues a signal to the status generation.

When a first level trigger is transmitted from the computer for this event, a L1 trigger is issued and the status vector indicates that an event is ready. In case no trigger is foreseen, the PDM-Simulator waits for a broadcast signal and changes the status vector to event ready after that.

After the CCB requests an event the SDRAM controller reads the data from the SDRAM and stores it into a second FIFO. This FIFO changes the bus ratio from 32 bit to 8 bit for the PDM to CCB interface. The interface is clocked with 40 MHz and not 60 MHz. To cross this clock domain border, the FIFO uses the system clock of 60 MHz to write and the interface clock of 40 MHz to read. The FIFO features a programmable full threshold, the read process is paused if less than 32 Byte are available. By using this buffer, buffer overflows are avoided. The buffer passes the data to the interface controller. After the packet is read entirely, the SDRAM controller waits either for new data or for the next request for data.

For the entire process an internal counter is increased on rising edge of the system clock to keep track of the time. When a threshold of $\sim 10\,\mu s$ is reached, the next time the SDRAM controller is idle an *AUTO REFRESH* is done and the counter is set to zero. In principle a refresh cycle every 15.625 µs is sufficient, but security margins were included because the performance of the SDRAM exceeds all requirements by the factor of four. After every burst the controller checks if the threshold is reached. The SDRAM chip itself keeps track which rows are refreshed so far so that no additional action on the controller's side is necessary. This secures the error free operation of the SDRAM.

In conclusion the SDRAM provides sufficient storage for the PDM-Simulator's functionality and the performance of the SDRAM controller is so good that it will not slow down the USB and more importantly the PDM to CCB interface. It has 8 MB of storage compared to a full event of about 350 kB. By using the FIFOs as a buffer, the interface is provided with a constant data stream with no interruptions. By storing the data in consecutive addresses and by avoiding interruptions the resource consumption is less than 5 % of the FPGAs logic slices.

3.6.4 PDM to CCB Interface

The code for the implementation of the interface on the PDM-Simulator side was developed together with Giuseppe Distratis who developed the interface on the CCB side. While the code used for the interface in the PDM-Simulator is not the same as in the real PDM, the functionality however is replicated exactly. In addition, the code is part of a simulation test bench for the CCB. Figure 3.8 shows a block diagram of the PDM-Simulator. It includes the interfaces to the CCB and the PC and gives an overview of the operations done on the PDM-Simulator.

The SPI ports and the data ports between CCB and PDM are connected directly to the interface component. The SPI commands are received by using a shift register and add together the serial bits to a vector. For 32 SPI clocks a bit from the MOSI line is added every rising edge of the clock to a shift register. After 16 µs (32 2 MHz clock cycles) the 32 bit wide shift register contains an entire command. When the command is complete, it is accessible by using the *command* port. A valid signal indicates whether the command is valid and new. Commands concerning the interface are interpreted in the interface component. This results in a *data_send*, *data_error* and *data_ack* signal. Most of the commands are not interpreted by the interface, for example all configuration commands for the PDM-board and the SPACIROC.

The moment a command is valid it is sent back, again using a shift register. This time bit by bit is fed to the MISO line every falling edge of the SPI clock so the receiver can sample on the rising edge. The result is an acknowledgment for the master (the CCB) if the command was received correctly. The PDM board sends the commands back when they are executed correctly. To give the slave enough time to do so, the SPI pauses for 1 µs. After that the next command or a status request is sent by the CCB.

The second functionality of the SPI is to receive a status vector from the PDM-board. So far only three states are defined using only the last two bits of one SPI packet. Both bits on low means the PDM-board is idle. The 32nd bit on high indicates that an event is ready to be received by the CCB. The 31st bits contains information whether the PDM-board is sending (high) or not (low). When the CCB has no command for the PDM-Simulator it sends a status request and receives the status vector. After a maximum of 33 µs (assuming the status change takes place when the transmission of an outdated status just started) the CCB receives a change of status. To detect a malfunction of the PDM-board the commands xFFFFFFFF or x00000000 are reserved. If the board fails, the MISO will be stuck on high or low and the CCB will sample one of these commands and interpret this as error.

There are two ways to activate a data transmission. Either the PDM-board issues an L1-trigger or the PDM-board receives a broadcast signal from the CCB in case a neighboring PDM-board

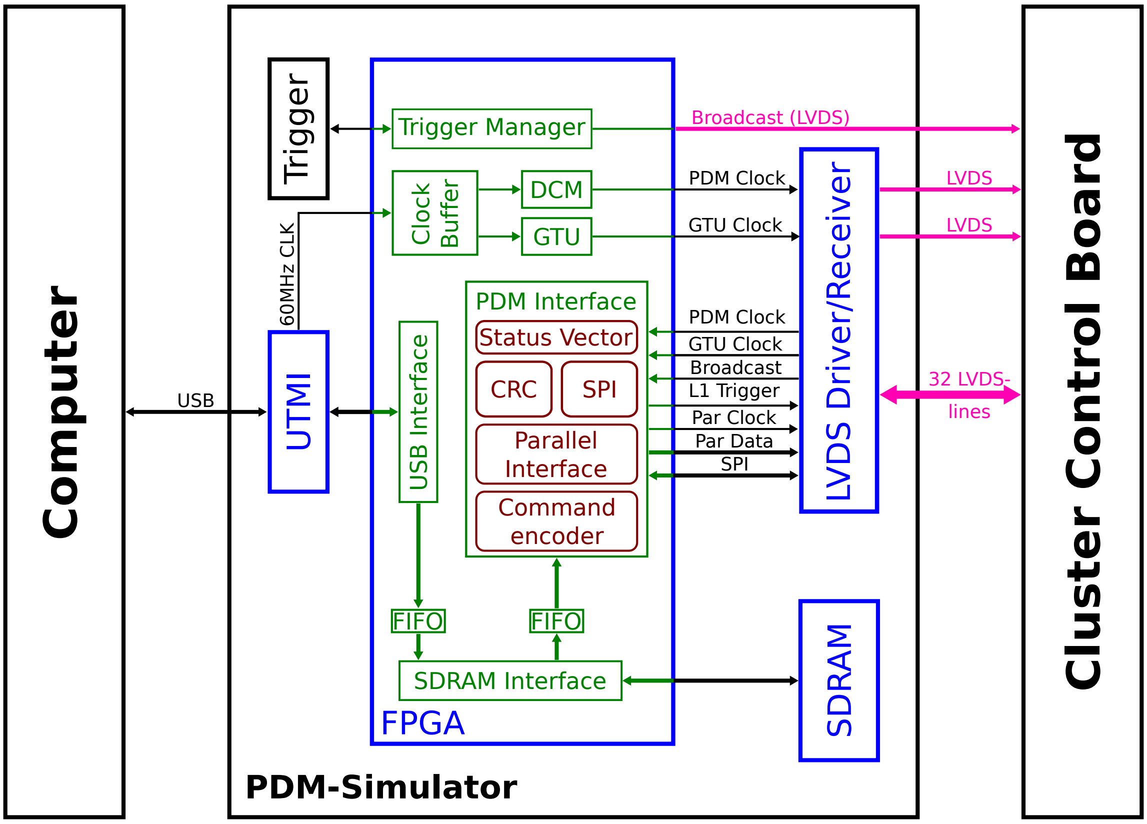

Figure 3.8: Block diagram of the PDM-Simulator: From left to right the computer providing the simulated data, the PDM-Simulator and the CCB are shown. The computer interfaces with the PDM-Simulator over USB. The UTMI chip also provides the 60 MHz system clock for the PDM-Simulator. This clock is connected to a global clock buffer of the Spartan 3, to a DCM to generate a 40 MHz clock and to a counter to provide the GTU clock. The 40 MHz clock, the GTU clock and the broadcast signal generated by the button here called *Trigger* are connected to the CCB. This is the clock board functionality of the PDM-Simulator. The data coming from the USB is buffered by a FIFO buffer and stored to the SDRAM. From there the PDM interface can request the data and after another FIFO the CRC is calculated and the data transmitted. For the PDM interface the clock coming from the CCB is used although the clock originally comes from the PDM-Simulator.

triggered it. The PDM-Simulator can either issue an L1 trigger if told so via USB or react on a broadcast signal like the PDM-board does. The real PDM-board freezes the data after a predefined exposure time and prepares the data after an L1 trigger or a broadcast signal. In both cases the PDM-Simulator reads the data from the SDRAM and fills the outgoing FIFO.

After the data is ready the status vector is changed. The CCB recognize this status change and issues a *data send* command via SPI. After the *data send* the interface component reads from the outgoing buffer, includes the read bytes in the CRC calculation (see 2.4.2) and generates a parallel clock. Together with the parallel clock the data is put on the outgoing parallel signals. The data is valid on rising edge of the parallel clock. The CRC is added to the data stream after the header and every GTU time frame by using counters.

In the design described here and used in the laboratory tests, the CCB sends either a *data acknowledge* signal after the data transfer is complete and the CRC calculation was successful or a *data error* after a failed CRC calculation. This can happen in the middle of the data stream. The transmission breaks up and the PDM-Simulator resets the outgoing buffer and reads the event again from the SDRAM. Unfortunately, in the case of the EUSO-TA experiment the data is lost after sending it because of the storage technology used on the real PDM. In the current CCB design, frames in which the CRC fails are marked, but the transmission does not stop.

