

# DSME-Injektions Septum

Gerätemodell und Softwareentwurf

Klaudia Herlo

<b>Änderungsprotokoll</b>			
Datum	GM-Version	Name	Kommentar
30.Oktober.2007	DSME.01	KHer	Erstellen des Dokumentes

# Inhaltsverzeichnis

<b>I</b>	<b>Das Gerätemodell</b>	<b>5</b>
1	Die Aufgabe des Gerätes	5
2	Die Hardware des Gerätes	5
3	Definition der Bits des Hardwarestatus	5
4	Die Steuerung des Gerätes	6
4.1	Aufgaben im Normalbetrieb	6
4.1.1	Plattenabstand	6
4.1.2	Verschieben des kompletten Plattensystems	7
4.1.3	Initiale Position	7
4.2	Zeitkritische Anforderungen	7
4.3	Einordnung in das Timing	7
4.4	Schrägstellung der Plattenelektroden	7
4.5	Festlegung von Startwerten	8
4.5.1	Kaltstarts	8
4.5.2	Warmstarts	8
4.6	Ableitung des Hardwarefehler-Bits aus dem Gerätestatus	8
5	Die Repräsentation des Gerätes	8
5.1	Kennzeichnung des Gerätemodells	9
<b>II</b>	<b>Der Entwurf der Software</b>	<b>11</b>
6	Softwareentwurf	11
7	Lokale Datenbasis	11
7.1	Tabelle der Konstanten	11
7.2	Konfigurationsparameter	12
8	Die Default-Properties	12
8.1	Die Default-Master-Properties	12
8.1.1	INFOSTAT	13
8.1.2	INIT	13
8.1.3	POWER	13
8.1.4	RESET	14
8.1.5	STATUS	14

8.1.6	CONSTANT	14
8.2	Die Default-Slave-Properties	14
<b>9</b>	<b>Die Gerätemodell-Properties</b>	<b>14</b>
9.1	Die Gerätemodell-Master-Properties	14
9.2	Die Gerätemodell-Slave-Properties	15
<b>10</b>	<b>DeviceAccess::EqModDSME::CallINITPOS Class Reference</b>	<b>15</b>
10.1	Detailed Description	15
<b>11</b>	<b>DeviceAccess::EqModDSME::ReadINFO Class Reference</b>	<b>16</b>
11.1	Detailed Description	16
<b>12</b>	<b>DeviceAccess::EqModDSME::CallSTOP Class Reference</b>	<b>17</b>
12.1	Detailed Description	18
<b>13</b>	<b>DeviceAccess::EqModDSME::ReadDISTS Class Reference</b>	<b>18</b>
13.1	Detailed Description	18
<b>14</b>	<b>DeviceAccess::EqModDSME::WriteDISTS Class Reference</b>	<b>19</b>
14.1	Detailed Description	19
<b>15</b>	<b>DeviceAccess::EqModDSME::ReadDISTI Class Reference</b>	<b>19</b>
15.1	Detailed Description	20
<b>16</b>	<b>DeviceAccess::EqModDSME::WritePOSIREL Class Reference</b>	<b>20</b>
16.1	Detailed Description	20
<b>17</b>	<b>DeviceAccess::EqModDSME::ReadPOSITS Class Reference</b>	<b>21</b>
17.1	Detailed Description	21
<b>18</b>	<b>DeviceAccess::EqModDSME::WritePOSITS Class Reference</b>	<b>22</b>
18.1	Detailed Description	22
<b>19</b>	<b>DeviceAccess::EqModDSME::ReadPOSITI Class Reference</b>	<b>22</b>
19.1	Detailed Description	23
<b>20</b>	<b>DeviceAccess::EqModDSME::ReadPOSIABSI Class Reference</b>	<b>23</b>
20.1	Detailed Description	23
<b>21</b>	<b>DeviceAccess::EqModDSME::ReadPOSIABSS Class Reference</b>	<b>24</b>
21.1	Detailed Description	24
<b>Index</b>		<b>27</b>

## Teil I

# Das Gerätemodell

## 1 Die Aufgabe des Gerätes

Das Injektions Septum besteht aus zwei Plattenelektroden (Anode und Kathode). Diese Platten werden durch vier Schrittmotoren in horizontaler Ebene verfahren. Eine Platte ist jeweils mit zwei Motoren ausgestattet, die immer parallel verfahren werden sollen.

Das Injektions Septum hat die Aufgabe die Strahlüberführung(Strahleinschuß) vom Unilac ins SIS18 zu verbessern.

Die nächsten Abbildungen verdeutlichen den Aufbau des Gerätemodells.

