

TESCAN Scanning Electron Microscope

SharkSEM Remote Control

FIB Extension

Revision 2.0.10
25th July 2012

SEM Models: Vela 3, Lyra 3, Fera 3

TESCAN, a.s.
Brno, Czech Republic



Copyright 2012, TESCOAN, a.s.
Libušina tř. 21, Brno, Czech Republic
Phone +420 547 130 411
info@tescan.cz

Table of Contents

Preface	4
SharkSEM Modifications	5
FIB Optics	6
FibEnumOptPars	6
FibGetOptParam.....	6
FibGetViewField.....	7
FibReadFCCurr	7
FibSetViewField.....	8
FIB Input Channels and Detectors	9
FibDtAutoSig	9
FibDtEnable	9
FibDtGetChann	10
FibDtGetGainBl	10
FibDtSetGainBl	11
FIB Scanning.....	12
FibScEnumSpeeds.....	12
FibScGetExtern	12
FibScGetSpeed	13
FibScScanXY	13
FibScSetExtern.....	15
FibScSetSpeed.....	15
FibScStopScan	16
FIB High Voltage	17
FibHVGetBeam.....	17
FibHVGetFilTime	17
FibHVGetVoltage	18
FIB Presets	19
FibEnumPresets.....	19
FibSetPreset.....	19
FIB Centering	21
FibEnumCent	21
FibGetCent	21
FibSetCent.....	22
FIB Image Geometry.....	23
FibEnumGeom	23
FibGetGeom.....	24
FibSetGeom.....	24

Preface

All TESCAN Scanning Electron Microscopes (SEMs) are equipped with remote control capability. It can be used by remote applications like an EDX system, or lithography system or other custom application.

The remote control interface is referred as SharkSEM Remote Control.

This document describes interface for controlling the FIB (Focused Ion Beam) column, a part of all Lyra, Fera, Vela systems.

For information about the SharkSEM protocol, refer to the main document. This document only describes the extension functions.

SharkSEM Modifications

The SharkSEM protocol for Vela, Lyra, Fera is binary compatible with the SharkSEM protocol present on tungsten or FE SEM.

Except for the new set of functions, there is only one important change. In the *Message header*, field *Request flags*, there are three new wait flags:

- Bit 12 Wait E
- Bit 13 Wait F
- Bit 14 Wait G

Wait E – FIB scanning. Set to 1 while the FIB is scanning.

Wait F – FIB optics change in progress. Set to 1 if the FIB optics is not in stable condition. This flag is intended for future extensions, currently it is not used.

Wait G – FIB automatic procedure in progress. Set to 1 during run of any automatic procedure.

FIB Optics

FibEnumOptPars

Get list of all available optical parameter.

Arguments

map **FibEnumOptPars**(void)

return value list of optical parameters

Remarks

The *map* has the following form:

```
param.0.name  
param.0.count  
param.0.unit  
param.1.name  
...  
param.X.name  
param.X.count  
param.X.unit
```

The key is separated by two dots. The middle number is an index of the parameter. Application should not expect that the indexes form a consecutive sequence. The index can be any integer number.

name	human-readable name.
count	1 or 2 (one if only the first value is used, two otherwise)
unit	%, mm, deg, or other physical unit

For future compatibility, application must silently ignore any *key* which does not have exactly the above form.

Call Context

Anytime.

Compatibility

1.x.x	2.x.x
1.0.10 and later	yes

FibGetOptParam

Get FIB optical parameters.

Arguments

void FibGetOptParam(in int index, out float x, out float y)

index index of the parameter – see *FibEnumOptPars()*
x, y one or two values

Remarks**Call Context**

Anytime.

FibGetViewField

Get FIB view field.

Arguments

float FibGetViewField(void)

return value view field in [mm]

Remarks**Call Context**

Anytime.

Compatibility

1.x.x	2.x.x
1.0.10 and later	yes

FibReadFCCurr

Measure FIB beam current in the Faraday cup.

Arguments

float FibReadFCCurr(void)

return value Faraday cup current in [pA]

Remarks**Call Context**

Scanning must be inactive.

Compatibility

1.x.x	2.x.x
1.0.10 and later	yes

FibSetViewField

Set FIB view field.

Arguments

*void **FibSetViewField**(in float vf)*

vf view field in [mm]

Remarks

If the requested value is out of range, it is silently limited.

