
EPICS-Steuerung der Spektrometer-Magnete

EPICS-control of the spectrometer magnets

Miniforschung von Alexander Hufnagel

20. November 2013



TECHNISCHE
UNIVERSITÄT
DARMSTADT

Fachbereich Physik
Institut für Kernphysik
AG Pietralla

Inhaltsverzeichnis

1	Einleitung	3
2	Grundlagen	5
2.1	Spektrometermagnete	5
2.2	Bestehendes Steuersystem und Anforderung an das neue System	6
2.3	EPICS	7
2.4	Control System Studio	10
3	Hardware	13
3.1	Netzteil	13
3.2	Umsetzer	14
4	Ergebnis	17
4.1	Finite-State Machine	17
4.2	User Interface	18
4.3	Neues Steuersystem	20
4.4	Bestimmung der Slope-Rate	23
5	Zusammenfassung und Ausblick	25
	Abbildungsverzeichnis	26
	Tabellenverzeichnis	26
	Literaturverzeichnis	27



1 Einleitung

Am supraleitenden Darmstädter Elektronenbeschleuniger S-DALINAC werden zwei Elektronenspektrometer betrieben. Zum einen das Lintott-Spektrometer, auch 169° -Spektrometer genannt, und zum anderen das QCLAM-Spektrometer [1]. Beide Spektrometer besitzen eine ähnliche Funktionsweise. Nachdem die Elektronen an einem Target gestreut wurden, fliegen sie durch den Spektromettermagneten, welcher ein großer Dipolmagnet ist, um letztendlich in einer Fokalebene von verschiedenen Detektorsystemen registriert zu werden. Durch einen konstanten Strom wird im Inneren des Dipols ein homogenes Magnetfeld erzeugt, welches senkrecht zur Bewegungsrichtung der Elektronen steht. Durch die wirkende Lorentzkraft werden die Elektronen auf eine Kreisbahn gelenkt, ohne eine Änderung ihrer Energie zu erfahren.

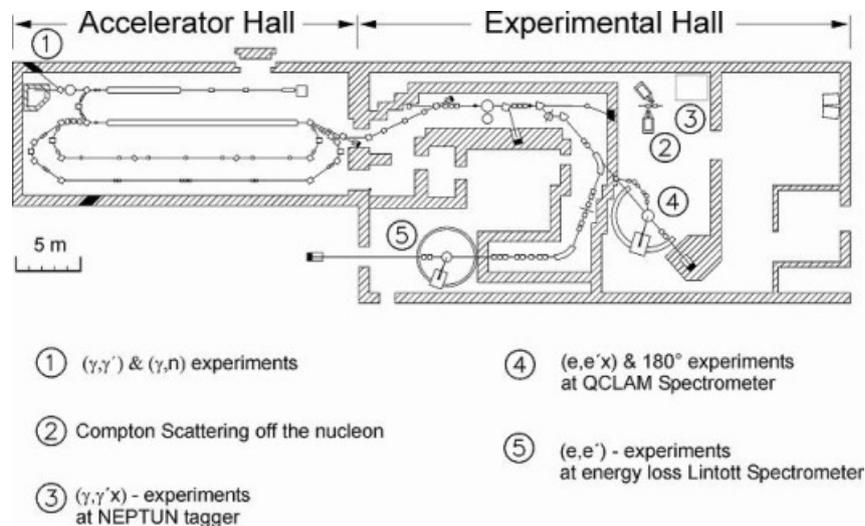


Abbildung 1.1: Das Bild zeigt einen schematischen Aufbau des S-DALINAC. Die Experimentierplätze sind mit Nummern gekennzeichnet [2].

Da das alte Steuersystem über die Jahre an Zuverlässigkeit verlor, ist das Thema dieser Miniforschung die Erstellung einer einfachen und bedienerfreundlichen Steuerung der Spektromettermagnete. Die neue Steuerung wird dabei in das schon bestehende EPICS-System des S-DALINAC eingebunden. Kapitel 2 gewährt einen Überblick über die erforderlichen Grundlagen. Dabei geht es neben den Anforderungen an das Magnetfeld insbesondere um die benötigten Werkzeuge wie EPICS, Stream Device und Control System Studio. In Kapitel 3 werden die verwendeten Geräte wie Netzteil und Umsetzer besprochen. Letztere bietet die Möglichkeit, den Cycle-Vorgang des Magneten in EPICS einzubinden. Die erzielten Ergebnisse werden in Kapitel 4 dargestellt. Ebenso wird hier das User Interface beschrieben und erklärt, sodass diese Miniforschung auch als eine Anleitung dienen kann. Eine Zusammenfassung und Ausblick findet sich in Kapitel 5.



2 Grundlagen

2.1 Spektrometermagnete

Um bei Experimenten am Lintott- bzw. QCLAM-Spektrometer am S-DALINAC [3] gute Ergebnisse zu erhalten, benötigt man in den Spektrometermagneten ein homogenes Magnetfeld. Dieses erreicht man durch einen möglichst konstanten Strom, welcher nur geringen Schwankungen unterliegt. Da Messungen oft auch mehrere Tage in Anspruch nehmen, erhöht dies ebenso die Anforderungen an das versorgende Netzgerät.

Zusätzlich benötigt man meist ein Magnetfeld einer gewissen Stärke. Durch den Einsatz von ferromagnetischen Stoffen bleibt jedoch immer eine Restmagnetisierung, Remanenz genannt, zurück. Um dennoch zu reproduzierbaren Magnetfeldern zu gelangen, verwendet man einen sogenannten Cycle-Vorgang [4]. Dabei stellt man abwechselnd einen Strom ein, welcher größer bzw. kleiner als der Zielstrom ist, um so die Remanenz zu minimieren. Als Zielstrom wird der gewünschte Strom bezeichnet, bei dem das Experiment durchgeführt werden soll (In Abbildung 2.1 als I_0 bezeichnet). Zwischen zwei Cycle-Schritten muss eine Zeit lang gewartet werden, sodass sich die Weißschen Bezirke dem Magnetfeld entsprechend ausrichten können. Man nähert sich so innerhalb einer Anzahl von Schritten an den gewünschten Strom von beiden Seiten an. Die dabei verwendete Cycle-Funktion hat die folgende Form.

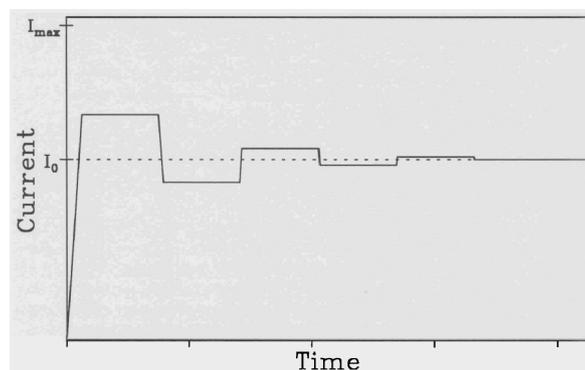


Abbildung 2.1: Die Abbildung zeigt den schematischen Verlauf des eingestellten Stroms während des Cycle-Vorgangs [4].

$$I_i = I_0 \left(1 + \frac{\text{overshoot}}{100} \frac{1}{(-\text{fractionrate})^{i-1}} \right) \quad (2.1)$$

Hierbei läuft die Variable i von 1 bis zur Anzahl der gewünschten Schritte, I_0 entspricht dem gewünschten Zielstrom, overshoot beschreibt die Größe der Über- bzw. Unterschreitung des Zielstroms und fractionrate ist ein Maß für die Annäherungsgeschwindigkeit an den Zielstrom. Die Werte, welche für gewöhnlich verwendet werden, sind in Tabelle 2.1 aufgelistet.

Die Spektrometermagnete des QCLAM- und Lintott-Spektrometers werden vom gleichen Netzgerät betrieben. Zusätzlich zum großen Dipolmagneten muss beim QCLAM ein Quadrupol, welcher im Grunde genommen ein Pentapol ist, mit Strom versorgt werden. Der Wechsel der Stromversorgung zwischen den zwei Experimentierplätzen erfolgt direkt am Netzgerät in der Experimentierhalle durch das Drücken entsprechender Schalter (siehe Abbildung 2.2). Das Netzgerät besitzt zwei Ausgänge (Ports); Port 1 für

Variable	Wert
Schritte	6
overshoot	25
fractionrate	2
Wartezeit	300 s

Tabelle 2.1: Diese Cycle-Variablen haben sich durch Testläufe als besonders empfehlenswert herausgestellt [4].

den jeweiligen Dipol und Port 2 für den Quadrupol des QCLAM. Die Dipole können mit einem maximalen Strom von 310 A und der Quadrupol mit einem maximalen Strom von 225 A betrieben werden [5].



Abbildung 2.2: Das Kontrollpanel des Netzteils. Am unteren Rand des Bildes ist die Umschaltung zwischen Lintott und QCLAM zu sehen. Am Kontrollbildschirm ist zu erkennen, dass das Netzteil im remote-Modus betrieben wird und auf Port 1 177,717 Ampere eingestellt sind.

2.2 Bestehendes Steuersystem und Anforderung an das neue System

Das alte Steuersystem lief auf einer Computereinheit, welches das Betriebssystem Virtual Memory System, kurz VMS, nutzte. Das Programm, das für die Steuerung zuständig war, wurde in C geschrieben. Angesteuert wurde das Programm durch eine Eingabekonsolle. Durch den Mangel einer grafischen Oberfläche oder sonstigen Anzeigen war der Bedienvorgang wenig intuitiv. Ein großer Nachteil des alten Systems war die Inkompatibilität mit moderner, kostengünstiger PC-Hardware, welche schon seit Jahren hier zum Einsatz kommt. Zudem benötigte das Crate unnötig viel Platz, gemessen an der Leistung

und Stabilität, die es lieferte. So passierte es oft, dass die Ansteuerung nicht zuverlässig funktionierte und sich Messungen wegen Ausfällen verzögerten. Zudem war es nötig, die gesamte Hardware neu zu starten, sobald die Verbindung verloren ging, was in letzter Zeit sehr häufig vorkam [6].

