

# INTEGRATION MANUAL



---

## TABLE OF CONTENTS

---

<b>Table of Contents</b> .....	<b>2</b>
<b>1. Introduction</b> .....	<b>3</b>
<b>2. Communication Interface</b> .....	<b>4</b>
2.1 Hardware .....	4
2.2 Software .....	4
<b>3. Software Interface</b> .....	<b>5</b>
3.1 General information .....	5
3.2 Command List .....	7
3.3 Command details .....	8
3.4 Error Codes .....	12
3.5 Extended programming information .....	13
<b>4. Hardware Interface</b> .....	<b>14</b>
4.1 Device Installation .....	14
4.2 Device Dimensions and Images .....	15
TiltStation .....	15
<b>5. Changelog</b> .....	<b>16</b>
<b>6. Notes</b> .....	<b>17</b>
<b>7. Support</b> .....	<b>20</b>

## 1. INTRODUCTION

QINSTRUMENTS devices are optimized to be integrated seamlessly into automation platforms. The simple and over all our devices standardized command set allows you to easily set and control process parameters and read out sensor values. Through the integrated microelectronics no other external components or control devices are necessary. All units are designed for continuous 24 hour hands-free operation when utilizing sound scientific methods

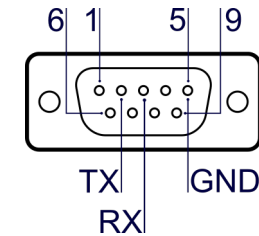
Providing long term stable hard- and software interfaces and supporting industry standard like SiLA<sup>®</sup>, paves the way to a superior level of lab automation. Due to the outstanding integration support we are happy to call the leading providers for lab automation our partners.

## 2. COMMUNICATION INTERFACE

### 2.1 HARDWARE

The RS232 interface is available through the 9-pin D-subminiature connector. Pins 2 (TX), 3 (RX) and 5 (GND) on the connector are used (see picture below).

Baud rate	9600
Parity	None
Data bits	8
Stop bits	1
Hardware or software handshake (XON/XOFF)	Not supported



The power connection is realized via a barrel connector (ID 2.5 mm x OD 5.5 mm). For all devices the power supply must fulfill the following requirements. Only use the delivered power supply (Mean Well GST120A24) to fully stay within the device certification.

Input	100 - 240 V AC   50 - 60 Hz
Output	24 V DC   I <sub>max</sub> : 5.0 A   P <sub>max</sub> : 120 Watt
Approvals	CE/UL/CS, 85-264 V AC, 47-63 Hz IEC/EN60320-1 C14



### 2.2 SOFTWARE

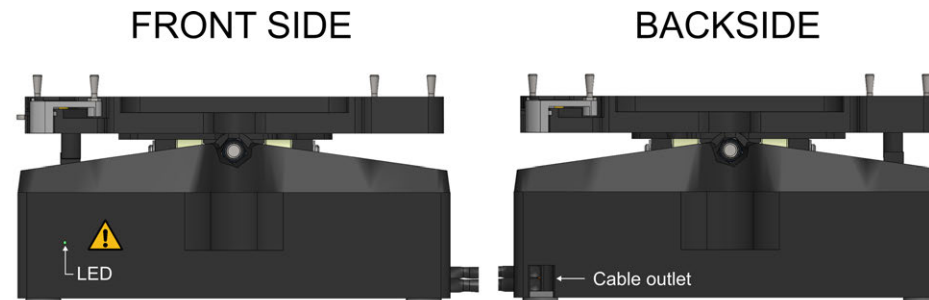
The device functions and parameters are controlled via the commands which are provided by QINSTRUMENTS and are explained in detail in chapter: "[Software Interface](#)" on the next page. Keep in mind that these commands are just the building blocks for a device driver. It is required to deal with device errors, logging, communication errors and lags, command timings and so on to realize a stable, solid and productive devices control. Additionally specific requirements that are related to the platform in which the unit is integrated might need to be addressed.

