SCANLAB
innovators for industry

excelli*SCAN* Scan Heads – Functional Principle of *SCAN*ahead Servo Control and Operation by RTC®6 Boards

SCANLAB AG
Siemensstr. 2a
82178 Puchheim
Germany

Tel. +49 (89) 800 746-0
Fax: +49 (89) 800 746-199

info@scanlab.de
www.scanlab.de

(Doc. Rev. 1.0.0 e - February 29, 2016)

SCANLAB
innovators for industry

Contents

1 Overview

1.1 Introduction

Because SCANLAB excelli*SCAN* scan heads use *SCAN*ahead servo control, they exhibit different dynamics than scan heads with conventional control.

In particular, you need to consider timing differences when synchronously operating your scan head, laser and peripheral equipment.

Here, the RTC®6 board command set provides special *SCAN*ahead commands that considerably simplify adapting of existing RTC applications (for conventional scan heads) into programs for excelli*SCAN* scan heads with *SCAN*ahead control. Basically, just add two lines of code to your existing RTC®6 program:

```
set_scanahead_params( 1, 1, 1, 0, 0, 0 );
activate_scanahead_autodelays( 1 );
```

All details about these and other commands are described in this manual.

1.2 About This Manual and Other Documents

This manual describes user-relevant differences between excelli*SCAN* scan heads with SCANahead servo control and scan heads with conventional control. And it describes all special RTC®6 commands reserved exclusively for operating *SCAN*ahead-based excelli*SCAN* scan heads.

Thus, this document expands upon the RTC®6 and RTC®5 manuals. To program complete RTC®6 applications for *SCAN*ahead-based excelli*SCAN* scan heads, you have to observe all three manuals. Also observe the notes in your scan head manual.

Notes

- All references to the RTC®5 manual refer to Doc. Rev. 1.10 e.

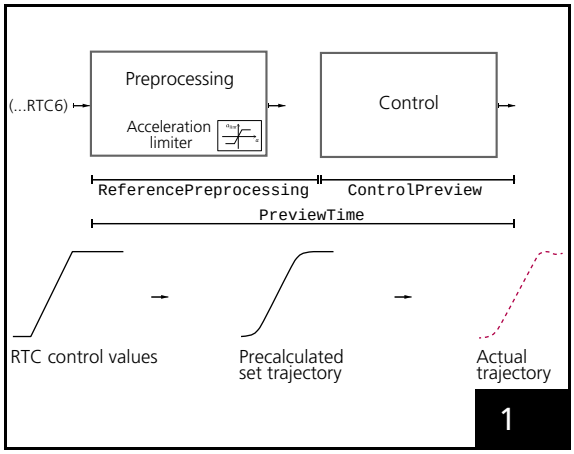
2 excelli*SCAN* Scan Heads with *SCAN*ahead Servo Control

2.1 Introduction

excelli*SCAN* scan heads are equipped with *SCAN*ahead servo control. This innovative control concept(1):

- Produces no tracking error and
- Exploits the maximum acceleration capability of the galvanometer scanners.

Figure 1 shows how *SCAN*ahead servo control is structured.



Structure of *SCAN*ahead servo control (timing depictions are not true-to-scale)

The RTC®6 board puts out RTC control values. The preprocessing unit analyzes associated theoretical accelerations. These are then limited to match the scan system's actual acceleration capabilities.

The result is a precalculated set trajectory of acceleration-limited control values.
The required time for this is depicted as `ReferencePreprocessing` in figure 1 and figure 4.

The galvanometer scanner control successfully traverses the precalculated set trajectory (actual trajectory in figure 1) – but only with a temporal offset (the control preview time `ControlPreview`, see figure 1 and figure 4). This enables full usage of scan system dynamics.

The entire temporal offset between RTC control values and galvanometer scanner motions (that is, `ReferencePreprocessing` plus control precalculation time `ControlPreview`) is called `PreviewTime`. The term `PreviewTime` highlights the excelli*SCAN* scan head's need to know the RTC control values in advance by this amount of time to ensure punctual execution.

For synchronous laser control, the laser control signals likewise need to take `PreviewTime` into account. This also applies for any other control signals intended to be transmitted synchronously with scanner motion.

The RTC®6 can ensure synchronous operation of galvanometer scanner motions, laser control, and other peripherals.

The precalculated set trajectories with limited, constant set accelerations cause acceleration time to vary in accordance with speed changes. This means scanner delays and laser delays needed for taking acceleration time into account are also dependent on speed.

The RTC®6 board can automatically calculate the required delays. Therefore, SCANLAB recommends the RTC®6 for operating excelli*SCAN* scan heads with *SCAN*ahead technology, see chapter 3.1 "Introduction", page 16.

(1) Thus, excelli*SCAN* scan heads cannot be operated like conventional scan heads.

2.2 Comparing excelli*SCAN* Scan Heads to Scan Heads with Conventional Servo Control (for example, intelli*SCAN*®)

2.2.1 What is Unchanged for excelli*SCAN* Scan Heads?

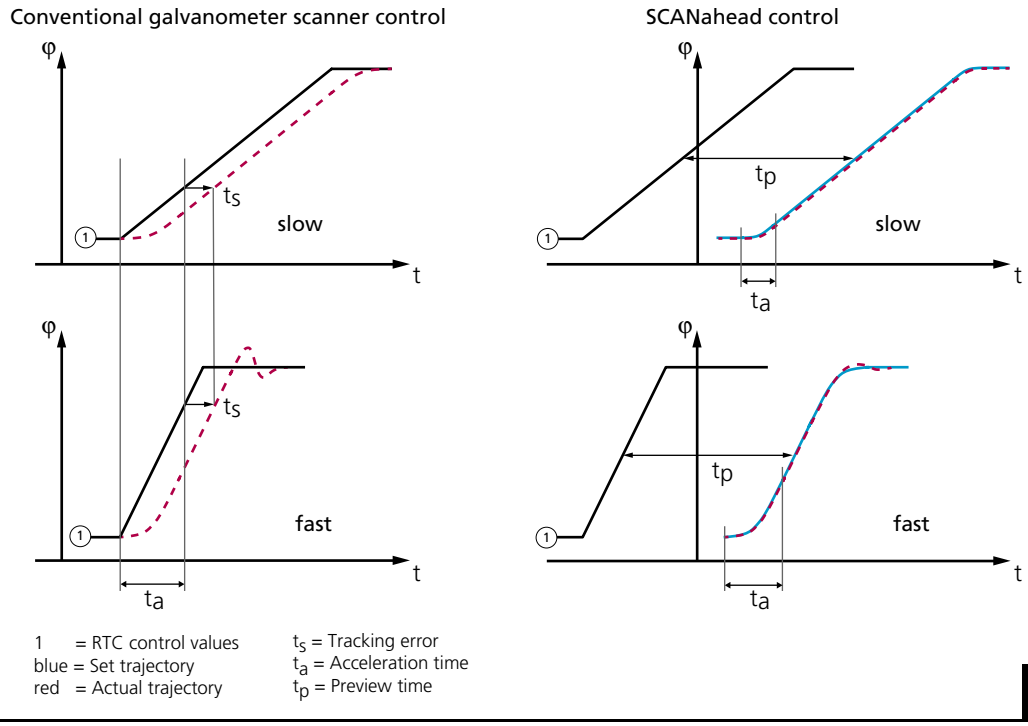
What is described in the RTC®5 manual chapter 7.2 "Delay Settings for Synchronizing Scan Head and Laser Control", page 108 also applies when controlling an excelli*SCAN* scan head by an RTC®6 board:

- The galvanometer scanner motions are temporally offset from the RTC control values by the `PreviewTime` (not: tracking error!). For details, see the table in chapter 2.2.2 "Comparing Scan Heads with and without SCANahead Technology", page 7.
- The temporal stream of actual values (=actual trajectory) is smoothed in comparison to the stream of RTC control values(1), due to the limited acceleration potential of galvanometer scanners.
- To compensate the temporal offset between RTC control values and actual trajectories when synchronizing galvanometer scanner motions and laser signals, the RTC®6 allows setting of scanner and laser delays.
  The meaning of delays for the excelli*SCAN* is the same as for scan systems without *SCAN*ahead technology (and identical for both the RTC®6 and RTC®5 boards). The same applies for the commands **set_scanner_delays** and **set_laser_delays** to set the delays.
- The preferable alternative is to let the RTC®6 board automatically calculate and set scanner delays and laser delays, see chapter 3.1 "Introduction", page 16.