Call Context

Anytime.

Compatibility

1.x.x	2.x.x
1.0.10 and later	yes

FIB Input Channels and Detectors

The FIB uses currently just secondary electron detector (SE). Wherever *channel* or *detector* field is required, its value must be zero.

For above reason, function equivalent to SEM *DtSelect()* is not present, because there are only single detector and single image channel.

This may change in the future, when FIB imaging will support more than one detector.

FibDtAutoSig

Invoke automatic brightness/contrast adjustment procedure.

Arguments

void FibDtAutoSig(in int channel)

channel

input video channel

Timing

Execution time may vary between 1 and 10 seconds (wait G).

Remarks

Call Context

Scanning must be inactive.

Compatibility

1.x.x	2.x.x
1.0.10 and later	yes

FibDtEnable

Enable/disable FIB input video channel.

Arguments

void FibDtEnable(in int channel, in int enable[, in int bpp])

channel

input video channel index

enable

0 – disable, 1 – enable

bpp

bits per pixel (optional)

8 or 16, default is 8

Remarks

Call Context

Scanning must be inactive.

FibDtGetChann

Get number of FIB input video channels.

Arguments

int **FibDtGetChann**(*void*)

return value

number of available input video channels

Remarks**Call Context**

Anytime.

Compatibility

1.x.x	2.x.x
1.0.10 and later	yes

FibDtGetGainBl

Get detector gain and black level.

Arguments

void **FibDtGetGainBl**(*in int detector, out float gain, out float black*)

detector

detector index

gain

gain (contrast) [percent]

black

black level (brightness) [percent]

Remarks**Call Context**

Anytime

Compatibility

1.x.x	2.x.x
1.0.10 and later	yes

FibDtSetGainBl

Set detector gain and black level.

Arguments

*void **FibDtSetGainBl**(in int detector, in float gain, in float black)*

detector

detector index

gain

gain (contrast) [percent]

black

black level (brightness) [percent]

Remarks

Call Context

Anytime

Compatibility

1.x.x	2.x.x
1.0.10 and later	yes

FIB Scanning

FibScEnumSpeeds

Get list of available FIB scanning speeds.

Arguments

map ***FibScEnumSpeeds***(void)

return value scanning speed list

Remarks

Map of speeds is returned.

The map *key* has the following form:

speed.1.dwell

speed.2.dwell

...

speed.N.dwell

Where **N** is the total number of speed indexes.

The map *value* contains **float** value describing pixel dwell time in microseconds for given speed index.

For future compatibility, application must silently ignore any *key* which does not have exactly the above form.

Call Context

Anytime

Compatibility

1.x.x	2.x.x
1.0.10 and later	yes

FibScGetExtern

Get FIB external scanning status.

Arguments

int ***FibScGetExtern***(void)

return value external scanning status

Timing

Executed immediately.

Remarks

External scanning can either be ON or OFF:

0 – external scanning is OFF.

1 – external scanning is ON.

See *FibScSetExtern()* for details.

Call Context

Anytime

Also Affected

-

Compatibility

1.x.x	2.x.x
no	2.0.5 and later

FibScGetSpeed

Get current FIB scanning speed index.

Arguments

int ***FibScGetSpeed***(void)

return value

scanning speed index

Remarks

Call Context

Anytime.

Compatibility

1.x.x	2.x.x
1.0.10 and later	yes

FibScScanXY

Start FIB scanning over rectangular area.

Arguments

int ***FibScScanXY***(
 in unsigned int frameid,
 in unsigned int width,

*in unsigned int height,
in unsigned int left,
in unsigned int top,
in unsigned int right,
in unsigned int bottom,
in int single,
[in unsigned dwell]*

);

return value	0 – ok, < 0 – failed, invalid parameters
frameid	unique frame id sent in the data callback
width, height	dimensions of the whole scanning window
left, top, right, bottom	definition of the visible region
single	0 – continual scanning, 1 – single frame
dwell	pixel dwell time in [ns]

*void **FibScData**(
in unsigned int frameid,
in int channel,
in unsigned int index,
in int bpp,
in char[] data*

);

frameid	frame id
channel	channel index
index	index of the first pixel in the data buffer
bpp	bits per pixel (8 / 16)
data	image data buffer

Remarks

Refer to *ScScanXY()*, in the main SharkSEM documentation, *FibScScanXY()* has almost the same behavior.