Das neue Steuersystem hingegen bietet neben einer grafischen Oberfläche und einer vereinfachten Bedienung auch eine volle Integration in das schon bestehende EPICS-Framework, welches in vielen Bereichen am S-DALINAC verwendet wird. So ist es möglich, die Archivierung von Prozessvariablen (siehe Kapitel 2.3) über bestehende Lösungen zu realisieren. Zudem wird durch das Verwenden von EPICS das Einfügen von Erweiterungen enorm erleichtert. So steht das neue System offen gegenüber möglichen Updates oder Änderungswünschen.

2.3 EPICS

Das **Experimental Physics and Industrial Control System** [7] bietet eine Sammlung von Softwarekomponenten, mit deren Hilfe Kontrollsysteme erstellt und verwaltet werden können. Die Anwendungsgebiete von EPICS erstrecken sich von großen Beschleunigeranlagen, wie dem Deutschen Elektronen Beschleuniger DESY [8] oder dem amerikanischen Fermilab [9], bis hin zu anderen großen Experimenten und Teleskopen. EPICS ist ein Open Source Projekt und steht kostenfrei und mit freier Lizenz zur Verfügung. Durch die weite Verbreitung wird das System in den unterschiedlichsten Gebieten verwendet, weshalb es ständig weiterentwickelt wird. Dabei ist es sehr wichtig, dass das System ohne größere Probleme und Ausfälle auch über lange Zeit konstant arbeitet, da ein Versagen an Beschleunigeranlagen zu großen Schäden führen könnte.

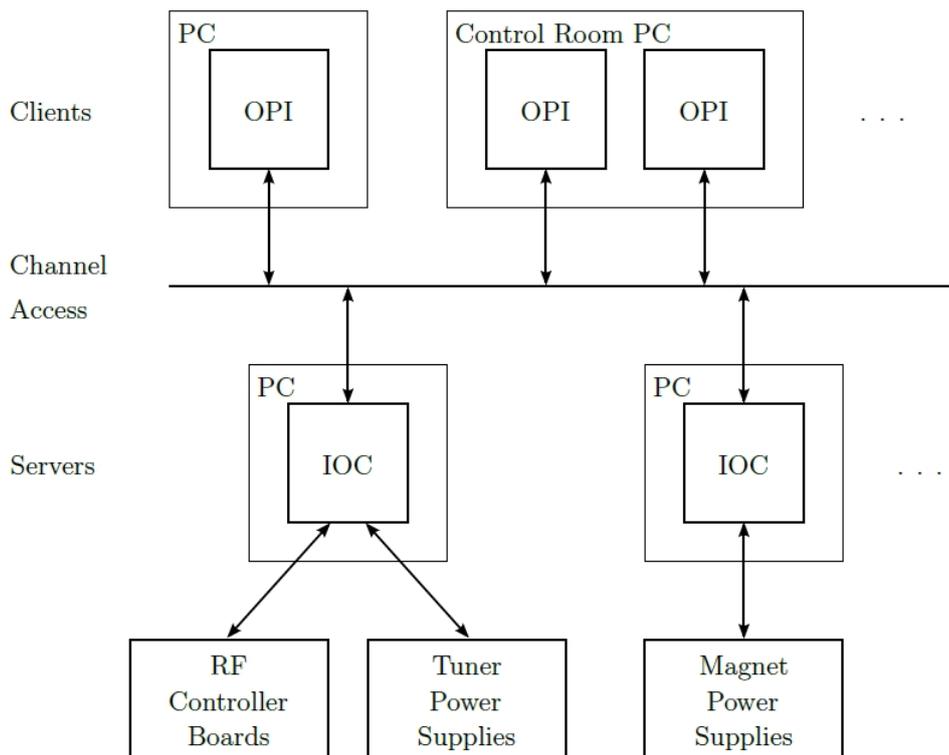


Abbildung 2.3: Das Schaubild beschreibt die EPICS-Architektur. Die Hardware kommuniziert mit den IOCs, welche ihre Informationen über Channel Access den verbundenen OPIs zur Verfügung stellen [10].

EPICS arbeitet mit Client/Server-Methoden. Auf der Server-Seite befinden sich die Input/Output-Controller, welche für die Kommunikation mit der Hardware verantwortlich sind. Diese IOCs arbeiten in

Echtzeit und stellen ihre gesammelten Informationen über ein Netzwerkprotokoll namens Channel Access den verbundenen Clients zur Verfügung. Die von der Hardware gesendeten Informationen werden mittels einer Prozessvariable (PV) für die verbundenen Clients zur Verfügung gestellt und können allein über dessen Namen identifiziert werden. Somit ist es nicht mehr nötig zu wissen, auf welchem Server die PV liegt, um sich mit ihr zu verbinden [7].

Die von den IOCs empfangenen Daten werden als Prozessvariablen gespeichert. Mehrere PVs können in einem record gespeichert werden. Ein record ist immer von einem bestimmten Typ, welcher sich durch dessen Funktion bestimmt. Dabei ist es ebenfalls von Bedeutung, in welche Richtung ein record kommuniziert. Ist dieser vom Typ input, so erwartet der record Informationen von der Hardware. Output-records senden üblicherweise Daten an die Hardware, bzw. kommunizieren mit anderen records. Eine Übersicht über die in dieser Arbeit am häufigsten verwendeten record-Typen liefert Tabelle 2.2.

Name	Datentyp	I/O Richtung
ai	Gleitkommazahl	input
ao	Gleitkommazahl	output
bi	Binärzahl	input
bo	Binärzahl	output
longin	Ganze Zahl	input
longout	Ganze Zahl	output
calc	abhängig vom Input	input

Tabelle 2.2: In dieser Tabelle sind die verschiedenen record-Typen dargestellt. Ein record besitzt einen Namen, einen Datentyp und eine eindeutige Kommunikationsrichtung.

Um einen record zu erstellen, reicht es, eine `.template` Datei zu erstellen und diese mittels Texteditor zu bearbeiten. Die typische Form eines records kann in Abbildung 2.4 betrachtet werden. Neben der Bezeichnung record steht in runden Klammern der Typ und der eindeutige Name. Dieser kann beliebig gewählt werden, jedoch dürfen nicht zwei oder mehrere records mit dem gleichen Namen existieren. Nach Möglichkeit sollte auch ein Name gewählt werden, welcher die Funktion erahnen lässt. Innerhalb der geschweiften Klammern können diverse Felder definiert werden. So sorgt das SCAN-Feld dafür, dass der record jede Sekunde prozessiert wird. Um das in Abschnitt 2.3.1 beschriebene Support Modul Stream Device zu verwenden, muss in den entsprechenden records, welche mit der Hardware kommunizieren sollen, das DTYP-Feld wie in Abbildung 2.4 hinzugefügt werden.

```
record(longin, "$(NAME):DP:rdError") {
  field(DTYP, "stream")
  field(INP, "@BrukerBMC20D.proto rdError(1) $(DEVICE)")
  field(SCAN, "1 second")
  field(SDIS, "$(NAME):deviceEnabled")
}
```

Abbildung 2.4: Ein record, welcher den Befehl rdError ausführt, jede Sekunde prozessiert wird, mit Stream Device arbeitet und nur aktiv ist, wenn das Gerät eingeschaltet ist [11].

Neben dieser Methode existiert noch die Möglichkeit mittels grafischer Oberfläche records zu erstellen. Das Visual Database Configuration Tool, kurz Visual DCT, ist ein kostenloses in Java geschriebenes Programm zur Erleichterung der Arbeit im Umgang mit EPICS. Die Vorteile bestehen u.A. in der stark erhöhten Benutzerfreundlichkeit und der besseren Übersicht [12]. Wie in Abbildung 2.5 zu sehen ist, fällt es wesentlich leichter, Zusammenhänge zwischen verschiedenen records zu überblicken.

Der obere record in Abbildung 2.5 ist vom Typ longout und ist über seinen Namen mit einem Button auf einem OPI verbunden. Wird nun der Knopf gedrückt, wird der record prozessiert. Da jedoch nichts weiteres im record steht, folgt die Prozesskette der weiteren Verknüpfung. Der Forward Link FLINK führt

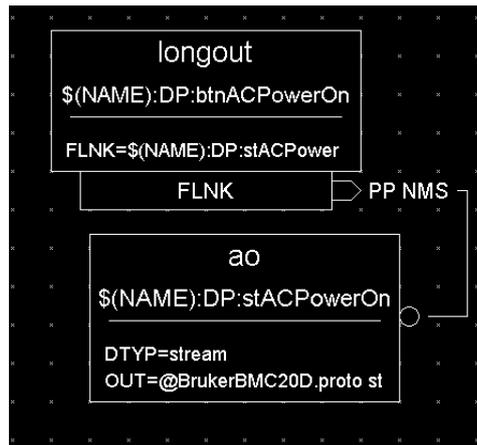


Abbildung 2.5: Eine record-Kette, welche das Einschalten der AC Power am Dipol übernimmt. Wird der obere record prozessiert, so wird durch diese Verknüpfung danach der untere record getriggert.

zum unteren record; durch die Eigenschaft PP, process passive, wird die Prozesskette nicht unterbrochen. Der untere record ist ein analog output, welcher mittels Stream Device kommuniziert. Im OUT-Feld wird auf die für Stream Device benötigte Protokoll-Datei verwiesen. Auf diesem Bild nicht mehr zu erkennen ist der Befehl `stACPowerOn(1) $(DEVICE)`. Eine Erklärung folgt in Abschnitt 2.3.1 [13].