After a successful transmission the PDM-Simulator waits for new data or a broadcast signal to send the stored data again. As long as the power supply is stable, the data on the SDRAM is not lost until a new event replaces the data. This makes long term tests of the dead time during transmission and stability of the data transmission with respect to the rate of occurring bit flips possible (see Chapter 4).

The PDM-Simulator receives a GTU clock from the CCB. As stated before, this clock has a period of 2.5 µs and is used to measure the lapsed time by increasing a counter. The interface components start every data transmission with a packet type byte and a four bytes long time stamp. By using the information from the Clock Board a reconstruction of the exact time of a recorded event is done during the analysis.

It is important to note that the interface component uses a clock provided by the CCB. The 40 MHz clock is part of the PDM-board to CCB interface. The CCB receives its clock from the Clock Board and sends it to the PDM-board. Although the PDM-Simulator can take over the Clock Board functionality (see Subsection 3.6.5), it uses the clock from the CCB to achieve a comparable setup for the laboratory tests. Therefore the clock for the interface is generated by the PDM-Simulator, sent to the CCB, sent back to the PDM-Simulator and used as interface clock.

3.6.5 Clock Board functionality

There is no oscillator on the CCB for clock generation. To test the board without the Clock Board an external function generator and an LVDS driver was used but the clock was not stable enough for the CCB. From the 40 MHz input clock the CCB generates a 100 MHz system clock and a 200 MHz SpaceWire clock. This generation is not tolerant to little shifts of the input clock. In addition, a broadcast signal and the GTU clock have to be fed to the CCB. To reduce the equipment needed for integration sessions in other laboratories and to provide a stable clock, it was decided to use the PDM-Simulator board as a replacement for the Clock Board as well.

The PDM-Simulator is supplied with a 60 MHz clock from the UTMI chip. This clock is used as an input for a DCM. By using a clock divider of 1.5 a 40 MHz clock is generated. This clock is put on one of the auxiliary lines of the PDM to CCB connector, but re-soldered to a LEMO connector. For two of the auxiliary lines LVDS drivers are placed on the board. The GTU clock is generated with a counter and put on the second auxiliary line and from there to another LEMO connector.

The broadcast signal can be issued by pushing a button on the PDM-Simulator. By pushing the button an input pin is pull from high to low. This change is sampled by the FPGA. The FPGA generates a broadcast signal and waits for 1 s before sampling the input again. By keeping the button pushed, the PDM-Simulator issues a broadcast signal with 1 Hz.

This interval can be changed in the configuration to reach higher broadcast rates for example another mode implemented issues a broadcast signal with 10 Hz after pushing the button once. For the broadcast signal an internal LVDS driver of the Spartan 3 chip is used. The LVDS lines are routed to the general purpose pins on the bottom of the board close together. Because these lines were never planned to be LVDS lines, there are some difference in length and distance between the lines. One of the LVDS pairs is used for the less timing sensible signal. A LEMO connector is soldered to this pins as well. In the end three LEMO connectors are placed on the front pane. The signals are connected to the CCB using a cable with LEMO connectors on one side and the Clock Board connector on the other.

4 Test-Setup and Integration for EUSO-TA

4.1 Overview

The original plan to build a test setup in Tübingen for the CCB was delayed because of a change in the schedule of the prototype of the JEM-EUSO hardware on the site of Telescope Array (EUSO-TA). EUSO-TA is a field test for the JEM-EUSO hardware planned for the end of April 2013. The complete chain of electronics from the MDP over the CCB down to one PDM is planned to be placed at the TA site in Utah including a smaller prototype of the optics. TA is a UHECR detector using fluorescence detectors and particle detectors on ground. The fluorescence detectors feature a trigger algorithm to identify an EAS in the FoV. The plan is to use the TA trigger as an external trigger for the EUSO-TA setup. A mean trigger rate of 3 Hz is expected for TA's fluorescence detectors. For EUSO-TA the use of internal hardware triggers as for JEM-EUSO are not foreseen. Three integration sessions were scheduled to integrate and test the entire hardware of EUSO-TA.

In the complete development process the PDM-Simulator played an important role to replace unavailable hardware in various locations. Subsections 4.3, to 4.6 describe the different steps in the integration process. Subsection 4.2 gives an overview of the Tübingen laboratory test setup and the tests performed so far.

Another path-finder of JEM-EUSO is the so-called EUSO-Balloon. A similar setup as for EUSO-TA is deployed in a balloon gondola to fly at $\sim 40\,\mathrm{km}$ above ground looking down to the atmosphere. The trigger algorithms are mandatory for this test. While the design of the EUSO-Balloon has been completed, the first steps for the integration of EUSO-Balloon are taking place right now. The launch is scheduled for mid 2014.

Before concentrating on the integration for EUSO-TA, a first test of the interface using single ended signals instead of LVDS failed because of the signal quality. Unfortunately, only one ground connection was used between the two boards due to limitations of the cable. This lead to unstable signals on some lines. The CRC failed after the header or sometimes after the first frame. This triggered the development of a dedicated base board for the PDM-Simulator to be able to use LVDS for all interface signals.

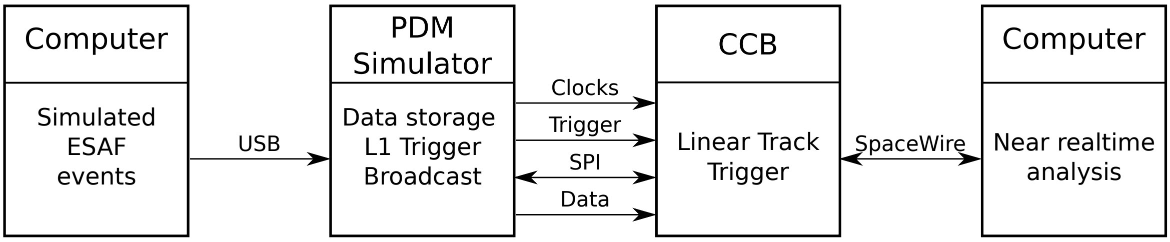

Figure 4.1: Schematic of the laboratory setup: From left to right the Computer providing the simulated data, the PDM-Simulator, the CCB and the Computer reading the data from the CCB are pictured. The first computer is connected to the PDM-Simulator via USB. Between the CCB and the PDM-Simulator the interface is plotted. The CCB is connected to the second computer with a SpaceWire link.

4.2 Test setup at the Institut für Astronomie und Astrophysik Tübingen (IAAT)

Figure 4.1 is a schematic of all involved hardware and the connections between them. The Low Voltage Power Supply (LVPS) is replaced by two separated power supplies for the CCB and the PDM-Simulator. The CCBs power supply is a sensitive laboratory equipment adjustable down to 10 mV. The PDM-Simulator's power supply is a commercial product, not very precise but due to the power converters onboard it is sufficient. The PC provides a SpaceWire connection to the CCB and replaces the real CPU in the MDP. It also is connected to the PDM-Simulator and provides the simulated ESAF data or a dummy data packet. The PC provides near real time visualization tools for the data from the CCB and is planned to be used for a quick look on the acquired data from TA and Balloon. The SPI to USB converter in combination with a C program can read the housekeeping data from the CCB. It can also issue a soft reset. The PDM-Simulator is a replacement for the PDM and the Clock Board. It uses the same connectors to the CCB as the real boards.

Several tests were done prior to the integration sessions. Before the integration with the MDP the main focus was on the SpaceWire connection. A lot of effort was put in the 13 Byte command and status messages. After several error corrections this part worked perfectly. Another topic was the request of dummy data transfer from the CCB using a SpaceWire command from the PC. The first tests were promising. But after making some changes in Naples, the dummy data did not work because of timing problems. This was solved shortly after the integration session with the MDP by optimizing the timing. Additional buffers were added, which increased the clock latency, but after this correction the data was transferred correctly and complete.

After sorting out the problem with the SpaceWire connection the PDM-board to CCB interface was tested. Commands were sent to the PDM-Simulator several times and without errors. Prior to the integration session with the PDM the main objective was a working interface with our setup to be used from this point on. Before leaving for Orsay the interface was stable and a test with a trigger rate of 10 Hz was conducted. For 3000 events the errors and acknowledges were counted on the PDM-Simulator and displayed on the PC using the JTAG debug interface and *Chipscope Pro*. For the entire 3000 events no error occurred.

This test was only conducted between the PDM-Simulator and the CCB. The trigger rate of 10 Hz exceeds the estimated trigger rate for EUSO-TA (3 Hz) and for the estimated first level trigger rate of the PDM (1 Hz). At this time the SpaceWire software on the PC was not stable. After one successful data transmission, the software crashed. This bug is solved but the PDM to CCB cable is currently not available to redo the test.

During the integration with the PDM all changes necessary were implemented and tested on the PDM-Simulator as well. Using the behavior for EUSO-TA, the PDM-Simulator is tested and functional for further tests of the LTT or the updated behavior of the interface for EUSO-Balloon. In conclusion, the laboratory setup fulfills all needs for tests necessary for the balloon, the update for the CCB to PDM interface and the L2 trigger algorithm.

4.3 Preintegration of the SpaceWire interface between the CPU and the CCB

Between the 24th and the 28th of September a first test of the SpaceWire interface between the CPU and the CCB was conducted in Tübingen. A preliminary structure of the data, command and status packets was defined including the format of the Cyclic Redundancy Check (CRC). Figure 4.2 shows a picture of the setup for the preintegration in Tübingen.

