



Notizen für das EPICS Training am PSI



Zur Anmeldung beim EPICS Training findet man mehr
Informationen auf der Seite
<http://epics.web.psi.ch/training/>
oder man schickt eine mail an
elke.zimoch@psi.ch

Inhaltsverzeichnis

Inhaltsverzeichnis	2
Ein kleines EPICS Wörterbuch.....	3
Linux Befehle.....	4
Was ist eigentlich Linux?.....	4
Wozu Linux in einem EPICS Training?	4
Ein paar nützliche Befehle	5
Die CA-Befehle	6
caget	6
caput.....	6
cado.....	6
cainfo.....	6
camon.....	7
Online Hilfe und Optionen.....	8
PSI Skripte und spezielle Befehle.....	9
swit.....	9
bootinfo	9
externalLinks.....	10
findrecord	10
rmc	10
Die Namenskonvention an der SLS	11
Allgemeine Hinweise.....	11
Beispiele für Record Namen an der SLS	12
Alarme und ihre Farben	13
NO_ALARM	13
MINOR	13
MAJOR.....	13
INVALID	14
not Connected	14
Wie man caQtDM am PSI benutzt.....	15
Wie man GUIs entwickelt.....	15
Wie man GUIs ausführt	15
Wie man medm GUIs zu caQtDM konvertiert.....	15
caQtDM auf Windows	15
Installation von caQtDM.....	15
Mehr Dokumentation	15
Übung: Ansteuerung eines Kühlers	16
Problembeschreibung.....	16
Die benötigten Records.....	16
Die Substitution Datei.....	17
Die Template Datei	17
Entwickle die Übung weiter.....	19
Mehr Informationen über EPICS	20
Zehn nette Tatsachen über EPICS	20

Ein kleines EPICS Wörterbuch

EPICS benutzt viele Spezialbegriffe, die nicht immer eine naheliegende Bedeutung haben. Eine kleine Übersicht soll hier ein bisschen Klarheit schaffen.

Channel	Eine andere Bezeichnung für Prozess Variable.
Channel Access	Das Kommunikations-Protokoll von EPICS mit dessen Hilfe die Informationen über das Netzwerk geschickt werden.
Channel Access Server	Software, die Zugriffe über Channel Access auf Prozess Variablen zur Verfügung stellt. Vereinfacht: Software, die Records zur Verfügung stellt. Eine IOC ist ein Channel Access Server.
Channel Access Client	Software, die über Channel Access auf Prozess Variablen zugreift, zum Beispiel <code>caQtDM</code> .
EPICS	Abkürzung für E xperimental P hysics and I ndustrial C ontrol S ystem.
Input	(wörtlich Eingabe) bedeutet innerhalb von EPICS, dass die Informationen von der Hardware zum Nutzer gehen. Oder technischer ausgedrückt: vom Channel Access Server zum Channel Access Client.
Input Output Controller (IOC)	Ein Computer auf dem <code>iocCore</code> läuft. Das ist eine Sammlung von EPICS Routinen, die benutzt werden um Prozess Variablen zu definieren und Algorithmen zu implementieren. <code>iocCore</code> benutzt Records, um Prozess Variablen und ihr Verhalten zu definieren. Eine IOC ist gleichzeitig ein Channel Access Server und ein Channel Access Client, das heisst, IOCs können miteinander Daten austauschen.
Output	(wörtlich Ausgabe) bedeutet innerhalb von EPICS, dass die Informationen vom Nutzer zur Hardware gehen. Oder technischer ausgedrückt: vom Channel Access Client zum Channel Access Server).
Process Variable (PV)	Ein Datensatz mit einem Namen, der über diesen eindeutigen Namen angesprochen werden kann. Der PV-Name setzt sich zusammen aus einem Recordnamen und einem Feldnamen mit einem Punkt als Trennzeichen. Zum Beispiel: <code>AIRDI-PCT:CURRENT.EGU</code> . Eine PV ist die Basis-Dateneinheit des Channel Access Protokolls.
Record	Ein Record ist eine Zusammenfassung von Pocess Variablen, die auf einer IOC implementiert ist. Es existieren dutzende von verschiedenen Record-Typen, jeweils mit eigenen Attributen (Feldern) und einer eigenen Funktionalität. Die gebräuchlichsten Typen sind in der EPICSBase Distribution enthalten.
Soft IOC	Ein Computer (meist ein PC), auf dem nicht ausschliesslich <code>iocCore</code> läuft oder laufen kann. <code>iocCore</code> ist dann nur ein Programm unter anderen. Teilweise ist keine besondere Hardware an diesen PC angeschlossen. Am PSI wird mit einer IOC ein VME-Rechner gemeint, während eine SoftIOC ein PC ist.
VME	VME ist kein EPICS zugehöriger Begriff! VME steht für VERSA Module Eurocard und ist eine Computer-Architektur. Im Unterschied zum PC sind alle Karten über einen gemeinsamen Bus verbunden.

Linux Befehle

Was ist eigentlich Linux?

- Linux wurde von Linus Torvalds nach dem Vorbild von Unix programmiert.
- Linux ist ein frei verfügbares Unix-ähnliches Betriebssystem. Es ist damit ein Konkurrenz zu Windows.
- Linux und die meisten Programme, die auf Linux Rechnern laufen, stehen unter der GPL (GNU-Public-License), die sicherstellt, dass freie Software auch frei bleibt. Frei heißt dabei insbesondere, dass der Programm-Quelltext verfügbar und veränderbar ist. Man spricht hier auch von Open Source.
- Linux läuft auf allen gängigen PCs. Bei neuer oder seltener Hardware kann es schwierig werden, den richtigen Treiber zu finden.