(1) Thus any curves within the control value trajectory get reduced.

2.2.2 Comparing Scan Heads with and without *SCAN*ahead Technology

*SCAN*ahead technology makes excelli*SCAN* scan heads behave differently in some respects than scan heads without *SCAN*ahead technology. These differences are contrasted in figure 2 and the following table.



Comparison: Conventional galvanometer scanner servo control and *SCAN*ahead servocontrol control

Scan Head without *SCAN*ahead Technology	Scan Head with *SCAN*ahead Technology, for example, excelli*SCAN*
• Scan heads without *SCAN*ahead technology exhibit a characteristic temporal offset called tracking error t_s, see figure 2. The tracking error t_s results from the implemented servo control structure. The tracking error duration t_s is a metric for the dynamics of a scan head. The tracking error t_s is (at "normal" speeds) practically independent of scan speed. It is also affected by the selected tuning.	• For scan heads with *SCAN*ahead technology, the temporal offset is not tracking error t_s. Instead it is the fixed, pre-defined precalculation time `PreviewTime` t_p, see figure 2. The precalculation time `PreviewTime` t_p is qualitatively different from tracking error t_s. And it is not a metric for scan head dynamics. Instead, a physically traversable trajectory gets precalculated within the `PreviewTime` t_p.

Scan Head without *SCAN*ahead Technology (Continued)	Scan Head with *SCAN*ahead Technology, for example, excelli*SCAN* (Continued)
• Tracking error t_s can introduce undesirable artifacts during marking, for example, "necking" in circles and arcs. You can partially compensate such artifacts by specifying over-dimensioned set radii.	• Vectors can be traversed even at very high speeds. Necking artifacts during circle/arc marking only start to occur at accelerations exceeding `Amax`(a). Therefore, these necking artifacts can *not* be compensated by specifying larger set radii!
• For tracking error t_s, a tolerance (maximum value) is specified (tested by SCANLAB). But the exact value can be serial-number-dependent (minor variation within the tolerances are possible). The exact value cannot be queried, and must instead be experimentally determined.	• The `PreviewTime` t_p is a serial-number-dependent value. It is permanently stored in the firmware and can be queried from the scan head.
• The trajectory of RTC control values gets smoothed by servo control. The result of smoothing is not precisely known in advance. And it is dependent on the selected tuning. In negative acceleration phases, significant undesirable overshoot may occur.	• From the trajectory of RTC control values, a traversable acceleration-limited set value trajectory gets calculated and transmitted to the servo control, see figure 1. Deviation of the final actual value trajectory from the RTC control value trajectory is therefore known in advance (because it was precalculated).
• *The duration of acceleration phases is practically speed-independent*. Thus the same amount of time is always required to reach the desired target speed. This means that, particularly at low speeds, the acceleration potential is not fully exploited and the process times of applications are therefore not optimal.	• *SCAN*ahead control ensures constant acceleration (at the maximum capabilities of th scan head) in acceleration phases. *The duration of acceleration phases is thus minimized, although speed-dependent*. Resultingly, scanner delays and laser delays need adjusting in accordance with marking speed.

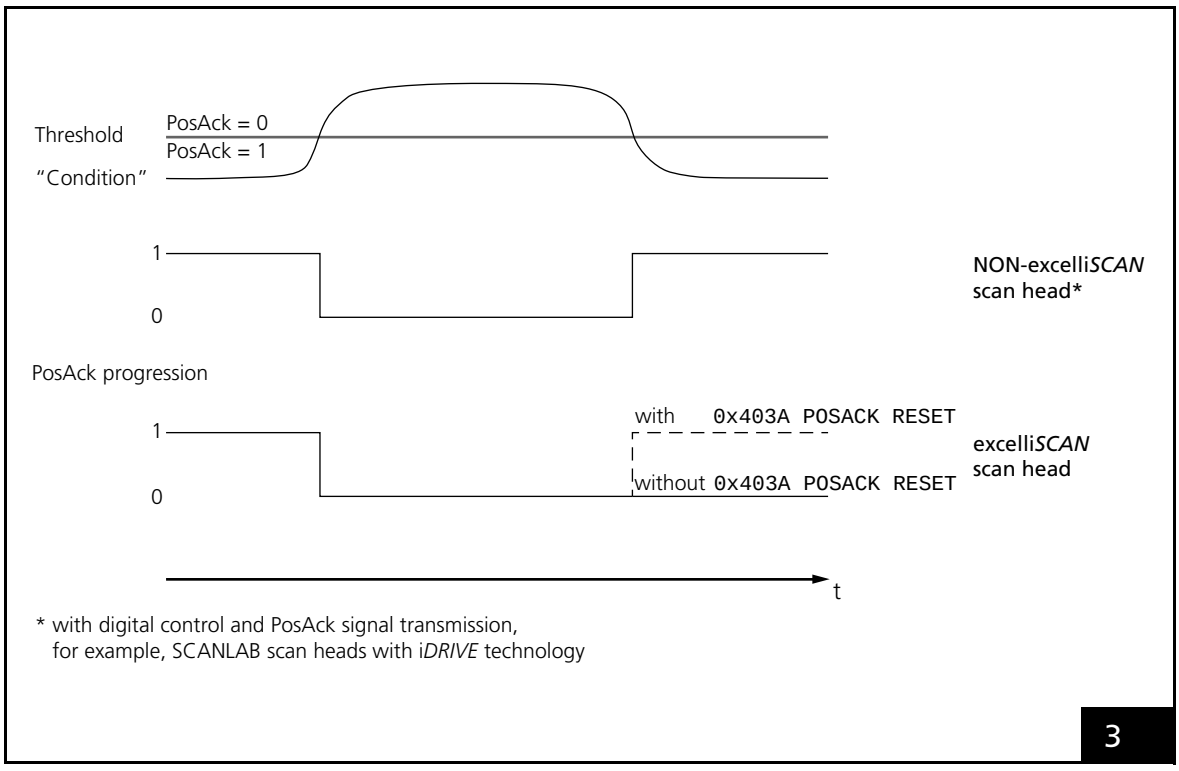
Scan Head without *SCAN*ahead Technology (Continued)	Scan Head with *SCAN*ahead Technology, for example, excelli*SCAN* (Continued)
• You can set scanner delays and laser delays: These do *not* need adjusting for changes in speed. There are rules of thumb for sizing the delays, but truly optimal delays always require empirical determination for each case of application. Furthermore, delay values need determining for each tuning.	• You can set scanner delays and laser delays: They must be set *differently in accordance with speed*. Thanks to constant acceleration and precalculated set positions, the relationship between speed and optimum delays is likewise mathematically known in advance. The RTC®6 board can automatically determine the optimal speed-dependent scanner delays and laser delays. It then also sets them dynamically in real time during list execution. For this, the user activates RTC®6 autodelay functionality by **set_scanahead_params** and **activate_scanahead_autodelays**`(1)` as described in chapter 3.1 "Introduction", page 16 with figure 5 and figure 6. If the user does *not* want to use RTC®6 autodelay functionality (**set_scanahead_params** and **activate_scanahead_autodelays**`(0)`), then: – Speed-dependent delays "must" be determined and – appropriately set before each change of speed

(a) Therefore, at very high speeds or for very small radii. See also empirical formula in "Notes on Arc Commands", page 22.

