EUROPEAN SOUTHERN OBSERVATORY

Organisation Européenne pour des Recherches Astronomiques dans l'Hémisphère Austral

Europäische Organisation für astronomische Forschung in der südlichen Hemisphäre

# VERY LARGE TELESCOPE

**VLT Software**

**Motor Engineering Interface**

**User Manual**

Doc.No. VLT-MAN-ESO-17210-0669

Issue 1.6

Date 02/10/98

Prepared P.Duhoux 02/10/98 ...........................................
Name Date Signature

Approved A.Longinotti /10/98 ...........................................
Name Date Signature

Released G.Raffi /10/98 ...........................................
Name Date Signature

## Change Record

| Issue/Rev. | Date | Section/Page affected | Reason/Initiation/Document/Remarks |
|---|---|---|---|
| 1.0 | 13/10/95 | All | First preparation |
| 1.1 | 12/12/95 | 3 | Modifications for new release of motei<br>New configuration panels |
| 1.2 | 23/12/95 | 3,5 | Updated for new release. |
| 1.3 | 11/04/97 | 3 and 4 | Update for MAY97 release.<br>New subsections 3.7 and 4.5 |
| 1.4 | 20/10/97 | 3 | Update for NOV97. |
| 1.5 | 26/02/98 | All | General upgrade for APR98 |
| 1.6 | 02/10/98 | | Update for OCT98 |
| | | 3.2.1 | Added LCU monitor description |
| | | 3.4 & 3.5 | Updated Config & Control panels |
| | | 4.1 | Updated versions |
| | | 5.1 & 5.2 | Scripts man-pages & Panel help-files updated |

# TABLE OF CONTENTS

# 1 INTRODUCTION

The software described in this manual is intended to be used in the ESO VLT project by ESO and authorized external contractors only.

While every precaution has been taken in the development of the software and in the preparation of this documentation, ESO assumes no responsibility for errors or omissions, or for damage resulting from the use of the software or for the information contained herein.

## 1.1 Purpose

This document provides the User Manual of the **motei** module, the Engineering Interface for the Motor Control Module (hereafter MCM).

It is intended to provide engineers who operate, monitor or configure motors with a stand-alone environment for motor configuration, test and tuning.

The author assumes that the reader has some experience working with motors.

In addition to the Introduction, this manual contains the following chapters:

- **Overview** general description of the motei module.
- **User Guide** describes panels and available options.
- **Installation** describes how to install the software on the WS.
- **Reference** manual pages of panels and scripts used.

## 1.2 Reference Documents

| | | | |
|---|---|---|---|
| [1] | VLT Software Programming Standards | VLT-PRO-ESO-10000-0228 | Issue 1.0 |
| [2] | CCS-LCU/MCM User Manual (Part I: API-ACI) | VLT-MAN-ESO-17210-0600 | Issue 1.7 |
| [3] | Central Common Configuration User Manual | VLT-MAN-ESO-17210-0855 | Issue 2.0 |
| [4] | VLT Common Software Installation Manual | VLT-MAN-ESO-17200-0642 | Issue 1.9 |
| [5] | VLT Software Problem Report Change Request | VLT-MAN-ESO-17200-0981 | Issue 1.0 |

## 1.3 Abbreviations and Acronyms

The following abbreviations and acronyms are used in this document:

| | |
|---|---|
| ACI | Application Command Interface |
| CCS | Central Control Software |
| HOS | High Level Operating Software |
| HW | Hardware |
| ICS | Instrument Control Software |
| IEEE | Institute of Electrical and Electronics Engineers |
| I/O | Input/Output |
| ISO | International Standardisation Organisation |
| LCC | LCU Common Software |

| | |
|---|---|
| LCU | Local Control Unit |
| MCM | Motor Control Module |
| N/A | Not Applicable |
| TBC | To Be Confirmed |
| SW | Software |
| TBD | To Be Defined |
| TCS | Telescope Control Software |
| UIF | (Portable) User Interface (Toolkit) |
| VLT | Very Large Telescope |
| WS | Workstation |

## 1.4 Stylistic Conventions

The following styles are used:

**bold** - in the text, for commands, filenames, prefixes/suffixes as they have to be typed.

*italic* - in the text, for parts that have to be substituted with the real content before typing.

`teletype` - for examples.

`<name>` in the examples, for parts that have to be substituted with the real content before typing.

**bold** and *italic* are also used to highlight words.

*Items which are subject to change in future versions are marked in this way (AuthorRemark).*

✗Very important items are marked in this way (Warning).

% - indicates a UNIX shell prompt

-> - indicates a VxWorks shell prompt

## 1.5 Naming Conventions

This implementation follows the naming conventions as outlined in the VLT Programming Standards [1].

## 1.6 Problem Reporting/Change Request

The form described in [5] shall be used.

# 2 OVERVIEW

This part of the document provides a general description of the Motor Engineering Interface.

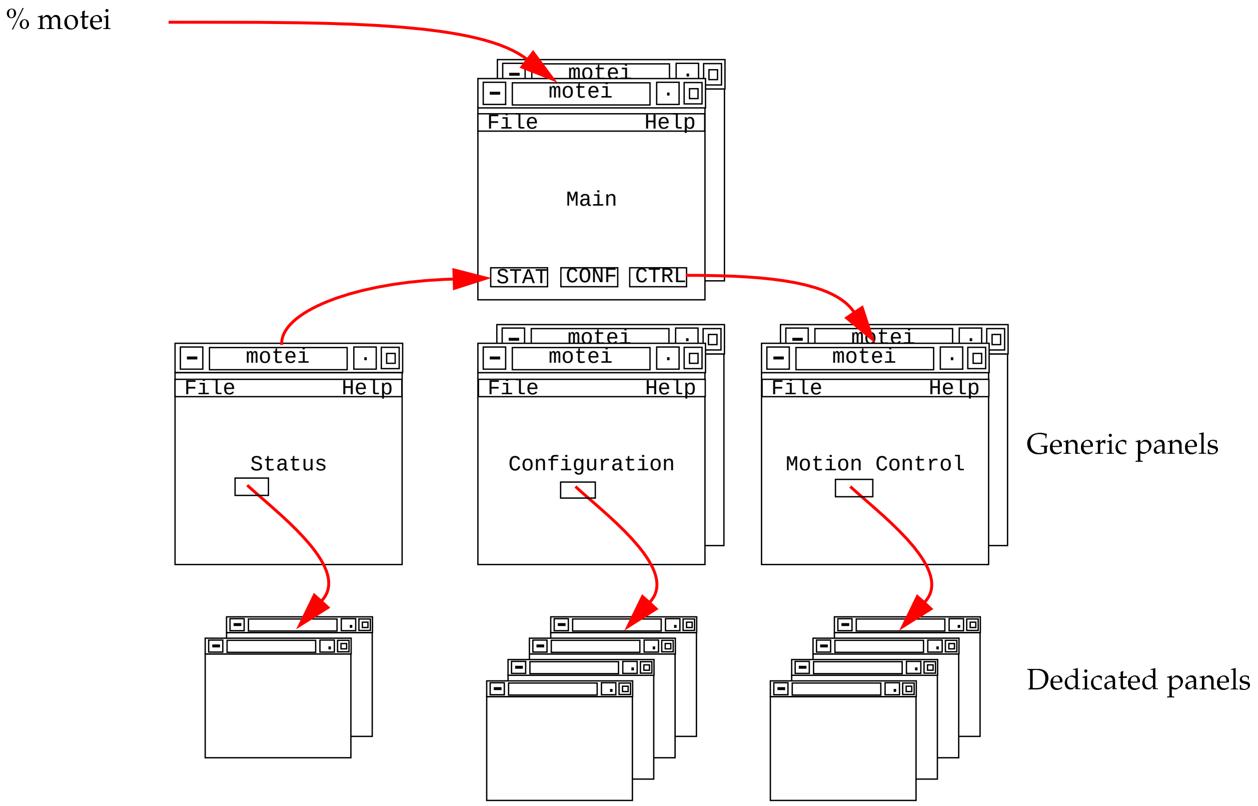
## 2.1 Description

The Motor Engineering Interface **motei** is a UIF-based tool and is available on CCS (RTAP) and CCS-lite WS environments.

It communicates with the LCU via the message system to the command interpreter of the Motor Control Module and allows the user to operate, monitor and configure motors. It is intended to be used by hardware and software engineers for configuration, test and tuning purposes, offering all the commands available in the ACI of the MCM.

### 2.1.1 Architecture

The architecture of the panels is shown in the figure below



## 2.2 Cook-Book

### 2.2.1 Starting with a new motor

1. Create/Configure the WS and/or LCU environment database (see Section 4.2)
    a. Instanciate one motor class or derived sub-class
    and/or
    b. Include the motei branch
2. Generate the database & start the WS environment (see Section 4.2)
   and/or
   Generate the database & configure + reboot LCU (see Sections 4.3 & 4.2)
3. Start motei (see Section 3.2 )
4. Select environment and motor
5. Load a configuration file (see Section 3.3.1)
   Invoke the configuration panel (See Section 3.4)
6. Modify/Set the configuration (see Sections 3.4.2 to 3.4.12)
7. Save the configuration in a file (see Section 3.3.2).

### 2.2.2 Tuning motor parameters

1. Start motei (see Section3.2)
2. Select environment and motor
3. Load a configuration file (see Section 3.3.1)
4. Invoke the Configuration panel (See Section 3.4)
5. Invoke Status and Control panels (See Sections 3.5 and 3.6)
6. Initialize and Move the motor (See Sections 3.5.1.2 to 3.5.1.5)
7. Tune parameters (See Sections 3.4.2 to 3.4.12) , Apply (and Save) configuration.
8. Repeat 6 to 7 until tuning satisfactory
9. Save the configuration in a file (see Section 3.3.2).

# 3 USER GUIDE

This chapter is dedicated to the general description of the Motor Engineering Interface application.

## 3.1 Introduction

It describes how to start the application **motei** , the main panel (section 3.2), the essential actions of loading and saving a motor configuration (section 3.3), the configuration panels (section 3.4), the control panels (section 3.5) and the status panels (section 3.6).

The first step to perform before invoking **motei** is to configure the working environment(s). They are of two kinds: CCS-RTAP environment on WorkStation, LCC environment on LCU. Whereas all actions are allowed on LCU environments, only configuration activities are available on WS.

The configuration of the environment consists solely in creating dedicated database branches in the environment database:

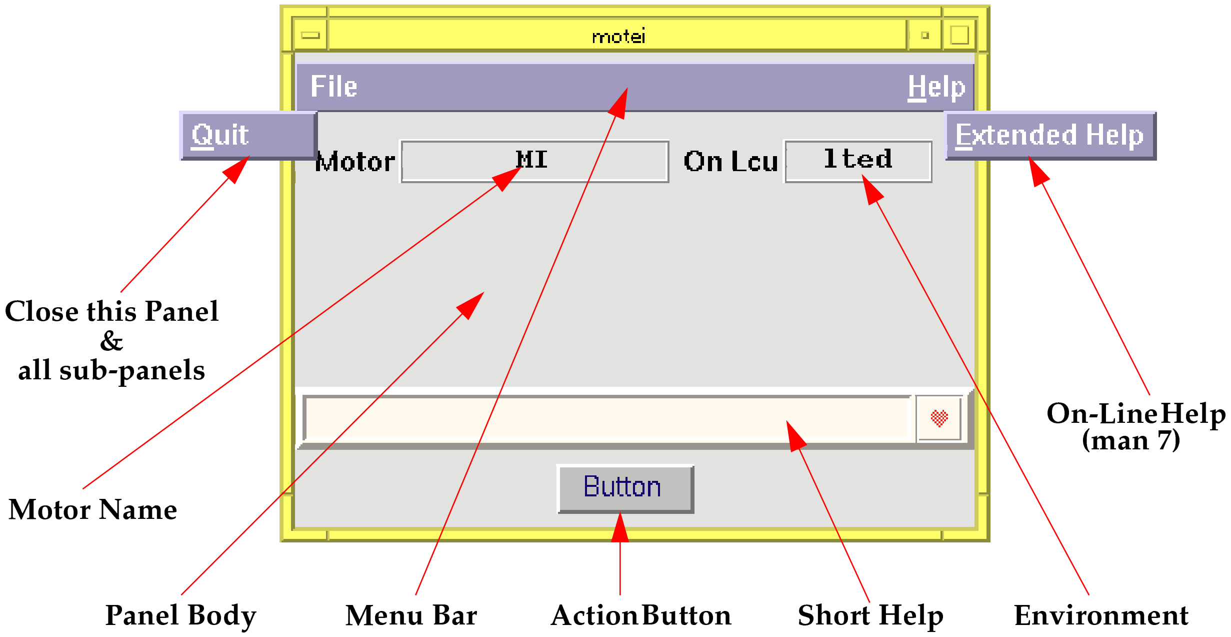
✗ **Refer please to Chapter 4, section 4.2 for the Database configuration.**

✗ **Refer please to Chapter 4, section 4.3 for the LCU configuration.**

Once the environment(s) have been created and started (see vccEnv(1) and vccConfigLcu(1)), you are ready to work with **motei**.

### 3.1.1 Panel Common Description

Each panel contains the items as shown below:



Each panel contains a menu bar, a short help field, action buttons and 2 output fields that show the selected motor and the working environment. The panel body contains all UIF elements needed by the dedicated function of the panel.

MenuBar 'File' :

Command 'Quit' : close this panel and all related sub-panels, i.e. panels invoked from this panel.

MenuBar 'Help' :

Command 'Extended Help' : show the associated help text, i.e. the man-page section 7 in a dedicated xterm.

**Note :** In the current version of the UIF toolkit, the time needed to build a panel can take a few 10 seconds depending on its size.

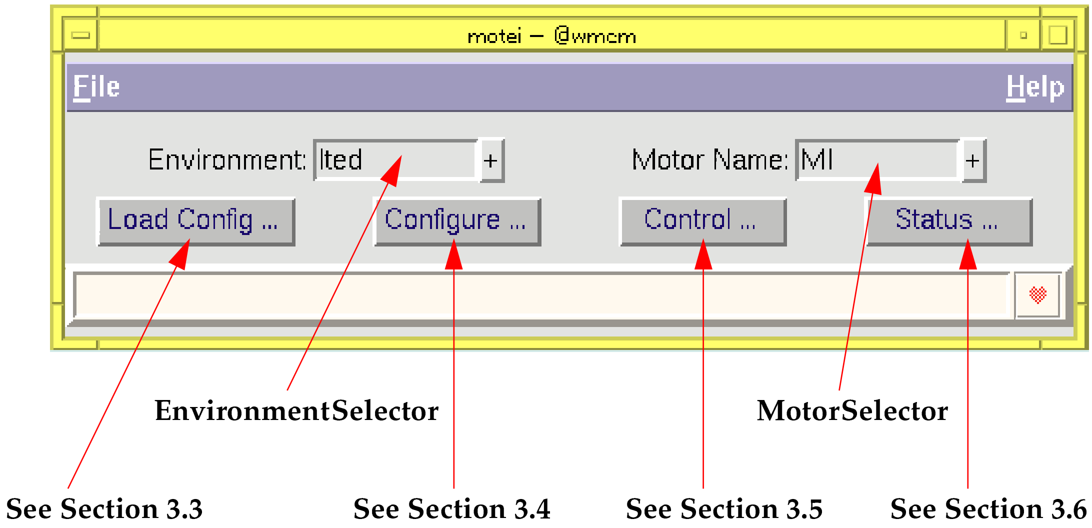
In the following sections, the environment will be referred to as **`<mcmEnv>`**, the motor as **`<motorName>`**.

## 3.2 Main Panel

The application **motei** (see 5.2.1 for man page) is invoked from the Unix shell command line by typing on the running environment **$RTAPENV** (CCS or CCS-lite) :

```
% motei
```

The main panel of **motei** is shown in the figure below and allows the selection of the working environment (LCU or WS) and of the Motor:



The **action buttons** located in the bottom part of the panel are:

- **Load Config...** to load the motor database configuration parameters from a backup file
- **Configure...** to configure motor and board parameters.
- **Control...** to control the motor.
- **Status...** to show the motor status.

### 3.2.1 Environment Selection

The environment selector consist of an entry and a button (**+**). The selection of the environment occurs either in typing in its name and **<Return>** or in invoking the pull-down menu via the **+** button and select the item. The environment defaults to **$RTAPENV**.

The environment may be preset using the environment variable **MOTENV**, e.g.

```
% setenv MOTENV wmcm
```

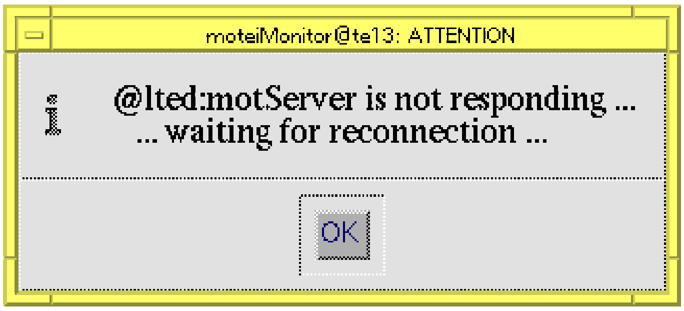
#### 3.2.1.1 Selecting WS Environments

It is also possible to prepare and browse motor configurations without a physical motor nor a LCU. The WS environment shall be of type **RTAP**, i.e. full CCS.

CCS-lite environment may only be used together with a LCU. The environment is only providing the communication interface between motei and the LCU, therefore it shall not be selected.

#### 3.2.1.2 Selecting LCU Environments

When selecting a LCU environment, an environment monitor is launched that checks every second if the environment is alive (**`PING`** command to **`motServer`**). If the LCU is not alive, the warning panel shown below pops up for acknowledge. The monitor is queried by the panels before closing.



@lted:motServer is not responding ...
... waiting for reconnection ...

<u>**Note:**</u> This feature is under development and will be enhanced so as to improve the reliability of the product when the LCU environment becomes not alive while operating a motor.

### 3.2.2 Motor Selection

The motor selector is enabled after an environment has been selected. This selector behaves identically to the environment selector. However, the list of available motors may not reflect the content of the database: since motor branches may be located anywhere in the database, there is no way a priori to retrieve all motor branches out of the only database. Two scenarii may take place:

1. the database (WS or LCU) includes the template motors as provided by moteiDATABASE.db(5). The motors are then located under **`:Appl_data:MOTEI`** for a WS, resp. under **`:MOTEI`** for a LCU environment. An automatic procedure will search for them and append them to the list.
2. the database (WS or LCU) does not include the template motors. The motors may thus be located anywhere. The list is empty and the user must type in the name of the selected motor.

On LCU environment only, a second procedure is invoked that searches for installed motors, see motInstall(1), and appends them to the list. When the template motors have been included, they are automatically installed. The selection of a motor implies its installation.

The motor may be preset using the environment variable **`MOTNAME`**, e.g.

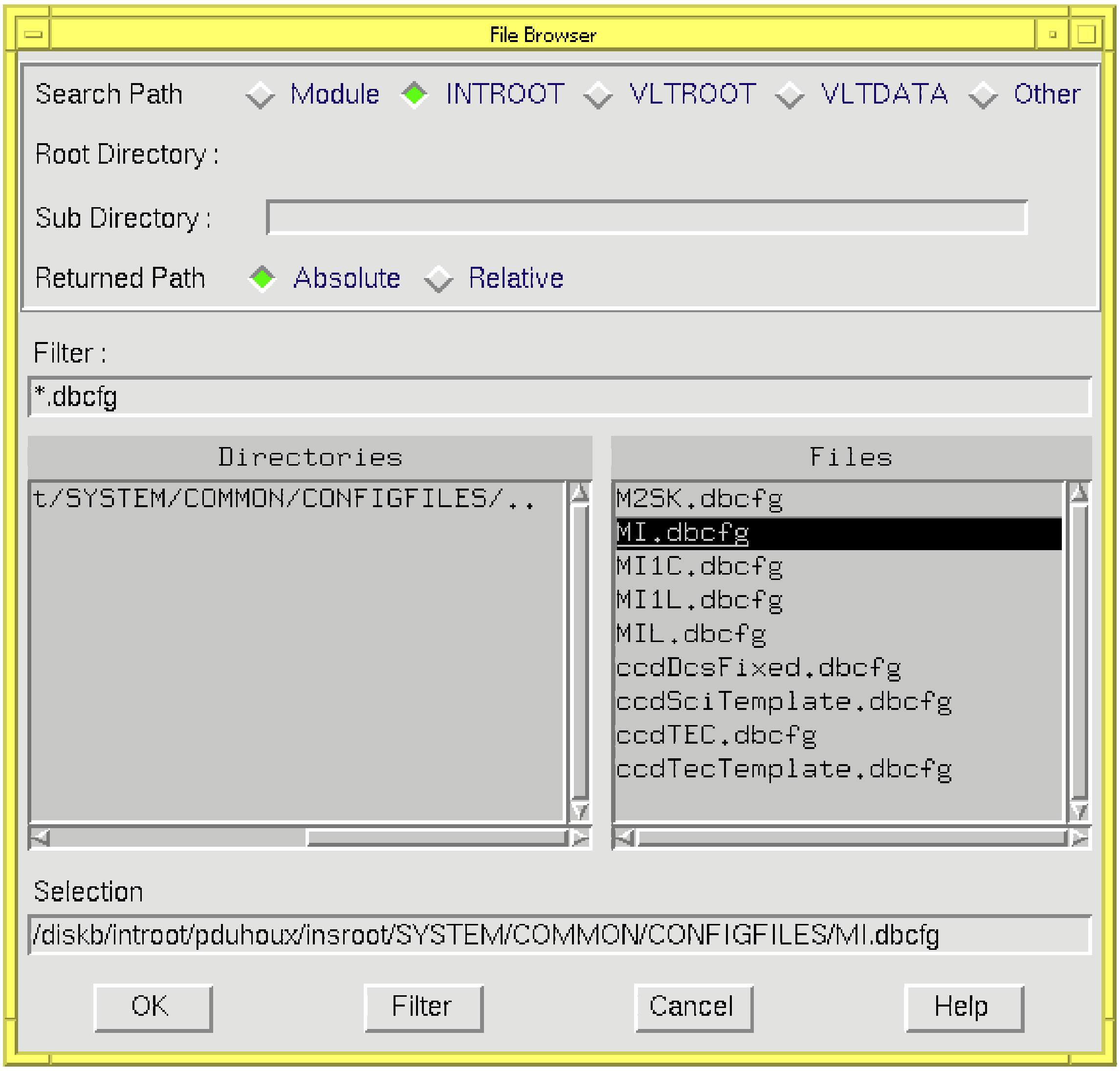
```
% setenv MOTNAME MI
```

## 3.3 Load & Save a Motor Configuration

### 3.3.1 Load a Configuration

The **Load Config ...** button of the Main panel allows to load the database branch of the selected motor with the parameters stored in a configuration file.

A *File selection Box* is displayed allowing the selection of the name of the configuration file.



File Browser

Search Path ◇ Module ◆ INTROOT ◇ VLTROOT ◇ VLTDATA ◇ Other

Root Directory :

Sub Directory :

Returned Path ◆ Absolute ◇ Relative

Filter :

*.dbcfg

Directories

t/SYSTEM/COMMON/CONFIGFILES/..

Files

M2SK.dbcfg
MI.dbcfg
MI1C.dbcfg
MI1L.dbcfg
MIL.dbcfg
ccdDcsFixed.dbcfg
ccdSciTemplate.dbcfg
ccdTEC.dbcfg
ccdTecTemplate.dbcfg

Selection

/diskb/introot/pduhoux/insroot/SYSTEM/COMMON/CONFIGFILES/MI.dbcfg

It is recommended to name these files as **`<motorName>.dbcfg`** and to locate them either:

- under the current application path **`<appl>/config`** for later archiving

or

- under **`$INS_ROOT/SYSTEM/COMMON/CONFIGFILES`** where all **`.dbcfg`** files are searched for (see motInitDb(1)).

### 3.3.2 Save the Configuration

The main configuration panel (see moteiConfig(7)) provides one additional button **`Save`** that is the combination of **`Apply`** + **`Save`**. In the same way as for **`Load Config...`**, a file selector is raised. The generated file should have the extension **`.dbcfg`** for compatibility and will contains the values of all configuration attributes as expected by motInitDb(1), motInstall(1):

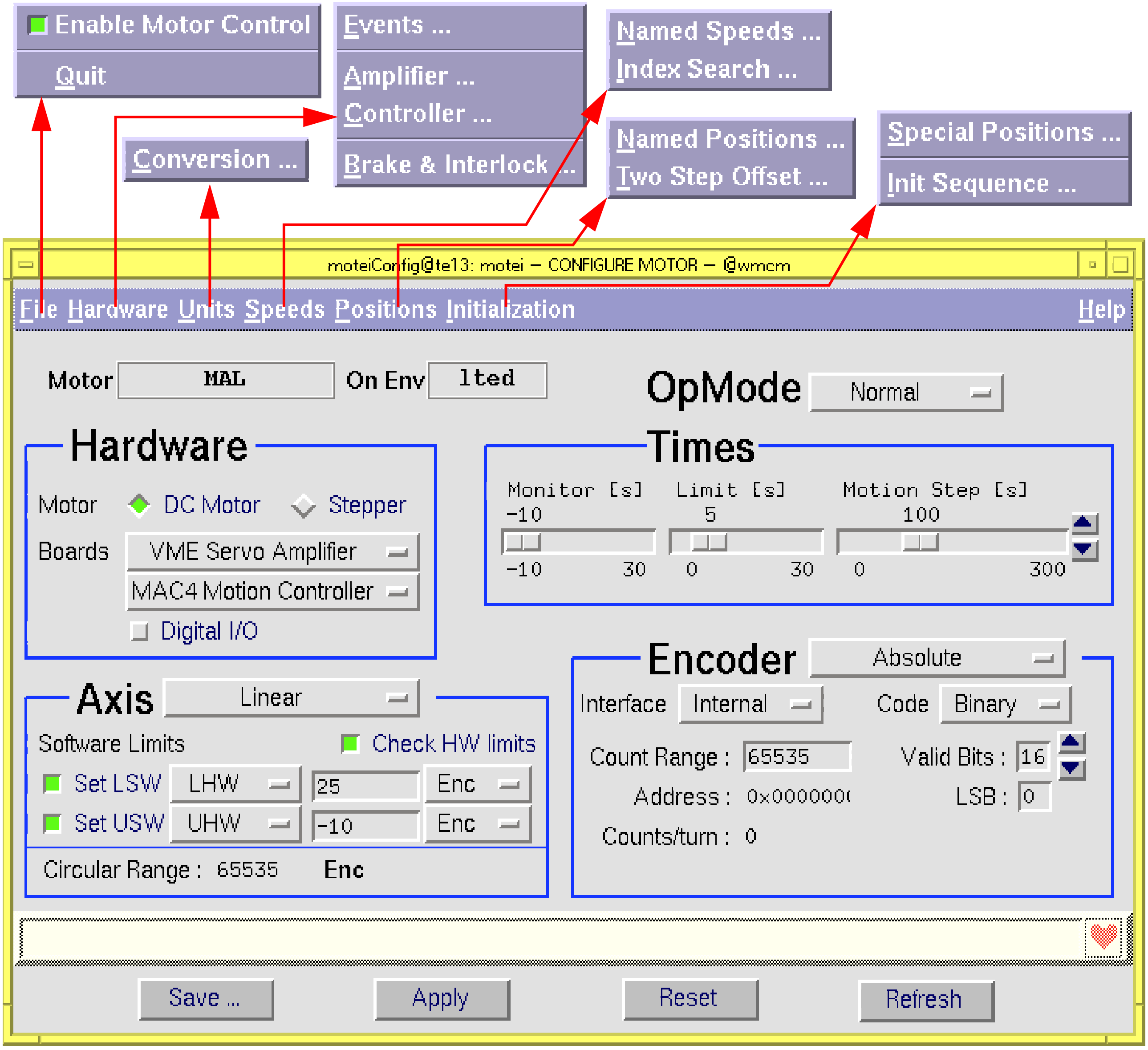
✗ **Each time the LCU is rebooted, any change in the configuration is lost: it is therefore recommended to save the actual configuration in a backup file after each major change. After the LCU is rebooted, the relevant configuration must be loaded in the database.**

## 3.4 Configure the Motor

The Configuration Panel is raised after clicking on the **Configure...** button of the Main panel.