2.3.1 Stream Device

Stream Device ist ein Support-Modul für EPICS, durch welches per Byte-Stream mit der Hardware kommuniziert werden kann, das heißt es ist möglich, über einfache Zeichenketten die Hardware zu steuern. Dies bringt große Vorteile, da hierdurch die Kommunikation erheblich vereinfacht wird, sofern die Hardware diese Art der Steuerung unterstützt. So kann bei dem vorliegenden Netzteil zum Beispiel mit RST/ der momentane Status abgefragt werden. Als Antwort erhalten wir die Zeichenfolge RST/ 0000512 [14]. Diese Art der Kommunikation kommt insbesondere bei seriellen Geräten zu Einsatz, so auch hier. Das Netzteil besitzt eine serielle Schnittstelle, welche via RS-232 kommuniziert [14, 15].

Um Stream Device zu verwenden, muss eine Protokoll-Datei geschrieben werden. In dieser wird festgehalten, welche Befehle gesendet werden und welche Antworten erwartet werden. Ein Beispiel ist in Abbildung 2.6 dargestellt. Dabei befindet sich der zu sendende Befehl hinter dem out-Operator und die zu erwartende Antwort hinter dem in-Operator. %d beschreibt einen Platzhalter, bei dem eine Zahl vom Typ *signed decimal*, das heißt eine vorzeichenbehaftete Dezimalzahl, erwartet wird. Stream Device bietet die gängigsten Zahlformate an, welche auch in der Programmiersprache C, bzw. C++ zu finden sind. Neben den vorhandenen Zahlkonvertierungen besteht auch die Möglichkeit, sich eigene Formate zu erstellen und einzubinden [15].

```
rdError {
    out "RST/\$1";
    in "RST/\$1 %d";
}
```

Abbildung 2.6: Ein Beispiel für einen Eintrag in der Protokoll-Datei. An die Hardware wird der Befehl "RST/1" gesendet und die Antwort "RST/1 0000512" erwartet, falls sich der Dipol im remote-Modus befindet [14].

Die Kommunikation funktioniert nun folgendermaßen: Falls ein record eine Anfrage an die Hardware senden möchte, muss dieser record mit der Protokoll-Datei verknüpft werden. Dies geschieht in diesem

Fall über das Input-Feld INP. Dort muss, wie in Abbildung 2.4 zu sehen ist, auf die Protokoll-Datei, auf den auszuführenden Befehl und auf das anzusprechende Gerät verwiesen werden. Verwendet man ein Output-record, so schreibt man diese Informationen in das OUT-Feld. Sobald nun der record prozessiert wird, wird die hinter dem out-Feld in der Protokoll-Datei stehende Zeichenkette an die Hardware gesendet. Nach der erfolgreichen Antwort wird der erhaltene Wert in das VAL-Feld des records geschrieben und steht dort zur weiteren Verwendung zur Verfügung [13].

Die Zeichenfolge `\$1` in Abbildung 2.6 dient als Platzhalter für eine Zahl. Da am QCLAM ein Dipol und ein Quadrupol gesteuert werden müssen und sich die Befehle bis auf eine Portnummer ähneln, wird hier ein Makro verwendet. Wie in Abbildung 2.4 zu sehen ist, lautet der Befehl `rdError(1)`. Durch das Einfügen von (1) (für Dipol), bzw. (2) (für Quadrupol) hinter dem Namen wird das Makro im Befehl der Protokoll-Datei durch eine 1, bzw. 2 ersetzt. Durch diese Methode kann der Code der Protokoll-Datei enorm reduziert werden. In diesem record sind noch zwei weitere Makros verwendet worden. `$(DEVICE)` verwendet man, wenn das gleiche Template verschiedene Geräte steuern soll und `$(NAME)` verwendet man, wenn man viele records hat, welche die gleiche Funktion besitzen, sich jedoch in ihrer Bezeichnung unterscheiden. Beide Optionen sind hier nur zur Demonstration verwendet worden. Die Unterscheidung zwischen Dipol und Quadrupol wird durch das oben erwähnte Makro durchgeführt.

2.4 Control System Studio

Das Control System Studio (CSS) ist Teil des Kontrollsystems. Es dient der Überwachung und Steuerung der Geräte, wie z.B. Vakuumpumpen oder Netzteile. Es wurde in einer Kollaboration zwischen verschiedenen physikalischen Instituten erstellt, darunter der DESY in Hamburg oder das BNL in New York, USA [16]. CSS steht in verschiedenen Versionen zur Verfügung. Die hier verwendete Version trägt die Bezeichnung NSLSII und verwendet als graphische Oberfläche BOY [11]. Die Bereiche am S-DALINAC, welche über die CSS Oberfläche gesteuert werden, erstrecken sich von einfachen Stromauslesen über Vakuum-Kontroll-Systeme bis hin zur Steuerung der Hochfrequenzcavities [10, 11].

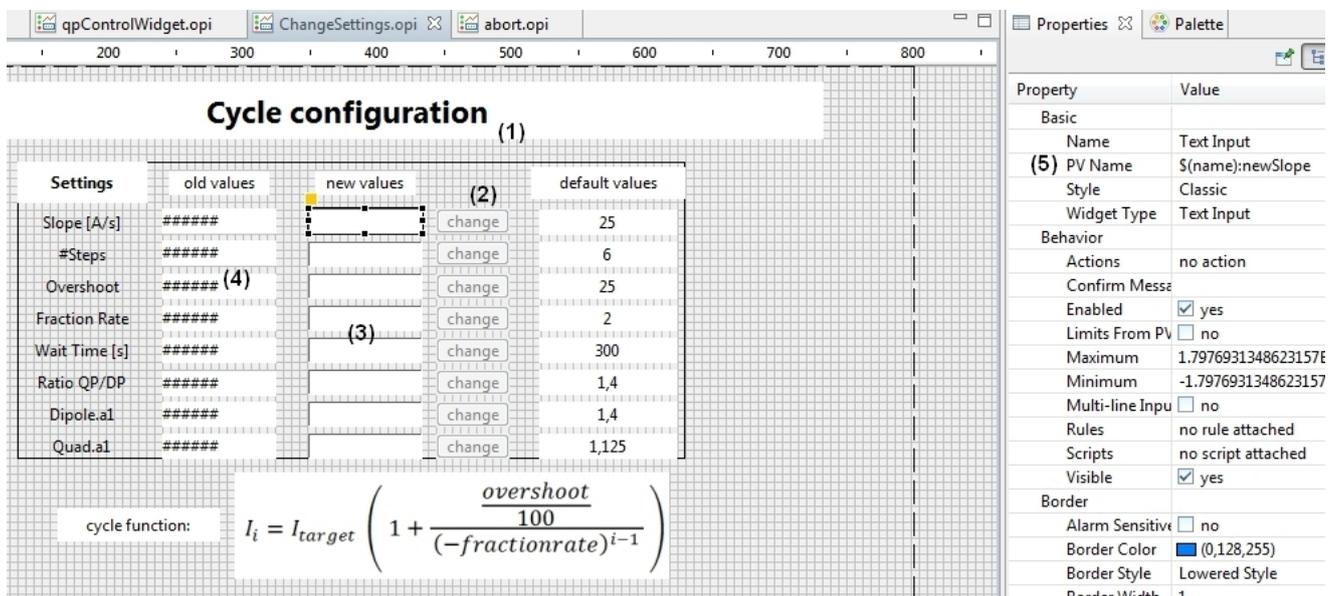


Abbildung 2.7: Ein Screenshot der Arbeitsoberfläche von CSS. Zu sehen sind u.a. erstellte Labels (1), Buttons (2), Textinput-Felder (3) sowie Textupdate-Felder (4). Die Objekte werden direkt mit dem Gerät über den PV Namen (5) verbunden.

In Abbildung 2.7 ist die Arbeitsoberfläche von CSS zu sehen. Um ein Operator Interface zu kreieren, muss man zuvor eine `.opi`-Datei erstellen. Das Programm ermöglicht es Objekte aus einer Palette auf einer Zeichenfläche zu platzieren. Die Anordnung passiert per intuitivem Drag and Drop, man kann

die Position jedoch auch genauer in den Properties (Eigenschaften) eines Objektes festlegen. Neben der Position lassen sich auch noch Aussehen, Größe und Verhalten bestimmen. Durch die Option "Verhalten" können so zum Beispiel Alarmzustände visuell auf der Oberfläche hervorgehoben werden.

Die wichtigste Eigenschaft eines Objektes ist der PV-Name. Über diesen Eintrag können Objekte auf dem OPI mit einem record aus der EPICS-Datenbank verknüpft werden. Sobald eine Verknüpfung hergestellt wurde und die .opi-Datei im Runtime-Modus ausgeführt wird, sendet das System Anfragen an die IOCs und verlangt nach Informationen über den entsprechenden record. Falls ein Objekt keine Verbindung zu seinem record aufbauen kann, so wandert das Objekt in einen Alarmzustand und wird als "disconnected" dargestellt. Auf diese Weise erfährt man, sobald ein Gerät keine Informationen mehr sendet, falls es ausgefallen ist oder falls kein IOC diese PV kennt [11].

Zu den wichtigsten Objekten gehören vor allem das Text-Update und das Text-Input. Möchte man die von der Hardware empfangenen Informationen, wie zum Beispiel Stromwerte, darstellen, so verwendet man ein Text-Update. Sind record und PV über den PV-Namen korrekt verbunden, so wird der Inhalt des VAL-Feldes des records im Text-Update dargestellt. Wird im record zusätzlich das EGU-Feld (Engineering unit) deklariert, so wird der erhaltene Wert auch mit Einheit angezeigt. Um beliebige Stromwerte zu setzen, gibt man die gewünschte Zahl in ein Text-Input Feld ein. Der verbundene record liest, sobald er prozessiert wird, das Input-Feld aus und setzt den Wert in sein VAL-Feld. Um bei diesem Programm einen Strom zu setzen, muss man seine Angabe per Kopfdruck bestätigen. Sobald ein Button gedrückt wird, wird der record prozessiert. Eine ausführliche Beschreibung der Verknüpfung folgt in Kapitel 4.3.