2.2.3 Returned Data Signal Differences: PosAck Status Signal Behavior after Limit Exceedance

Both scan heads with i*DRIVE*® technology and excelli*SCAN* scan heads return the PosAck status signal at 10 µs intervals. But the following differences apply after exceedance of the threshold value, see figure 3:

- For non-excelli*SCAN* scan heads, the PosAck bit gets automatically reset to 1 as soon as the position error is smaller than the limit value.
- For excelli*SCAN* scan heads, the PosAck bit is not automatically reset to 1 as soon as the position error is smaller than the limit value. That is, the PosAck bit remains at 0.
  This lets you determine after the event whether a limit exceedance has been occurred.
  To reset the PosAck bit to 1, you must execute
  `control_command (Data = 0x403A).`



PosAck signal for non-excelli*SCAN* and excelli*SCAN* scan heads

2.2.4 Returned Data Signal Differences: New Status Value HEAD_BUSY

A new addition to the RTC®6 board in comparison with the RTC®5 is the returnable status value HEAD_BUSY (bit #23).

This status value only gets returned by excelli*SCAN* scan heads. It can be queried by the **get_status** control command, see also the RTC®5 manual chapter 6.4.3 "List Execution Status", page 77.

A prerequisite for this is that **set_scanahead_params** with valid parameters has been called. That is, the RTC®6 board is configured to operate excelli*SCAN* scan heads.

The HEAD_BUSY status value is returned when:

- A list is being processed
- A list has been finished, but scan head output is still in progress
  (temporal offset = `PreviewTime`)
- The marking's final still-in-progress `LaserOffDelay` has not yet expired, that is, so long as the laser is still on

This allows waiting until the actual end of marking before an equipment controller closes a shutter or power is cut to the scan head, etc.

In contrast, if the laser was manually switched on (**laser_signal_on**) after the list ended but before expiration of `PreviewTime`, then its LASEROFF is not be waited for.

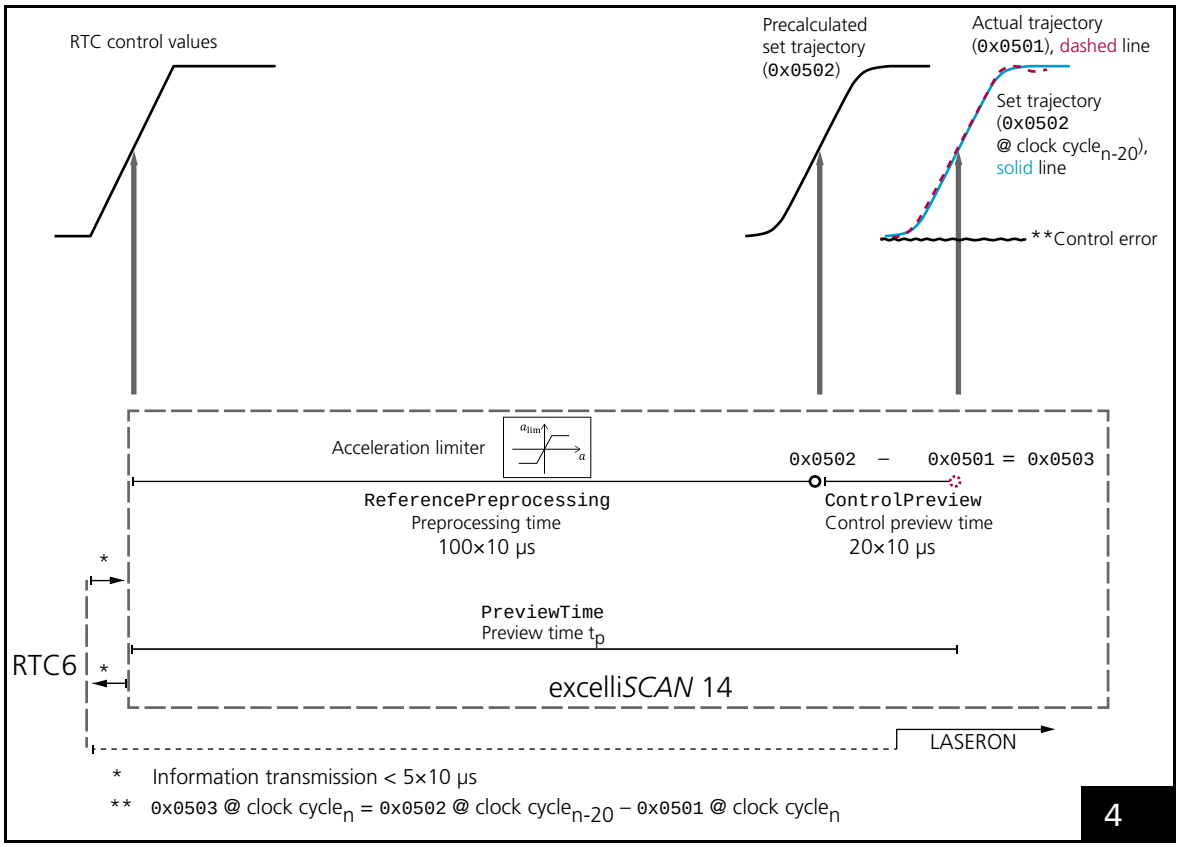
2.2.5 Returned Data Signal Differences: Change of Meaning for 0x0502 Signal

The value `0x0502` returned by excelli*SCAN* scan heads has a different meaning than for non-excelli*SCAN* scan heads, see chapter 2.3 "excelliSCAN-Specific Changes of Returned Data with the control_command Command" and chapter 2.2.7 "Timing Diagram of excelliSCAN 14 Scan Head with SCANahead Technology", page 12.

2.2.6 New Returned Data Signal: New Type for SetControlDefinitionMode

In response to the **SetControlDefinitionMode** command, excelli*SCAN* scan heads return the value 4 (*SCAN*ahead trajectory control), see page 15 in chapter 2.3 "excelliSCAN-Specific Changes of Returned Data with the control_command Command".

2.2.7 Timing Diagram of excelli*SCAN* 14 Scan Head with *SCAN*ahead Technology



Timing diagram of excelli*SCAN* 14 scan head

Note: In the timing diagram (figure 4), specified absolute time values are for the excelli*SCAN* 14.

For excelli*SCAN* scan heads, the signal `0x0502` is the already acceleration-limited precalculated set position (angular position). This value can be queried with `control_command (Data = 0x0502)`.

This `0x0502` signal is delayed with respect to RTC control values by `ReferencePreprocessing` (excelli*SCAN* 14: 100×10 µs clock cycles, that is, 1.0 ms).

Each precalculated set position value `0x0502` is reached after the control preview time `ControlPreview` (excelli*SCAN* 14: exactly 20×10 µs clock cycles, that is, 0.2 ms).

The trajectory of actual positions `0x0501` therefore lags the trajectory of precalculated set positions `0x0502` by 0.2 ms and lags the RTC control values by 1.2 ms (`PreviewTime`).