✗**This is the first operation to be done before starting to use a motor**

The panel is shown in the figure below; Menu bar options are inset above the panel. See 5.2.2 for man page.



Motor MAL On Env lted

OpMode Normal

Hardware

Motor ● DC Motor ○ Stepper

Boards VME Servo Amplifier

MAC4 Motion Controller

□ Digital I/O

Times

Monitor [s] -10 (-10 to 30)

Limit [s] 5 (0 to 30)

Motion Step [s] 100 (0 to 300)

Axis Linear

Software Limits ☑ Check HW limits

☑ Set LSW LHW 25 Enc

☑ Set USW UHW -10 Enc

Circular Range : 65535 Enc

Encoder Absolute

Interface Internal Code Binary

Count Range : 65535 Valid Bits : 16

Address : 0x000000( LSB : 0

Counts/turn : 0

The panel is to be read from the part **Hardware**, down to **Axis**, **Encoder** and **Times**. According to the current settings, some items may not be enabled, some values may be modified as a consequence of a previous setting.

The **Hardware** part is dedicated to the description of the hardware: types of motor and boards.

The **Axis** part is dedicated to the description of the axis: type (linear/circular) and SW limits.

For linear axis with motion controller, the SW limits may be defined absolute or relative to Special positions (see 3.4.11). The limits are set at completion of the HW Initialization (see 3.4.12). In addition, the HW limits may be checked on request for simultaneous activation indicating a possible HW problem with the motor cables.

For circular axis, the range must be set strictly positive.

The **Encoder** part is dedicated to the description of the encoder: type (incremental/absolute) and associated parameters.

The **Times** part is dedicated to the setting of the status sampling period and timeouts.

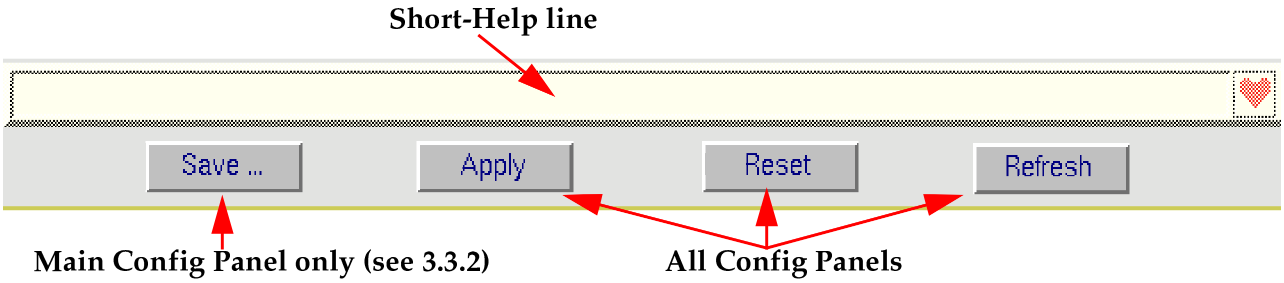
In most cases the setting of these configuration items from one of the template motor allows the motor to become initialized and may be sufficient to perform some motions. Hardware tuning can be achieved via the MenuBar Hardware; advanced configuration is made via the other Menubar entries.

✗In the current version of **motei**, any change in the description of the axis and/or encoder must be propagated to the motion controller MAC4, by invoking the associated panel (Menubar ***Hardware*** option ***Controller***) and **Apply**. When both panels are active, one must first **Apply** in the Main Configuration panel and then **Refresh** + **Apply** in the Motion Controller MAC4 panel.

### 3.4.1 Common panel items

#### 3.4.1.1 Action Buttons

Each configuration panel provides the 3 buttons as shown below:



The actions may be invoked at any time from any panel and apply only on the values as available from this panel:

**Apply** write the configuration values of this panel into the corresponding attributes of the database. Depending on the menubar ***File*** option ***Enable Motor Control*** (in main configuration panel), the **Apply** action will instruct MCM to update the configuration; the motor operational state is then reset to **NOT INIT**.

**Reset** restore the configuration as when the panel was invoked. The restored configuration is then Applied.

**Refresh** the panel by reading the configuration from the database, but the current attributes. It is recommended especially when units, axis or HW configuration have been changed.

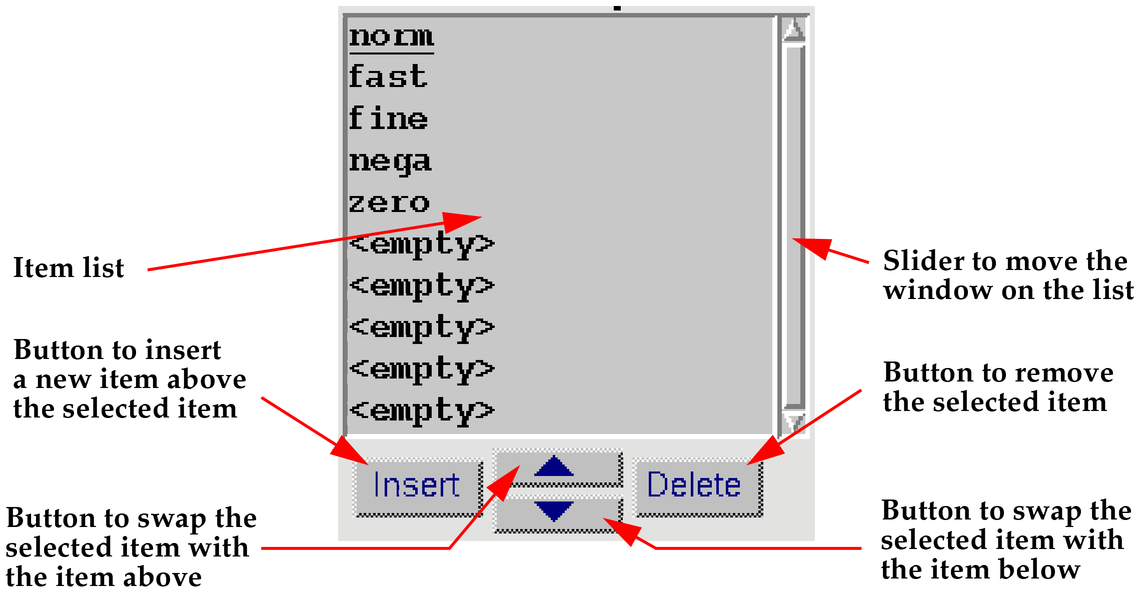
Any change in the configuration panels is not considered by MCM until **Apply** or **Save** is invoked.

✗**When Quitting the panel, not Applied values are lost.**

✗**The last Applied configuration is valid until a new configuration is loaded into the database or the LCU is rebooted.**

### 3.4.1.2 List Boxes

Some of the sub-panels (see 3.4.6, 3.4.7, 3.4.9 and 3.4.12) are built around List Boxes. Below the item list, 4 buttons are provided as shown below:


norm
fast
fine
nega
zero
<empty>
<empty>
<empty>
<empty>
<empty>
Insert
Delete
Item list
Button to insert a new item above the selected item
Button to swap the selected item with the item above
Slider to move the window on the list
Button to remove the selected item
Button to swap the selected item with the item below

The selection of an item is made via the Mouse Button 1 (left).

The **`Insert`** button inserts an empty item above the selected one, but for the first item.

The new item is preset to meaningful values.

The **`Arrow Up`** swaps the selected item with the one above, but for the first item.

The **`Arrow Down`** swaps the selected item with the one below, but for the last item.

The **`Delete`** button removes the selected item.

The selected item is automatically updated with the current values when invoking **`Insert`**/**`Arrow Up`**/**`Arrow Down`**, **`Apply`** (**`Save`**) or selecting an item.

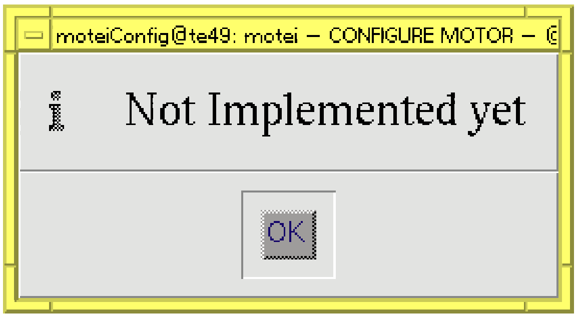
Depending on the panel, some additional restrictions may apply that will be mentioned in the corresponding sections.

**Note :** This kind of selection widget is used for database attributes of type TABLE. The table contains physically a predefined number of entries. It is therefore obvious that this number can not be modified neither with Insert nor with Delete. This means that the insertion is only possible when at least one empty entry is available; the deletion removes the selected entry and appends an empty one at the end of the list.

### 3.4.2 Menu Hardware : Events ...

These features are not implemented yet, however the default configuration as preset in the base motor classes is coherent and does not prevent the operation of the motor.

The dialog box window as shown below is raised; Click on **OK** to acknowledge.



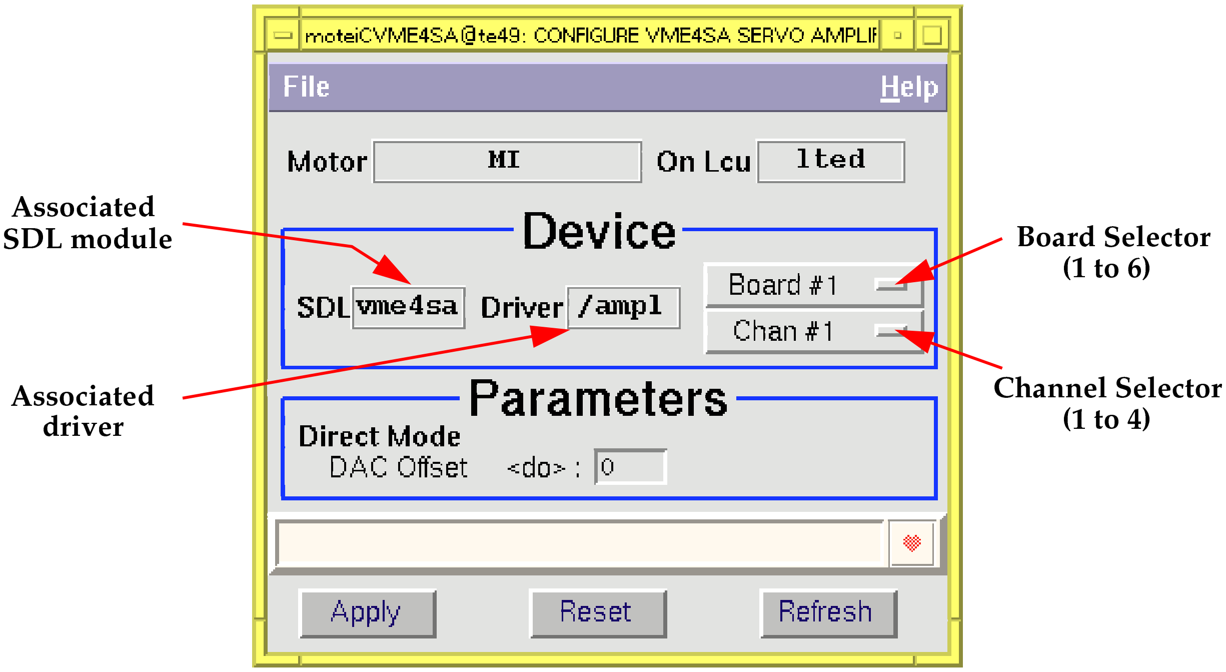
Not Implemented yet

### 3.4.3 Menu Hardware : Amplifier ...

#### 3.4.3.1 Servo Amplifier VME4SA

This panel is only available when using the standard ESO Servo Amplifier VME4SA, see 5.2.3 for man page.

A detailed description of the board parameters is given in [2] and in the VME4SA User Manual.



moteiCVME4SA@te49: CONFIGURE VME4SA SERVO AMPLIF

Motor MI On Lcu lted

Device

SDL vme4sa Driver /ampl

Board #1

Chan #1

Parameters

Direct Mode

DAC Offset <do> : 0

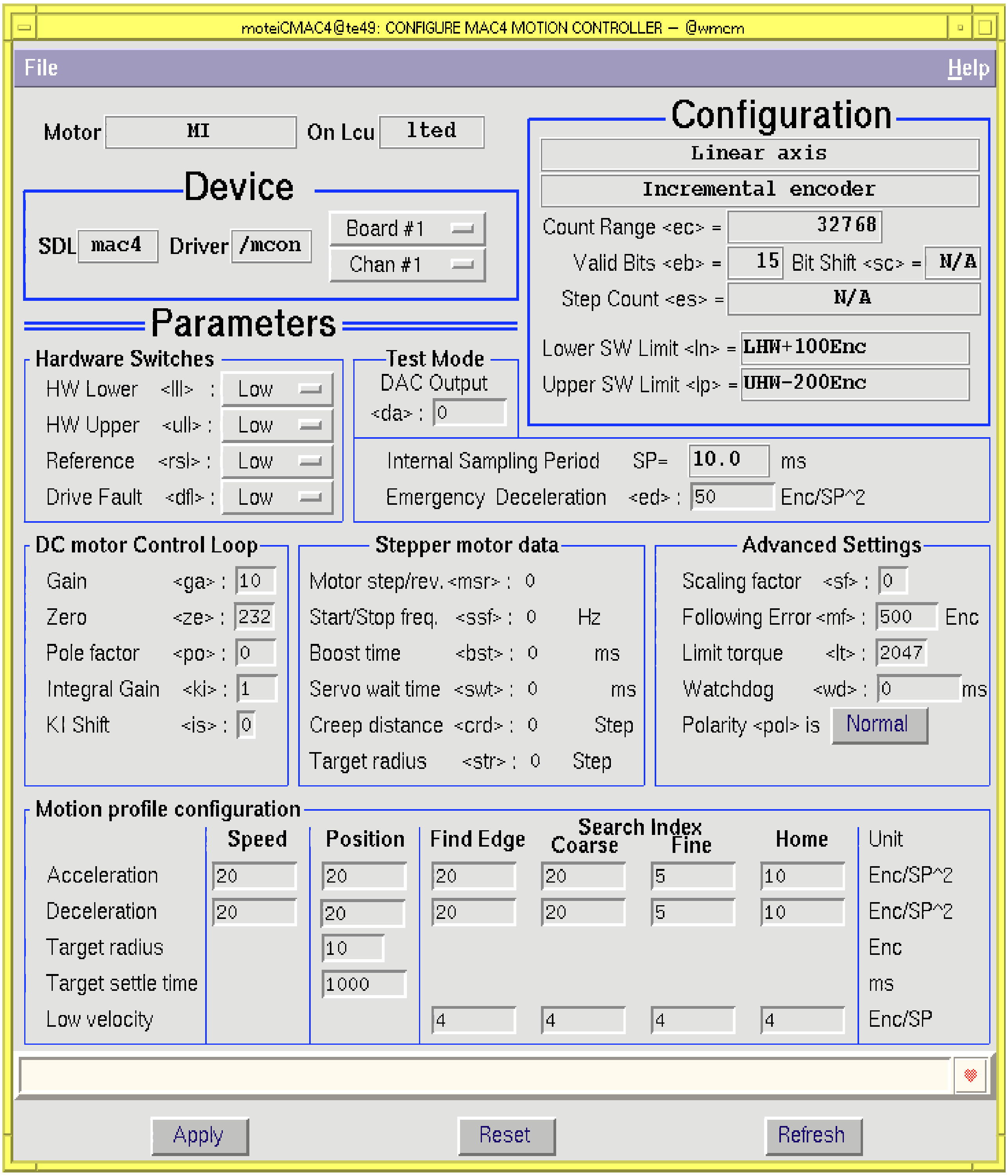
The **Device** part allows the setting of the associated hardware channel. The board selector refers to the board index with increasing base address; the channel selector refers to the outputs as labelled on the backplane and on the board front panel.

### 3.4.4 Menu Hardware : Controller ...

#### 3.4.4.1 Motion Controller Maccon MAC4

This panel is only available when using the standard ESO Motion Controller Maccon MAC4, see 5.2.4 for man page.

A description of the board parameters is given in [2] and in the MACCON MAC4 User Manual.



moteiCMAC4@te49: CONFIGURE MAC4 MOTION CONTROLLER – @wmcm

Motor MI On Lcu lted

**Device**

SDL mac4 Driver /mcon Board #1 Chan #1

**Configuration**

Linear axis

Incremental encoder

Count Range <ec> = 32768

Valid Bits <eb> = 15 Bit Shift <sc> = N/A

Step Count <es> = N/A

Lower SW Limit <ln> = LHW+100Enc

Upper SW Limit <lp> = UHW-200Enc

**Parameters**

Hardware Switches

| | | |
|---|---|---|
| HW Lower | <ll> : | Low |
| HW Upper | <ul> : | Low |
| Reference | <rsl> : | Low |
| Drive Fault | <dfl> : | Low |

Test Mode
DAC Output
<da> : 0

Internal Sampling Period SP= 10.0 ms

Emergency Deceleration <ed> : 50 Enc/SP^2

DC motor Control Loop

| | | |
|---|---|---|
| Gain | <ga> : | 10 |
| Zero | <ze> : | 232 |
| Pole factor | <po> : | 0 |
| Integral Gain | <ki> : | 1 |
| KI Shift | <is> : | 0 |

Stepper motor data

| | | | |
|---|---|---|---|
| Motor step/rev. | <msr> : | 0 | |
| Start/Stop freq. | <ssf> : | 0 | Hz |
| Boost time | <bst> : | 0 | ms |
| Servo wait time | <swt> : | 0 | ms |
| Creep distance | <crd> : | 0 | Step |
| Target radius | <str> : | 0 | Step |

Advanced Settings

| | | | |
|---|---|---|---|
| Scaling factor | <sf> : | 0 | |
| Following Error | <mf> : | 500 | Enc |
| Limit torque | <lt> : | 2047 | |
| Watchdog | <wd> : | 0 | ms |
| Polarity <pol> is | | Normal | |

Motion profile configuration

| | Speed | Position | Find Edge | Search Index Coarse | Search Index Fine | Home | Unit |
|---|---|---|---|---|---|---|---|
| Acceleration | 20 | 20 | 20 | 20 | 5 | 10 | Enc/SP^2 |
| Deceleration | 20 | 20 | 20 | 20 | 5 | 10 | Enc/SP^2 |
| Target radius | | 10 | | | | | Enc |
| Target settle time | | 1000 | | | | | ms |
| Low velocity | | | 4 | 4 | 4 | 4 | Enc/SP |

The **`Device`** part allows the setting of the associated hardware channel. The board selector refers to the board index with increasing base address; the channel selector refers to the outputs as labelled on the backplane and on the board front panel.

The **`Configuration`** part recalls the configuration as defined in the Main Configuration panel, (see moteiConfig(7)). It shows the current configuration values for the **ESO Standard MAC4** controller board and allows to change or tune any parameter handled by the board. The label preceding each input field shows both the *human-readable* and, (between **`< >`**) the *internal-board* parameter name.

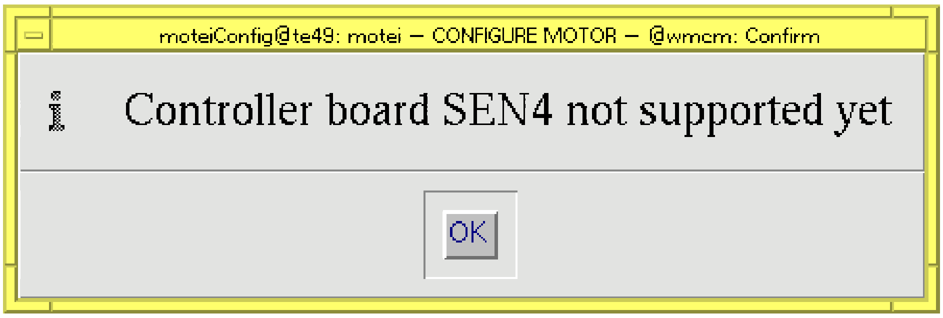
A description of the meaning of each board parameter is given in [2].

✗In the current version of **`motei`**, any change in the description of the axis and/or encoder must be propagated to the motion controller MAC4, by invoking the associated panel (Menubar ***Hardware*** option ***Controller***) and **`Apply`**. When both panels are active, one must first **`Apply`** in the Main Configuration panel and then **`Refresh`** + **`Apply`** in the Motion Controller MAC4 panel.

#### 3.4.4.2 Other Motion Controllers

The motion controller CAMAC (used on NTT-EMMI) is not supported.

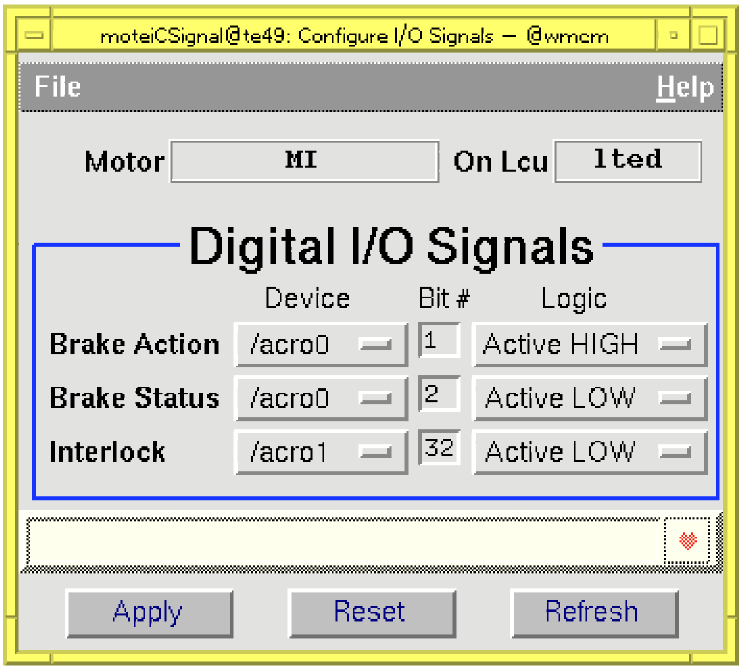
The dialog box window as shown below is raised; Click on **`OK`** to acknowledge.



Controller board SEN4 not supported yet

### 3.4.5 Menu Hardware : Brake & Interlock ...

This panel is available only when a Digital I/O board has been declared in the HW configuration, see 5.2.5 for man page..



moteiCSignal@te49: Configure I/O Signals — @wmcm

Motor MI On Lcu lted

Digital I/O Signals

| | Device | Bit # | Logic |
|---|---|---|---|
| Brake Action | /acro0 | 1 | Active HIGH |
| Brake Status | /acro0 | 2 | Active LOW |
| Interlock | /acro1 | 32 | Active LOW |

For each of the three signals, the associated device (**`/acro0`** to **`/acro3`**) shall be selected.

Each signal is mapped by one (1) bit which shall be input (from 1 to 64).

The signal **`Brake Action`** is an Output; the other two are Inputs (status lines).

At last, the logic of the signals shall be specified :

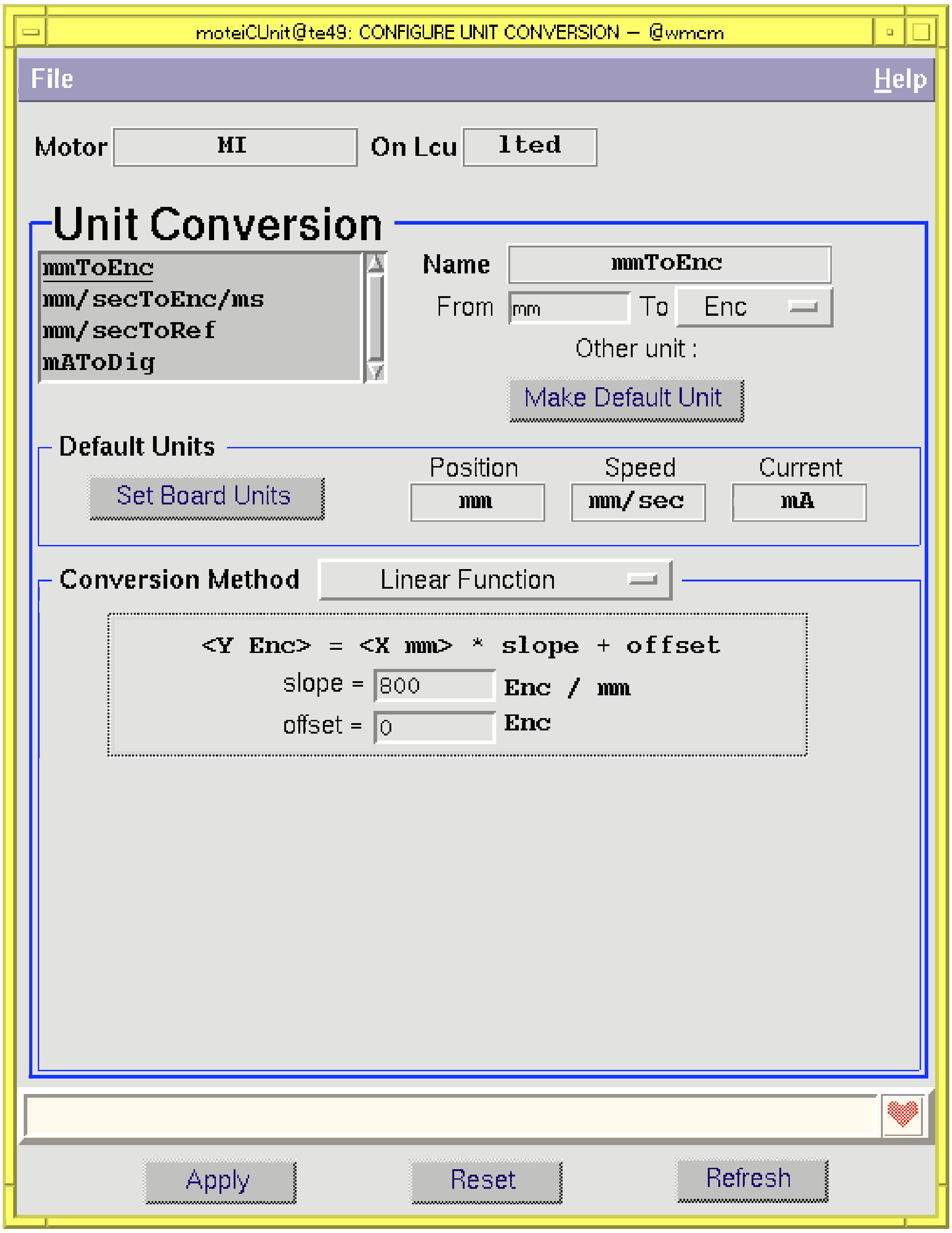
| | |
|---|---|
| **`OFF`**: | the signal is not used |
| **`Active LOW`**: | negative logic |
| **`Active HIGH`**: | positive logic |

### 3.4.6 Menu Units : Conversion ...

One of the important features of MCM is the handling ability of units (See [2]). This panel is dedicated to the configuration of the unit conversion methods to be used by MCM and the application.