If the *dwell* parameter is omitted, current scanning speed is used instead. Refer to *FibScSetSpeed()*.

Note that the FIB scanning and the SEM scanning are fully independent.

Call Context

Scanning must be inactive.

Compatibility

1.x.x	2.x.x
1.0.10 and later	yes

The optional *dwell* argument is supported since version 2.0.6.

FibScSetExtern

Enable/disable FIB external scanning.

Arguments

void FibScSetExtern(in int enable)

enable

enable flag

Timing

Executed immediately.

Remarks

This function is typically used by an external pattern generator.

If external scanning is ON, FIB scanning amplifiers switch their input to external scanning ramp.

External scanning can either be ON or OFF:

0 – external scanning is OFF.

1 – external scanning is ON.

This function controls simple switching circuit (multiplexer). If *FibScSetExtern(1)* is executed, external ramp input is activated. If internal scanning is active at this moment, it is stopped. If *FibScSetExtern(0)* is executed, internal scanning input is activated but scanning remains stopped.

Besides the scanning ramp input, this function also selects the beam blanker input which corresponds to the active scanning ramp input.

Call Context

Anytime

Also Affected

-

Compatibility

1.x.x	2.x.x
no	2.0.5 and later

FibScSetSpeed

Set current FIB scanning speed index.

Arguments

void FibScSetSpeed(in int speed)

speed

scanning speed index

Remarks

Call Context

Anytime.

Compatibility

1.x.x	2.x.x
1.0.10 and later	yes

There is a difference between 1.x.x and 2.x.x protocol versions. In the 2.x.x version, the scanning speed is not changed if scanning is in progress. Scanning must be restarted to apply the new scanning speed value.

FibScStopScan

Stop FIB scanning.

Arguments

void FibScStopScan(void)

Remarks

If scanning was initialized using one of the scanning functions, it must be stopped later using *FibScStopScan()*. It does not matter if single scan or continual scanning was requested.

It is not guaranteed that the image data has already been sent at the moment this call is completed. Application must use the frame id identifier to distinguish which frame the image data belongs to.

Call Context

Scanning must be active when calling this function.

Compatibility

1.x.x	2.x.x
1.0.10 and later	yes

FIB High Voltage

Following set of functions is intended for main accelerating voltage control.

Currently there are no control functions, just read access is provided.

FibHVGetBeam

Get the beam status.

Arguments

int FibHVGetBeam(void)

return value	0	beam is off
	1	beam is on

Timing

Executed immediately.

Remarks

Determine the current beam status.

Call Context

Anytime

Also Affected

-

Compatibility

1.x.x	2.x.x
no	2.0.4 and later

FibHVGetFilTime

Get the amount of consumed filament since the last exchange.

Arguments

float FibHVGetFilTime(void)

return value	consumed filament [As]
--------------	------------------------

Timing

Executed immediately.

Remarks

The amount is not specified in terms of time, but rather as a product of time and emission current.

Call Context

Anytime

Also Affected

-

Compatibility

1.x.x	2.x.x
no	2.0.4 and later

FibHVGetVoltage

Read current accelerating voltage.

Arguments

float ***FibHVGetVoltage***(*void*)

return value

accelerating voltage [V]

Timing

Executed immediately.

Remarks

Returns current FIB accelerating voltage. In conjunction with *FibHVGetBeam()*, client application can determine if the FIB beam is ready for operation.

Call Context

Anytime

Also Affected

-

Compatibility

1.x.x	2.x.x
no	2.0.4 and later

FIB Presets

FIB preset is a group of FIB parameters (e.g. aperture, centering, condenser excitation, working distance). The presets are defined by user in the FIB GUI. Each preset is identified by unique, user-defined name. SharkSEM can enumerate the presets and apply them.

FibEnumPresets

Get list of all available FIB presets.

Arguments

char[] **FibEnumPresets**(void)

return value

string containing list of presets

Timing

Executed immediately.

Remarks

The returned string contains list of preset names. The string is divided into lines, there is one preset name per line. If there are no presets, the string is empty.

Call Context

Anytime

Compatibility

1.x.x	2.x.x
no	2.0.10 and later

FibSetPreset

Activate preset.

