

Masterthesis

Entwicklung von FPGA basierter Ausleseelektronik für den HADES RICH upgrade und CBM RICH Detektor

- online TDC Kalibrierung und DAC Schwellen Erzeugung -

Developement of FPGA based readout electronics for the
HADES RICH upgrade and CBM RICH detector

- online TDC calibration and DAC threshold settings -

Zur Erlangung des akademischen Grades
Master of Science - Physik



vorgelegt am II. Physikalischen Institut
Fachbereich 07 - Mathematik und Informatik, Physik, Geographie

Autor: **Adrian Amatus Weber**

Matrikelnummer: **4006583**

Abgabe: **September 2017**

1. Gutachter: **Prof. Dr. Claudia Höhne**

2. Gutachter: **Prof. Dr. Wolfgang Kühn**

Selbstständigkeitserklärung

Hiermit versichere ich, die vorgelegte Thesis selbstständig und ohne unerlaubte fremde Hilfe und nur mit den Hilfen angefertigt zu haben, die ich in der Thesis angegeben habe. Alle Textstellen, die wörtlich oder sinngemäß aus veröffentlichten Schriften entnommen sind, und alle Angaben die auf mündlichen Auskünften beruhen, sind als solche kenntlich gemacht. Bei den von mir durchgeführten und in der Thesis erwähnten Untersuchungen habe ich die Grundsätze guter wissenschaftlicher Praxis, wie sie in der "Satzung der Justus-Liebig-Universität zur Sicherung guter wissenschaftlicher Praxis" niedergelegt sind, eingehalten. Gemäß § 25 Abs. 6 der Allgemeinen Bestimmungen für modularisierte Studiengänge dulde ich eine Überprüfung der Thesis mittels Anti-Plagiatssoftware.

Datum

Unterschrift

Zusammenfassung

Im Rahmen der durchgeführten Master Thesis wurde ein online System zur FPGA basierten Kalibration von TDCs für die HADES und CBM RICH readout Elektronik entwickelt. Eine Überprüfung der bisherigen Kalibrierungsverfahren hat verdeutlicht, dass bereits eine lineare Kalibrierung der FineTime eine gute Zeitpräzision im Bereich von Maximal 15 ps ermöglicht. Die Methode der linearen Kalibrierung wurde für FPGAs implementiert und mit Daten aus einem Pulser AddOn überprüft. Die Zeitpräzision von Software basierten und FPGA basierten Verfahren sind im Rahmen der Fehler identisch. Desweiteren zeigt sich, dass die Kalibrierung zu einer hervorragenden Temperaturstabilität führt. Die Zeitpräzision bleibt über einen gemessenen Temperaturbereich von 5 bis 50 Grad Celsius konstant. Die Nutzung von Offsets in den Pulserkanälen zeigt zusätzlich, dass die Zeitpräzision mit steigenden Offsets stetig schlechter wird. Je nach Offset und Kanal konnte eine Zeitpräzision von 15 ps bis 33 ps erreicht werden.

Zusätzlich wurden zwei FPGAs des neuen DiRICH2 Boards zur Schwellenerzeugung für die weitere Auslese der MAPMTs programmiert. Mit Hilfe des TrbNet werden Schwellenwerte an den zentralen FPGA des DiRICH2 gesendet. Über ein SPI Interface werden die Schwellenwerte in den Flash Speicher der zusätzlichen FPGAs geschrieben bzw. gelesen. Die Schwellen werden als Referenzsignal an einen Diskriminator gegeben, wodurch das einfallende Signal der MAPMTs vom Untergrundrauschen separiert werden kann. Durch das neue Layout und die neuen FPGAs konnte das Rauschen der DiRICH Boards deutlich reduziert werden.

Abstract

An online FPGA based calibration system for TDCs of the HADES and CBM RICH readout electronic was developed. A measurement of the current software based calibration systems showed, that a linear calibration of the finetime leads to a time precision of 15 ps at maximum. The linear calibration was implemented on an FPGA and tested with a pulser AddOn. The timing precision of the software based and FPGA based calibration was within errors identical. The measurement of a temperaturescan between 5 and 50 degree celsius showed an outstanding temperature stability of the timing precision. The use of offsets in the pulser channels showed a decrease of the timing precision with increasing offsets. Depending on the offset a precision between 15 ps and 33 ps was reached.

In addition two FPGAs for the new DiRICH2 boards for the thresholds setting of the MAPMT signals were programmed. TrbNet is used to send the threshold values to the main FPGA of the DiRICH2. Via an SPI interface the thresholds are either written or read out from the flash memory of the additional FPGAs. The threshold is used as a reference signal for the discriminator and helps to separate signals of the MAPMTs from background noise. The new layout and the new FPGAs improved the noise substantially.

Inhaltsverzeichnis

1	Theorie	1
1.1	Standardmodell	1
1.2	Das Quark-Gluon-Plasma	3
1.3	Das HADES Experiment	6
1.3.1	START and VETO Detektoren	7
1.3.2	RICH Detektor	7
1.3.3	MDC (Tracking) Detektoren	8
1.3.4	Time-of-Flight (ToF) Detektoren	8
1.3.5	PreShower Detektor	8
1.3.6	Data Acquisition (DAQ)	8
1.4	Das CBM Experiment	9
1.5	Datenauslese der HADES und CBM RICH Detektoren	11
1.5.1	Field Programmable Gate Arrays (FPGA)	12
1.5.2	Time to Digital Converter (TDC)	16
2	FPGA basierte TDC Online Kalibrierung	19
2.1	Funktion der Kalibrierung	25
2.2	Slow Control und Flash Werte	30
2.3	Auswertung ohne Offset	31
2.4	Auswertung mit Offset	36
2.5	Zusammenfassung	41
3	DAC FPGA für den DiRICH2	42
3.1	SPI-Slave	43
3.2	DAC Implementierung durch eine Δ - Σ -Modulation.	43
3.3	Flash Speicher	44
3.4	Slow Control	45
3.4.1	Lesen	45
3.4.2	Schreiben	46
3.5	Ergebnisse von Testmessungen im realen HADES Detektoraufbau . .	47
4	Appendix	50
4.1	Magnetfeldboard	50
4.2	Umweltsensorboard	52
4.3	Laboraufbau zur TDC Kalibrierung	54

1 Theorie

1.1 Standardmodell

Das Standardmodell der Elementarteilchenphysik ist ein sehr erfolgreiches Modell, welches eine quantitative Beschreibung aller Interaktionen fundamentaler Teilchen liefert. Beim Standardmodell handelt es sich um eine renormalisierbare relativistische Quantenfeldtheorie, die auf der nicht-Abelschen Symmetrie der Eichgruppe $SU(3)_c \times SU(2)_L \times U(1)_Y$ beruht. Sie lässt sich dadurch in die vereinheitlichte Theorie der elektroschwachen Wechselwirkung, als auch in die Theorie der starken Wechselwirkung, die Quantenchromodynamik (QCD), aufteilen. Die Gravitation als vierte grundlegende Kraft ist bislang, zusammen mit dem Graviton g als Austauscheteilchen, nicht in das Standardmodell integriert.

Die elektroschwache Theorie beschreibt dabei die elektromagnetische und schwache Interaktion zwischen Quarks und Leptonen mit einem chiralen nicht-abelschen Isospin und einer abelschen Hyperladungs-Eichsymmetrie - $SU(2)_L \times U(1)_Y$.

Bei der Quantenchromodynamik handelt es sich hingegen um eine Vektor-Eichsymmetrie. Sie beschreibt die $SU(3)_c$ Farbwechselwirkung zwischen Quarks und Gluonen, den Austauscheteilchen der starken Wechselwirkung.

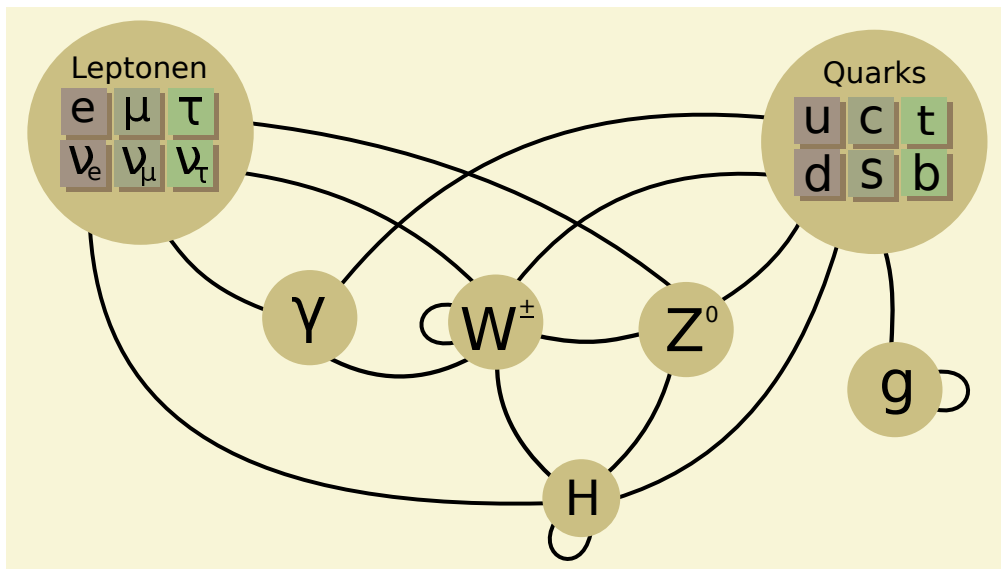


Abbildung 1: Schematische Darstellungen des Standardmodells und der Wechselwirkungen.

Das Standardmodell lässt sich in seinem schematischen Aufbau in 3 Bereiche teilen - Fermionen, Eichbosonen und das Higgs-Boson.

Fermionen (Spin $1/2$) umfassen im Standardmodell die Quarks, sowie die Leptonen und teilen sich in je 3 Familien auf. Pro Familie gibt es ein Quark mit $+2/3$ und ein Quark mit $-1/3$ Ladung. Daraus ergeben sich die 3 Familien, die aus up und down, charm und strange sowie top und bottom Quarks bestehen. Alle Quarks nehmen sowohl an der elektroschwachen, als auch an der starken Wechselwirkung teil. Das

heißt wiederum, dass diese auch Farbladungen tragen.

Jede der 3 Generationen der Leptonen besteht aus einem Elektron, Myon bzw. Tau, sowie seinem jeweiligen Neutrino-partner, dem ν_e , ν_μ bzw. ν_τ . Elektronen, Myonen und Taus sind allesamt geladen und nehmen an der elektroschwachen Wechselwirkung teil. Sie tragen allerdings keine Farbladung, wodurch sie nicht an der starken Wechselwirkung beteiligt sind. Neutrinos nehmen eine spezielle Stellung ein. Sie sind elektrisch neutral und tragen keine Farbladung. Somit nehmen sie nur an der schwachen Wechselwirkung teil.

Alle Fermionen im Standardmodell kommen zusätzlich als Anti-Teilchen vor.

Wechselwirkung	koppelt an	Austausch-Teilchen	Masse (GeV/c ²)	J^P
stark	Farbe	8 Gluonen (g)	0	1^-
elektromagn.	elektrische Ladung	Photon (γ)	0	1^-
schwach	schwache Ladung	W^\pm	$80,376 \pm 0,033$	1
		Z^0	$91,1876 \pm 0,0021$	

Tabelle 1: Übersichtstabelle der 3 Wechselwirkungen des Standardmodells. [1, 2]

Die Wechselwirkungen zwischen den einzelnen Teilchen werden im Standardmodell über Eichbosonen beschrieben. Die Eichbosonen der elektromagnetischen Wechselwirkung sind die masselosen Photonen γ . Sie sind elektrisch neutral und koppeln somit nicht an sich selbst.

In der schwachen Wechselwirkung gibt es zwei geladene Eichbosonen, das W^+ und das W^- . Hinzu kommt noch ein neutrales Z^0 Eichboson. Diese drei Eichbosonen tragen Masse (siehe Tabelle 1), was im Endeffekt zur schwachen Kopplung und der Namensgebung führt.

In der vereinheitlichten elektroschwachen Theorie wird die Wechselwirkung über ein neu postuliertes W-Boson Triplet (W^+ , W^- und W^0), sowie einem B^0 (Singlett) erzeugt. Durch die Mischung der Eigenzustände über den Weinbergwinkel führt dies zu dem bekannten Z^0 Boson und den Photonen.

Die Gluonen als Eichbosonen der starken Wechselwirkung sind masselos und tragen Farbladung. Ein Gluon trägt immer eine Farbe und eine Anti-Farbe, was zu 8 unterschiedlichen Gluonen führt. Dadurch, dass Gluonen selbst Farbladung tragen, können diese an sich selbst koppeln und z.B. Glueballs entstehen.

Durch die starke Wechselwirkung können sich Quarks, die Farbladung tragen, zu Hadronen zusammenschließen. Aufgrund der geforderten Farbneutralität solcher Gebilde sind nur bestimmte Kombinationen aus Elementarteilchen erlaubt. Mesonen bestehen aus einem Quark und einem Anti-Quark. Baryonen hingegen aus 3 Quarks oder 3 Anti-Quarks. Allerdings sind theoretisch auch weitere Strukturen wie Glueballs, Tetraquarks, etc. möglich.

Das Higgs-Boson nimmt im Standardmodell eine wichtige Rolle ein. Es ist kein Eichboson, sondern eine elementare Anregung des Higgs-Feldes. Das Higgs-Boson

folgt aus einer spontanen Symmetriebrechung.

Durch den von Null verschiedenen Vakuumerwartungswert des Higgs-Feld bekommen die eigentlich masselosen Eichbosonen der elektroschwachen Wechselwirkung (W^\pm , Z^0) ihre hohe Masse. Das Higgs-Feld sorgt zusätzlich dafür, dass Quarks und Leptonen Masse erhalten.

1.2 Das Quark-Gluon-Plasma

Genau wie in Wasser können auch in anderer Materie, speziell auch in stark wechselwirkender Materie, Phasenübergänge, unterschiedliche Phasen, sowie kritische Endpunkte auftreten. Durch Variation von thermodynamischen Parametern, wie der Temperatur T , dem Druck P oder dem chemischen Potential μ kann man zwischen den einzelnen Phasen wechseln [3].

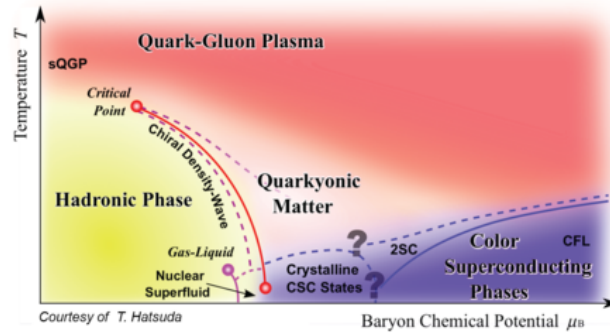


Abbildung 2: Phasendiagramm stark wechselwirkender Materie [4].

In stark wechselwirkender Materie ist, genau wie in elektromagnetisch wechselwirkender Materie, die Erzeugung eines Plasmas möglich. Anstatt wie im elektromagnetischen Plasma elektrische Ladungen zu trennen, werden hier freie Farbladungsträger (Quark, Antiquark, Gluon) erzeugt. Das Potential der starken Wechselwirkung ermöglicht erst auf sehr kleinen Abständen eine asymptotische Freiheit. Entfernt man zwei Quarks voneinander, so muss mit steigendem Abstand immer mehr Energie aufgebracht werden um den Abstand weiter zu vergrößern. Ab einem gewissen Punkt wird es energetisch günstiger ein neues Quark-Antiquark Paar zu erzeugen. Dieses Prinzip wird als *confinement* beschrieben. Um das *confinement* aufzulösen, muss die nukleare Materie entweder erhitzt oder komprimiert werden. Sind die erreichten Drücke und/oder Temperaturen groß genug, kommt es zum *deconfinement* und es lassen sich quasi-freie Quarks und Gluonen beobachten. Dieser neu erzeugte Zustand wird als Quark-Gluon-Plasma bezeichnet.

Trägt man die Parameter Temperatur und baryochemisches Potential wie in Abb. 2 gegeneinander auf, so ergibt sich, in Analogie zu Wasser, ein Phasendiagramm. Im Bereich kleiner baryochemischer Potentiale μ ($\mu \approx 0$) wird ein kontinuierlicher Übergang zwischen hadronischer Phase und QGP erwartet. Aus Simulationen mit nicht-pertubativer lattice-QCD lässt sich als kritische Temperatur T_c für diesen Übergang ein Wert von $T_c=150$ bis 170 MeV errechnen [5, 6, 7, 8].

Bisher durchgeführte Experimente in diesem Bereich des Phasendiagramms weisen auf eine kritische Temperatur von $T_c \approx 160$ MeV hin [5].

Geht man im Phasendiagramm zu höheren Dichten wird je nach Modell ein kritischer Endpunkt, wie auch ein Phasenübergang 1. Ordnung erwartet [9]. Die genaue Position des Endpunktes sowie die genaue Lage des Übergangs ist bisher nicht bekannt. Um diesen Bereich näher zu untersuchen, wird u.a. in Zukunft das CBM Experiment an FAIR Daten sammeln.

Andere Bereiche wie der bei hohen Temperaturen und niedrigem baryochemischen Potential werden mit Hilfe großer Beschleunigeranlagen wie dem LHC am CERN untersucht.

Um ein Quark-Gluon Plasma im Labor zu erzeugen, werden an Beschleunigeranlagen schwere und große Ionen beschleunigt und in *Collider* oder *fixed-target* Experimenten zur Kollision gebracht. Damit genügend Volumen und Materie zur Bestimmung der thermodynamischen Parameter vorhanden ist, werden in diesen Experimenten Schwerionenstrahlen genutzt.

Die Kollision von Schwerionen ist in Abb. 3 schematisch dargestellt. Zwei lorentz-kontrahierte Schwerionen fliegen mit einem Überlapp aufeinander zu. Der Überlapp der beiden Schwerionen, der über den *Impact Parameter* b bestimmt wird, klassifiziert die Zentralität der Kollision. Liegt der Impact Parameter bei $b=0$, so handelt es sich um eine zentrale Kollision. Der überlappende Bereich der Kollision bildet einen elliptischen Raum aus, der eine hohe Temperatur und/oder Dichte besitzt. Die Nukleonen in diesem Bereich werden *Participants* genannt. Die Nukleonen, die ohne Kollision weiterfliegen, sind *Spectators*.

Durch die Kollision der Schwerionen kommt es innerhalb kurzer Zeiträume zu hohen Energiedichten und zur Produktion energiereicher Teilchen in den Nukleon-Nukleon Kollisionen. Nach kurzer Zeit wird lokal ein thermisches Gleichgewicht erreicht und ein sogenannter Feuerball entsteht. Unter gewissen Umständen bildet sich in diesem Feuerball ein Quark-Gluon-Plasma aus. Die Lebensdauer eines Quark-Gluon-Plasmas, bevor es zur Hadronisierung kommt, liegt bei $\approx 10^{-23}$ s bis $\approx 10^{-22}$ s [5, 10].

Damit sich ein Quark-Gluon Plasma ausbilden kann, muss die erzeugte Materie korreliert und zumindest annähernd in einem thermischen Gleichgewicht sein. Durch die Kollision entsteht ein Druckgradient in Richtung des Vakuums. Je nach Überlapp der Kollision expandiert der entstandene Feuerball in longitudinaler stärker Richtung als in transversaler Richtung.

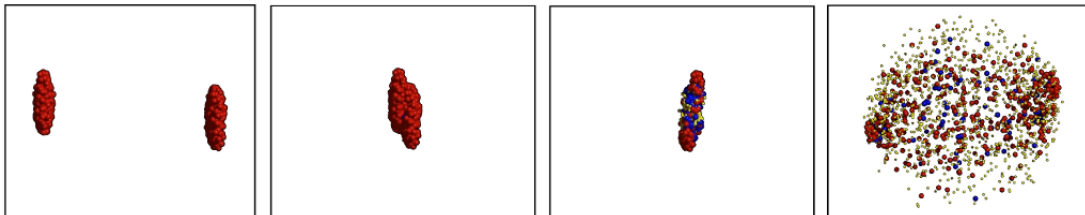


Abbildung 3: Bildsequenz einer zentralen Au+Au Kollision bei 23 AGeV Strahlenergie in UrQMD Simulationen [11].

Der dabei entstehende sogenannte elliptische Fluß ist messbar, indem man die Anzahl an Teilchen pro Winkel registriert. Der elliptische Fluß zeichnet sich dann durch eine annähernd sinusförmige Verteilung der Teilchen aus.

Über eine Fourier-Transformation lässt sich die Amplitude der Winkelverteilung v_2 ermitteln. Wird v_2 gegen den Transversalimpuls p_t

$$p_t = p \cdot \sin(\Theta)$$

aufgetragen, so folgt Abb. 4. Die Messdaten dieser Zuordnung stimmen sehr exakt mit theoretischen Beschreibungen der Hydrodynamik überein.

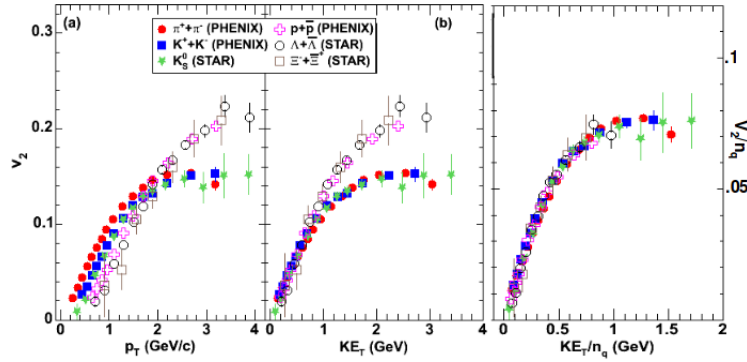


Abbildung 4: Messdaten der RHIC Experimente STAR und PHENIX zum elliptischen Fluß: v_2 gegen p_t (links), sowie gegen KE_t (Mitte) aufgetragen. Zu sehen sind jeweils Messungen verschiedener Hadronen. Die Messung fand bei zentralen Au+Au Kollision mit $\sqrt{s} = 200\text{GeV}$ statt. Rechts: v_2 aufgetragen gegen KE_t , dividiert durch die Anzahl an Konstituentenquarks n_q . [12]

Die Daten legen eine Beschreibung des Quark-Gluon-Plasmas als ideale Flüssigkeit (Viskosität=0) nahe. Für eine hydrodynamische Beschreibung ist ein thermisches Gleichgewicht Voraussetzung, was wiederum ein Zeichen dafür ist, dass sich auch das QGP im thermodynamischen Gleichgewicht befindet.

Abb. 4 zeigt weiter, dass es eine Massenordnung bzw. eine Ordnung nach Mesonen und Baryonen gibt. Trägt man v_2 gegen KE_t , normiert auf die Anzahl an Konstituentenquarks (v_2/n_q gegen KE_t/n_q), auf, so ergibt sich das Bild wie in Abb. 4 rechts.

$$KE_t = m_t - m_0$$

Alle Messwerte fallen nun auf eine Kurve, was die Interpretation nahelegt, dass die Quarks die relevanten Freiheitsgrade bei der Ausbildung des Flusses bilden. Genau dieses zeichnet das QGP aus und diese Messung ist damit ein sehr illustrativer Nachweis desselben. Mit anderen Observablen werden dann andere Eigenschaften des Quark-Gluon-Plasmas, wie zum Beispiel Temperatur, Dichte und die Propagation von schweren Quarks, wie den c-Quarks, gemessen.

Bei hohen Kollisionsenergie ($\mu_B \approx 0$, hohes T) wird ein Quark-Gluon-Plasma erzeugt. Neueren Experimenten am RHIC und LHC gelingt es das QGP zunehmend genauer zu charakterisieren.

In Schwerionenkollisionen tritt dieses wahrscheinlich erst als Phasenübergang bei $\approx 20\text{--}30$ AGeV Strahlenergie auf [13].

Die Struktur und Charakterisierung der Materie bei großem μ_B und moderater Temperatur T ist weitgehend unklar. Die Untersuchung dieses Bereichs wird durch die HADES und CBM Experimente an GSI/FAIR durchgeführt.

1.3 Das HADES Experiment

Das *High Acceptance DiElectron Spectrometer* (HADES) Experiment ist eines der größten fixed-target Experimente der GSI Helmholtzzentrum für Schwerionenforschung in Darmstadt und an den SIS18 Teilchenbeschleuniger der GSI angeschlossen. HADES ermöglicht es Kollisionen von p+p, p+A, π +A, aber auch A+A bei verschiedenen Energien zu untersuchen. Die Untersuchung von Au+Au Kollisionen bei 1,23 AGeV Strahlenergie wurde erst durch ein Upgrade der Datenaquisition (DAQ) und des TOF Detektors ermöglicht. Die DAQ des HADES Experiments wurde hierbei durch die Verwendung von *Field Programmable Gate Arrays* (FPGAs) und der Verwendung von Glasfaser-Technologie für die hohen Datenraten bei Au+Au Kollisionen vorbereitet.

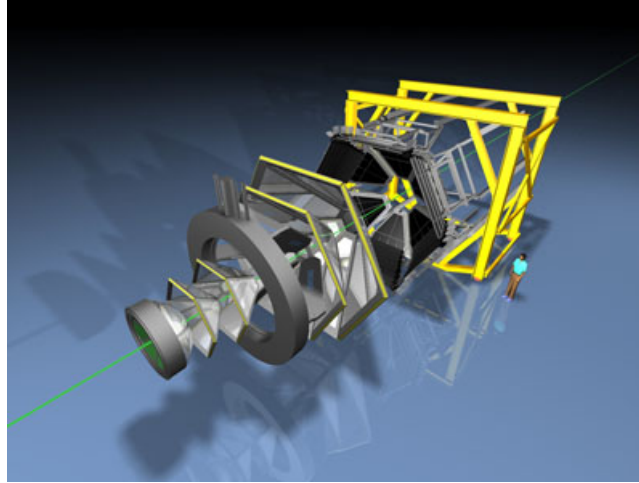


Abbildung 5: Das Hadesspektrometer. [14]

Das Spektrometer ermöglicht durch sein Design die präzise Untersuchung von in-medium Modifikationen leichter Vektor-Mesonen, d.h. ω , ρ oder ϕ -Mesonen durch die Messung von e^+e^- -Paaren. Dileptonen bieten sich bei Schwerionenkollisionen als vielversprechende Proben an, da diese nicht stark wechselwirken und so Informationen direkt aus der Kollisionszone unbeeinflusst meßbar machen. Der dileptonische

Zerfallskanal von Vektor-Mesonen ermöglicht so die Messung ihrer Eigenschaften in dichter Kernmaterie. Da allerdings die Zerfallsbreite in diesem Kanal sehr klein ist, sind Dielektronen eine sehr seltene Observable.