Beside the 4 board units (**Enc**, **Enc/ms**, **Ref** and **Dig**) for Position, Velocity and Current, it is possible to define user units to be used as default units, i.e. the unit in which position and velocity may be defined, and in which position, velocity and current actual values may be expressed in the status.

For each user unit declared as default unit, the associated conversion method to one of the board units must be provided.



moteiCUnit@te49: CONFIGURE UNIT CONVERSION – @wmcm

Motor MI On Lcu lted

Unit Conversion

```
mmToEnc
mm/secToEnc/ms
mm/secToRef
mAToDig
```

Name mmToEnc

From mm To Enc

Other unit :

Default Units

| Position | Speed | Current |
|---|---|---|
| mm | mm/sec | mA |

Conversion Method Linear Function

```
<Y Enc> = <X mm> * slope + offset
```

slope = 800 Enc / mm

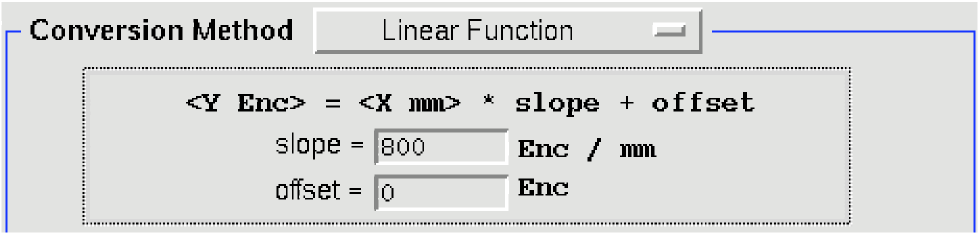
offset = 0 Enc

The panel shown above displays the default set of units and the table of unit conversion methods of the selected motor, see 5.2.6 for man page. The listBox displays the names of the conversion method or **`<empty>`** if the row is available. Whenever a method is selected (by clicking on it with the mouse) the panel is updated with the source unit, target unit, conversion method and its related parameters as from the database.

It is possible to set a unit as the default for *Speed*, *Position* or *Current* (See [2] for the requirements) by invoking the **`Make Default`** button.

The general procedure to define/modify a unit conversion method is:

1. Using the mouse, select an entry from the listbox.
2. Edit the **`From`** entry. It is the name of the source unit.
3. Select the **`To`** unit from the predefined set of units or select **`Other`** to define another target unit, in the later case type in the entry below the **`To`** selector the name of the target unit. This conversion method can however not be used by MCM.
4. Define the conversion method to apply (Linear, Interpolation or External function)
5. Define the parameters required for the conversion method selected.
   a. For linear conversion method, enter the offset and slope.



<Y Enc> = <X mm> * slope + offset

slope = 800 Enc / mm

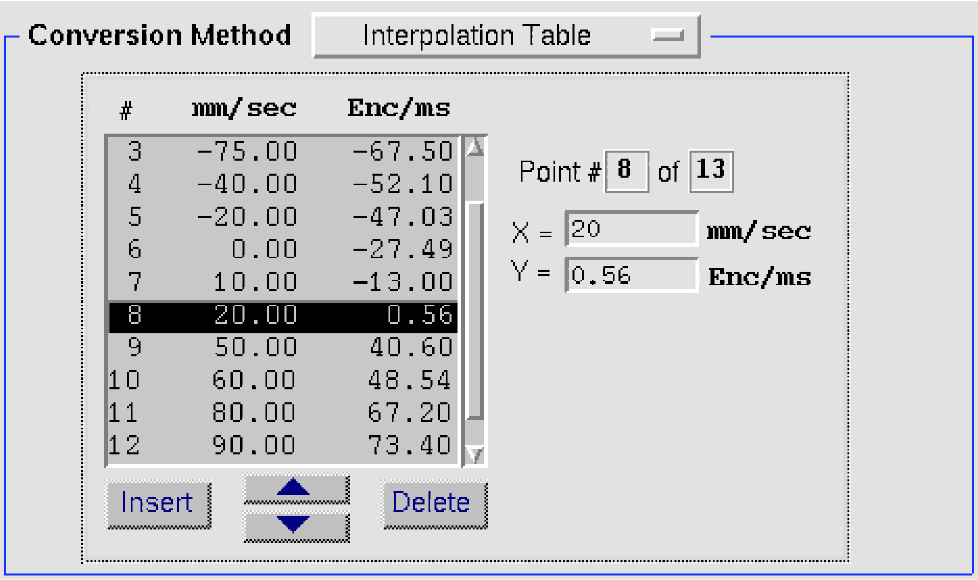
offset = 0 Enc

The slope must verify:

$$|slope| > 10^{-38}$$

   b. For interpolation method: minimum 2 up to 50 points shall be defined. The set of interpolation points must cover the whole variation domains in X and Y. The function must be monotonous. The points shall be entered either in X or Y increasing/decreasing order (use the arrows to sort the list). Add/remove points using the **`Insert`**/**`Delete`** buttons. The list is automatically updated with the current point (X,Y) when invoking the buttons **`Insert`**/**`Apply`** and the arrows.

.



Conversion Method: Interpolation Table

| # | mm/sec | Enc/ms |
|---|---|---|
| 3 | -75.00 | -67.50 |
| 4 | -40.00 | -52.10 |
| 5 | -20.00 | -47.03 |
| 6 | 0.00 | -27.49 |
| 7 | 10.00 | -13.00 |
| 8 | 20.00 | 0.56 |
| 9 | 50.00 | 40.60 |
| 10 | 60.00 | 48.54 |
| 11 | 80.00 | 67.20 |
| 12 | 90.00 | 73.40 |

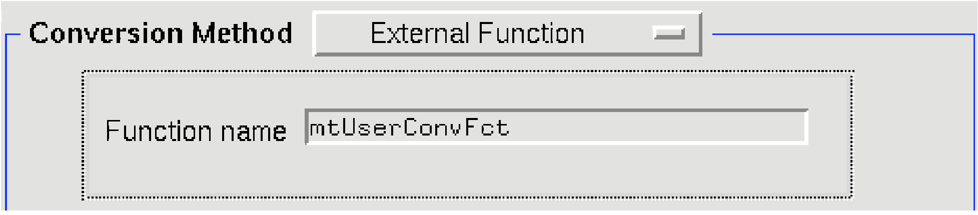
Point # 8 of 13

X = 20 mm/sec

Y = 0.56 Enc/ms

**Minimum 2 points, function must be monotonous**
**Edges of conversion domain must be included**

c. For user function, enter the name of the user function. The name must be found in the Global Symbol table of the LCU and must comply the type **`motCONV_FUNC`** (defined in **`mot.h`**). The names are resolved at Install time, see motInstall(1).



Conversion Method: External Function

Function name: mtUserConvFct

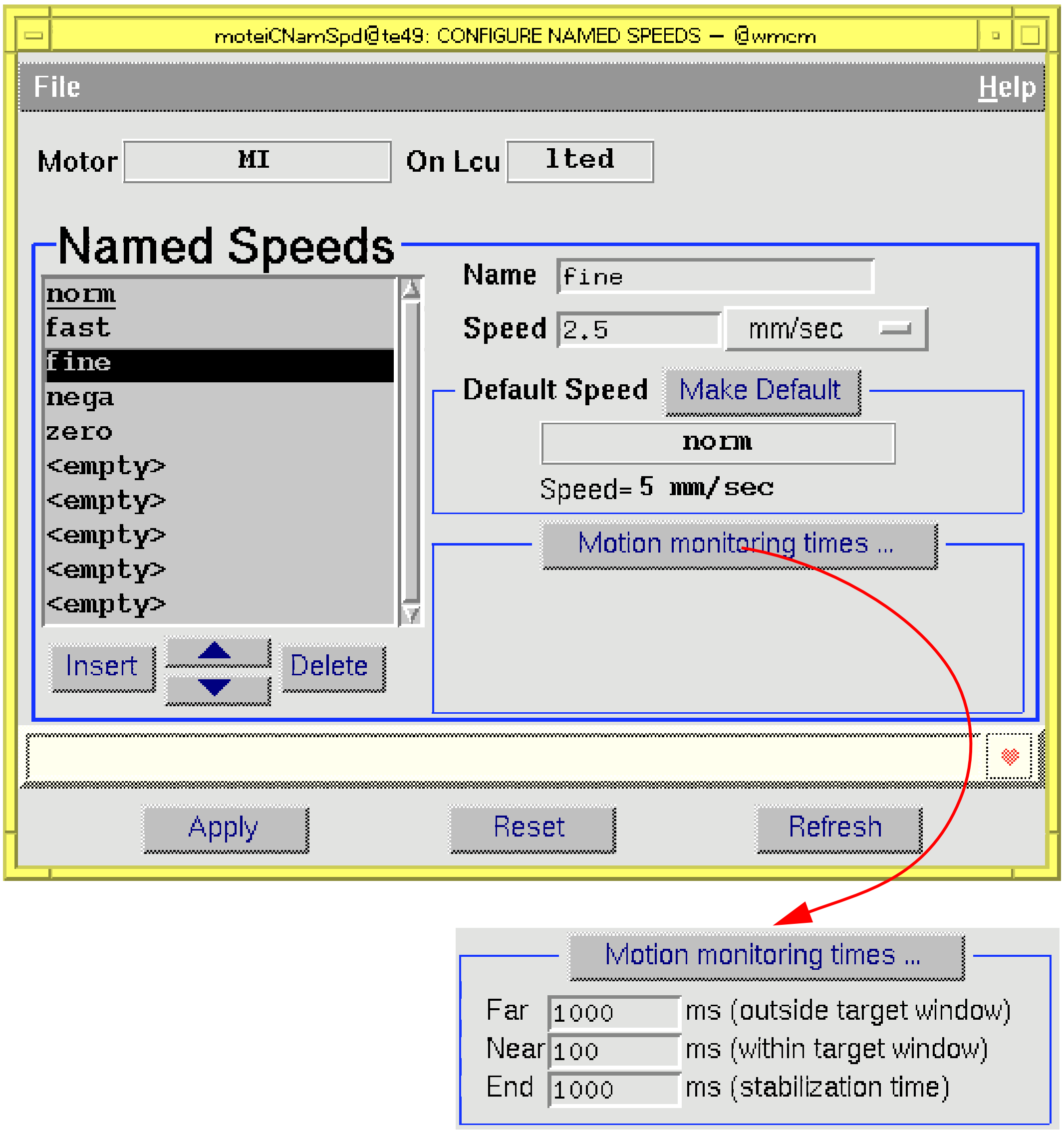
The unit conversion description is applied automatically when selecting an other method or invoking the **`Apply`** button.

<u>**Important Note:**</u> the definition of user units as default units enforces the independence of the configuration with the hardware, i.e. in case the encoder shall be replaced and the gear ratio is modified, only the update of the corresponding conversion methods is necessary so as to update the whole configuration.

✗However, the declaration of user units as default units and their use in the other configuration items forces the user not to change them. Changing the name of a user unit must be propagated through all other configuration items whenever referenced. Automatically, the unit will be forced to the corresponding board unit if the unit was not declared.

### 3.4.7 Menu Speeds : Named Speeds ...

Named Speeds are symbolic names assigned to physical velocities. The panel below shows the table of named speeds, see 5.2.7 for man page.


moteiCNamSpd@te49: CONFIGURE NAMED SPEEDS — @wmcm

Motor MI On Lcu lted

Named Speeds

- norm
- fast
- fine
- nega
- zero
- <empty>
- <empty>
- <empty>
- <empty>
- <empty>

Name fine

Speed 2.5 mm/sec

Default Speed norm

Speed= 5 mm/sec

Motion monitoring times ...

Far 1000 ms (outside target window)

Near 100 ms (within target window)

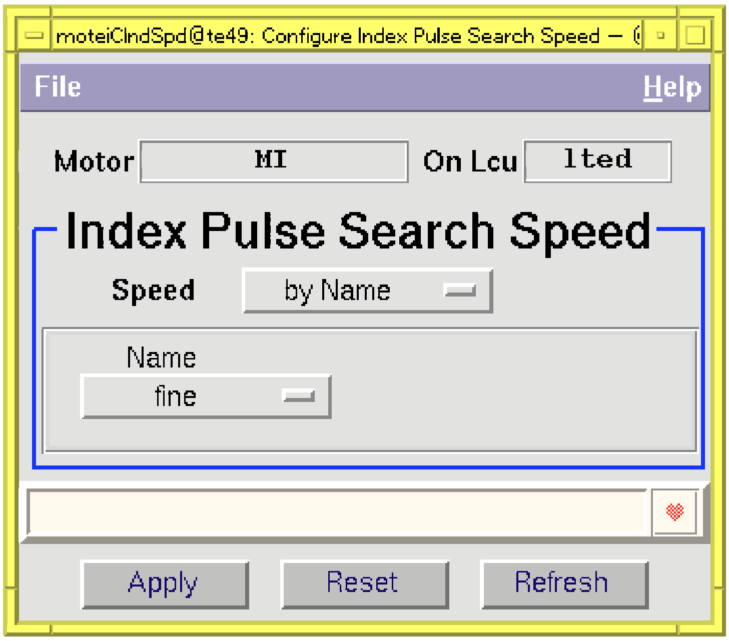
End 1000 ms (stabilization time)

The ListBox shows the list of all defined Named Speeds or **`<empty>`** if the row is available. Whenever an entry is selected (by clicking on it with the mouse), the panel is updated with the speed name and its parameters. The default speed is also displayed and it is possible to set its value by pressing the **`Make Default`** button, the settings currently present in the panel will be used.

<u>**Note**</u>: the default speed is recorded as the index of the associated speed in the table.

In addition, for each speed the motion monitoring parameters may be specified by clicking the button **`Motion monitoring times ...`** These times are defaulted to 1000, 200 and 1000ms resp. See [2] for details.

### 3.4.8 Menu Speeds : Index Search ...



moteiCIndSpd@te49: Configure Index Pulse Search Speed –

Motor: MI
On Lcu: lted

Index Pulse Search Speed

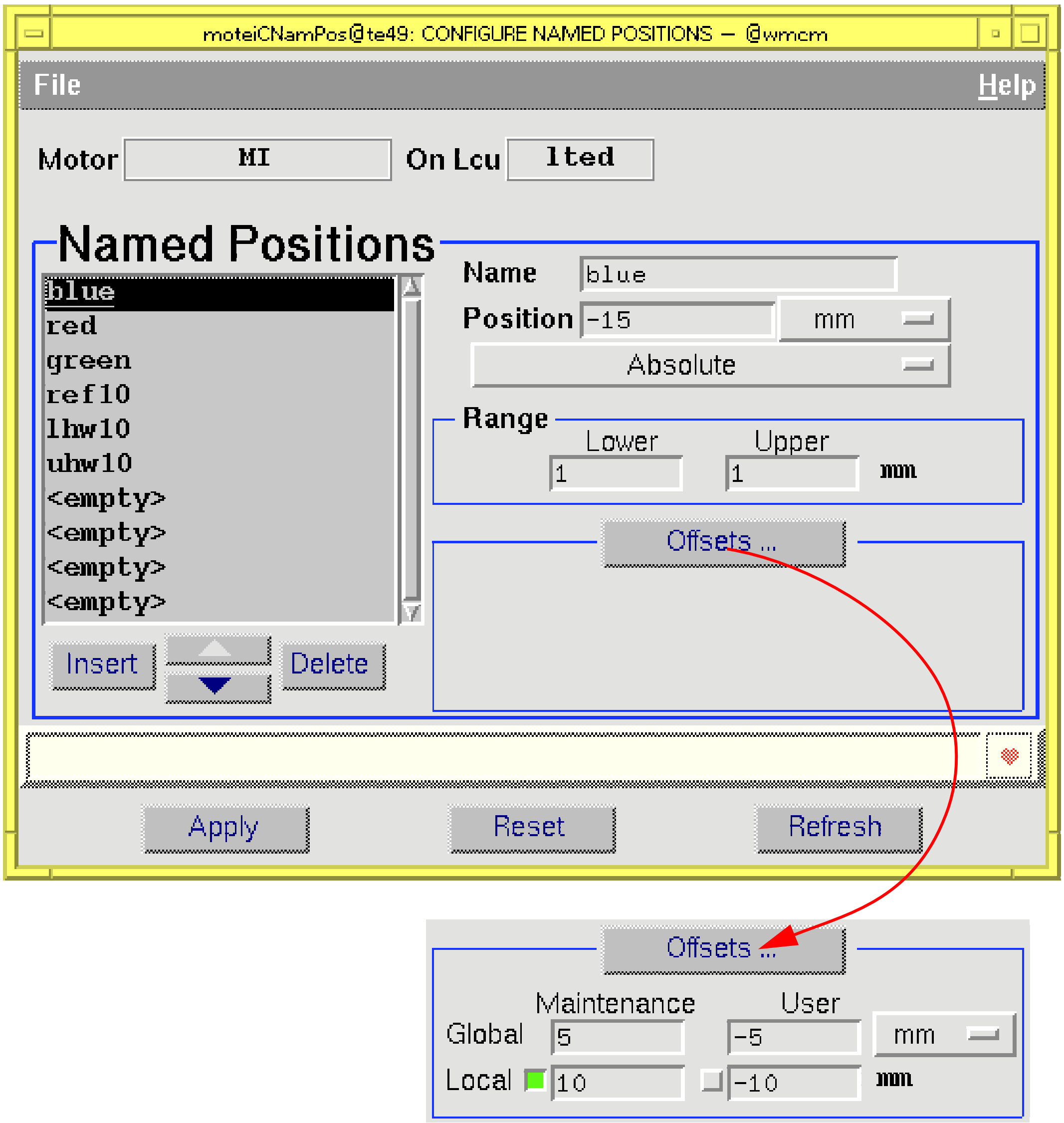
Speed: by Name

Name: fine

This panel is dedicated to the definition of the speed to be applied when performing the second step of the search motion to the index pulse of the encoder. See 5.2.8 for man page.

### 3.4.9 Menu Positions : Named Positions ...

Named Positions are symbolic names assigned to physical positions. The panel below shows the table of named positions, see 5.2.9 for man page.



moteiCNamPos@te49: CONFIGURE NAMED POSITIONS – @wmcm

Motor: MI  On Lcu: lted

Named Positions

- blue
- red
- green
- ref10
- lhw10
- uhw10
- <empty>
- <empty>
- <empty>
- <empty>

Name: blue

Position: -15 mm

Absolute

Range

| | Lower | Upper | |
|---|---|---|---|
| | 1 | 1 | mm |

Offsets ...

| | Maintenance | | User | |
|---|---|---|---|---|
| Global | 5 | | -5 | mm |
| Local | 🟩 10 | □ | -10 | mm |

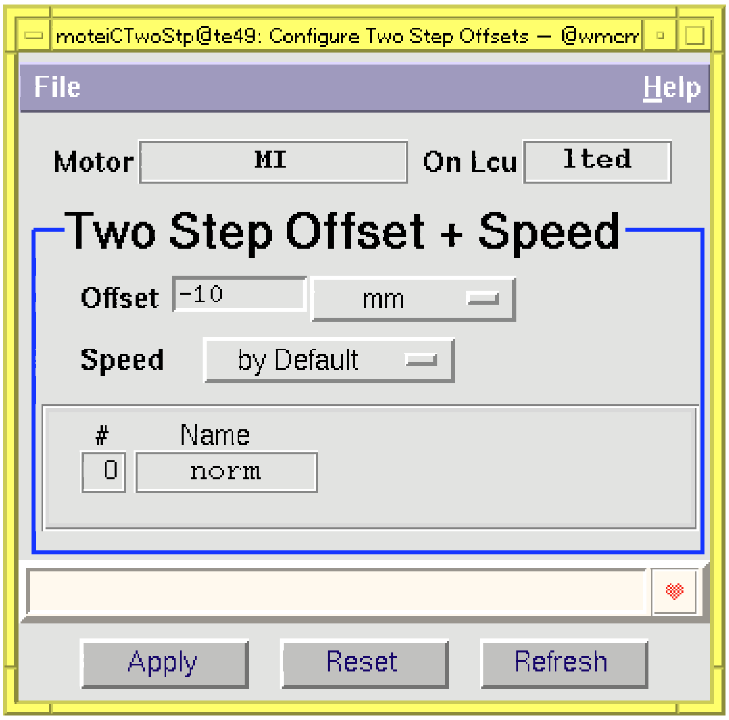
The ListBox shows the list of all defined Named Positions or **`<empty>`** if the row is available. Whenever an entry is selected (by clicking on it with the mouse), the panel is updated with the position name and its parameters. Each named position shall be described as a Nominal position and the two Lower and Upper range values that define the domain in which the name shall be valid.

A named positions shall be defined either as an Absolute position or relative to one of the Special Positions (see 3.4.11). The named positions must be fully resolved at completion of the HW Initialization (see 3.4.12).

In addition, offsets may be specified for the sections Maintenance and User-defined as described in [2], by invoking the **`Offsets...`** button. By default, both offsets are off, i.e. the global offsets are valid. The small inset panel allows the setting of both the global and the local offsets: the Global offsets are defined for all named positions, the Local offsets override the latter whenever enabled.

### 3.4.10 Menu Positions : Two Step Offset ...

This panel is dedicated to the description of the Offset and Speed to be applied for Two-Step type motions, see 5.2.10 for man-page:


moteiCTwoStp@te49: Configure Two Step Offsets – @wmcm

Motor MI On Lcu lted

Two Step Offset + Speed

Offset -10 mm

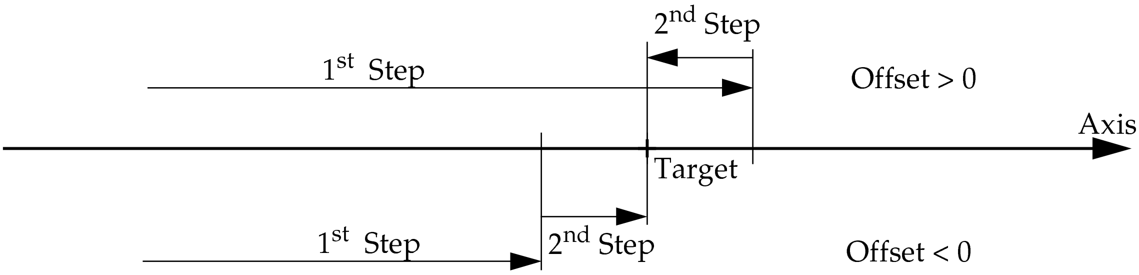
Speed by Default

| # | Name |
|---|---|
| 0 | norm |

The offset is a relative value which will be added to the target position value so as to define the position to move to during the first step:

offset > 0 : the first motion will end *after* the target, i.e. the target will always be reached from the *right* side (decreasing position).

offset < 0 : the first motion will end *before* the target, i.e. the target will always be reached from the *left* side (increasing position).

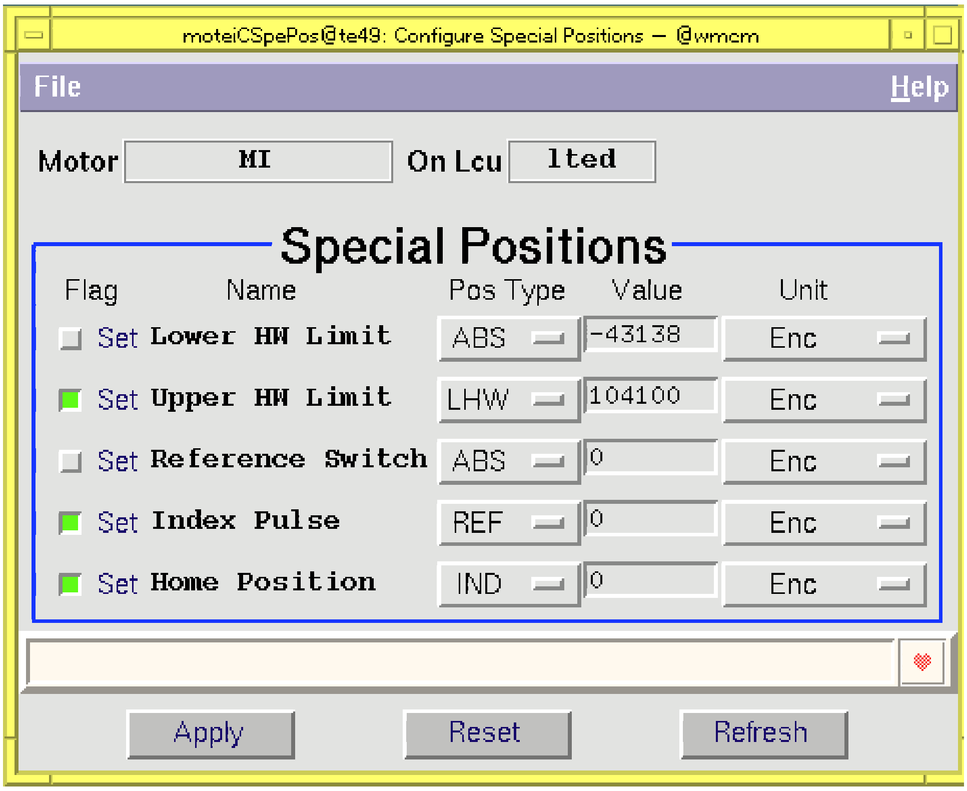


This type of motion is meant to correct for eventual mechanical backlash, and thus increase the precision of the positioning.

The speed to be applied during the second step shall not be too high so as to limit possible oscillations and achieve a better positioning accuracy.