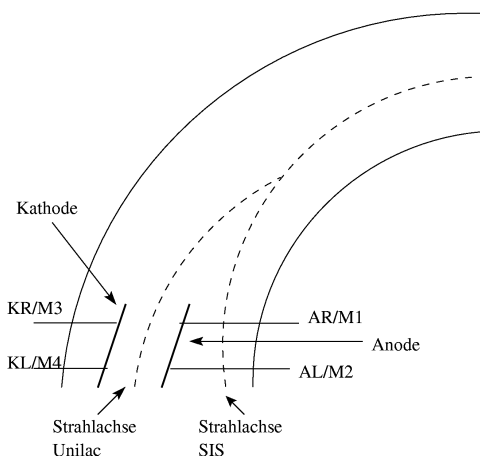


Abbildung 1: Strahlüberführung vom Unilac in den SIS18

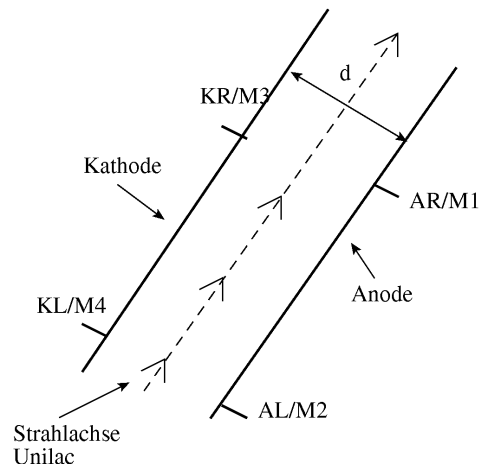


Abbildung 2: Skizze vom Aufbau des Septums

## 2 Die Hardware des Gerätes

Die Bedienung des Gerätes erfolgt über zwei Rechereinheiten. Die Recheneinheit microIOC-M-Box-PMAC (mBox) beinhaltet die Gerätesoftware, die Treibersoftware für die Schrittmotoren, verschiedene Softwarebibliotheken und stellt verschiedene notwendigen Service zur Verfügung. Die andere Recheneinheit microIOC-M-Box-PMAC-Extension-Box (auch PDC genannt für Power Drive Case) beinhaltet den Leistungstreiber für die Schrittmotorsteuerung und dient als Schnittstelle zum Gerät.

## 3 Definition der Bits des Hardwarestatus

Die Geräte liefern 3 Bytes Statusinformation. Die Funktionscodes zum Lesen der entsprechenden Bytes sind in der folgenden Tabelle zusammengefasst.

Die Bits 0 . . . 7 sind die systemweiten sogenannten generierten Softwarestatusbits (in engl. derived status bits). Die Statusbits im Einzelnen sind in der folgenden Tabelle zusammengefasst.

Bit	Name	Bedeutung	
		High (1)	Low (0)
0	Power	on	off
1	Remote/Local	Remote	Local
2		reserved	
3		reserved	
4		reserved	
5	Interlock	no	yes
6	HW Error	no	yes
7	SW Error	no	yes
8	Motors moving	all park	at least one is moving
9	Motor 1 inner limit switch	not reached	reached
10	Motor 1 outer limit switch	not reached	reached
11	Motor 1 moving	in park	moving
12	Motor 2 inner limit switch	not reached	reached
13	Motor 2 outer limit switch	not reached	reached
14	Motor 2 moving	in park	moving
15	Motor 3 inner limit switch	not reached	reached
16	Motor 3 outer limit switch	not reached	reached
17	Motor 3 moving	in park	moving
18	Motor 4 inner limit switch	not reached	reached
19	Motor 4 outer limit switch	not reached	reached
20	Motor 4 moving	in park	moving
21	Minimal distance	not reached	reached
22	Maximal distance	not reached	reached
23	Anode angle warning	no	yes
24	Anode angle error	no	yes
25	Kathode angle warning	no	yes
26	Kathode angle error	no	yes
27	Interlock	deaktiv	aktiv
28		reserved	
29		reserved	
30		reserved	
31		reserved	

## 4 Die Steuerung des Gerätes

Hier wird beschrieben, wie das Gerät (die Hardware) bedient werden muss. Das beinhaltet die Anforderungen *vom* Gerät als auch die Anforderungen *an das* Gerät.