Um eine hohe Statistik und hohe Akzeptanz zu gewährleisten, wurde HADES mit sehr schnellen Sub-Detektoren für hohe Event-Raten konstruiert. Desweiteren wird eine hohe Akzeptanz von HADES durch 18° - 85° Öffnungswinkel in polarer Richtung und voller azimuthaler Abdeckung erzielt. Durch ein geringes Materialaufkommen im Detektor wird zusätzlich der Untergrund möglichst klein gehalten. Die Detektoren sind auf eine hohe Impulsauflösung ausgelegt, um z.B. eine gute Auflösung der invarianten Masse von leichten Vektormesonen zu erhalten. Diese hohe Impulsauflösung wird insbesondere durch die Verwendung von Multi-Wire Drift Chambers (MDCs) vor und nach dem Magnetfeld erreicht.

1.3.1 START and VETO Detektoren

Der START und der VETO Detektor des HADES Experiments werden zur Erzeugung eines Trigger-Signals, aber auch als Startsignal für die Time-of-Flight Messung genutzt. Beide aus Diamant gefertigten Detektoren sitzen vor, bzw. hinter den Targets. Jedes Teilchen im einkommenden Strahl erzeugt somit ein Signal im Startdetektor. Der VETO Detektor hingegen registriert alle Teilchen, die nicht mit dem Target interagiert haben. Aus dem Zusammenspiel beider Detektoren lässt sich so bestimmen, ob eine Kollision stattgefunden hat und ob ein Trigger-Signal ausgesendet wird.

1.3.2 RICH Detektor

Um Elektronen und Positronen zu detektieren, wurde ein RICH Detektor in HADES integriert. Der RICH Detektor ist um das Target herum positioniert, sodass die produzierten Teilchen den Radiator des RICH Detektors durchfliegen. Bewegen sich die Teilchen mit einer Geschwindigkeit durch den Radiator, der die Mediumlichtgeschwindigkeit überschreitet, so werden Photonen über den Cherenkov-Effekt emittiert. Die über einen Spiegel auf eine Photodetektor Ebene reflektierten und fokussierten Photonen werden als Ringe auf dieser Ebene abgebildet. Aus deren Radius und dem Teilchenimpuls lassen sich Teilchen identifizieren. Durch eine geschickte Wahl des Radiators können so insbesondere Elektronen identifiziert werden. Der bisher verwendete RICH Detektor wurde mit C_4F_{10} als Radiatorgas betrieben und konnte so für die Detektion von Elektronen und Positronen in einem Impulsbereich zwischen 0.1 und 1.5 GeV/c optimiert werden. [15]

Mit dem nächsten Update von HADES wird die Photodetektorebene des RICH Detektors erneuert. Der bisherige Gasdetektor wird durch H12700 Hamamatsu MAPMTs ersetzt. Die dafür in Kooperation mit CBM und PANDA entwickelte Auslese basiert auf TDCs, die in FPGAs implementiert werden. Diese neu entwickelten DiRICH Boards werden Signale mit sehr guter Zeitauflösung bei moderater Amplitudeninformation liefern. Insgesamt wird mit dem RICH upgrade die Effizienz der Dielektronen Identifizierung in HADES entscheidend verbessert [16].

1.3.3 MDC (Tracking) Detektoren

Der HADES Detektor nutzt zum Tracking der Teilchen eine *Mini Drift Kammer* in Kombination mit einem Magneten. Hierbei sind 2 der MDCs vor und 2 hinter dem Magneten angebracht. Die aus Feld- und Sensor-Drähten bestehende Driftkammer registriert Teilchen, die durch das eingebrachte Gas fliegen und dieses ionisieren. Die dadurch erzeugten Ladungen werden über die Sensor-Drähte registriert. Durch die Überlagerung mehrerer Ebenen aus gegeneinander rotierten Drähten wird so, mit Hilfe von TDCs zur Auslese, die Spur eines Teilchens rekonstruiert. Die im Rahmen dieser Thesis entwickelte Kalibrierung kann auch für die TDC Auslese des MDC Tracking verwendet werden.

1.3.4 Time-of-Flight (ToF) Detektoren

Der HADES Detektor besitzt zwei unterschiedliche Flugzeitwände. Zum einen befinden sich unter kleinem Azimutwinkel *Resistiv-Plate-Chambers* (RPCs), deren Granularität die Messung bei hohen Spurdichten wie in Au+Au Kollisionen erlaubt. Unter größerem Azimutwinkel ist eine Wand aus *Szintillatorstäben* und Photomultipliern positioniert. Durch ein Signal des START Detektors und den Stoppsignalen der beiden Time-Of-Flight-Wände kann die Flugzeit der Teilchen ermittelt werden und so eine effektive Identifikation unterschiedlichster Teilchen mit hinreichendem Massenunterschied ermöglicht werden.

1.3.5 PreShower Detektor

Zusammen mit den beiden Flugzeitwänden bildet der PreShower Detektor den sogenannten *Multiplicity and Electron Trigger Array* (META) Detektor. Der PreShower Detektor dient zur Unterscheidung zwischen Elektronen und Hadronen. Er besteht aus 3 Drahtkammern, die durch Bleiwände getrennt sind. Innerhalb der Bleiwände erzeugen die Teilchen Bremsstrahlung, die zu elektromagnetischen Schauern führen. Da die Bremsstrahlung nur bei Elektronen signifikant ist, lassen sich über die gemessene Ladung Elektronen von Hadronen unterscheiden. Unter kleinem Azimutwinkel, unter dem der PreShower Detektor steht, ist diese Unterscheidung durch die anderen Detektoren nicht möglich.

1.3.6 Data Acquisition (DAQ)

Die Daten Akquise (DAQ) des HADES Experiments beruht auf an der GSI selbst entwickelter Elektronik. Das Trigger und Readout Board (TRB) wird in verschiedensten Anwendungen genutzt. Es kann zum einen als Hub, aber auch als TDC verwendet werden. Durch die Nutzung von FPGA Technologie ist der Anwendungsbereich sehr flexibel. Die Daten aus den einzelnen Detektoren werden über das speziell entwickelte TrbNet Protokoll übermittelt.

Die Daten der einzelnen Sub-Detektoren werden über Hubs verbunden und über Gigabit-Ethernet an den sogenannten Event-Builder geleitet. Dieser bereitet die Daten auf und speichert sie ggf. als Hades List Data (HLD) Dateien ab.

1.4 Das CBM Experiment

Das *Compressed Baryonic Matter* (CBM) Experiment ist Bestandteil der im Bau befindlichen FAIR Beschleunigeranlage bei der GSI Helmholtzzentrum für Schwerionenforschung in Darmstadt. Das CBM Experiment wird an SIS100 p+A und A+A Kollisionen mit Strahlenergien von 2 bis 11 AGeV für Au-Ionen messen. Dadurch wird es möglich, bis zu 10 mal höhere Dichten als in normaler Kernmaterie zu erzeugen.

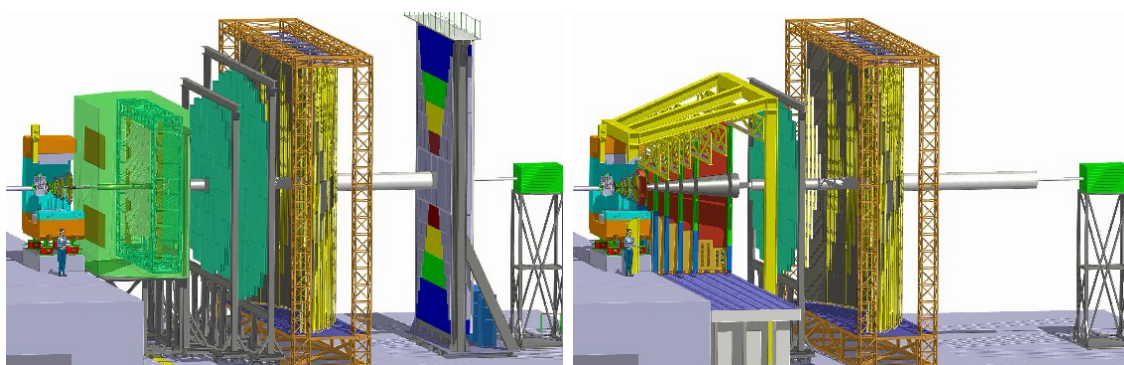
Durch Kollisionen bei solch hoher Dichte, aber moderaten Temperaturen, können bisher weitgehend unbekannte Bereiche des QGP-Phasendiagramms vermessen und auf mögliche Phasenübergänge hin untersucht werden. Unabhängig von möglichen Phasenübergängen wird CBM generell stark komprimierte baryonische Materie charakterisieren.

Zur Untersuchung der Materie (ver)misst man unterschiedliche Observablen. Eine wichtige Observable sind durchdringende, elektromagnetische Sonden, die in der hochdichten Phase erzeugt wurden. Diese durchdringen den Feuerball im weiteren Verlauf weitgehend ohne Wechselwirkung und sind so eine ideale Quelle um Informationen aus der hochdichten Phase zu erhalten.

Eine Messung des ρ -Mesons im dielektrischen Zerfallskanal kann so z.B. Aufschluss über die Eigenschaften des ρ -Mesons im Medium geben, da das ρ -Meson eine Lebensdauer hat, die kleiner als die des Feuerballs ist.

Ein anderer Zugang ist die Messung der charm-Produktion und die Verteilung der charm Quarks auf die unterschiedlichen Hadronen in nuklearer Materie. Um insbesondere die charm-Produktion nachweisen zu können, muss das CBM Experiment einige Herausforderungen bewältigen. Neben einer sehr guten Teilchenrekonstruktion und -identifikation ist insbesondere eine hohe Auslesegeschwindigkeit erforderlich. Um die hohe Kollisionsrate von 10^7 Au+Au Kollisionen pro Sekunde zu erreichen, muss eine sehr schnelle und noch dazu strahlenharte Ausleseelektronik entwickelt werden. Elektronik und Detektoren sind für Raten bis zu 10MHz ausgelegt.

Der Aufbau des CBM Experiments ist in Abb. 6 zu sehen. Der erste Subdetektor des Experiments ist der Micro Vertex Detector (MVD), der zur hoch präzisen Messung von Vertices sekundärer Zerfälle konzipiert ist.



(a) Version mit RICH-Detektor

(b) Version mit Myonon-Kammer

Abbildung 6: Schematischer Aufbau des CBM Detektors [3].

Auf den MVD folgt das Silicon Tracking System. Es besteht aus 8 Layern und wird zur Spur- und Impulsmessung aller aus der Produktion aus dem Target stammenden geladenen Teilchen genutzt. Zusammen mit dem Magneten, der den STS umgibt, kann aus der Krümmung der Teilchenspur der Impuls rekonstruiert werden. Der folgende Detektor ist ein RICH Detektor. Er ist für die Identifikation von Elektronen mit Impulsen <10 GeV/c ausgelegt. Als Photomultiplier werden Hamamatsu 12700 MAPMTs zum Einsatz kommen. Diese sind bereits bestellt und werden schon jetzt für den HADES RICH Detektor upgrade genutzt.

Der RICH Detektor kann gegen eine Myon Chamber (MUCH) ausgetauscht werden, damit es insbesondere möglich wird Myonen aus Schwerionenkollisionen mit niedrigem Impuls zu identifizieren. Der MUCH ist so kompakt wie möglich gebaut und besteht aus 6 Eisen Absorbern, sowie 18 Gas-Tracking Kammern.

Um die Elektronenidentifikation zu verbessern, wird zusätzlich ein Transition Radiation Detektor entwickelt. Um Hadronen zu identifizieren, wird eine Flugzeitmessung durchgeführt. Wie auch bei HADES kommen hierzu RPCs zum Einsatz. Diese sind in 10 Meter Entfernung zum Target als Array angeordnet. Zum Abschluss von CBM wird ein elektromagnetisches Kalorimeter im klassischen Schaschlik Design genutzt. Es wird im Experiment direkte Photonen und neutrale Mesonen, die in Photonen zerfallen, messen.

1.5 Datenauslese der HADES und CBM RICH Detektoren

Die Datenauslese des HADES und des CBM RICH Detektors basiert auf DiRICH Boards. Signale aus Hamamatsu H12700 MAPMTs werden durch 32 Kanäle je DiRICH verarbeitet. Die gemessene Präzision (RMS) der MAPMTs liegt bei ca. 650 ps [16]. Die Signale werden verstärkt und mit einem einstellbaren Schwellenwert diskriminiert, um so das Signal vom Rauschen zu separieren. Aus den Signalen, die diese Diskrimination überstanden haben, werden in Field Programmable Gate Array (FPGA) basierten Time-to-Digital Convertern (TDC) Zeitstempel erzeugt. Jedes Signal erzeugt eine Zeitinformation mit seiner steigenden und eines mit seiner fallenden Flanke. Aus diesen Zeitinformationen lässt sich eine Time-over-Threshold (ToT) errechnen. Dieses Konzept ermöglicht eine Hit Auslese mit guter Zeitauflösung und moderater Amplitude.

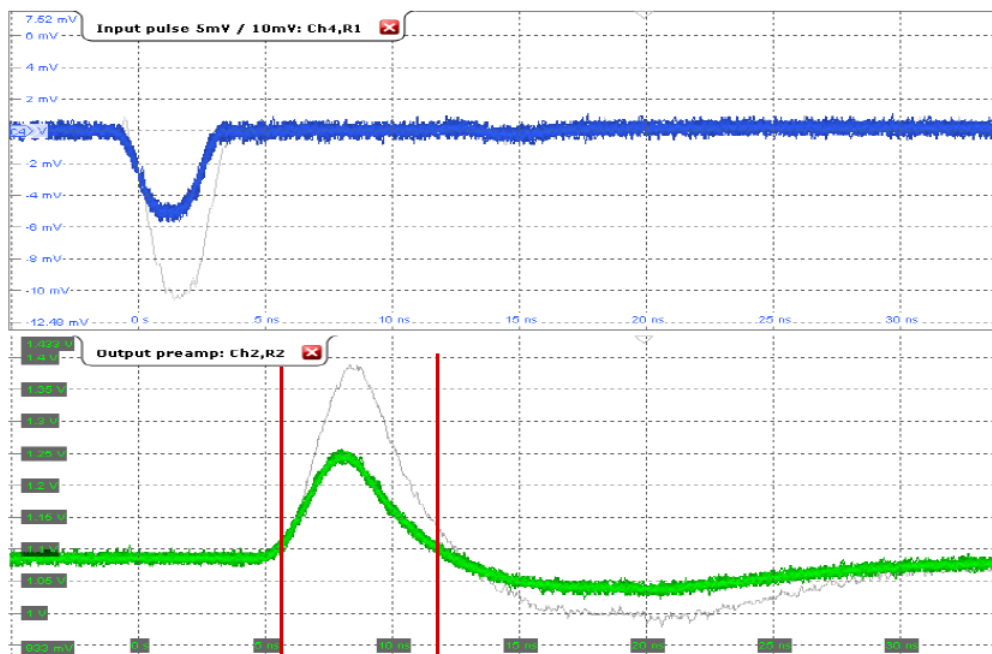


Abbildung 7: Vergleich des unverstärkten und vorverstärkten MAPMT Signals [17].

Abbildung 7 zeigt im oberen Bereich ein Signal, welches direkt aus einem MAPMT kommt. Dieses ist in seiner Amplitude sehr moderat und wird auf dem DiRICH Vorverstärkt, sodass ein Signal entsteht, wie es in Abb. 7 unten zu sehen ist. Mit Hilfe zweier roten Linien wurde beispielhaft ein Time-over-Threshold Bereich des Signals eingezeichnet. In diesem Bereich ist das Signal oberhalb des definierten Schwellenwertes.

Jedes DiRICH Board ist mit einem zentralen Lattice ECP5 FPGA ausgestattet. Zusätzlich werden zwei Lattice MachXO3 FGAs zur Schwellenerzeugung verbaut. Je zwei der DiRICH Boards lesen einen MAPMT aus. Sie sind auf einer speziell entwickelten Backplane montiert. Eine Seite dieser Backplane trägt die MAPMTs, die andere Seite die nötige Elektronik (siehe Abb. 8). Zu den 12 DiRICH Boards

auf einer Backplane kommt zusätzlich noch ein Power Modul für die Versorgung der Elektronik mit Niederspannung und der MAPMTs mit Hochspannung. Ein Concentrator Board sammelt u.a. alle Zeitinformationen und sendet diese über das TrbNet zur weiteren Datenverarbeitung.

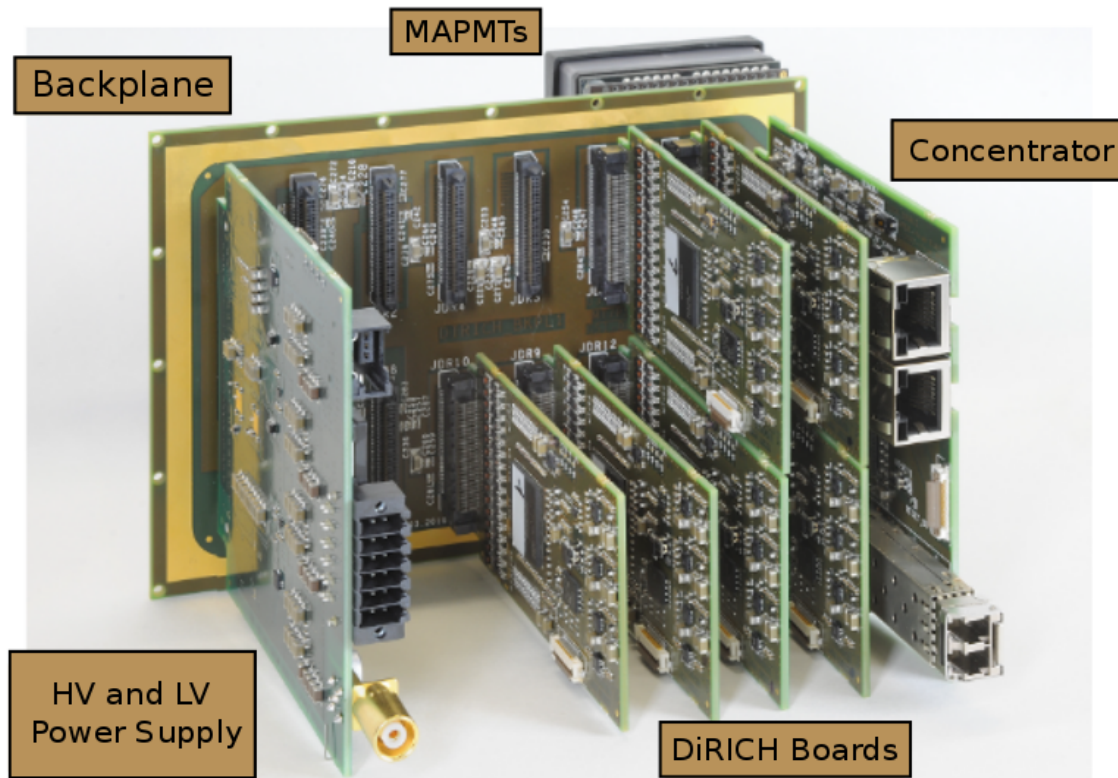


Abbildung 8: Aufbau der Elektronik/DAQ für das HADES RICH upgrade [18].

1.5.1 Field Programmable Gate Arrays (FPGA)

Field Programmable Gate Arrays (FPGA) werden im HADES Experiment an diversen Stellen in der Ausleseelektronik eingesetzt. Sie dienen zur Erstellung von Schwellenwerten, werden als Time-To-Digital Converter (TDC) eingesetzt oder dienen in anderen unterschiedlichsten Anwendungsbereichen als variabler Elektronikbaustein. FPGAs sind Silizium basierte Bausteine, die wiederum mit unterschiedlichen Elektronik Strukturen versehen werden. Jeder FPGA besitzt als zentrale Struktur sogenannte Common Logic Blocks (CLBs). Diese CLBs sind in großer Anzahl zu finden. Die Unterstruktur dieser CLBs ist in Slices eingeteilt. Somit besteht der Großteil eines FPGAs aus einer Matrix an Slices. All diese Slices werden durch eine immense Anzahl an Drähten untereinander verbunden. Ein slice besteht typischerweise aus einer *Look Up Table* (LUT) mit 4 (oder auch mehr) Eingängen. Diese 4 Eingänge werden je nach entwickelter Logik auf ein Ausgangssignal gelegt. Zusätzlich enthält ein Slice in der Regel einen Flip-Flop (FF) als Speicherelement (Abb. 9).

Ein FPGA besitzt desweiteren Input-/Output-Blocks (IOB), Block RAM, dedizierte arithmetische Einheiten zur Durchführung größerer Berechnungen oder auch digitale Clock Manager (DCM) (siehe Abb. 10). Mit Hilfe der IOBs ist es möglich externe Signale in den FPGA oder auch interne Signale an die externe Elektronik zu leiten. Ein FPGA ist für sich ohne Programmierung nicht zu gebrauchen. Erst durch die Erzeugung einer Logik, die in den FPGA gespielt wird, wird der FPGA zu einem mächtigen Werkzeug in der Elektronik. FPGAs bieten die Möglichkeit nahezu jede Logik zu implementieren. Durch geschicktes Programmieren kann so ein und derselbe Elektronik Baustein diverse Aufgaben erfüllen. Genau hier liegt auch der Vorteil eines FPGAs. Durch die hohe Flexibilität sind diese sehr gut für klein-Serien und Entwicklungsarbeit geeignet. Dabei halten sie die Produktionspreise im Vergleich zu ASICS relativ gering. Erst bei großen Stückzahlen würde sich eine Entwicklung eines ASICS lohnen.

FPGAs laufen typischerweise mit einer Clock Geschwindigkeit von lediglich 50-200 MHz. Diese Frequenzen sind vergleichsweise gering, allerdings relativiert sich dies durch die Möglichkeit der parallelen Nutzung.

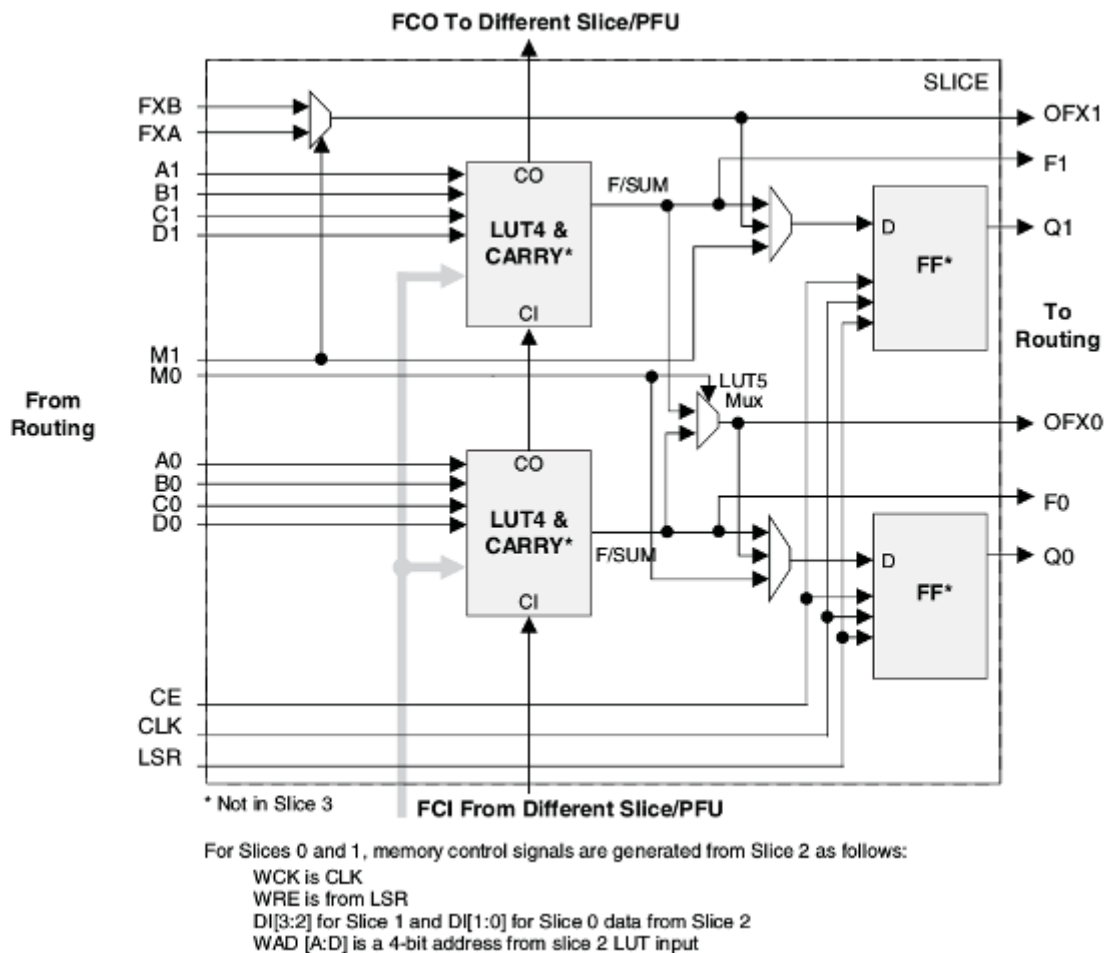


Abbildung 9: Diagramm eines Slice auf einem Lattice ECP3 FPGAs. Dieser ist bei TrbSc und Trb3 Boards verbaut. Für die DiRICH Boards kommt der Nachfolger ECP5 zum Einsatz. [19]

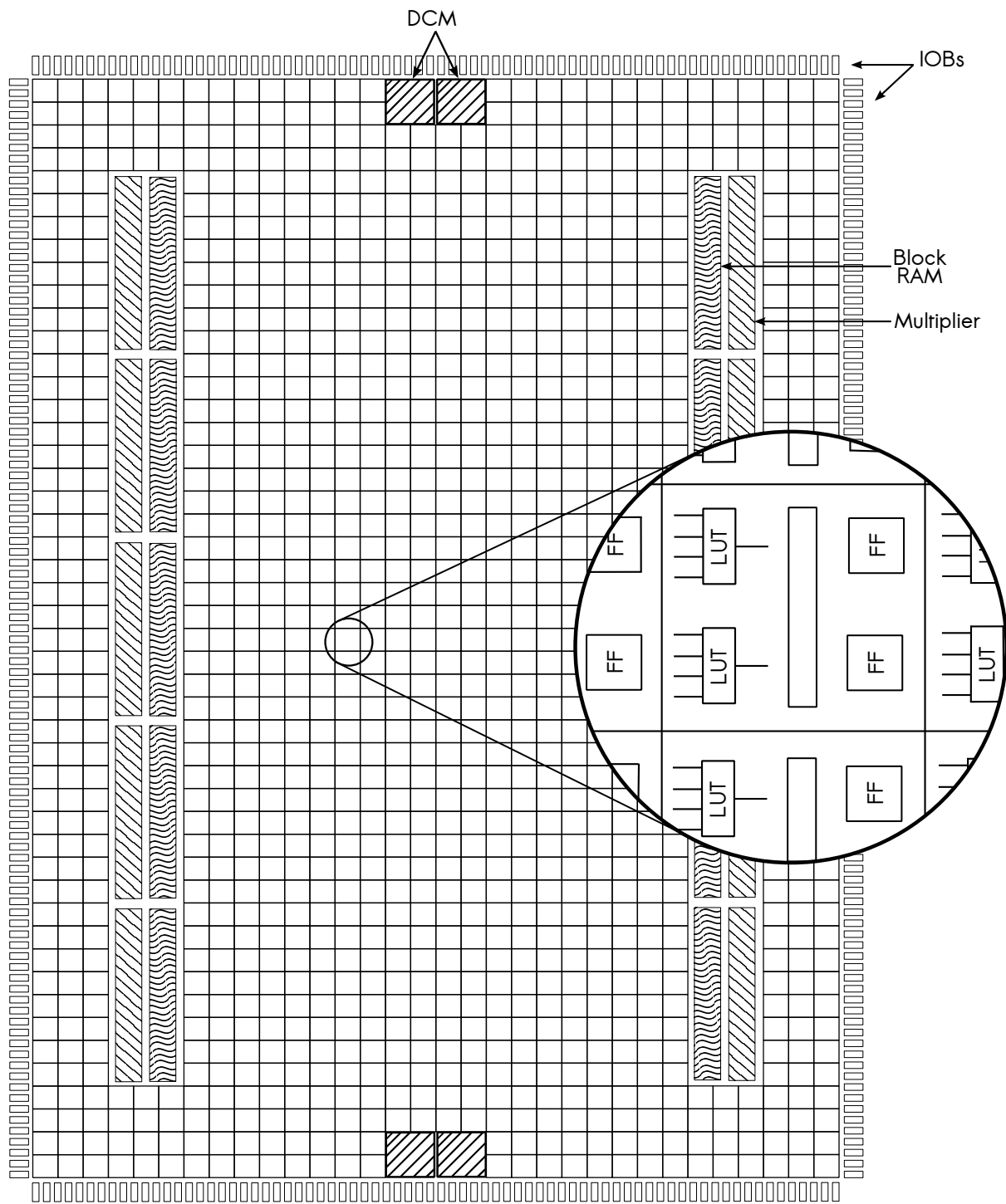


Abbildung 10: Schematische Darstellung eines allgemeinen FPGAs. Eine exakte Darstellung des Lattice ECP3 ist in [19] zu finden.