### 3.4.11 Menu Initialization : Special Positions ...

The panel below is dedicated to the description of the Hardware special positions, see 5.2.11 for man-page:



Motor MI On Lcu lted

Special Positions

| Flag | Name | Pos Type | Value | Unit |
|---|---|---|---|---|
| □ | Set Lower HW Limit | ABS | -43138 | Enc |
| ☑ | Set Upper HW Limit | LHW | 104100 | Enc |
| □ | Set Reference Switch | ABS | 0 | Enc |
| ☑ | Set Index Pulse | REF | 0 | Enc |
| ☑ | Set Home Position | IND | 0 | Enc |

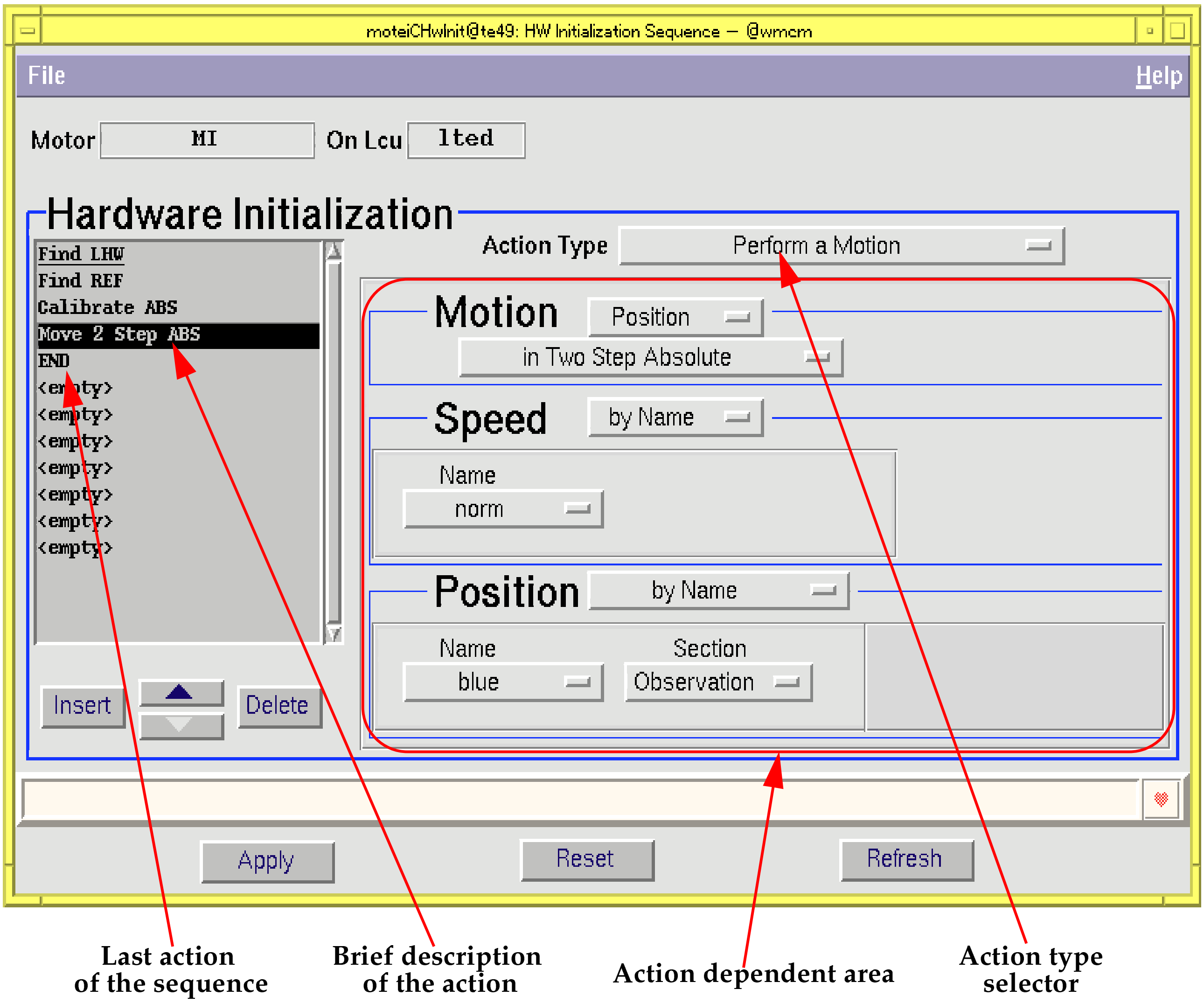
For each of the 5 positions, the Flag indicates if the position shall be made known to the system at completion of the HW initialization sequence if no action had lead to move to. All positions which are not flagged as such must have been moved to before they may be referenced by named positions or in the target definition. This table will be processed at completion of the HW initialization sequence in the order shown in the panel, thus any entry may refer to any HW position reached in a previous step. Finally the SW limits and Named Positions will be resolved for Special Position references.

In order to operate the motor in Simulation mode, the position values of the unflagged positions must be given here so as to instruct MCM about the physical position which would have been reached in normal mode.

This table is to be carefully configured together with the HW Initialization sequence, since they both describe how the motor will be initialized/calibrated.

### 3.4.12 Menu Initialization : Init Sequence ...

The panel below is dedicated to the configuration of the sequence of actions to be performed during the initialization of the motor, see 5.2.12 for man page:


moteiCHwInit@te49: HW Initialization Sequence – @wmcm
File
Help
Hardware Initialization
Insert
Delete
Apply
Reset
Refresh
Last action of the sequence
Brief description of the action
Action dependent area
Action type selector

Motor MI On Lcu lted

Find LHW
Find REF
Calibrate ABS
Move 2 Step ABS
END
<empty>
<empty>
<empty>
<empty>
<empty>
<empty>
<empty>

Action Type: Perform a Motion

Motion: Position, in Two Step Absolute

Speed: by Name — Name: norm

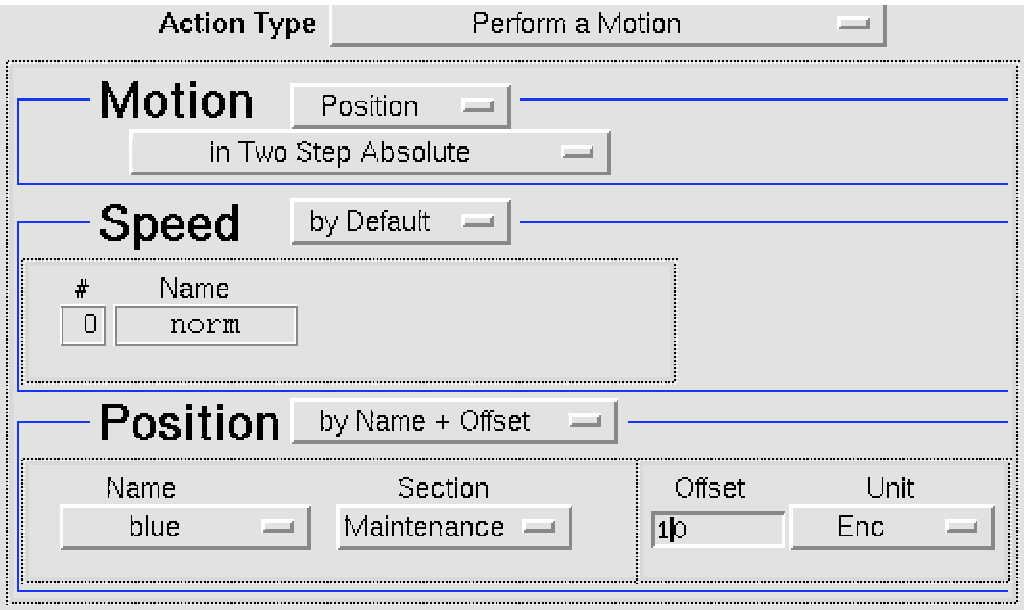
Position: by Name — Name: blue, Section: Observation

The ListBox shows the list of all defined Actions or **<empty>** if the row is available. Whenever an entry is selected (by clicking on it with the mouse), the panel is updated with the action type and its parameters.

The last action of the sequence shall be **END**. A maximum of 12 actions may be defined. Two-step motions (Move 2 Step / Search Index Pulse) count as 2 actions, thus the effective size of the table is reduced by one every time a two-step motion is defined in the initialization sequence.

The panel insets as of the action type, displayed in the Action dependent area, are shown in the subsections below.

#### 3.4.12.1 Perform a Motion



Action Type: Perform a Motion

Motion: Position
in Two Step Absolute

Speed: by Default

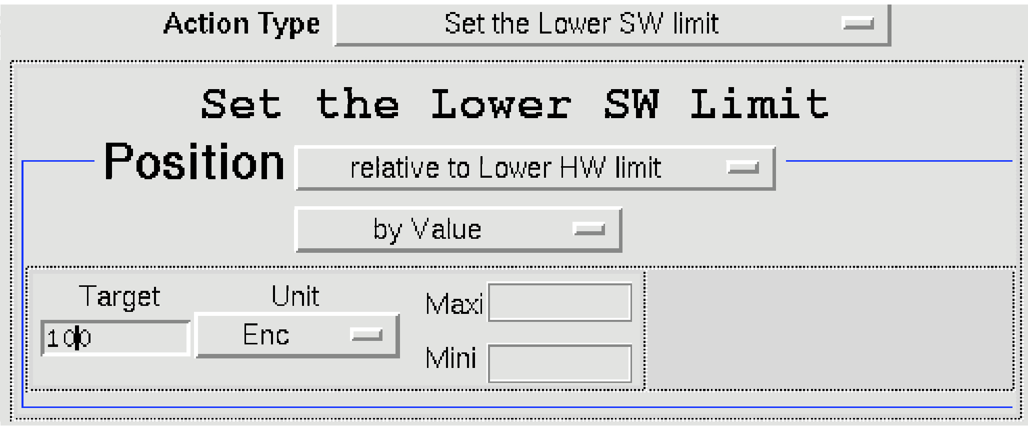
| # | Name |
|---|---|
| 0 | norm |

Position: by Name + Offset

| Name | Section | Offset | Unit |
|---|---|---|---|
| blue | Maintenance | 10 | Enc |

This is the main action dedicated to the description of motions. At this stage none of the special positions is known therefore care should be taken when defining the target positions when not absolute. As well, absolute positions do make sense only after the axis has been calibrated, i.e. a known position is assigned a known value (e.g. find the Reference Switch and calibrate the position as 0 Enc).

#### 3.4.12.2 Set the Lower/Upper SW Limit



Action Type: Set the Lower SW limit

Set the Lower SW Limit

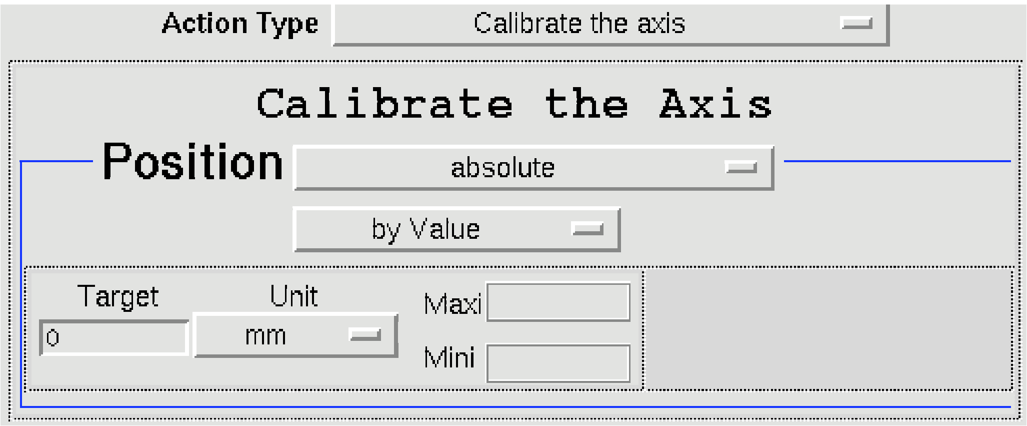
Position: relative to Lower HW limit
by Value

| Target | Unit |
|---|---|
| 100 | Enc |

Maxi
Mini

Only on Linear Axis, this setting are overridden by the SW limit settings as of the Main Configuration panel (see 3.4).

#### 3.4.12.3 Calibrate the Axis



Action Type: Calibrate the axis

Calibrate the Axis

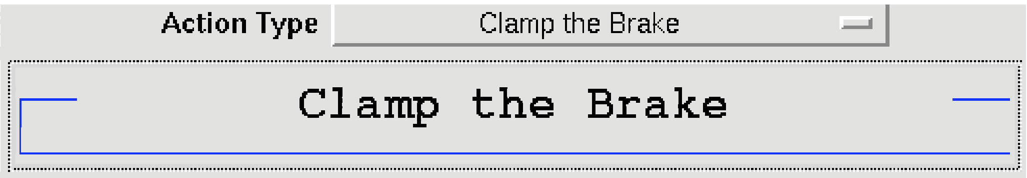
Position: absolute

by Value

| Target | Unit | Maxi | Mini |
|---|---|---|---|
| 0 | mm | | |

The current position will be forced to the position described in the inset above. All special positions will be accordingly corrected of the offset.

#### 3.4.12.4 Clamp the Brake

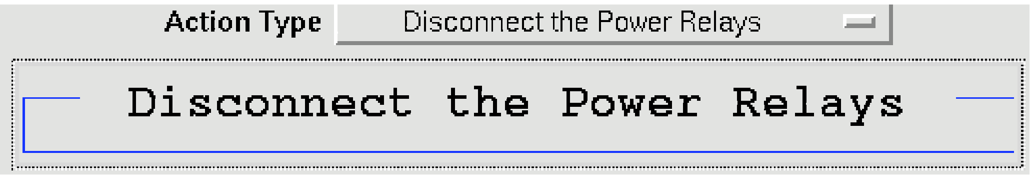


Action Type: Clamp the Brake

Clamp the Brake

Only together with Digital I/O board accordingly configured (see 3.4.5).

#### 3.4.12.5 Disconnect the Power Relays

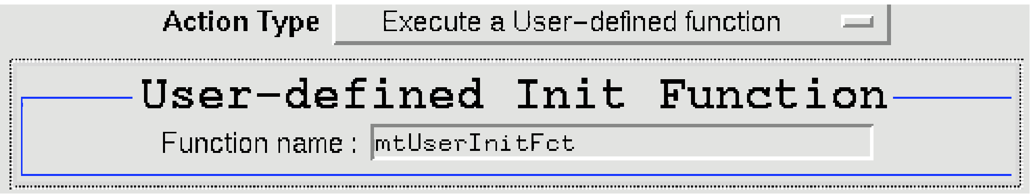


Action Type: Disconnect the Power Relays

Disconnect the Power Relays

Only together with ESO VME4SA Servo Amplifier.

#### 3.4.12.6 Execute a User-defined function



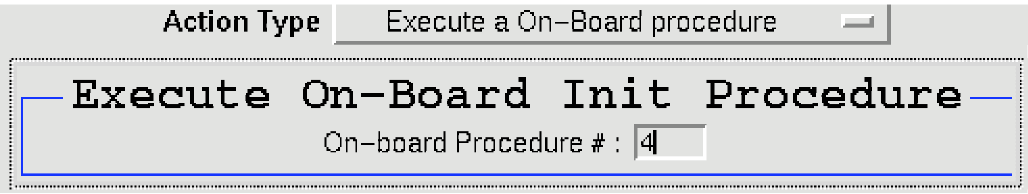
Action Type: Execute a User-defined function

User-defined Init Function

Function name : mtUserInitFct

For user function, enter the name of the user function. The name must be found in the Global Symbol table of the LCU and must comply the type **`motINIT_FUNC`** (defined in **`mot.h`**). The names are resolved during the HW Initialization. See also section 3.5.7 for argument passing mechanism.

#### 3.4.12.7 Execute a On-Board procedure



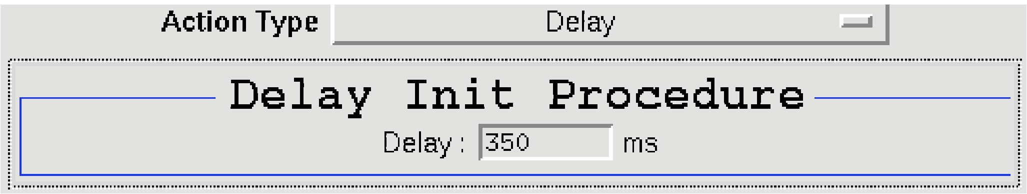
Action Type: Execute a On–Board procedure

Execute On-Board Init Procedure

On–board Procedure # : 4

Only on Motion Controller supporting this feature (e.g. CAMAC).

#### 3.4.12.8 Delay Init Procedure



Action Type: Delay

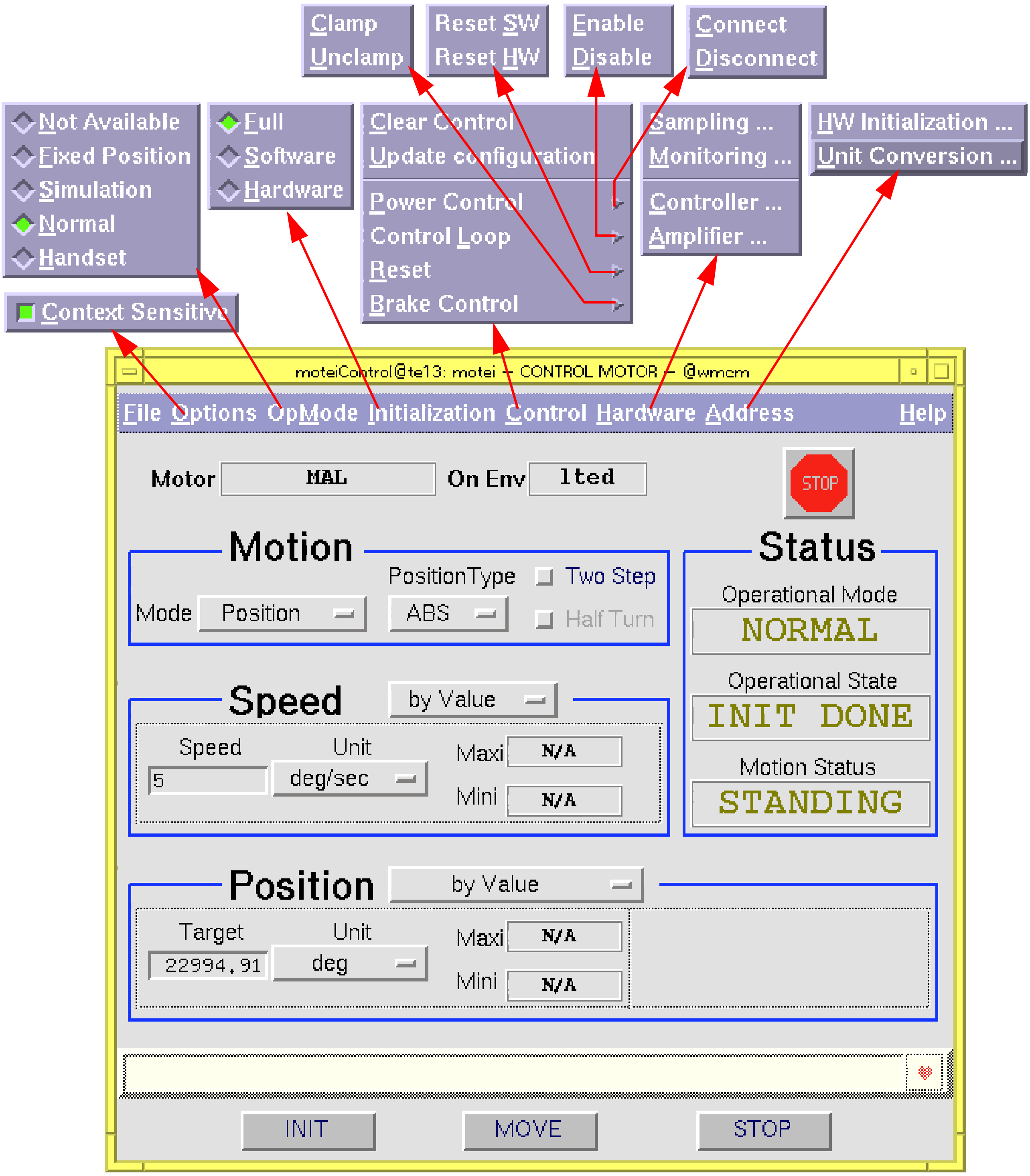
Delay Init Procedure

Delay : 350 ms

## 3.5 Control the Motor

The Control Panel is raised after clicking on the **`Control...`** button of the Main panel.

It controls operation on motors: Initialization, Motion and a set of on-line control/parametrization actions. The panel is shown in the figure below; Menu bar options are inset above the panel. See 5.2.13 for the man-page.



Motor MAL On Env lted

Mode Position PositionType ABS □ Two Step □ Half Turn

Speed by Value

Speed 5 Unit deg/sec Maxi N/A Mini N/A

Position by Value

Target 22994.91 Unit deg Maxi N/A Mini N/A

Operational Mode NORMAL

Operational State INIT DONE

Motion Status STANDING

### 3.5.1 Active Control Operations

This panel is driven context-sensitive, it means that the panel parts/elements are automatically enabled/disabled according to the context, i.e. the state of the system. This feature is intended to help the user by simplifying the panel as the mandotory inputs are concerned. However, it is always possible to disable this behaviour and operate the panel with all elements/parts accessible: in this case, most of the checks are disabled, thus this mode is to be handled with care.

The **`STOP`** buttons are **ALWAYS** enabled.

#### 3.5.1.1 Operational Mode selection

To switch the motor to any operational mode (Normal, Simulation and Handset) the wanted mode must be selected.

1. **Simulation** Operation without hardware: all actions enabled, simulated and performed instantaneously.
2. **Normal** Operations on the hardware: all actions are enabled.
3. **Handset** Manual control through the amplifier board.

A complete description of the operational modes is given in [2].

#### 3.5.1.2 Initialize the Motor

Before starting to move a motor, it must be initialized from the SW and HW point of view. This task is performed by pressing the **`INIT`** button.

The MenuBar ***Initialization*** allows to choose between **Full** initialization (default) and **Software** or **Hardware** initialization only.

The hardware initialization can be performed only when the Software initialization has been previously performed (Operational State is **`INIT SOFT`**). During the Hardware initialization, the sequence as defined in Section 3.4.12 will be performed; while executing, the Operational State is **`INIT RUN`**.

At completion, the Operational State is **`INIT DONE`**.

#### 3.5.1.3 Define a Motion

This part of the Control panel is only accessible when the motor has been successfully initialized: Operation State is **`INIT DONE`**.

- Motion Type Selection

  A motion may be of three (3) types, which define how the motor shall be moved:

  **Direct** via the amplifier to a HW position: only the speed can be specified.

  **Position** via the motion controller at a given speed to a target position defined by a relative or absolute value

  **Speed** via the motion controller at a given speed to a HW position.

A complete description of the motion modes is given in [2].

- Position Type Selection

  Depending on the motion mode, the motor shall be moved to the following positions according to the axis type:

☞**Direct Mode:** the target is a HW position, the motion ends on the switch. The last type (**SPD**) has no completion condition, thus it ends either on a **STOP** command or fails on reaching **LHW** or **UHW** whenever available:

- **LHW** move to the *Lower Hardware Limit* (only Linear axis)
- **UHW** move to *Upper Hardware Limit* (only Linear axis)
- **SPD** move at the given *speed*

☞**Position Mode:** the target is a defined position. When the target position is not absolute (**REL**, **LHW**, **UHW**, **REF**, **IND** or **HOM**)), the target shall be defined as an offset to the reference position whenever available:

- **ABS** move to the specified *absolute position*
- **REL** move to the specified *relative position* from the current position
- **LHW** move of the specified offset from the *Lower Hardware Limit* (only Linear axis)
- **UHW** move of the specified offset from the *Upper Hardware Limit* (only Linear axis)
- **REF** move of the specified offset from the *Reference Switch*
- **IND** move of the specified offset from the *Encoder Index Pulse*
- **HOM** move of the specified offset from the *Home Position*

☞**Speed Mode:** the target is a HW position, the motion ends on the edge of the switch. The last type (**SPD**) has no completion condition, thus it ends either on a **STOP** command or fails on reaching **LHW** or **UHW** whenever available:

- **LHW** find the *Lower Hardware Limit* (only Linear axis)
- **UHW** find the *Upper Hardware Limit* (only Linear axis)
- **REF** find the *Reference Switch*
- **IND** find the *Encoder Index Pulse* (only when delivered by incremental encoder)
- **HOM** find the *Home Position* (only when delivered by incremental encoder)
- **SPD** move at given *speed*

- Speed Definition

  The second item to define is the speed to apply for the motion step. This area is always accessible while the Operational State is **INIT DONE**.

- Target Position

  Only in Position mode, the Position area becomes accessible. The target position may be defined in 5 ways. Beside the setting by Value, Name or Index, 2 additional possibilities are provided that allow the target to be defined as a Named/Indexed position + offset.

#### 3.5.1.4 Initiate the Motion

The button **MOVE** initiates the motion.

While the motor is moving, the Motion Status goes to **MOVING** and the menuBar is disabled.

During a motion, speed and position can be redefined; they are activated with **MOVE**.

#### 3.5.1.5 Stop the Motor

The button **STOP** and the Stop sign instruct MCM to stop the motor immediately.

### 3.5.2 Other Control Operations

#### 3.5.2.1 Menu Control

Two sets of control commands are available:

1. General commands:

   **Clear Control** Clear all connections to motor, reset internal tables, establish new connection. The Operational State is reset to **`NOT INIT`**.

   **Update Configuration** Force re-installation of motor, read the database configuration (see motDeinstall/motInstall(1)). The Operational State is reset to **`NOT INIT`**.

2. Specialized commands:

   **Power Control** Connect/Disconnect the power relays (only with VME Amplifier)

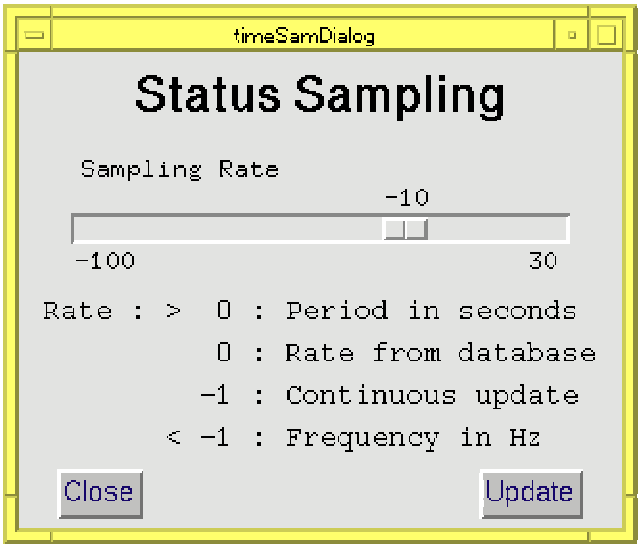
   **Loop Control** Enable/Disable the position control loop (only with Controller)

   **Reset** Reset Axis/Boards. The Operational State is reset to **`NOT INIT`**.

   **Brake Control** Clamp/Unclamp the brake (only with Digital I/O board)

### 3.5.3 Menu Hardware : Sampling ...

The slider as shown below allows the on-line setting the Status Sampling Period, i.e. the period at which the whole motor status shall be refreshed in the database.