### 4.1 Aufgaben im Normalbetrieb

#### 4.1.1 Plattenabstand

Der Plattenabstand ist der Abstand zwischen der Plattenelektrode *Kathode* und der Plattenelektrode *Anode*. Der Plattenabstand( $d$ ) ist in der Abbildung 1 ersichtlich. Der Plattenabstand kann beliebig eingestellt werden, wobei ein minimaler Plattenabstand  $d_{min}$  und ein maximaler Platten-

abstand  $d_{max}$  eingehalten werden muss. Beide Variablen  $d_{min}$  und  $d_{max}$  erhalten ihren Wert aus einer Datenbank, in der alle gerätespezifischen Gerätekonstanten stehen. Siehe auch die Beschreibung der Property [8.1.6](#).

#### 4.1.2 Verschieben des kompletten Plattensystems

Das komplette Plattensystem kann auf eine neue Position verschoben werden. Dabei bleibt der Plattenabstand unverändert. Die neue Position des Septums wird über eine neue absolute Position der Anode angegeben. Siehe auch die Property [19.1](#).

#### 4.1.3 Initiale Position

Das komplette Plattensystem kann auf eine vorher definierte Position verfahren werden. Die initiale Positionen für alle Motoren stehen in der Datenbank, in der alle gerätespezifischen Gerätekonstanten stehen. Siehe auch die Beschreibung der Seite [8.1.6](#).

### 4.2 Zeitkritische Anforderungen

Bei dem Septum sind keine zeitkritischen Anforderungen zu berücksichtigen. Die Platten werden mit einer Geschwindigkeit von 0.5mm/sec verfahren.

### 4.3 Einordnung in das Timing

Das Septum nimmt nicht an der PPM (Puls-zu-Puls-Modulation) teil und ist damit unabhängig vom Timing.

### 4.4 Schrägstellung der Plattenelektroden

Eine gesteuerte Schrägstellung der Plattenelektroden wird von der Software nicht unterstützt. Trotzdem wird die Parallelität beider Plattenelektroden überprüft um eine große Schrägstellung zu verhindern. In der Abbildung [3](#) sieht man eine Schrägstellung der Platten.

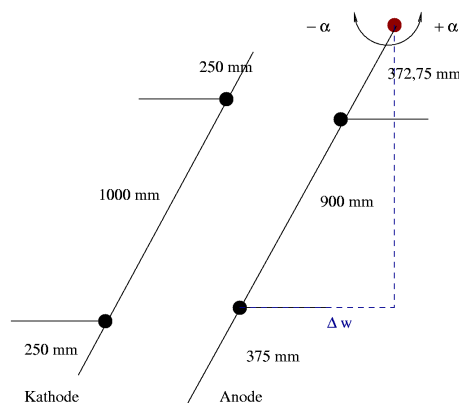


Abbildung 3: Schrägstellung des Septums

Der Maximalwinkel jeder Plattenelektrode beträgt +/- 3mrad. Größere Winkel können zu einem erheblichen Schaden des Septums führen.

Deshalb wird im Treiber der Schrittmotoren und in der Gerätesteuerung eine Schrägstellung der Platten überprüft. Momentan wird eine Schrägstellung von maximal 1 mm zugelassen. Das bedeutet ist die Differenz zwischen zwei Motoren an einer Platte größer als die zugelassene Differenz vom 1mm, ist ein weiteres Verfahren dieser Platte nicht mehr möglich.

## 4.5 Festlegung von Startwerten

### 4.5.1 Kaltstarts

Bei einem Kaltstart werden folgende Aktionen durchgeführt:

- Das Gerät wird initialisiert.
- Alle Sollwerte werden auf die Istwerte gesetzt.
- Hardwarestatus wird aktualisiert.

### 4.5.2 Warmstarts

Bei einem Warmstart werden folgende Aktionen durchgeführt:

- Der Hardwarestatus wird aktualisiert.

## 4.6 Ableitung des Hardwarefehler-Bits aus dem Gerätestatus

Ein Hardwarefehler (angezeigt im Hardwarefehler-Bit des Status) liegt vor, wenn eines der folgenden Bits des Hardwarestatus *nicht* den angegebenen Wert (nicht OK) anzeigt.

Bit	Name	Wert
9	Motor 1 -Inner end position	1
10	Motor 1 -Outer end position	1
12	Motor 2 -Inner end position	1
13	Motor 2 -Outer end position	1
15	Motor 3 -Inner end position	1
16	Motor 3 -Outer end position	1
18	Motor 4 -Inner end position	1
19	Motor 4 -Outer end position	1
21	Minimal distance	1
22	Maximal distance	1
23	Anode angle warning	1
24	Anode angle error	1
25	Kathode angle warning	1
26	Kathode angle error	1

## 5 Die Repräsentation des Gerätes

Dieses Kapitel beschreibt, wie das Gerät nach höheren Ebenen hin abgebildet wird.