Wozu Linux in einem EPICS Training?

Wir benutzen Linux PCs als Terminals, also zur Darstellung der Vorgänge auf VME Computern, zur Entwicklung von EPICS Datenbanken und zur Erzeugung von graphischen Oberflächen. Das Betriebssystem Linux wird auch in den SLS Kontrollräumen und an den Beamlines verwendet. Es ist damit die Computerumgebung, die auch beim täglichen Umgang mit EPICS an der SLS häufig Verwendung findet.

Wir benutzen im Training nur sehr wenige Linux Befehle, so dass Vorkenntnisse nicht nötig sind. Im wesentlichen sind das die unten aufgeführten Befehle und einige spezielle Programme, die erklärt werden, da es sich nicht um Standard Linux Befehle handelt.

Wir arbeiten im wesentlichen mit einer Shell, die es erlaubt, Befehle direkt einzutippen und auszuführen. Die meisten Befehlsnamen sind bei Linux (wie bei Unix) Kürzel, die nur entfernt an einen Namen erinnern. Es sieht manchmal so aus, als wenn sich die Programmierer von Unix keine Namen mit mehr als fünf Buchstaben merken konnten. Die häufig benutzten Befehle haben deutlich weniger Zeichen.

In einer Shell ist es ein Unterschied, ob Befehle und Namen gross oder klein geschrieben werden. Die Befehle werden klein geschrieben, viele EPICS Namen werden gross geschrieben. Es ist wichtig, genau auf die Schreibweise zu achten.

Ein paar nützliche Befehle

- cd** "change **d**irectory" Wechselt das Verzeichnis. Wird hinter dem Befehl nichts angegeben, so wird in das Home Verzeichnis des Nutzers gewechselt. Wird ein Verzeichnisname angegeben, so wird in das entsprechende Verzeichnis gewechselt. Mit „cd ..“ wechselt man ein Verzeichnis nach oben. Beispiel: `cd G/TRAINING/T1`
- cp** "copy" Kopiert eine Datei woanders hin oder erstellt eine Kopie unter einem anderen Namen. Erst wird das Original angegeben, dann der gewünschte Name der Kopie oder das Verzeichnis in das kopiert werden soll. Beispiel: „`cp MTEST-VME-T1_startup.script test`“
Erstellt eine Kopie der Datei MTEST-VME-T1_startup.script die dann test heisst.
- ll** "long list" Ist eine Abkürzung für den Befehl "ls -l", der den Inhalt des derzeitigen (oder des angegebenen) Verzeichnisses in der langen Form ausgibt. Eine ausgegebene Zeile sieht etwa so aus
`-rw-rw-r-- 1 zimoch e sls 686 Mar 30 08:29 startup.script`
die ersten 10 Zeichen geben die verschiedenen Rechte der Datei an (und ob es eine Datei oder ein Verzeichnis ist), dann kommt ein Zähler, dann der Name des Besitzers der Datei und seine Gruppe. Die Zahl 686 in dem Beispiel gibt die Grösse der Datei in Bytes an, danach steht das Datum, an dem die Datei zuletzt geändert wurde. Als letztes steht schliesslich der Name der Datei.
- pwd** "print working **d**irectory" Zeigt den vollständigen Namen des aktuellen Verzeichnisses an. Der Befehl braucht keine weiteren Argumente.
- rm** "remove" L'öscht eine oder mehrere Dateien. ACHTUNG: diese Dateien sind weg. Sie werden nicht in einen Papierkorb verschoben, aus dem man sie im Notfall wieder heraus fischen kann, sondern sie werden ganz schlicht und einfach gelöscht! Immer noch mal genau überlegen!
Beispiel: `rm test`
- nedit &** Einer der vielen Editoren unter Linux. Der nedit ist einfach zu bedienen und wird daher im Training benutzt. Wenn man hinter dem Befehl einen Datei-Namen angibt, dann wird diese Datei in dem Editor geöffnet. Ansonsten kann man auch über das Menü *File* (und dann *Open...*) eine Datei öffnen.
Beispiel: `nedit G TRAINING T1 COOLER.template &`

Die CA-Befehle

Die folgenden Befehle können benutzt werden, um aus der Kommandozeile einer Shell auf einem Linux-Rechner in einem Maschinen-Netzwerk EPICS-Records abzufragen und Werte zu setzen:

```
caget <RecordName>  
caput <RecordName> <Wert>  
cado <RecordName>  
cainfo <RecordName>  
camon <RecordName>
```

Aufgrund der Gateway-Konfiguration können vom Büro aus nur Werte gelesen werden und vom Beamline-Netzwerk aus können nur EPICS-Records im Beamline-Netz geschrieben werden. Diese Regeln gibt es aus Sicherheitsgründen.

caget

caget - den Wert eines Records auslesen. Es wird ein formatierter Wert mit Einheit angegeben, wenn nicht nach einem spezifischen Feld des Records gefragt wird. Zusätzlich werden Status- und Severity-Feld angegeben, wenn ein Alarmzustand vorliegt.

```
zimoche@pc4859[/work] ~ > caget ARIDI-PCT:CURRENT  
ARIDI-PCT:CURRENT 301 mA
```

caput

caput - setzt den Wert eines Records. Natürlich sollte es sich bei dem angegebenen Wert um den richtigen Wertetyp handeln (Zahl, Zeichen, ...). Der Wert wird ohne Einheit angegeben. Zur Bestätigung wird der zurückgelesene Wert mit Einheit angezeigt nachdem das Schreiben beendet ist. Damit steht eine einfache Möglichkeit zur Verfügung um zu testen ob der Befehl erfolgreich war.

```
zimoche@pc4859[/work] ~ > caput V13BL-OP-SH1:SIZE-SET 5  
V13BL-OP-SH1:SIZE-SET 5.000 mm
```

cado

cado - setzt ein Record auf 1. Dies ist vor allem bei Start- oder Stopp-Records sinnvoll. Der Befehl erzeugt keine Antwort in der Eingabezeile..

```
zimoche@pc4859[/work] ~ > cado V13BL-OP-SH1:STOP  
zimoche@pc4859[/work] ~ >
```

cainfo

cainfo - fragt einige Record-Felder ab. Damit erhält man z.B. Informationen, auf welcher IOC das Record existiert und welche Alarmschwellen und Grenzen gesetzt sind.