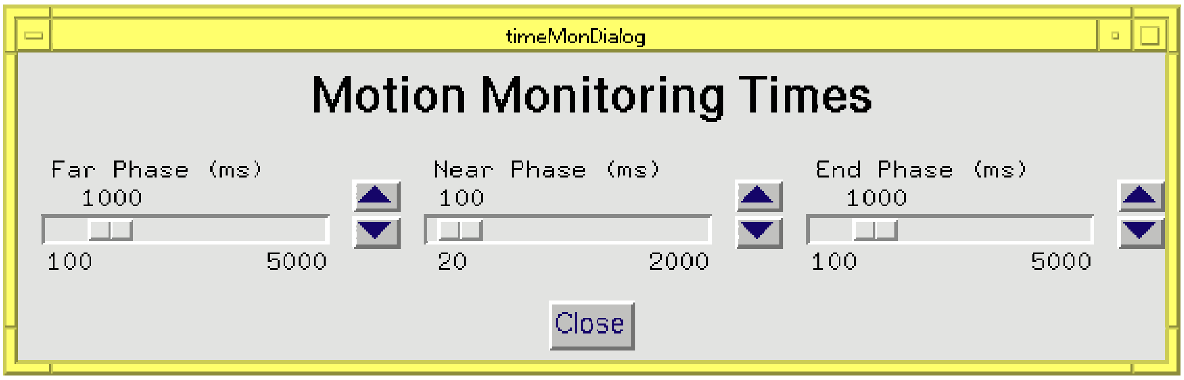
Sampling Rate -10

```
Rate : >  0 : Period in seconds
          0 : Rate from database
         -1 : Continuous update
       < -1 : Frequency in Hz
```

**Note:** High sampling rate implies also a higher CPU load. In the standard HW configuration MCM needs approx. 9.5ms to refresh the status. When many motors are controlled in parallel, high rates may not be effective.

### 3.5.4 Menu Hardware : Monitoring ...

The 3 sliders as shown below allows the setting the Motion Monitoring Times.



Far Phase (ms) 1000

Near Phase (ms) 100

End Phase (ms) 1000

These times are defaulted to 1000, 200 and 1000ms resp. and are applied when defining the speed by value.

**Note:** High polling rates imply also a higher CPU load. When many motors are controlled in parallel, high rates may not be effective. For more details, see [2] Section 2.4.7.

### 3.5.5 Menu Hardware : Amplifier ...

This sub-panel allows the examination/setting of any board parameter.

Not implemented yet.

### 3.5.6 Menu Hardware : Controller ...

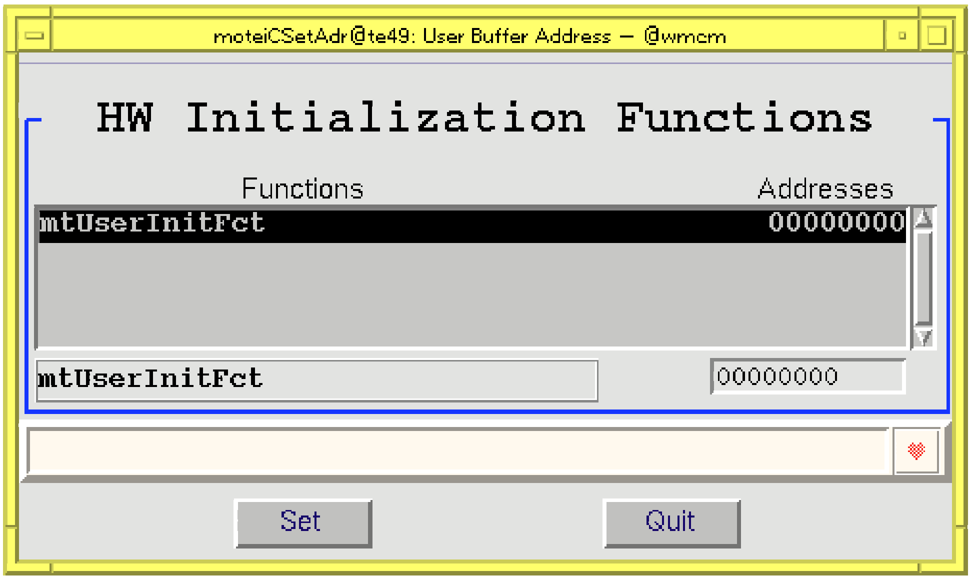
This sub-panel allows the examination/setting of any board parameter.

Not implemented yet.

### 3.5.7 Menu Address

These 2 panels are accessible when reference to external function(s) is made either in the Unit Conversion methods (see 3.4.6) or in the HW Init Sequence (see 3.4.12). See 5.2.14 for the man-page.

#### 3.5.7.1 HW Initialization ...

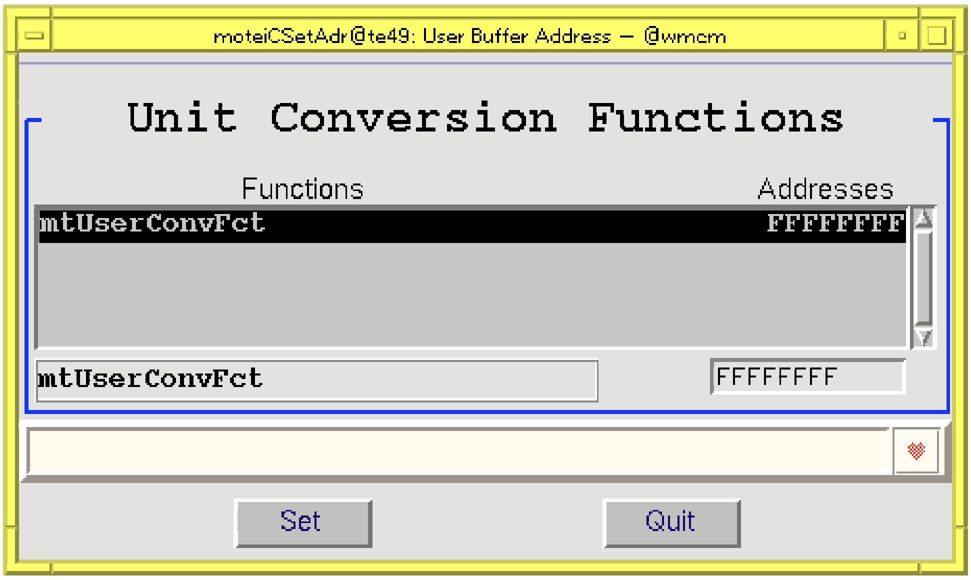

moteiCSetAdr@te49: User Buffer Address – @wmcm
HW Initialization Functions
Functions
Addresses
Set
Quit

mtUserInitFct 00000000

mtUserInitFct 00000000

**Address in HEX must point a readable memory buffer. NULL is always accepted.**

#### 3.5.7.2 Unit Conversion ...


moteiCSetAdr@te49: User Buffer Address – @wmcm
Unit Conversion Functions
Functions
Addresses
Set
Quit

mtUserConvFct FFFFFFFF

mtUserConvFct FFFFFFFF

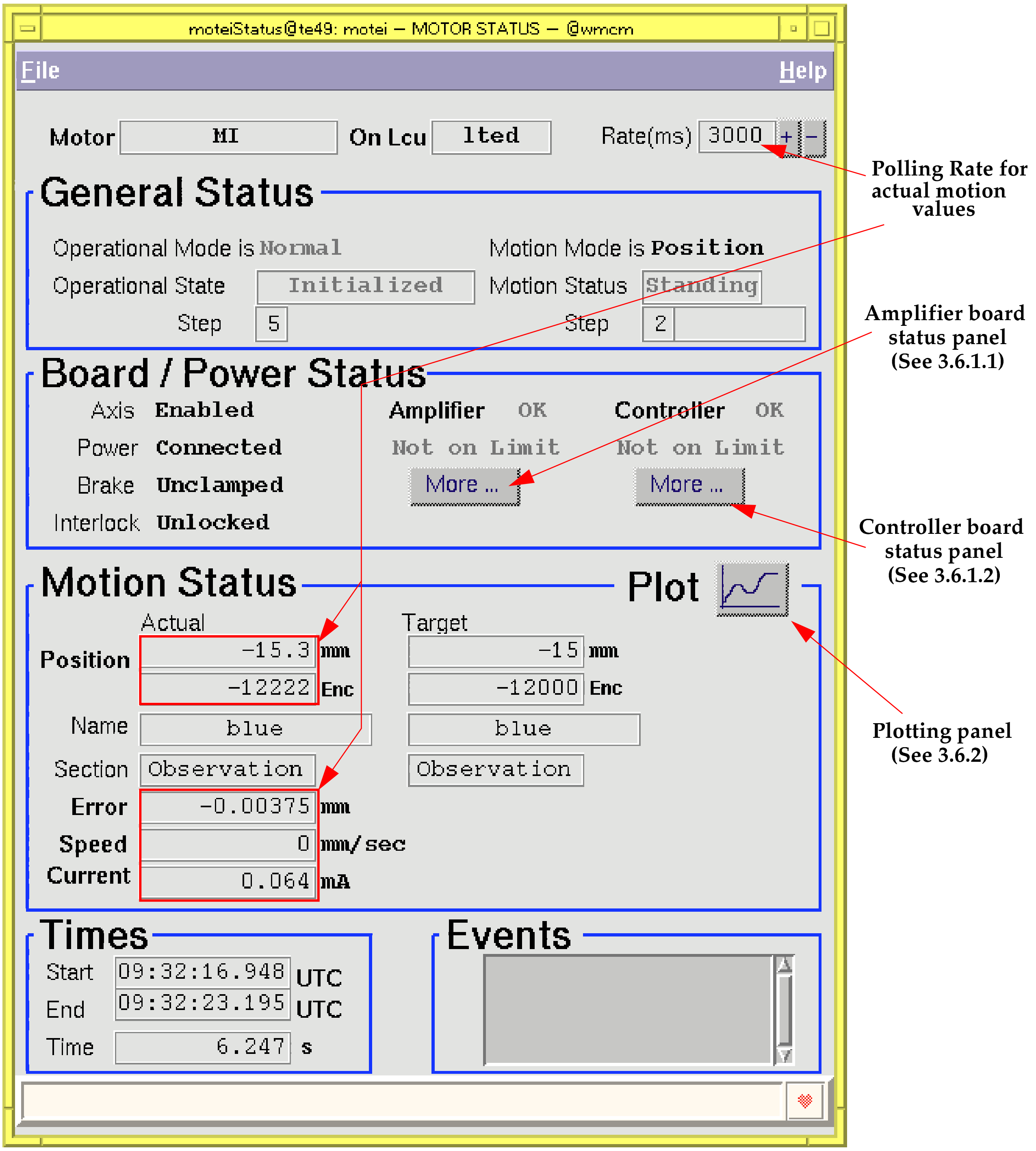
**Address in HEX must point a readable memory buffer. NULL is always accepted.**

### 3.6 Motor Status

The Status Panel is raised after clicking on the **`Status...`** button of the Main panel.

The motor status is only accessible on LCU environments.

The panel (See 5.2.15 for man page), as shown in the figure below, displays the full status of the selected motor, as stored in the database.



moteiStatus@te49: motei – MOTOR STATUS – @wmcm

File Help

Motor MI On Lcu lted Rate(ms) 3000 + –

**General Status**

Operational Mode is Normal Motion Mode is Position

Operational State Initialized Motion Status Standing

Step 5 Step 2

**Board / Power Status**

Axis Enabled Amplifier OK Controller OK

Power Connected Not on Limit Not on Limit

Brake Unclamped More ... More ...

Interlock Unlocked

**Motion Status** Plot

| | Actual | | Target | |
|---|---|---|---|---|
| Position | -15.3 | mm | -15 | mm |
| | -12222 | Enc | -12000 | Enc |
| Name | blue | | blue | |
| Section | Observation | | Observation | |
| Error | -0.00375 | mm | | |
| Speed | 0 | mm/sec | | |
| Current | 0.064 | mA | | |

**Times**

Start 09:32:16.948 UTC

End 09:32:23.195 UTC

Time 6.247 s

**Events**

Polling Rate for actual motion values

Amplifier board status panel (See 3.6.1.1)

Controller board status panel (See 3.6.1.2)

Plotting panel (See 3.6.2)

It is divided into five parts: from top to bottom, **General**, **Board/Power**, **Motion**, **Time** and **Events Status**. The information displayed is assumed to be self-explicative. The **Position** information displays both **Actual** and **Target** position in **User** and **Encoder** units.

### 3.6.1 Buttons More ...

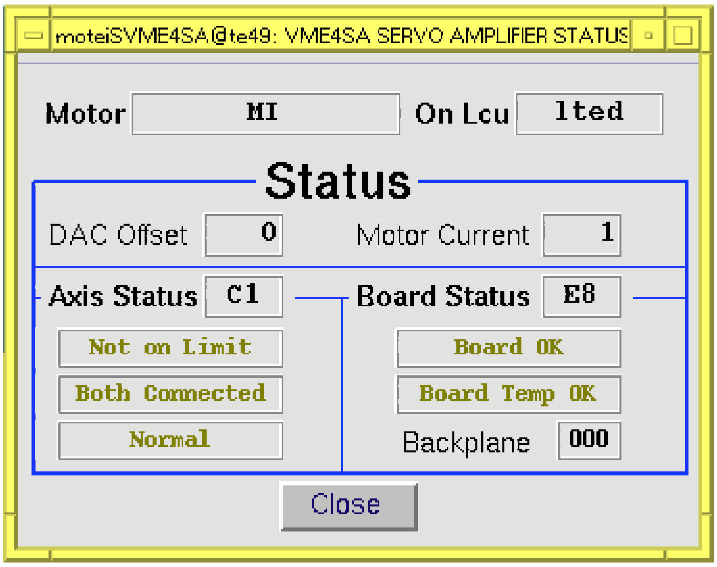
These 2 buttons raise dedicated status panels for the Amplifier and Controller boards.

No Extended-Help is available since the information is reflecting the hardware, therefore please refer to the corresponding manuals.

**Note:** The values shown on the two following panels are refreshed on database change events. This may lead to high message traffic between the LCU and the WS.

#### 3.6.1.1 Amplifier board status

Only the ESO VME4SA Servo Amplifier is supported, especially this panel can not be displayed for the Stand-Alone amplifier ESO VME4ST (for Stepper motors).



moteiSVME4SA@te49: VME4SA SERVO AMPLIFIER STATUS

Motor MI On Lcu lted

Status

DAC Offset 0 Motor Current 1

Axis Status C1 Board Status E8

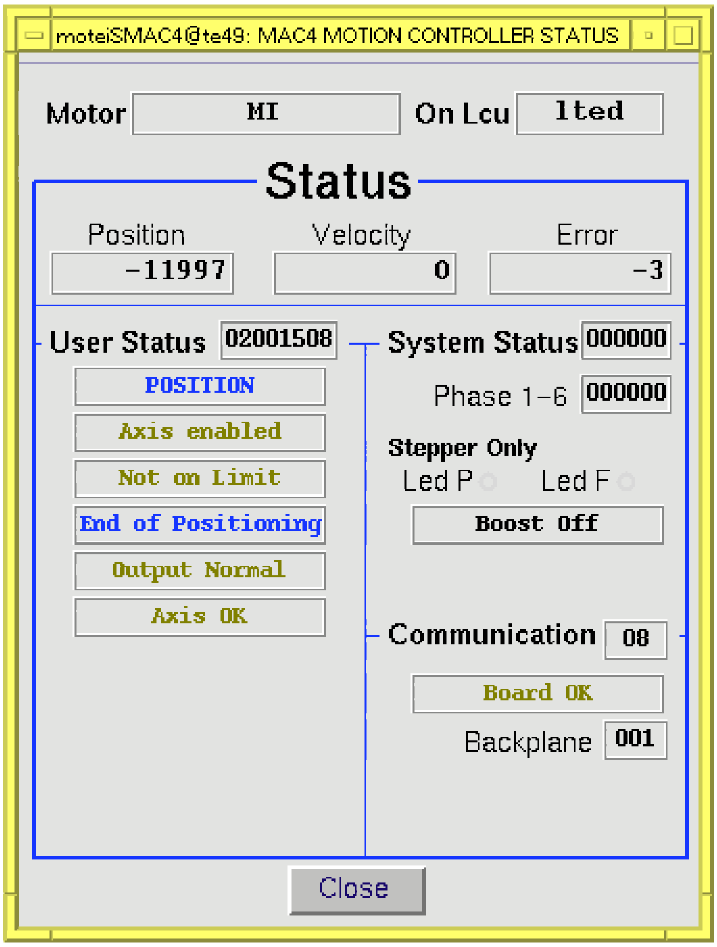
Not on Limit Board OK

Both Connected Board Temp OK

Normal Backplane 000

### 3.6.1.2 Controller board status

Only the Maccon MAC4 Motion Controller is supported.



moteiSMAC4@te49: MAC4 MOTION CONTROLLER STATUS

Motor MI On Lcu lted

Status

Position -11997

Velocity 0

Error -3

User Status 02001508

POSITION

Axis enabled

Not on Limit

End of Positioning

Output Normal

Axis OK

System Status 000000

Phase 1-6 000000

Stepper Only

Led P ○ Led F ○

Boost Off

Communication 08

Board OK

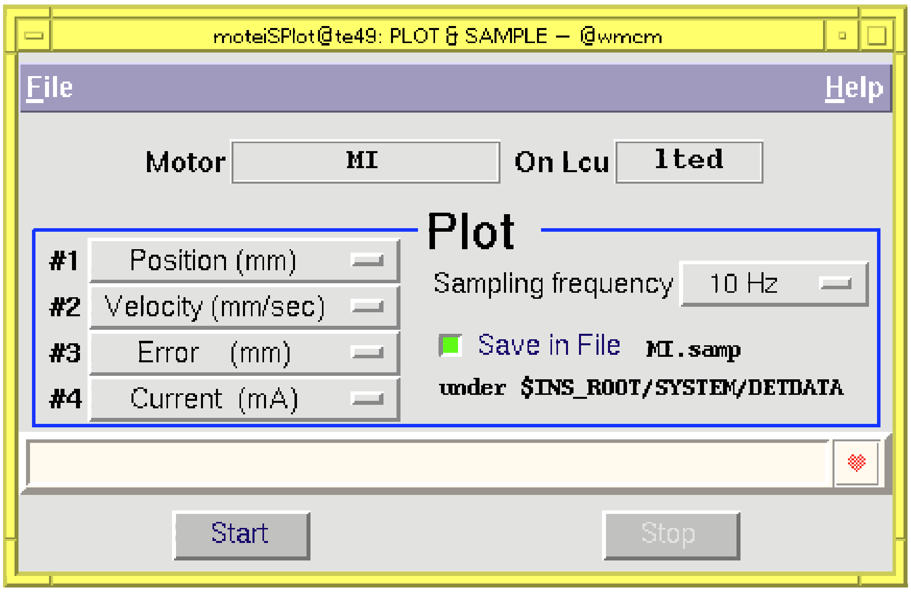
Backplane 001

### 3.6.2 Icon Plot : Plotting Facility

This icon button raise a panel that provides item selectors for plotting. This feature is only available when the WS environment is of type **RTAP** (full-CCS), since it refers to some database points.

This facility may help tuning the control loop parameters in order to reduce possible oscillations by plotting the following error and the actual position as monitored from the motion controller (EP & CP). When plotting high level quantities (not in board units), one shall considere the status sampling period and the motion monitoring times from which the effective update rate is determined.

Up to four (4) quantities may be plotted on the same graph. The item list is available for each channel. See 5.2.18 for the man-page.



moteiSPlot@te49: PLOT & SAMPLE – @wmcm

Motor: MI

On Lcu: lted

Plot

#1 Position (mm)
#2 Velocity (mm/sec)
#3 Error (mm)
#4 Current (mA)

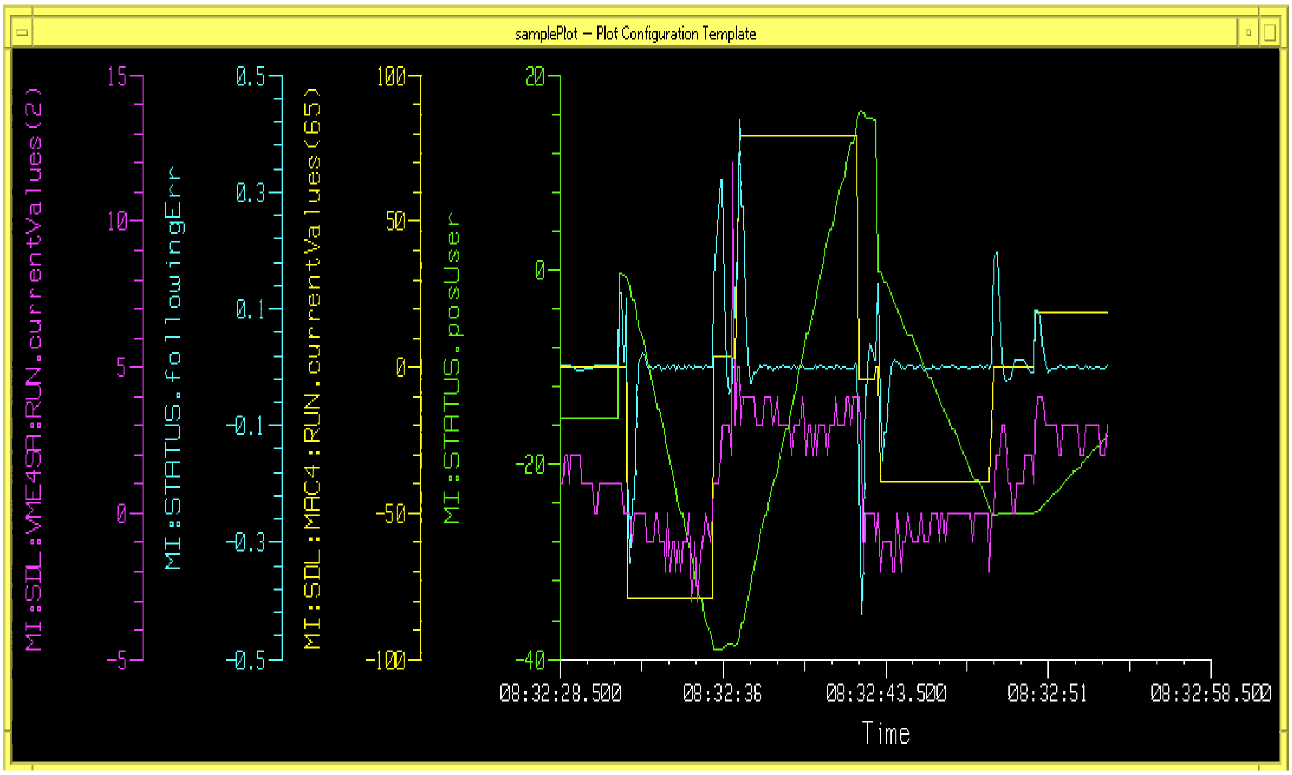
Sampling frequency: 10 Hz

🟩 Save in File MI.samp

under $INS_ROOT/SYSTEM/DETDATA

**Note :** CP, EP & AV only with Motion Controller MAC4;
MC only with Servo Amplifier VME4SA.

#### 3.6.2.1 Example



This diagram shows the time evolution of the position (mm), velocity (Enc/ms), following error (mm) and current (Dig) for a typical HW initialization sequence.

## 3.7 Shutdown the Motor Engineering Interface

Select menubar ***File*** option ***Quit*** in the Main Panel. All panels will be shutdown. This operation may take some time until the whole application has terminated. The connection to the motor is detached, however the motor is not de-installed.

Following the same principle when ***Quit***ting one of the main 3 panels (Status, Config or Control) all related panels underlying the panel to close will be closed.

# 4 INSTALLATION

The generation and installation of the **motei** module follows VLT standards as described in [2].

## 4.1 General Requirements

The **motei** module requires the availability of Hardware and Software detailed in [2], **CCS** or **CCS-lite** and the following modules as of VLT Common Software Release OCT98:

- **seq** version 2.25 or higher
- **panel** version 4.12 or higher
- **uif** version 3.9 or higher
- **samp** version 1.48 or higher
- **mcm** version 2.7 or higher

## 4.2 Database configuration

In both LCU & WS environments (not on CCS-lite), the database shall contain MCM motor branches. As described in [2], chapter 4, section 4, a motor branch is instanciated from the base class **`motMOTOR`**, or from any specialized sub-class, such as **`motDVA0X`**, **`motDVAMA`**, **`motDVAMI`**, **`motDXXMA`**, **`motDXXMI`** & **`motSVSMS`**, derived from the main class **`motMOTOR`**, by overloading.

Dedicated to the Engineering Interface, 6 motors have been instanciated resp. from each sub-class. These branches are collected in the file **`moteiDATABASE.db(5)`** (see 5.3.1) that may be included in the environment configuration file **`$VLTDATA/ENVIRONMENTS/<mcmEnv>/dbl/DATABASE.db`**.

On WS/CCS environment, the motor branches are located under **`:Appl_data:MOTEI`**
On LCU environment under **`:MOTEI`**

The motors are aliased according to their class names : **`DVA0X`**, **`DVAMA`**, **`DVAMI`**, **`DXXMA`**, **`DXXMI`** & **`SVSMS`**.

The tables default to the maximum number of entries and preset the HW initialization sequence to one single action **`END`**, implying that the motor may be initialized without motion therefore enabling the normal motions for tuning.

The following table describes the 6 motors as of their HW configuration, all axis are linear, the board parameters are preset to their default values:

| MOTOR | Motor | Encoder | Servo Amplifier | Motion Controller |
|---|---|---|---|---|
| DVAMI | DC | Incremental | VME4SA | MAC4-INC |
| DVAMA | DC | Absolute | VME4SA | MAC4-SSI |
| DVA0X | DC | none | VME4SA | none |
| DXXMI | DC | Incremental | stand-alone | MAC4-INC |
| DXXMA | DC | Absolute | stand-alone | MAC4-SSI |
| SVSMS | Stepper | Incremental | stand-alone[a] | MAC4-STP |

[a] VME4ST or compatible

Below is an example of the file **`$VLTDATA/ENVIRONMENTS/<mcmEnv>/dbl/DATABASE.db`** :

```
//*******************************************************************************
//* E.S.O. - VLT project
//*
//* “@(#) $Id: DATABASE.db,v 1.16 1997/04/01 12:29:01 vltsccm Exp $”
//*
//* who       when      what
//* --------  --------  ----------------------------------------------
//* NNNNNNNN  dd/mm/yy  created
//*

//**************************************************************************
// VLT COMMON SOFTWARE
//**************************************************************************

// Loads classes definition from standard file
#include “CCS.db”

//
// USER.db contains necessary points for CCS, but requiring editing to match
// installed configuration.
#include “USER.db”

//**************************************************************************
// APPLICATION SOFTWARE
//**************************************************************************
//
// Load branches needed by your application
//
#ifdef MAKE_VXWORKS
#define motMOTORROOT  :MotorINC
#else
#define motMOTORROOT  :Appl_data:MotorINC
#endif
#define motMOTORTYPE  motDVAMI
#define motMOTORNAME  MI
#define motNUMBER_NAMED_POSITIONS 10
#define motNUMBER_NAMED_SPEEDS    10
#include “motor.db”
//
// Load motei branch
//
#include “moteiDATABASE.db”

// ___oOo___
```

The database will contain the 6 template motors instanciated in **moteiDATABASE.db** and one motor aliased **MI**.