## 3. SOFTWARE INTERFACE

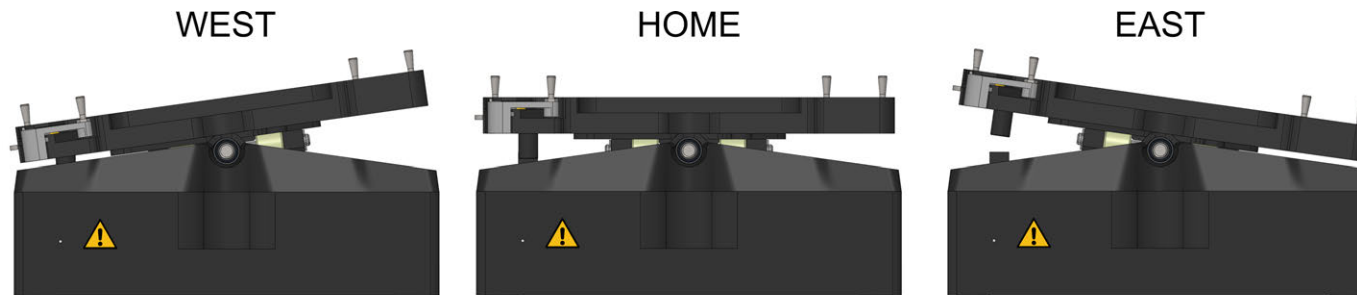
### 3.1 GENERAL INFORMATION

#### Device Orientation

The TiltStation has 3 defined positions . They are called: WEST | HOME | EAST . These positions are defined in regard to the device orientation described in the pictures below. Front side is where the LED is situated, backside is where the outlet is situated.



Looking at the FRONT SIDE the positions are defined as described in the picture below. Moving to the WEST or EAST position (`tiltGoWest`, `tiltGoEast`) means that the tablar is lowered at this side, see picture below.



If required it is possible to switch the definition of WEST and EAST with the command `setTiltPositionSwitch`. Switching the positions is not persistent and needs to be done after each restart or reset of the device.

---

**Information**


---

All commands need to be terminated with: <CR>\*

All return values are terminated with: <CR><LF>\*\*

Sending an unknown command returns: u->'unknown command'<CR><LF>

After each switch on of the device or call of the `reset` command it is required to call `tiltInit` before any tilt motions can be executed. All commands that initiate a tilt motion before calling `tiltInit` will return an error; e.

Initialization sets the target speed to 2 oscil/min, afterwards the last setting for speed (`stiopmt<value>`) is kept

Initialization sets the target acceleration to 1 sec , afterwards the last setting for acceleration (`stia<value>`) is kept

If the device has an error **or** the command which is send interferes with the current status of the device, the device returns: e<CR><LF>\*\*\*

If an error occurs (smart LED turns red), the device must be reset, to clear its internal failure memory and continue working normally.

When the power supply is connected and active, the system is automatically started (boot process) and all hardware components will be checked. This process takes about 30 seconds.

After setting a parameter value or changing the device status with a `set` command, recheck the parameter value | status with the related `get` command. This ensures a correct operation of hardware and software.

The waiting time for status requests with `get` command is minimum 100 milliseconds.

If the device is not used for longer than 15 min, please switch to the ECO mode with the command `setEcoMode<CR>` to save energy and decrease abrasion. Wake up the device with `leaveEcoMode<CR>`.

The following commands are changing a device parameter permanently. This means the value is saved to EEPROM and kept even if the device is powered off.  
`disableCLED` | `enableCLED`

---

\* <CR> is the term for the control character "carriage return" in ASCII code (decimal 13, hexadecimal 0x0D)

\*\* <LF> is the term for the control character "line feed" in ASCII code (decimal 10, hexadecimal 0x0A)

\*\*\* Getting e as return value does not necessarily mean that the device has an error. Device errors are written to the error list, which can be accessed with the `gel` command. An example for status interference is when the command `tiltOn` is send while the device is already oscillating.

## 3.2 COMMAND LIST

	Long Form<CR>	Short Form<CR>	Return value<CR><LF>		Long Form<CR>	Short Form<CR>	Return value<CR><LF>
INITIALIZATION	disableCLED		ok	TILT	setTiltAcceleration<value>	stia<value>	ok
	enableCLED		ok		setTiltOscillationsPerMinuteTarget<value>	stiopmt<value>	ok
	getDescription		<unit type>		setTiltPosition	stip<value>	ok
	getErrorList	gel	{'e <sub>1</sub> '; ... ;'e <sub>n</sub> '}		setTiltPositionSwitch<value>	stips<value>	ok
	getSerial		<device serial number>		tiltEmergencyOff	tieoff	ok
	getVersion		<fw version>		tiltGoEast	tige	ok
	info		<boot text>		tiltGoHome	tigh	ok
	resetDevice	reset	ok		tiltGoWest	tigw	ok
	tiltInit	tii	ok		tiltOff	tioff	ok
	version	v	<unit type + fw version>		tiltOn	tion	ok
ECO	leaveEcoMode	lem	ok	tiltOnWithOscillations<value>	tionwo<value>	ok	
	setEcoMode	sem	ok	tiltOnWithRuntime<value>	tionwr<value>	ok	
TILT	getTiltAcceleration	gtia	<value>	ELM	getElmState	ges	<value>
	getTiltAccelerationMax	gtiamax	<value>		getElmStateAsString	gesas	<value>
	getTiltAccelerationMin	gtiamin	<value>		setElmLockPos	selp	ok
	getTiltMaxOpm	gtimax	<value>		setElmUnlockPos	seup	ok
	getTiltMinOpm	gtimin	<value>				
	getTiltOscillationsPerMinuteActual	gtiopma	<value>				
	getTiltOscillationsPerMinuteTarget	gtiopmt	<value>				
	getTiltPosition	gtip	<value>				
	getTiltPositionSwitch	gtips	<value>				
	getTiltRemainingOscillations	gtiro	<value>				
	getTiltRemainingTime	gtirt	<value>				
	getTiltState	gtis	<value>				
	getTiltStateAsString	gtisas	<value>				