Ausserdem kann der Befehl benutzt werden um die möglichen Eingaben einer Aufzählung zu erfahren. Der Befehl kann auch auf Felder eines Records, z.B. das SCAN Feld angewendet werden.

```
zimoeh_e@pc4859[/work] ~ > cainfo ARIDI-PCT:CURRENT
ARIDI-PCT:CURRENT:
NAME ARIDI-PCT:CURRENT
VAL 300.96249157
EGU mA
SIZE 1
SEVR NO_ALARM
STAT NO_ALARM
TIME {07/23/04 13:21:18.695144729}
TYPE DBF_DOUBLE
HOPR 500
LOPR 0
DRVH 500
DRVL 0
HIHI 0
HI 0
LOLO 0
LO 0
PREC 0
ACCESS R
IOC slscag02.psi.ch
RTYP ai
DESC ""
zimoeh_e@pc4859[/work] ~ >
```

camon

camon - es wird ein Monitor für ein Record gesetzt. Die angezeigten Daten sind dieselben wie bei caget. Sobald der Wert des Records sich ändert, wird dieser geänderte Wert auch dargestellt. Mit der Tastenkombination <CTRL><C> kann der Befehl beendet werden.

```
zimoeh_e@pc4859[/work] ~ > camon ARIDI-PCT:CURRENT
ARIDI-PCT:CURRENT 301 mA
ARIDI-PCT:CURRENT 300 mA
ARIDI-PCT:CURRENT 301 mA
ARIDI-PCT:CURRENT 300 mA
ARIDI-PCT:CURRENT 301 mA
ARIDI-PCT:CURRENT 300 mA
ARIDI-PCT:CURRENT 301 mA
...
```

Online Hilfe und Optionen

Wenn man die genauen Befehle vergessen hat, kann man eine beliebigen der CA-Befehle ohne Argument angeben und damit eine englische Online Hilfe aufrufen.

```
zimoeh_e@pc4859[/work] ~ > caget
usage: caget [flags] <channel> [<channel> ...]
cagets [flags] <channel> [<channel> ...]
caput [flags] <channel> <value> [<channel> <value> ...]
caputq [flags] <channel> <value> [<channel> <value> ...]
cainfo [flags] <channel> [<channel> ...]
camon [flags] <channel> [<channel> ...]
cado [flags] <channel> [<channel> ...]
cawait [flags] <channel> '<condition>' [<channel>
'<condition>'...]
caget reads and formats values from channels (arrays too)
cagets writes 1 to .PROC and reads after processing has
finished
caput writes, waits until processing finishes and reads
back
caputq writes but does not wait for processing
cainfo reads additional information
camon starts monitors (terminate with CTRL-C)
cado writes 1 but does not wait for processing
cawait waits until any condition ('>4.3', '!3...5', etc)
matches
accepted flags:
-date add record execution date
-localdate add host date
-time add record execution time
-localtime add host time
-noname don't add channel name
-nounit don't add units
-stat always add severity and status
-nostat never add severity and status
-hex show integer values as hex
-prec <digits> override the PREC field
-timeout <sec> timeout cawait after <sec> seconds
-version print version and exit
-help print this help text and exit
zimoeh_e@pc4859[/work] ~ >
```


PSI Skripte und spezielle Befehle

Auf den Entwicklungsservern (Linux) sind einige nützliche Skripte installiert. Man benutzt sie genau wie Befehle, d.h. man gibt sie in einer Kommandozeile ein. Dementsprechend muss man auch hier auf Gross- und Kleinschreibung achten. Bei einigen Skripten kann man zusätzliche Optionen angeben, die nach dem Skript Namen, durch Leerzeichen getrennt, angegeben werden.

Die hier aufgeführten Programme sind auf allen Entwicklungsservern und auf allen Konsolen innerhalb der Maschinennetze installiert. Das bedeutet, sie stehen überall zur Verfügung, wo EPICS Record Entwicklung stattfinden soll. Auf den Entwicklungsservern kann man sich mit ssh oder NX Client remote einloggen (z.B. auf sls-lc oder fin-lc)

swit

Installiert IOC Anwendungen vom Arbeitsverzeichnis oder vom CVS Speicher nach \$INSTBASE. Im EPICS Training wird dieser Befehl verwendet, um die substitution und template Dateien in das boot Verzeichnis der IOC zu installieren.

Der Befehl muss in dem Verzeichnis ausgeführt werden, in dem sich auch die Dateien befinden, die installiert werden sollen. Für das EPICS Training ist es das Verzeichnis ~/G/TRAINING/T1 (für die erste Trainings IOC).

Eine Kurzfassung der verfügbaren Optionen zusammen mit ein paar dokumentierten Beispielen bekommt man so: `swit -h`

Beispiel:

```
swit -V -ioc MTEST-VME-T1
```

Mehr Informationen findet man auf der Webpage:

<http://city.psi.ch/city/index.php/swit/index.php>

bootinfo

Mit diesem Befehl kann man Informationen zum letzten boot Vorgang einer oder mehrerer IOCs abfragen. Zum Beispiel werden das Datum und die Uhrzeit, sowie die geladene EPICS Version angezeigt. Das ist zum einen nützlich, um die vorgenommenen Einstellungen zu überprüfen, zum anderen kann man aber auch einfach nur nachsehen, was letzte Nacht gemacht wurde.