The first transmission tests between the CPU and the CCB were performed as well, starting with an echo test and a small mock packet exchange. With a loop back test from the CPU the performance of the SpaceWire interface was measured. A constant data rate of $\sim 18\,\mathrm{MB/s}$ was achieved. For a transmission test from the CCB to the CPU the measured data rate was 17.11 MB/s.

A list of message types and a list of commands have been defined. For further testing the CCB's VHDL implementation has been successfully synthesized in a Xilinx Spartan-6 board to be used in Rome as a CCB emulator. A summary of the integration session can be found in A.2.

4.4 Integration between CCB and MDP in Naples

The first step of the EUSO-TA integration took place in Naples in November 2012, integrating the CCB, the MDP and the housekeeping board. The PDM-Simulator provided a replacement of the real PDM-board. In advance of the integration session a laboratory test of the data interface between CCB and PDM-Simulator in Tübingen was not possible due to the higher priority of

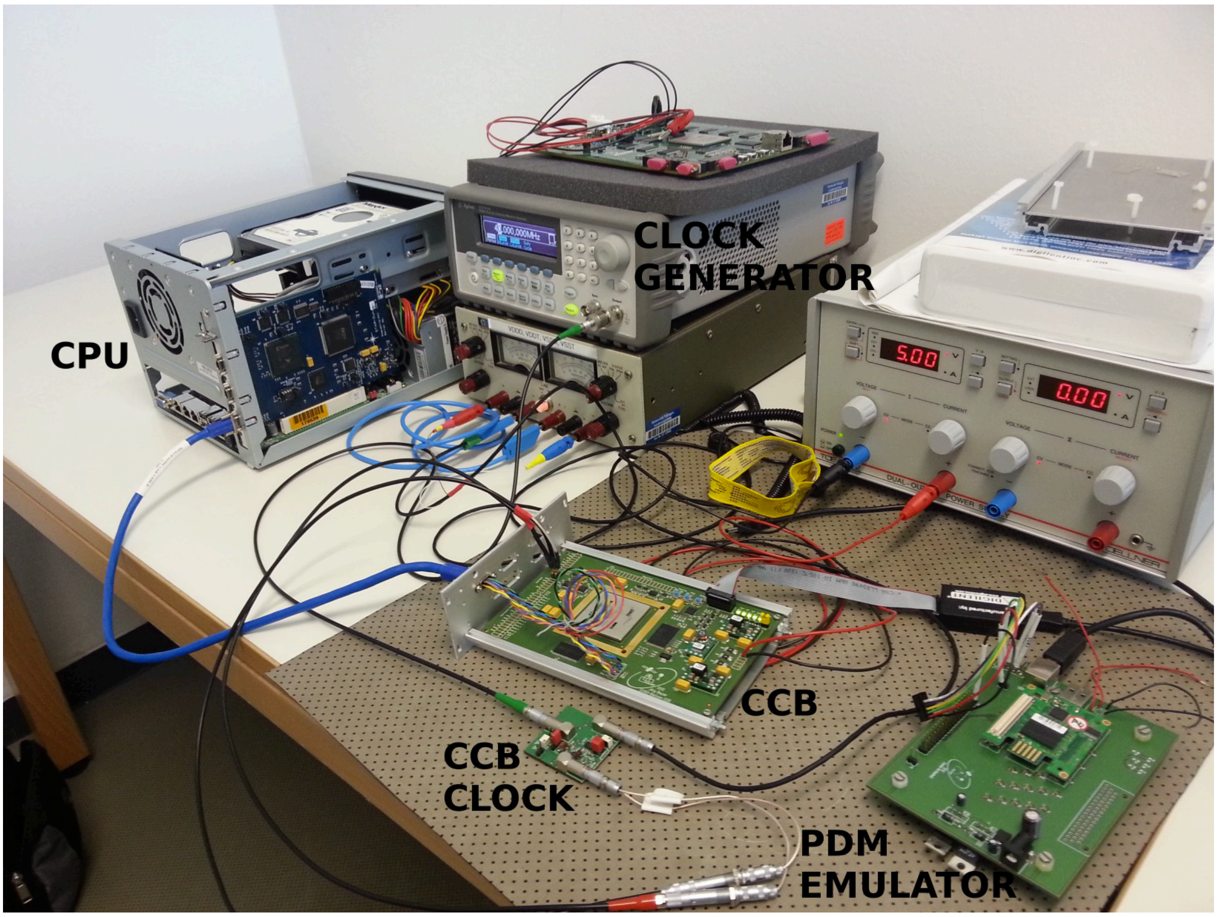

Figure 4.2: Picture of the setup of the CPU to CCB preintegration.

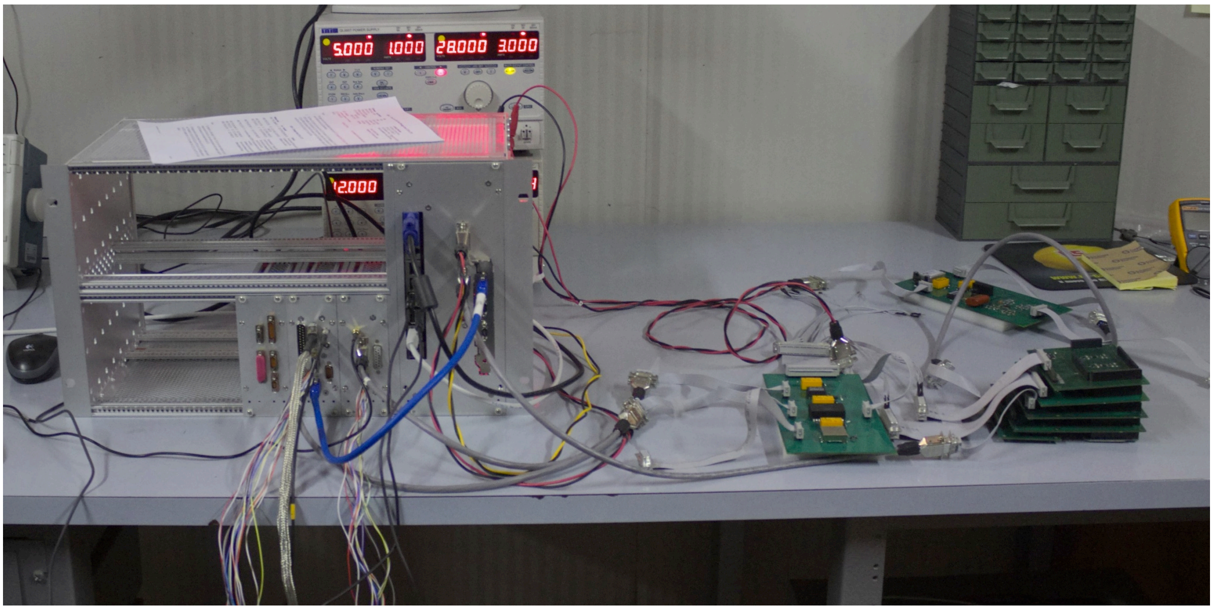

Figure 4.3: Picture of the setup in Naples: In the rack in the lower row from left to right is the CCB, the clock board and the GPS board mounted. Next to it on the right the CPU is placed over two rows. On the table next to the rack the LVPS and housekeeping boards are laying. Behind the setup the power supply is visible. The PDM-Simulator will be connected to the CCB to the connector with the red protection. The casing of the PDM-Simulator is constructed in a was so it can be also mounted to this rack.

the SpaceWire connection to the MDP.

Figure 4.3 is a picture of the setup in Naples. The mechanical integration caused no problems. The CCB is mounted in an industrial standard module for a rack of different boards. The cable connections were tested as well and worked apart from the housekeeping cable that had to be soldered in Naples.

One open topic for EUSO-Balloon is the thermal management. Because of the thin atmosphere the thermal dissipation is only radiative and the current cooling will not be sufficient. Studies about the thermal dissipation are going on.

The first test with the Clock Board was successful. The clock provided by the board was translated to the 100 MHz system clock and the 200 MHz SpaceWire clock. Also timestamps of the SpaceWire packets were tested by using the GTU clock provided by the Clock Board.

After some minor adaptions to the housekeeping board it was possible to read out all necessary values from the CCB using the SPI between the housekeeping board and the CCB. The generation of the alarm vector was still an open topic that was defined and implemented during the integration. The LVPS controlled by the housekeeping board was successfully tested for the CCB.

While testing the housekeeping interface a problem with the reset signals was discovered. The soft reset and the hard reset are directly connected to the resets of the CCB. After a hard reset is issued the FPGA reloads the firmware from the PROM and is deactivated for some time. The soft reset brings the FPGA in a safe state after aborting any operations. While the housekeeping board was not powered or the cabled was unplugged, the signal lines began to float. The CCB was not operational in these cases because of continuous resets.

A comparable problem came up with the broadcast signal from the Clock Board. At first the broadcast signal was inactive on the Clock Board which resulted in a great number of broadcast signals to the CCB. The CCB tried to receive data from the PDM-board. This lead to a number of errors that could not be explained at first. These problems are targeted for EUSO-Balloon by introducing fail safes to the LVDS receivers.

The LVDS drivers generate a signal by putting a current on the two lines. In case of no current the fail safe guarantees a stable signal either on low or high. The design for the PDM-Simulator does not have these problems because external receivers were used which have a fail safe already implemented. At first the logic analyzers were not sensitive enough to locate this problem, but after reconfiguring, the problem was obvious.