## 5.1 Kennzeichnung des Gerätemodells

Das Gerätemodell hat die Bezeichnung **DSME**.

Die Gerätemodellnummer ist  $77_{dez}$ .



## Teil II

# Der Entwurf der Software

## 6 Softwareentwurf

Bei der Steuerung des Injektions Septums müssen einige sicherheitsrelevanten Aspekte berücksichtigt werden. Nachfolgend werden alle sicherheitsrelevanten Aspekte, die in der Software behandelt werden, beschrieben.

1. Es kann immer nur eine Schreibe/Write Property zu einer Zeit ausgeführt werden.
2. Vor jedem Fahrbefehl wird überprüft, ob sich dieser in einem zulässigen Bereich befindet. Der minimale und maximale Plattenabstand wird überprüft. Sowie die minimalen und maximalen Positionen aller Motoren wird überprüft.
3. Alle Fahrbefehle werden in 1 mm Fahrbefehle zerteilt. Nach jedem 1 mm Fahrbefehl wird die Parallelität kontrolliert.
4. Die Parallelität der Platten: Bei einer Differenz von mehr als 1 mm zwischen zwei Motoren an einer Platte, werden alle Motoren gestoppt. Diese Überprüfung findet in der Gerätesoftware statt und auch in der darunter liegenden Treiberschicht. Das bedeutet man hatte eine doppelte Sicherheit, dass eine Schrägstellung der Platten kontrolliert und verhindert wird.

## 7 Lokale Datenbasis

### 7.1 Tabelle der Konstanten

Für das Gerät gibt es eine Beschreibung aus 10 Elementen in der Konstantentabelle der lokalen Datenbasis. Die Elemente haben in der Reihenfolge folgende Bedeutung:

- 1:** Initiale Position von Motor 1 in 1/10mm. Diese Position wird vom Motor 1 gezielt beim Aufruf der Property **10.1** angefahren.
- 2:** Initiale Position von Motor 2 in 1/10mm. Diese Position wird vom Motor 2 gezielt beim Aufruf der Property **10.1** angefahren.
- 3:** Initiale Position von Motor 3 in 1/10mm. Diese Position wird vom Motor 3 gezielt beim Aufruf der Property **10.1** angefahren.
- 4:** Initiale Position von Motor 4 in 1/10mm. Diese Position wird vom Motor 4 gezielt beim Aufruf der Property **10.1** angefahren.
- 5:** Minimaler Plattenabstand der zwischen den Plattenelektroden Anode und Kathode zugelassen wird.
- 6:** Maximaler Plattenabstand der zwischen den Plattenelektroden Anode und Kathode zugelassen wird.
- 7:** Grenzwert in 1/10mm Warnung- Schrägstellung Anode
- 8:** Grenzwert Warnung- Schrägstellung Kathode

**9:** Grenzwert Fehler- Schrägstellung Anode

**10:** Grenzwert Fehler- Schrägstellung Kathode

Siehe auch die Beschreibung der Property Read Constant [8.1.6](#).

## 7.2 Konfigurationsparameter

Für das Gerät gibt es eine Beschreibung aus 20 Parametern. Die Parameter werden aus einer Konfigurations-Datei gelesen und in der lokalen Datenbasis abgespeichert. Die Parameter haben in der Reihenfolge folgende Bedeutung.

**1:** Motor 1 dreht im Uhrzeigersinn (1) oder gegen den Uhrzeigersinn (-1)

**2:** Motor 2 dreht im Uhrzeigersinn (1) oder gegen den Uhrzeigersinn (-1)

**3:** Motor 3 dreht im Uhrzeigersinn (1) oder gegen den Uhrzeigersinn (-1)

**4:** Motor 4 dreht im Uhrzeigersinn (1) oder gegen den Uhrzeigersinn (-1)

**5:** Die Länge des Potentiometers in mm

**6:** Anzahl von Motorschritten für einen mm

**7:** Minimale Position der Kathode in 1/10mm

**8:** Maximale Position der Kathode in 1/10mm

**9:** Minimale Position der Anode in 1/10mm

**10:** Maximale Position der Anode in 1/10mm

**11:** Beschleunigung in ms von der Startfrequenz bis zum Erreichen der Maximalfrequenz

**12:** Maximalfrequenz

**13:** Offset von Motor 1

**14:** Offset von Motor 2

**15:** Offset von Motor 3

**16:** Offset von Motor 4

**17:** Volt per mm von Motor 1

**18:** Volt per mm von Motor 2

**19:** Volt per mm von Motor 3

**20:** Volt per mm von Motor 4

Siehe auch die Beschreibung der Property Read Info [11.1](#).