Das Programm greift auf eine Oracle Datenbank zu und sollte unabhängig von der EPICS Umgebung sein. Es gibt verschiedenen Optionen, die mit dem Befehl `bootinfo -h` angezeigt werden können.

Generell sieht die Syntax des Befehls wie folgt aus:

```
bootinfo [options] [pattern]
```

Wobei `pattern` ein IOC Name oder ein Teil eines IOC Namens ist. Wenn nur ein Teil-Name angegeben wird, listet das Programm Informationen über alle Systeme auf, die diesen Namensteil enthalten.

Beispiel:

```
bootinfo MTEST-VME-T1
```

externalLinks

Dieses Script wird auf eine Substitution Datei angewendet. Es listet alle Records auf, die in links benutzt werden, aber nicht in der Datei selbst erzeugt werden. Vor allem für Tippfehler in Record Namen ist dies sehr nützlich.

Zusätzlich wird die Syntax des gesamten Substitution/Template Komplexes überprüft. Wenn `externalLinks` nicht korrekt funktioniert, wird die IOC im Allgemeinen nicht korrekt starten. Das hilft dabei, alle fehlenden Klammern und Anführungszeichen zu finden. Ausserdem wird festgestellt, wenn ein Feld nicht zum Record Typen gehört. Angewendet wird der Befehl `externalLinks` in dem Verzeichnis, in dem sich die Substitution Datei und die zugehörigen Template Dateien befinden. Das ist während des EPICS Trainings zum Beispiel das Verzeichnis `~/G/TRAINING/T1` für die erste Trainings Station.

Beispiel:

```
externalLinks MTEST-VME-T1_example.subs
```

findrecord

Manchmal kann man sich nur noch an einen Teil des Record Namens erinnern. Dann ist der Befehl `findrecord` in der Lage den gesamten Namen zu finden. Das Script durchsucht eine Oracle Datenbank, die jeweils beim booten einer IOC auf den neusten Stand gebracht wird. Man kann also nur nach aktuellen Namen suchen.

Der Befehl kann in einem beliebigen Verzeichnis angewendet werden, er sollte von der EPICS Umgebung unabhängig sein. Die generelle Syntax des Befehls ist `findrecord [options] [pattern]`

Wobei `pattern` der Teil des Record-Namens ist, an den man sich noch erinnert. Es werden dann alle Record Namen angezeigt, in denen `pattern` vorkommt. Es können auch Platzhalter (wildcards) eingesetzt werden: `%` ersetzt beliebig viele Zeichen, `_` ersetzt genau ein Zeichen.

Es gibt bei `findrecord` noch verschiedene Optionen. Die Dokumentation dazu kann man so anzeigen lassen: `findrecord -h`

Beispiel:

```
findrecord MTRT1
```

rmc

Die Abkürzung steht für **remote minicom**. Der Befehl stellt eine direkte serielle Verbindung (minicom) zum Debug Ausgang der VME IOC her. In dem neuen Fenster kann man dann mit EPICS und VxWorks Befehlen den Zustand seiner IOC überprüfen. Eine `rmc` Verbindung ist nur innerhalb des gleichen Netzwerks möglich. Abhängig vom Netzwerk wird man nach einem Passwort gefragt oder muss einen gültigen Namen (und zusätzlich das Passwort) angeben.

Beispiel:

```
rmc MTEST-VME-T1
```

Die Namenskonvention an der SLS

EPICS Records müssen eindeutige und einmalige Namen tragen. Dies kann nur durch eine Namenskonvention erreicht werden, die für alle Records verbindlich ist. Diese Konvention ist von Institut zu Institut unterschiedlich, es gibt also keine immer gültige Regelung. Innerhalb des PSI gibt es für jede Maschine eine eigene Konvention.

An der SLS bestehen Namen aus fünf Teilen:

[1][2] – [3] – [4] : [5]

1. Königreich (1 Buchstabe):
A=Accelerator/Beschleuniger, X=Beamline, I=Infrastruktur, M=Test
2. Domäne (4 Buchstaben):
Die Domäne ist eine funktionelle Region, wobei die Bedeutung je nach Königreich unterschiedlich ist:
 - Alle Beschleuniger (Linac, Ring, Booster, etc.) haben die gleichen Subsysteme wie Magnete, Vakuum, Diagnostik und so weiter. Daher setzt sich der Domänen-Name aus zwei Buchstaben, die den Beschleuniger bezeichnen (LI, RI, BO...), und zwei Buchstaben, die das Subsystem bezeichnen (MA, VA, DI...), zusammen. Die Domäne RIDI steht also für Diagnose-Komponenten im Speicherring.
 - Bei den Beamlines beginnt die Domäne mit der zweistelligen Nummer des Sektors in dem die Beamline liegt. Dann folgt ein Buchstabe, der die Lichtquelle beschreibt: S=small straight (kurze Gerade), M=medium straight (mittellange Gerade), L=long straight (lange Gerade) und D=Dipol für die Beamlines an Ablenkmagneten. Abgeschlossen wird der Domänenname durch einen je Sektion fortlaufenden Buchstaben, der die Nummer der Beamline in diesem Sektor beschreibt. Beispiel: 04SA
3. Gerät (Optional):
Da verschiedene System auch verschiedenen aufgebaut sind, können die beiden Namensteile 3 und 4 unterschiedliche Bedeutungen haben. Bei den Beamlines ist es üblich, das der dritte Teil den Beamlinebereich angibt: FE=Front End, OP=Optik und ES=Experimental Station. Je nach Aufbau kann noch eine Nummer das Subsystem genauer spezifizieren.
4. Komponente:
Da verschiedene System auch verschiedenen aufgebaut sind, können die beiden Namensteile 3 und 4 unterschiedliche Bedeutungen haben.
5. Funktion (Optional):
Eine willkürliche Abkürzung für die Funktion die das Record bereit stellt..