3 Hardware

3.1 Netzteil

Das Steuergerät, welche die beiden Spektromettermagneten des Lintott und QCLAM mit Strom versorgt, kommt von der Firma Bruker und trägt die Bezeichnung B-MC 20D. In Steuergerät sind die Netzteile B-MN 170/310 und B-MN 36/255 den Ports 1 und 2 zugeordnet [14]. Dabei versorgt das erste Netzteil die Dipole mit einem maximalen Strom von 310 Ampere und das zweite Netzteil den Quadrupol des QCLAM mit einem maximalen Strom von 255 Ampere [17]. Das Steuergerät kann sowohl im lokalen als auch im remote-Modus betrieben werden. Wegen der großen Ausgangsleistung müssen die Netzteile wassergekühlt werden.

In Abbildung 2.2 ist die Vorderseite des Netzteils zu sehen. Um jedoch den remote-Modus zu verwenden, bietet das Gerät an der Rückseite eine serielle Schnittstelle, welche per RS-232 und RS-422 kommunizieren kann. In diesem Fall wird das Netzteil per RS-232 mit dem Umsetzer verbunden, welcher wiederum per Ethernet mit dem hauseigenen Netzwerk verbunden ist. Auf diese Weise ist es möglich die Spektromettermagnete beispielsweise per telnet-Verbindung zu steuern.

Mit Hilfe des telnet-Befehls wird eine direkte Verbindung zum Netzteil aufgebaut. Durch Eingabe der entsprechenden Befehle ist eine direkte Steuerung der Magnete möglich. Bei erfolgreicher Kommunikation gibt das Netzteil den gesendeten Befehl als Antwort wieder aus. In Tabelle 3.1 findet man eine Auflistung der wichtigsten Befehle und deren Beschreibung.

Befehl	Beschreibung
PRT/	Liest den aktuellen Port aus
PRT=cc	Setzt den Port auf den Wert cc (1 oder 2)
RST/	Liest den aktuellen Status aus
ACP=cc 0(1)	Schaltet die ACPower in Port cc aus (ein)
DCP=cc 0(1)	Schaltet die DCPower in Port cc aus (ein)
CUR/cc	Liest den aktuellen Strom von Port cc aus
CUR=cc nn	Setzt den Strom von Port cc auf nn
RAT/cc	Liest die Sloperate von Port cc aus
RAT=cc nn	Setzt die Sloperate von Port cc auf nn
UPL=cc nn	Setzt die obere Schranke von Port cc auf nn
LWL=cc nn	Setzt die untere Schranke von Port cc auf nn
STP=3	Startet den Ramp-Vorgang zur oberen Schranke
STP=4	Startet den Ramp-Vorgang zur unteren Schranke
STP=0	Stoppt den Ramp-Vorgang

Tabelle 3.1: Dies sind die wichtigsten Befehle zur Steuerung des Netzteils [14].

Bei der Stromeingabe muss darauf geachtet werden, dass der Wert hinter dem CUR=1 nn in Einheiten von Skalenteilen eingegeben wird. Da im Netzteil ein 16 bit-ADC verarbeitet worden ist, entspricht ein Wert von 65535 dem maximalen Strom. Dies ergibt für den Dipol, mit einem maximalen Strom von 310 A, den Umrechenfaktor 211.4032. Möchte man einen Strom von 100 A einstellen, so lautet der

Befehl CUR=1 21140. Am Quadrupol hingegen beträgt der maximale Strom nur 225 Ampere, weshalb der Umrechnungsfaktor 291.2667 beträgt. Um die zukünftige Eingabe nicht unnötig kompliziert zu machen, werden die Umrechnungen von EPICS im Hintergrund übernommen, sodass im fertigen Programm der gewünschte Strom in Ampere eingegeben werden kann.

Während der Testphasen kam es bei der Kommunikation mit dem Netzteil zu einem Problem. Im fertigen Programm sollen Werte wie Strom, Spannung und Status sekundlich abgefragt werden. Dabei hat es den Anschein, dass das Netzteil korrekte gesendete Befehle manchmal nicht versteht und als Antwort einen Fehler ausgibt. Somit erschienen im Sekundentakt Fehlermeldungen, was sehr untypisch ist, da selbst ältere Geräte mit dieser Anzahl an Anfragen zurechtkommen sollten [11]. Als mögliche Ursache wurden falsche Kommunikationseinstellungen (siehe Kapitel 3.2) bzgl. der baud-Rate oder den bit-Einstellungen in Betracht gezogen. Leider brachten Änderungen in diesen Einstellungen keine Besserung. Wurden diese Einstellungen geändert, war es überhaupt nicht mehr möglich, das Netzteil zu steuern. Die Kontaktaufnahme mit dem Hersteller brachte ebenfalls keine Lösung, da ihm dieses Problem nicht bekannt war [17].

Die technische Ursache wurde mangels Zeit noch nicht entdeckt. Die Flut an Fehlermeldungen konnte jedoch mit Hilfe eines Eintrages in der Protokoll-Datei gestoppt werden. Die Fehlermeldung des IOCs kommt im wesentlichen durch eine gesendete Antwort zustande, welche der IOC nicht versteht. Falls man mit Stream Device arbeitet, muss die zu erwartende Antwort bekannt sein. Wie oben erwähnt, gibt das Netzteil als Antwort den gesendeten Befehl mit einer Zahl aus. Wird nun eine "falsche" Antwort empfangen, so gibt der IOC einen Fehler aus. Für einen solchen "Mismatch"-Fall gibt es für Stream Device eine Lösung.

```
rdCurrent {
    out "CUR/\$1";
    in "CUR/\$1 %d";
    @mismatch {
        in "CUR/\$1 %d";
    }
}
```

Abbildung 3.1: Durch den Mismatch-Eintrag in der Protokoll-Datei werden unverständliche Antworten durch die erwartete Antwort überschrieben.

Mit Hilfe des "Mismatch"-Operators können im Falle einer Nichtübereinstimmung die fehlerproduzierenden Einträge überschrieben werden. Da in unserem Fall die tatsächlich erhaltene Antwort nicht immer der erwarteten Antwort in Zeile 3 in Abbildung 3.1 entspricht, wird die erhaltene Antwort durch die in Zeile 5 ersetzt. Durch diesen Umweg können die Fehlermeldungen beseitigt werden. Praktisch beeinflusst dieses Problem den Betrieb nur wenig, da der Großteil der gesendeten Befehle vom Gerät verstanden wird.

3.2 Umsetzer

Um mit der Hardware über das Netzwerk kommunizieren zu können, wird ein Umsetzer zwischen serieller Schnittstelle und Ethernet benötigt. Hierfür wurde das Modell NPort IA5450A-T der Firma MOXA ausgewählt. Der Umsetzer besitzt vier serielle Anschlüsse und zwei Ethernet-Verbindungen. Im Hinblick auf zukünftige Projekte wurde ein Umsetzer mit vier seriellen Anschlüssen gewählt, sodass die Möglichkeit besteht, weitere Geräte über diesen Umsetzer anzusprechen. Zum Betrieb werden 12-48 Volt Gleichspannung benötigt, welche durch das Gerät STEP-PS/1AC/24DC/2.5 der Firma Phoenix Contact zur Verfügung gestellt werden. Die Verkabelung wurde von der hauseigenen Elektronik-Werkstatt vorgenommen.

Bevor der Umsetzer den Betrieb aufnehmen kann, muss vorher die IP-Adresse geändert werden. Das Gerät wird über die IP 192.168.2.100, Subnetzmaske 255.255.255.0 und Gateway 192.168.2.251



Abbildung 3.2: Der Umsetzer wird von einer 24 Volt Gleichspannungsquelle versorgt. Die vier seriellen Anschlüsse bieten Platz für zukünftige Geräte.

erreichbar sein [18]. Zusätzlich zu diesen Netzwerkeinstellungen müssen weitere Einstellungen vorgenommen werden. Darunter fallen die Kommunikationsrate mit 9600 baud, 8 data bits und 1 stop bit. Mittels RTS/CTS wird der Datenfluss kontrolliert [14]. Diese Werte werden durch die verbundene Hardware vorgeschrieben. Ein Abweichen von diesen Werten kann zu möglichen Fehlfunktionen, bzw. zu einem Totalausfall führen.



4 Ergebnis

4.1 Finite-State Machine

Der in Kapitel 2 beschriebene Cycle-Vorgang ist ein System, für dessen Umsetzung eine Finite-State Machine verwendet werden muss. Die State-Machine ist besonders nützlich, falls ein System mit einer begrenzten, bzw. endlichen Anzahl unterschiedlicher Zustände gesteuert werden soll. Um von einem Zustand in einen anderen zu wechseln, müssen Bedingungen erfüllt werden. Durch das Wechseln in einen anderen Zustand können Operationen ausgeführt werden. So eignet sich die Finite-State-Machine für den Einsatz zur Steuerung des Cycle-Vorgangs der Magnete. Die Umsetzung ohne eine Finite-State Machine wäre sehr schwierig und nicht effizient.