Die Implementierung von Logik für einen FPGA wird typischerweise mit Hilfe der *Very high speed integrated circuit **H**ardware **D**escription **L**anguage* (VHDL) durchgeführt. Wie der Name schon sagt, ist VHDL keine echte Programmiersprache. Mit VHDL wird mehr eine Beschreibung der Logik geliefert, die wiederum durch eine Vielzahl spezieller Werkzeuge in die passende Konfiguration des FPGAs übersetzt wird. VHDL bietet die Möglichkeit beliebige Logiken zu entwickeln und auch den Funktionsumfang durch die Nutzung von Librarys zu erweitern - ähnlich zu `#include` in C.

Die so entwickelte Logic lässt sich als Bitstream über verschiedene Kommunikationswege, wie z.B. JTAG, auf den FPGA übermitteln. Die einzelnen sub-Elemente des FPGA wie z.B. LUTs oder die interne "Verkabelung" werden nach diesen Informationen konfiguriert.

1.5.2 Time to Digital Converter (TDC)

Ein *Time-To-Digital Converter* (TDC) ist ein elektronisches Bauteil, welches eine Zeitinformation in ein digitales Signal wandelt. Diese Funktion lässt sich auf verschiedenste Möglichkeiten mit unterschiedlichsten Präzisionen implementieren. Für HADES und CBM wird der TDC mit Hilfe eines FPGAs implementiert. Läuft ein FPGA mit einer Clock von 200 MHz, so kann, basierend auf einer synchronen Logik, alle 5 ns ein Signal verarbeitet werden. Diese Zeiteinheit wird als grobe Schrittgröße eines TDC genommen und im Folgenden als *CoarseTime* bezeichnet. Für einen genauen TDC ist eine Schrittweite von 5 ns allerdings viel zu groß. Um eine Schrittweite von 10 ps (im Fall von Trb3 TDCs) zu erreichen, wird im FPGA die interne physikalische Verzögerung der einzelnen Logik Bausteine genutzt. Jeder Logikbaustein braucht eine bestimmte Zeit, um eine Ausgabe zu produzieren. Werden nun viele solche Bausteine aneinander gereiht, so kann man die Zeit zwischen den 5 ns entfernten CoarseTimes in kleine Zeitintervalle, die sogenannte *FineTime*, unterteilen. Ein Bin der *FineTime* wird so einem Verzögerungselement zugeordnet.

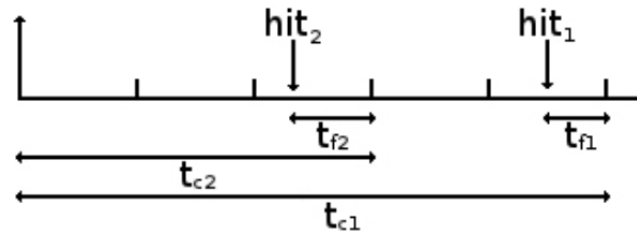


Abbildung 11: Schematische Darstellungen der Coarse- (t_c) und der Fine-Time (t_f) verschiedener Hits [20].

Um eine Verzögerung im FPGA zu generieren werden bei HADES Volladdierer genutzt. Jeder Volladdierer hat 2 Eingänge A und B, sowie ein Übertragsbit C. Das Ergebnis der Berechnung wird auf der Ausgabe S ausgegeben. Der neue Übertrag wird auf das Übertragsbit des nächsten Volladdierers gelegt. Vergleicht man diesen Aufbau mit Abb. 9, so sieht man schon, dass die LUT Einheit eines Slices bei ECP3 FPGAs unter entsprechender Programmierung genau diese Funktion bietet. Verbindet man einzelne Volladdierer nun wie in Abb. 12 mit den dazugehörigen Werten auf A und B, so ist die Ausgabe der Kette der Volladdierer ohne einfallendes Signal eine Reihe von Einsen. Wird ein Hit registriert und in den ersten Volladdierer geführt, so ändert sich im ersten Zeitschritt nur dessen Ausgabe und Übertrag. Bei N Volladdierern ist nun die Ausgabe eine Null, gefolgt von (N-1) Einsen. Mit wenigen Pikosekunden Verzögerung ändert sich nun auch die Ausgabe und der Übertrag vom zweiten Volladdierer. Mit leichter Verzögerung werden so nach und nach alle Einsen zu Nullen gewandelt.

Damit nun auch wirklich die Zeitinformationen der FineTime extrahiert wird, wird die Ausgabe aller N Volladdierer genau dann in einem FlipFlop gespeichert, wenn der Oszillator des FPGAs eine neue steigende Flanke hat, also sich die CoarseTime ändert. Der so entstandene Thermometercode aus den FlipFlops wird durch einen Encoder in eine binäre Zahl gewandelt und durch einen FIFO gepuffert. Um die

Da es sich bei der Verbindung der Slices auch um eine physikalische Verbindung handelt, ändern sich die Länge der Verbindungen und andere Eigenschaften mit der Temperatur. Auch die Positionierung der Slices durch die Werkzeuge der FPGA Programmierung führt zu unterschiedlichen Abständen zwischen Slices.

Diese Variationen führen zusammen zu verschiedenen Verzögerungen der einzelnen Elemente eines TDC. Diese Verzögerung sorgt dafür, dass sich bei der Datennahme mehr Hits in bestimmten FineBins ansammeln, als in anderen Bins.

Dieses Verhalten führt zu einer immensen Verschlechterung der Zeitpräzision des TDC mit steigender Temperatur. Um diesem Problem entgegen zu wirken muss jeder Kanal eines TDCs auf jedem FPGA kalibriert werden.

Um die Präzision des TDC zu bestimmen, werden Differenzen zwischen verschiedenen Kanälen gebildet. Die daraus resultierende Verteilung ist beispielhaft in Abb. 13 dargestellt. Der RMS Wert dieser Verteilung gibt die Präzision an. Die Differenz wird über die Formel

$$\Delta t = (t_{coarse1} - t_{coarse2}) - (t_{fine1} - t_{fine2})$$

berechnet.

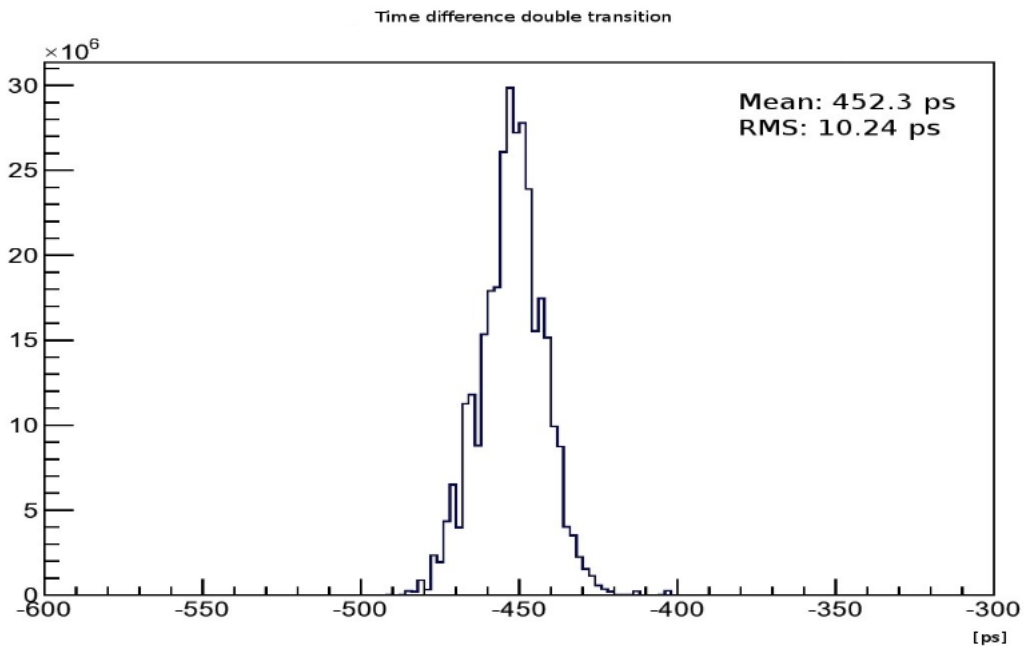


Abbildung 13: Verteilung der Zeitdifferenz einer Zeitmessung des selben Signals in 2 Kanälen [20].

2 FPGA basierte TDC Online Kalibrierung

FPGA basierte TDC weisen in ihrer Präzision eine erhebliche Temperaturabhängigkeit auf. Um dieser entgegen zu wirken, muss jeder Kanal eines TDC eines jeden FPGAs kalibriert werden. Bisher wurde hierzu eine Software basierte Lösung genutzt, die nach der Datennahme ausgeführt wurde [21]. Es besteht die Möglichkeit diese Software-Kalibrierung auf zwei verschiedene Arten auszuführen. Die erste Herangehensweise ist eine *lineare Kalibrierung*. Hierzu wird jeder Kanal analysiert und der Wert des kleinsten nicht leeren Bins (das Minimum), sowie der Wert des größten nicht leeren Bins (das Maximum) gesucht. Aus diesen Werten lässt sich nun eine Gerade erzeugen und mit dessen Steigung eine Kalibrierung durchführen.

$$FineBin_{Cal}(n) = 5ns \cdot \frac{n - Min}{Max - Min} \quad (2.1)$$

Eine solche lineare Kalibrierung geht davon aus, dass jeder Bin in der Verteilung der FineTime gleich breit ist. Da dies nicht der Realität entspricht, ist die lineare Kalibrierung nicht die ideale Methode. Allerdings ist sie durch ihre geringe Komplexität nicht sehr speicherintensiv. Um eine möglichst hohe Präzision zu erreichen, wurde eine weitere komplexere Kalibrierung in der Software eingeführt. Diese Kalibrierung bezieht jedes einzelne Bin der Verteilung der FineTime in die Kalibrierung mit ein und gleicht dadurch die unterschiedlichen Breiten der Bins aus. Diese Kalibrierung folgt der Funktion

$$FineBin_{Cal}(n) = \frac{(\sum_{i=0}^n Statistic(i)) - \frac{Statistic(n)}{2}}{(\sum_{i=0}^{NoOfBins} Statistic(i)) \cdot 5ns} \quad (2.2)$$

Die zweite Kalibrierungsmethode, im Folgenden Bin-by-Bin genannt, ist für eine FPGA basierte Kalibrierung nicht praktikabel, da hierzu über einen großen Wertebereich hinweg alle Bin-Einträge (*statistic*) zwischengespeichert und über arithmetische Berechnungen miteinander verarbeitet werden müssen. Hierzu reicht die zur Verfügung stehende Hardware eines TDC-FPGAs nicht aus. Die Implementierung einer linearen Kalibrierung benötigt hingegen lediglich zwei Werte, die für jeden Kanal gespeichert werden müssen. Dies vereinfacht das Problem dramatisch und eine FPGA basierte Lösung wäre realistisch.

Beide Software Kalibrierungsmethoden wurden im Labor getestet und verglichen. Insbesondere ist die Temperaturabhängigkeit der Präzision mit beiden Methoden getestet worden. Durch die Gegenüberstellung lässt sich quantitativ beurteilen, in wie fern die lineare Kalibrierung schlechter als die Bin-by-Bin Kalibrierung ist und ob eine FPGA basierte Lösung mit hinnehmbar guter Präzision möglich ist. Dem Ergebnis entsprechend wird eine FPGA basierte Implementierung geplant und umgesetzt.

Mittels einer Heiz-/Kühlanlage wurde im Labor ein TrbSc¹ Board, mit einem Lattice ECP3 FPGA ausgestattet ist, bei unterschiedlichen Temperaturen mit Pulsen versorgt und Daten genommen. Aus der Verteilung der Zeitdifferenz zweier Kanäle lässt sich ein RMS Wert zur Bestimmung der Präzision errechnet (vgl. Abb. 13). Die Versorgung mit Pulsen lief über ein AddOn² und ein zweites TrbSc Board zur Vermeidung von Korrelation mit der internen Clock des TrbSc. Der Trigger wurde auf 10 kHz eingestellt. Ein Zeitfenster für die Triggersignale wurde deaktiviert (TriggerWindow). Die Pulse wurden mit einer Breite von 41 ns und einer Periode von 100.000 ns auf die Kanäle des TDC vom TrbSc geführt. Zur Datennahme wurden alle 11 verfügbaren TDC Kanäle aktiviert. Die Temperatur des FPGAs wurde zwischen 5 °C und 50 °C in 5 °C Schritten variiert.

Um die genommenen Daten auszuwerten, wurden einzelne Kanäle mit einander verglichen. Jeder genannte Kanal besitzt somit einen sogenannten Referenz-Kanal, der Tabelle 2 zu entnehmen ist. Zur Auswertung wurde eine Bin-by-Bin und eine lineare Kalibrierung durch die Software erzeugt. Für die lineare Kalibrierung wurden mittels eines C Programms die minimalen und maximalen Werte eines jeden Kanals ausgelesen und anschließend an die korrekte Stelle in die Datei first.C des Go4 Frameworks geschrieben.

Kanal	ref. Kanal	Kanal	ref. Kanal
Ch 2	Ch 1	Ch 7	Ch 2
Ch 3	Ch 2	Ch 8	Ch 1
Ch 4	Ch 1	Ch 9	Ch 7
Ch 5	Ch 4	Ch 10	Ch 5
Ch 6	Ch 3	Ch 11	Ch 9

Tabelle 2: Übersicht über die Vergleichskanäle eines jeden Kanals. Diese Zuweisung ist für alle Auswertungen in dieser Arbeit genutzt worden.

In Abbildung 14 - 18 sind die Ergebnisse der Messung zu sehen. Abbildung 14 zeigt eine Gegenüberstellung von 3 möglichen Kalibrierungen. Es wurden eine lineare und eine Bin-by-Bin Kalibrierung durchgeführt. Als dritte Möglichkeit wurde eine Bin-by-Bin Kalibrierung bei 20 °C erstellt und zur Kalibrierung aller weiteren Temperaturen genutzt. Die gezeigten Werte sind Mittelwerte aus allen 11 Kanälen. Der angegebene Fehler ist die Standardabweichung der einzelnen Kanäle. Abbildung 15 und 16 zeigen die gleiche Darstellung für die Kanäle 5 und 8.

Anhand der Abbildungen ist festzustellen, dass die Bin-by-Bin Kalibrierung mit etwa 10 ps Präzision die beste Methode darstellt. Wird aber keine neue Kalibrierung bei schwankender Temperatur erstellt, so steigt der RMS Wert massiv an. Eine Kalibrierung darf also nicht über zu große Temperaturbereiche genutzt werden. Die lineare Kalibrierung weist in der Messung, genau wie die Bin-by-Bin Kalibrierung, eine hohe Konstanz über die Temperaturbereiche hinweg auf.

¹TrbSc: Kleine Version eines TRB3 Boards (1/4 des TRB3). Es handelt sich um eine Eigenentwicklung der GSI zur Datennahme.

²AddOn: Aufsteckbares Modul zur Verbindung mit einem externen Pulser.

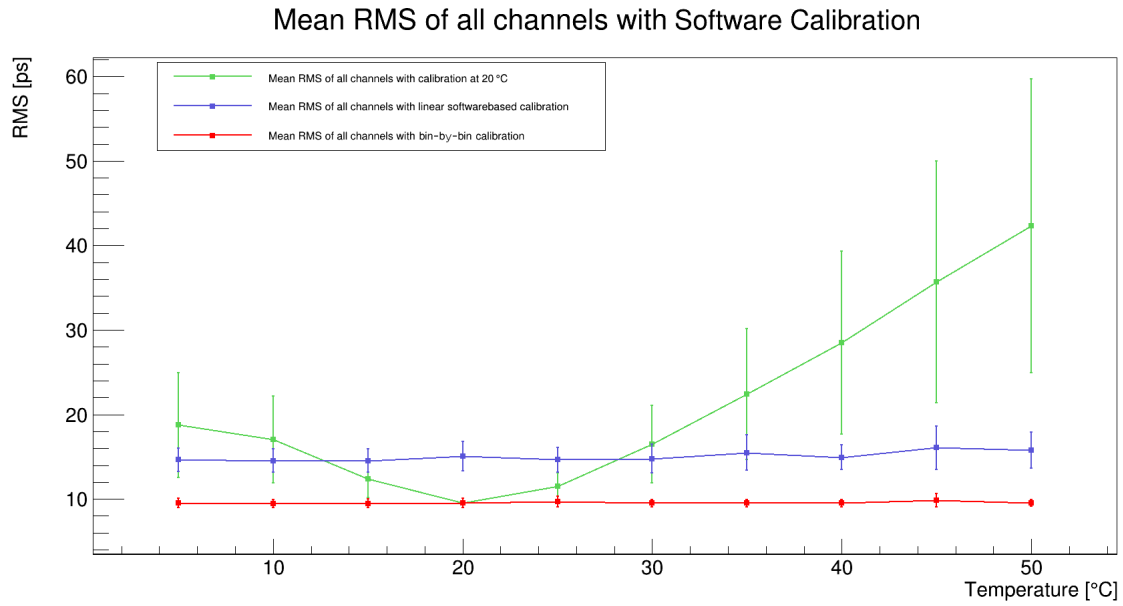


Abbildung 14: Der Mittelwert des RMS Werts, ausgerechnet aus allen zur Verfügung stehenden 11 Kanälen für die Kalibrierung basierend auf den Kalibrierungsdaten einer Bin-by-Bin Kalibrierung bei 20 °C (grün), einer linearen Kalibrierung (blau) und der Bin-by-Bin Kalibrierung (rot). Die Werte wurden für alle gemessenen Temperaturen zwischen 5 °C und 50 °C berechnet. Der angegebene Fehler ist die Standardabweichung der Kanäle zum Mittelwert.

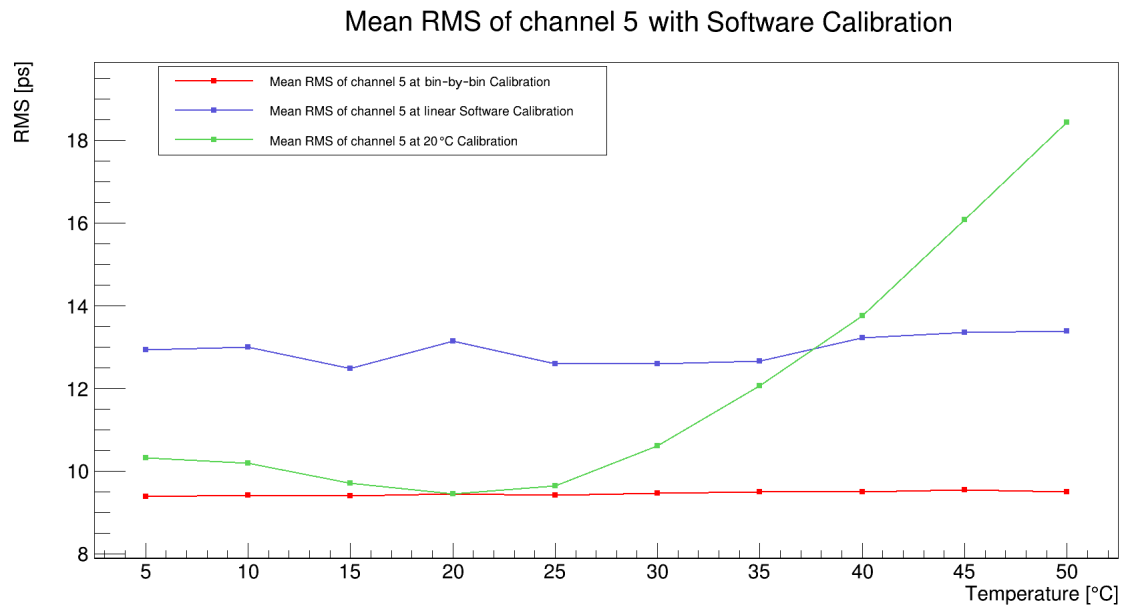


Abbildung 15: Der Mittelwert des RMS Werts von Kanal 5 für die Kalibrierung basierend auf den Kalibrierungsdaten einer Bin-by-Bin Kalibrierung bei 20 °C (grün), einer linearen Kalibrierung (blau) und der Bin-by-Bin Kalibrierung (rot). Die Werte wurden für alle gemessenen Temperaturen zwischen 5 °C und 50 °C berechnet.

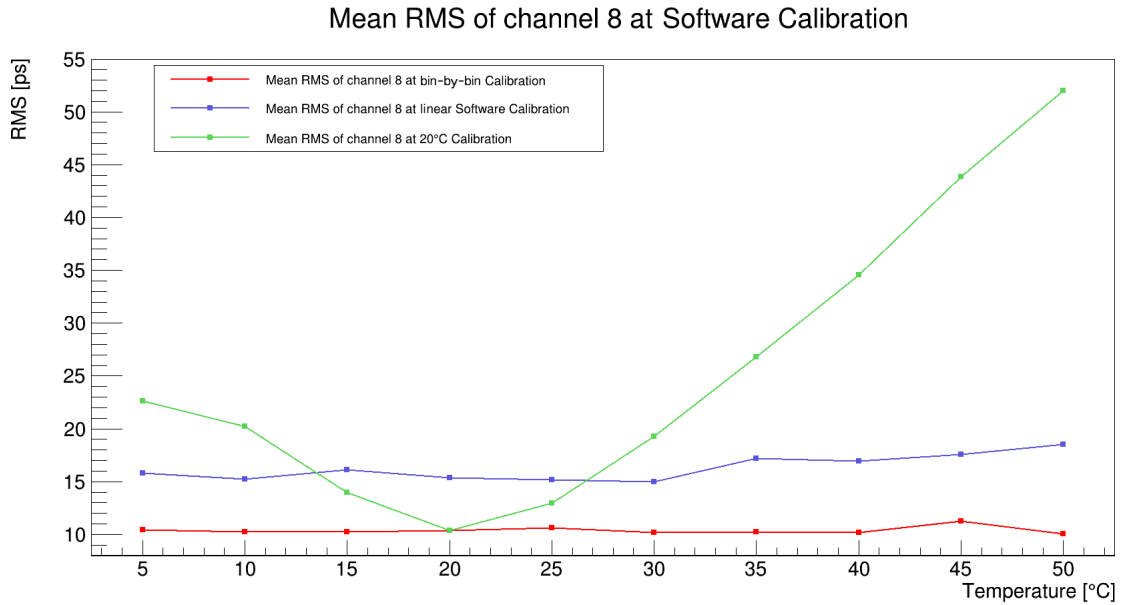


Abbildung 16: Der Mittelwert des RMS Werts von Kanal 8 für die Kalibrierung basierend auf den Kalibrierungsdaten einer Bin-by-Bin Kalibrierung bei 20 °C (grün), einer linearen Kalibrierung (blau) und der Bin-by-Bin Kalibrierung (rot). Die Werte wurden für alle gemessenen Temperaturen zwischen 5 °C und 50 °C berechnet.