Allgemeine Hinweise

- Trennzeichen: Es wurde der Bindestrich (-) (ASCII 45) gewählt, um Probleme mit anderer Software zu vermeiden, die die Namen benutzen muss: Der Punkt (.) wird von EPICS für andere Zwecke benutzt, der Slash (/) erzeugt Probleme bei der

Nutzung von MS-EXCEL, der Unterstrich (_) wurde als hässlich aussortiert, der Backslash (\) als unbequem und weitere (, ; ^ > = +) als zu ungebräuchlich. Trotzdem wird gerade beim letzten Namensteil auch häufiger der Unterstrich verwendet.

- Es wird keine Vorgabe zur Gross- oder Kleinschreibung (ausser den Teilen 1 und 2, die gross geschrieben werden) gemacht.
- Die Maximal-Länge der Namen ist auf 27 Buchstaben inklusive Trennzeichen limitiert, da EPICS (bis Version 3.14) keine längeren Namen zulässt.
- Es ist zu empfehlen, sich zunächst die Namen von bereits existierenden Records anzuschauen, wenn man neue Records erzeugt. Häufig haben sich in einem Bereich Standards gebildet, die dann auch berücksichtigt werden sollten.
- Soll in einem Bereich ein bereits existierendes System auf EPICS umgestellt werden, so sollte der ursprüngliche Name so weit wie möglich übernommen werden um die Umstellung zu erleichtern.

Beispiele für Record Namen an der SLS

Die hier genannten Beispiele wurden zufällig ausgewählt..

- ARIDI-PCT:CURRENT
 1. Königreich: A - Das Record gehört zum Beschleuniger
 2. Domäne: RIDI - RI für Speicherring, DI für Diagnose
 3. Gerät: nicht verwendet
 4. Komponente: PCT - Gerätetyp für Strommessungen
 5. Funktion: CURRENT – Strom

Das Record ARIDI-PCT:CURRENT liest den aktuellen Strahlstrom des Speicherrings von einem PCT Gerät.
- X10SA-ID-GAP:SET
 1. Königreich: X - Das Record gehört zu einer Beamline
 2. Domäne: 10SA - Name der Beamline (Sector zehn (10), kurze Gerade (S), erste Beamline (A))
 3. Gerät: ID - Geräte Gruppe ist Undulator (insertion device)
 4. Komponente: GAP - Die Komponente ist das Gap (der Magnetabstand) des Undulators
 5. Funktion: SET - Die Funktion erlaubt ein setzen des Wertes.

Mit dem Record X10SA-ID-GAP:SET kann ein Benutzer den Wert für den Magnetabstand (das ist auch die Magnetstärke) des Undulators in Sektor zehn (kurze Gerade) gesetzt werden, der Licht für die erste Beamline dieses Sektors liefert.
- ALIRF-A2-KLY:COMP
 1. Königreich: A - Das Record gehört zum Beschleuniger
 2. Domäne: LIRF - LI für Linac, RF für Hochfrequenz (radio frequency)
 3. Gerät: A2 - Es gibt zwei RF Stationen, dieses Record gehört zur zweiten
 4. Komponente: KLY - Die Komponente ist das Klystron
 5. Funktion: COMP - Es werden zwei Werte verglichen (compare)

Das Record ALIRF-A2-KLY:COMP vergleicht zwei Werte des Klystrons von der RF Station A2 im Linac.

Alarmer und ihre Farben

Ein EPICS Record kann fünf verschiedene "Alarm"-Zustände annehmen:

Status Bezeichnung	Farbe
NO ALARM	Grün
MINOR	Gelb
MAJOR	Rot
INVALID	Weiss
not Connected	Weisser Kasten oder Violett

Diese Zustände können zum Beispiel mit dem Alarmhandler (CA-Client) überwacht werden. Aber auch normale GUIs (Graphical User Interfaces) stellen sie häufig farblich dar.

Um konsistent zu bleiben sollten für neue Oberflächen die gleichen Farben verwendet werden.

Es ist wichtig, den Alarmzustand eines Records in selbstgeschriebenen Programmen mit abzufragen und dann auch zu verwenden. Allein am Wert eines Records ist nicht zu erkennen, ob es sich um einen "guten" oder einen "schlechten" Wert handelt.

Da EPICS die Möglichkeit bietet, Schwellen für Warnungen und Fehler einzutragen, ist es besser, dies zu nutzen, als in jedem Client-Programm die Vergleiche selbst vorzunehmen. Wenn die Schwellen im Record, also auf dem Server konfiguriert werden, werden Alarme für alle Clients gleich berechnet - selbst wenn der Client nur ein Kommandozeilen Befehl ist. Zusätzlich erleichtert es die spätere Anpassung der Schwellen, da nur an einer Stelle geändert werden muss.

NO_ALARM

Dies sollte der Normalzustand für ein Record sein. Wenn er farblich hervorgehoben werden soll, dann wird Grün verwendet. Wenn man die CA-Befehle benutzt, dann wird dieser "Alarm"-Zustand nicht mit angezeigt.