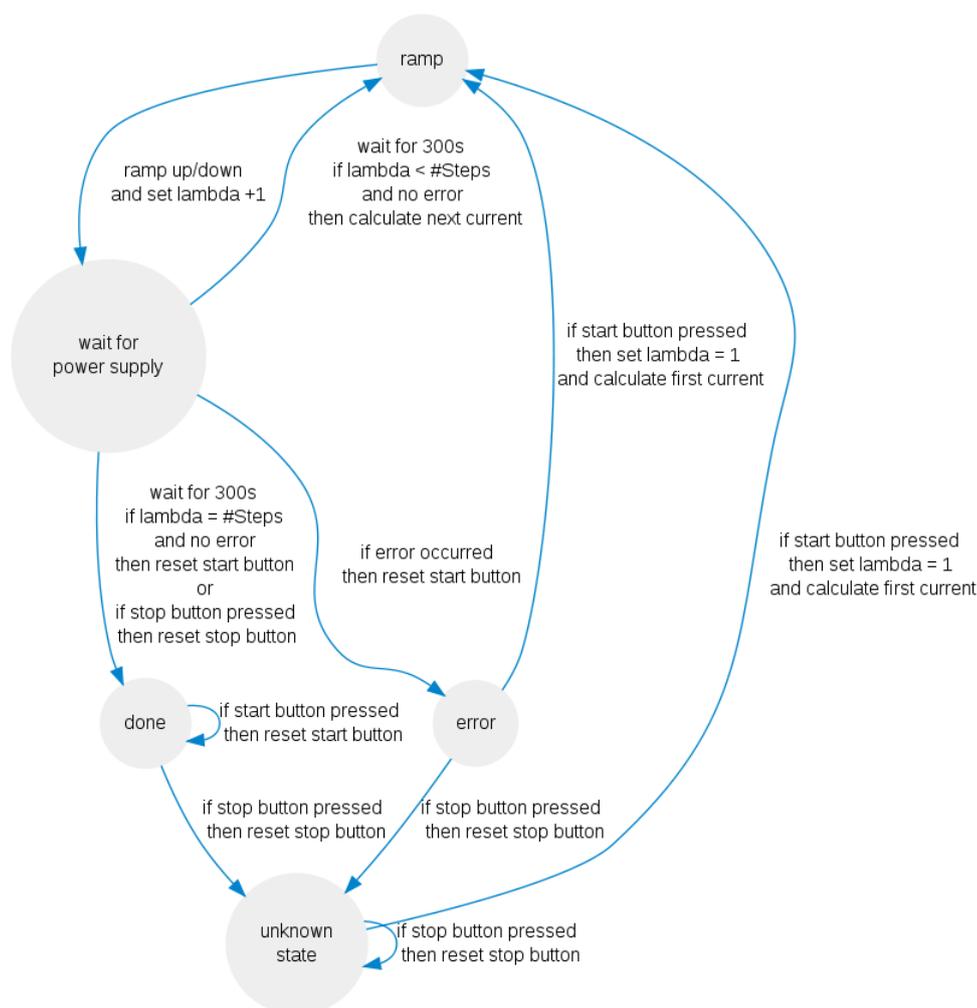


Abbildung 4.1: Die Finite-State Machine für den Cycle-Vorgang der Magnete. Durch die Verwendung der Cycle-Funktion in Gleichung 2.1 nähert sich das Programm mit jedem Schritt dem gewünschten Strom an.

Wie in Abbildung 4.1 zu sehen, kann der Cycle-Vorgang fünf verschiedene Zustände durchlaufen. Die Bedingungen für das Wechseln in einen anderen Zustand stehen an den jeweiligen Pfeilen. Zusätzlich

zu den Bedingungen sind die Aktionen beschrieben, welche bei einem Wechsel ausgeführt werden. Falls sich so zum Beispiel das Programm im Zustand "Warte auf power supply" befindet, kann es nur in den Zustand "Rampe" wechseln, falls 300 Sekunden gewartet wurde, Lambda kleiner als die Anzahl der Cycle-Schritte ist und es zu keinem Fehler kam. Sind diese Bedingungen erfüllt, so kommt es zu einem Wechsel und der nächste Stromwert des Cycle-Vorgangs wird berechnet. Nach erfolgreichem Durchlaufen der Cycle-Schritte, welche vom Experimentator frei wählbar sind, stoppt das Programm automatisch und befindet sich dann im Zustand "Fertig". Tritt während des Cycle-Vorgangs ein Fehler auf oder wird manuell abgebrochen, so wandert das Programm in den Zustand "Fehler", bzw. in den Zustand "Fertig" und von dort aus in "Unbekannter Zustand".

4.2 User Interface

Das neue Interface wurde, wie in Kapitel 2.4 beschrieben, mit dem Control System Studio erstellt. Das Design und die Anordnung der Elemente wurde so gewählt, dass eine möglichst hohe Übersicht und Einsteigerfreundlichkeit gewährleistet werden kann. Das in Abbildung 4.2 gezeigte Interface zeigt die Steueroberfläche für das QCLAM. In der oberen Bildhälfte ist die Steuerung für den Dipol zu sehen und in der unteren Bildhälfte die Steuerung für den Quadrupol. Das User Interface für den Lintott-Dipol ist das gleiche wie beim QCLAM-Dipol. Da das Lintott-Spektrometer keinen Quadrupol besitzt, wird die untere Hälfte bei der Lintott-Oberfläche nicht angezeigt.

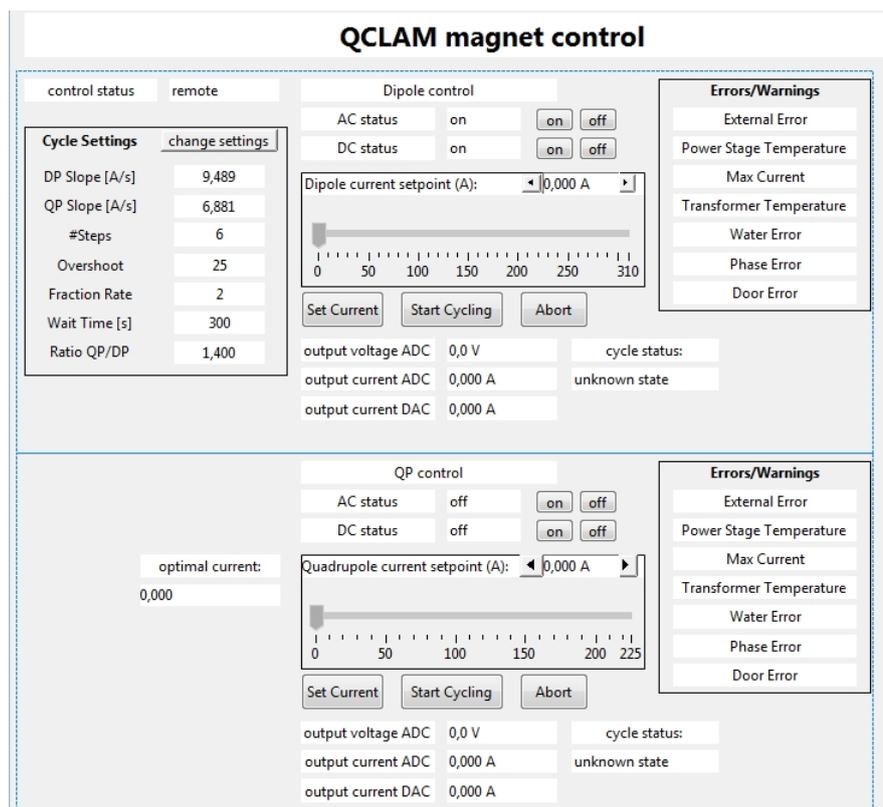


Abbildung 4.2: Das Operator Interface für den QCLAM-Spektromettermagneten. Zu sehen sind die Cycle-Einstellungen links, Fehler und Warnungen rechts und die Steuerung für den Strom in der Mitte.

Bevor man einen Strom setzen möchte, sollte man zu Beginn einen Blick auf den "control status" werfen. Dieser zeigt an, ob das Netzteil im remote oder local Modus betrieben wird. Befindet sich das Netzteil im local Modus, funktioniert die Steuerung über das Interface nicht. Aus Sicherheitsgründen ist es nur möglich, lokal am Netzteil zwischen den Modi zu wechseln. Im Normalfall sollte aber remote

eingestellt sein. Zusätzlich müssen für den Betrieb "AC Power" und "DC Power" auf "on" stehen. "AC Power" wird für geräteinterne Aufgaben benötigt und "DC Power" sorgt für den tatsächlichen Stromfluss zum Magneten. Beide Elemente können über die anliegenden Buttons gesteuert werden.



Abbildung 4.3: In den Cycle-Einstellungen lassen sich alle Werte separat ändern.

Zentral in der Mitte findet man die eigentliche Stromsteuerung. Möchte man einen Strom setzen, hat man die Möglichkeit, den Strom per Schieberegler, per Scrollbar oder per direkter Eingabe einzustellen. Erst durch das Drücken von "Set Current" wird der Strom gesetzt. Hinter dieser Aktion befindet sich kein Ramp-, bzw. Cycle-Vorgang. Die Option eignet sich besonders gut für Testzwecke, bei denen kein fest definiertes Magnetfeld benötigt wird, oder für das Herunterfahren des Magnetfeldes.