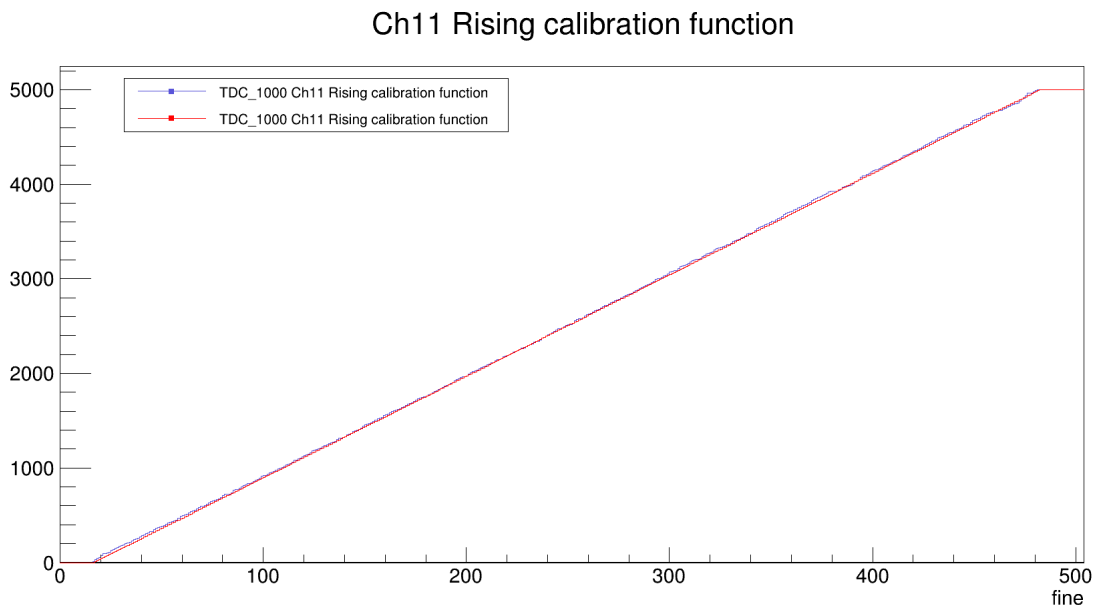


Abbildung 17: Blau: Die von der Software berechnete Kalibrierungskurve der Bin-by-Bin Kalibrierung. Rot: Die von der Software berechnete Kalibrierungskurve der linearen Kalibrierung. Die Kalibrierung wurde an Kanal 11 bei 25 °C durchgeführt.

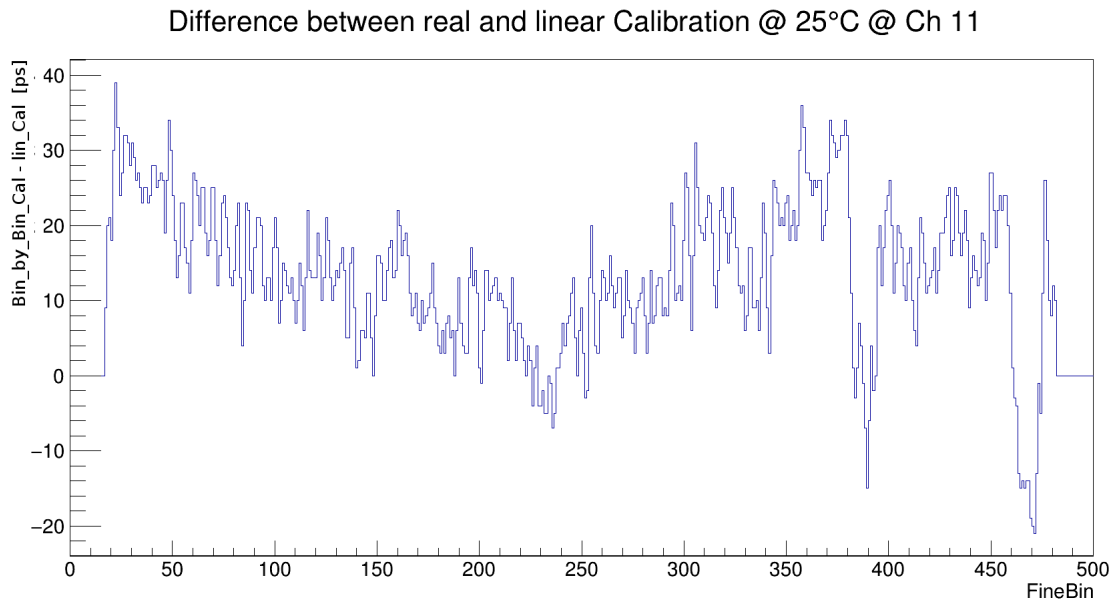


Abbildung 18: Die Differenz zwischen Bin-by-Bin und linearer Kalibrierung. Die Kalibrierung wurde an Kanal 11 bei 25 °C durchgeführt.

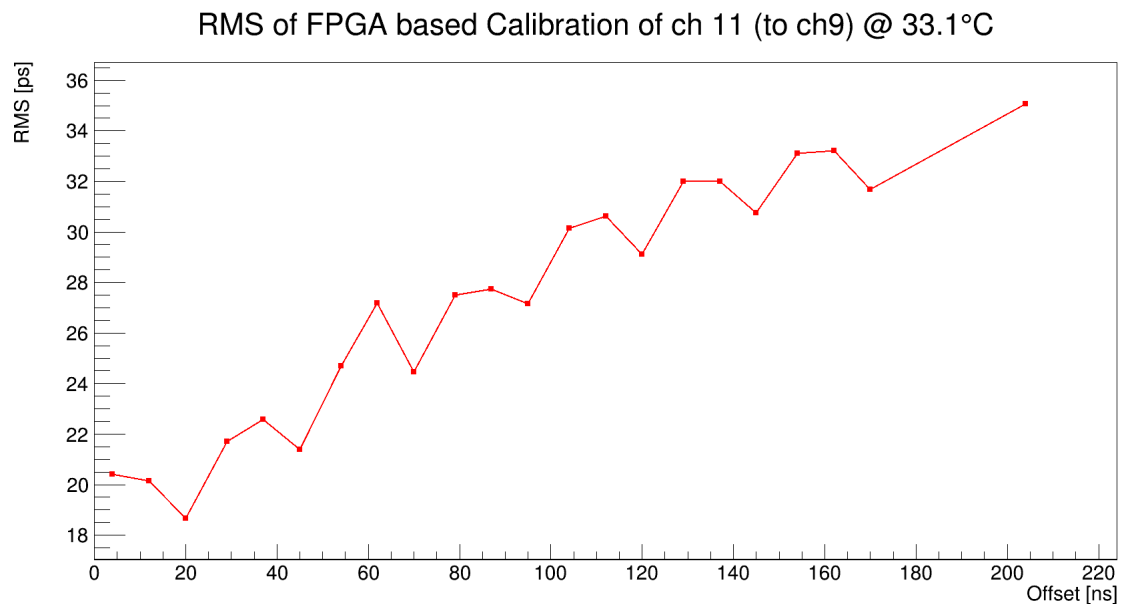


Abbildung 19: Gemessener RMS Wert für Kanal 11 bei verschiedenen Offsets des Pulzers. Die Offsets lassen sich in 4 ns Schritten einstellen. Gewählt wurden 8 ns Schritte. Die Messung fand unter konstanter Temperatur bei 33,1 °C statt.

Der RMS Wert liegt hier bei 16 ps und weicht somit nur um 6 ps von der besten Methode ab. Da 16 ps für die Anwendung bei HADES und CBM eine ausreichend gute Präzision ist, ist die lineare Kalibrierung ausreichend (intrinsische Zeitpräzision der MAPMTS: ≈ 650 ps [16]).

Kanal	Offset [ns]	Kanal	Offset [ns]
Ch 0	0	Ch 6	37
Ch 1	0	Ch 7	58
Ch 2	8	Ch 8	54
Ch 3	12	Ch 9	0
Ch 4	4	Ch 10	0
Ch 5	66	Ch 11	158

Tabelle 3: Eingestellter Offset des Pulsers für die jeweiligen Kanäle des TDC.

Somit ist die Kalibrierung auch auf den FPGA auslagerbar und kann damit bereits während der Datennahme durchgeführt werden. Die Messungen für Kanal 5 und 8 zeigen allerdings auch, dass die Werte bei allen Kalibrierungsmethoden vom Kanal abhängen. Die Präzision kann von Kanal zu Kanal im Bereich von einigen Pikosekunden variieren. Abb. 17 und Abb. 18 zeigen die Differenz der beiden Kalibrierungsfunktionen für die Bin-by-Bin und die lineare Kalibrierung. Die Bin-by-Bin Kalibrierung weicht im Verlauf nur minimal von der linearen Kalibrierung ab. Diese minimale Abweichung führt allerdings bereits zur Verbesserung der Präzision um die besagten 6 ps.

Da die Auswertung auf Daten beruht, die zur gleichen Zeit vom Pulser kamen, wurde für die Auswertung der FPGA basierten Kalibrierung zusätzlich eine weitere Analyse mit Daten durchgeführt, die mit einem Offset im Pulser genommen wurden. Der Offset im jeweilige Kanal ist Tabelle 3 zu entnehmen. Die sonst für alle Kanäle gleichzeitig kommenden Pulse können über einen Offset in 4 ns Schritten verzögert werden. Dadurch kommen die Pulser Signale aus dem zweiten TrbSc nicht mehr alle gleichzeitig, sondern zu ganz unterschiedlichen Zeiten. Die Nutzung des Offsets führt zu realistischeren Bedingungen. Abb. 19 zeigt hierzu anhand von Kanal 11 wie sich der RMS Wert in Abhängigkeit des Offsets verändert. Das Verhalten weist einen stetigen Anstieg des RMS Wertes auf. Allerdings ist deutlich zu sehen, wie der RMS Wert im Bereich von Messungen mit einem Offset, der ein vielfaches der CoarseTime ist, wieder besser wird.

2.1 Funktion der Kalibrierung

Um eine lineare Kalibrierung implementieren zu können, müssen die TDC-Daten extrahiert werden. Hierzu wird in der Entität des VHDL Codes des TDC die Entität zur linearen Kalibrierung (*Calibration*) zwischen die Datenausgabe geschaltet. So werden Daten erst in die Kalibrierung und dann wieder in den normalen Datenfluss, in den die Daten normalerweise nach der TDC Entität gehen, geschleust. Desweiteren ist die Kalibrierung an die TrbNet Slow Control angeschlossen. Dies bietet einige Vorteile. Es wird so möglich, den Inhalt einzelner Register zu ändern und auszulesen. Außerdem kann hierüber der Flash-Speicher des TrbSc (Lattice ECP3 FPGA) ausgelesen werden.

Die aufgenommenen Daten aus dem TDC teilen sich in verschiedene Typen auf. Prinzipiell werden immer 32-Bit lange Daten aus dem TDC geschrieben. Je nach Bit Konfiguration haben diese aber völlig unterschiedliche Funktionen. Um echte Daten aus dem TDC zu bekommen muss Bit 31 der Daten einer 1 entsprechen (siehe Tabelle 4).

Bits	Erklärung
31	Marker für die Zeit Daten
30 - 29	reserviert
28 - 22	Kanal Nummer
21 - 12	FineTime Wert
11	Typ der Kante; '1': rising edge '0': falling edge
10 - 0	CoarseTime

Tabelle 4: Aufbau des Zeitdaten-Pakets des TDC.

Ist dies nicht der Fall, so wurden andere Daten wie z.B. Fehlernummern ausgesendet. Damit solche Daten nicht zur Kalibrierung herangezogen werden, wird lediglich dann eine Kalibrierung durchgeführt, wenn Bit 31 = 1 ist.

In diesem Fall wird eine Flag *DoCal* auf 1 gesetzt. Der exakte Aufbau eines 32 Bit TDC-Datenpakets ist in Tabelle 4 dargestellt. Weitere übertragene Datenpakete sind der TDC-Header, Debug Daten, der EPOCH-Counter und ein TDC-Trailer. Diese Daten werden allerdings für die Kalibrierung nicht genutzt und ohne Manipulation durch die Kalibrierungs-Entität geführt. Dies ist nötig, damit es keine Probleme im Timing der Daten gibt (vgl. Abb. 20).

Die eigentliche Kalibrierung ist in zwei Abschnitte geteilt: Um eine lineare Kalibrierung durchführen zu können, muss für jeden Kanal der kleinste Wert der FineTime (Min) und der größte Wert der FineTime (Max) bekannt sein. Hieraus lässt sich die Differenz errechnen und die Steigung aus einer Look-Up Tabelle auslesen. Die Min und Max Werte variieren allerdings mit der Temperatur und müssen so immer neu erzeugt werden. Deshalb werden alle einkommenden TDC Timer-Daten (Bit 31 = 1) zum einen mit einer bestehenden (**current**) Kalibrierung kalibriert, zum ande-

ren aber auch für die neue (**next**) Kalibrierung genutzt. Die Anzahl an Daten zur Erzeugung einer neuen Kalibrierung ist über die Slow Control einstellbar (Register `0xe001`), jedoch durch ein Minimum von 10.000 limitiert.

Abbildung 20 zeigt den schematischen Aufbau der Kalibrierung mit allen Sub-Entitäten des VHDL Codes. Da die Daten zu jeder steigenden Flanke der Clock neu anliegen können, werden die Daten von Entität zu Entität weiter gereicht. Aufgaben, die parallel erledigt werden können, werden im gleichen Zeitschritt abgearbeitet. Zu Beginn der Kalibrierung wird in der Entität **read_cnt** aus den eingehenden Zeitdaten die Kanalnummer extrahiert. Desweiteren wird die Flag *DoCal* gesetzt und die Nummer des FPGAs ausgegeben. Im Falle eines TrbSc gibt es nur einen FPGA und dieser Wert bleibt dauerhaft auf 0. Die FPGA Nummer ist für eine mögliche Implementierung in ein Trb3 vorgesehen. In der Entität **compare_old** wird geschaut, ob es sich bei den Daten um den gleichen FPGA und Kanal wie im Zeitschritt zuvor handelt. In diesem Fall wird die Flag *read* auf 0 gesetzt und es wird im folgenden auf bereits geladene Daten zurückgegriffen.

Um nun eine Kalibrierung durchzuführen, muss der bisherige Wert des Kanals für die Min und Max Werte gelesen werden. Damit die nächste Kalibrierung vorbereitet werden kann, muss zusätzlich der aktuelle Min und Max Wert für die nächste Kalibrierung ausgelesen werden und anschließend ggf. geändert werden. Um überhaupt zu wissen, wann eine neue Kalibrierung gebraucht wird, muss zusätzlich für jeden Kanal eines jeden FPGA ein Zähler ausgelesen werden. Dies geschieht alles in den Entitäten **cnt_val**, **Memory_curr** und **Memory**. Die Hauptarbeit der Kalibrierung findet in der Entität **Cal_Limits_v2** statt. Diese Entität hat 4 parallele Aufgaben: Die Entität zählt die kalibrierten Daten mit und speichert den Zählerstand im nötigen Moment in den RAM. Desweiteren wird hier die Auswertung der aktuellen und nächsten Kalibrierung vollzogen. Die ausgelesenen Werte für die aktuelle Kalibrierung werden zu einem Delta Wert verrechnet und an die nächste Entität weitergegeben. Damit allerdings auch der Übergang zwischen zwei Kalibrierungen klappt, wird hier die Verarbeitung zwischen aktuellen (*current*) und nächsten Min/Max (*next*) Werten vollzogen. Die Bereitstellung der Min und Max Werte für die nächste Kalibrierungsphase wird ebenfalls in dieser Entität vollzogen. Ist der aktuelle Wert der Zeitdaten größer als der bisherige größte Wert für die nächste Kalibrierung, so wird der Max Wert durch die aktuellen Zeitdaten ersetzt und abgespeichert. Im Fall von Min-Werten wird geschaut, ob die aktuellen Daten kleiner sind. Diese Berechnungen sind nicht trivial, da es durch die schnelle Abfolge der Daten zu Problemen mit der Speicherung kommen kann. Kommen z.B. Daten aus Kanal 1, gefolgt von Kanal 2 und dann wieder Kanal 1, so ist der Min Wert des 1. Kanals zum Zeitpunkt der zweiten Abfrage für Kanal 1 noch nicht gespeichert. Hierzu werden u.a. Min, Max, FPGA und Kanal Werte gepuffert und in diesen Fällen genutzt.

Ein wichtiger Spezialfall für die Kalibrierung ist der Start der Datennahme. Zum Start des FPGAs existiert keine aktuelle Kalibrierung, die genutzt werden kann. Um nach dem Start oder einem Reset des FPGA trotzdem Daten kalibrieren zu können, wurden Standardwerte implementiert. Der FPGA liest beim Start spezielle Bereiche des Flash Speichers aus und nutzt diese Werte für die erste Kalibrierung. Auf diese Funktion wird später detaillierter eingegangen.

Ist der Min und Max Wert, sowie der Delta Wert der Kalibrierung, bekannt, wird die nötige Steigung der linearen Kalibrierung aus einer Look Up Table (LUT) ausgelesen. Da es sich bei der Berechnung einer Steigung um eine Division handelt, die sich in einem FPGA nur unter großem Aufwand implementieren lässt, wurde auf LUTs zurückgegriffen. Aus den bekannten Werten, zusammen mit der FineTime aus Bit 21-12 der Zeitdaten des TDC, wird nach Formel 2.3 die Kalibrierung durchgeführt. Um Rundungsprobleme und Ungenauigkeiten zu umgehen, wurde ein Faktor 1024 mit den Daten multipliziert und der Wert 512 addiert. Nach der Berechnung werden die letzten 10 Bits des kalibrierten Wertes gelöscht. Dadurch wurde die Multiplikation des Wertes 1024 rückgängig gemacht und durch die Addition der Zahl 512 auch die Rundung korrekt durchgeführt.

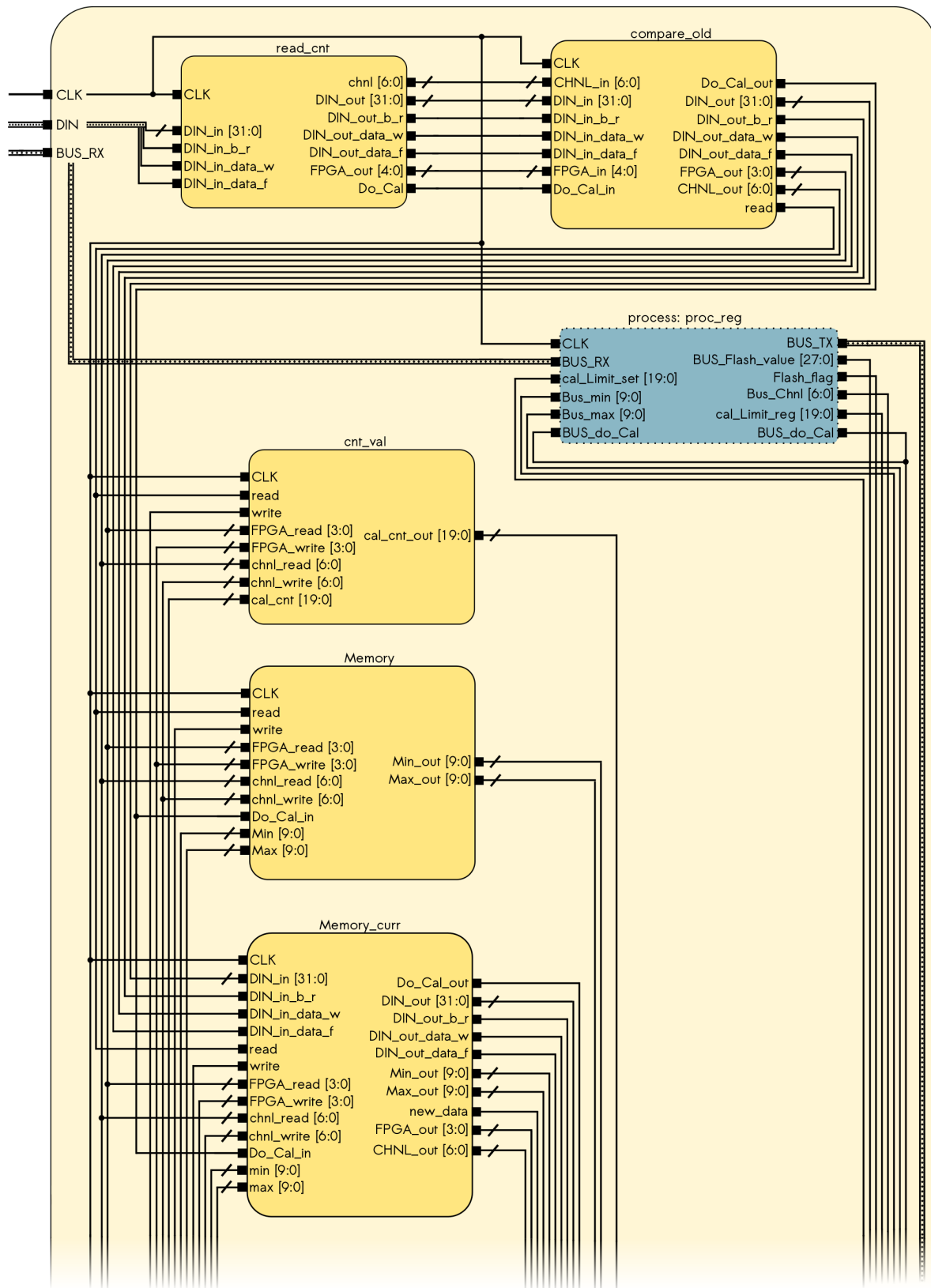
$$FineTime_{cal} = (FineTime - Min) \cdot slope + 512 \quad (2.3)$$

Die Ausgabewerte der Kalibrierung liegen nun zwischen 0 und 1000. Dies wurde so gewählt, da im Datenpaket der Zeitdaten 10 Bit für die FineTime vorgesehen sind. Um diese 10 Bit auch voll auszunutzen, wurde der Bereich 0 bis 1000 gewählt. Bei einer Software basierten Kalibrierung liegt die Ausgabe zwischen 0 und 500. Dies ist bei der Bearbeitung der Daten zu berücksichtigen. Ist eine FineTime kleiner als der Min Wert der Kalibrierung, so wird sie auf die Stelle 1010 gelegt. Ist die FineTime größer als der Max Wert, so wird sie auf die Stelle 1015 geschrieben.

Bei der Auswertung der Daten (z.B. Mit Go4³) ist darauf zu achten, dass die Daten, welche größer als 1000 sind, nicht genutzt werden sollten. Aus statistischer Fluktuation heraus gibt es ab und an Einträge an den Stellen 1010 und 1015.

Bei der Kompilierung des VHDL Codes kommt es leider zu Kollisionen mit Sittes aus der TDC Implementierung. Ohne die Unterstützung eines versierten TDC Entwicklers war es nicht möglich mehr als 11 Kanäle auf einem TrbSc zu implementieren.

³Go4: Objekt orientiertes, root basiertes System zur Analyse und und GUI.
https://www.gsi.de/en/work/research/experiment_electronics/data_processing/data_analysis/the_go4_home_page.htm



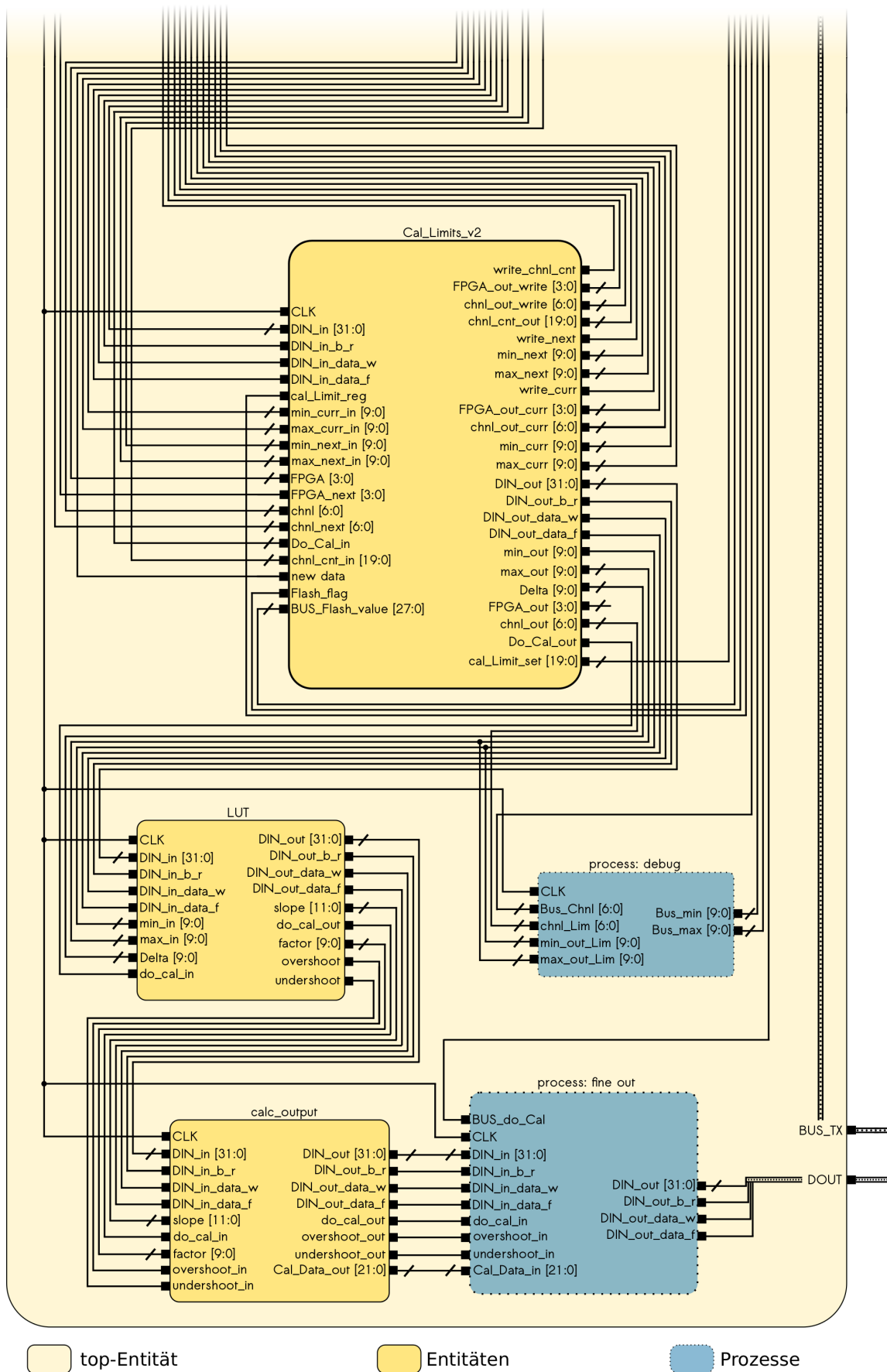


Abbildung 20: Schematic der TDC Kalibrierung.