MINOR

Dieser Alarm zeigt an, dass eine erste Sicherheitsschwelle über- oder unterschritten wurde. Der Wert für diese Schwelle wird im Record festgelegt: beim AO und AI Records heissen die Felder HIGH und LOW (wobei zusätzlich die Felder HSV und LSV auf den Wert "MINOR" gesetzt werden müssen).

Ein MINOR-Alarm ist eine Warnung und sollte dazu dienen, den Operateur auf ein mögliches Problem frühzeitig aufmerksam zu machen. Bei einer Darstellung sollte Gelb verwendet werden.

MAJOR

Dies ist ein wirklicher Alarm, der anzeigt, dass eine Sicherheitsschwelle über- oder unterschritten wurde. Der Wert für diese Schwelle wird im Record festgelegt: beim AO und AI Records heissen die Felder HIHI und LOLO (wobei zusätzlich die Felder HHSV und LLSV auf den Wert "MAJOR" gesetzt werden müssen).

Ein MAJOR-Alarm ist ein Fehler im Beschleuniger oder in der Gerätefunktion, der vom Operateur behoben werden sollte. Bei einer Darstellung wird Rot verwendet.
Ein MAJOR-Alarm ist kein Fehler des Kontrollsystems!

INVALID

Wenn ein Record den Status INVALID annimmt, bedeutet dies, dass der Wert des Records nicht mehr wahr ist. Dies tritt zum Beispiel auf, wenn die Hardware nicht funktioniert oder nicht angeschlossen ist. Bei Darstellungen wird der Wert weiss dargestellt.

Direkt nach dem booten sind übrigens alle Records im Status INVALID um anzuzeigen, dass dieses Record noch nicht prozessiert wurde. Zum Beispiel wurde das Record X10SA-OP-BPM3:FIT-H0 noch nicht prozessiert:

```
zimoch_e@pc4859[/work] ~ > caget X10SA-OP-BPM3:FIT-H0  
X10SA-OP-BPM3:FIT-H0 -3082.0000 (SEVR:INVALID STAT:UDF)
```

Hier wird das Feld SEVR (Severity bedeutet Ernsthaftigkeit) und das Feld STAT (Status) angezeigt; UDF steht für "Undefined" also nicht definiert. Nach dem ersten prozessieren (z.B. Lesen von der Hardware) ändert sich der Zustand automatisch.

not Connected

Dies ist streng genommen gar kein Zustand des Records, sondern eine Meldung eines CA-Clients, dass das Record nicht gefunden werden konnte. Dies ist der Fall, wenn sich innerhalb einer Wartezeit kein CA-Server gemeldet hat. Auftreten kann dies bei Netzwerkproblemen, einer IOC die gebootet wird oder schlicht bei einem Schreibfehler im Recordnamen.

CaQtDM stellt einen nicht verbundenen Kanal als weisses Kästchen dar, einige andere Programme im Kontrollsystem verwenden Pink.

Dieser Alarm-Zustand sollte nicht dauerhaft vorliegen, da der Client bei jedem Server der bootet erneut einen Broadcast abschickt und so das Netzwerk unnötig belastet.

Wie man caQtDM am PSI benutzt

caQtDM ist standardmässig auf allen GFA SL6 (Linux) Computern installiert.

Wie man GUIs entwickelt

Um den Qt-Designer zu starten, benutzt man folgendes Kommando:

```
caqtdm_designer
```

Wie man GUIs ausführt

Um am PSI ein caQtDM GUI auszuführen (zu starten), benutzt man folgendes Kommando:

```
caqtdm
```

Wie man medm GUIs zu caQtDM konvertiert

Um ein medm GUI, das *xxx.adl* heisst, zu caQtDM zu konvertieren, benutzt man folgendes Kommando:

```
adl2ui xxx.adl
```

Dies erzeugt ein caQtDM GUI mit dem Namen *xxx.ui*.

caQtDM auf Windows

Installation von caQtDM

- Aus dem AIT Software Kiosk das caQtDM Paket installieren
- Nun existiert ein caQtDM Menue unter Start/Programs/caQtDM/caQtDM-Menue. von dem aus man den Developer und die Ausführung von GUIs starten kann.

Mehr Dokumentation

Die Hauptseite für caQtDM Dokumentation ist

<http://epics.web.psi.ch/software/caqtdm/>

Übung: Ansteuerung eines Kühlers

Ursprünglich wurde diese Übung von John Mclean vom Argone National Laboratory für die "Getting started with EPICS" Vorlesungsreihe entwickelt.