Status and command messages between the CCB and CPU were transmitted perfectly and a dummy packet requested by the CPU from the CCB was transmitted as well. A chain without the PDM-board or the PDM-Simulator was built up and after a broadcast signal induced by a function generator, the CPU was able to ask for data and receive it from the CCB. But after checking the data, it turned out to be incomplete. After a great number of tests using a logic analyzer it was shown that there was a timing problem with the SpaceWire connection. Parts of the data packet got lost on the CCB. This problem could not be solved in the short time of the integration session. After further testing in Tübingen it was possible to solve this problem. Additional buffers for the data were added and the handshakes between the different components of of the CCB were optimized.

One goal of the integration was to test the chain with the PDM-Simulator. Due to the problems with the SpaceWire interface this could not be done. But after finding a logic error in the SPI of the CCB, it was at least possible to send commands to the PDM-Simulator from the CPU.

In the meantime the SpaceWire software of the workstation with the near real time analysis software was optimized to match the performance and functionality of the real CPU. In addition, a visualization of acquired data for the German PC was tested with simulated ESAF data. In the end the German PC should provide near real time analysis capability.

In conclusion, the first integration session was successful. It gave the possibility to work on the flaws in the design and it provided important information for the second design revision for EUSO-Balloon like the fail safe for the LVDS receiver. From a PDM-Simulator point of view the integration was also a success because the command interface was tested and worked the first time. As a result of the integration the redesign for the PDM-Simulator to provide Clock Board functions was planned and implemented ahead of the integration session with the PDM-board. It was foreseen that the Clock Board would not be available for this second integration session.

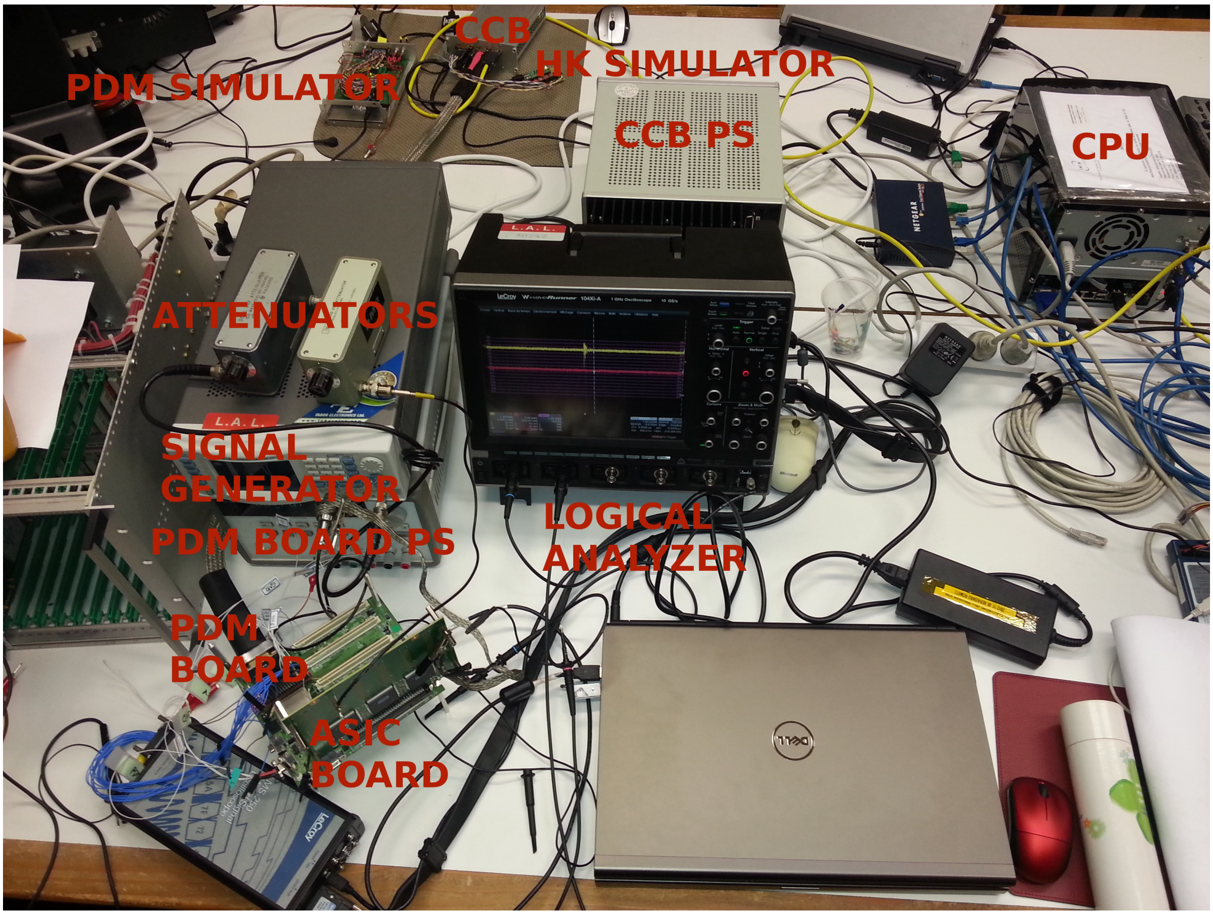

Figure 4.4: Picture of the setup of the CCB and PDM integration: On the bottom left the PDM, ASIC board and the function generator for the charge injection are pictured. For debugging reasons a logic analyzer is connected to the PDM. On the right a prototype of the CPU is placed with the SpaceWire connection to the CCB on top of the picture. Connected to the CCB is the PDM-Simulator working as a Clock Board replacement. In addition, a PC is used as a logic analyzer connected through the JTAG.

4.5 Integration between CCB and PDM at the LAL

At the Laboratoire de l'Accélérateur Linéaire (LAL) the second integration between CCB, CPU and PDM-board took place. Figure 4.4 shows a picture of the setup with all involved hardware. The goal was to have a working command interface using SPI, the parallel interface for data and a successful data recording after a broadcast signal.

Before the CCB arrived, the integration between the SPACIROC and the PDM-board took place. A comparable setup was tested in Tübingen using the near real time analysis workstation and the PDM-Simulator together with the CCB. During the entire integration the PDM-Simulator worked as a replacement for the Clock Board. It provided the system clock and GTU clock and was able to provide a broadcast signal by pushing a button. This was the first occasion where the CCB and the real PDM board were actually connected. The first step of the integration was to discuss the test chain with the PDM-Simulator and especially the timing.

From the CCB side it was foreseen that the SPI is sending continuous commands to the PDM-board. It was expected that the PDM-board is answering with the status most of the time interrupted by the acknowledgment of configuration commands for the PDM-board. The PDM-board needs time to analyze the commands from the CCB and act on them. It was requested that after every 32 bit there is an interrupt in the SPI for 50 ns to 100 ns. The PDM developer did not expect a continuous request for the status of the PDM. It took two days to implement these changes on both boards, after that the first successful commands were transmitted and acknowledged by the PDM-board. These tests were successful using the real CPU and the workstation with SpaceWire card from Tübingen.

After establishing a command interface, the parallel interface for the data had to be tested. The CRC was not implemented on the PDM-board which lead to transmission errors on the CCB. After turning the CRC off the timing of the data interface was not correct, but it was possible to request a dummy data packet from the real PDM-board. After analyzing the received data it was possible to account for all problems. The result was a stable data connection tested several times.

The next step was to connect the EC-board to the PDM. The first test was to change the thresholds for the photon trigger. This could be measured directly on the EC-board. After that, a charge was injected in the EC board that should trigger a signal for one pixel. This signal was transmitted to the CPU and analyzed. Apparently the test was successful, but the data was not consistent with what was expected. Instead of one activated pixel, the two neighboring pixels were active. But after several tests of the CCB design, the conclusion was that the interfaces were working.

This integration session ended successful apart from the CRC that was left to be implemented before the next integration session. After understanding the data handling on the PDM-board, several changes had to be made for the CCB and in parallel for the PDM-Simulator. To reproduce the behavior in the laboratory, the SPI was altered to fit the behavior of the real PDM-board. Because the real PDM-board uses FIFO buffers for the data, it is not possible to read an event again after a transmission error. After reading data from a FIFO it is lost. This makes a data error message from the CCB to the PDM-board pointless because it is not followed by any action. It was decided to finish the transmission of a data packet in any case and mark the frames for which a CRC error occurred. These major redesigns are still under development and being tested, for EUSO-TA the old design will be used. For the old design events with a CRC error are lost.