2.2 Slow Control und Flash Werte

Über den Slow Control BUS des TrbNet ist es möglich, Werte an bestimmte Stellen des FPGA zu schreiben. Der Kalibrierung wurde hierzu ein 12 Bit Adressraum zugewiesen. Mit der Registeradresse 0xE000 bis 0xEFFF wird die Kalibrierung gesteuert. Die Adresse 0xE000 wird genutzt, um die Kalibrierungsfunktion zu aktivieren und zu deaktivieren. Ist der Datenwert in Register 0xE000 gerade, so ist die Kalibrierung deaktiviert. Ist der Wert ungerade, so ist sie aktiviert. Mit Adresse 0xE001 wird das Limit der Kalibrierung eingestellt. Dieser Wert gibt an, wie viele Daten für eine Kalibrierung benötigt werden. Der Standardwert ist 0x186A0 (100.000₁₀). Die Adresse 0xE003 ist für die Auswahl eines Kanals vorgesehen. Wird hier z.B. der Wert 4 eingetragen, so wird auf Adresse 0xE004 der Min und auf Adresse 0xE005 der aktuelle Max Wert von Kanal 4 zum Auslesen bereitgestellt. Die Register werden über den Befehl

```
trbcmd r <TrbAdresse> <RegisterAdresse>
```

ausgelesen. Um in die Register zu schreiben, wird der Befehl

```
trbcmd w <TrbAdresse> <RegisterAdresse> <Daten>
```

verwendet.

Ab Adresse 0xE007 wird der Standardwert der Kalibrierung der einzelnen Kanäle gesetzt. 0xE007 entspricht Kanal 0 von FPGA 0, 0xE008 ist Kanal 1 von FPGA 0 und so weiter. Bisher wird nur ein FPGA genutzt und somit wurde die Funktionalität auch lediglich für diesen einen implementiert. Eine Erweiterung auf mehrere FPGAs ist sehr schnell zu implementieren. Um den Min und den Max Wert zu übertragen, muss der Min Wert in den Bits 0-9 und der Max Wert in den Bits 10-19 des Datenfeldes der Slow Control Adressierung stehen. Da diese Adressen primär für den Flash-Speicher gedacht sind, ist es nicht zu empfehlen, Werte manuell zu hinterlegen. Um in den Flash zu schreiben, wird das Programm *flash_settings.pl* aus *daqtools/tools*⁴ genutzt. Dem Programm wird der Name einer Liste der Registeradressen mit den entsprechenden Werten und die TrbAdresse übergeben.

Die im Flash hinterlegten Werte zur Kalibrierung werden bei jeden (Neu-)Start des Trb Boards ausgelesen und für den ersten Kalibrierungsblock (bis durch die eingehenden Daten eine neue Kalibrierung erzeugt wurde) verwendet. Um einer falschen Kalibrierung zu Beginn durch Temperaturwechsel entgegen zu wirken, ist es zu empfehlen die Min und Max Werte des Flash-Speichers bei signifikanten Änderungen (der vom FPGA gemessenen Temperatur) zu erneuern.

⁴<http://jspc29.x-matter.uni-frankfurt.de/git/?p=daqtools.git;a=tree>

2.3 Auswertung ohne Offset

Die neue FPGA Konfiguration mit der Kalibrierungsfunktionalität wurde auf das TrbSc aufgespielt. Es wurden Messungen mit und ohne Offset (siehe Tabelle 3) durchgeführt. Hierbei wurden erneut die Temperaturen zwischen 5 °C und 50 °C in 5 °C Schritten verändert. Um die Pulse zu vergleichen, wurde für jeden Kanal ein Referenzkanal festgelegt. Dieser ist Tabelle 2 zu entnehmen. Die Ergebnisse der Datenaufnahme wurden mit der softwareseitigen linearen Kalibrierung verglichen und sind im Folgenden zu sehen. Abbildung 21 zeigt die RMS Werte der einzelnen Kanäle des TDCs für alle gemessenen Temperaturen bei einer linearen Software-Kalibrierung ohne Offset. In Abb. 22 ist das gleiche Histogramm für die FPGA basierte online Kalibrierung zu sehen.

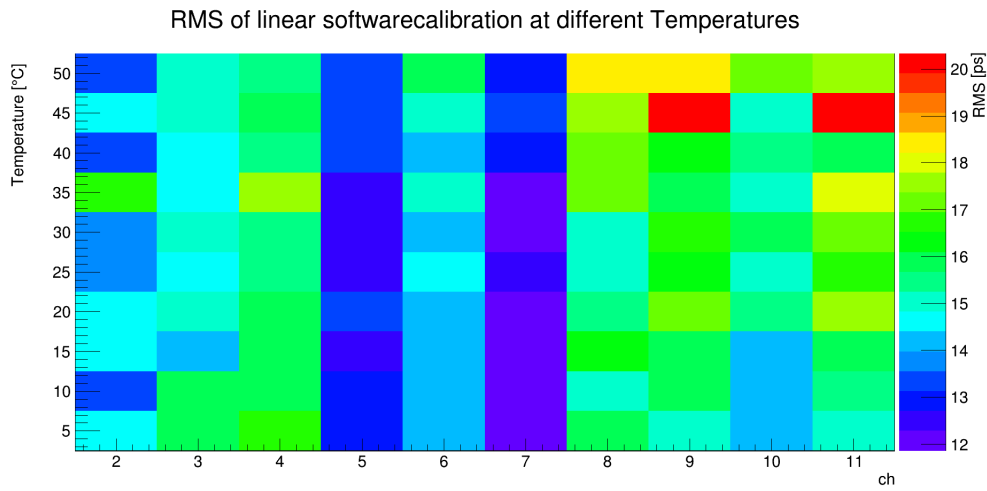


Abbildung 21: Übersicht über die einzelnen RMS Werte in Pikosekunden für alle Kanäle zu allen gemessenen Temperaturen mit Hilfe der Software basierten Kalibrierung.

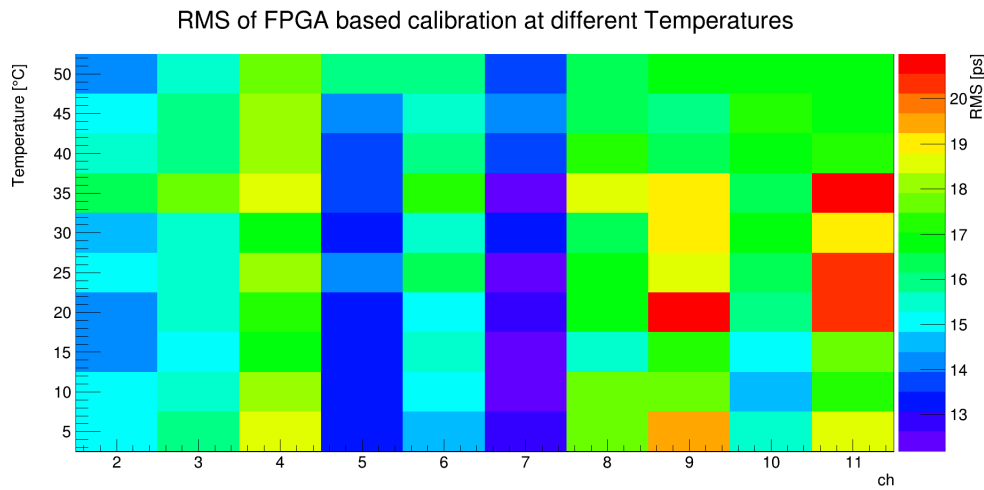


Abbildung 22: Übersicht über die einzelnen RMS Werte in Pikosekunden für alle Kanäle zu allen gemessenen Temperaturen mit Hilfe der FPGA basierten Kalibrierung.

Wie man im Vergleich beider Histogramme gut sehen kann, variieren in beiden Verfahren die RMS Werte eines Kanals über verschiedenen Temperaturen hinweg nur minimal. Schaut man sich einzelne Kanäle an, so ist die sehr hohe Konstanz des RMS Wertes in Abhängigkeit der Temperatur noch deutlicher sichtbar. Abb. 23 zeigt die RMS Werte für verschiedene Temperaturen in Kanal 5. In Abb. 24 ist das gleiche für Kanal 11 zu sehen. In beiden Fällen ist deutlich zu erkennen, dass die Präzision über den kompletten Temperaturbereich innerhalb von ca. 3 ps stabil bleibt. Desweiteren fällt auf, dass die Präzision für die FPGA basierte Kalibrierung in der Regel um 1-3 ps geringer ist als bei der Software basierten linearen Kalibrierung.

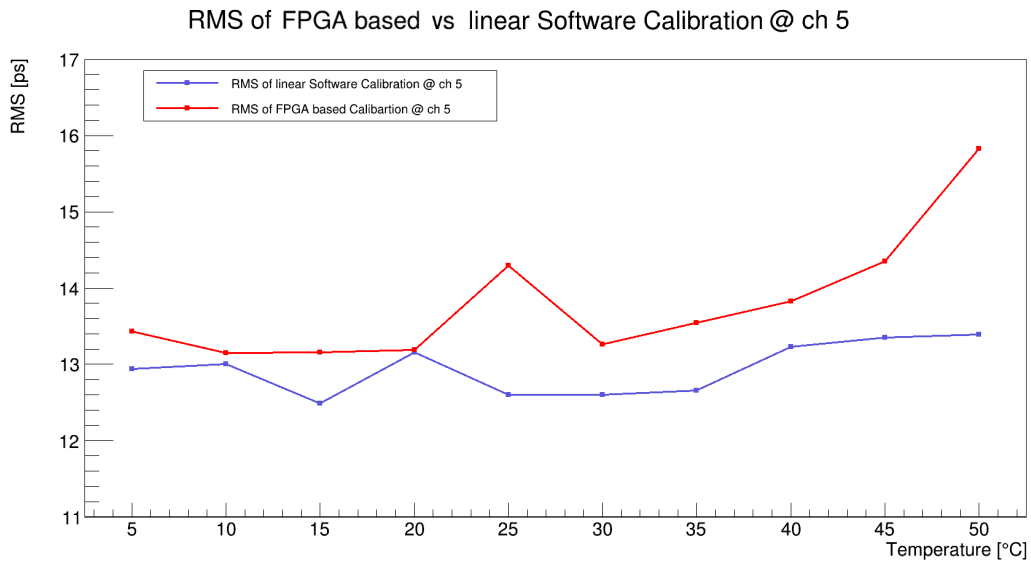


Abbildung 23: Gemessene RMS Werte von Kanal 5 eines TrbSc Boards bei allen betrachteten Temperaturen im Vergleich der zwei Kalibrierungsmethoden.

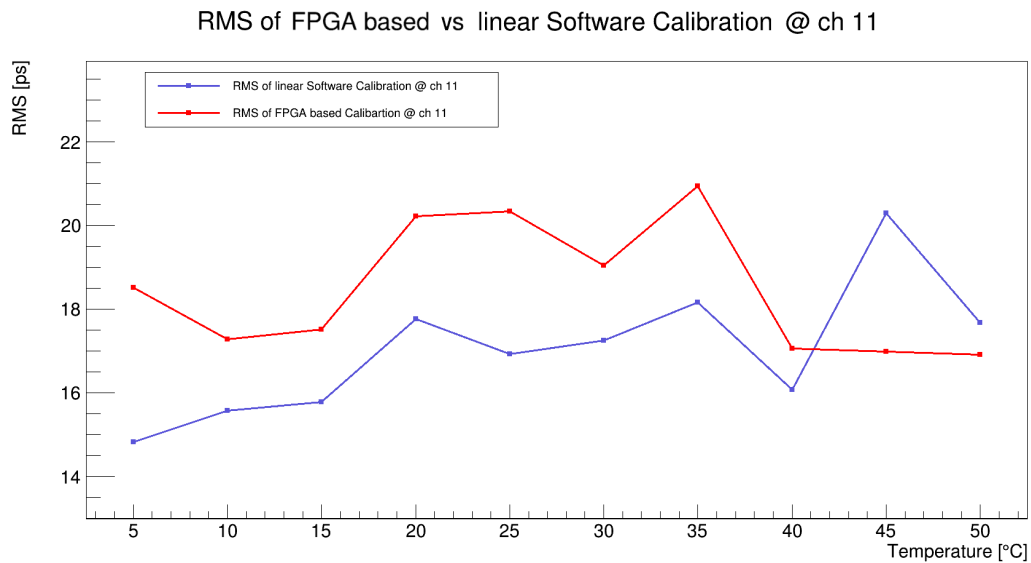


Abbildung 24: Gemessene RMS Werte von Kanal 11 eines TrbSc Boards bei allen betrachteten Temperaturen im Vergleich der zwei Kalibrierungsmethoden.

Dies lässt sich auf die leicht unterschiedliche Festlegung der Min und Max Werte und die Bedingung bei der Datennahme zurückführen. Im Falle der FPGA basierten Lösung wurden nach je 100.000 Einträgen Min und Max Werte festgestellt. Im Fall der Software Lösung werden alle 2 Millionen Einträge das Maximum und das Minimum ausgewählt. Diese Auswahl bezieht sich allerdings auf die Daten der steigenden Flanke. Bei der FPGA basierten Lösung wird nicht zwischen steigender und fallender Flanke unterschieden. Der größte Einfluss ist aber durch die Festlegung der Min und Max Grenzen aus 2 Millionen Daten auszumachen. Da aufgrund der Heiz-/Kühl-Apparatur die Temperatur während der Messung im Bereich von bis zu $\pm 1^\circ\text{C}$ variieren, ist die Festlegung eines solchen Min und Max Wertes mit einem großen Fehler behaftet. Somit weichen die Werte der Software-Lösung potentiell stärker von dem “echten” Maximum ab. Die Werte des Minimums bleiben konstant. Ist der Maximum Wert zu groß gewählt, so zeigt sich durch Tests, dass dadurch der RMS Wert besser wird. Eine Abweichung um lediglich 1 Bin führt schon zu Variationen um etwa 2 ps im RMS Wert. Dieser systematische Fehler ist bei der Betrachtung der Daten immer zu berücksichtigen. Damit ergeben beide Methoden innerhalb dieses systematischen Fehlers dasselbe Ergebnis.

In Abb. 25 und 26 ist zur Verdeutlichung beispielhaft die Verteilung der FineTime von Kanal 5 eines TDC für die unkalibrierte und die (FPGA basierte) kalibrierte Variante dargestellt. Die FPGA basierte Version zeigt eine gleichmäßige Verteilung, die genau im vorgesehenen Bereich von 0 bis 1000 liegt. Die unkalibrierte Ausgabe des TDC weist ebenfalls eine gleichmäßige Verteilung auf. Der Bereich der FineTime Bins variiert aber je nach Kanal und Temperatur stark. Die Verteilung bewegt sich in der Regel in einem Bereich von 16 bis 480, kann aber auch leicht bei 470 oder 490 liegen.

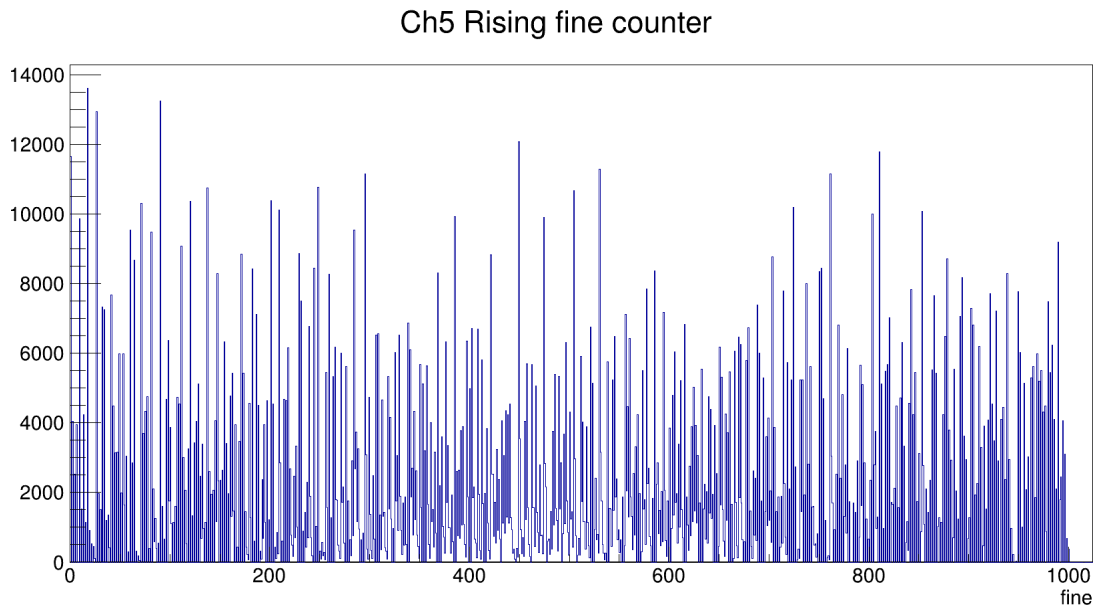


Abbildung 25: Verteilung der Einträge in den FineTime Bins für Kanal 5 der FPGA basierten Lösung mit Kalibrierung.

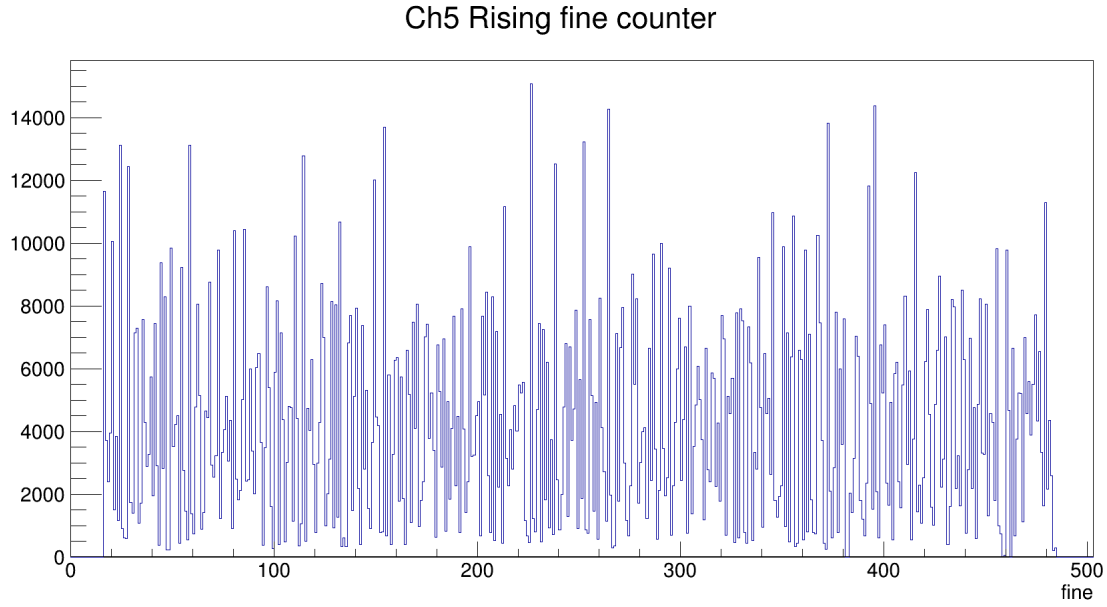


Abbildung 26: Verteilung der Einträge in den FineTime Bins für Kanal 5 ohne Kalibrierung.

In Abb. 27 ist der Mittelwert der Präzision (RMS) über alle Temperaturen im vermessenen FPGA gezeigt. Die Software basierte als auch die im FPGA implementierte Kalibrierung ergeben mittlere Präzisionen zwischen 12 ps und 19 ps. Der Unterschied zwischen beiden Methoden ist marginal.

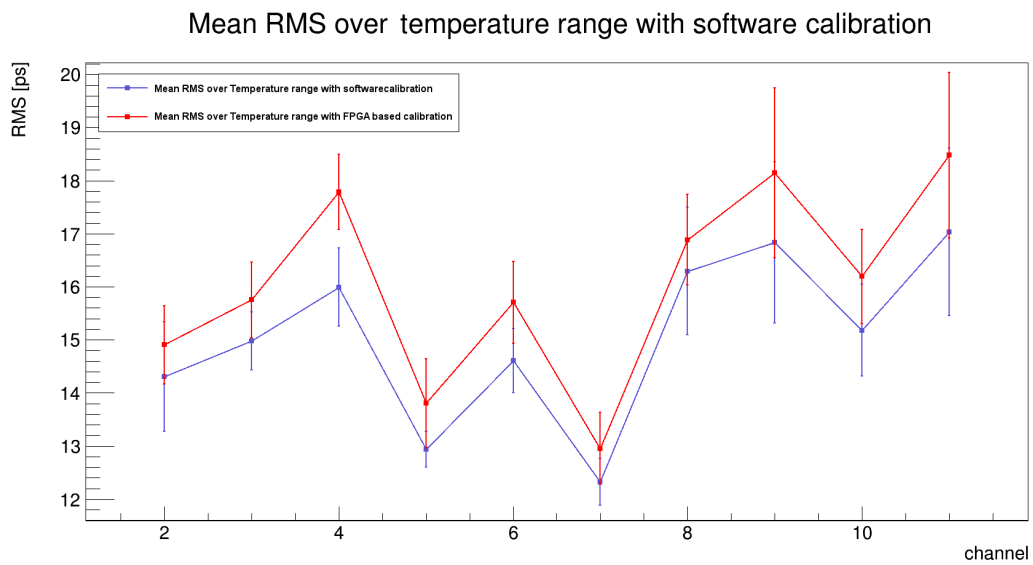


Abbildung 27: Übersicht über den mittleren RMS Wert aller gemessenen Temperaturen für jeden Kanal des TDC.

Der RMS Wert der beiden Kalibrierungen liegt unter konstanten Laborbedingungen, also bei einer FPGA-Temperatur von 34,4 °C, noch näher zusammen. Eine solche Messung ist in Abschnitt 2.4 in Abbildung 37 zu sehen. Die größte Stärke der FPGA basierten Methode zeigt sich bei einer Datenaufnahme mit einem schnellen Temperaturabfall von 50 °C auf 5 °C. Hierzu wurde die Datenaufnahme bei 50 °C gestartet. Anschließend wurde die Temperatur kontinuierlich reduziert, so dass nach etwa 5 Minuten 5 °C erreicht wurden. Daraufhin wurde die Datennahme gestoppt. In Abb. 28 sind die dazugehörigen RMS Werte für die Software und FPGA basierte Kalibrierung dargestellt. Man sieht deutlich, dass die FPGA basierte Lösung durch die ständige Neukalibrierung mit den einkommenden Daten gegen Änderungen der Temperatur nahezu immun ist. Im Vergleich dazu ist die Präzision der Software basierten Kalibrierung deutlich schlechter bei gleichzeitig starken Fluktuationen von Kanal zu Kanal.

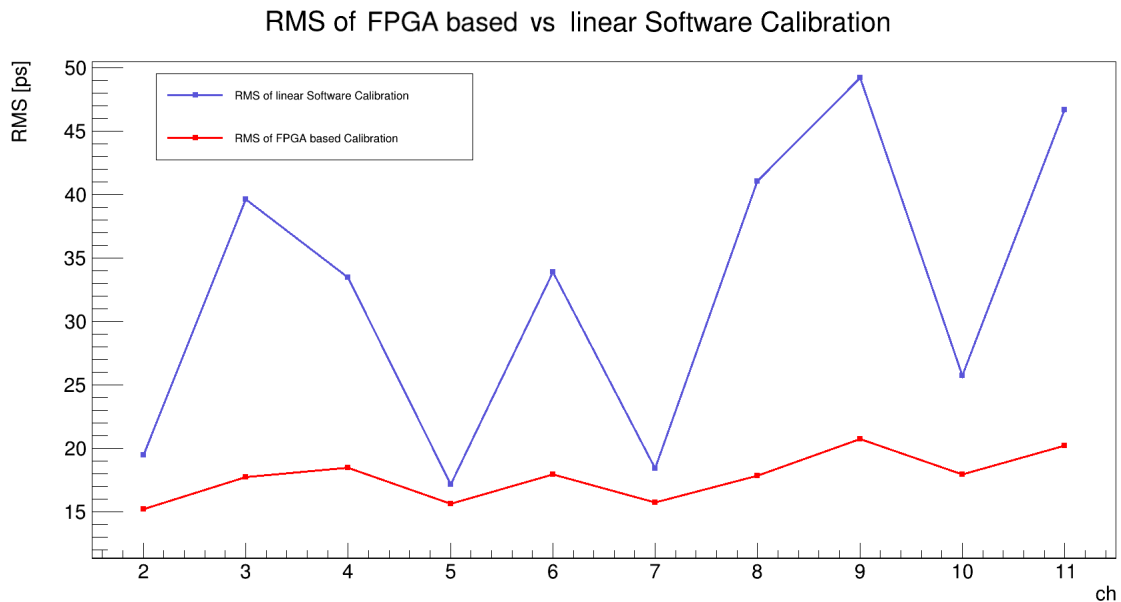


Abbildung 28: RMS Werte der einzelnen Kanäle bei einem Temperaturabfall zwischen 50 °C und 5 °C. blau: Software basierte Kalibrierung. rot: FPGA basierte Kalibrierung.

2.4 Auswertung mit Offset

Um realistischere Bedingungen zu schaffen, wurden Offsets auf die Pulser-Signale gegeben (siehe Tabelle 3). Unter diesen Bedingungen wurde nun die gleiche Analyse wie ohne Offset durchgeführt. Abbildung 29 und Abbildung 30 zeigen, wie sich durch den Offset die Präzision in den betroffenen Kanälen verschlechtert hat. Allerdings bleibt die Präzision weiterhin über den Temperaturbereich hinweg stabil.

Diese Konstanz verdeutlicht sich anhand von Abb. 31 und 32. Für Kanal 5 und auch Kanal 11 ist der RMS Wert sehr konstant, aber deutlich höher als ohne Offset. Dieser Effekt kommt aus dem anhand von Abb. 19 bereits aufgezeigten Zusammenhang aus Offset und RMS Wert.