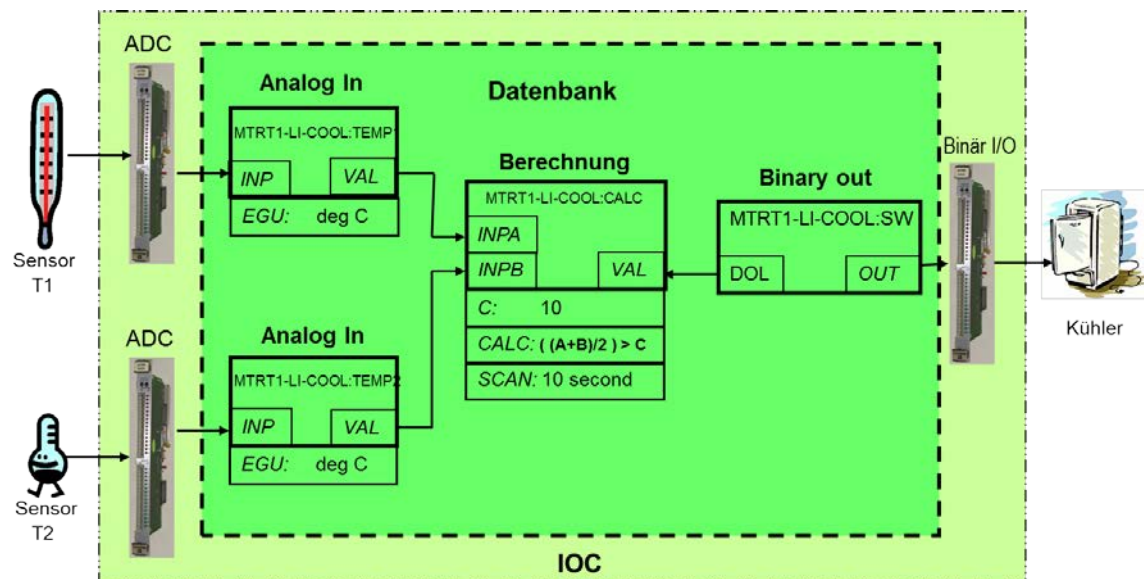
Problembeschreibung

Im Linac gibt es einen Wasserkühler, der eingeschaltet werden muss, wenn die durchschnittliche Temperatur der beiden Temperatursensoren über einen Grenzwert steigt. Der Grenzwert ist normalerweise 10 °C.

Im Training greifen wir auf unser Vorstellungsvermögen zurück und betrachten die ersten zwei Potentiometer als Temperatursensoren. Dies erlaubt uns eine bessere Kontrolle über die "Temperaturen" als echte Messfühler. Der Schaltzustand des Kühlers wird durch die erste LED dargestellt.

Die benötigten Records

Streng genommen sind nur die Analog Input und das Binary Output Record tatsächlich notwendig, wenn die eigentliche Regelung zum Beispiel von einem Skript übernommen würde. EPICS bietet aber die Möglichkeit, einfache Rechnungen direkt in einem Record auszuführen. Damit wird verhindert, dass jemand die Regelung aus Versehen abschaltet.

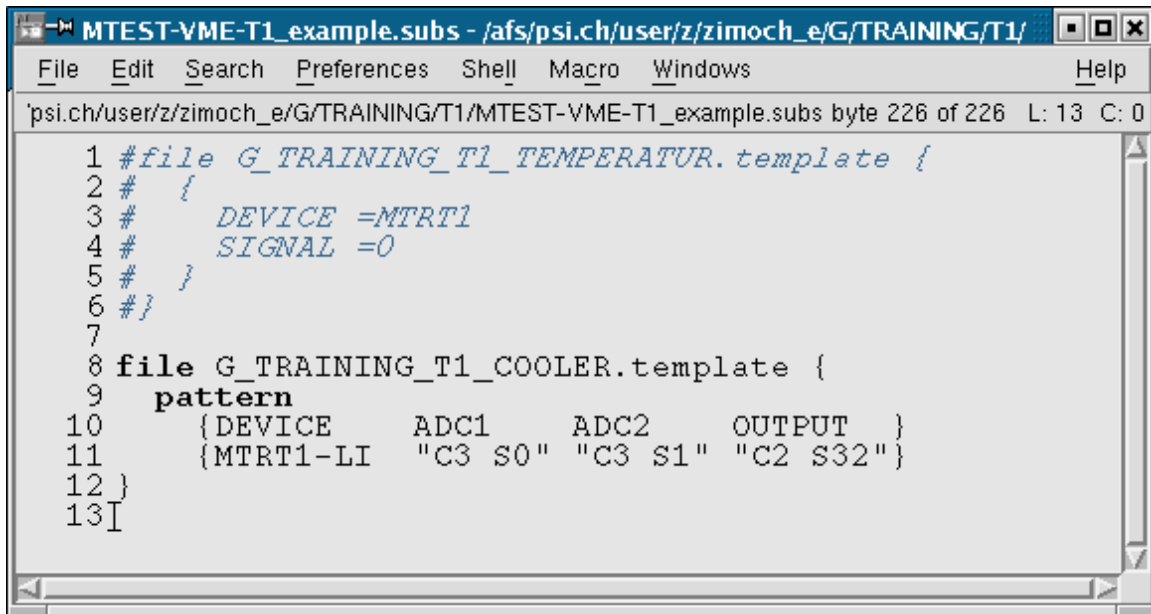


Die Record Namen sind:

- MTRT1-LI-COOL:TEMP1
- MTRT1-LI-COOL:TEMP2
- MTRT1-LI-COOL:COMPARE
- MTRT1-LI-COOL:SW

Die Substitution Datei

Die Substitution Datei findet sich im Verzeichnis ~/G/TRAINING/T1 und heisst MTEST-VME-T1_example.subs



```
1 #file G_TRAINING_T1_TEMPERATUR.template {
2 # {
3 #   DEVICE =MTRT1
4 #   SIGNAL =0
5 # }
6 #}
7
8 file G_TRAINING_T1_COOLER.template {
9   pattern
10  {DEVICE ADC1 ADC2 OUTPUT }
11  {MTRT1-LI "C3 S0" "C3 S1" "C2 S32" }
12 }
13 |
```

Es gibt zwei equivalente Möglichkeiten in der Substitution Datei Macros zu definieren:

1. **file** example.template {
 {
 DEVICE = MTRT5
 SIGNAL = "C3 S1"
 }
}
2. **file** example.templeate {
 pattern
 {DEVICE SIGNAL }
 {MTRT5 "C3 S1" }
}

Die erste Methode sollte man für wenige Macros und wenige Aufrufe der Template Datei verwenden. Die zweite Methode ist eher für viele Macros und viele Aufrufe geeignet.