For excelli*SCAN*, the signal `0x0503` (position error) is the difference between the precalculated set position `0x0502` (at 1.0 ms after the RTC control value) and the actual position `0x0501` at 1.2 ms after the RTC control value (and 0.2 ms after the precalculated set value). This value can be queried by `control_command (Data = 0x0503)`.

The galvanometer scanner position error – even during acceleration phases – is very small (that is, practically a straight line). You can therefore select a very small PosAck threshold value (in comparison to conventional scan heads) without the limit value being exceeded and the PosAck bit getting set.

As the example timing diagram shows, the position error is calculated as follows: `0x0503` @ clock cycle$_n$ = `0x0502` @ clock cycle$_{n-20}$ – `0x0501` @ clock cycle$_n$. Users who intend to calculate the control error themselves from log files have to observe this relationship.

With logging software, the signal `0x0503` is therefore depicted as x-axis-parallel line (that is, nearly horizontal line at 0).

2.3 excelli*SCAN*-Specific Changes of Returned Data with the control_command Command

Ctrl Command	**control_command**			
Function	Sends a control command to an excelli*SCAN* scan head (firmware versions >5050).			
Call	`control_command( Head, Axis, Data )`			
Parameters	`Head`	See RTC5 manual.		
	`Axis`	See RTC5 manual.		
	`Data`	See RTC5 manual.		
		Code$_H$	***Command and parameter values (Code$_L$)***	
		05_H	***SetMode***: See RTC5 manual. As described there: – Each *Code$_L$* parameter value corresponds to a specific data type. – The standard setting is *Code$_L$* = 00_H (XY2-100 status word). excelli*SCAN* scan systems have an SL2-100 interface. They therefore return a signed 20-bit status value.	
			Code$_L$	***Data type returned by scan system***
			01_H	For NON-excelli*SCAN* and excelli*SCAN* scan heads, the following applies: Actual position (angular position of galvanometer scanners). Bits [-2^{19} … $2^{19}-1$]. See chapter 2.2.7 "Timing Diagram of excelliSCAN 14 Scan Head with SCANahead Technology", page 12 and figure 4.
			02_H	For NON-excelli*SCAN* scan heads, the following applies: Set position (galvanometer scanners' angular position). Bits [-2^{19} … $2^{19}-1$]. For excelli*SCAN* scan heads, the following applies: Precalculated set position (angular position of galvanometer scanners). Bits [-2^{19} … $2^{19}-1$]. See chapter 2.2.7 "Timing Diagram of excelliSCAN 14 Scan Head with SCANahead Technology", page 12 and figure 4.
			03_H	For NON-excelli*SCAN* and excelli*SCAN* scan heads, the following applies: Position error (= set position – actual position). Bits [-2^{19} … $2^{19}-1$]. See chapter 2.2.7 "Timing Diagram of excelliSCAN 14 Scan Head with SCANahead Technology", page 12 and figure 4.

Ctrl Command	control_command			
		$0A_H$	***UpdatePermanentMemory***. This command has no effect on excelli*SCAN* scan heads.	
		$0E_H$	***SetControlDefinitionMode*** As described in the RTC®5 manual. Additionally, the value 4 is the tuning type for excelli*SCAN*s.	
			Code$_L$	***Value returned by scan head***
			00_H… 03_H	Bits #4…7 Tuning type: (…) = 4: *SCAN*ahead servo control.
		11_H	***SelectControlDefinition***. This command has no effect on excelli*SCAN* scan heads. Only one tuning is available for excelli*SCAN* scan heads.	
		12_H	***SetPositionScale***. This command has no effect on excelli*SCAN* scan heads. With excelli*SCAN* scan heads, the effective calibration cannot be changed.	
		40_H	***ResetPOSACK***. For excelli*SCAN* scan heads, this command lets you reset the PosAcknowledge warning. The only allowed parameter value for this command is *Code$_L$* = $3A_H$. See also RTC®5 manual chapter 8.1.6 "Configuring the PosAcknowledge Threshold Value", page 178, page 279, page 282 as well as in this document chapter 2.2.3 "Returned Data Signal Differences: PosAck Status Signal Behavior after Limit Exceedance", page 10. Note: Because the ***UpdatePermanentMemory*** command has no effect on excelli*SCAN* scan heads, the PosAcknowledge threshold value cannot be saved for subsequent new starts or resets.	
Notes	*General notes:* • –			
Version info	Status: Version DLL 601.			
References	**get_value, get_values, get_head_status, get_value, set_trigger, set_trigger4, get_waveform, get_last_error**			

3 RTC®6 *SCAN*ahead Functions

3.1 Introduction

As previously explained, the excelli*SCAN* scan head internally calculates a set trajectory with limited, constant accelerations, whereby the acceleration time depends on speed changes. Resultingly, the scanner and laser delays needed for taking acceleration time into account depend on speed.

The RTC®6 board can automatically calculate the required delays. For this and other reasons, SCANLAB recommends the RTC®6 for operating excelli*SCAN* scan heads and other *SCAN*ahead-based scan heads.

Two commands "activate" RTC®6 *SCAN*ahead functionality for suitable control of excelli*SCAN* scan heads:

(1) **set_scanahead_params** `(Mode =1)`

The RTC®6 board queries relevant information from the scan head, processes it if needed (coordinate transformations etc.) and holds it for further usage. Additionally, the board's laser control section gets prepared for taking `PreviewTime` into account. This affects LASER_DIGITAL_OUT1/2 etc., see "Notes on Controlling Peripherals" on page 24.

(2) **activate_scanahead_autodelays** `(1)`
The RTC®6 is instructed:

- To use the results of (1).
- To ignore delays set in source code by **set_scanner_delays** and **set_laser_delays**.
- To automatically calculate (for high contour fidelity) optimal speed-dependent scanner delays and laser delays[1].
- To also set these dynamically in real time during list execution.
- To likewise automatically calculate all required parameters when sky writing is used (for example, run-in and run-out motions), see also "Notes on Sky Writing", Seite 23.

Thus, users can effortlessly and immediately create markings with high contour fidelity [2] (without the need to determine and optimize delays).

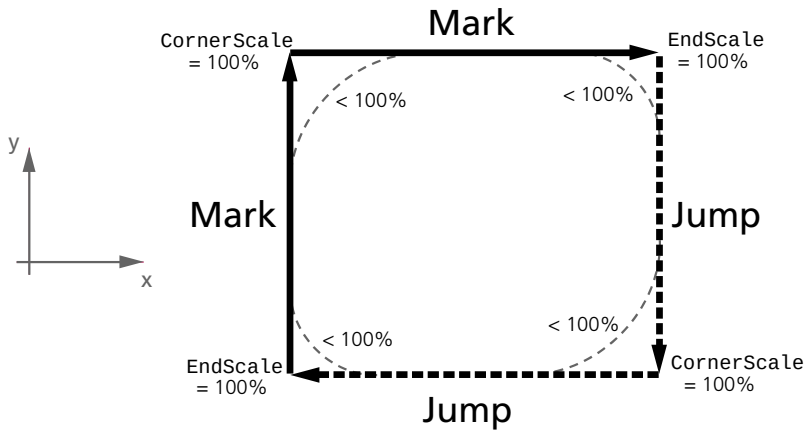
Of course, users are also free to tailor marking results using additional suitable RTC®6 commands:

- *Scanner delays* are influenced by the `CornerScale` and `EndScale` parameters of the two commands **set_scanahead_line_params** and **set_scanahead_line_params_list**.
  The effects of these parameters are shown in figure 5.
  You can choose between optimal contour fidelity and optimal process speed at the cost of contour fidelity (*process optimization*).
- *Laser delays* are influenced by the `AccScale` parameter of the two commands **set_scanahead_line_params** and **set_scanahead_line_params_list**.
  The effect of this parameter is shown in figure 6.
  Here, you can influence energy deposition at line ends by partially or wholly show/hide acceleration phases.
- Also available for *fine-tuning* laser delays are the RTC®6 commands **set_scanahead_laser_delay_shift** and **set_scanahead_laser_delay_shift_list**. They let you set a temporal offset for the laser switching time points, for example, to compensate signal propagation times.