Arguments

void **FibSetPreset**(*char[]* preset)

preset

name of the preset to be activated

Timing

This call can take arbitrarily long time, depending on the previous state of the FIB subsystem. Before further processing, check FIB wait flags using *IsBusy()* call, or use conditional command execution.

Remarks

The preset table is searched for the specified preset name. If it is found, the preset is activated. If the preset is not found, the request is ignored.

Call Context

No FIB automatic procedure must be running, the FIB optics must be stable.

Compatibility

1.x.x	2.x.x
no	2.0.10 and later

FIB Centering

FIB column centering allows enumeration of column centering parameters and r/w access to their values.

Note: currently there are no centering parameters accessible. The functions in this chapter will respond correctly, but no actual parameters are defined.

FibEnumCent

Get list of all available column centering parameters.

Arguments

map **FibEnumCent**(void)

return value

list of centering parameters

Timing

Executed immediately.

Remarks

See also *FibGetCent()*, *FibSetCent()*.

Call Context

Anytime.

Compatibility

1.x.x	2.x.x
no	2.0.10 and later

FibGetCent

Get FIB column centering parameter.

Arguments

void **FibGetCent**(in int index, out float x, out float y)

index

index of the parameter – see *FibEnumCent()*

x, y

centering values in X and Y axis

Timing

Executed immediately.

Remarks

See also *FibEnumCent()*, *FibSetCent()*.

Call Context

Anytime.

Compatibility

1.x.x	2.x.x
no	2.0.10 and later

FibSetCent

Set FIB column centering parameter.

Arguments

void FibSetCent(in int index, in float x, in float y)

index	index of the parameter – see <i>FibEnumCent()</i>
x, y	centering values in X and Y axis

Timing

Variable execution time, *Wait F* flag is set.

Remarks

See also *FibEnumCent()*, *FibGetCent()*.

Call Context

Anytime

Also Affected

-

Compatibility

1.x.x	2.x.x
no	2.0.10 and later

FIB Image Geometry

FIB scanning (imaging) allows several image transformations. The most common are image shift and image rotation.

FibEnumGeom

Get list of all available image geometry transformations.

Arguments

map **FibEnumGeom**(void)

return value

list of geometric transformations

Timing

Executed immediately.

Remarks

Generally, the *map* looks like:

```
geom.0.name  
geom.0.count  
geom.0.unit  
geom.1.name  
...  
geom.X.name  
geom.X.count  
geom.X.unit
```

The number in the middle is an index of the parameter. Application should not expect that the indexes form a consecutive sequence. The index can be any integer number.

name human-readable name.

count 1 or 2 (one if only the first value is used, two otherwise)

unit %, mm, deg, or other physical unit

For future compatibility, application must ignore any unknown *key*.

As of version 2.0.10, following transformations are defined:

```
geom.0.name=Image Shift  
geom.0.count=2  
geom.0.unit=mm  
geom.1.name=Image Rotation  
geom.1.count=1  
geom.1.unit=deg
```

Image Shift is offset in the scanning coils. Its range is usually only +/- few μm .

Image Rotation range is 0 degrees to 360 degrees, the positive orientation is counterclockwise.

See also *FibGetGeom()*, *FibSetGeom()*.

Call Context

Anytime.

Compatibility

1.x.x	2.x.x
no	2.0.10 and later

FibGetGeom

Get FIB geometry transformation.

Arguments

void FibGetGeom(in int index, out float x, out float y)

index	index of the parameter – see <i>FibEnumGeom()</i>
x, y	geometry values in X and Y axis

Timing

Executed immediately.

Remarks

See also *FibEnumGeom()*, *FibSetGeom()*.

Call Context

Anytime.

Compatibility

1.x.x	2.x.x
no	2.0.10 and later

FibSetGeom

Set FIB geometry transformation.

Arguments

void FibSetGeom(in int index, in float x, in float y)

index	index of the parameter – see <i>FibEnumGeom()</i>
x, y	geometry values in X and Y axis

Timing

Variable execution time, *Wait F* flag is set.

Remarks

See also *FibEnumGeom()*, *FibGetGeom()*.

Call Context

Anytime

Also Affected

-

Compatibility

1.x.x	2.x.x
no	2.0.10 and later