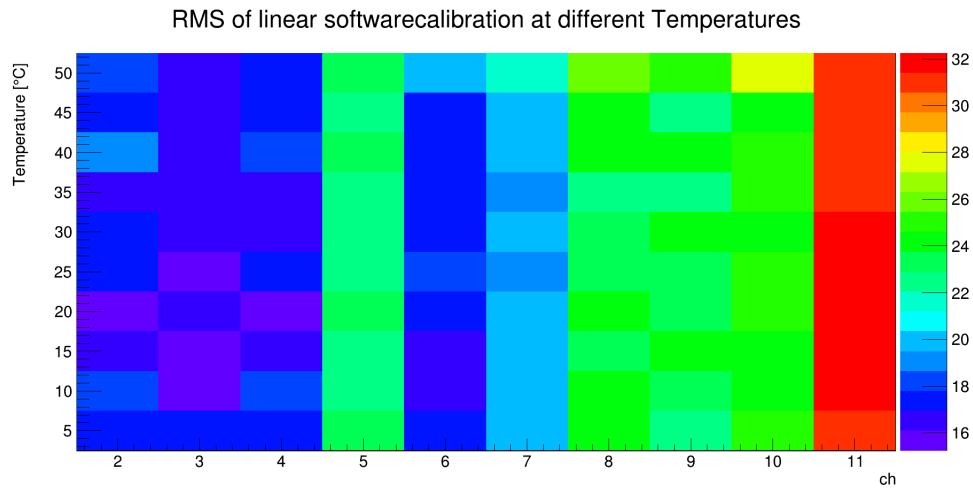


Abbildung 29: Übersicht über die einzelnen RMS Werte in Pikosekunden für alle Kanäle in Abhängigkeit der Temperatur mit Hilfe der Software basierten Kalibrierung.

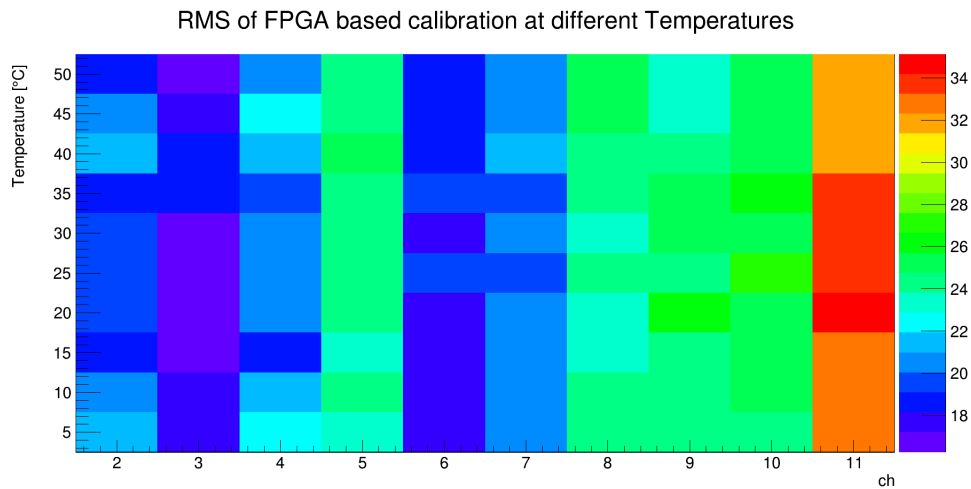


Abbildung 30: Übersicht über die einzelnen RMS Werte in Pikosekunden für alle Kanäle in Abhängigkeit der Temperatur mit Hilfe der FPGA basierten Kalibrierung.

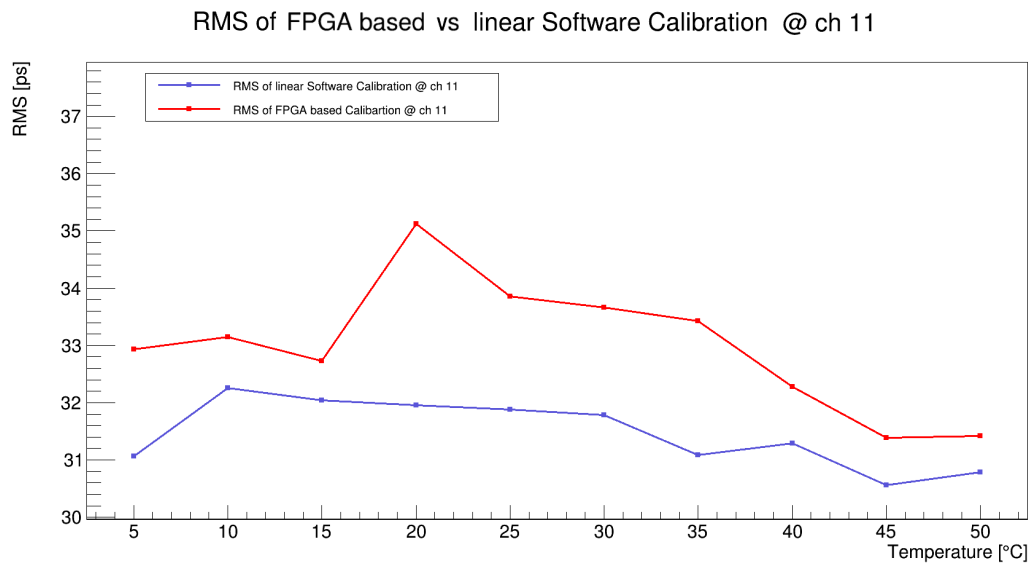


Abbildung 31: Übersicht über die einzelnen RMS Werte in Pikosekunden in Abhängigkeit der Temperatur in Kanal 11.

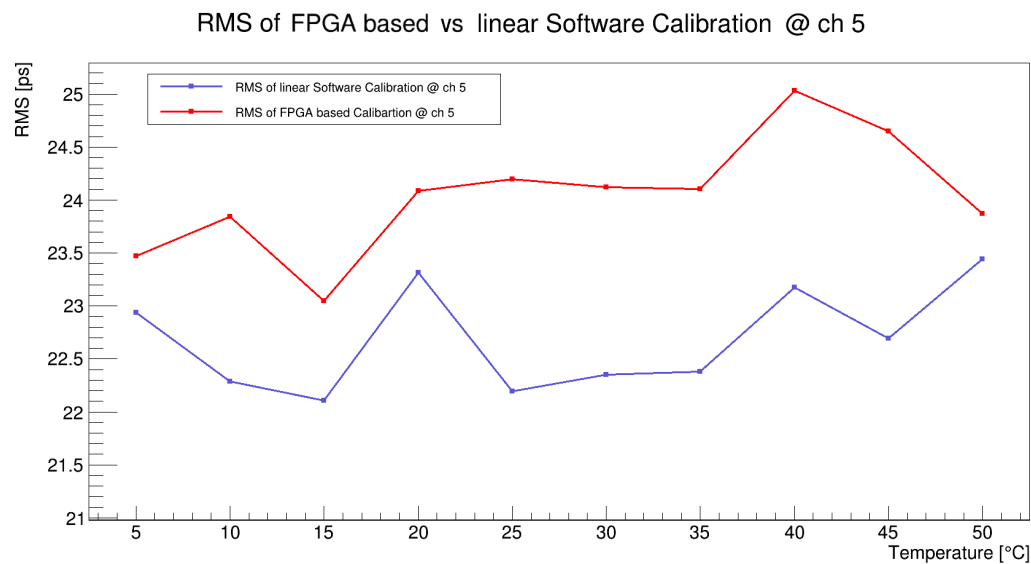


Abbildung 32: Übersicht über die einzelnen RMS Werte in Pikosekunden in Abhängigkeit der Temperatur in Kanal 5.

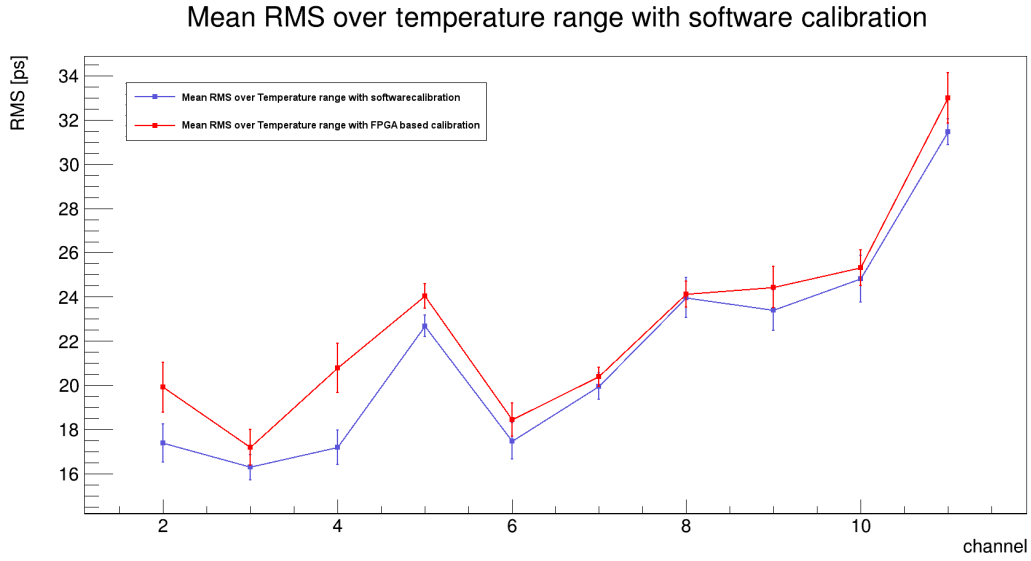


Abbildung 33: Übersicht über die Mittelwerte des RMS über alle gemessenen Temperaturen für alle Kanäle.

In Abbildung 33 ist der Mittelwert aus den Messungen bei Temperaturen von 5 °C bis 50 °C für jeden Kanal des TrbSc gezeigt. Analog zur Datennahme ohne Offset sieht man deutlich, dass die Software und FPGA basierte Kalibrierung sich nur um 1-3 ps unterscheiden. Selbst bei einem großen Offset von 158 ns in Kanal 11 ist die Präzision mit ca. 33 ps immer noch sehr gut, allerdings 20 ps schlechter als mit der Bin-by-Bin Kalibrierung ohne Offset.

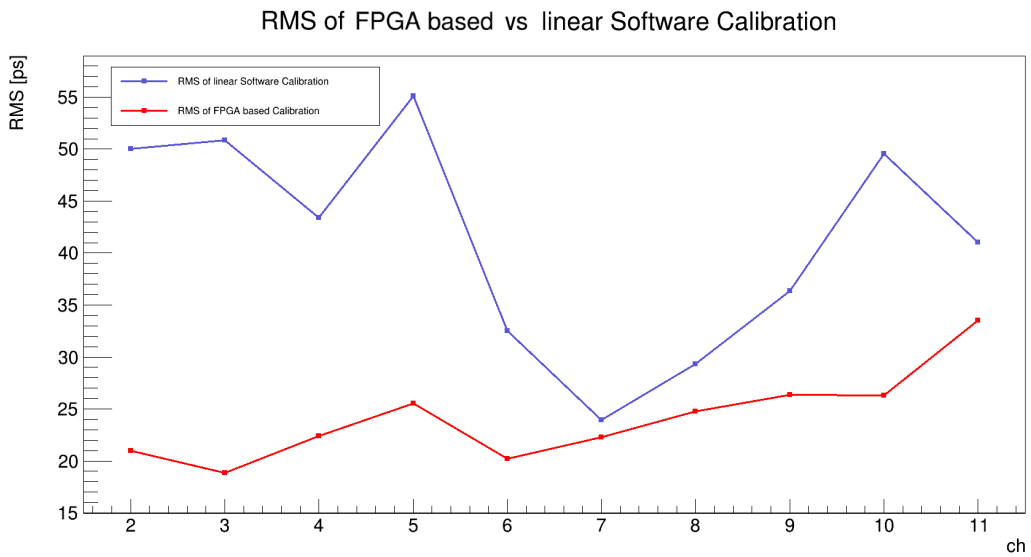


Abbildung 34: RMS Werte für jeden Kanal bei einem schnellen Temperaturabfall zwischen 50°C und 5°C.

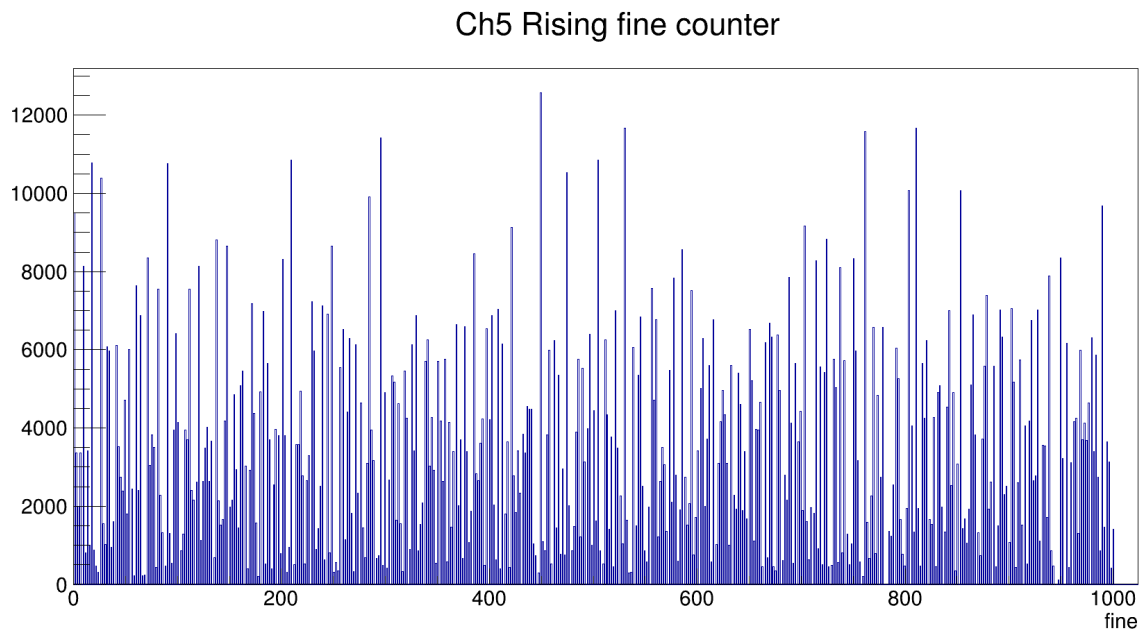


Abbildung 35: Verteilung der FineTime Bins in Kanal 5 bei der steigenden Flanke für die FPGA basierte Kalibrierung.

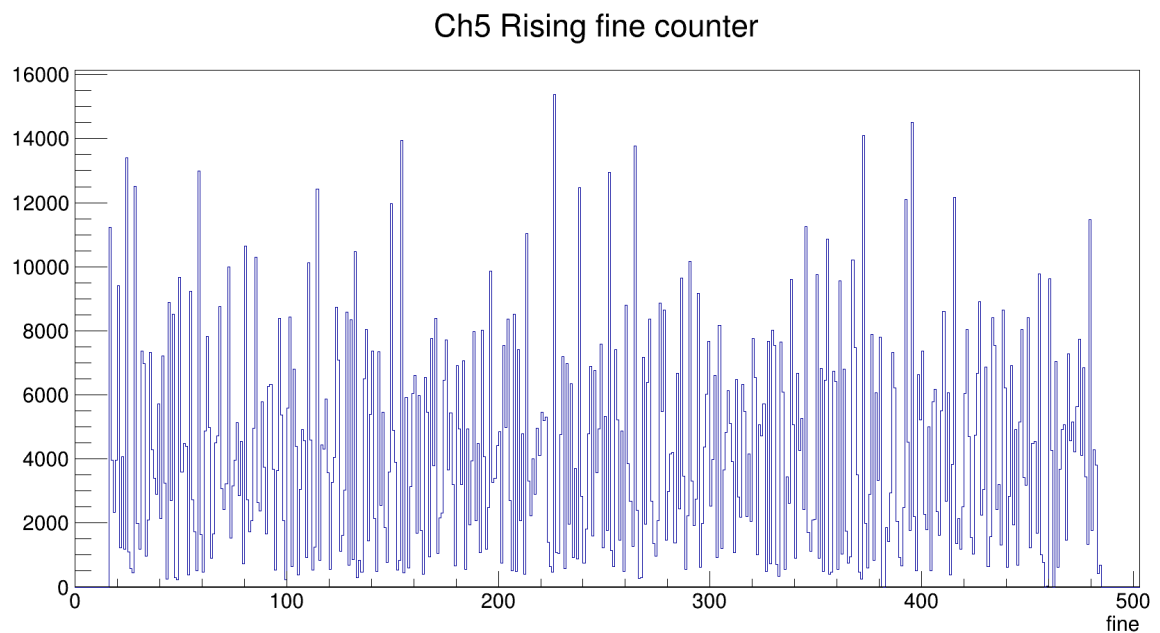


Abbildung 36: Verteilung der FineTime Bins in Kanal 5 bei der steigenden Flanke für die Software basierte Kalibrierung.

Die Präzision ist im Falle eines durchgehenden Temperaturabfalls zwischen 50 °C und 5 °C erneut gemessen worden. Wie man in Abb. 34 sieht, liegt die FPGA basierte Kalibrierung in einem Bereich zwischen 20 ps und 30 ps. Im Vergleich zu Abb. 31, 32 und 33 zeigt sich, dass Kanal-spezifisch ähnliche Präzisionen erreicht werden wie bei konstanter Temperatur und dass die Schwankungen zwischen 20ps und 30ps von der Charakteristik der einzelnen Kanäle bestimmt sind. Die Software basierte Kalibration zeigt aufgrund der schlechteren Fähigkeit der Temperaturänderung zu folgen wiederum eine schlechte Präzision. Allerdings sieht man auch, dass in einigen Kanälen der Unterschied, im Vergleich zur Messung ohne Offset, deutlich geringer ausfällt.

Die FineTime Verteilung in Abb. 35 und Abb. 36 zeigt - analog zur Messung ohne Offset - , dass die Bins gleichmäßig ohne auffällige Häufungen in bestimmten Bereichen verteilt sind. Im Falle der FPGA basierten kalibrierten FineTime ist ebenfalls zu sehen, dass die Bins 1010 und 1015 leer sind und keine Daten außerhalb der Kalibrierung lagen.

Um zu verdeutlichen, wie der Einfluss der Temperaturschwankungen im Bereich von $\pm 1^\circ\text{C}$ während der Messung ist, wurde zusätzlich eine Messung unter konstanten Laborbedingungen mit einer Temperatur von konstanten 34,4 °C durchgeführt. Abbildung 37 zeigt die RMS Werte für eine solche Messung. Hierbei wurden drei Messungen mit der FPGA basierten Kalibrierung (rot) und zwei Messungen mit der Software basierten Kalibrierung (blau) durchgeführt. Bis auf in Kanal 2 und 4 stimmen die Messwerte des RMS bei allen Messungen überein. Der Unterschied zwischen der Software und FPGA basierten Kalibrierung ist verschwindend gering. Im Vergleich zur Messung mit leicht variierender Temperatur ist der Unterschied zwischen beiden Methoden nahezu komplett verschwunden.

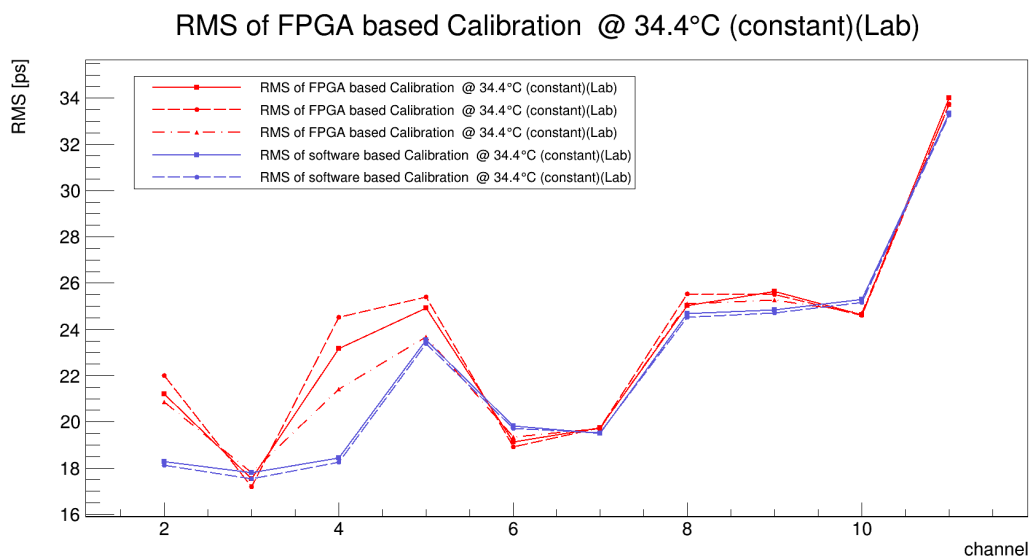


Abbildung 37: Messung der RMS Werte aller Kanäle bei konstanten 34,4 °C. Drei Messungen wurden für die FPGA basierte (rot) und zwei Messungen für die Software basierte (blau) Kalibrierung durchgeführt (mit Offset).

2.5 Zusammenfassung

Zur Verbesserung der TDC Präzision wurde eine Zeitkalibrierung entwickelt und an einem TrbSc mit einem Lattice ECP3 FPGA analysiert. Die Funktionalität kann später auch auf DiRICH Boards zur Auslese von MAPMT Signalen des H12700 für den HADES und CBM RICH portiert werden. Zunächst wurden bereits vorhandene Software basierte Kalibrierungs (linear und Bin-by-Bin) Verfahren verglichen und daran gezeigt, dass eine lineare Kalibrierung eine ausreichende Präzision liefert. Der Unterschied beider Methoden beträgt 5 ps bis 10 ps, je nach betrachtetem Kanal. Basierend auf dieser Feststellung konnte eine FPGA basierte online Kalibrierung implementiert werden, die zu gleichen Ergebnissen wie die bisherige Software basierte lineare Kalibrierung führt. Die Präzision ist hierbei im Bereich von hervorragenden 12 ps bis 19 ps. Geht man allerdings von einem nicht idealen System aus, in dem die einzelnen Kanäle zu unterschiedlichen Zeiten Signale bekommen, so wird die Präzision mit zunehmendem Offset schlechter. Selbst bei einem Offset von 158 ns verschlechtert sich die Präzision nur um etwa 10 ps. Systematische Untersuchungen der Präzision in Abhängigkeit der Temperatur, auch bei sich schnell ändernden Werten, zeigen eine hervorragende Stabilität. Leider stößt die FPGA basierte Kalibrierung derzeit durch die Implementierung des TDC beim Erstellen des FPGA Designs an seine Grenzen und es konnte bisher keine Implementierung mit mehr als 11 Kanälen erzeugt werden. Dieses Problem lässt sich in Zukunft durch den Einsatz eines versierten FPGA Entwicklers, der mit der TDC Implementierung vertraut ist, lösen. Zum Zeitpunkt der Anfertigung dieser Arbeit stand leider noch kein solcher Programmierer zur Verfügung.

Der Code der FPGA basierten Implementierung ist online unter http://jspc29.x-matter.uni-frankfurt.de/git/?p=trb3sc.git;a=tree;f=code/tdc_calib abrufbar.

3 DAC FPGA für den DiRICH2

Die RICH Detektoren des HADES und des CBM Experimentes werden in Zukunft beide mit Hilfe von DiRICH Boards in der Auslese betrieben. Nachdem es bei der ersten Iteration des DiRICH Boards zu Rauschen kam, wurde für die zweite Iteration viel Aufwand in die Beseitigung dieses Rauschens gesteckt. Neben der Optimierung des Layouts wurden zwei zusätzliche Lattice MachXO3-LF-4300 FPGAs auf dem Board aufgebracht (siehe rote Rechtecke in Abb. 38 rechts). Diese werden genutzt, um eine konstante Schwellenspannung ohne Temperaturschwankungen zu erzeugen und übernehmen damit eine DAC (digital to Analog converter) Funktionalität.

Die so erzeugte Schwelle wird als Referenz auf den Diskriminator des zentralen FPGA gegeben. Dieser Diskriminator ist durch die LVDS Eingänge des zentralen ECP5 FPGA implementiert. Die Signale der angeschlossenen MAPMTs werden über einen Amplifier verstärkt und als eigentliches Signal des Diskriminators auf die LVDS Eingänge geführt. Durch das Einstellen des Schwellenwertes kann so das zu detektierende Signal vom Rauschen separiert werden.

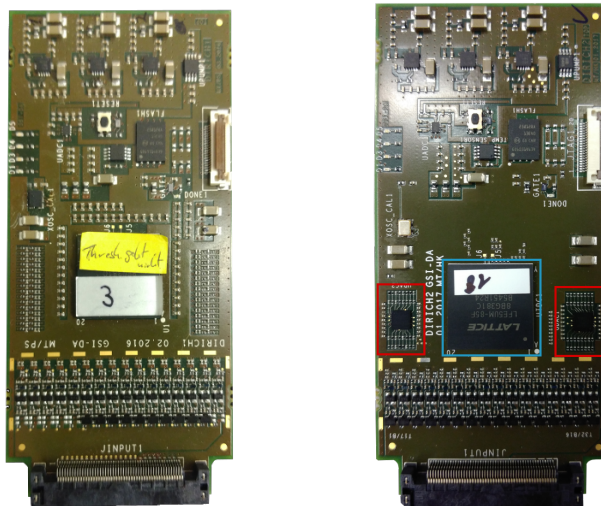


Abbildung 38: Links: DiRICH Version 1. Rechts: DiRICH Version 2. Beim zentralen FPGA handelt es sich um einen Lattice ECP5 (blau), bei den DAC FPGAs um Lattice MachXO3-LF-4300 (rot).

Beide DAC-FPGAs (siehe rote Rechtecke in Abb. 38 rechts) kommunizieren über das *Serial Peripheral Interface* (SPI) mit dem zentralen Lattice ECP5 FPGA, in dem der TDC untergebracht ist. Für jeden der DAC-FPGAs ist dazu auf dem zentralen FPGA ein SPI Interface hinterlegt. Die Designs beider DAC-FPGAs sind identisch. Lediglich die Umlegung der Schwellen auf die Kanäle unterscheidet sich. Hierzu wurde ein Pin angelegt, der durch die anliegende Spannung dem FPGA sagt, wie die Weitergabe der Schwellen erfolgen soll. Jeder der zwei FPGAs setzt die Schwellen für 16 Kanäle des DiRICH Boards. Das komplette DiRICH Board besitzt 32 Kanäle.