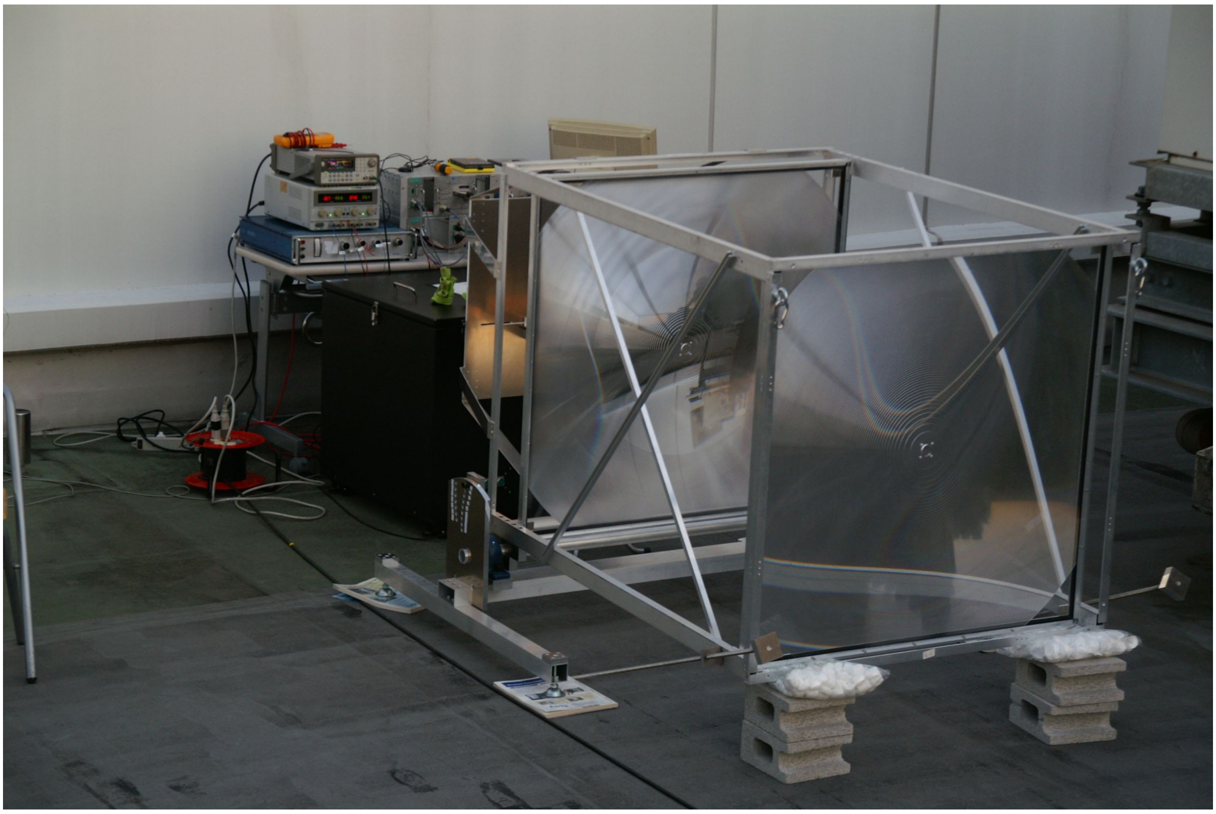

Figure 4.5: Picture of the setup on the roof of RIKEN: On the table the setup with all components connected is shown. The optic for EUSO-TA is seen in front of the picture pointing at a mirror for the laser to test the read out.

4.6 Final Integration in Japan

For the last integration session the PDM board and the CCB were sent to RIKEN in Japan. The PDM-Simulator was again part of the integration as a Clock Board replacement and as an external trigger. At first the connection between the CCB and PDM-board was tested again, new errors occurred and had to be sorted out.

This was done in the first few days, later on also including the CRC. The goal for this integration was to integrate a simplified optics for the TA setup and the MAPMTs with the focal surface electronics and the MDP. After establishing a reliable interface between CCB and PDM-board, one MAPMT was calibrated.

On top of the roof of the RIKEN facility the sky was recorded at night. Figure 4.5 shows the setup including the optics. In total 5000 external triggers were used to record 5000 events. In the process no transmission error occurred. After that another set of 5000 events was recorded without error. The interface developed during this work proved to be reliable and stable. In addition, a test using a laser was done, also successfully. A laser was pointed to a mirror and reflected back to the setup. The laser was moving to see a change in the exposer. This setup is

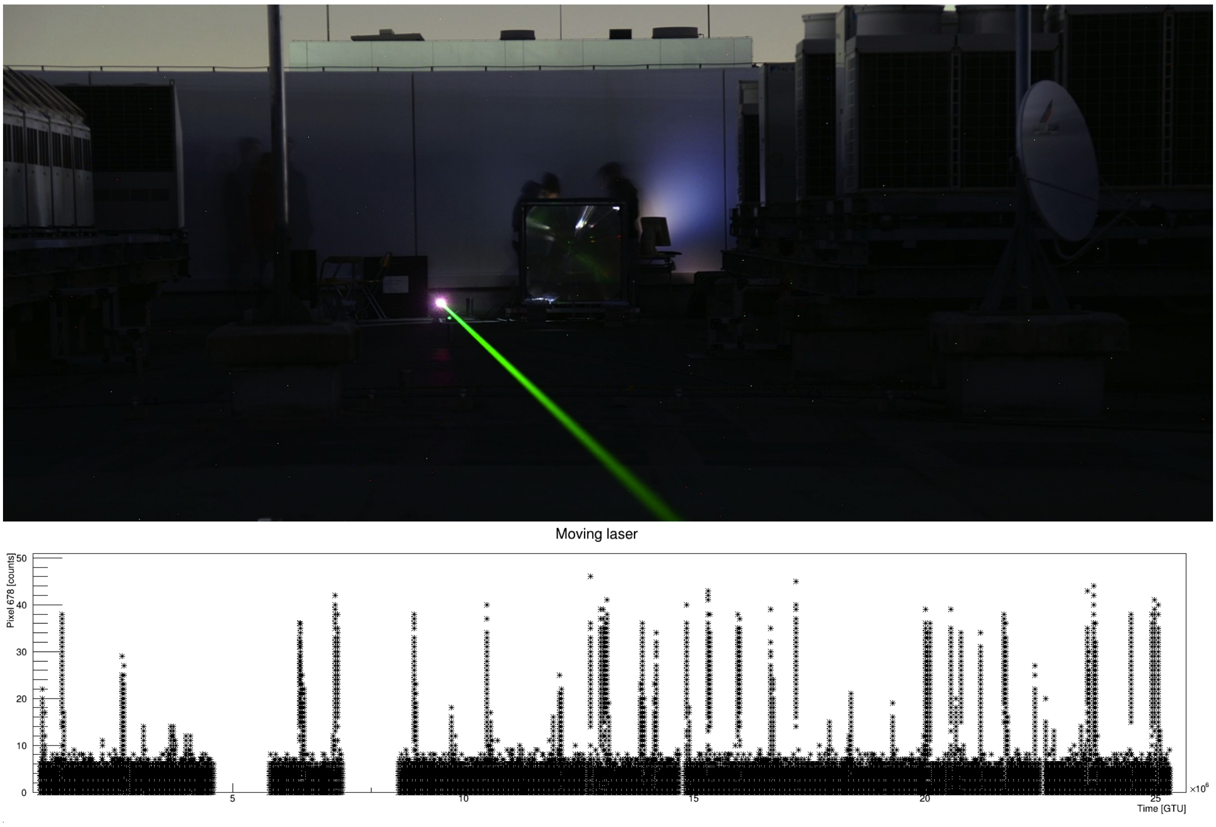

Figure 4.6: Picture of the test with a laser on the roof of RIKEN: On top a picture of the laser pointing to the mirror and reflected back to the setup is shown. The graphic shows the counts for one pixel over 30 GTU. The laser was moving to see a change in the exposer.

shown in Figure 4.6

Prior to the EUSO-TA campaign three additional MAPMTs had to be integrated and calibrated. This work was still under way when the participant from Tübingen left Japan. From a CCB point of view this integration proved to be very successful. For now the design is frozen, changes described in Subsection 4.5 are implemented for the Balloon.

5 Conclusions

In the context of the Diplomarbeit summarized in this thesis a PDM-Simulator board, to be used in laboratory and field tests of the detector prototype of the JEM-EUSO mission was designed, assembled, tested and extensively used. The PDM-Simulator has been used in fact in many different type of tests and has been improved and adapted to fit the experiment needs in a very flexible way. The current version can be used as a replacement of the PDM board to feed the JEM-EUSO CCB with simulated data in order to test and optimize the interface between the various components the CCB is connected to, and to verify the performance of second level trigger. The PDM-Simulator has proven to be an essential tool during the integration sessions performed for EUSO-TA. During the tests with the MDP, it was possible to conduct the first tests of the command interface to the PDM without the actual PDM hardware. This is a major point since the command relay from the CPU via the CCB is one of the key aspects of the functionality of the CCB. Moreover, during the integration of the PDM with the CCB, the PDM-Simulator replaced the clock board and provided external triggers with no major problems. In addition, the functioning interface between the CCB and the PDM-Simulator gave the possibility to simulate the behavior of the interface expected by the CCB allowing the preparation of further improvements. This made the integration much easier.

The key feature of the PDM-Simulator is the possibility to test the LTT of the CCB in a laboratory environment without the actual implementation of the focal surface electronics. Simulated events from the EUSO Simulation and Analysis Framework (ESAF) can be transmitted to the PDM-Simulator via USB and stored to a SDRAM. By using an interface identical to the real one between the PDM board and the CCB, the data is transmitted to the CCB. The setup is functioning properly and will be very soon used to test the LTT performances. The interface was tested extensively by conducting a long term test with a trigger rate exceeding the requirements. 3000 events were transmitted without a single error.

In the near future, two new CCB boards will be produced for the EUSO-Balloon experiment. The PDM-Simulator will ease the task to test these new boards just in the laboratory setup.

The design of the PDM-Simulator meets all requirements defined in the context of the JEM-EUSO phase A study and exceeds them by adding the additional clock board functionality. From the system clock of the PDM-Simulator the 40 MHz clock for the CCB and a 400 kHz GTU clock is generated and is fed to the CCB.