## 8 Die Default-Properties

### 8.1 Die Default-Master-Properties

Property	Klasse	Parameter		Daten		Größe	
		Anz.	Typ	Anz.	Typ	Einh.	Exp.
INFOSTAT	RA	0	–	25	BitSet32	1	0
INIT	N	0	–	0	–	–	–
POWER	R/W	0	–	1	BitSet16	1	0
RESET	N	0	–	0	–	–	–
STATUS	R	0	–	1	BitSet32	1	0
VERSION	RA	0	–	36	BitSet8	1	0
CONSTANT	RA	0	–	3	RealF	1	0

### 8.1.1 INFOSTAT

Bedeutung: Diese Property liefert einige wichtige Geräteinformationen in einem Zugriff.

Parameter: Keine.

Daten: Die 25 Langworte enthalten im Einzelnen:

- 1:** Gerätestatus (wie in der Property STATUS)
- 2:** Gibt in den oberen 16 Bits an, welcher virtuelle Beschleuniger aktiv gesetzt ist (ein Bit pro Beschleuniger). Das niederwertigste Bit (Bit 16) gibt den Beschleuniger 15 an, das Bit 31 den Beschleuniger 0. Die unteren 16 Bit sind nicht verwendet. Dabei bedeutet Null, dass der Beschleuniger inaktiv ist und Eins, dass der Beschleuniger aktiv ist.
- 3:** Master-Fehler. Hier ist derjenige Master-Gerätefehlercode mit dem schwersten Fehlergrad eingetragen. Bei mehreren Fehlern mit dem gleichen Fehlergrad wird der erste eingetragen, der gefunden wurde.
- 4:** Slave Fehler für virtuellen Beschleuniger 0. Entsprechend dem Master-Fehler wird hier der nach dem Fehlergrad schwerste Slave-Gerätefehlercode für den Beschleuniger 0 eingetragen.
- 5:** Entsprechend Punkt 4, aber für virtuellen Beschleuniger 1.
- ⋮
- 19:** Entsprechend Punkt 4, aber für virtuellen Beschleuniger 15.
- 20:** Reserviert für Erweiterungen.
- ⋮
- 25:** Reserviert für Erweiterungen.

### 8.1.2 INIT

Bedeutung: Initialisierung des Gerätes (Kaltstart). Für die dabei durchzuführenden Aktionen siehe Abschnitt [4.5.1](#) auf Seite [8](#).

Parameter: Keine.

Daten: Keine.

### 8.1.3 POWER

Bedeutung: Gibt an, ob der Leistungsteil des Gerätes ein- oder ausgeschaltet ist bzw. werden soll.

Parameter: Keine.

Daten: Das Datum kann nur zwei Werte annehmen. Null heißt, das Gerät ist eingeschaltet bzw. soll eingeschaltet werden. Eins heißt, das Gerät ist ausgeschaltet bzw. soll ausgeschaltet werden.

**Anmerkung:** Das Injektions Septum hat kein Leistungsteil. Das Datum hat immer den Wert 1.

#### 8.1.4 RESET

Bedeutung: Reset des Gerätes (Warmstart). Für die dabei durchzuführenden Aktionen siehe Abschnitt 4.5.2 auf Seite 8.

Parameter: Keine.

Daten: Keine.

#### 8.1.5 STATUS

Bedeutung: Auslesen des 32bit Gerätestatus.

Parameter: Keine.

Daten: Das 32bit Statuswort. Die Bits entsprechen den Statusbits, wie sie im Abschnitt 3 auf der Seite 5 erklärt sind.

#### 8.1.6 CONSTANT

Bedeutung: Liefert die gerätespezifischen Konstanten.

Parameter: Keine.

Daten: Die 10 Integerwerte bedeuten im Einzelnen:

- 1: Initiale Position von Motor 1
- 2: Initiale Position von Motor 2
- 3: Initiale Position von Motor 3
- 4: Initiale Position von Motor 4
- 5: Minimaler Plattenabstand
- 6: Maximaler Plattenabstand
- 7: Grenzwert Warnung- Schrägstellung Anode
- 8: Grenzwert Warnung- Schrägstellung Kathode
- 9: Grenzwert Fehler- Schrägstellung Anode
- 10: Grenzwert Fehler- Schrägstellung Kathode

## 8.2 Die Default-Slave-Properties

Für dieses Gerätemodell gibt es keine Default-Slave-Properties.