(1) For all commands that require delays, that is, jump, mark, arc commands.

(2) Provided that the specified marking speed makes sense. However, the jump speed is automatically reduced to suitable values.

Scanner delays



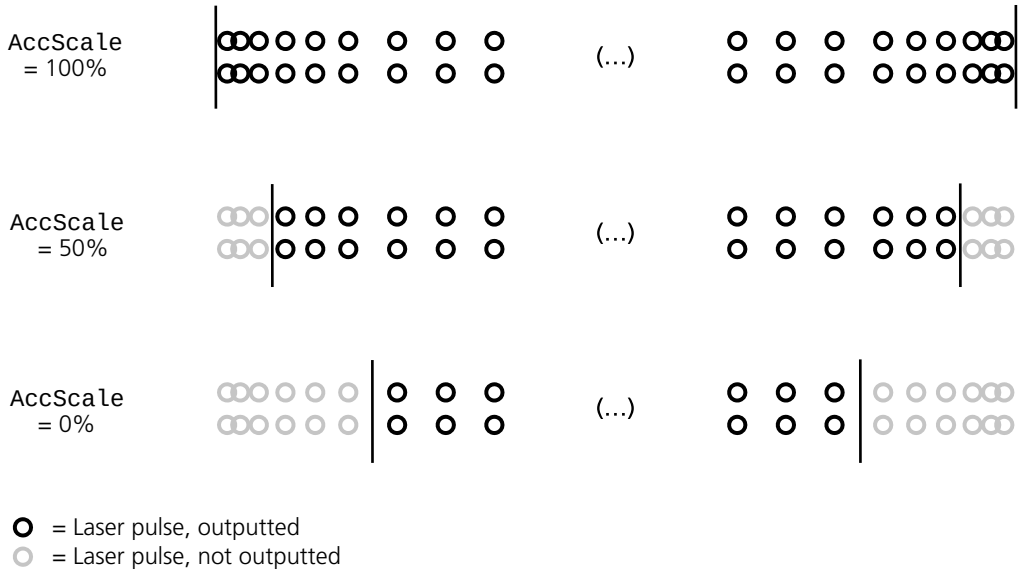
`CornerScale` controls the rounding of corners within polylines.
100% = sharp corners.

`EndScale` controls the marking accuracy at mark/jump and jump/mark transitions.
100% = straight line ends.

5

Command **set_scanahead_line_params**(_list), parameter `CornerScale` and parameter `EndScale` (influence scanner delays)

Laser delays



6

Command **set_scanahead_line_params**(_list), parameter `AccScale` (influences laser delays)

3.2 Quick Migration Guide

This chapter gives advanced users the following practical information for quickly transitioning to an excelli*SCAN* scan head with an RTC®6 board:

- Example source code for a user application, see chapter 3.2.1 “Example Source Code“, page 19
- Tips on adapting existing RTC user applications for the excelli*SCAN*, see chapter 3.2.2 “Modifying Existing User Programs for excelliSCAN Scan Heads – Basic Steps“, page 20
- A summary of what is no longer applicable compared to the RTC®5 or scan heads without *SCAN*ahead technology, see chapter 3.2.3 “Modifying Existing User Programs for excelliSCAN Scan Heads – Taking Restrictions into Account“, page 21

Further details are provided starting with chapter 3.3 “Modifying Existing User Programs – Further Steps, Background Info on SCANahead and excelliSCAN Functions“, page 22.

3.2.1 Example Source Code

The following example code snippets illustrate basic elements of an RTC®6 user application to initialize the RTC6DLL.DLL and RTC®6 board (that is, the example is not compilable).

Because this example uses implicit linking, the files RTC6impl.h and RTC6DLL.LIB are required. Additionally, the program needs to be able to call RTC6DLL.DLL. If the operating system can not find RTC6DLL.DLL during program start, then it responds with an error message and terminates the program.

```
// SAMPLE SOURCE CODE SNIPPETS: TO MARK A SQUARE.
//
// RTC®6 header file for implicit linking to RTC6DLL.DLL.
// also link to Visual C++ import library RTC6DLL.LIB, to compile an executable
#include "RTC6impl.h"

// scanahead line params
CornerScale = 100;    // polyline: 100%: max precision, 0% min execution time
EndScale = 100;       // mark, jump: 100%: max precision, 0% min execution time
AccScale = 100;       // 100%: laser is on during acceleration and deceleration
                      //   0%: laser is off during acceleration and deceleration

// enable/disable variable polygon delay or sky writing
VariablePolygonDelay = 0;
SkyWriting = 0;
int main()
{

// initialize
init_rtc6_dll();
if( 0 != load_program_file( NULL ) ) { return -1; }           // load RTC firmware
if( 0 != load_correction_file( NULL, 1, 2 ) ) { return -2; }
set_mark_speed_ctrl( 25000 );
set_jump_speed_ctrl( 50000 );

// laser control
set_laser_control( 0 );                                      // high active laser signals
set_laser_mode( 1 );                                         // mode 1 supports YAG
set_firstpulse_killer( 640 );
set_laserpulses_ctrl( 320, 128 );

// scanahead: calculate delays automatically. excelliSCAN scan head must be connected and powered.
set_scanahead_params( 1, 1, 1, 0, 0, 0 );                     // place anywhere after load_correction_file
activate_scanahead_autodelays( 1 );
set_scanahead_line_params( CornerScale, EndScale, AccScale );
set_scanahead_laser_shifts( 50, 50 );                         // fine tuning LaserON and LaserOFF

// sky writing and variable polygon delay
set_delay_mode( VariablePolygonDelay, 0, 2^30, 0, 0 );
set_sky_writing_para( SkyWriting, 0, 0, 0 );
set_sky_writing_mode( 3 );
set_sky_writing_limit( 0.5 );

// create and execute list
set_start_list( 1 );
jump_abs( 50000,  -50000 );
mark_abs( 50000, 50000 );
mark_abs( -50000, 50000 );
mark_abs( -50000, -50000 );
mark_abs( 50000, -50000 );
jump_abs( 0, 0 );
set_end_of_list();
execute_list( 1 );
}

// END OF SAMPLE SOURCE CODE SNIPPETS.
```

3.2.2 Modifying Existing User Programs for excelli*SCAN* Scan Heads – Basic Steps

To operate an excelli*SCAN* with the RTC®6, SCANLAB recommends modifying existing user programs as described below. For more details, see the following chapters, as well as the RTC®6 manual. It is assumed that the user's existing program is a complete RTC®6 or RTC®5 application for operating a SCANLAB scan head without *SCAN*ahead technology, for example, an intelli*SCAN*®.

(1) After the **load_correction_file** command at the beginning of your user program, be sure to insert the command **set_scanahead_params** (for parameters, see the command description).

This activates the RTC®6 board's *SCAN*ahead functionality (incl. laser control!). `Mode = 1` initiates direct querying of the scan head for scan-head-specific information on `PreviewTime`, `Vmax` and `Amax`. Therefore, an excelli*SCAN* scan head must be connected to the RTC®6 and already and switched-on during user-program runtime.