In spite of the excellent performance of the current version, producing a second version of the base board to correct minor flaws is a foreseen continuation of this work. In addition, the VHDL code could benefit from optimizations especially of the SDRAM interface. To test the behavior of the LTT in the more general case of two PDMs, the design of the simulator has also to be extended to a second version. The SDRAM's performance is sufficient and the FPGA has still enough resources left to host two PDM to CCB interfaces.

The next important step of the JEM-EUSO project, is the launch of the EUSO-Balloon in Summer 2014. The integration of the instrument begins in March 2013. One of the open key topics is the implementation and test of the LTT on the CCB. In particular, it must be still clarified whether the pixel map of the PDM meets the expectation of the current design of the LTT or whether the pixels have to be resorted on the CCB. Also the implementation of the L1 trigger on the PDM-board is another key topics to be tested in the near future.

In view of developing the JEM-EUSO mission, successful results from the path finder EUSO-TA and EUSO-Balloon are crucial. Hopefully, after finishing these two campaigns the work for JEM-EUSO can continue with the goal of a launch in 2017.

Bibliography

Abbasi, R.U.; Abu-Zayyad, T.; Allen, M.; Amann, J.F.; Archbold, G.; et al., 2010: Analysis of Large-scale Anisotropy of Ultra-high Energy Cosmic Rays in HiRes Data. *The Astrophysical Journal Letters*, 713:L64–L68.

Abbasi, R.U.; Abu-Zayyad, T.; Allen, M.; Amman, J.F.; Archbold, G.; et al., 2008: First Observation of the Greisen-Zatsepin-Kuzmin Suppression. *Phys. Rev. Lett.*, 100:101101.

Abraham, J.; Abreu, P.; Aglietta, M.; Aguirre, C.; Allard, D.; et al., 2008: Correlation of the highest-energy cosmic rays with the positions of nearby active galactic nuclei. *Astroparticle Physics*, 29(3):188 – 204. ISSN 0927-6505.

Adams Jr., J.; Ahmad, S.; Albert, J.N.; Allard, D.; Ambrosio, M.; et al., 2013: An evaluation of the exposure in nadir observation of the JEM-EUSO mission. *Astroparticle Physics*, 44(0):76 – 90. ISSN 0927-6505.

Ahmad, P.; Plin, S.; Dagoret, S.; Dulucq, F.; de La Taille, C.; et al., 2010: JEM-EUSO ASIC: SPACIROC measurements results.

Aloisio, R.; Berezinsky, V.; Kachelriess, M., 2006: Status of superheavy dark matter. *Phys. Rev. D*, 74:023516.

Axford, W.I.; Leer, E.; Skadron, G., 1977: The acceleration of cosmic rays by shock waves. International Cosmic Ray Conference, vol. 11 of *International Cosmic Ray Conference*, pages 132–137.

Bayer, J., 2011a: Development of a Cluster Control Board for the JEM-EUSO Mission.

Bayer, J., 2011b: The Cluster Control Board of the JEM-EUSO mission. International Cosmic Ray Conference, vol. 3 of *International Cosmic Ray Conference*, page 168.

Bell, A.R., 1978: The acceleration of cosmic rays in shock fronts. I. *Monthly Notices of the Royal Astronomical Society*, 182:147–156.

Bittermann, C., 2010: Simulation of UHE Neutrinos for the JEM-EUSO Mission.

Blandford, R.D.; Ostriker, J.P., 1978: Particle acceleration by astrophysical shocks. *The Astrophysical Journal Letters*, 221:L29–L32.

Casolino, M.; The Pamela collaboration, 2009: The Pamela Cosmic Ray Space Observatory: Detector, Objectives and First Results. *ArXiv e-prints.*

Distratis, G., 2012: The interface between CCB and PDM.

European Cooperation for Space Standardization, 2008: ECSS-E-ST-50-12C - SpaceWire - Links, nodes, routers and networks.

Fermi, E., 1949: On the Origin of the Cosmic Radiation. *Physical Review*, 75:1169–1174.

Gebhardt, H., 2009: Development of Data Acquisition and Detector Controller Electronics for the Low Energy X-Ray Detector of the Simbol-X Space Mission.

Greisen, K., 1966: End to the Cosmic-Ray Spectrum? *Phys. Rev. Lett.*, 16:748–750.

Hamamatsu Photonics, 2006: Photomultiplier Tubes: Basics and Applications.

Hanlon, W.F., 2008: The Energy Spectrum of Ultra High Energy Cosmic Rays measured by the High Resolution Fly's Eye Observatory in Stereoscopic Mode. Ph.D. thesis.

Hillas, A.M., 1984: The Origin of Ultra-High-Energy Cosmic Rays. *Annual Review of Astronomy and Astrophysics*, 22:425–444.

Hinton, J., 2004: The status of the HESS project. *New Astronomy Reviews*, 48(5â€“6):331 – 337. ISSN 1387-6473. 2nd VERITAS Symposium on the Astrophysics of Extragalactic Sources.

JEM-EUSO Collaboration, 2010: Report on the Phase A study.

JEM-EUSO Collaboration, 2012: The JEM-EUSO Mission: Status and Prospects in 2011. *ArXiv e-prints.*

Koopman, P., 2004: Cyclic Redundancy Code (CRC) Polynomial Selection For Embedded Networks.

Krymskii, G.F., 1977: A regular mechanism for the acceleration of charged particles on the front of a shock wave. *Akademiia Nauk SSSR Doklady*, 234:1306–1308.

Lemoine, M.; Sigl, G., 2001: Physics and astrophysics of ultra high energy cosmic rays.

Longair, M.S., 2011: High energy astrophysics, Cambridge Univ. Press, Cambridge [u.a.], 3. ed., 1. publ. edn. ISBN 978-0-521-75618-1.

Matthiae, G., 2008: The Auger Experiment Status and Results. M. Barone; A. Gaddi; C. Leroy; L. Price; P.G. Rancoita; R. Ruchti, editors, Astroparticle, Particle and Space Physics, Detectors and Medical Physics Applications, pages 229–238.

MEDINA TANCO, G.A.; DE GOUVEIA DAL PINO, E.M.; HORVATH, J.E., 1998: Deflection of Ultra-High-Energy Cosmic Rays by the Galactic Magnetic Field: From the Sources to the Detector. *The Astrophysical Journal*, 492(1):200.

MERNIK, T., 2009: Reconstruction of UHECR Events for the JEM-EUSO mission.

SEO, E.S.; AHN, H.S.; BEATTY, J.J.; COUTU, S.; CHOI, M.J.; ET AL., 2004: Cosmic-ray energetics and mass (CREAM) balloon project. *Advances in Space Research*, 33:1777–1785.

SIMPSON, J.A., 1983: Elemental and Isotopic Composition of the Galactic Cosmic Rays. *Annual Review of Nuclear and Particle Science*, 33:323–382.

SOKOLSKY FOR THE HIRES COLLABORATION, P., 2010: Final Results from the High Resolution Fly's Eye (HiRes) Experiment. *ArXiv e-prints*.

TAMEDA, Y.; TAKETA, A.; SMITH, J.D.; TANAKA, M.; FUKUSHIMA, M.; ET AL., 2009: Trigger electronics of the new Fluorescence Detectors of the Telescope Array Experiment. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 609(2â€“3):227 – 234. ISSN 0168-9002.

TENZER, C.; SANTANGELO, A., 2013: Beobachtungstechniken der Astrophysik.

TESHIMA, M., http://www.asi.riken.jp/en/laboratories/chieflabs/astro/images/fig2.jpg: Simulation of protons with different energies in the Galactic magnetic field.

UNSOELD, A.; BASCHEK, B., 2002: Der neue Kosmos, Springer, 7. ed. edn.

USB IMPLEMENTORS FORUM, 2000: Universal Serial Bus Specification.

ZATSEPIN, G.T.; KUZ'MIN, V.A., 1966: Upper Limit of the Spectrum of Cosmic Rays. *Soviet Journal of Experimental and Theoretical Physics Letters*, 4:78.

A Appendix

A.1 Notes on the logic for the JEM-EUSO Cluster Control Board

The following documentation is written by Guiseppe Distratis and describes all messages, commands and data packages for the SpaceWire interface, the housekeeping and the PDM to CCB interface.

EUSO-INST-CCB-IAAT 1.2

Notes on the logic for the JEM-EUSO Cluster Control Board

This document describes briefly the logic behavior of the CCB device.
Detailed technical information can be found in the Instrument Definition Document (IDD)

L1-Trigger and broadcast signals

When the PDM fulfills the criteria for a L1 trigger, it will assert the l1_trig signal high for the time of 2 to 4 cycles of *pdm_clk* (*50* to *100 ns*) and should freeze the acquired data into one of its buffers.
The download procedure will be started by the CCB by sending the predefined command "CMD_DATA_REQUEST" to the PDM via the serial port (see below).
The PDM will either mark the GTU frame in which the L1 trigger has been issued with a flag bit in the frame header or will ensure that this frame will always be located at the same position in the data package (for instance as frame number 50 of 128).