# 9 Die Gerätemodell-Properties

## 9.1 Die Gerätemodell-Master-Properties

Property	Klasse	Parameter		Daten		Größe	
		Anz.	Typ	Anz.	Typ	Einh.	Exp.
INITPOS	N	0	-	0	-	-	-
INFO	R	0	-	20	RealF	-	-
STOP	N	0	-	0	-	-	-
DISTS	R/W	0	-	1	BitSet16	m	-4
DISTI	R	0	-	1	BitSet16	m	-4
POSIREL	W	0	-	1	BitSet16	m	-4
POSITS	R/W	0	-	1	BitSet16	m	-4
POSITI	R	0	-	1	BitSet16	m	-4
POSIABSI	R	0	-	4	BitSet16	m	-4
POSIABSS	R	0	-	4	BitSet16	m	-4

## 9.2 Die Gerätemodell-Slave-Properties

Für dieses Gerätemodell gibt es keine Gerätemodell-Slave-Properties.

Die Beschreibung aller Gerätemodell-Properties erfolgt direkt über die USRs in den folgenden Abschnitten.

## 10 DeviceAccess::EqModDSME::CallINITPOS Class Reference

**Call INITPOS** (Move to initial position.)

```
#include <dsme-usrs.hh>
```

### Public Member Functions

- **CallINITPOS** (DSMEDevice \*dev, const string &proprname="INITPOS")
- virtual AccDevRetStatus **call** (SLong vrtAcc, const AccData &rcvPara)

### 10.1 Detailed Description

**Call INITPOS** (Move to initial position.)

This call property moves anode and kathode to a known initial positions. The initial position for the four motors are part of the device constants.

Category: master

Parameters: none

Data: none

Medlock: all

Access right: device

The documentation for this class was generated from the following file:

- `dsme-usrs.hh`

## 11 DeviceAccess::EqModDSME::ReadINFO Class Reference

**Read INFO** (Read Information )

```
#include <dsme-usrs.hh>
```

### Public Member Functions

- **ReadINFO** (DSMEDevice \*dev, const string &propname="INFO")
- virtual AccDevRetStatus **read** (SLong vrtAcc, const AccData &rcvPara, AccData &sndData, AccStamp &stamp, AccEFICD &eficd)

### 11.1 Detailed Description

**Read INFO** (Read Information )

This property deliver device specific Information.

Category: master

Parameters: none

Data: 20



Start	Name	Type	Description
0	mrot1	Float32	Rotation Motor 1
1	mrot2	Float32	Rotation Motor 2
2	mrot3	Float32	Rotation Motor 3
3	mrot4	Float32	Rotation Motor 4
4	potlength	Float32	Length of Poti for device position [mm]
5	steps	Float32	number of steps per mm
6	kathmin	Float32	Kathode, min. Position in 1/10 mm
7	kathmax	Float32	Kathode, max. Position in 1/10 mm
8	anodmin	Float32	Anode, min. Position in 1/10 mm
9	anodmax	Float32	Anode, max. Position in 1/10 mm
10	acc	Float32	Acceleration
11	velyo	Float32	Velocity (maximum frequency)
12	offset1	Float32	Offset of Motor 1
13	offset2	Float32	Offset of Motor 2
14	offset3	Float32	Offset of Motor 3
15	offset4	Float32	Offset of Motor 4
16	volts1	Float32	Volts per mm, Motor 1
17	volts2	Float32	Volts per mm, Motor 2
18	volts3	Float32	Volts per mm, Motor 3
19	volts4	Float32	Volts per mm, Motor 4

Medlock: none

Access right: free

The documentation for this class was generated from the following file:

- dsme-usrs.hh

## 12 DeviceAccess::EqModDSME::CallSTOP Class Reference

### Call STOP

```
#include <dsme-usrs.hh>
```

### Public Member Functions

- **CallSTOP** (DSMEDevice \*dev, const string &proprname="STOP")

- virtual AccDevRetStatus **call** (SLong vrtAcc, const AccData &rcvPara)

## 12.1 Detailed Description

### Call STOP

By call the Stop property all movements will immediately stop. The nominal value will not refresh.

Category: master

Parameters: none

Data: none

Medlock: all

Access right: device

The documentation for this class was generated from the following file:

- **dsme-usrs.hh**

## 13 DeviceAccess::EqModDSME::ReadDISTS Class Reference

**Read DISTS** (distance of the two plates)

```
#include <dsme-usrs.hh>
```

### Public Member Functions

- **ReadDISTS** (DSMEDevice \*dev, const string &propname="DISTS")
- virtual AccDevRetStatus **read** (SLong vrtAcc, const AccData &rcvPara, AccData &sndData, AccStamp &stamp, AccEFICD &eficd)

## 13.1 Detailed Description

**Read DISTS** (distance of the two plates)

The property read the set value (nominal position) of the distance from the two plates Anode and Kathode.