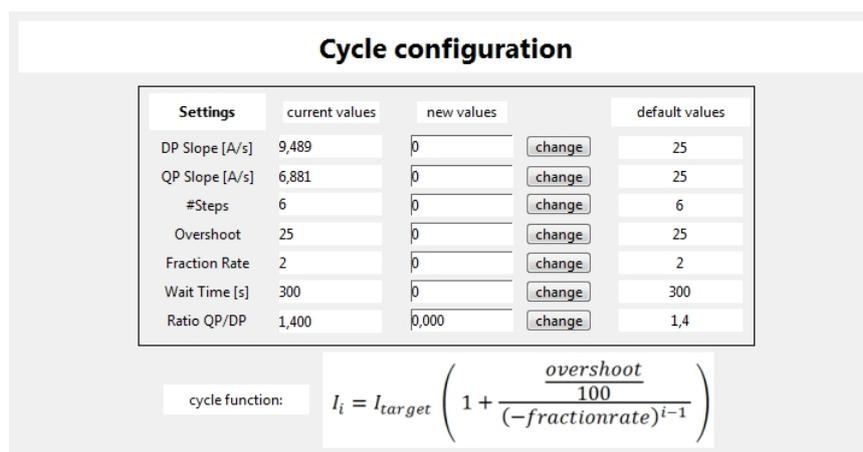


Abbildung 4.4: Durch das Einfügen eines extra Abbruch-Fensters werden ungewollte Abbrüche vermieden

Um ein Magnetfeld für Experimente aufzusetzen und somit den Cycle-Vorgang zu nutzen, muss lediglich der "Start cycling"-Button verwendet werden. Dabei ist zu berücksichtigen, dass der Cycle-Vorgang, falls nicht anders eingestellt, 30 Minuten in Anspruch nimmt. Die ursprünglichen Einstellungen sind dabei so gewählt, dass nach jedem der 6 Cycle-Schritte 300 Sekunden gewartet wird. Beide Einstellungen sind frei wählbar, jedoch haben sich diese Einstellungen als besonders empfehlenswert herausgestellt [4]. Um einen Cycle-Vorgang abubrechen, kommt man über den Knopf "Abort" in ein neues Fenster, in welchem man das Abbrechen erneut bestätigen muss (siehe Abbildung 4.3). Dies dient der Sicherheit, sodass durch einen unbeabsichtigten "Fehlklick" der Cycle-Vorgang nicht abgebrochen werden kann.

Falls man dennoch die Cycle-Einstellungen ändern möchte, so öffnet sich hierzu ein separates Fenster nach einem Klick auf "change settings". In dem neuen Fenster (siehe Abbildung 4.4) sind alle änderbaren Einstellungen und deren ursprüngliche Werte einzusehen. Um eine Änderung vorzunehmen, gibt man den gewünschten Wert in die mittlere Spalte unter "new values" ein und bestätigt seine Eingabe mit dem Button "change". Ebenfalls zu sehen ist die Cycle-Funktion, sodass auch Personen, welche nicht mit dem Thema vertraut sind, einen leichteren Einstieg mit dem Umgang des Magneten erhalten.

Eine Übersicht über die möglichen Hardware-Fehler, welche auftreten könnten, befindet sich, wie in Abbildung 4.2 zu sehen, auf der rechten Seite des Kontrollinterfaces. Die Funktionsweise wird in Kapitel 4.3 erläutert.

4.3 Neues Steuersystem

Die neue EPICS-Steuering der Spektromettermagnete umfasst einen einfachen Stromsetzvorgang, den Cycle-Vorgang inkl. der Einstellung der Cycle-Parameter sowie die Fehlerauslese und -Darstellung auf dem User Interface. In diesem Kapitel werden die Funktionsweisen der wichtigsten records und deren Verknüpfungen beschrieben. Das in den Abbildungen zu sehende $\$(NAME)$ wird mit Hilfe eines Makros durch die Bezeichnung QCLTSPKT ersetzt. Wie schon in Kapitel 2.3 beschrieben, betrachten wir die records mit Visual DCT, um eine bessere Übersicht über deren Zusammenspiel zu erhalten. Zu sehen sind Typ, Name, Attribute und die Verknüpfungsknoten zwischen den records. Die anliegenden Abkürzungen PP, NPP, MS und NMS stehen dabei für process passive bzw. no process passive, maximize severity und no maximize severity. PP und NPP entscheiden darüber, ob der verknüpfte record getriggert wird oder nicht. MS und NMS entscheiden darüber, ob ein möglicher Alarmzustand berücksichtigt wird oder nicht.

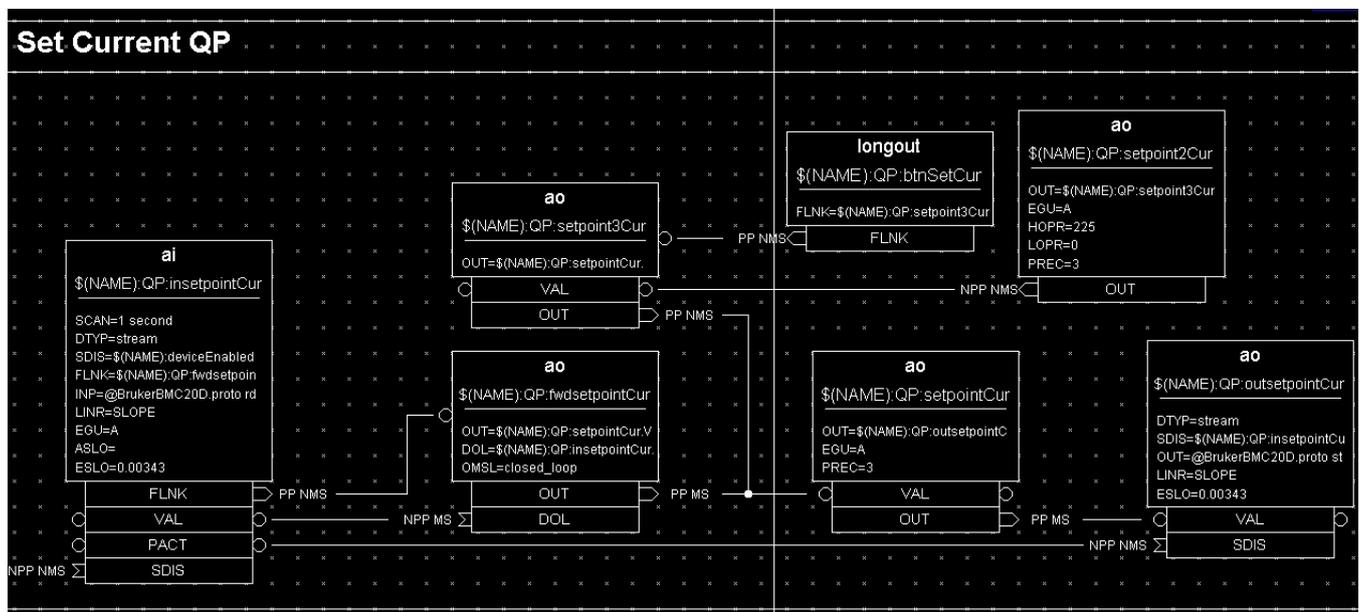


Abbildung 4.5: Durch die Verknüpfung dieser 6 records wird die Stromauslese und -Eingabe inkl. Button-Steuerung am Quadrupol realisiert. Der linke record übernimmt die Stromauslese und der rechte record sendet den Befehl zum Setzen des Stroms.

In Abbildung 4.5 ist die Steuerung der Stromauslese und Stromeingabe zu sehen. Die vier records in der unteren Ebene reichen im Grunde genommen für diese Aufgaben schon aus. Um den Strom durch einen aktiven Knopfdruck steuern zu können, mussten die oberen drei records zusätzlich hinzugefügt werden. Der record auf der linken Seite liest jede Sekunde den Strom des Netzteils aus. Da dieser in Skalenteilen ausgegeben wird, wird mit Hilfe einer linearen Umrechnung der Wert in Ampere umgewandelt. Da es sich bei diesem record um einen Analog Input handelt, muss der Stromwert mit Hilfe eines zweiten records an den dritten weitergeleitet werden. Der record mit dem Namen $\$(NAME):QP:setpointCur$ ist mit dem OPI verbunden, sodass der erhaltene Wert über diese Verknüpfung auf der Oberfläche dargestellt wird (siehe Abbildung 4.2 bei output current DAC). Der erhaltene Wert wird zudem weiter an den vierten record geleitet, welcher eigentlich für das Setzen eines Stromwertes verantwortlich ist. Damit es zu keinem Kreisschluss kommt, sorgt das PACT-Feld (processing active) dafür, dass der vierte record nur prozessieren kann, sofern der erste record gerade nicht aktiv ist. Da jedoch alle records per PP miteinan-

der verbunden sind, gelten alle als aktiv, bis die Prozesskette durchgelaufen ist. Der gewünschte, auf der Oberfläche eingegebene, Stromwert wird durch den oberen rechten record in das System geholt. Durch Drücken des Buttons "Set Current" (siehe Abbildung 4.2) wird ein record zur Weiterleitung getriggert, welcher den Wert an den dritten record der vorherigen Kette weitergibt. Von da an passiert das gleiche wie vorher, nur mit dem Unterschied, dass der erste record der unteren Reihe nicht aktiv ist und der Strom gesetzt werden kann.

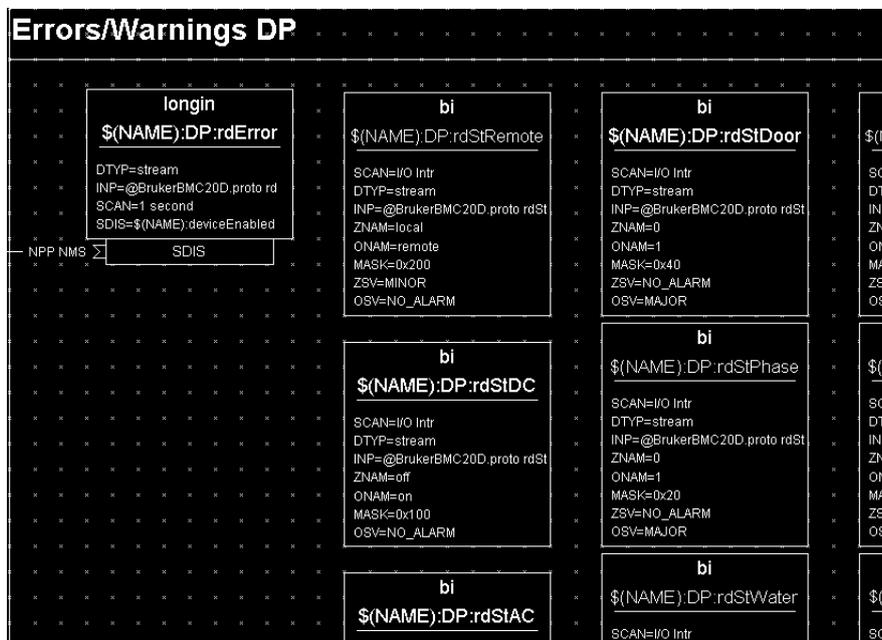


Abbildung 4.6: Der record auf der linken Seite übernimmt die sekundliche Statusabfrage. Durch eine hexadezimale Maske wird die erhaltene Zahl von den einzelnen records entschlüsselt. In den records sind zusätzlich die Zustände und ihre Dringlichkeit definiert.