PDM-Command interface and status register

The command interface to the PDM is based on a standard SPI and is used to send commands or configuration sequences to the PDM and to receive the content of the PDM status register.
The SPI has been currently configured for a 32-bit word and a spi_sclk frequency of 2 MHz, but can be easily re-configured to any length between 8 and 32 bits and a wide range of frequencies up to 20 MHz
Asserting HIGH[1] the *spi_ssel* strobe line initiates the activity on *spi_sclk* and therefore the command transmission. The SPI master (CCB side) shifts the command word LSB first to the *spi_mosi* line on the falling edge of *spi_sclk*, the SPI slave (PDM side) should sample these bits on the rising edge.

and at the same time shifts on the same edge the status register LSB first to the spi_miso line. Since the status register will most probably not contain as much bits as the command word, it will be padded to the same length with leading zeros.
By sending the "CMD_NULL" command, a binary zero, the CCB reads out the status register on a regular basis after 1 GTU of inactivity in order to keep the local copy of it up to date and to sense that the PDM (at least its SPI) is "alive". The PDM-Team could also provide a status bit to indicate the "alive" status of the PDM as a whole or other parts of it, for instance via toggling this bit on and off with every GTU clock.
The "CMD_NULL" command should be ignored by the PDM logic and must not have any effect on its operation.
The transition of the spi_ssel line to high level after completing the transmission will work on the SPI slave like an asynchronous reset and will keep the interface in a definite idle status. This way it is possible, to implement a command to soft-reset the rest of the PDM. The command CMD_RESET

[1] The widely accepted SPI quasi-standard for the SSEL-Line is LOW-Active. This deviation from the standard has been explicitly requested by the PDM developers. It can be accepted since the SPI will not be operated in Multi-Slave mode. In case it will be experimentally needed, the polarity and phase of any SPI signal may be changed on CCB and PDM accordingly.

has been defined for the time being as a full word of '1' but can be changed to any other value other than zero.

The status registers of the PDM will be stored into the CCB and read out by the HK system.

PDM-Data interface and data package

The event data are transferred to the CCB via the 8-bit wide, source synchronous data bus par_data working at 40 MHz. The clock line par_clk is active only when transmitting data, which will be switched to the bus by the PDM on the rising edge and sampled by the CCB on the falling edge.

After receiving the request command from the CCB, the PDM will start the data upload within a time lapse of *20 μs*. If the PDM does not comply within this time, the CCB will communicate the malfunction to the HK.

The data package is composed of one "Event Summary" (256 bytes data plus 2 bytes CRC16) followed by 128 GTU data frames.

Each GTU data frame consists of 2304 bytes (48x48 pixels, 8 bit data) for PMT data and 288 bytes KI data, then the 2 bytes of CRC16.

In case the CRC algorithm detects an error, the affected PDM will be instructed to stop the current transmission via the CMD_DATA_NAK command sent over the SPI. The PDM should keep the data and wait for another request from the CCB.

If all data are received correctly the command CMD_DATA_ACK will be sent and the PDM is allowed to clear the event buffer.

Assuming the size of 256 bytes for the Event Summary and 2 bytes for its CRC, 2304 bytes for the pixel data and 288 bytes for KI data and again 2 byte for the frame CRC, the total size of one event will be 256+2+128*(2304+288+2)=332290 bytes and it will take about 8.3 ms to transfer it over the PDM-Data bus.

HK-Interface

The communication to the HK-System is established over a SPI-Slave with 32-bit word length. It will transfer to the HK-System a selection of digitalized voltages, currents and temperatures as well as the contents of the PDM and CCB status registers.

CPU-Interface (SpaceWire)

The communication with the CPU has been realized using the free available SpaceWire Light VHDL-Core available at: http://opencores.org/project,spacewire_light

It has been configured for an Rx/Tx-frequency of 200 MHz, which translates into a TX data rate of max 200 Mbit/sec.

The complete data traffic between the CPU and the CCB as well as the PDM connected to it takes place over this interface in form of standardized data packets called "Messages".

The Event Message, which structure is described in Table 1, represents the highest traffic load.

All messages leaving the CCB are provided with a 4-byte large CRC32 in order to allow the CPU to assure their data integrity.

Table 1 – Structure of the Event Message from CCB to CPU

Element	**Sub-Element**	**Size**	**Content, Remarks**
MESSAGE TYPE		1 Byte	Full Event, Dummy Event, ...
DEVICE ID		1 Byte	10*CCB_ID + PDM_ID
TRIGGER TIME		4 Bytes, LSB First	GTU Number of the L1 Trigger for this Event
PDM EVENT SUMMARY		252 Bytes	
GTU FRAME #1		2628 Bytes total	Total Size of the GTU Frame
	PDM PIXELS	2304 Bytes	Matrix of 48x48 Pixels
	PDM KI DATA	324 Bytes	Size of the KI Data

GTU FRAME #		
… continuous stream of 128 frames		
GTU FRAME #128		
ERROR MAP	17 Bytes, LSB First	Bitmap with CRC error markers (bit 0 = Event Summary, bit 1 to 128 = GTU Frames). Bit values: 0 = CRC OK, 1 = CRC Error
CRC32	4 Bytes, LSB First	Error detection for the whole data packet

Notes:

- The trigger time in the event header is based on the GTU counter in the CCB that is reset at every Orbit/Run from Clock Distribution.
- The Device ID is given by the sum of the CCB ID number [0-19] multiplied by 10 and the PDM ID number [1-9] within the CCB. This provides a unique Device Identifier for every CCB and PDM of the future JEM-EUSO version. For the TA and Balloon versions the only used IDs are x"00" for the CCB and x"01" for the PDM.

Table 2 – Messages from CCB to CPU

Message Types	**Hex**	**Message Content**	**Notes**
MSG_TYPE_L1_TRIG	06	DEV_ID+GTU_TS	L1 Trigger received from PDM
MSG_TYPE_L2_TRIG	08	CCB_ID+GTU_TS	LTT issued L2 Trigger
MSG_TYPE_EVT_RDY	17	DEV_ID+GTU_TS	The Event (Real or Dummy) is ready for Download
MSG_TYPE_DUMMY	19	Full Event (s. Table 1)	Dummy Data
MSG_TYPE_FULL_EVT	24	Full Event (s. Table 1)	Event Data
MSG_TYPE_05	2A		Available code
MSG_TYPE_06	35		Available code
MSG_TYPE_07	3B		Available code
MSG_TYPE_CMD_ACK	42	ACKed Command	All commands except MSG_TYPE_GET_EVT
MSG_TYPE_CMD_NAK	4C	NAKed Command	Command NOT Acknowledged
MSG_TYPE_REG_CONT	53	DEV_ID+GTU_TS +REG_ID+CONTENT	Content of the Register REG_ID
MSG_TYPE_ERROR	5D	DEV_ID+GTU_TS+ERR	Error codes see Table 5
MSG_TYPE_PDM_STS	60	DEV_ID+GTU_TS+STATUS	PDM Status register
MSG_TYPE_PDM_RBK	6E	DEV_ID+CMD+MSG_ID	PDM Command Readback
MSG_TYPE_14	71		Available code
MSG_TYPE_15	7F		Available code

Notes:

- Every message type other than the Event Message consists of the Message Type Indicator (1 byte) followed by the 9 bytes of content (padded with x"00" if shorter) and the 4 byte wide CRC32.
- All messages other than the Event Message are queued in an output FIFO and sent to the CPU whenever the SpaceWire component is in an idle state.

Table 3 – Messages from CPU to CCB

Message Types	**Hex**	**Parameters**	**Remarks**
MSG_TYPE_COMMAND	83	DEV_ID+PDM_CMD+MSG_NR	The 4-byte wide PDM command ist just passed to the PDM
MSG_TYPE_SET_REG	8D	CCB_ID+REG_ID+CONTENT +MSG_NR	Set the CCB Registers ID see Table 4
MSG_TYPE_GET_REG	92	CCB_ID+REG_ID+MSG_NR	Get the CCB Registers
MSG_TYPE_MUTE_L1	9C	DEV_ID+STATE+MSG_NR	Stop receiving of L1 Triggers (State: zero=Off, non-zero=On, default=On)

MSG_TYPE_20	A1	DEV_ID+STATE +MSG_NR	Available code
MSG_TYPE_GET_STS	AF	DEV_ID+MSG_NR	Get the PDM Status Register
MSG_TYPE_MK_DUMMY	B0	CCB_ID+MSG_NR	Generates a dummy event into RAM
MSG_TYPE_GET_EVT	BE	DEV_ID+MSG_NR	Starts the event download from RAM
MSG_TYPE_24	C7		Available code
MSG_TYPE_25	C9		Available code
MSG_TYPE_26	D6		Available code
MSG_TYPE_27	D8		Available code
MSG_TYPE_28	E5		Available code
MSG_TYPE_29	EB		Available code
MSG_TYPE_30	F4		Available code
MSG_TYPE_31	FA		Available code
MSG_TYPE_SOFT_RST	FF	CCB_ID+MSG_NR	Software triggered DCM reset

Notes:

- The command as a fixed length of 8 bytes. The first byte always indicates the type of message sent and is followed by a 2-byte wide serial number, which is needed to identify the ACK/NAK answer but is not evaluated within the CCB.
- All messages should be addressed properly either with the CCB_ID (a number between 0 and 19 multiplied by 10 for the JEM-EUSO, just x"00" for the TA and Balloon version) or the PDM_ID (number from 1 to 9 to be added to the CCB_ID, just x"01" for TA and Balloon).