The property write DISTS write an new value of the distance from the two plates Anode and Kathode. It is to regard the minimum and maximum plate distance. See also the constants of the device.

Category: master

Parameters: none

Data: 1

Start	Name	Type	Description
0	dist	SWord	Distance, set value

Medlock: none

Access right: free

The documentation for this class was generated from the following file:

- `dsme-usrs.hh`

## 14 DeviceAccess::EqModDSME::WriteDISTS Class Reference

**Write DISTS** (distance of the two plates)

```
#include <dsme-usrs.hh>
```

### Public Member Functions

- **WriteDISTS** (DSMEDevice \*dev, const string &proprname="DISTS")
- virtual AccDevRetStatus **write** (SLong vrtAcc, const AccData &rcvPara, const AccData &rcvData)

### 14.1 Detailed Description

**Write DISTS** (distance of the two plates)

The property read the set value (nominal position) of the distance from the two plates Anode and Kathode.

The property write DISTS write an new value of the distance from the two plates Anode and Kathode. It is to regard the minimum and maximum plate distance. See also the constants of the device.

Category: master

Parameters: none

Data: 1

Start	Name	Type	Description
0	dist	SWord	Distance, set value

Medlock: all

Access right: device

The documentation for this class was generated from the following file:

- `dsme-usrs.hh`

## 15 DeviceAccess::EqModDSME::ReadDISTI Class Reference

**Read DISTI** (Read the actual plates distance)

```
#include <dsme-usrs.hh>
```

## Public Member Functions

- **ReadDISTI** (DSMEDevice \*dev, const string &proprname="DISTI")
- virtual AccDevRetStatus **read** (SLong vrtAcc, const AccData &rcvPara, AccData &sndData, AccStamp &stamp, AccEFICD &eficd)

### 15.1 Detailed Description

**Read DISTI** (Read the actual plates distance)

Read the actual plates distance directly from the potentiometer in 1/10mm .

Category: master

Parameters: none

Data: 2

Start	Name	Type	Description
0	distfront	SWord	Distance at Front Side in 1/10mm
1	distback	SWord	Distance at Back Side in 1/10mm

Medlock: none

Access right: free

The documentation for this class was generated from the following file:

- **dsme-usrs.hh**

## 16 DeviceAccess::EqModDSME::WritePOSIREL Class Reference

**Write POSIREL** (Move Kathode relatively.)

```
#include <dsme-usrs.hh>
```

### Public Member Functions

- **WritePOSIREL** (DSMEDevice \*dev, const string &proprname="POSIREL")
- virtual AccDevRetStatus **write** (SLong vrtAcc, const AccData &rcvPara, const AccData &rcvData)

### 16.1 Detailed Description

**Write POSIREL** (Move Kathode relatively.)

Move Kathode relatively in 1/10mm in positiv and negativ direction.

Category: master

Parameters: none

Data: 1

Start	Name	Type	Description
0	relmov	SWord	Relative movements for Kathode in 1/10mm.

Medlock: all

Access right: device

The documentation for this class was generated from the following file:

- **dsme-usrs.hh**

## 17 DeviceAccess::EqModDSME::ReadPOSITS Class Reference

### Read POSITS

```
#include <dsme-usrs.hh>
```

### Public Member Functions

- **ReadPOSITS** (DSMEDevice \*dev, const string &proprname="POSITS")
- virtual AccDevRetStatus **read** (SLong vrtAcc, const AccData &rcvPara, AccData &sndData, AccStamp &stamp, AccEFICD &eficd)

### 17.1 Detailed Description

#### Read POSITS

The property read Posits read the set value (nominal position) of anode.

The property write Posits move complete system of Kathode and Anode to new given absolute position of Anode.

Category: master

Parameters: none

Data: 1

Start	Name	Type	Description
0	posAnod	SWord	New absolute position of Anode in 1/10mm

Medlock: none

Access right: free

The documentation for this class was generated from the following file:

- **dsme-usrs.hh**

## 18 DeviceAccess::EqModDSME::WritePOSITS Class Reference

### Write POSITS

```
#include <dsme-usrs.hh>
```

### Public Member Functions

- **WritePOSITS** (DSMEDevice \*dev, const string &proprname="POSITS")
- virtual AccDevRetStatus **write** (SLong vrtAcc, const AccData &rcvPara, const AccData &rcvData)

### 18.1 Detailed Description

#### Write POSITS

The property read Posits read the set value (nominal position) of anode.