Die Template Datei

Die Template Datei findet sich im Verzeichnis ~/G/TRAINING/T1 und heisst G_TRAINING_T1_COOLER.template

```

record (ai, "$(DEVICE)-COOL:TEMP1")
{
    field (DESC, "Sensor T1")
    field (DTYP, "Hy8401")
    field (INP, "#$(ADC1) @")
    field (EGU, "Grad C")
    field (PREC, "1")
    field (LINR, "LINEAR")
    field (EGUF, "100")
    field (EGUL, "-100")
    field (HOPR, "100")
    field (LOPR, "0")
    field (HIGH, "51")
    field (HIHI, "52")
    field (SCAN, ".1 second")
}

record (ai, "$(DEVICE)-COOL:TEMP2")
{
    field (DESC, "Sensor T2")
    field (DTYP, "Hy8401")
    field (INP, "#$(ADC2) @")
    field (EGU, "Grad C")
    field (PREC, "1")
    field (LINR, "LINEAR")
    field (EGUF, "100")
    field (EGUL, "-100")
    field (HOPR, "100")
    field (LOPR, "0")
    field (HIGH, "51")
    field (HIHI, "52")
    field (SCAN, ".1 second")
}

record (calc, "$(DEVICE)-COOL:COMPARE")
{
    field (DESC, "Calculation")
    field (INPA, "$(DEVICE)-COOL:TEMP1")
    field (INPB, "$(DEVICE)-COOL:TEMP2")
    field (INPC, "10")
    field (CALC, "((A+B)/2) > C")
    field (SCAN, "10 second")
    field (FLNK, "$(DEVICE)-COOL:SW")
}

```

```

record (bo, "$(DEVICE)-COOL:SW")
{
  field (DESC, "Switch for Cooler")
  field (DOL, "$(DEVICE)-COOL:COMPARE")
  field (ZNAM, "OFF")
  field (ONAM, "ON")
  field (DTYP, "Dim8001")
  field (OUT, "#$(OUTPUT) @")
  field (OMSL, "closed_loop")
}

```

Entwickle die Übung weiter

Die folgenden Anleitungen sind Vorschläge, wie die Übung weiter ausgestaltet werden könnte. Es ist aber in jedem Fall zu empfehlen, zunächst ein GUI (Graphical User Interface) zu entwickeln, da es die Fehlersuche vereinfacht.

- Entwickle ein GUI auf dem folgendes angezeigt wird:
 - beide Temperaturen mit Bezeichnung
 - der Zustand des Kühlers (ob er On oder Off ist)
 - die Alarm-Zustände der Temperatur-Sensoren – hier ist es eventuell notwendig zuerst einige Felder der Temperatur Records zu ändern
 - eine Grafik mit dem zeitlichen Verlauf der beiden Temperaturen

Weitere Aufgaben, die etwas mehr mit Records zu tun haben:

- erzeuge ein Record für die Durchschnittstemperatur und zeige den Wert an
- erzeuge ein Record für die Grenztemperatur, so dass diese verändert werden kann (sie sollte aber zwischen 10 und 45 °C bleiben), und stelle den Wert dar
- für einige Wartungsarbeiten soll es möglich sein, die Regelung auszuschalten und den Kühler unabhängig von den Temperaturen ein und aus zu schalten
- ein neues Gerät, eine Heizung, soll eingeschaltet werden, wenn die durchschnittliche Temperatur unter einem weiteren Grenzwert liegt, der zwischen 0 und 20 °C eingestellt werden kann. Es muss sicher gestellt sein, dass niemals der Kühler und die Heizung gleichzeitig an sind. Die Heizung benutzt den gleichen Treiber wie der Kühler und den Output Link "#C2 S33 @"
Dies alles soll auch im GUI angezeigt werden.
- Ermögliche die Regelschleife auch mit anderen EPICS Standard Perioden zu betreiben (um diese Standards heraus zu finden benutzt man den Befehl "caget MTRT1-LI-COOL:TEMP1.SCAN")
- Es gibt eine weitere Kühler/Heizung Kombination, diesmal im Speicherring. Die Records müssen dupliziert werden, die Namen sollen aber jetzt mit MTRT1-SR anfangen.

Mehr Informationen über EPICS

EPICS ist eine gemeinschaftlich entwickelte Sammlung von Open Source Software (Werkzeuge, Libraries, Anwendungen), mit der weltweit verteilte Echtzeitsysteme aufgebaut werden. Genutzt wird EPICS vor allem in Wissenschaftlichen Grossanlage wie Teilchenbeschleuniger und Teleskope.

Mehr Informationen und internationale Kontaktadressen finden sich auf der Internetseite

<http://www.aps.anl.gov/epics/>

Zehn nette Tatsachen über EPICS

1. Es ist gratis. Keine Lizenzgebühren, keine neuen Gebühren für jede neue Version. EPICS kann man einfach aus dem Internet herunterladen.
2. Es ist Open Source (Quellcode ist zugänglich). Änderungen und Anpassungen an spezielle Bedürfnisse sind dadurch möglich.
3. Es gibt eine Menge Leute die es benutzen, dass bedeutet, es ist gut getestet.
4. Alles was der Client wissen muss um auf die Daten zuzugreifen ist der PV Name (Prozess Variablen Name). Es gibt keinen einzelnen Rechner, der alle Namen kennt und so einen Engpass darstellt..
5. Man kann unter den besten Werkzeugen auswählen ...
6. ... oder seine eigenen schreiben.
7. Der langweilige Teil ist schon gemacht (z.B. ist die Channel Access Schnittstelle stabil und gut getestet)
8. Es gibt bereits eine Menge Erfahrungen in verschiedenen Instituten..
9. Eine gute Erweiterung findet internationale Aufmerksamkeit.
10. Es ist egal, ob man 10 oder 10 Millionen PVs braucht. EPICS ist in dieser Beziehung fast beliebig skalierbar.