Der Status des Netzteils kann über den Befehl RST/ abgefragt werden. Diese Aufgabe erledigt der record in der linken oberen Ecke aus Abbildung 4.6 sekundlich. Abbildung 4.6 zeigt nur einen Ausschnitt aus allen Status-records. Tabelle 4.1 zeigt alle möglichen Fehlermeldungen mit ihren zugehörigen Werten. Sind mehrere Zustände aktiv, so gibt das Netzteil die Summe der Werte aus.

Zustand	Wert	hexadezimaler Wert
External error	1	001
Power stage temperature error	2	002
Current error	4	004
Transformer temperature error	8	008
Water error	16	010
Phase error	32	020
Door error	64	040
AC on	128	080
DC on	265	100
Remote active	512	200

Tabelle 4.1: In dieser Tabelle sind die verschiedenen Fehler bzw. Zustände dargestellt. Jedem Zustand ist dabei eine Zahl zugeordnet. Sind mehrere Zustände aktiv, so wird die Summe ausgegeben.

Befindet sich das Netzteil zum Beispiel im remote-Modus und sind ACPower und DCPower an, so wie es im Normalfall sein sollte, gibt das Netzteil den Wert 896 aus. Mit Hilfe einer hexadezimalen Maske wird jedem Zustand seine entsprechende Zahl zugewiesen. Wird so zum Beispiel die Sicherheitstür des Netzteils geöffnet, gibt das Netzteil eine 64 aus, was im hexadezimalen Zahlensystem als 40 dargestellt wird. Dadurch wandert der \$(NAME):DP:rdStDoor-record in einen 1-Zustand und seine Severity (OSV) wechselt zu MAJOR. Die hohe Dringlichkeit wird danach vom OPI interpretiert und das Label "Door Error" färbt sich rot (siehe Abbildung 4.2).

In Abbildung 4.7 ist ein Ausschnitt aus der Steuerung der Cycle-Einstellungen zu sehen. Die nicht zu sehenden records sind mit dem gleichen Muster aufgebaut. Allein die Einstellungen für die Slope unterscheiden sich vom Rest, da hier explizit ein Wert dem Netzteil übergeben werden muss. Alle anderen Einstellungen beziehen sich allein auf den Cycle-Vorgang. Um die Slope-Rate auszulesen, wurde erneut ein Analog Input-record verwendet, welcher alle 10 Sekunden eine Anfrage sendet. Über ein Text-Update-Feld wird der Wert auf dem OPI dargestellt. Um die Slope-Rate zu verändern, gibt man den gewünschten Wert in ein Text-Input-Feld ein (siehe Abbildung 4.4) und bestätigt seine Eingabe per Knopfdruck auf den Button "change". Der Button triggert einen record, welcher den Wert an einen dritten record sendet, welcher diesen Wert dann an die Hardware mit dem entsprechenden Befehl sendet.

Für die programminternen Einstellungen funktioniert die Steuerung analog. Der Unterschied besteht darin, dass hierbei die Werte nicht regelmäßig ausgelesen werden müssen und zudem ein Standard-Wert voreingestellt ist. Möchte man eine Einstellung dennoch ändern, so gibt man sie ebenfalls in ein Text-Update-Feld ein und bestätigt durch einen Knopfdruck. Der Wert wird dann an den record, welcher mit der State Machine verbunden ist, weitergeleitet. Das Feld FLINK (forward link) dient der Verknüpfung zweier records. Durch das PP-Attribut wird der verbundene record im Anschluss an den ersten record prozessiert.

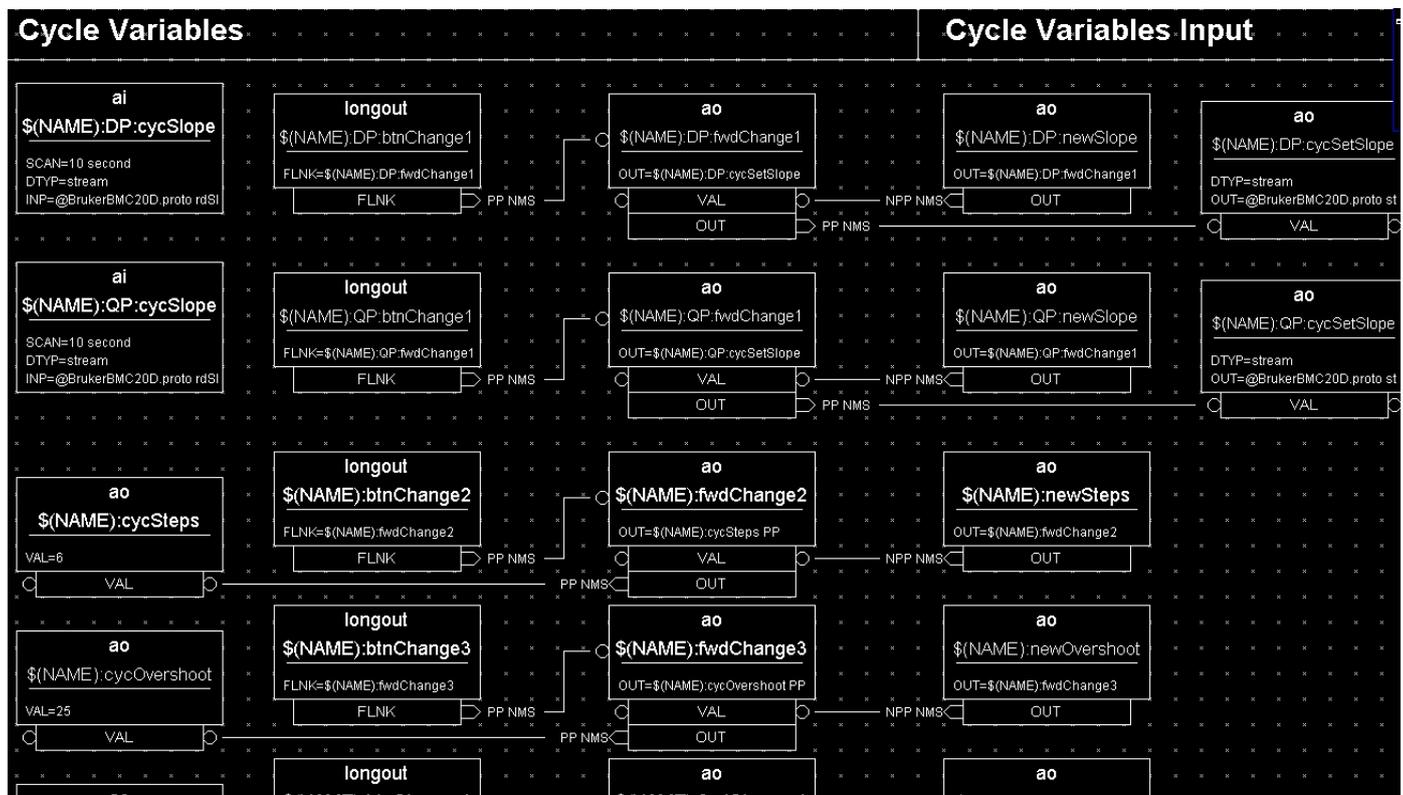


Abbildung 4.7: Die record-Ketten, welche für die Cycle-Einstellungen zuständig sind, funktionieren alle gleich. Durch das Drücken eines Buttons wird der Wert aus dem Eingabefeld der Oberfläche in den entsprechenden record geschrieben.

4.4 Bestimmung der Slope-Rate

Da während den Testphasen auffiel, dass die Slope-Rate nicht mit den Angaben aus dem Handbuch übereinstimmten, musste die tatsächliche Ramp-Geschwindigkeit experimentell bestimmt werden. Aus dem Handbuch geht hervor, dass die Slope-Rate über die Befehlskette `RAT=cc nn` bestimmen wird. Hierbei steht "cc" für einen Kanal, bzw. Port und "nn" beschreibt die Ramp-Geschwindigkeit in Einheiten von A/s. Nach kurzen Testläufen stellte sich heraus, dass zwischen der gedachten und der tatsächlichen Geschwindigkeit ein Faktor von 1381,22 liegt. Dabei wurde für 20 Sekunden in einem bestimmten Intervall bei einer festgelegten Geschwindigkeit gerampt. Das Intervall, die Geschwindigkeit und die Dauer wurden dabei nach jeder Messreihe variiert, sodass eine Abhängigkeit dieser Variablen ausgeschlossen werden konnte. Das Ergebnis war, dass das Netzteil nach der Eingabe von `RAT=1 25` mit einer Geschwindigkeit von 0,0181 A/s statt 25 A/s rampt. Um dennoch die gewünschte Ramp-Geschwindigkeit in A/s eingeben zu können, wurde innerhalb von EPICS mit Hilfe eines `calcout-records` eine Umrechnung hinzugefügt. Zusätzlich wurde bei einem Testlauf explizit auf die Linearität des Ramp-Vorgangs geachtet. Hierzu wurde von 0 bis 200 A mit einer Geschwindigkeit von 0,527 A/s gerampt. Da hierbei alle 5 Sekunden der Wert des Stroms abgefragt wurde, muss die Zeitskala ebenfalls mit dem Faktor 5 verrechnet werden (siehe Abbildung 4.8). Dabei wurde der berechnete Skalierungsfaktor von 1381,22 erneut bestätigt.