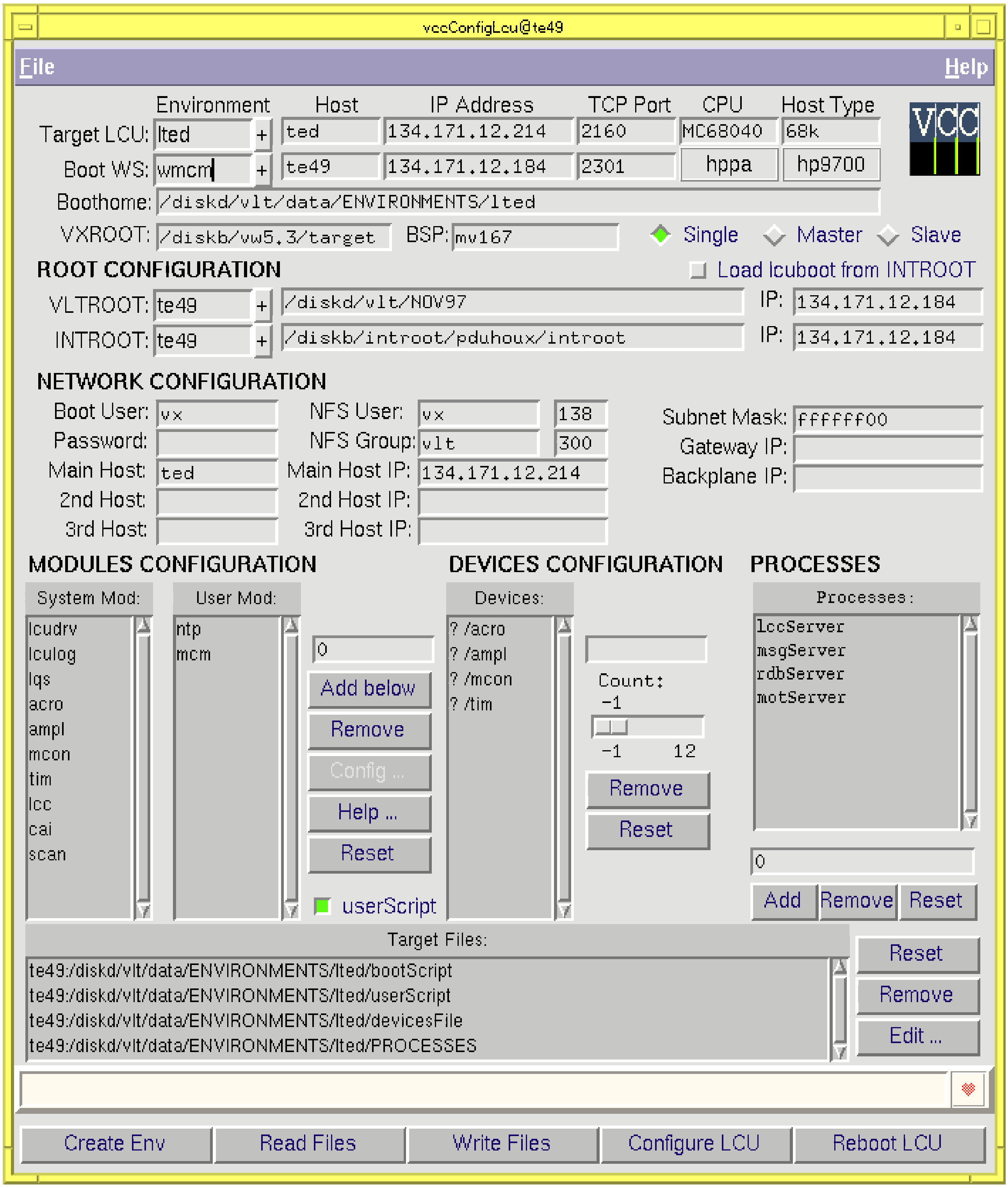
## 4.3 LCU Configuration

It is recommended to use the tool **vccConfigLcu(1)** to set-up the modules and processes required in the LCU-side of MCM:

1. **System Modules:**
   a. lcudrv, lculog, lqs
   b. drivers: ampl, mcon
   c. optional drivers : acro (for signals), tim
   d. lcc, cai
   e. scan (optional)
2. **User Modules:**
   a. mcm
   b. ntp (optional)
3. **Processes:**
   a. lccServer, msgServer, rdbServer
   b. motServer

See [3] or the corresponding man-pages for instructions to use the **vccConfigLcu** tool.

The figure shown below is an example of a LCU configuration for the MCM Engineering Interface:



| | Environment | | Host | IP Address | TCP Port | CPU | Host Type |
|---|---|---|---|---|---|---|---|
| Target LCU: | lted | + | ted | 134.171.12.214 | 2160 | MC68040 | 68k |
| Boot WS: | wmcm | + | te49 | 134.171.12.184 | 2301 | hppa | hp9700 |

Boothome: /diskd/vlt/data/ENVIRONMENTS/lted

VXROOT: /diskb/vw5.3/target BSP: mv167 ● Single ○ Master ○ Slave

ROOT CONFIGURATION □ Load lcuboot from INTROOT

| | | | | | |
|---|---|---|---|---|---|
| VLTROOT: | te49 | + | /diskd/vlt/NOV97 | IP: | 134.171.12.184 |
| INTROOT: | te49 | + | /diskb/introot/pduhoux/introot | IP: | 134.171.12.184 |

NETWORK CONFIGURATION

| | | | | | | |
|---|---|---|---|---|---|---|
| Boot User: | vx | NFS User: | vx | 138 | Subnet Mask: | ffffff00 |
| Password: | | NFS Group: | vlt | 300 | Gateway IP: | |
| Main Host: | ted | Main Host IP: | 134.171.12.214 | | Backplane IP: | |
| 2nd Host: | | 2nd Host IP: | | | | |
| 3rd Host: | | 3rd Host IP: | | | | |

MODULES CONFIGURATION

System Mod:
lcudrv
lculog
lqs
acro
ampl
mcon
tim
lcc
cai
scan

User Mod:
ntp
mcm

0

■ userScript

DEVICES CONFIGURATION

Devices:
? /acro
? /ampl
? /mcon
? /tim

Count:
-1
-1 12

PROCESSES

Processes:
lccServer
msgServer
rdbServer
motServer

0

Target Files:
te49:/diskd/vlt/data/ENVIRONMENTS/lted/bootScript
te49:/diskd/vlt/data/ENVIRONMENTS/lted/userScript
te49:/diskd/vlt/data/ENVIRONMENTS/lted/devicesFile
te49:/diskd/vlt/data/ENVIRONMENTS/lted/PROCESSES

# 5 REFERENCE

This chapter provides the manual pages of the scripts, panels and database files.

## 5.1 Scripts

### 5.1.1 motei(1)

**NAME**

```
motei - Invoke the Motor Engineering Interface
```

**SYNOPSIS**

```
motei [<Environment> [<motorName>]]
```

**DESCRIPTION**

```
This script invokes the main panel of MCM Engineering Interface.
Two optional arguments may be specified:
<Environment> : name of the environment to work on.
    WS  Env : any RTAP environment running on this host.
    LCU Env : any LCU environment connected to the RTAP environment
              from which the script has been launched.
    When not given, <Environment> defaults to $MOTENV;
    otherwise to $RTAPENV.
<motorName> : name of the motor to work on.
    The name may be given only together with an environment.
    The motor name is the alias of the database point containing
    the motor branch.
    When not given, <motorName> defaults to $MOTNAME,
    otherwise to '<none>'.
```

**FILES**

```
motei*.hlp extended help ASCII files associated to each panel.

Each panel motei<Topic> is the association of two files:
    motei<Topic>.pan containing the panel description
    motei<Topic>.tcl containing all associated Tcl procedures
```

**ENVIRONMENT**

```
RTAPENV and eventually some LCU environments accessible from RTAPENV
MOTENV, MOTNAME
```

**EXAMPLES**

```
Example 1 :
===========
Let's assume the RTAP environment 'wmcm' which database contains the
motor branch aliased 'MI':
> setenv MOTNAME MI
> motei wmcm
    The fields <Environment> is set to 'wmcm'
    and <Motor Name> is set to 'MI'.
    Both selectors are active.
    The buttons 'Load Config ...' and 'Configure ...' are enabled.

Example 2 :
```

```
===========
Let's assume the LCU 'ted' configured to report to RTAPENV and which
database contains possible motor branches not located under :MOTEI
> setenv MOTENV lted
> motei
    The fields <Environment> is set to 'lted'
    and <Motor Name> is set to '<none>'.
    Only the Environment selector is active.
    All buttons are disabled.

Example 3 :
===========
Let's assume the LCU 'ted' configured to report to RTAPENV and which
database contains the branch :MOTEI and possible motor branches not
located under :MOTEI
> setenv MOTENV lted
> motei
    The fields <Environment> is set to 'lted'
    and <Motor Name> is set to 'DVAMI'.
    Both selectors are active. The MotorName selector gives the
    choice between the 6 standard motei motors DVA0X, DVAMA, DVAMI,
    DXXMA, DXXMI and SVSMS.
    All buttons are enabled.

Example 4 :
===========
Let's assume the LCU 'ted' configured to report to RTAPENV and which
database contains the motor aliased 'MI':
> setenv MOTNAME MI
> motei lted
    The fields <Environment> is set to 'lted'
    and <Motor Name> is set to 'MI'.
    Both selectors are active.
    All the buttons are enabled.
```

**SEE ALSO**

```
Extended help associated to each panel, motei*(7).
```

```
- - - - - -
Last change:  30/09/98-15:01
```

### 5.1.2 moteiShowStatInvoke(1)

**NAME**

```
moteiShowStatInvoke - call motei motor configuration panel
```

**SYNOPSIS**

```
moteiShowStatInvoke -lcu <lcuEnv> -motor <motorName> [-update <secs>]
moteiShowStatInvoke -?
```

**DESCRIPTION**

```
moteiShowStatInvoke is a script which calls the motei's motor
configuration panel connecting it to the lcu environment and
motor given as arguments.

The script checks the availability of the lcu environment and
validate the motor alias by trying to set the cwp to it.
If any of the above checks fails the scripts aborts printing an
error message.
```

**FILES**

```
None.
```

**ENVIRONMENT**

```
None.
```

**RETURN VALUES**

```
0 on success, 1 otherwise.
```

**CAUTIONS**

```
moteiControlServer is spawned if not present.
```

**EXAMPLES**

```
moteiShowStatInvoke -lcu lted -motor DVAMI
```

```
- - - - - -
Last change:  23/05/96-09:46
```

## 5.2 Panels

The man-pages associated to the panels are listed below in the order of their respective accessibility from the menubars of the main panels.

| | | |
|---|---|---|
| 1. Main panel | moteiMain(1) | on page 59 |
| 2. Motor Main Configuration | moteiConfig(7) | on page 61 |
| a. Servo Amplifier VME4SA | moteiCVME4SA(7) | on page 65 |
| b. Motion Controller MAC4 | moteiCMAC4(7) | on page 67 |
| c. Digital Signals | moteiCSignal(7) | on page 70 |
| d. Unit Conversion methods | moteiCUnit(7) | on page 71 |
| e. Named Speeds | moteiCNamSpd(7) | on page 74 |
| f. Index Search Speed | moteiCIndSpd(7) | on page 76 |
| g. Named Positions + Offsets | moteiCNamPos(7) | on page 77 |
| h. Two-Step offset & speed | moteiCTwoStp(7) | on page 80 |
| i. Special Positions | moteiCSpePos(7) | on page 82 |
| j. HW Initialization Sequence | moteiCHwInit(7) | on page 84 |
| 3. Motor Control panel | moteiControl(7) | on page 88 |
| a. External address setting | moteiCSetAdr(7) | on page 93 |
| 4. Motor Status panel | moteiStatus(7) | on page 94 |
| a. Servo Amplifier VME4SA | moteiSVME4SA(7) | on page 96 |
| b. Motion Controller MAC4 | moteiSMAC4(7) | on page 96 |
| c. Plotting Facility | moteiSPlot(7) | on page 96 |

These man-pages (section 7) are accessible on-line from the respective menubar ***Help*** option ***Extended Help***.

### 5.2.1 moteiMain(1)

**NAME**

```
moteiMain - Motor Engineering Interface
```

**EXTENDED HELP**

```
This panel is the main panel from which all sub-panels can be invoked.

It contains a menu bar, 2 input selectors and 4 buttons.

==============================================================================
Selectors:
==========
Selector 'Environment' :
    The selector offers the list of all environments accessible from this host.
    For WS environments, it is recommended to work on local ones,
    ie resident on this host.
    On selection, a procedure is invoked that searches for motors.
    On WS environment, it searches for motor branches located below the
        point :Appl_data:MOTEI
    On LCU environment, it searches for motor branches located below the
        point :MOTEI
        In addition, it sends the command MOTORS to the process 'motServer'
        on the LCU that returns the names of all installed motors.

    Is the list of motors not empty, the selector 'Motor Name' is configured
    with the list and the first name of the list appears in the entry field.
    Is the list empty, the motor selector is preset to '<none>',
    the + button is disabled.

Selector 'Motor Name' :
    The selector offers the list of all motors accessible from this
    environment.
    If a motor is not in the list but has a branch in the database, its name
    (alias of the DB point) can be entered.
    On LCU environment, the command INSTALL is issued that verifies the
      consistency of the database branch.
    On WS environment, no check is performed.

==============================================================================
Buttons:
========
Button 'Load Config' :
    Invokes the DB restore procedure that loads the database with the
    configuration values contained in a so-called '.dbcfg' file.

Button 'Configure' :
    Starts the Control Server (only on LCU environment)
    Invokes the main Configuration panel.

Button 'Control' : only on LCU environment
    Starts the Control Server
    Invokes the Control panel.

Button 'Status' : only on LCU environment
    Invokes the Status panel.

==============================================================================
MenuBar:
========
MenuBar 'File' :
    Command 'Quit' :
        Close this panel and ALL sub-panels,
        Terminates the Control Server process when running.

MenuBar 'Help' :
    Command 'Extended Help' : shows this text.

==============================================================================
                         Panel common description
==============================================================================
Each panel contains 2 output fields that show the selected motor and the
working environment and a menu bar.

MenuBar 'File' :
    Command 'Quit' : close this panel and all sub-panels.
MenuBar 'Help' :
    Command 'Extended Help' : show the associated help text.

                               ___oOo___
```

```
- - - - - -
Last change:  30/09/98-15:01
```

### 5.2.2 moteiConfig(7)

**NAME**

```
moteiConfig - Motor Configuration
```

**EXTENDED HELP**

```
This panel is the main configuration part. It provides all configuration
items for a basic usage of a motor, assumed the base configuration is one of
the standard configurations provided by motei for each generic type of motor:
    DVA0X: DC motor + ESO VME4SA amplifier
    DVAMI: DC motor + MACCON MAC4-INC controller + ESO VME4SA amplifier
           Incremental encoder connected to the controller
    DVAMA: DC motor + MACCON MAC4-SSI controller + ESO VME4SA amplifier
           Absolute encoder connected to the controller
    DXXMI: DC motor + MACCON MAC4-INC controller + stand-alone amplifier
           Incremental encoder connected to the controller
    DXXMA: DC motor + MACCON MAC4-SSI controller + stand-alone amplifier
           Absolute encoder connected to the controller
    SVSMS: Stepper motor + MACCON MAC4-STP controller +
           ESO VME4ST stand-alone amplifier
           Incremental encoder connected to the controller
(see moteiMain(7) for loading a configuration file)

==============================================================================
OpMode : configure is the Operational Mode in which the motor shall be set.
========

Selector 'OpMode':
    Not Available : The function is not available
    Fixed Position: The function is available but the motor shall not be
        moved and is located at a specified fixed position.
        A sub-panel is invoked that allows the setting of this position.
    Simulation    : The function is available and all actions are enabled.
        However no access to the HW.
    Normal        : The function is available and all actions are enabled.
    Handset       : The function is available but all actions are disabled,
        since the motor is controlled via the handset.

==============================================================================
The panel is split in 4 parts and is to be read counter-clockwise as
described below:

==============================================================================
Hardware : description of the hardware configuration
==========
    Motor type : DC or Stepper
    Boards :
        Selector 'Amplifier'  : type of amplifier
            VME Servo Amplifier : this is the standard ESO amplifier VME4SA
            Stand-Alone Amplifier : any kind of amplifier working in
                stand-alone mode (i.e. not connected on the VME-bus)
                It includes the ESO amplifier VME4ST for Stepper Motors
        Selector 'Controller' : type of motion controller
            No Motion Controller : for motors to be controlled via a VME
                amplifier only
            MAC4 Motion Controller : this is the standard ESO controller
                It includes all the versions (INC, SSI and STP)
            CAMAC Controller : this is the EMMI controller used at NTT
                (ESO internal use only)
        CheckBox 'Digital I/O' : indicates if the ESO standard digital I/O
            board is plugged in.
            This board is used to handle the Brake (1 OUT & 1 IN binary lines)
            and Interlock (1 Input binary line) signals.
```

```
==============================================================================
Axis : description of the kind of axis
======
    The function may be either
    - Linear (moving between two limits, e.g Carriage) or
    - Circular (no limits, e.g Filter Wheel).

    Selector 'Axis' :
        Linear : the function is Linear, SW limits may be set
        Circular :  the function is Circular, No SW limits,
            but the Circular Range must be specified.
        Circular Optimized : idem Circular; however, when using the ESO
            standard controller MAC4, the shortest way to the target will be
            chosen.

    Software Limits : for Linear Axis only
        The SW limits are positions within which the motor is allowed to move.
        They may be defined as Absolute (ABS) or Relative positions.
        For Relative position, it may be either Relative (REL) to the current
        position or to one of the Special Positions (see moteiCSpePos(7) and
        MenuBar 'Initialization'->'Special Positions').
        The SW limits will be set at completion of the HW Initialization,
        therefore all referenced special positions must be "known" by the
        system to allow their computation/setting.

    Circular Range : for Circular Axis only preset to 86400.
        It is the number of Encoder counts (Enc) per revolution of the axis.
        (this value is given by the mechanics).
        For Stepper motor without encoder, its the number of motor steps
        needed per revolution of the axis.
        For DC motor without controller, the value is not relevant and is
        preset to 1.

==============================================================================
Encoder : description of the encoder
=========
    Selector 'Encoder' : select the type of encoder
        None : No encoder. Only for Stepper motor or system w/o controller
        Incremental : The encoder is incremental.
            For the ESO standard controller, it must be of type MAC4-INC
        Absolute : The encoder is absolute.
            For the ESO standard controller, it must be of type MAC4-SSI
        Absolute Relative : The encoder is absolute but its data is to be
            processed as relative.
            For the ESO standard controller, it must be of type MAC4-SSI

    Selector 'Interface' : specifies where the encoder is connected
        Internal : the encoder is connected to the controller
        CPU : a dedicated task on the CPU takes care of reading the encoder
            data and to write it into the controller on interrupt request
        External : the encoder data is accessible at a VME address

        For not 'Internal' encoders, any type of controller MAC4 can be used:
        e.g. Incremental CPU on MAC4-SSI
          or Absolute External on MAC4-INC/STP

        For MAC4-STP only incremental internal encoders are supported.

    Selector 'Code' : Not supported yet.
    Entry 'Count Range' : Absolute encoder only or external encoder
        Number of counts per revolution of the encoder (= resolution)
    Entry 'Valid Bits' : Absolute encoder only or external encoder
```

```
        Number of bits needed to code the encoder data (up to 24)
    Entry 'LSB' : Absolute encoder only or external encoder
        Position of the LSB in the encoder data word
    Entry 'Address' : CPU or External encoder interface only

    Entry 'Counts/turn' : Stepper motor only
        Number of steps per revolution of the motor

==============================================================================
Times : description of the times used internally as timeout or period
=======
    Slider 'Monitor' : it is the status sampling period
        It indicates how often the full motor status must be retrieved
        This activity is handled asynchronously by the task tPoll_<motorName>

    Slider 'Limit' : timeout for the motor to move out of the limit domain
        It sets the maximum time allowed for a motor to move inside the valid
        domain when standing on a limit switch (linear axis only).

    Slider 'Motion Step' : maximum time allowed for a motion step
        It sets the maximum time allowed for a motor to perform a motion step.
        This time prevents the system to loose control on a motor when a
        motion is initiated that does not complete (e.g. due to oscillations
        in the position control loop, the motor does come to stabilize).

==============================================================================
Action Buttons: in addition to the standard ones (see below)
===============
Button 'Save'    : apply the configuration (see button 'Apply') and
                   save the configuration in a file of type .dbcfg via a
                   file selector.

==============================================================================
Menu Bar : its pull-down menues allow the advanced configuration of the motor
==========

Menu Bar 'File'
---------------
    Command 'Generate DBL' : Not supported anymore.
    Command 'Enable Motor Control' : when unset, the configuration will not
        be updated in MCM on Apply or Save. If the control panel is active,
        the configuration update is ignored until the command MenuBar
        'Control'->'Update Configuration' is explicitely issued.
        When set, any Apply or Save of the configuration implies an automatic
        'Update Configuration' in the control panel. The operational state
        is reset to 'NOT INIT'. The dialog box pops up with the message:
        "Database has been modified. Remember to INIT again the motor".

Menu Bar 'Hardware'
-------------------
    Sub-Panel 'Events' : Not implemented yet.
    Sub-Panel 'Amplifier'  : For the selected amplifier, it invokes the
        associated configuration panel.
    Sub-Panel 'Controller' : For the selected controller, it invokes the
        associated configuration panel. (Only MACCON MAC4).
    Sub-Panel 'Brake & Interlock' : Not implemented yet.

Menu Bar 'Units'
----------------
    Sub-Panel 'Conversion' : It invokes the panel for configuration of the
        unit conversion methods when user-defined units have to be used
        instead of the physical board units (Enc for position, Enc/ms or Ref
        for speed and Dig for current).
```

```
Menu Bar 'Speeds'
-----------------
    Sub-Panel 'Named Speeds' : It invokes the panel for configuration of the
        the named speeds.
        These are logical names associated to physical speeds.
    Sub-Panel 'Index Search' : It invokes the panel for configuration of the
        speed to be applied to perform the second step of the search procedure
        that looks for the Index pulse signal of the encoder.

Menu Bar 'Positions'
--------------------
    Sub-Panel 'Named Positions' : It invokes the panel for configuration of
        the named positions.
        These are logical names associated to physical positions.
    Sub-Panel 'Two Step Offset' : It invokes the panel for configuration of
        the offset and speed to be applied for Two-Step type motions.
        Two-Step motion prevent backlash errors in forcing the positioning
        from the same "side" of the target.

Menu Bar 'Initialization'
-------------------------
    Sub-Panel 'Special Positions' : It invokes the panel for configuration of
        the special positions.
        These are positions associated to hardware switches/signals.
    Sub-Panel 'Init Sequence' : It invokes the panel for configuration of the
        sequence of actions to be performed while initializing the motor.
        Mainly it includes find motions to HW positions and positioning to
        a given position.

==============================================================================
               Configuration sub-panel common description
==============================================================================
Each Configuration panel contains 3 action buttons:

Button 'Apply'   : write the configuration as described in the panel to the
                   database.
Button 'Reset'   : restore the original configuration as from before the
                   panel was invoked.
Button 'Refresh' : read the configuration from the database but the items
                   the panel is responsible for. Update the panel accordingly.

                               ___oOo___
```

```
- - - - - -
Last change:  30/09/98-15:01
```

### 5.2.3 moteiCVME4SA(7)

**NAME**

```
moteiCVME4SA - Configure Servo Amplifier ESO 'VME4SA'
```

**EXTENDED HELP**

```
This panel provides the configuration items of the amplifier.

The panel is split in two parts :

==============================================================================
Device:
=======
    For this type of amplifier, the name of the Single Device Library (SDL)
    interfacing MCM with the driver is preset to 'vme4sa', as well the driver
    is preset to '/ampl'. Remain to indicate which HW device is assigned to
    the motor:

    Selector 'Board #' : choice from 1 to 6
        Board #<b> : VME short-IO base address 0x3<b>00 [256 Bytes]
                     (see address selector on the board)

    Selector 'Chan #' : choice from 1 to 4
        Chan #<c>  : index on the board as labelled on the backplane
                     and on the board front panel
                     (1 at the top, 4 at the bottom)

    The effective device number /ampl<d> is computed as :
        <d> = 4*(<b>-1)+(<c>-1)

    On the VME bus, the boards should be located with increasing address
    range at the right side of the CPU:
    E.g. CPU    Board     on Slot #1,
         TIM    Board     on Slot #2,
         MCon   Board #1  on Slot #3,
         VME4SA Board #1  on Slot #4,   -> devices /ampl0 to /ampl3
         MCon   Board #2  on Slot #5,
         VME4SA Board #2  on Slot #6,   -> devices /ampl4 to /ampl7
         etc...
    where <MCon> stands for the associated Motion Controller board.

    Note: For VME4SA-11 boards, two slots must be foreseen (8 TE).

==============================================================================
Parameters:
===========
    One single parameter may be configured:

    DO : DAC Offset in Direct mode only
        It ranges between -128 and +127.
        This value is applied as Velocity Reference as soon the Amplifier
        is set to DAC mode (ie Direct motion mode).
```