(2) Insert **activate_scanahead_autodelays**`(1)`.
This switches on automatic calculation of scanner delays and laser delays.
All **set_scanner_delays** and **set_laser_delays** commands in subsequent code lines have no effect, see also notes for **activate_scanahead_autodelays**.

(3) If the original program is an RTC®5 user program, then also make the following changes (for more detailed and comprehensive information, see the RTC®6 manual):
- change `init_rtc5_dll` to `init_rtc6_dll`
- change `rtc5_count_cards` to `rtc6_count_cards`
- change `free_rtc5_dll` to `free_rtc6_dll`
- do *NOT*(1) change `set_rtc5_mode` and `set_rtc4_mode` to `set_rtc6_mode`.

In practice, these modifications should already suffice for successful execution of your user program.

Any source code sections for variable polygon delays and sky writing should thereby also be runnable.

Check whether your code is affected by the restrictions mentioned next, in chapter 3.2.3 "Modifying Existing User Programs for excelliSCAN Scan Heads – Taking Restrictions into Account", page 21.

(1) This section is intended as a quick guide: for detailed information (for example, 3D applications etc.), see the RTC®6 manual.

3.2.3 Modifying Existing User Programs for excelli*SCAN* Scan Heads – Taking Restrictions into Account

While continuing to modify your user program for the excelli*SCAN* scan head, keep in mind the following:

Non-Usable Functionalities of the RTC®6 Board and RTC®6 DLL

- *After activation of SCAN*ahead *functionality by* **set_scanahead_params**, the following RTC®6 functionalities are *not* usable (status: DLL ≤601):
  - Processing-on-the-fly
  - Softstart mode
  - Galvanometer-speed-dependent automatic laser control(1)
  - Encoder-speed-dependent automatic laser control(2)
  - **laser_on_pulses_list**
  - **load_varpolydelay**

Non-Usable i*DRIVE*® Functionalities

- Unlike with intelli*SCAN*® scan heads, the following i*DRIVE*® functions are not available for excelli*SCAN* scan heads:
  - Switching between tunings (**SelectControlDefinition**)
    `control_command (Data = 0x11xx).`
  - Changing the effective calibration (**SetPositionScale**)
    `control_command (Data = 0x12xx).`

See chapter 2.3 "excelliSCAN-Specific Changes of Returned Data with the control_command Command", page 14.

(1) Described in RTC®5-Manual chapter 7.4.9 "Automatic Laser Control", page 156.

(2) Described in RTC®5-Manual section "Encoder-Speed-Dependent Laser Control" on page 165.

3.3 Modifying Existing User Programs – Further Steps, Background Info on *SCAN*ahead and excelli*SCAN* Functions

While continuing to modify your user program for the excelli*SCAN* scan head, keep in mind the following:

Notes on Using Commands for Scanner Delays and Laser Delays

When using commands with the excelli*SCAN* to synchronize scan motions and laser control signals, observe the following:

- All relevant *SCAN*ahead parameters (article-number-specific) are stored in the excelli*SCAN* scan head during manufacturing. Although **set_scanahead_params** `(Mode = 2,…)` allows other values, we recommend only using the values queried from the excelli*SCAN* scan head. Mode 2 is primarily intended for software development without an attached excelli*SCAN* scan head.
- For excelli*SCAN* scan heads, the actual position ot the scan motion lags the control values by `PreviewTime`, see figure 1(1). For synchronous laser control, the laser control signals also need to take `PreviewTime` into account.
  This can be done manually by appropriately modifying parameters for the **set_scanner_delays** and **set_laser_delays** commands. But here, you must take the excelli*SCAN*'s speed-dependent acceleration time into account.
  The preferable alternative is to simply use the **activate_scanahead_autodelays** command for automatic calculation of scanner delays and laser delays. Then, the following applies:
  – The existing command parameters for **set_scanner_delays** and **set_laser_delays** are invalidated by `activate_scanahead_autodelays( Mode = 1 )` and not modified. They become immediately effective again if automatic delay calculation gets deactivated by `activate_scanahead_autodelays( Mode = 0 )`.
  – The command `set_scanahead_params( Mode = 1 or 2, HeadNo, TableNo, PreviewTime, Vmax, Amax)` defines parameters for automatic calculation of scanner delays and laser delays together with `activate_scanahead_autodelays( Mode = 1)`.
- If needed, throughput of laser-marked workpieces can be increased, if lower quality is acceptable ("process optimization"): see **set_scanahead_line_params** `( CornerScale, EndScale, AccScale )`.
  See also figure 5 and figure 6.
- For *fine-tuning* laser delays (for example, to compensate signal propagation times to the laser or to take laser system switching behavior into account), see **set_scanahead_laser_delay_shift** `( LaserOnDelay, LaserOffDelay )`.

Notes on Arc Commands

- excelli*SCAN* scan heads have a nominal tracking error of 0(2). Tracking-related artifacts (for example, necking during small, quick circular motions) do not occur so long as the maximum

(1) This temporal offset differs quantitatively and qualitatively from tracking error in scan heads without *SCAN*ahead technology.

(2) The output delayed by `PreviewTime` is not tracking error in the ususal sense.

acceleration of the head is not exceeded (for example, when the source code selects an excessive speed in relation to the circular diameter).

- To estimate maximum acceleration within the image field, you can query the maximum acceleration scaled to an image-field-based bits/10 ms² by `get_scanahead_params (HeadNo = 256)`.Note: Do not use this queried and rescaled value for `set_scanahead_params(Mode = 2)`.
- To select a suitable speed for a particular circular radius, you can apply the following rule of thumb:
  $v^2/r < 0.7$ `Amax`(1)
  (valid for excelli*SCAN* 14 with any objective. Insert either control bits or programming bits into the formula). See also page 22.

Notes on Polygon Delays

- On the excelli*SCAN*, variable polygon delays work similarly to scan heads without *SCAN*ahead technology.
  To activate variable polygon scanner delays, you still must call the command **set_delay_mode** with `VarPoly` > 0.
  If automatic delay calculation (**activate_scanahead_autodelays**) is switched on, its parameters `EdgeLevel`, `MinJumpDelay` and `JumpLengthLimit` have no effect. But they become immediately effective if automatic delay calculation gets switched off.
  In contrast, the `DirectMove3D` parameter is always effective.
- For excelli*SCAN*, the **load_varpolydelay** command has no effect. Currently (status: DLL 601) user-defined variable polygon delay tables are not available.

Notes on Sky Writing

- For excelli*SCAN*, sky writing functionality has the same effect as on scan heads without *SCAN*ahead technology. All three modes are selectable.
  To activate sky writing functionality, you must still call the command **set_sky_writing** or **set_sky_writing_para** with `Timelag` > 1/4 µs.
  If automatic delay calculation (**activate_scanahead_autodelays**) is switched on, then `Timelag` (except for the actual activation of sky writing) and the parameters `Nprev` and `Npost` have no effect. But they immediately become effective, if automatic delay calculation gets switched off.
  In contrast, the `LaserOnShift` parameter is always effective.

(1) `Amax` is queried by `get_scanahead_params(HeadNo = 256)`. Observe the units specified in the command description for **get_scanahead_params**.

Notes on Functions for Automatic Laser Control

- Functions for automatic laser control are currently not usable (status: DLL 601).

Notes on Controlling Peripherals

When using commands to control peripherals (LASER connector pins ANALOG OUT 1 and ANALOG OUT 2, EXTENSION 1 and EXTENSION 2 headers), observe the following:

If the *SCAN*ahead functionality of the RTC®6 board has been activated by **set_scanahead_params** (`PreviewTime` > 0), then:

(1) the control commands **write_io_port** and **write_8bit_port** are immediately executed (that is, not simultaneously with the galvanometer scanners; “asynchronous”).