3.1 SPI-Slave

Für die SPI Kommunikation wurde kein festes Protokoll definiert. Dadurch unterscheiden sich diverse Implementierungen von SPI. Der Aufbau des verwendeten SPI Protokolls ist in Abb. 39 gezeigt. Es werden 32 Bit große Datenpakete übergeben. Die Bits 31-24 enthalten die Adresse des Registers, das angesprochen werden soll. Bit 23 enthält die Funktionalität (Modus). Ist dieses Bit = 1, so wird geschrieben. Ist es 0, wird gelesen. Die Stellen 22 bis 16 sind ohne Funktion. Die Bits 15 - 0 werden für die Daten bereit gehalten.



Abbildung 39: Schematische Übersicht des Aufbaus der 32 Bit Daten der SPI Verbindung.

3.2 DAC Implementierung durch eine Δ - Σ -Modulation.

Für die MAPMT Auslese generieren die DAC-FPGAs eine konstante Spannung, die in für den Diskriminator des TDC FPGA (Low Voltage Differential Signal (LVDS) Eingang) als Schwelle verwendet wird. Der gewünschte Schwellenwert wird extern bestimmt und im TDC-FPGA hinterlegt und an die DAC-FPGAs weitergeleitet. Die Erzeugung des gewünschten Schwellenwertes erfolgt über eine sogenannte Δ - Σ -Modulation. Die Ausgabe ähnelt der einer Pulsweiten-Modulation (PWM). Ein digitales Signal wird hierbei in ein analoges Signal gewandelt, indem durch die Ausgabe von 0 oder 1 ein Verhältnis zwischen maximaler Ausgangsspannung eines Pins und keiner Spannung an einem Pin gebildet wird. Liegt zum Beispiel dauerhaft eine 1 an, so misst man die maximale Ausgangsspannung. Die theoretische Umsetzung eines solchen Digital zu Analog Wandlers (DAC), basierend auf der Δ - Σ -Modulation, ist in Abb. 40 dargestellt. Durch die Addition in Kombination mit einem Flip-Flop wird ein Wert bis auf eine bestimmte Schwelle inkrementiert.

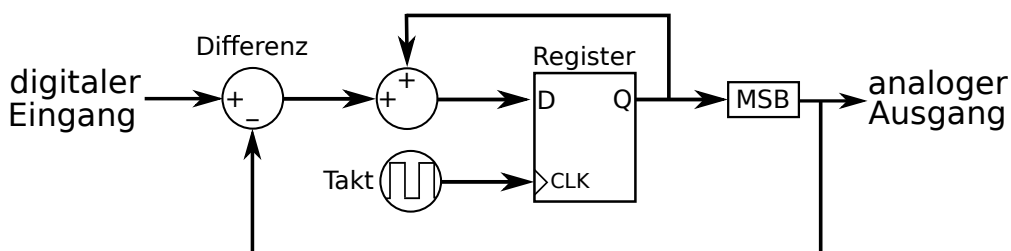


Abbildung 40: Schematischer Aufbau einer Δ - Σ -Modulation [22].

Wird diese Schwelle überschritten (meist einfach das Most Signifikant Bit (MSB)), wird eine 1 ausgegeben und diese auf den Eingang zurückgeführt, damit wieder unterhalb der Schwelle addiert wird.

Im FPGA ist dies ähnlich implementiert. Durch die Addition einer 16 Bit Zahl (der Schwellenwert) auf einen 17-Bit Wertebereich, wird solange addiert bis es eine Änderung im 17. Bit des 17-Bit-Wertebereichs gibt. Wird weiter gezählt, als der 17-Bit Wertebereich her gibt, so kommt es zu einem Überlauf und es wird Modulo $0x2000$ weiter gezählt. Jeder Wechsel des 17. Bit erzeugt in der Ausgabe der PWM nun eine Eins. Wechselt der Wert nicht, so wird eine Null ausgegeben. Die Größe des 16-Bit Schwellenwerts bestimmt so das Verhältnis aus 0 und 1 auf der Ausgabe. Das Maximum ist in dieser Implementierung auf $0xFFFF$ festgelegt.

Abbildung 41 zeigt anhand eines Beispiels die PWM Ausgabe des FPGA. Der mit roten Balken markierte Bereich zeigt gerade eine Periode. Wird ein Startwert (16-Bit Zahl) von z.B. $0x6000$ gesetzt, so ist die Ausgabe während 62.5% der Periode eine 0, während 37.5% eine 1. Damit wird die Ausgangsspannung auf 37.5% des maximalen Wertes von 2.5 V gelegt. Sie beträgt also 0.9375V. Diese Spannung entspricht der an den TDC-FPGA weitergeleiteten Schwelle für die MAPMT Signale.

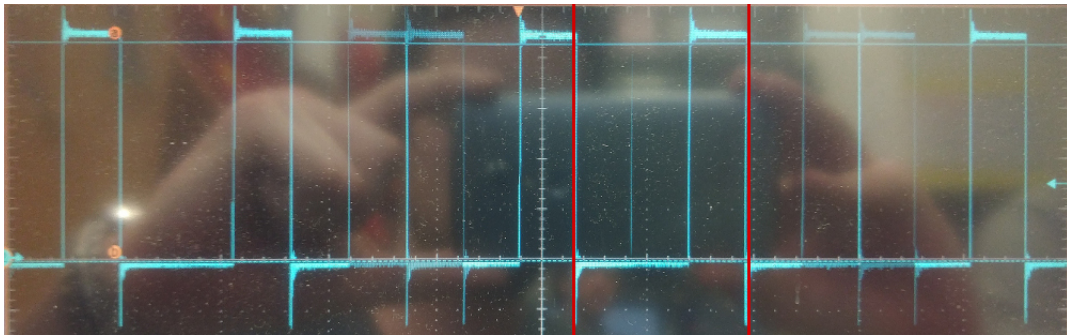


Abbildung 41: Oszilloskop Aufnahme der Ausgabe eines PWM Ports eines DAC-FPGAs. Der rot markierte Bereich ist eine Periode der PWM.

3.3 Flash Speicher

Im späteren HADES und CBM RICH Experiment werden mehr als 1000 DiRICH Boards zum Einsatz kommen. Aufgrund der starken Belastung des Netzes wäre es nicht sehr praktikabel, nach jedem Reset alle Schwellen neu per SPI zu schreiben. Als Lösung wird der interne Flash des MachXO3LF-4300 zur Speicherung der Schwellenwerte genutzt. Bei jedem Reset des DiRICH Boards wird so automatisch der Flash ausgelesen und die dort hinterlegten Werte in die PWM (Δ - Σ -Modulation) geführt. Damit der Flash nicht dauerhaft beschrieben wird, wird nicht jeder übergebene Wert direkt abgespeichert. Es wird lediglich dann gelesen bzw. geschrieben, wenn ein Reset stattfindet oder über SPI der nötige Befehl an den FPGA übermittelt wurde. In den Flash werden nur die Werte geschrieben, die im RAM gespeichert sind. Wird der Flash ausgelesen, werden die gelesenen Werte zum einen an die PWM übergeben, zum anderen in den RAM geschrieben.

3.4 Slow Control

Um mit den 2 DAC-FPGAs des DiRICH2 zu kommunizieren, werden über TrbNet die nötigen Informationen an den zentralen FPGA (TDC-FPGA) geschickt. Dieser leitet die zu übertragenden Daten anhand eines sogenannten *Chain* Wertes über das korrekte SPI Interface an den gesuchten DAC-FPGA weiter.

3.4.1 Lesen

Um Werte aus dem DAC-FPGA zu lesen, muss der Lese-Modus der SPI-Kommunikation aktiviert werden. Hierzu wird Bit 23 der SPI Kommunikation auf 0 gesetzt.

Auslesbar sind nur die Werte, die im RAM des DAC-FPGA stehen. Um auf diese zuzugreifen, muss die SPI-Adresse zwischen 0x10 und 0x1F liegen. Möchte man Adresse 0x10 auslesen, so wird der Schwellenwert aus dem RAM für Kanal 0 ausgelesen. Liest man Adresse 0x1F aus, so wird der Schwellenwert aus dem RAM für Kanal 15 ausgelesen. Die Adresse 0xEE kann genutzt werden, um Bit 0 bis 15 des Debug-Kanals von *sedcheck* auszulesen. Bit 16 bis 31 lässt sich mit der Adresse 0xEF auslesen.

Adresse	Beschreibung
0x00 bis 0x0F	keine Funktion
0x1Y	Ram Wert des Y-ten Kanals des FPGAs
0xEE	Bit 0 bis 15 des Debug Kanals von sedcheck
0xEF	Bit 16 bis 31 des Debug Kanals von sedcheck

Tabelle 5: Belegung des Adressbereichs zum Auslesen des FPGAs.

Die Funktion `PadiwaSendCmd`, welche in der PERL-Library in `daqtools.git/dmon/code/Dmon.pm`⁵ zu finden ist, kann zum Kommunizieren genutzt werden. Der Befehl muss wie folgt aufgebaut sein:

```
PadiwaSendCmd(<SPI-Kommando>,<DiRICH-Adresse>,<ID>);
```

Um z.B. den Schwellenwert von Kanal 12 des 2. FPGA (ID=1) auf dem DiRICH Board der Nummer 0x1202 zu lesen, muss der Befehl

```
PadiwaSendCmd(0x1C000000,0x1202,1);
```

lauten.

Der SPI Befehl setzt sich in diesem Fall aus der Register Adresse für Kanal 12 (0x1C), dem Lese-Modus gefolgt von den unbenutzten Bits (0x00) und dem Datenpaket, welches beim Lesen beliebig sein kann, zusammen.

⁵<http://jspc29.x-matter.uni-frankfurt.de/git/?p=daqtools.git;a=blob;f=dmon/code/Dmon.pm>

3.4.2 Schreiben

Damit man Werte in den DAC-FPGA schreiben kann, muss der Schreib-Modus (Bit 23 = 1) gewählt werden. Ist dieser gewählt, so ergeben sich nun diverse Optionen zum Schreiben.

Aus Gründen des Debugging ist es möglich, nur die Schwellen der PWM für einen Kanal zu setzen. Der gesetzte Wert wird nicht in den RAM geschrieben. Für diese Option ist der Adressbereich von 0x00 bis 0x0F reserviert. Will man den Schwellenwert eines Kanals ändern und den Wert in den RAM schreiben - dies ist das empfohlene Vorgehen -, so ist der Adressbereich von 0x10 bis 0x1F zu wählen. Damit die Werte aus dem RAM auch in den Flash geschrieben werden, muss ein beliebiger Datenwert an die SPI-Adresse 0x20 gesendet werden. Schreibt man einen beliebigen Wert an die Adresse 0x21, so wird der Flash ausgelesen. Die Adresse 0xEE bietet zusätzlich die Möglichkeit, die Control-Werte an die Entität *sedcheck* zu übermitteln.

Adresse	Beschreibung
0x00 bis 0x0F	Schwellenwert der PWM des X-ten Kanals des FPGAs.
0x10 bis 0x1F	RAM- und Schwellenwert des X-ten Kanals des FPGAs.
0x20	Befehl zum Beschreiben des Flash-Speichers.
0x21	Befehl zum Auslesen des Flash-Speichers.
0xEE	Bit 0 bis 15 des Debug Kanals von <i>sedcheck</i>

Tabelle 6: Belegung des Adressbereichs zum Beschreiben des FPGAs.

Um z.B. den Schwellenwert von Kanal 12 des 2. FPGA (ID=1) auf dem DiRICH Board der Nummer 0x1202 auf den Wert 0x6543 zu setzen, muss der Befehl

```
PadiwaSendCmd(0x1C806543,0x1202,1);
```

übermittelt werden.

Der SPI Befehl setzt sich in diesem Fall aus der Registeradresse für Kanal 12 (0x1C), dem Schreib-Modus gefolgt von den unbenutzten Bits (0x80) und dem Datenpaket (0x6543) zusammen.

Will man im gleichen FPGA auf dem gleichen DiRICH hingegen den Flash auslesen, so muss man den Befehl

```
PadiwaSendCmd(0x21800000,0x1202,1);
```

nutzen. Die übertragenen Daten im Datenfeld sind hierbei nicht von Belang.

3.5 Ergebnisse von Testmessungen im realen HADES Detektoraufbau

Die Funktionalität des DAC-FPGAs wurde im HADES-RICH test setup an der GSI getestet. Hierzu wurde eine Backplane mit MAPMTs, DiRICH2, Power Modul und Concentrator Board bestückt. Die Schwellen der DAC-FPGAs wurden mit Hilfe eines Schwellenscans von J. Förtsch [23] ermittelt und gesetzt. Beim Schwellenscan wird schrittweise für jeden Kanal des DiRICH Boards die Schwelle über den DAC-FPGA erhöht und die Rate der gemessenen Signale ausgelesen. Durch die Variation der Rate zeigt diese Messung unter anderem, dass die Schwellenwerte korrekt gesetzt werden. Die Abbildungen zeigen je drei DiRICH1 und fünf DiRICH2 Boards. Die DiRICH Adressen 0x120X sind die der DiRICH1 Boards, die mit 0x12XX die der DiRICH2.

Die Messergebnisse beider Boards sind vergleichbar, da die Verstärkung in beiden Versionen vergleichbar ausgeführt wird und die Größe der Schwellenwerte im gleichen Bereich liegt.

In Abb. 42 ist der Schwellenscan für alle Boards mit je allen 32 Kanälen gezeigt. Die Ordinate zeigt den eingestellten Schwellenwert. Die Farbcodierung zeigt wiederum die gemessene Rate. Bei diesen Messungen lag keine Hochspannung an den MAPMTs an, sodass nur das elektrische Rauschen zu sehen ist. Wie man der Abbildung entnehmen kann, ist das gemessene Rauschband bei den DiRICH1 und DiRICH2 Board unterschiedlich. Die Verbesserung am DiRICH2 Board zur Verbesserung des Rauschens zeigen hier ihre Wirkung. Das Rauschband ist im Vergleich zu den DiRICH1 Boards deutlich schmaler geworden.

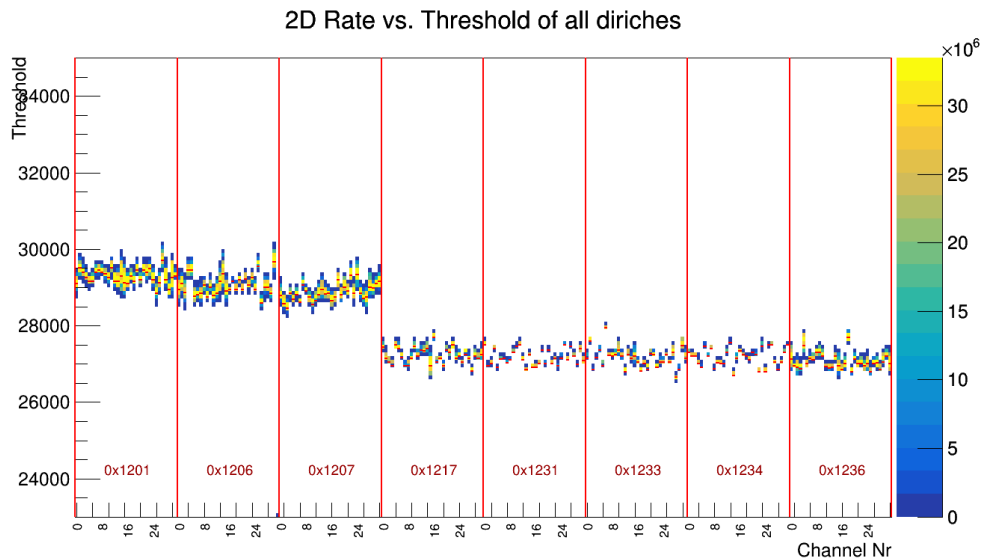


Abbildung 42: Ratenmessung bei variierten Schwellen für alle Kanäle. Linke drei Felder: DiRICH1 (ohne zusätzliche DAC-FPGAs); Rechte fünf Felder: DiRICH2 (mit reduziertem Rauschen und DAC-FPGAs) [23].

Dies kann man in Abb. 43 noch besser erkennen. Hier ist die Breite des Rauschbandes für alle Kanäle der DiRICH Boards in Millivolt gezeigt. Die gemessene Breite des Rauschens ist für das DiRICH2 ca. 10 mV breit und somit deutlich kleiner als die (vorverstärkte) erwartete MAPMT Signal Amplitude, die nach Abb. 7 bei etwa 150 mV (bei 5mV des ursprünglichen Signals) liegt.

Eine Messung mit MAPMTs bei 1000V und montierter Ringmaske führt bei einer Beleuchtung mit einer gedimmten LED zu Abb. 44. Die gezeigte Abbildung wurde ohne Verbesserungen durch Time-Over-Threshold-Cuts und ohne Trigger-Fenster aufgenommen. Selbst ohne diese Cuts ist das durch die Maske erzeugte Muster klar zu erkennen, was die Leistungsfähigkeit der DiRICH2 Boards und die erfolgreiche Implementierung der Schwellenwertsetzung durch die DAC-FPGAs belegt.

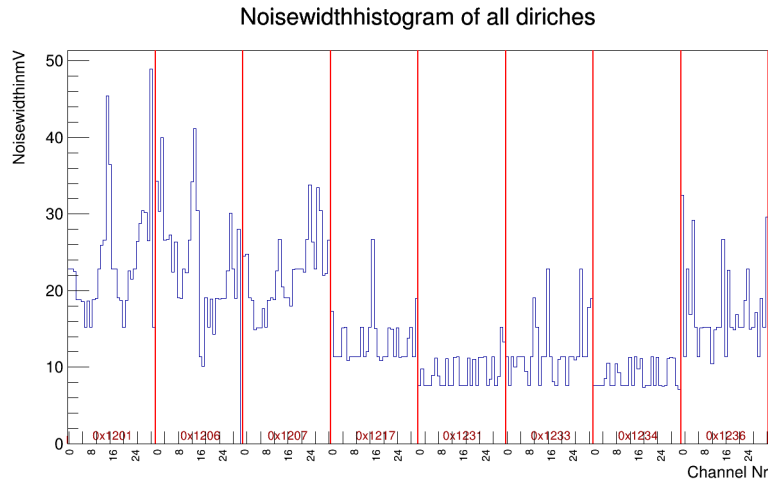


Abbildung 43: Breite des Rauschens in mV für alle DiRICH Boards. Linke drei Felder: DiRICH1; Rechte fünf Felder: DiRICH2 [23].

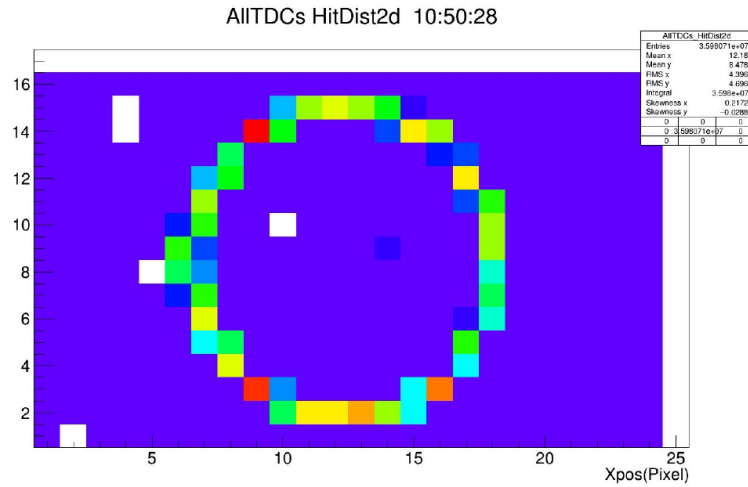


Abbildung 44: Zeitintegrierte Aufnahme eines Rings mit Hilfe einer Maske bei einer vollen Bestückung der Backplane mit 6 MAPMTs und einer Beleuchtung durch eine gedimmte LED [23].

4 Appendix

Im Rahmen des HADES RICH upgrade Projekts wurden verschiedene zusätzliche Kontrollmessungen nötig, die ebenfalls im Rahmen dieser Arbeit durchgeführt wurden und hier kurz erläutert werden sollen.

4.1 Magnetfeldboard

Der HADES RICH ist an seiner Position innerhalb des Experiments einem leichten Magnetfeld ausgesetzt. Um dieses zu überwachen wurde ein Magnetfeldsensor-Board entwickelt. Es wird im Experiment in den inneren Speichen des RICH Detektors montiert (siehe Abb. 45). Jedes Board wird über die Gesamtlänge von 21 cm mit vier Drei-Achsen-Magnetfeldsensoren des Typs MLX90393 bestückt (siehe Abb. 46). Es besteht die Option das Board auf 4 cm zu verkleinern und so mit lediglich einem Magnetfeldsensor zu arbeiten. Die Sensoren werden mit einem ATmega168Pa Mikrocontroller gesteuert. Dieser wird über UART die gemessenen Magnetfeld Daten übertragen. Die Kommunikation zwischen Magnetfeldsensor und Mikrocontroller wird über das I²C Protokoll implementiert.

Über I²C wird jeder einzelne Magnetfeldsensor angesprochen. Während der Initialisierung wird geprüft, welcher Sensor eine Antwort sendet. Kommt keine Antwort, wird angenommen, dass der betroffene Sensor nicht da ist. Es wird im Fall von Magnetfeldsensor 0 (I²C-Adresse: 0x0C) eine Nachricht

ERR Mag 0 NO_DEVICE

ausgegeben. Für die Sensoren 1,2 und 3 würde die gleiche Nachricht mit der jeweiligen Sensornummer ausgegeben.

Tritt hingegen während der Laufzeit ein Fehler auf, so wird die Nachricht

ERR Mag 0 CONNECTION

für den jeweiligen Sensor ausgegeben. Sollte hingegen kein Fehler auftreten, wird ebenfalls eine Nachricht übertragen:

ERR Mag 0 NO

Diese Nachricht wird an die EPICS Slow Control übertragen. So wird es ermöglicht zu erkennen, welcher Sensor funktionstüchtig ist und welcher nicht.

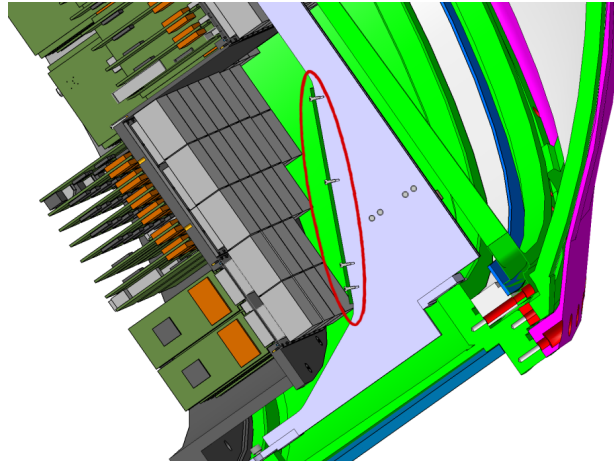


Abbildung 45: Position innerhalb des HADES RICH Detektors nach dem upgrade. Die genaue Position ist rot gekennzeichnet [24].

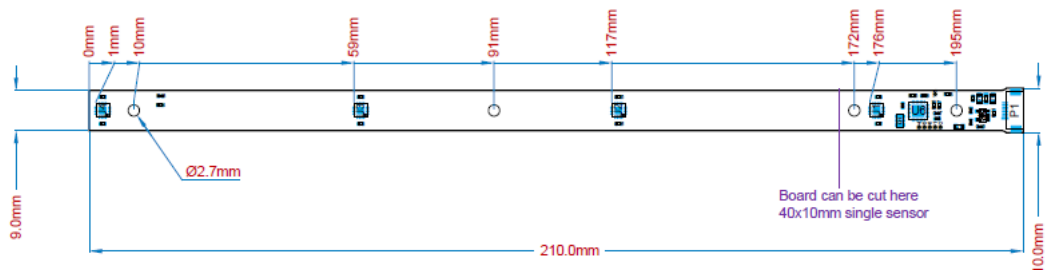


Abbildung 46: Maße des Magnetfeldboards [25].

Funktionieren alle Sensoren, wird die Temperatur sowie die X-,Y- und Z-Achsen Information des Magnetfeldsensors übermittelt. Die Übertragung findet in Mikrotesla statt. Zusätzlich wird die Temperatur in °C gemessen und übertragen. Die übertragenen Informationen werden mit M für das Magnetfeldboard versehen. Anschließend wird die Nummer des Sensors und ein Unterstrich gesendet. Um die Sensorwerte zu unterscheiden, wird nun im Fall der Temperatur ein T und bei den Magnetfeldern die Achse (X,Y oder Z) übertragen. Nachdem diese Identifikation übermittelt wird kommt der Sensorwert, gefolgt von einem Zeilenumbruch.

Die Übertragung erfolgt via UART bei einer Baudrate von 19200 und einer Mikrocontroller Frequenz von 16 MHz. Eine korrekte Übertragung für Sensor 0 sieht z.B. wie folgt aus:

```
M0_T 33.42
M0_X 99.323
M0_Y -23.123
M0_Z 2.947
ERR Mag 0 NO
```

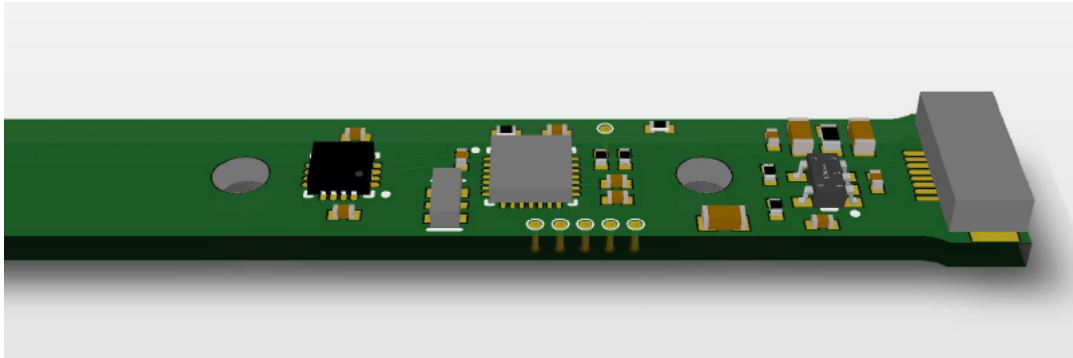


Abbildung 47: 3D Darstellung des Magnetfeldsensorboards. Gezeigt ist lediglich der hintere Teil des Boards [25].