___oOo___

```
- - - - - -
Last change:  30/09/98-15:01
```

### 5.2.4 moteiCMAC4(7)

**NAME**

```
moteiCMAC4 - Configure Motion Controller MACCON 'MAC4'
```

**EXTENDED HELP**

```
This panel provides the configuration items of the controller.

The panel is split in two parts :

==============================================================================
Device:
=======
    For this type of controller, the name of the Single Device Library (SDL)
    interfacing MCM with the driver is preset to 'mac4', as well the driver
    is preset to '/mcon'. Remain to indicate which HW device is assigned to
    the motor:

    Selector 'Board #' : choice from 1 to 6
        Board #<b> : VME base address 0xC<b>0000 [64 kBytes]
                     (see address selectors on the board)

    Selector 'Chan #' : choice from 1 to 4
        Chan #<c>  : index on the board as labelled on the backplane
                     and on the board front panel
                     (1 at the top, 4 at the bottom)

    The effective device number /mcon<d> is computed as :
        <d> = 4*(<b>-1)+(<c>-1)

    On the VME bus, the boards should be located with increasing address
    range at the right side of the CPU:

    E.g. CPU    Board     on Slot #1,
         TIM    Board     on Slot #2,
         MAC4   Board #1  on Slot #3,   -> devices /mcon0 to /mcon3
         Ampl   Board #1  on Slot #4,
         MAC4   Board #2  on Slot #5,   -> devices /mcon4 to /mcon7
         Ampl   Board #2  on Slot #6,
         etc...
    where <Ampl> stands for the associated VME Amplifier board.

==============================================================================
Parameters:
===========
    All configuration parameters are R/W accessible unless specified.
    Since the configuration of the MACCON MAC4 motion controller is complex,
    only high level information is given here below,
    see MACCON MAC4 User Manual for details.

    Common Configuration Parameters:
    --------------------------------
        Acceleration/Deceleration:
        --------------------------
        SA SD : Speed mode acc./dec.
        PA PD : Position mode acc./dec.
        FA FD : Speed mode to LHW/REF/IND/HOM/UHW (Find mode) acc./dec.

        Test mode:
        ----------
        DA : DAC output (Test mode only)

        Axis and Encoder:
```

```
    -----------------
    AX : Axis type
    CR : Circular Range (> 0)
    LL UL : Software limits
    ET : Encoder type
    EB : Encoder bits
    EC : Encoder count / revolution
    EO : Encoder offset (not used)

    Hardware Switches:
    ------------------
    ULL : Lower HW limit signal logic state
    LLL : Upper HW limit signal logic state
    RSL : Reference Switch signal logic state
    DFL : Drive Fault signal logic state

    Adcanced Settings:
    ------------------
    POL : Output Polarity (WR only)
    WD  : Watchdog (not used, should be set to 0ms)
          Maximum time between command - 0 means not watchdog
    MF  : Maximum Following Error
          Maximum allowed difference between effective and
          theoretical motion profile
          Range from 0 to 32000 Enc
    ED  : Emergency Deceleration
          Deceleration applied on HW/SW limit, Reset, Drive Fault,
          Following Error
    TR  : Target Radius
          Define the tolerance window around the target position
          Range from 0 to 32000 Enc
    TT  : Target Settle Time
          Time in milli-seconds to settle up within the target window
          before notifying motion end
    SF  : Scaling Factor
          Effective scaling factor is 2^SF
          Apply to all accel/decel and speeds
    LV  : Low velocity for Find Edge, Search Index & Home modes

Specific Parameters:
--------------------
MAC4-INC:
    Acceleration/Deceleration:
    -------------------------
    CA CD : Index Search Coarse mode acceleration/deceleration
    IA ID : Index Search Fine mode acceleration/deceleration
    HA HD : Home mode acceleration/deceleration
    Control Loop:
    -------------
    GA : Gain in 0.25 step - Range from 0 to 255 (ga=63.75)
    ZE : Zero in step of 1/256 - Range from 0 to 255 (ze=255/256)
    KI : Integral Gain in step of 1/256 - Range from 0 to 255 (ki=255/256)
    IS : Right bit shift to apply to KI - Range from 0 to 8
    PO : Pole in step of 1/256 - Range from 0 to 255 (po=255/256)
    LT : Limit Torque
         Acts as a speed limiter - Range from 1 to 2047
         Maximum DAC unsigned output (2047 = 10V)

MAC4-SSI:
    Control Loop:
    -------------
    GA, ZE, KI, IS, PO, LT : idem INC
    Encoder:
            --------
            SC : LSB for encoder data in word

        MAC4-STP:
            Acceleration/Deceleration:
            --------------------------
            CA CD, IA ID, HA HD : idem INC
            Stepper:
            -------
            ES  : Encoder counts / motor turn - Range 1 to 1000000 Incr
                  1 Incr = 4 Enc
            MSR : Steps / motor turn - Range 1 to 10000000
            SSF : Start/Stop frequency - Range from 1 to 2000 Hz
            BST : Boost Time - Range 1 to 10000 ms
                  Delay before Boost signal active, start at SSF
            CRD : Creep Distance - Range 0 to 30000 steps
                  Distance from target driven at SSF (Boost off)
            SWT : Servo Wait Time
                  Position control refresh period with encoder

    ==============================================================================
    The following parameters are used during motions and for status purpose:

        Run-Time parameters:
        --------------------
            SV,PV,CV,IV,HV,FV,VT : Velocities in all modes
            AP RP : Absolute/Relative target position (Position mode)
            AT RT : Absolute/Relative target position (Tracking mode)
            EA : Encoder Address (external encoder only)
            DL : Switch for Find mode
            IP : Init position (calibration) (WR only)

        Read-Only parameters:
        ---------------------
            CP  : Current Position
            AV  : Actual velocity
            EP  : Error Position
            VER : Firmware version

        Status (Read only):
        -------------------
            USR : User status
            SYS : System status
            COM : Communication status

                                    ___oOo___



- - - - - -
Last change:  30/09/98-15:01
```

### 5.2.5 moteiCSignal(7)

**NAME**

```
moteiCSignal - Configure the Digital I/O Signals
```

**EXTENDED HELP**

```
This panel allows the configuration of the Digital I/O Signals for Brake and
Interlock. Each of the 3 signals is mapped by one (1) bit of the Digital I/O
board.

==============================================================================
Brake:
======
  Two signals shall be configured:
    Brake Action : is the output signal commanding the brake device
    Brake Status : is the input  signal reflecting the brake status
  When the Brake is clamped, no motion can be initiated.

==============================================================================
Interlock:
==========
  One signal shall be configured:
    Interlock : is the input signal reflecting the interlock status
  When the Interlock is set, no motion can be initiated.

==============================================================================
For each signal to be configured, following fields must be set:

Selector 'Device' : choice from /acro0 to /acro3
    It indicated which Digital I/O board shall be addressed.

Entry 'Bit #'     : range from 1 to 64
    It indicated which I/O bit shall map the signal.

Selector 'Logic'  : choice 'OFF', 'Active LOW', 'Active HIGH'
    It indicates how the signal shall be interpreted.
```

**EXAMPLES**

___oOo___

```
- - - - - -
Last change:  30/09/98-15:01
```

### 5.2.6 moteiCUnit(7)

**NAME**

```
moteiCUnit - Configure Unit Conversion Methods
```

**EXTENDED HELP**

```
Besides the common widgets, it contains a list box, a method name edition part,
a section 'Default Units', a 'Conversion Method' selector and a method
dependent part.

==============================================================================
Unit Conversion Methods:
========================
ListBox 'Unit Conversion': the names of all conversion methods are listed.
    At initialization, the first method is selected (highlighted).
    Mouse button 1 (left) is used to select the method.

On Method selection, Apply & Refresh, the current row is taken and replaces
the previous settings.

On selection, the name of the conversion method is displayed in the 'name'
output field and can be modified as described below:
Entry 'From' : this is the "user" defined unit. (max 7 characters)
Selector 'To': this is the board unit selector
    'Enc' : the method will describe how to convert position values from the
         user unit to the position unit 'Enc' and reverse.
    'Enc/ms' : the method will describe how to convert speed values from the
         user unit to the speed unit 'Enc/ms' (all motion modes but Direct)
         and reverse.
    'Ref' : the method will describe how to convert speed values from the
         user unit to the speed unit 'Ref' (Direct motion mode) and reverse.
    'Dig' : the method will describe how to convert current values from the
         user unit to the amplifier current unit 'Dig' and reverse.

    Only these 4 methods and associated user units can be used by the system.
    Other units (e.g. mmToHz will not be used, however they may be invoked by
    the application)

    'Other': any unit (max 7 characters).
         this unit can not be used by the system.

==============================================================================
Button 'Make Default Unit':
===========================
    Any user unit for which a conversion method to one of the board units is
    provided can be set as default unit. This will be the unit in which the
    corresponding quantities will be expressed in the status.
    E.g. 'mmToEnc' allows 'mm' to be set as default position unit.
    This button is disabled whenever the operation is not legal.
    E.g. 'mmToHz' does allow 'mm' to be set as default unit.
        The button is disabled.

==============================================================================
Button 'Set Board Units':
=========================
    Reset the default units to the board units:
    'Enc'    for positions
    'Enc/ms' for speeds ('Ref' is taken for Direct motions)
    'Dig'    for current

==============================================================================
Selector 'Conversion Method':
=============================
    Selection of the conversion method between:
    Linear Function:
    ----------------
        The formula to describe is:
            <Y> = <X> * slope + offset
        where <Y> is expressed in board unit (e.g. Enc)
        and   <X> is expressed in user  unit (e.g. mm)

        The two parameters 'slope' and 'offset' shall be set.
        Note : the value of the slope that must comply:
            |slope| > 1E-38
        to allow the operation:
            <X> = (<Y> - offset) / slope

    Interpolation Table:
    --------------------
        The interpolation table shall contain up to 50 points covering the
        WHOLE conversion domain, including the edges so that the method
        can ALWAYS locate the value to convert between 2 neighbours.
        The first value X shall be expressed in user unit,
        the second one Y in board unit.
        The points must be input in increasing or decreasing order along
        X or Y. The function MUST be bijective:
            Y = f(X) <=> X = ~f(Y)

        The configuration part consists in:
        List Box : shows all interpolation points (X,Y)
            at initialization, the first point is selected (highlighted).
            Mouse button 1 (left) is used to select the point.
        Button 'Insert' : insert a point above the selected one.
                          (X,Y) is preset to (0,0).
        Arrow  'Up'     : exchange selected point with the one above
                          (but first).
        Arrow  'Down'   : exchange selected point with the one below
                          (but last).
        Button 'Delete' : delete selected point (minimum two valid points).

        On Insert, Move Up, Move Down, Apply & Refresh, the current position
        is taken and replaces the previous settings.

        On selection, the values X & Y of the point are displayed
        and can be modified, as well its index in the list versus the total
        number of valid points.

    External Function:
    ------------------
        An external function must comply with the interface motCONV_FUNC
        and MUST provide both conversions ways.
        It belongs to the application and must be found in the global symbol
        table of the LCU at INSTALL time.
        The address of the function will be retrieved from the symbol table
        and stored in the database.
        The user-buffer address is preset to 0xFFFFFFFF.
        The application must explicitely set this parameter before proceeding
        with the initialization.

        (see: Panel Control: MenuBar 'Addresses'->'Unit Conversion' or
         moteiCSetAdr(7), command SETADDR in motServer.cdt).

        See also MCM User Manual VLT-MAN-ESO-17210-0600 section 2.4.12 p.35ff
```

___oOo___

```
- - - - - -
Last change:  30/09/98-15:01
```

### 5.2.7 moteiCNamSpd(7)

**NAME**

```
moteiCNamSpd - Configure Named Speeds
```

**EXTENDED HELP**

```
This panel provides configuration facilities for the so-called Named Speeds.
Named Speeds are logical speeds defined by the user so as to simplify the
motion definition by using pre-defined speeds such as 'coarse' for coarse-type
motions (e.g. find lower HW limit switch), 'normal' for e.g. all positioning
motions, 'fine' for small displacements etc...
These logical speeds avoid as well to modify the OS code in case the encoder
or the gearbox have to be exchanged.

==============================================================================
Named Speeds:
=============
Besides the common widgets, it contains a list box with the 4 associated
buttons, an option button and a configuration dependent part.

List Box : at initialization, the first speed is selected (highlighted).
    Mouse button 1 (left) is used to select the named speed.
Button 'Insert' : insert a new named speed above the selected one:
                  The name is preset to 'speed#<rowNumber>',
                  the speed is preset to <rowNumber> Enc/ms.
Arrow  'Up'     : exchange selected speed with the one above (but first)
Arrow  'Down'   : exchange selected speed with the one below (but last)
Button 'Delete' : delete selected speed.

On Insert, Move Up, Move Down, Apply & Refresh, the current row is taken and
replaces the previous settings.

On selection, the name of the speed is displayed in the 'name' entry field
and can be modified:

Entry 'Speed'   : nominal value of the speed (floating point value).

Selector 'Unit' : unit in which the speed is expressed.
    For named speeds intended to be used in all modes but Direct, the unit
    shall be either a User unit (assumed the conversion method to 'Enc/ms'
    is provided), or the controller speed unit 'Enc/ms'.
    For named speeds intended to be used in Direct mode, the unit shall be
    either a User unit (assumed the conversion method to 'Ref' is provided),
    or the amplifier speed unit 'Ref'.

==============================================================================
Default Speed:
==============
This is the index of the named speed to be used when referencing the speed
'By Default' in the speed definition of a motion.

Button 'Make Default': Take the selected named speed as the default speed.
    The name and nominal speed will be displayed below.

==============================================================================
Motion Monitoring Times:
========================
For each named speed, it is needed to set the monitoring times for
the 3 phases of the motion:
    far    : until the motor has reached the target window (TW)
    near   : from the moment on when the motor has reached the TW
    stable : delay before the controller issues the Motion End
             after the target window has been reached.
    These times are defaulted resp. to 1000, 200 and 1000ms.

==============================================================================
Button 'Motion Monitoring Times':
=================================
The entry fields associated to the 3 times are shown and enabled for
modification.

                               ___oOo___
```

```
- - - - - -
Last change:  30/09/98-15:01
```

### 5.2.8 moteiCIndSpd(7)

**NAME**

```
moteiCIndSpd - Configure Index Pulse Search Speed
```

**EXTENDED HELP**

```
This panel provides configuration facilities for the speed to be applied to
search for the index pulse of the encoder. It is available on system using
an incremental internal encoder supporting this signal.
The description below is complying the search procedure as supported by the
Motion Controller MACCON MAC4.
For other controllers, step 3 may not be performed.

The index pulse search procedure is a 3-step motion consisting in:
- step 1 : coarse motion using the speed as defined for the motion and
           ending after the pulse has been detected.
- step 2 : fine motion backwards using this speed ending on the pulse
           detection.
- step 3 : very fine motion forward using the parameter 'Low Velocity' LV
           of the Motion Controller. This step is made automatically by
           the motion controller MACCON MAC4 at completion of step 2.

==============================================================================
Index Pulse Search Speed:
=========================
Selector 'Speed': indicates how the speed is defined
    By Default : when a default speed is defined
        The index and name of the default speed are displayed.
    By Value   : the speed is defined by a value
        Following widgets are shown:
        Entry 'Speed'   : nominal value of the speed (floating point value).
        Selector 'Unit' : unit in which the speed is expressed.
            The unit shall be either a User unit (assumed the conversion
            method to 'Enc/ms' is provided), or the controller speed
            unit 'Enc/ms'.
        Outputs 'Extrema' : not updated yet, preset to 'N/A' (Not Applicable).
    By Name    : when named speeds have been defined
        A selector is shown that lists all defined named speeds by their name
    By Index   : when named speeds have been defined
        A selector is shown that lists all defined named speeds by their index
```

___oOo___

```
- - - - - -
Last change:  30/09/98-15:01
```

### 5.2.9 moteiCNamPos(7)

**NAME**

```
moteiCNamPos - Configure Named Positions
```

**EXTENDED HELP**

```
This panel provides configuration facilities for the so-called Named Positions.
Named Positions are logical positions defined by the user so as to refer
physical positions such as Filter Wheel slot, Grating emission line wavelength,
or any other position OS may need to refer to without knowing the absolute
physical position. These logical positions avoid as well to modify the OS code
in case the encoder has to be exchanged, recalibrated or offset.

Besides the common widgets, it contains a list box with the 4 associated
buttons, an option button and a configuration dependent part.

==============================================================================
Named Positions:
================
List Box : at initialization, the first position is selected (highlighted).
    Mouse button 1 (left) is used to select the named position.
Button 'Insert' : insert a new name above the selected one.
                  The name is preset to 'position#<rowNumber>',
                  the position is preset to Absolute <rowNumber> Enc,
                  the lower/upper range is set to 1 Enc.
Arrow  'Up'     : exchange selected position with the one above (but first)
Arrow  'Down'   : exchange selected position with the one below (but last)
Button 'Delete' : delete selected position.

On Insert, Move Up, Move Down, Apply & Refresh, the current row is taken and
replaces the previous settings.

On selection, the name of the position is displayed in the 'name' entry field
and can be modified:

Entry 'Position': nominal value of the position (floating point value).

Selector 'Unit' : unit in which the position is expressed.
    The unit shall be either a User unit (assumed the conversion
    method to 'Enc' is provided), or the controller unit 'Enc'.

Selector 'Position Type': indicates how the position must be interpreted
    Absolute : it is an absolute position
    Relative to Lower HW Limit: it is a positive offset to the position of
        the Lower HW limit.
    Relative to Upper HW Limit: it is a negative offset to the position of
        the Upper HW limit.
    Relative to Reference Switch: it is an offset to the position of
        the Reference Switch.
    Relative to Index Pulse: it is an offset to the position of
        the Index Pulse of the encoder.
    Relative to Home Position: it is an offset to the position of
        the Home Position.

    For relative positions, it is assumed that the reference position is known
    by the system at completion of the HW Initialization.
    (see Panel Configuration : MenuBar 'Initialization'->'Special Positions',
     and moteiCSpePos(7), moteiCHwInit(7) accessible from the Extended Help
     of the associated panels).

==============================================================================
Entries 'Range':
================
    Lower : it is the tolerance <L> subtracted from the nominal position <N>
        giving the minimal position value above which the current position
        will be associated to this name when retrieving the status.
    Upper : it is the tolerance <U> added to the nominal position <N> giving
        the maximal position value below which the current position will be
        associated to this name when retrieving the status.

                              Name
                         <------------>
    ---------------------=========*====--------------> axis
                         ^        ^   ^
                        N-L       N   N+U

    The range applies for all sections.

==============================================================================
Button 'Offsets':
=================
The positions are defined for 3 sections: Observation (Nominal), Maintenance
and User-defined.
The Maintenance section is dedicated to operation related to maintenance
activities, such as changing a filter on a wheel. Since it may not be easy to
access this position when in the light beam, its maintenance position shall
be offset so as to facilitate access to this position.
The User-defined section is free for any other offsetting.

One shall distinguish 2 types of offsets:
- global offsets for Maintenance and User-defined sections that apply for any
  named position.
- local offsets that override the global offsets and are associated to this
  named position only.

When clicking the button 'Offsets', one can set both local and global offsets
for both Maintenance and User-defined sections.
In case the local offset shall override the global one, the associated check
box shall be enabled.
For global offsets, the unit is to be selected, whereas for local offsets the
unit is preset to the unit selected for the nominal position.

Note: the setting of the global offset applies for ALL named positions.
-----
```

## EXAMPLES

```
     Let's assume the named position 'blue' absolute -15 mm +1/-2mm
     global offsets 5 & -6 mm for Maintenance resp. User-defined sections
     local offsets 10 & -11 mm, only Maintenance offset enabled.
This gives the domains:               Lower     Upper   limit
     Observation  : [-17,-14]mm   (= -15   -2, -15   +1)
     Maintenance  : [ -7, -4]mm   (= -15+10-2, -15+10+1)
     User-defined : [-23,-20]mm   (= -15- 6-2, -15- 6+1)

Let's move the motor to named position 'blue' Observation:
     the target position is -15mm
Let's move the motor to named position 'blue' Maintenance:
     the target position is -15+10=-5mm
Let's move the motor to named position 'blue' User-defined:
     the target position is -15-6=-21mm

Now, let's retrieve the status for the motor being at position:
     -16mm   : named position 'blue' Observation since within the range
     -4.1mm  : named position 'blue' Maintenance
     -21.5mm : named position 'blue' User-defined
```

```
                                   ___oOo___


- - - - - -
Last change:  30/09/98-15:01
```

### 5.2.10 moteiCTwoStp(7)

**NAME**

```
moteiCTwoStp - Configure Offset & Speed for Two-Step Motion
```

**EXTENDED HELP**

```
This panel allows the setting of the speed and offset to apply for Two-Step
motions. Two-Step motions prevent backlash errors in forcing the positioning
from the same "side" of the target.

==============================================================================
Offset: offset off the target for the first step
=======
Entry 'Offset' : this value indicates how 'far' the target of the first motion
    step shall be.
    - Negative offset indicates that the target will always be
      reached in positive direction;
    - Positive offset indicates that the target will always be
      reached in negative direction

                                              offset<0
    --------------------------*------------|------------*-----> axis
                              ^            ^            ^
                             CP      Target+offset      Target
                              ------------->------------>
                                   step 1      step 2

                                              offset>0
    --------------------------*------------*------------|-----> axis
                              ^            ^            ^
                             CP          Target   Target+offset
                              -------------------------->
                                        step 1
                                           <-------------
                                               step 2

    Where <CP> is the Current Position

Selector 'Unit' : unit in which the offset is expressed.
    The unit shall be either a User unit (assumed the conversion
    method to 'Enc' is provided), or the controller unit 'Enc'.

==============================================================================
Speed: speed to apply in the second step
======
Selector 'Speed': indicates how the speed is defined
    By Default : when a default speed is defined
        The index and name of the default speed are displayed.
    By Value   : the speed is defined by a value
        Following widgets are shown:
        Entry 'Speed'   : nominal value of the speed (floating point value).
        Selector 'Unit' : unit in which the speed is expressed.
            The unit shall be either a User unit (assumed the conversion
            method to 'Enc/ms' is provided), or the controller speed
            unit 'Enc/ms'.
        Outputs 'Extrema' : not updated yet, preset to 'N/A' (Not Applicable).
    By Name    : when named speeds have been defined
        A selector is shown that lists all defined named speeds by their name
    By Index   : when named speeds have been defined
        A selector is shown that lists all defined named speeds by their index
```

**EXAMPLES**

```
Let's considere the Two-Step offset = -10mm, speed 'fine'

1 - Let the motor move from absolute position 200mm to 150mm in two-step
    at speed 'normal'.

    The motor moves forwards at speed 'normal' to position 150-10=140mm,
    then it moves forwards at speed 'fine' to the position 150mm.

2 - Let the motor move from absolute position 200mm to 300mm in two-step
    at speed 'normal'.

    The motor moves forwards at speed 'normal' to position 300-10=290mm,
    then it moves forwards at speed 'fine' to position 300mm.

Let's considere the Two-Step offset = +10mm, speed 'fine'

1 - Let the motor move from absolute position 200mm to 150mm in two-step
    at speed 'normal'.

    The motor moves forwards at speed 'normal' to position 150+10=160mm,
    then it moves backwards at speed 'fine' to position 150mm.

2 - Let the motor move from absolute position 200mm to 300mm in two-step
    at speed 'normal'.

    The motor moves forwards at speed 'normal' to position 300+10=310mm,
    then it moves backwards at speed 'fine' to the position 300mm.
```

___oOo___

```
- - - - - -
Last change:  30/09/98-15:01
```

### 5.2.11 moteiCSpePos(7)

**NAME**

```
moteiCSpePos - Configure the Special Positions
```

**EXTENDED HELP**

```
This panel allows the configuration of the so-called special positions.

Special positions are positions defined by the hardware:
    LHW : lower HW limit switch (for linear axis only)
    UHW : upper HW limit switch (for linear axis only)
    REF : reference HW switch
    IND : Index Pulse (delivered by the encoder when supported)
    HOM : Home Position defined as being the next Index Pulse found
          in positive direction after the Reference Switch edge has been
          detected (the reference switch should be on)

==============================================================================
Special Positions:
==================
A configuration line is assigned to each position:

Field 'Flag' : This check button tells the system if, at completion of the HW
    Initialization Sequence, the position has to be made known to the system.
    If the HW Initialization Sequence forsees a motion to this position,
    it is then known by the system. No change is done.

Field 'Pos Type' : This option button tells the system how to interpret the
    position value.
    ABS : the value is absolute
    otherwise the value is an offset to :
    REL : the current position
    LHW/UHW : the lower/upper HW limit position
    REF : the Reference Switch position
    IND : the Index Pulse position
    HOM : the Home position

Fields 'Value' and 'Unit' : define the position in encoder units 'Enc' or
    user defined position unit (configured as default position unit).
```

**CAUTIONS**

```
The configuration of these special positions is to be made carefully together
with the HW Initialization Sequence. All positions must be resolved at
completion of the HW Initialization in order to be allowed to reference
them later.
The table will be processed in the order shown in the panel:
this means that any position defined as offset to another assumes that
the latter is known by the system at this time.
For well-tuned systems, this panel allows to reduce the number of motions
necessary to make all special positions known to the system to a minimum.
E.g. knowing the physical distance between LHW and UHW leads to save a motion
to the UHW by defining UHW as LHW+offset and moving to LHW only.

The special positions may be referenced by the named positions or
during motions by value in position mode.
```

**EXAMPLES**

```
    Let's considere the following sequence:
    Action #1 : Find Lower HW Limit (LHW)
    Action #2 : Find the Reference Switch (REF)
```

```
Action #3 : Calibrate the encoder to 0 Enc
Action #4 : Move to absolute position +6000 Enc
Action #5 : END

Let's now considere the Special Positions:
Set LSW = LHW + 100Enc
Set USW = UHW - 200Enc
Don't set LHW
Set UHW = LHW + 10000Enc
Don't set REF
Don't set IND
Don't set HOM

and the Named Position:
RED = UHW - 3000Enc

At completion of the HW initialization sequence, the following positions
will be known and set in this order:
LHW = -1000Enc
UHW = +9000Enc
REF =     0Enc
LSW =  -900Enc
UHW = +8800Enc

The named position RED will as well be resolved as:
RED = +6000Enc

The actual position name shown in the status will be RED.

As one can see, it was not necessary to physically move to UHW in order
to resolve the named position RED.
```