Table 4 – Operation registers

Registers:	**Hex**	**Function**	**Notes**
REG_SPW_DIV	00	SpW TX-Clock divider [1 Byte]	TX-Clk=(200 MHz)/(DIV+1)
REG_RAM_WS	01	RAM Waitstates	High Nibble=Write WS (Default=0) Low Nibble=Read WS (Default=1)

Table 5 – Error codes for MSG_TYPE_ERROR message

Error Codes:	**Hex**	**Meaning**
ERR_SPW_DISC	01	SpW Disconnect
ERR_SPW_PAR	02	SpW Parity
ERR_SPW_ESC	03	SpW Escape
ERR_SPW_CRED	04	SpW Credit
ERR_SPW_RXTO	05	SpW Receiver Timeout
ERR_SPW_TXTO	06	SpW Transmitter Timeout
ERR_PDM_DOWN	10	PDM not working
ERR_PDM_TO	11	PDM Timeout
ERR_NO_DATA	20	Requested Event NOT Ready

A.2 Minutes of the preintegration between the CPU and the CCB

EUSO-Balloon CPU/CCB Pre-Integration Report
Tübingen 24th-28th September 2012

Participants
University & INFN Rome - Tor Vergata: C. De Santis, N. De Simone, A. Pesoli
Universität Tübingen: G. Distratis, C. Tenzer, J. Bayer

Summary

- CPU/CCB setup (Fig. 1)
- preliminary transmission tests on the SpaceWire interface (echo, small mock packet exchange)
- discussion about data packet format sent from CCB to CPU. The following decisions have been made:
    - science data transmission from CCB to CPU is started by the CCB directly (no command exchange from CPU to CCB)
    - packet format re-defined in order to improve data-rate performance (all GTU frames transmitted together)
    - CRC-32 IEEE 802.3 (Ethernet)
    - estimated science data packet size ~ 332 KB
- CPU loop-back tests with data packet (data-rate ~18 MB/s).
- Data packet transmission tests from CCB to CPU. Several configuration parameters have been extensively tested (FIFO size, transmission speed, transmission delay). Few bugs have been fixed. In the standard nominal configuration (200 Mbit/s CCB TX interface) with no delay the following results for the data-rate have been obtained:
    - 17.11 MB/s (real time)
    - 21.36 MB/s (CPU time)
- After many tests on both sides on CRC-32 IEEE 802.3 algorithm the following decisions concerning how the CCB calculates CRC have been made:
    - inverted packet input for CRC computation (bitwise)
    - inverted CRC output (bitwise)
    - CRC transmission as little endian
    - time-stamp transmission as little endian
- Discussion about commands.
    - a list of message types concerning command exchange from CPU to CCB and PDM (via CCB broadcast) has been defined
    - packet number has to be included in payload field for command acknowledgement identification
    - a list of commands for the CCB has been defined. CCB registers shall be configured by means of a specific command.
    - command payloads to be defined
    - no packet retransmission shall be implemented for the following reasons
        - implementation of such a mechanism costs transfer time and is FGPA resource expensive
        - low packet/command transmission error rate
        - fast packet overwriting in CCB cache
- CCB VHDL has been successfully synthesized in a Xilinx Spartan-6 board to be used in Rome as a CCB emulator

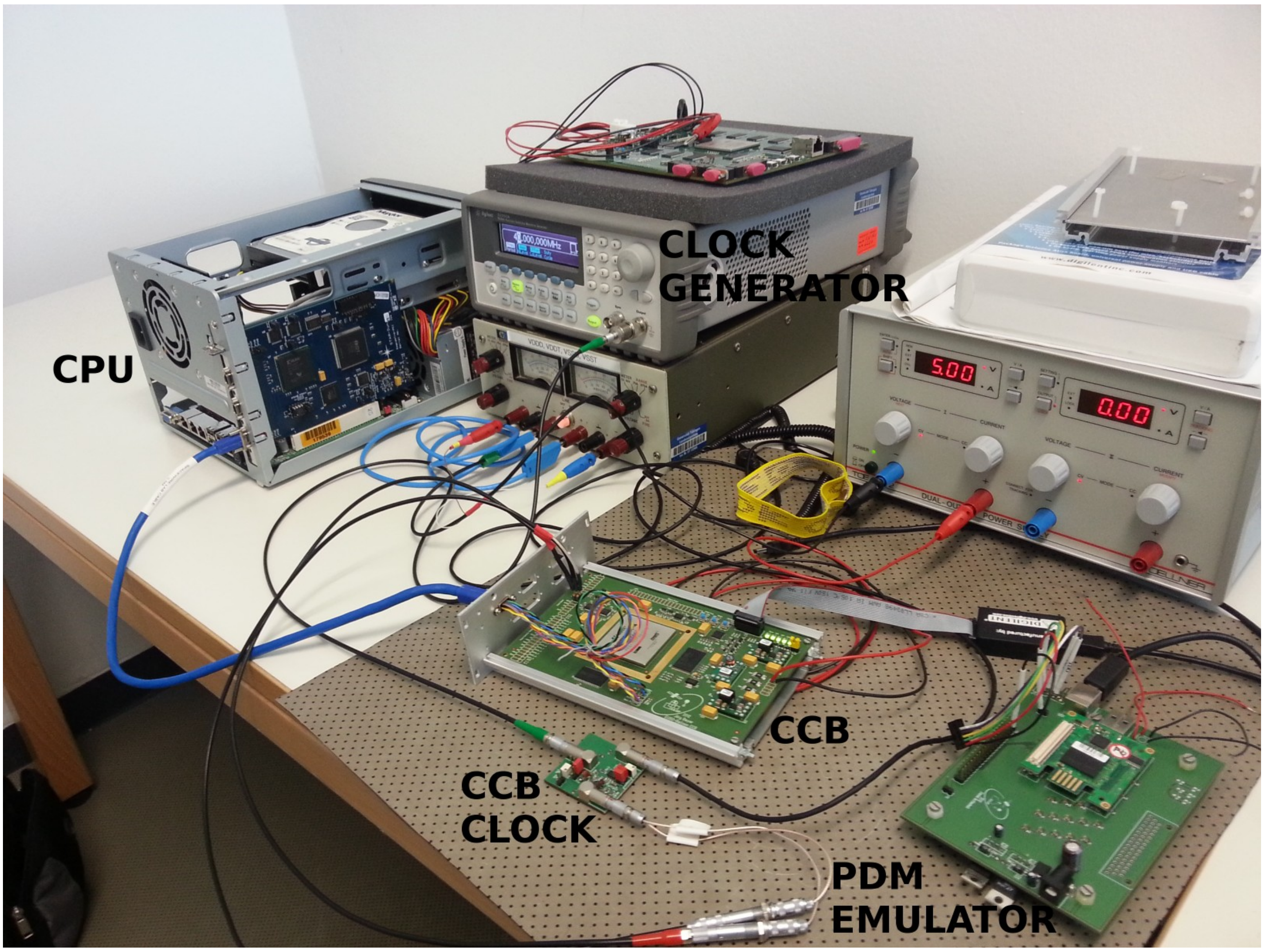

Fig. 1: EUSO-Balloon CPU/CCB pre-integration setup.

Eidesstattliche Erklärung

Ich, Daniel Gottschall, Matrikel-Nr. 28 91 592, versichere hiermit, dass ich meine Diplomarbeit mit dem Thema

Development of a PDM-Simulator Board for JEM-EUSO

nach § 18 Abs. 7 der Diplom-Prüfungsordnung selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe, wobei ich alle wörtlichen und sinngemäßen Zitate als solche gekennzeichnet habe. Die Arbeit wurde bisher keiner anderen Prüfungsbehörde vorgelegt und auch nicht veröffentlicht.

Tübingen, den 25. März 2013

Daniel Gottschall

Danksagung

An dieser Stelle möchte ich mich bei all jenen bedanken, die diese Arbeit möglich gemacht haben und mich in dem Prozess unterstützt haben.

Prof. Dr. Andrea Santangelo: Vielen Dank, dass Du es mir möglich gemacht hast an diesem interessanten Projekt mitzuwirken. Dein Interesse an meinem Fortschritt bei den regelmäßigen JEM-EUSO Treffen hat mich sehr gefreut. Die Unterstützung während der letzten Wochen meiner Arbeit war sehr wichtig für mich.

Dr. Christoph Tenzer: Danke, dass ich mit allen Problemen zu Dir kommen konnte und danke für die Korrekturen meiner schriftlichen Arbeit.

Alejandro, Thomas, Elias, Beppe und Jörg: Liebe Kollegen, vielen Dank für die Unterstützung bei meinem Projekt. Besonderer Dank gebührt Jörg. Ohne Deine Hilfe beim Design der Platine und bei der VHDL Entwicklung wäre ich nicht so weit gekommen.

Werkstatt: Vielen Dank für die unkomplizierte Hilfe mit meinem Gehäuse.

Gabi: Danke, dass Du dir immer Zeit für mich genommen hast.

Büsra: Durch Dich war es nicht immer so langweilig in unserem Büro.

Espresso- und Kaffeerunde: Die Pausen mit Euch waren der perfekte Ausgleich für meine Arbeit am Institut.

Besonders möchte ich mich bei meiner Familie bedanken. Ohne Eure Hilfe in den letzten 7 Jahren wäre mein Studium nicht möglich gewesen. Sowohl finanziell als auch bei jeder Art von Problemen konnte ich immer auf Euch zählen.

Das Beste zum Schluss: Liebe **Ruth**, vielen Dank für alles. Ohne Dich hätte ich es nicht geschaft. Ich liebe Dich.