The property write Posits move complete system of Kathode and Anode to new given absolute position of Anode.

Category: master

Parameters: none

Data: 1

Start	Name	Type	Description
0	posAnod	SWord	New absolute position of Anode in 1/10mm

Medlock: all

Access right: device

The documentation for this class was generated from the following file:

- **dsme-usrs.hh**

## 19 DeviceAccess::EqModDSME::ReadPOSITI Class Reference

**Read POSITI** (Gives actual position of Anode)

```
#include <dsme-usrs.hh>
```

### Public Member Functions

- **ReadPOSITI** (DSMEDevice \*dev, const string &proprname="POSITI")
- virtual AccDevRetStatus **read** (SLong vrtAcc, const AccData &rcvPara, AccData &sndData, AccStamp &stamp, AccEFICD &eficd)

## 19.1 Detailed Description

**Read POSITI** (Gives actual position of Anode)

Gives actual position of Anode directly from the potentiometer in 1/10mm.

Category: master

Parameters: none

Data: 2

Start	Name	Type	Description
0	posfront	SWord	Actual position of Anode in 1/10mm, front side
1	posrear	SWord	Actual position of Anode in 1/10mm, rear side

Medlock: none

Access right: free

The documentation for this class was generated from the following file:

- `dsme-usrs.hh`

## 20 DeviceAccess::EqModDSME::ReadPOSIABSI Class Reference

**Read POSIABSI** (Gives the actual position of all motors)

```
#include <dsme-usrs.hh>
```

### Public Member Functions

- **ReadPOSIABSI** (DSMEDevice \*dev, const string &proprname="POSIABSI")
- virtual AccDevRetStatus **read** (SLong vrtAcc, const AccData &rcvPara, AccData &sndData, AccStamp &stamp, AccEFICD &eficd)

## 20.1 Detailed Description

**Read POSIABSI** (Gives the actual position of all motors)

Gives the actual position of all motors directly from the potentiometer in 1/10mm.

Category: master

Parameters: none

Data: 4

Start	Name	Type	Description
0	pos1	SWord	Actual Position in 1/10mm, of motor 1
1	pos2	SWord	Actual Position in 1/10mm, of motor 2
2	pos3	SWord	Actual Position in 1/10mm, of motor 3
3	pos4	SWord	Actual Position in 1/10mm, of motor 4

Medlock: none

Access right: free

The documentation for this class was generated from the following file:

- `dsme-usrs.hh`

## 21 DeviceAccess::EqModDSME::ReadPOSIABSS Class Reference

**Read POSIABSS** (Gives the nominal position of all motors. )

```
#include <dsme-usrs.hh>
```

### Public Member Functions

- **ReadPOSIABSS** (DSMEDevice \*dev, const string &propname="POSIABSS")
- virtual AccDevRetStatus **read** (SLong vrtAcc, const AccData &rcvPara, AccData &sndData, AccStamp &stamp, AccEFICD &eficd)

### 21.1 Detailed Description

**Read POSIABSS** (Gives the nominal position of all motors. )

Gives the nominal position of all motors in 1/10mm.

Category: master

Parameters: none

Data: 4

Start	Name	Type	Description
0	pos1	SWord	Nominal position in 1/10mm, of motor 1
1	pos2	SWord	Nominal position in 1/10mm, of motor 2
2	pos3	SWord	Nominal position in 1/10mm, of motor 3
3	pos4	SWord	Nominal position in 1/10mm, of motor 4

Medlock: none

Access right: free



The documentation for this class was generated from the following file:

- **dsme-usrs.hh**





- POWER ..... 13
- RESET ..... 14
- STATUS ..... 14

Proprty

- Schreiben ..... 11
- Write ..... 11

— R —

ReadDisti ..... 20

ReadDists ..... 19

ReadInfo ..... 17

ReadPosiabsi ..... 24

ReadPosiabss ..... 25

ReadPositi ..... 22

ReadPosits ..... 21

Repräsentation des Gerätes ..... 8

Reset ..... 8

— S —

Softwareentwurf ..... 11

Softwarestatus ..... 5

Startwerte ..... 8

Statusbits ..... 5

Steuerung des Gerätes ..... 6

— T —

Timing ..... 7

— W —

Warmstarts ..... 8

WriteDists ..... 19

WritePosirel ..... 21

WritePosits ..... 23

— Z —

Zeitkritische Anforderungen ..... 7