4.2 Umweltsensorboard

Eine weitere Entwicklung im Zuge des HADES RICH upgrade ist ein kleines Umweltsensorboard (siehe Abb. 48 und Abb. 49). Dieses wurde entwickelt, um mit einem Bosch BME280 Sensor Änderungen der Luftfeuchte, der Temperaturen und des Drucks zu messen. Zusätzlich bietet das Board auch einen Luminositätssensor TSL2591 zur Überwachung der Lichtkontamination innerhalb des RICH Detektors. Als 3. Komponente wurde auch ein MLX90393 Magnetfeldsensor auf das Board gesetzt. Die Steuerung der Sensoren wird über I²C von einem ATmega32U4 ausgeführt. Die Daten können hierbei über UART und über microUSB ausgelesen werden. Als zusätzliche Option ist ein Sockel für ein ESP8266 WLAN-Modul vorgesehen (siehe Abb. 50). Dieses Modul bietet die Möglichkeit eines Telnet Client oder auch eines kleinen Servers, so dass die Werte z.B. direkt via Smartphone angezeigt werden können. Desweiteren sind 3 Analogeingänge bis 3 V vorgesehen. Diese können auch als herkömmliche I/O Pins genutzt werden.

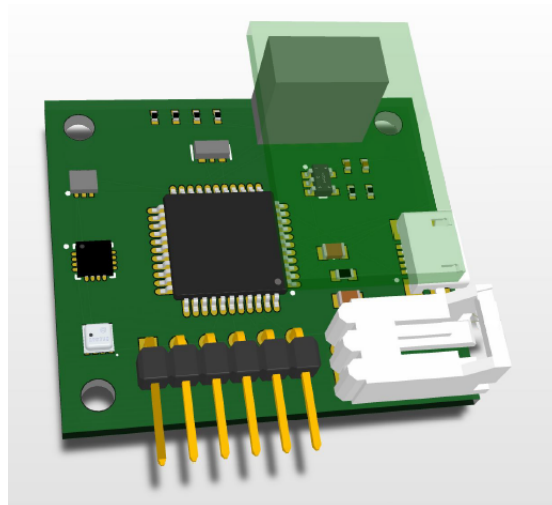


Abbildung 48: Layout des Magnetfeldboards. Im Zentrum: ATmega32U4. Links unten: BME280. Links Mitte: MLX90393. Links oben: TSL2591 [25].

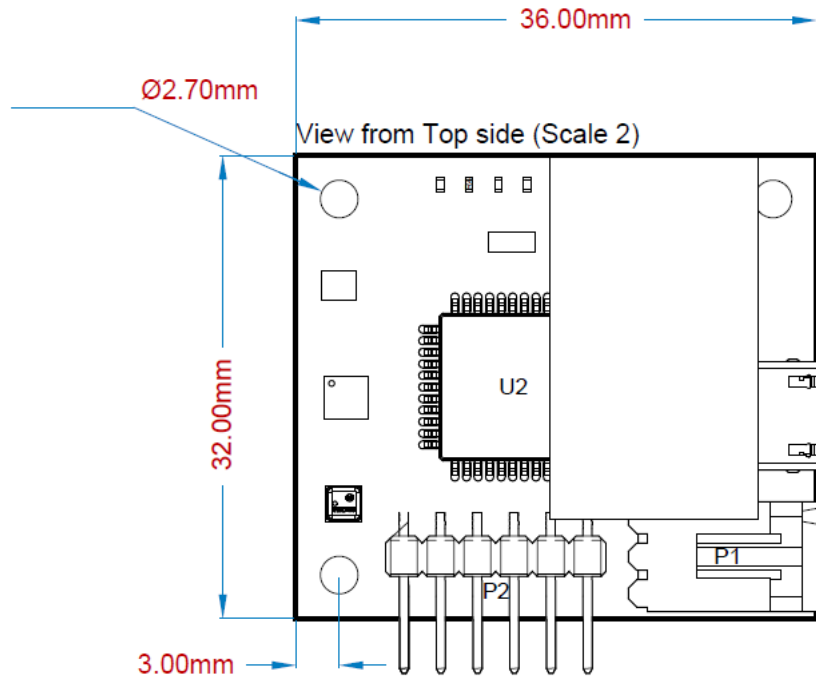


Abbildung 49: Maße des Umweltsensorboards [25].

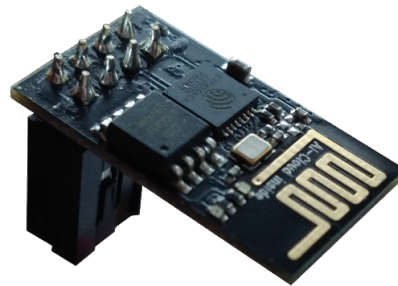


Abbildung 50: ESP8266 Wifi-Modul.

Die Datenausgabe des Umweltsensorboards folgt dem gleichen Prinzip wie das Magnetfeldboard. Für den BME280 Sensor wird die Ausgabe über vorangestellte Wörter **Temp**, **Humid**, und **Pres** getätigt. Parallel wird derzeit ein kommerziell erhältliches Board mit ähnlicher Funktionalität getestet (Silicon Labs SLTB001A). Je nach Ausgang der Testmessungen soll dann entweder das Silicon Labs SLTB001A oder das hier vorgestellte Umweltsensorboard im HADES RICH implementiert werden. In diesem Fall wäre mit wenig Aufwand noch das Protokoll mit dem des Magnetfeldsensorboards zu vereinheitlichen.

4.3 Laboraufbau zur TDC Kalibrierung

Im folgenden werden einige Bilder des Laboraufbaus für die Messungen zur TDC Kalibrierung (Kapitel 2) gezeigt.

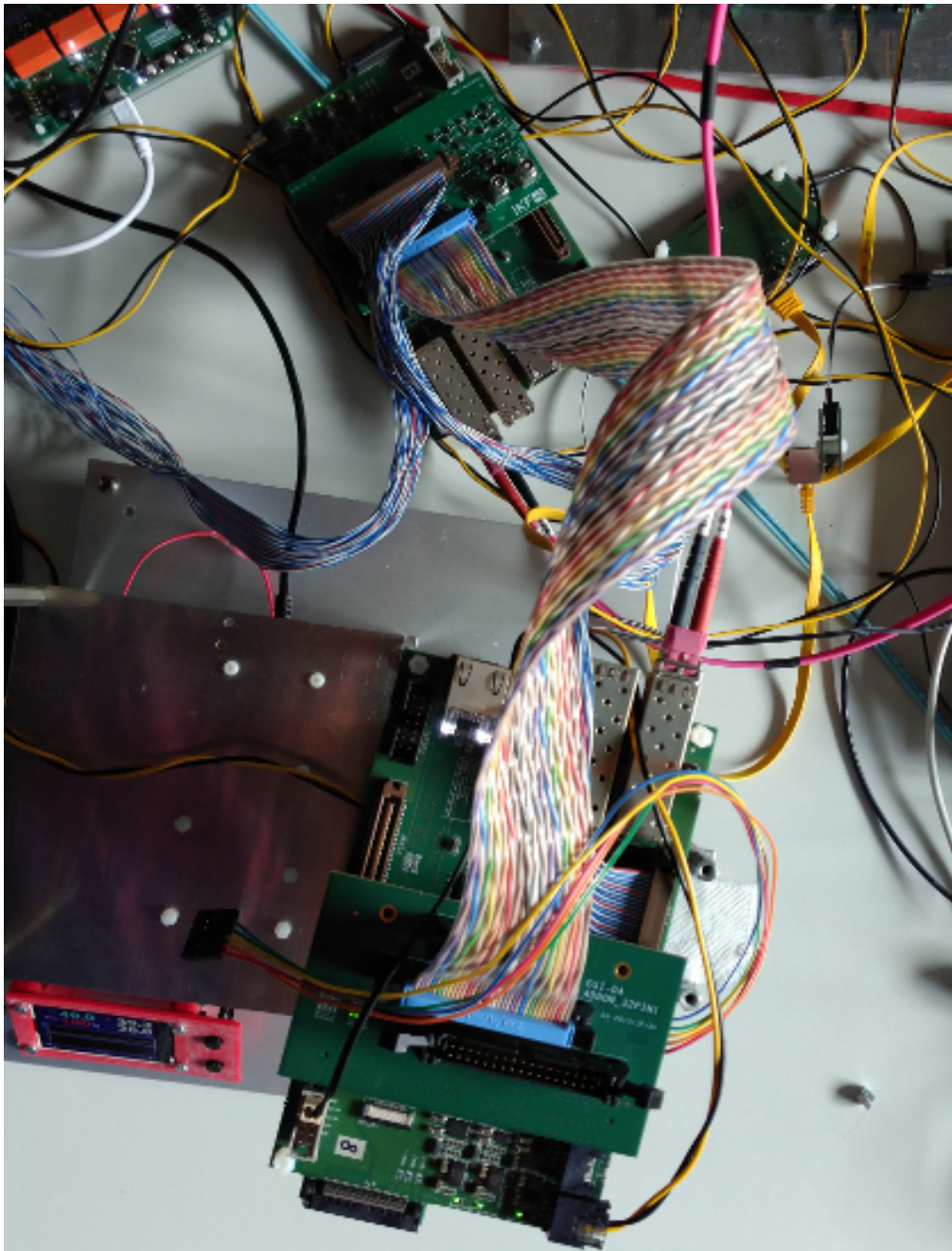


Abbildung 51: Verbindung der beiden TrbSc Boards. Das obere TrbSc dient als Pulsgenerator, das Untere läuft mit dem TDC-Kalibrierungs-Design.

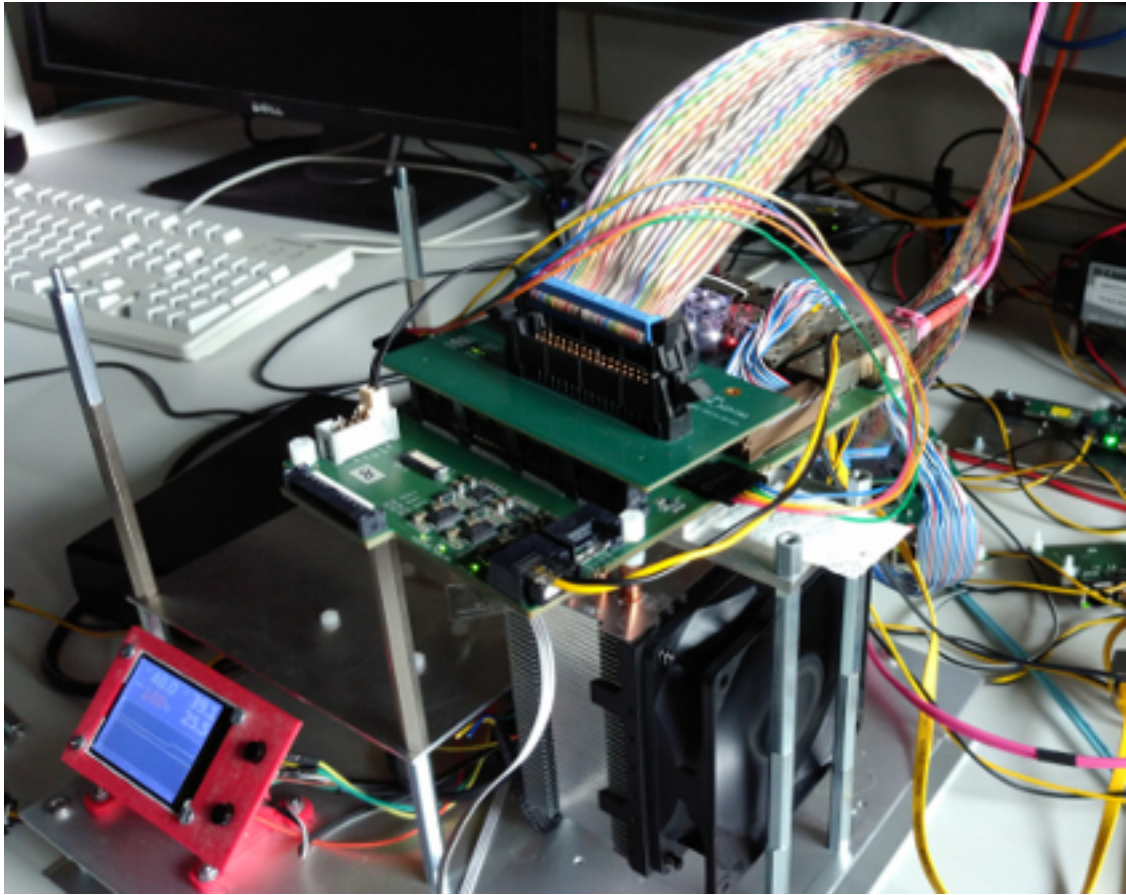


Abbildung 52: Aufbau aus Heizung/Kühlung und TrbSc Board.

Danksagung

Ich bedanke mich bei der kompletten Arbeitsgruppe Höhne für die tolle Zusammenarbeit in den letzten Jahren. Insbesondere möchte ich mich bei Frau Prof. Dr. Claudia Höhne dafür bedanken, dass sie mir die Möglichkeit gibt auf solch einem Interessanten Gebiet der Physik zu arbeiten und mir dabei noch die Möglichkeit eröffnet abwechslungsreichen Tätigkeiten nachzugehen.

Ein sehr großer Dank geht außerdem an Dr. Jan Michel, der mir mit Rat und Tat zur Seite stand und durch die Nutzung seines Labors erst den Zugang zur nötigen Elektronik ermöglicht hat. Desweiteren bedanke ich mich bei Dr. Michael Traxler und der ganzen Trb-Community für die tolle Zusammenarbeit.

Ein weiterer Dank geht an Tariq, Christina, Semen, Jan sowie Jordan für die tolle Stimmung im Büro und der ganzen Arbeitsgruppe.

Ein besonders großes Danke geht an meine Familie und Freunde, die mich immer unterstützt haben und mir den Rücken gestärkt haben.

Abbildungsverzeichnis

1	Schematische Darstellungen des Standardmodells und der Wechselwirkungen.	1
2	Phasendiagramm stark wechselwirkender Materie [4].	3
3	Bildsequenz einer zentralen Au+Au Kollision bei 23 AGeV Strahlenergie	4
4	Messdaten der RHIC Experimente STAR und PHENIX zum elliptischen Fluß.	5
5	Das Hadesspektrometer. [14]	6
6	Schematischer Aufbau des CBM Detektors	9
7	Vergleich des unverstärkten und vorverstärkten MAPMT Signals [17].	11
8	Aufbau der Elektronik/DAQ für das HADES RICH upgrade [18]. . .	12
9	Diagramm eines Slice auf einem Lattice ECP3 FPGAs. Dieser ist bei TrbSc und Trb3 Boards verbaut. Für die DiRICH Boards kommt der Nachfolger ECP5 zum Einsatz. [19]	13
10	Schematische Darstellung eines allgemeinen FPGAs. Eine exakte Darstellung des Lattice ECP3 ist in [19] zu finden.	14
11	Schematische Darstellungen der Coarse- (t_c) und der Fine-Time (t_f) verschiedener Hits [20].	16
12	Schematische Darstellung der Funktionsweise eines TDCs. Zu sehen ist, wie das Signal nach jeder kurzen Verzögerung eine Null im nächsten Volladdierer erzeugt. Zur neuen CoarseTime wird die Position der N Ausgänge in die FlipFlops geschrieben, Binär kodiert und ausgeschrieben.	17
13	Verteilung der Zeitdifferenz einer Zeitmessung des selben Signals in 2 Kanälen [20].	18
14	Der Mittelwert des RMS Werts, ausgerechnet aus allen zur Verfügung stehenden 11 Kanälen für die Kalibrierung basierend auf den Kalibrierungsdaten einer Bin-by-Bin Kalibrierung bei 20 °C (grün), einer linearen Kalibrierung (blau) und der Bin-by-Bin Kalibrierung (rot). Die Werte wurden für alle gemessenen Temperaturen zwischen 5 °C und 50 °C berechnet. Der angegebene Fehler ist die Standardabweichung der Kanäle zum Mittelwert.	21
15	Der Mittelwert des RMS Werts von Kanal 5 für die Kalibrierung basierend auf den Kalibrierungsdaten einer Bin-by-Bin Kalibrierung bei 20 °C (grün), einer linearen Kalibrierung (blau) und der Bin-by-Bin Kalibrierung (rot). Die Werte wurden für alle gemessenen Temperaturen zwischen 5 °C und 50 °C berechnet.	21
16	Der Mittelwert des RMS Werts von Kanal 8 für die Kalibrierung basierend auf den Kalibrierungsdaten einer Bin-by-Bin Kalibrierung bei 20 °C (grün), einer linearen Kalibrierung (blau) und der Bin-by-Bin Kalibrierung (rot). Die Werte wurden für alle gemessenen Temperaturen zwischen 5 °C und 50 °C berechnet.	22

17	Blau: Die von der Software berechnete Kalibrierungskurve der Bin-by-Bin Kalibrierung. Rot: Die von der Software berechnete Kalibrierungskurve der linearen Kalibrierung. Die Kalibrierung wurde an Kanal 11 bei 25 °C durchgeführt.	22
18	Die Differenz zwischen Bin-by-Bin und linearer Kalibrierung. Die Kalibrierung wurde an Kanal 11 bei 25 °C durchgeführt.	23
19	Gemessener RMS Wert für Kanal 11 bei verschiedenen Offsets des Pulsers. Die Offsets lassen sich in 4 ns Schritten einstellen. Gewählt wurden 8 ns Schritte. Die Messung fand unter konstanter Temperatur bei 33,1 °C statt.	23
20	Schematic der TDC Kalibrierung.	29
21	Übersicht über die einzelnen RMS Werte in Pikosekunden für alle Kanäle zu allen gemessenen Temperaturen mit Hilfe der Software basierten Kalibrierung.	31
22	Übersicht über die einzelnen RMS Werte in Pikosekunden für alle Kanäle zu allen gemessenen Temperaturen mit Hilfe der FPGA basierten Kalibrierung.	31
23	Gemessene RMS Werte von Kanal 5 eines TrbSc Boards bei allen betrachteten Temperaturen im Vergleich der zwei Kalibrierungsmethoden.	32
24	Gemessene RMS Werte von Kanal 11 eines TrbSc Boards bei allen betrachteten Temperaturen im Vergleich der zwei Kalibrierungsmethoden.	32
25	Verteilung der Einträge in den FineTime Bins für Kanal 5 der FPGA basierten Lösung mit Kalibrierung.	33
26	Verteilung der Einträge in den FineTime Bins für Kanal 5 ohne Kalibrierung.	34
27	Übersicht über den mittleren RMS Wert aller gemessenen Temperaturen für jeden Kanal des TDC.	34
28	RMS Werte der einzelnen Kanäle bei einem Temperaturabfall zwischen 50 °C und 5 °C. blau: Software basierte Kalibrierung. rot: FPGA basierte Kalibrierung.	35
29	Übersicht über die einzelnen RMS Werte in Pikosekunden für alle Kanäle in Abhängigkeit der Temperatur mit Hilfe der Software basierten Kalibrierung.	36
30	Übersicht über die einzelnen RMS Werte in Pikosekunden für alle Kanäle in Abhängigkeit der Temperatur mit Hilfe der FPGA basierten Kalibrierung.	36
31	Übersicht über die einzelnen RMS Werte in Pikosekunden in Abhängigkeit der Temperatur in Kanal 11.	37
32	Übersicht über die einzelnen RMS Werte in Pikosekunden in Abhängigkeit der Temperatur in Kanal 5.	37
33	Übersicht über die Mittelwerte des RMS über alle gemessenen Temperaturen für alle Kanäle.	38
34	RMS Werte für jeden Kanal bei einem schnellen Temperaturabfall zwischen 50°C und 5°C.	38

35	Verteilung der FineTime Bins in Kanal 5 bei der steigenden Flanke für die FPGA basierte Kalibrierung.	39
36	Verteilung der FineTime Bins in Kanal 5 bei der steigenden Flanke für die Software basierte Kalibrierung.	39
37	Messung der RMS Werte aller Kanäle bei konstanten 34,4 °C. Drei Messungen wurden für die FPGA basierte (rot) und zwei Messungen für die Software basierte (blau) Kalibrierung durchgeführt (mit Offset).	40
38	Links: DiRICH Version 1. Rechts: DiRICH Version 2. Beim zentralen FPGA handelt es sich um einen Lattice ECP5 (blau), bei den DAC FPGAs um Lattice MachXO3-LF-4300 (rot).	42
39	Schematische Übersicht des Aufbaus der 32 Bit Daten der SPI Verbindung.	43
40	Schematischer Aufbau einer Δ - Σ -Modulation [22].	43
41	Oszilloskop Aufnahme der Ausgabe eines PWM Ports eines DAC-FPGAs. Der rot markierte Bereich ist eine Periode der PWM.	44
42	Ratenmessung bei variierten Schwellen für alle Kanäle. Linke drei Felder: DiRICH1 (ohne zusätzliche DAC-FPGAs); Rechte fünf Felder: DiRICH2 (mit reduziertem Rauschen und DAC-FPGAs) [23].	47
43	Breite des Rauschens in mV für alle DiRICH Boards. Linke drei Felder: DiRICH1; Rechte fünf Felder: DiRICH2 [23].	48
44	Zeitintegrierte Aufnahme eines Rings mit Hilfe einer Maske bei einer vollen Bestückung der Backplane mit 6 MAPMTs und einer Beleuchtung durch eine gedimmte LED [23].	48
45	Position innerhalb des HADES RICH Detektors nach dem upgrade. Die genaue Position ist rot gekennzeichnet [24].	51
46	Maße des Magnetfeldboards [25].	51
47	3D Darstellung des Magnetfeldsensorboards. Gezeigt ist lediglich der hintere Teil des Boards [25].	52
48	Layout des Magnetfeldboards. Im Zentrum: ATmega32U4. Links unten: BME280. Links Mitte: MLX90393. Links oben: TSL2591 [25].	52
49	Maße des Umweltsensorboards [25].	53
50	ESP8266 Wifi-Modul.	53
51	Verbindung der beiden TrbSc Boards. Das obere TrbSc dient als Puls-generator, das Untere läuft mit dem TDC-Kalibrierungs-Design.	54
52	Aufbau aus Heizung/Kühlung und TrbSc Board.	55

Tabellenverzeichnis

1	Übersicht über die 3 Wechselwirkungen des Standardmodells.	2
2	Übersicht über die Vergleichskanäle eines jeden Kanals. Diese Zuweisung ist für alle Auswertungen in dieser Arbeit genutzt worden. . . .	20
3	Eingestellter Offset des Pulsers für die jeweiligen Kanäle des TDC. . .	24
4	Aufbau des Zeitdaten-Pakets des TDC.	25
5	Belegung des Adressbereichs zum Auslesen des FPGAs.	45
6	Belegung des Adressbereichs zum Beschreiben des FPGAs.	46

Literatur

- [1] Bogdan Povh. “*Teilchen und Kerne: eine Einführung in die physikalischen Konzepte*“. Springer-Lehrbuch. Springer, Berlin [u.a.], 8. aufl. edition, 2009.
- [2] C. Patrignani et al. “*Review of Particle Physics*“. *Chin. Phys.*, C40(10):100001, 2016.
- [3] Bengt Friman et al. “*The CBM Physics Book: Compressed Baryonic Matter in Laboratory Experiments*“. Lecture Notes in Physics. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.
- [4] Kenji Fukushima and Tetsuo Hatsuda. “*The phase diagram of dense QCD*“, 2011.
- [5] Peter Braun-Munzinger and Johanna Stachel. “*The quest for the quark-gluon plasma*“. *Nature*, 448:302–309, 2007. DOI: 10.1038/nature06080.
- [6] Szabolcs Borsanyi, Gergely Endrodi, Zoltan Fodor, Antal Jakovac, Sandor D. Katz, Stefan Krieg, Claudia Ratti, and Kalman K. Szabo. “*The QCD equation of state with dynamical quarks*“. *JHEP*, 11:077, 2010, 1007.2580.
- [7] Szabolcs Borsanyi, Zoltan Fodor, Christian Hoelbling, Sandor D. Katz, Stefan Krieg, and Kalman K. Szabo. “*Full result for the QCD equation of state with 2+1 flavors*“. *Phys. Lett.*, B730:99–104, 2014, 1309.5258.
- [8] A. Bazavov et al. “*Equation of state in (2+1)-flavor QCD*“. *Phys. Rev.*, D90:094503, 2014, 1407.6387.
- [9] Christian S. Fischer, Jan Luecker, and Christian A. Welzbacher. “*Locating the critical end point of QCD*“. *Nucl. Phys.*, A931:774–779, 2014, 1410.0124.
- [10] J. D’Auria et al. “*Nuclear Physics News*“, Vol 16 No.1 , 2006, ISSN 1050-6896.
- [11] <http://th.physik.uni-frankfurt.de/~weber/CERNmovies/u23u.avi>. Stand: 18/08/2015.
- [12] Roy A. Lacey and Arkadij Taranenko. “*What do elliptic flow measurements tell us about the matter created in the little bang at RHIC?*“. *PoS*, CFRNC2006:021, 2006, nucl-ex/0610029.
- [13] C. Alt et al. “*Pion and kaon production in central Pb + Pb collisions at 20-A and 30-A-GeV: Evidence for the onset of deconfinement*“. *Phys. Rev.*, C77:024903, 2008, 0710.0118.
- [14] https://hades-new.gsi.de/sites/default/files/web/media/page/main/HadesFull3_400pix.jpg. Stand: 31/08/2017.
- [15] Heidi Schuldes. “*Charged Kaon and φ Reconstruction in Au+Au Collisions at 1.23 AGeV*“. <http://publikationen.ub.uni-frankfurt.de/frontdoor/index/index/docId/42489>.

-
- [16] CBM Collaboration. “*CBM Progress Report 2016*“.
 - [17] Vivek Patel. “*First Hands on DiRICH*“, Universität Wuppertal, HADES-Meeting, 21/9/2016.
 - [18] Vivek Patel. “*DiRICH readout electronics - status and first measurements*“, Universität Wuppertal.
 - [19] http://www.latticesemi.com/~media/LatticeSemi/Documents/DataSheets/Lattice/LatticeECP3EAFamilyDataSheet.pdf?document_id=31998.
Stand: 31/08/2017.
 - [20] Cahit Uğur. “*FPGA Based Data Digitisation with Commercial Elements*“. <https://indico.scc.kit.edu/indico/event/21/material/slides/0?contribId=4>.
Stand: 31/08/2017.
 - [21] <http://go4.gsi.de>.
Stand: 31/08/2017.
 - [22] http://www.beis.de/Elektronik/DeltaSigma/DeltaSigma_D.html.
Stand: 31/08/2017.
 - [23] Jörg Förtsch. Universität Wuppertal, *Private communication*.
 - [24] Mike Faul. GSI, *Private communication*.
 - [25] Jan Michel. Universität Frankfurt, *Private communication*.