(2) execution of the list commands **write_io_port_list** and **write_write_8bit_port_list** lags by the `PreviewTime` (that is, simultaneous with the galvanometer scanners; “axissynchronous”).

If the *SCAN*ahead functionality of the RTC®6 board has not been activated (`PreviewTime` = 0), then asynchronous and axissynchronous execution (1) and (2) are identical; therefore control commands and list commands are handled identically and immediately executed.

With the new laser control mode of the RTC®6 (`set_dsp_mode( 3 )`), LASERON delays during a series of short vectors can extend across one or several subsequent vectors. The RTC®6 in DSP mode 3 eliminates synthetic scanner delays that prevent this. Output is then synchronous with each vector’s corresponding LASERON delay. The list commands **write_da_1_list**, **write_da_2_list**, **write_da_x_list** and the control commands **write_da_1**, **write_da_2**, **write_da_x** are synchronized with the corresponding laser control signals (that is, executed simultaneously with the laser control signals; “lasersynchronous”). With `set_dsp_mode( 2 )` (RTC®5-compatible laser control), axissynchronous and lasersynchronous execution are identical.

Notes on Unique Characteristics of Returned Data Signals

Some of the excelli*SCAN*’s returned data signals differ from those of other i*DRIVE*® scan heads, for example, intelli*SCAN*®.

- **PosAck Status Value with excelli*SCAN***
  If the threshold is exceeded, then the PosAck bit gets set. It remains set so long as it does not get reset by `control_command (Data = 0x403A)`.
  See chapter 2.2.3 “Returned Data Signal Differences: PosAck Status Signal Behavior after Limit Exceedance”, page 10.
- **Position error value with excelli*SCAN***
  `control_command (Data = 0x0503)`
  This is the difference between the precalculated set position (at 100 clock cycles after the RTC control value) and the actual position (at 120 clock cycles after the control value or 20 clock cycles after the set value) (values apply to excelli*SCAN* 14).
  See chapter 2.2.7 “Timing Diagram of excelliSCAN 14 Scan Head with SCANahead Technology”, page 12

Notes on PosAck Status Signal Behavior

- See chapter 2.2.3 “Returned Data Signal Differences: PosAck Status Signal Behavior after Limit Exceedance”, page 10.

3.4 RTC®6 Commands - Exclusively for *SCAN*ahead Systems

This chapter describes RTC®6 commands intended exclusively for scan heads with *SCAN*ahead technology, for example, the excelli*SCAN*.

This command type is identifiable by the name element "**_scanahead_**".

All are also available as multi-board commands (status: DLL 601). Multi-board command names begin with the prefix **n_**.

(1) con control command

(2) us undelayed short list command

Note

- Universally usable RTC®6 commands are described in the RTC®6 manual.

Ctrl Command	**activate_scanahead_autodelays**
Function	Switches on or off the automatic (dynamic) calculation of the scanner delays and laser delays. Returns the current mode after the command has been executed.
Call	`CurrentMode = activate_scanahead_autodelays ( Mode )`
Parameters	`Mode` Signed 32-bit value. `-1`: Only returns the current mode. The current mode is not changed. `0`: Switches off automatic calculation. `1`: Switches on automatic calculation (`Mode = 1`) if **set_scanahead_params** has been called previously. Otherwise, `Mode` remains 0 after the command has been executed.
Result	Mode as unsigned 32-bit value.
Comments	• If `Mode = 0`, then normal scanner delays and laser delays are used (see **set_scanner_delays** and **set_laser_delays**). If `Mode = 1`, then scanner delays and laser delays are calculated and set automatically. Values for `Mode = 0` do not get overwritten. After switch-off, they remain available for usage (as with non-excelli*SCAN* scan heads). • **activate_scanahead_autodelays** also affects Sky Writing: the sky writing command parameters `Timelag` (except for the actual activation of sky writing), `Nprev` and `Npost` have no effect. However, they are going to have immediate effect once the automatic delay calculation is switched off. In contrast, the parameter `LaserOnShift` is always effective.
RTC®5→ RTC®6	New command.
Version info	Available as of version DLL 600, OUT 600.
References	**activate_scanahead_autodelays_list**, **set_scanner_delays**, **set_laser_delays**

Undelayed Short List Command	**activate_scanahead_autodelays_list**
Function	Identical with **activate_scanahead_autodelays**, but an undelayed short list command.
Call	`CurrentMode = activate_scanahead_autodelays_list ( Mode )`
Parameters	Parameter as unsigned 32-bit value. For values, see **activate_scanahead_autodelays**.
Comments	• The **activate_scanahead_autodelays_list** command takes effect upon the next to-be-calculated delay. • `Mode = -1` is not available with **activate_scanahead_autodelays_list**.
RTC®5→ RTC®6	New command.
Version info	Available as of version DLL 600, OUT 600.
References	**activate_scanahead_autodelays**

Ctrl Command	**get_scanahead_params**	
Function	Queries certain parameters from the specified scan head. The attached scan head must be an excelli*SCAN*.	
Restriction	The command can be executed only if no list is currently active. Otherwise, the **get_last_error** return code gets set to `RTC6_BUSY` and error code 5 is returned.	
Call	`Error = get_scanahead_params ( Headno, &PreviewTime, &Vmax, &Amax )`	
Parameters	`HeadNo`	Scan head connector number as unsigned 32-bit value. Allowed values: = 1: scan head connector 1. = 2: scan head connector 2 (activation required). = 256: RTC®6 board (no scan head connector). The parameters `Vmax` and `Amax` are returned converted to image field coordinates.
Returned parameter values	`PreviewTime`	Pre-calculation time for galvanometer scanner control in [10 µs].
	`Vmax`	For `HeadNo` 1, 2: Velocity limit as control bits (galvanometer scanner rotation angle) [bits/10 µs]. For `HeadNo` 256: Velocity limit as programming bits (image field coordinates) [bits/ms].
	`Amax`	For `HeadNo` 1, 2: Acceleration limit as control bits (galvanometer scanner rotation angle) [bits/10 µs²]. For `HeadNo` 256: Acceleration limit as programming bits (image field coordinates) [bits/ms²].
	All values are pointers to unsigned 32-bit values.	

Ctrl Command	get_scanahead_params
Result	Error code as unsigned 32-bit value. Value — Description 3 — No excelli*SCAN* scan head is attached or its excelli*SCAN* tuning is not active (**get_last_error** return code: `RTC6_PARAM_ERROR`). 5 — A list is currently active (**get_last_error** return code: `RTC6_BUSY`). 6 — `HeadNo` = `0` or `HeadNo` >2 has been specified as parameter (**get_last_error** return code: `RTC6_PARAM_ERROR`). 8 — The RTC®6 Board is not responding. Probably a program has not been loaded yet (**get_last_error** return code: `RTC6_TIMEOUT`). 11 — A PCI error occurred (**get_last_error** return code: `RTC6_SEND_ERROR`).
Comments	• The returned parameter values are also used by the **set_scanahead_params** command. • `Vmax` is the maximum angular velocity of the galvanometers. `Vmax` *is not* the marking speed in the image plane. The returned parameter value unit type is therefore [bits/10 µs] and *not* [bits/ms]. Distortion by the correction file is not yet taken into account in the returned parameter value (see **set_scanahead_params**). • `HeadNo` = `256` only returns meaningful values if **set_scanahead_params** was previously called.
RTC®5→ RTC®6	New command.
Version info	Available as of version DLL 600, OUT 600.
References	**set_scanahead_params**, **get_last_error**