Durch den großen Skalierungsfaktor müssen dementsprechend große Slope-Rate-Werte dem Netzteil übergeben werden. Leider kam es in diesem Bereich zu einer Diskrepanz zwischen den eingegebenen und den empfangenen Werten. Wurde eine Slope-Rate von 34525 eingestellt, was einer tatsächlichen Geschwindigkeit von 25 A/s entspricht, so gab das Netzteil bei einer erneuten Abfrage mit `RAT/1` den Wert 52428 aus. Diese Zahl ist um den Faktor 1,51 größer als die eingegebene Zahl. Wählt man eine Slope-Rate aus einem Bereich von 5000 bis 20000 Skalenteilen aus, so wird der Wert mit dem Faktor 1,31 multipliziert. Je näher man sich dabei der 0 nähert, desto geringer wird die Diskrepanz. Da die Ramp-Geschwindigkeit keinen erheblichen Einfluss auf die Güte des Magnetfeldes hat, wurde darauf verzichtet die Ursache zu suchen.

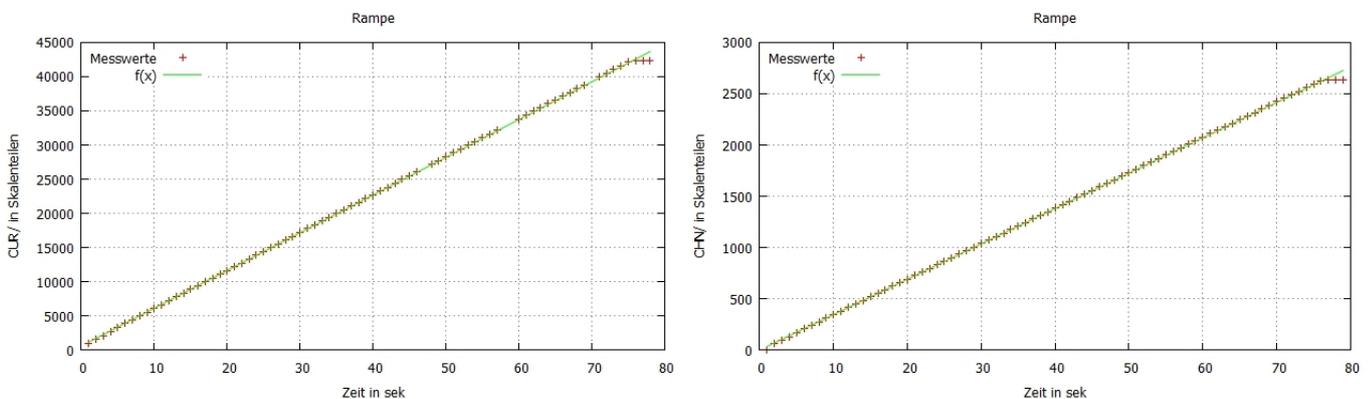


Abbildung 4.8: Die Abbildung zeigt einen Testlauf, bei dem von 0 bis 200 A mit einer Slope-Rate von 0,527 A/s gerampt wurde. In der linken Hälfte wurde der Strom mit dem Befehl `CUR/` angefragt und in der rechten Hälfte mit dem Befehl `CHN/`



5 Zusammenfassung und Ausblick

Die in dem Rahmen dieser Miniforschung erstellten Kontrolloberflächen erleichtern den Betrieb der Spektrometernagnete enorm. Durch die Verwendung von EPICS wurde zudem sichergestellt, dass die neue Steuerung auch in Zukunft offen für Änderungen bzw. Updates ist. EPICS zeichnet sich zudem durch eine hohe Wartungsfreundlichkeit aus, was es auch fachfremden Personen erlaubt, sich relativ schnell in das Feld einzuarbeiten. Die neue Kontrolloberfläche erlaubt es, durch wenige Klicks den gewünschten Strom einzustellen oder, falls gewünscht, mit Hilfe eines Cycle-Vorgangs ein bestimmtes Magnetfeld aufzubauen. Zudem werden nun alle möglichen Fehler, welche den reibungslosen Operationsbetrieb stören könnten, auf dem Operator Interface grafisch dargestellt. Durch eine eingebaute Sicherung ist es nun schwieriger geworden, den Cycle-Vorgang unbeabsichtigt zu unterbrechen.

Während der Testphase wurde das Problem der Fehlkommunikation aufgedeckt, welches mangels Zeit nicht direkt behoben werden konnte. Falls der langfristige Betrieb den erwarteten Kriterien nicht entspricht, sollte der Fehler näher untersucht werden. Des Weiteren konnte eine gewünschte Aufzeichnen-Funktion noch nicht realisiert werden. Falls es möglich wäre, ein Aktivitätenprotokoll zu erstellen, würde das die Reproduzierbarkeit zusätzlich erhöhen. Zudem konnte die Slope-Rate nicht exakt bestimmt werden, da die in der Bedienungsanleitung gedruckte Beschreibung mit den Beobachtungen nicht übereinstimmt. Die tatsächliche Slope-Rate ist dabei um ca. vier Größenordnungen kleiner als erwartet. Um die Durchführung zukünftiger Experimente zu verbessern, kann zusätzlich eine Eichgerade bestimmt werden, welche den Zusammenhang zwischen Elektronenenergie und Kanal des elastischen Peaks beschreibt.

Abbildungsverzeichnis

1.1	Übersicht über den S-DALINAC	3
2.1	Abbildung der Cycle-Funktion	5
2.2	Kontrollpanel des Netzteils	6
2.3	EPICS-Architektur	7
2.4	Beispiel für einen record	8
2.5	Beispiel für eine record-Verknüpfung	9
2.6	Beispiel für einen Eintrag in der Protokoll-Datei	9
2.7	Überblick über die CSS Oberfläche	10
3.1	Mismatch-Eintrag in der Protokoll-Datei	14
3.2	Umsetzer mit Stromversorgung	15
4.1	Finite-State Machine	17
4.2	OPI für die QCLAM Magnete	18
4.3	OPI für den Abbruch des Cycle-Vorgangs	19
4.4	OPI für die Cycle-Einstellungen	19
4.5	records für die Stromauslese und -Eingabe	20
4.6	records für die Fehlerauslese	21
4.7	records für die Cycle-Einstellungen	22
4.8	Testlauf zur Überprüfung der Linearität der Slope-Rate	23

Tabellenverzeichnis

2.1	Cycle-Variablen	6
2.2	Übersicht über die verwendeten record-Typen	8
3.1	Befehle zur Netzteilsteuerung	13
4.1	Übersicht über die möglichen Fehler bzw. Zustände	21

Literaturverzeichnis

- [1] KNIRSCH, Martin: *Konzeption, Aufbau und Erprobung eines hochauflösenden QCLAM-Elektronenspektrometers mit grossem Raumwinkel und hoher Impulsakzeptanz am Elektronenbeschleuniger S-DALINAC*, TU Darmstadt, Diss., 1991
- [2] TU DARMSTADT: *Der supraleitende Darmstädter Elektronenbeschleuniger S-DALINAC*. Aufgerufen am: 15.10.2013. http://www.ikp.tu-darmstadt.de/sdalinac_ikp/
- [3] RICHTER, Achim: *Operational experience at the S-DALINAC*. EPAC '96, Sitges, 1996
- [4] SCHMIDT, Werner: *Messung kinematischer Koinzidenzen an Protonen und Entwicklung eines Computerprogramms zur Magnetfeldsteuerung am QCLAM-Spektrometer*, TU Darmstadt, Diplomarbeit, 1992
- [5] BRUKER: *Prüf- und Messprotokoll*. Sep 1988
- [6] KRUGMANN, Andreas: *Persönliche Mitteilung*
- [7] ARGONNE NATIONAL LABORATORY: *Experimental Physics and Industrial Control System*. Aufgerufen am: 14.10.2013. <http://www.aps.anl.gov/epics/>
- [8] CLAUSEN, Matthias: *EPICS at DESY. KEK/J-PARK*, 2004
- [9] LACKEY, Charon L.: *ACNET vs. EPICS at Fermilab*. Mar 2006
- [10] KONRAD, Martin: *Development and commissioning of a digital rf control system for the S-DALINAC and migration of the accelerator control system to an EPICS-based system*, TU Darmstadt, Diss., 2013
- [11] BURANDT, Christoph: *Persönliche Mitteilung*
- [12] COSYLAB: *Visual Database Configuration Tool*. Aufgerufen am: 14.10.2013. http://cosylab.com/resources/downloads_/
- [13] ARGONNE NATIONAL LABORATORY: *EPICS Wiki*. Aufgerufen am: 14.10.2013. https://wiki-ext.aps.anl.gov/epics/index.php/RRM_3-14
- [14] BRUKER: *Description of B-MC 20D, local and remote control of Power Supplies*. Karlsruhe, Germany: Bruker Analytische Messtechnik, 1988
- [15] PSI: *Stream Device*. Aufgerufen am: 15.10.2013. <http://epics.web.psi.ch/software/streamdevice/doc/index.html>
- [16] COLLABORATION OF SNS, FRIB, BNL, DESY, ITER: *Control System Studio*. Aufgerufen am: 16.10.2013. <http://controlsystemstudio.github.io/index.html>
- [17] LEOPLES, David: *Persönliche Mitteilung*
- [18] BIRKHAN, Jonny: *Persönliche Mitteilung*