___oOo___

- - - - - -
Last change:  30/09/98-15:01

### 5.2.12 moteiCHwInit(7)

**NAME**

```
moteiCHwInit - Configure HW Initialization Sequence
```

**EXTENDED HELP**

```
This panel provides configuration facilities for the HW init sequence.

The HW initialization sequence consist of a number of actions that are needed
by the system to be calibrated so as to be able to go ONLINE.
Typically, the sequence contains a number of motions dedicated to locate some
positions defined by the hardware such as switches or encoder signal edge.
In addition, it is used to move the motor to a known safe position.

Besides the common widgets, it contains a list box with the 4 associated
buttons, an option button and a configuration dependent part.

==============================================================================
HW Initialization Sequence:
===========================
List Box : at initialization, the first action is selected (highlighted).
    Mouse button 1 (left) is used to select the action.
Button 'Insert' : insert an END action above the selected one.
Arrow  'Up'     : exchange selected action with the one above (but first)
Arrow  'Down'   : exchange selected action with the one below (but last)
Button 'Delete' : delete selected action (but END).

Move Up, Move Down, Delete are only allowed when the current action or the
next (Move Down) is not END.
On Insert, Move Up, Move Down, Apply & Refresh, the current row is taken and
replaces the previous settings.

On selection, the option button 'Action Type' is updated and the configuration
dependent part is displayed accordingly.

Option Button 'Action Type' :
    Perform a motion : see Motion, Speed and Position Definitions below
    Set Lower SW limit : see Software Limits & Position Definition below
    Set Upper SW limit : see Software Limits & Position Definition below
    Calibrate the axis : assign a calibartion position to the actual
        encoder position. See Position Definition below
    Clamp the Brake : only if Brake Action signal is available
    Disconnect the Power Relays : only with VME-bus Amplifier.
    Execute a User-defined function : the sub-window asks then for the
        name of the function to invoke (see User Function below)
    Execute a On-Board Procedure : only on Motion Controller when supported
        the sub-window asks then for the index of the procedure to execute
    Delay : delay the sequence by the amount of milli-seconds as set.
    END : last step of the sequence. Extra actions will be ignored.

Important Note: since the purpose of the HW initialization sequence is to
    determine the position of relevant special positions, it is evident that
    the definition of target positions relative to a HW special position
    implies that this position is known to the system at this stage.
    In particular, this is true for named positions.

==============================================================================
Software Limits:
================
    The SW limits may be set during the HW initialization. However they will
be overwritten if they are defined in the main configuration panel
see Panel Configuration: section 'Axis', and moteiConfig(7).
When setting the SW limit, one must be sure that the motor is not outside the
```

```
valid domain (e.g. if the motor is at position LHW + 100Enc, it is illegal to
attempt to set the Lower SW limit to LHW + 200Enc; a failure would occur).

See Position Definition below.

==============================================================================
Motion Definition: description of the type of motion to perform
==================
    The two following selectors have to be understood as the descriptor of
    the type of motion. Speed and Position are consequently the physical
    description of the parameters needed to perform this kind of motion.

    Selector 'Mode' :
        Direct   : The motor is driven by velocity via the amplifier only.
            A motion consists in a Speed definition.
        Position : The motor is driven in velocity and position via the
            motion controller.
            A motion consists in a Speed and a Target definitions.
        Speed    : The motor is driven in velocity via the motion controller.
            A motion consists in a Speed definition.

    Selector 'Position Type' :
        to Absolute position : Position mode only.
            The target is an absolute position.
        Relative to current position : Position mode only.
            The target description is relative to the current position.
        to Lower HW limit : Direct & Speed modes only
        relative to Lower HW limit : Position mode only
            Direct mode: the motor is instructed to move in negative direction
                until detection of the Lower HW switch of the amplifier.
            Position mode : The target description is an offset to the
                physical position of the Lower HW limit.
            Speed mode: the motor is instructed to move in negative direction
                until detection of the Lower HW switch of the controller.
        to Upper HW limit : Direct & Speed modes only
        relative to Upper HW limit : Position mode only
            Direct mode: the motor is instructed to move in positive direction
                until detection of the Upper HW switch of the amplifier.
            Position mode : The target description is an offset to the
                physical position of the Upper HW limit.
            Speed  mode: the motor is instructed to move in positive direction
                until detection of the Upper HW switch of the controller.
        to Reference Switch : Speed mode only
        relative to Reference Switch : Position mode only
            Position mode : The target description is an offset to
                the physical position of the Reference switch.
            Speed  mode: the motor is instructed to move in positive
                direction until detection of the edge of the Reference switch.
        to Index Pulse : Speed mode only
        relative to Index Pulse : Position mode only
            Position mode : The target description is an offset to
                the physical position of the Index Pulse.
            Speed  mode: the motor is instructed to move in positive
                direction until detection of the index pulse of the encoder.
            The index pulse search is a 3 step intrisic motion:
            The input speed is applied until the pulse is detected,
            then the Index Search speed defined in the Configuration is
            applied to move in negative direction off the pulse.
            (see Config: MenuBar 'Speeds' -> 'Index Search')
            The third step is made using the Low Velocity parameter as
            configured for the Motion Controller MACCON MAC4.
            (see Config: MenuBar 'Hardware' -> 'Controller')
        to Home Position : Speed mode only
        relative to Home Position : Position mode only
            Position mode : The target description is an offset to
                the physical position of the Home position.
            Speed  mode: the motor is instructed to move in positive
                direction until detection of the next encoder index pulse
                following the edge of the Reference switch.
        in Two-Step Absolute/Relative: Position mode only
            The target position will be reached in two steps.
            (see Configuration Panel: MenuBar Positions->'Two Step Offset'
             Extended Help or moteiCTwoStp(7)).
        Off Reference Switch: Position mode only
            In some HW configurations, the location of the Reference switch
            (hereafter REF) is close to the UHW so that REF is active on
            UHW. In order not to perform any motion to the HW limits,
            one may want to detect only the position of the edge of REF and
            derive all other positions from this location. Therefore this type
            of motion checks if REF is active. When active, the system performs
            a motion in Position mode of the defined offset (negative) which
            must be set so as that REF goes inactive at motion completion.
            The next step should then be a motion in speed mode to REF.

See Position & Speed Definitions below.

==============================================================================
Speed Definition: description of the speed to apply during the motion
=================
    Selector 'How' :
        By Default : when configured, use this named speed
        By Value   : enter the speed value and select the speed unit
             as the board unit (Enc/ms) or as the user unit
             configured as the default unit.
             The speed maxima are not yet available
        By Name/Index: when configured, use one of the named speeds
             referenced by its name/index in the list.
        Named speeds & default speed :
        (see Config: MenuBar 'Speeds' -> 'Named Speeds', or moteiCNamSpd(7))

==============================================================================
Position Definition: SW limit, calibration or target position description
====================
Selector 'Position Type': indicates how the position must be interpreted
    Absolute : it is an absolute position
    Relative to current position: it is a positive offset to the actual
        position of the motor.
    Relative to Lower HW Limit: it is a positive offset to the position of
        the Lower HW limit.
    Relative to Upper HW Limit: it is a negative offset to the position of
        the Upper HW limit.
    Relative to Reference Switch: it is an offset to the position of
        the Reference Switch.
    Relative to Index Pulse: it is an offset to the position of
        the Index Pulse of the encoder.
    Relative to Home Position: it is an offset to the position of
        the Home Position.

    Selector 'How' :
        By Value   : enter the position/offset value and
             select the position unit as the board unit (Enc)
             or as the user unit configured as the default unit.
             The maxima are displayed in the selected unit
             whenever available; otherwise N/A.
        By Name/Index : Only for absolute position :
             when configured, use one of the named positions
```

```
               referenced by its name/Index in the list.
               The section is one of Observation, Maintenance or
               User-defined indicating the offset to apply relative
               to the nominal position (=Observation).
          By Name/Index + Offset : Only for absolute position
               Idem By Name, resp. By Index +
               Enter the offset value and select the position unit
               as the board unit (Enc) or as the user unit
               configured as the default unit.
          Named positions & Offsets:
          (see Config: MenuBar 'Positions' -> 'Named Positions')


==============================================================================
User Function:
==============
    The external function must comply with the interface motINIT_FUNC.
    It belongs to the application and must be found in the global symbol
    table of the LCU at HW INIT time.
    The address of the function will be retrieved from the symbol table.
    The user-buffer address is preset to 0x00000000.

    For setting the address of the user buffer, cf.
    - Panel Control: MenuBar 'Addresses'->'HW Initialization' or
    - moteiCSetAdr(7),
    - command SETADDR in motServer.CDT).

    See also MCM User Manual VLT-MAN-ESO-17210-0600 section 2.4.12 p.35ff


                                ___oOo___
```

```
- - - - - -
Last change:  30/09/98-15:01
```

### 5.2.13 moteiControl(7)

**NAME**

```
moteiControl - Motor Control
```

**EXTENDED HELP**

```
This panel contains all functions needed for the active control of a motor
on a LCU. It can only be invoked when the environment is refering to a LCU.

Since the number of combinations accessible with the panel is quite high,
the following description does not state whether a motion is possible or not
when the hardware configuration may affect the range of the possible motions.
E.g. if the encoder does not deliver the Index Pulse signal, it is evident
that motions to this position are not supported although the description
below may state it is allowed.

The behaviour of this panel is by default context-sensitive, i.e. the buttons,
entry fields and menu bar items are enabled/disabled according to the actual
state of the motor (e.g. the button MOVE is disabled when the Operational
state is not 'INIT DONE') and to the type of motion currently selected
(e.g. the part 'Position' is disabled for motion in Speed mode).
(see Menu Bar 'Options').
The STOP buttons are always enabled.

The panel is split in 4 parts:

==============================================================================
Motion : description of the type of motion to perform
========
    The two following selectors have to be understood as the descriptor of
    the type of motion. Speed and Position are consequently the physical
    description of the parameters needed to perform this kind of motion.

    Selector 'Mode' :
        Direct   : The motor is driven by velocity via the amplifier only.
            A motion consists in a Speed definition.
        Position : The motor is driven in velocity and position via the
            motion controller.
            A motion consists in a Speed and a Target definitions.
        Speed    : The motor is driven in velocity via the motion controller.
            A motion consists in a Speed definition.

    Selector 'Position Type' :
        ABS : Position mode only.
            The target is an absolute position.
        REL : Position mode only.
            The target description is relative to the current position.
        LHW : All motion modes
            Direct mode: the motor is instructed to move in negative direction
                until detection of the Lower HW switch of the amplifier.
            Position mode : The target description is an offset to the
                physical position of the Lower HW limit.
            Speed mode: the motor is instructed to move in negative direction
                until detection of the Lower HW switch of the controller.
        UHW : All motion modes
            Direct mode: the motor is instructed to move in positive direction
                until detection of the Upper HW switch of the amplifier.
            Position mode : The target description is an offset to the
                physical position of the Upper HW limit.
            Speed  mode: the motor is instructed to move in positive direction
                until detection of the Upper HW switch of the controller.
        REF : Position & Speed modes only
            Position mode : The target description is an offset to
```

```
                the physical position of the Reference switch.
            Speed  mode: the motor is instructed to move in positive
                direction until detection of the edge of the Reference switch.
        IND : Position & Speed modes only
            Position mode : The target description is an offset to
                the physical position of the Index Pulse.
            Speed  mode: the motor is instructed to move in positive
                direction until detection of the index pulse of the encoder.
            The index pulse search is a 3 step intrisic motion:
            The input speed is applied until the pulse is detected,
            then the Index Search speed defined in the Configuration is
            applied to move in negative direction off the pulse.
            (see Config: MenuBar 'Speeds' -> 'Index Search')
            The third step is made using the Low Velocity parameter as
            configured for the Motion Controller MACCON MAC4.
            (see Config: MenuBar 'Hardware' -> 'Controller')
        HOM : Position & Speed modes only
            Position mode : The target description is an offset to
                the physical position of the Home position.
            Speed  mode: the motor is instructed to move in positive
                direction until detection of the next encoder index pulse
                following the edge of the Reference switch.

    Check Button 'Two Step'  : Position Mode only
        When enabled, the motion is performed in 2 steps as configured
        (see Config: MenuBar 'Positions' -> 'Two Step Offset')

    Check Button 'Half-Turn' : Position Mode & Circular axis only

==============================================================================
Speed : description of the speed to apply during the motion
=======
    Selector 'How' :
        By Default : when configured, use this named speed
        By Value   : enter the speed value and select the speed unit
             as the board unit (Enc/ms) or as the user unit
             configured as the default unit.
             The speed maxima are not yet available
        By Name/Index: when configured, use one of the named speeds
             referenced by its name/index in the list.
        Named speeds & default speed :
        (see Config: MenuBar 'Speeds' -> 'Named Speeds')

==============================================================================
Position : description of the target position
==========
    Selector 'How' :
        By Value   : enter the target position/offset value and
             select the position unit as the board unit (Enc)
             or as the user unit configured as the default unit.
             The maxima are displayed in the selected unit
             whenever available; otherwise N/A.
        By Name/Index : Only for absolute target position :
             when configured, use one of the named positions
             referenced by its name/Index in the list.
             The section is one of Observation, Maintenance or
             User-defined indicating the offset to apply relative
             to the nominal position (=Observation).
        By Name/Index + Offset : Only for absolute target position
             Idem By Name, resp. By Index +
             Enter the offset value and select the position unit
             as the board unit (Enc) or as the user unit
             configured as the default unit.
        Named positions & Offsets:
        (see Config: MenuBar 'Positions' -> 'Named Positions')

==============================================================================
Status : display of the 3 main status items
========
    Operational Mode  : indicating the control mode of the motor
        DETACHED   : no control via MCM
        NOT AVAIL  : motor not available, all functions disabled
        FIXED      : all functions disabled, fixed position
        SIMULATION : all functions simulated (no HW access)
        NORMAL     : all functions enabled on HW
        HANDSET    : control via handset (no control via MCM)
    Operational State : indicating the current operational state
        NOT INIT   : the motor is not initialized
        INIT SOFT  : SW initialization done
        INIT RUN   : HW initialization in progress
        INIT DONE  : the motor is fully initialized, motions allowed
    Motion Status     : indicating the current state of the motion
        STANDING   : the motor is not moving, motion completed
        MOVING     : motion in progress
        ABORTED    : motion failed
        TIMEOUT    : motion did not complete within allowed time

==============================================================================

In addition the panel contains a Menu Bar and 3 action buttons:

==============================================================================
Action Buttons:
===============

Button INIT : Perform the initialization as specified in the Menu-Bar
    selector 'Initialization'. The Operational State is updated according
    to the initialization stage reached or in progress.

Button MOVE : Perform the motion a described in the parts Motion, Speed &
    Position.

Button STOP : Stop current motion. Button has little effect,
    if the motor is not moving.

==============================================================================
Menu Bar :
==========

Menu Bar 'Options'
------------------
    Command 'context sensitive' : when disabled, all widgets are enabled
        regardless of the actual state of the motor.

Menu Bar 'OpMode'
-----------------
    Selection of the Operational Mode.
    The Operational State of the motor is reset to 'Not Init'.

Menu Bar 'Initialization'
-------------------------
    Selection of the initialization procedure to apply:
    Option 'Full'     : both SW and HW initialization are performed.
        On success, the operational state goes to 'Init Done'.
    Option 'Software' : Only the software initialization is performed.
        On success, the operational state goes to 'Init Soft'.
    Option 'Hardware' : Only the hardware initialization is performed.
        The operational state must be at least 'Init Soft'.
        On success, the operational state goes to 'Init Done'.

Menu Bar 'Control'
------------------
    Command 'Clear Control'
        The command CLEAR <motor> is issued to the LCU process motServer
        that clears possible concurrent accesses to the motor.
    Command 'Update configuration'
        Together with the configuartion panel, in order to force the
        reinstallation of the motor and therefore the update of its
        configuration. The operational state goes to 'Not Init'.
    Command 'Power Control >' (on amplifier)
        Option 'Connect'    : Connects the power relays
        Option 'Disconnect' : Disconnects the power relays
    Command 'Control Loop >' (on controller)
        Option 'Enable'     : Enables the control loop
        Option 'Disable'    : Disables the control loop
    Command 'Reset >'
        Option 'Reset SW'   : idem Init SW
        Option 'Reset HW'   : issues a RESET command on the HW boards
    Command 'Brake Control >' (on digital I/O board)
        Option 'Clamp'      : sets the signal to clamp the brake
        Option 'Unclamp'    : resets the signal to umclamp the brake

Menu Bar 'Hardware'
-------------------
    Sub-panel 'Sampling ...'   :
        on-line setting of the status sampling period
        As indicated on the panel, the negative values have to be understood
        as the sampling frequency in Hertz (e.g. -10 = 10 Hz)
        The setting of the sampling period must be made carefully considering
        the overall load of the CPU (Retrieving the status of a motor takes
        approx. 10ms whereby 6 values must be read from the motion controller).

    Sub-panel 'Monitoring ...'
        when the speed is defined by value, it is needed to set
        the monitoring times for the 3 phases of the motion:
        far    : until the motor has reached the target window (TW)
        near   : from the moment on when the motor has reached the TW
        stable : delay before the controller issues the Motion End
             after the target window has been reached.
        These times are defaulted resp. to 1000, 200 and 1000ms.

    Sub-panel 'Controller ...'
        on-line setting/tuning of Motion Controller parameters
        Not available yet.
    Sub-panel 'Amplifier ...'
        on-line setting/tuning of Servo Amplifier parameters
        Not available yet.

Menu Bar 'Address'
------------------
    Sub-panel 'HW Initialization ...'
    Sub-panel 'Unit Conversion ...'
        When using user-defined function for Initialization and/or
        Unit conversion, one may have to give the physical address
        of the buffer to be passed to the function.
        For each function to be invoked, these panels show the current
        setting and by clicking on the requested function name they
        enable the setting of a new address.
```

```
==============================================================================
Emergency Stop:
===============
STOP sign button : acts as an Emergency Stop button.
        Any motion in progress is stopped immediately.


==============================================================================
Commands and Control outputs:
=============================
On the terminal from which motei has been launched, all commands sent to
<lcu>:motServer are echoed.
As well all relevant information
```

___oOo___

- - - - - -
Last change:  30/09/98-15:01

### 5.2.14 moteiCSetAdr(7)

**NAME**

```
moteiCSetAdr - Set User-Buffer address
```

**EXTENDED HELP**

```
    All external HW Initialization & Unit Conversion functions belong to the
    application and must be found in the global symbol table of the LCU.

    Since the API interface of the external user-hook functions foresees the
    passing of the address of a User parameter buffer, this panel allows the
    setting of the addresses for each function to be invoked.

    The addresses can be modified at any time.
    The NULL pointer (0x00000000) is always accepted, any other address must
    be valid and is tested for reading prior to access.

    See MCM User Manual VLT-MAN-ESO-17210-0600 section 2.4.12 p.35ff

==============================================================================
HW Initialization:
==================
    The external HW Initialization functions are resolved at HW INIT time.
    The user-buffer address is preset to 0x00000000.

==============================================================================
Unit Conversion:
================
    The external unit concersion functions are resolved at INSTALL time.
    The user-buffer address is preset to 0xFFFFFFFF.
    The application must explicitely set this parameter before proceeding
    with the initialization.

==============================================================================

The list box shows all functions and the associated addresses.
To modify and set a new address:
- click on the function (Mouse button 1 left)
- modify the address in the foreseen entry field
- click on button 'Set'

On success, the list will be updated accordingly.

Note: the addresses MUST be entered in HEXADECIMAL code w/o the leading 0x.

                                ___oOo___
```

```
- - - - - -
Last change:  30/09/98-15:01
```

### 5.2.15 moteiStatus(7)

**NAME**

```
moteiStatus - Motor Status
```

**EXTENDED HELP**

```
This panel displays continuously the complete status of the motor.
All status values are updated on change of the database attributes but the
actual values of Position/Speed/Current and Error, the latter are updated
on polling whereby the polling rate can be set from 1000ms in steps of 500ms
via the +/- buttons located at the top right corner of the panel; the actual
polling rate is displayed at their left hand side.

The status values are displayed in :
- black / green : for the normal behaviour
- red           : for abnormal states or preventing the motor to move.
- blue          : for some temporary states (e.g. on Reference Switch)

The panel is structured in 5 sections:

General Status:
===============
These values summarize the global state of the motor.
On the left side, the Operational Mode and the Operational State are displayed.
On the right side, the motion mode and status are displayed.
The steps of the HW initialization/motion sequence are shown. In addition,
the status indicator 'in Position' informs about the motion completion.

Board / Power Status:
=====================
In this section, all hardware indicators are gathered. They give an overview
of the state of all components.
If no brake is used, the Brake status is indicated as 'Unclamped'.
If no Interlock line is connected, the Interlock status is set to 'Unlocked'.
If a Stand-Alone Amplifier is used, the Power status is set to 'Connected' and
the Amplifier status to 'OK'.

Below the Amplifier and Controller status line, the position of the motor
relative to the HW special positions is indicated.

Motion Status & Plot:
=====================
This section is split in two parts: left the actual motor position, speed and
current; right the target position.
Depending on the default position unit, the position is indicated in both
Encoder unit (Enc) and default position unit.
The following error (difference between the theoretical and the physical
positions, whereby the theoretical is computed by the motion controller and
the physical is given by the encoder) is displayed in default position unit.
Whenever a name can be associated to a position, its label is displayed
together with the selected section.
The speed and current are displayed in default speed, resp. current units.

In addition, the button 'Plot' invokes a panel from which the user can select
the quantities to be plotted. See moteiSPlot(7), accessible also via the
Extended Help of this panel.

Times:
======
Each motion step is monitored in time. Its start and end UTC times are recorded
and displayed there. If a motion is performed and ends after 24:00 UTC,
the end time refers to the following day.
In addition, the motion duration is computed in seconds.
```

```
While moving, the end time is set to 00:00:00.000 UTC and
the duration to 'Undetermined'.

Events:
=======
All events occuring on the motor are listed in this widget. They show up
100ms longer than their real duration for visibility.

    Motion End        : Successfull motion completion
    Drive Fault       : Drive Fault signal raised
    Emergency Stop    : Emergency Stop signal active
    Over Temperature  : Amplifier temperature too high
    Over Current      : Too much current required from the amplifier
    Positioning Error : Motor could not follow the theoretical motion profile
    On Limit          : Motor on HW limit (Lower or Upper)
    Normal Stop       : Motor has been stopped
    Interlock Active  : Interlock signal active
    New Motion        : Motion has been redefined
    Board Init Ready  : On-Board HW Init procedure completed
```

___oOo___

```
- - - - - -
Last change:  30/09/98-15:01
```

### 5.2.16 moteiSVME4SA(7)

**This panel does not provide any on-line help, therefore no man-page has been issued.**
**Since the information shown in this panel reflects the status of the Servo Amplifier board VME4SA, the user is invited to refer to the associated User Manual.**

### 5.2.17 moteiSMAC4(7)

**This panel does not provide any on-line help, therefore no man-page has been issued.**
**Since the information shown in this panel reflects the status of the Motion Controller board MAC4, the user is invited to refer to the associated User Manual.**

### 5.2.18 moteiSPlot(7)

**NAME**

```
      moteiSPlot - Status Plotting Facility
```

**EXTENDED HELP**

```
      This panel interfaces motei with RtapPlotDisplay.

      It provides 4 selectors for the quantities to plot;
      all values expressed in default unit:
          - Position                        [STATUS.posUser]
          - Position (in encoder unit Enc)  [STATUS.posEnc]
          - Speed                           [STATUS.speed]
          - Error                           [STATUS.followingErr]
          - Current                         [STATUS.current]

      in addition, depending on the hardware configuration,
      and expressed in board units:
          from Motion Controller MAC4:
          - Position (CP)                   [SDL:MAC4:RUN.currentValues(64)]
          - Velocity (AV)                   [SDL:MAC4:RUN.currentValues(65)]
          - Error    (EP)                   [SDL:MAC4:RUN.currentValues(66)]
          from Servo Amplifier VME4SA:
          - Current  (MC)                   [SDL:VME4SA:RUN.currentValues(2)]

      The sampling frequency can be selected from 1 to 100Hz, whereby this setting
      is highly correlated with the status sampling period and the motion monitoring
      times.
      Higher sampling frequency has little effect on the plot precision, however it
      leads to an increase of the message load between WS and LCU.

      The two buttons Start and Stop initiates/stops the plotting.
      The option 'Save in file' dumps all values to the ASCII file '<motorName>.samp'
      located under $INS_ROOT/SYSTEM/DETDATA, if $INS_ROOT exists;
      otherwise under /tmp.
```

___oOo___

```
- - - - - -
Last change:  30/09/98-15:01
```

## 5.3 Database template files

### 5.3.1 moteiDATABASE.db(5)

**NAME**

```
moteiDATABASE.db - Database for MOTEI
```

**SYNOPSIS**

```
#include "moteiDATABASE.db"
```

**PARENT CLASS**

```
BASE_CLASS
```

**DESCRIPTION**

```
This file includes one instance of each base motor class
defined in motMOTORS.class:
  motDVA0X, motDVAMI, motDVAMA, motDXXMI, motDXXMA & motSVSMS

The motor branches are located under:
  :Appl_data:MOTEI   on WS
  :MOTEI             on LCU

The creation of this database branch is intended to provide standard
motor branches for motei users who do not have yet defined the motors
to be used in their application, but solely preparing/testing motor
configuration for future use.

All motors share the following characteristics:
  Number of Named Positions          : motMAX_NAMED_POSITION
  Number of Named Speeds             : motMAX_NAMED_SPEED
  Number of HW Init Actions          : motMAX_INIT_ACTION
  First HW Init Action               : motACTION_END
  Number of Unit Conversions methods : motMAX_CONV_METHOD
and all preset values as defined in the class definition.
```

**EXAMPLES**

```
Database of LCU lted : $VLTDATA/ENVIRONMENTS/lted/dbl/DATABASE.db
//********************
// VLT COMMON SOFTWARE
//********************
// Loads classes definition from standard file
#include "CCS.db"
// USER.db contains necessary points for CCS, but requiring editing to
// match installed configuration.
#include "USER.db"
//*********************
// APPLICATION SOFTWARE
//*********************
#include "moteiDATABASE.db"
// ___oOo___
```

**SEE ALSO**

```
motei*.db(5), mot*.class(5), motor.db(5)
```

```
- - - - - -
Last change:  30/09/98-15:01
```

### 5.3.2 moteiDVA0X.db(5)

**NAME**

```
      moteiDVA0X.db - Instance for DC motor (VME4SA-X1 only)
```

**SYNOPSIS**

```
      #include "moteiDVA0X.db"
```

**PARENT CLASS**

```
      motDVA0X
```

**DESCRIPTION**

```
      This file inserts an attribute named DVA0X instance of the motor
      class motDVA0X and aliased DVA0X.
```

**EXAMPLES**

```
      see moteiDATABASE.db(5)
```

**SEE ALSO**

```
      moteiDATABASE.db(5), motDVA0X.class(5), motor.db(5)
```

```
- - - - - -
Last change:  30/09/98-15:01
```

### 5.3.3 moteiDVAMI.db(5)

**NAME**

```
      moteiDVAMI.db - Instance for DC motor (VME4SA-X1 + MAC4-INC)
```

**SYNOPSIS**

```
      #include "moteiDVAMI.db"
```

**PARENT CLASS**

```
      motDVAMI
```

**DESCRIPTION**

```
      This file inserts an attribute named DVAMI instance of the motor
      class motDVAMI and aliased DVAMI.
```

**EXAMPLES**

```
      see moteiDATABASE.db(5)
```

**SEE ALSO**

```
      moteiDATABASE.db(5), motDVAMI.class(5), motor.db(5)
```

```
- - - - - -
Last change:  30/09/98-15:01
```

### 5.3.4 moteiDVAMA.db(5)

**NAME**

```
      moteiDVAMA.db - Instance for DC motor (VME4SA-X1 + MAC4-SSI)
```

**SYNOPSIS**

```
      #include "moteiDVAMA.db"
```

**PARENT CLASS**

```
      motDVAMA
```

**DESCRIPTION**

```
      This file inserts an attribute named DVAMA instance of the motor
      class motDVAMA and aliased DVAMA.
```

**EXAMPLES**

```
      see moteiDATABASE.db(5)
```

**SEE ALSO**

```
      moteiDATABASE.db(5), motDVAMA.class(5), motor.db(5)
```

```
- - - - - -
Last change:  30/09/98-15:01
```

### 5.3.5 moteiDXXMI.db(5)

**NAME**

```
      moteiDXXMI.db - Instance for DC motor (MAC4-INC, Stand-Alone amplifier)
```

**SYNOPSIS**

```
      #include "moteiDXXMI.db"
```

**PARENT CLASS**

```
      motDXXMI
```

**DESCRIPTION**

```
      This file inserts an attribute named DXXMI instance of the motor
      class motDXXMI and aliased DXXMI.
```

**EXAMPLES**

```
      see moteiDATABASE.db(5)
```

**SEE ALSO**

```
      moteiDATABASE.db(5), motDXXMI.class(5), motor.db(5)
```

```
- - - - - -
Last change:  30/09/98-15:01
```

### 5.3.6 moteiDXXMA.db(5)

**NAME**

```
moteiDXXMA.db - Instance for DC motor (MAC4-SSI, Stand-Alone amplifier)
```

**SYNOPSIS**

```
#include "moteiDXXMA.db"
```

**PARENT CLASS**

```
motDXXMA
```

**DESCRIPTION**

```
This file inserts an attribute named DXXMA instance of the motor
class motDXXMA and aliased DXXMA.
```

**EXAMPLES**

```
see moteiDATABASE.db(5)
```

**SEE ALSO**

```
moteiDATABASE.db(5), motDXXMA.class(5), motor.db(5)
```

```
- - - - - -
Last change:  30/09/98-15:01
```

### 5.3.7 moteiSVSMS.db(5)

**NAME**

```
moteiSVSMS.db - Instance for Stepper motor (VME4ST + MAC4-STP)
```

**SYNOPSIS**

```
#include "moteiSVSMS.db"
```

**PARENT CLASS**

```
motSVSMS
```

**DESCRIPTION**

```
This file inserts an attribute named SVSMS instance of the motor
class motSVSMS and aliased SVSMS.
```

**EXAMPLES**

```
see moteiDATABASE.db(5)
```

**SEE ALSO**

```
moteiDATABASE.db(5), motSVSMS.class(5), motor.db(5)
```

```
- - - - - -
Last change:  30/09/98-15:01
```

**___oOo___**