Ctrl Command	set_scanahead_laser_delay_shift
Function	Shifts the LaserOn signals and LaserOff signals temporally forward or back.
Call	`set_scanahead_laser_delay_shift ( LaserOnDelay, LaserOffDelay )`
Parameters	`LaserOnDelay` Time shift of LaserOn signals in [0.5 µs]. `LaserOffDelay` Time shift of LaserOff signals in [0.5 µs]. Shifts as signed 32-bit values.
Comments	• The **set_scanahead_laser_delay_shift** command lets you *fine-tune* the laser delay, for example, to compensate laser signal propagation times or to take the laser system switching behavior into account. • If **set_scanahead_params** has not been already called and automatic calculation has not been activated (**activate_scanahead_autodelays** `Mode = 0`), then the values only get stored, but not applied. But they are applied as soon as required conditions are fulfilled.
RTC®5→ RTC®6	New command.
Version info	Available as of version DLL 600, OUT 600.
References	**set_scanahead_laser_delay_shift_list**, **set_scanahead_params**, **activate_scanahead_autodelays**

Undelayed Short List Command	set_scanahead_laser_delay_shift_list
Function	Identical with **set_scanahead_laser_delay_shift**, but an undelayed short list command.
Call	`set_scanahead_laser_delay_shift_list ( LaserOnDelay, LaserOffDelay )`
Parameters	See **set_scanahead_laser_delay_shift**.
Comment	• The **set_scanahead_laser_delay_shift_list** command takes effect upon the next to-be-calculated delay.
RTC®5→ RTC®6	New command.
Version info	Available as of version DLL 600, OUT 600.
References	**set_scanahead_laser_delay_shift**

Ctrl Command	set_scanahead_line_params
Function	Influences the quality of marking results at runtime. Smaller percent values increase throughput at the expense of quality.
Call	`set_scanahead_line_params ( CornerScale, EndScale, AccScale )`
Parameters	`CornerScale` Corner sharpness in percent. 100% = sharp corners. `EndScale` Marking accuracy at mark/jump and jump/mark transitions. 100% = straight line ends. `AccScale` Determines the portion of the acceleration *time* (not: distance traversed) in which the laser is active, in percent. 100% = entire acceleration time. Parameters as unsigned 32-bit values.
Comments	• Only values from 0% to 100% are useful. Higher values do not improve quality (that is, corners cannot be even sharper than sharp), but instead only extend marking times. • If **set_scanahead_params** has not been already called, then the values only get stored, but not applied.
RTC®5→ RTC®6	New command.
Version info	Available as of version DLL 600, OUT 600.
References	**set_scanahead_line_params_list**

Undelayed Short List Command	set_scanahead_line_params_list
Function	Identical with **set_scanahead_line_params**, but an undelayed short list command.
Call	`set_scanahead_line_params_list ( CornerScale, EndScale, AccScale )`
Parameters	See **set_scanahead_line_params**.
Comment	• The **set_scanahead_line_params_list** command takes effect upon the next to-be-calculated delay.
RTC®5→ RTC®6	New command.
Version info	Available as of version DLL 600, OUT 600.
References	**set_scanahead_line_params**

Ctrl Command	set_scanahead_params
Function	Activates the RTC®6-*SCAN*ahead functionality suitable to control an excelli*SCAN* scan head. The laser Control and digital inputs/outputs for the peripherals are prepared for an excelli*SCAN* scan head, that is, delayed by `PreviewTime`. Thereby scanner delays and laser delays are not automatically calculated. For this purpose serves **activate_scanahead_autodelays**.
Restriction	The command can only be executed if no list is currently active. Otherwise, the **get_last_error** return code gets set to `RTC6_BUSY` and error code 5 is returned.
Call	`Error = set_scanahead_params ( Mode, HeadNo, TableNo, PreviewTime, Vmax, Amax )`
Parameters	`Mode` = 0: Deactivates the RTC®6-*SCAN*ahead functionality. The additional parameters `TableNo, PreviewTime, Vmax, Amax` are not taken into account. Mode, for example, for operating intelli*SCAN*® scan heads. = 1: Queries parameters of the scan head with the specified `HeadNo` and applies them. The parameters `PreviewTime, Vmax, Amax` are not taken into account. Default mode for operating attached excelli*SCAN*scan heads. = 2: Apply the parameters of this command. Mode, for example, for software development. An excelli*SCAN* scan head does not need to be connected. `HeadNo` Number of the scan head connector as unsigned 32-bit value. Allowed values: = 1: scan head connector 1. = 2: scan head connector 2 (activation required). `TableNo` Allowed range 1–4. A correction file should have been loaded by **load_correction_file**(…, `TableNo`, …). This is used for converting the parameters `Amax` and `Vmax` from control bits (galvanometer scanner rotation angle) into programing bits (image field coordinates). See **get_scanahead_params**. `PreviewTime` Pre-calculation time for galvanometer scanner control in [10 μs]. This parameter is ignored in `Mode = 0` and `Mode = 1`. `Vmax` Velocity limit as control bits (galvanometer scanner rotation angle) [bits/10 μs]. This parameter is ignored in `Mode = 0` and `Mode = 1`. `Amax` Acceleration limit as control bits (galvanometer scanner rotation angle) [bits/10 μs^2]. This parameter is ignored in `Mode = 0` and `Mode = 1`. All parameters as unsigned 32-bit values.

Ctrl Command	set_scanahead_params
Result	Error code as unsigned 32-bit value. Value — Description 3 — No excelli*SCAN* scan head attached or excelli*SCAN* tuning is not active (**get_last_error** return code: `RTC6_PARAM_ERROR`). 5 — A list is currently active (**get_last_error** return code: `RTC6_BUSY`). 6 — `HeadNo` = `0` and `HeadNo` >2 (**get_last_error** return code: `RTC6_PARAM_ERROR`). 7 — The determined scaling factor exceeds 16 or is below 1/16 (possibly no correction file was loaded). The scaling factor is set to 1. 8 — The RTC®6 board is not responding. Probably a program has not been loaded yet (**get_last_error** return code: `RTC6_TIMEOUT`). 11 — A PCI error occurred (**get_last_error** return code: `RTC6_SEND_ERROR`)
Comments	• The `Vmax` and `Amax` parameters are used for automatic calculation of scanner delays and laser delays, see **activate_scanahead_autodelays**. • The values for `Vmax` and `Amax` are specified as control bits (galvanometer scanner rotation angles). Information is read-out from the correction file and used for an internal conversion into programing bits (sample values). Therefore, the correction file needs to be specified in the `TableNo` parameter. At conversion time, the correction file must already be loaded on the board, but does not yet need to be assigned. The system cannot reliably detect correction files that have not been loaded. The `Vmax` and `Amax` values which have been converted into programing bits can be queried by **get_scanahead_params**(`HeadNo` = `256`). • For error return information, see **get_scanahead_params**. • `Mode` = `1` results in aborted command execution (`Error` = `3`) if no excelli*SCAN* is attached and active. All *SCAN*ahead functionality is deactivated. Parameters specified for `Mode` = `2` are not applied! • The **set_scanahead_params** command waits until the last session has finished traversing, that is, the HEAD_BUSY signal (bit #23) from **get_status** is no longer set. In any case, it waits no longer than 255 × [10 µs].
RTC®5→ RTC®6	New command.
Version info	Available as of version DLL 600, OUT 600.
References	**get_scanahead_params**, **activate_scanahead_autodelays**, **get_last_error**, **get_status**, **load_correction_file**

Notes