### 3.3 COMMAND DETAILS

	Command<CR>   Short Form<CR> Command description	<Example Execution><CR>	<Example Return Value><CR><LF> Additional information
INITIALIZATION	<b>disableCLEd</b> Permanent deactivation of the LED indication lights. The instrument will reset after this command.	disableCLEd >>	ok
	<b>enableCLEd</b> Permanent activation of the LED indication lights. The instrument will reset after this command.	enableCLEd >>	ok
	<b>getDescription</b> Returns model type.	getDescription >>	Q.Q.MTP-TILTSTATION
	<b>getErrorList   gel</b> Returns a semicolon-separated list with errors and warnings which can occur during processing.	getErrorList >>	{303} <i>Note: Please see also section "Error control" in this document.</i>
	<b>getSerial</b> Returns the device serial number.	getSerial >>	0000012345
	<b>getVersion</b> Returns the firmware version number.	getVersion >>	1.8.00
	<b>info</b> Returns the boot screen text.	info >>	<boot screen text>
	<b>resetDevice   reset</b> Restarts the controller.	resetDevice >>	ok <i>Note: This takes about 30 seconds for BS units and 5 for the Q1, CP models</i>
	<b>tiltInit</b> Initialize the TiltStation. Executing the command is obligatory after each switch on or reset of the device	tii >>	ok <i>Note: This is obligatory before device can start tilt motion</i>
	<b>version   v</b> Returns model type and fw version number.	version >>	Q.MTP-TILTSTATION v1.8.00
	ECO	<b>leaveEcoMode   lem</b> Leaves the economical mode and switches into the normal operating state.	leaveEcoMode >>
<b>setEcoMode   sem</b> Switches the device into economical mode. It will reduce electricity consumption by deactivation the solenoid that locks the home position and deactivation of the ELM function.		setEcoMode >>	ok <i>Note: Home position is not locked!</i> <i>Note: All commands other than leaveEcoMode will return: e</i>
TILT	<b>getTiltAcceleration   gtia</b> Returns the set acceleration/deceleration value in seconds.	getTiltAcceleration >>	5
	<b>getTiltAccelerationMax   gtiamax</b> Returns the maximum accelration/deceleration time in seconds.	getTiltAccelerationMax >>	30
	<b>getTiltAccelerationMin   gtiamin</b> Returns the minimum accelration/deceleration time in seconds.	getTiltAccelerationMin >>	1



Command<CR>   Short Form<CR> Command description	<Example Execution><CR>	<Example Return Value><CR><LF> Additional information
<b>getTiltMaxOpm   gtimax</b> Returns the maximum oscillations per minute value.	getTiltMaxOpm >>	100
<b>getTiltMinOpm   gtimin</b> Returns the minimum oscillations per minute value.	getTiltMinOpm >>	2
<b>getTiltOscillationsPerMinuteActual   gtiopma</b> Returns the current oscillations per minute (speed).	getTiltOscillationsPerMinuteActual >>	5
<b>getTiltOscillationsPerMinuteTarget   gtiopmt</b> Returns the target oscillations per minute (speed).	getTiltOscillationsPerMinuteTarget >>	10
<b>getTiltPosition   gtip</b> Returns current tilt position.	getTiltPositionSwitch >>	2 2 Home Position 3 East Position 4 West Position 9 Unknown Position
<b>getTiltPositionSwitch   gtips</b> Returns if the WEST and EAST positions are switched.	getTiltPositionSwitch >>	0 Positions are not switched 1 Positions are switched
<b>getTiltRemainingOscillations   gtiro</b> Returns the remaining oscillations if the device was started with the command: tiltOnWithOscillations.	getTiltRemainingOscillations >>	7
<b>getTiltRemainingTime   gtirt</b> Returns the remaining time in seconds if the device was started with the command: tiltOnWithRuntime.	getTiltRemainingTime >>	6
<b>getTiltState   gtis</b> Returns device state.	getTiltState >>	3 0 Running 1 Detected a stop command 2 Stopping is imminent 3 Stopped and is locked at position 4 Manual mode for external control 5 Accelerates 6 Decelerates 7 Decelerates to stop 8 Accelerates to stop 90 ECO mode 99 Boot process running   tiltInit needs to be executed 100 Error detected. Use getErrorList command.

TILT

Command<CR>   Short Form<CR> Command description	<Example Execution><CR>	<Example Return Value><CR><LF> Additional information
<b>getTiltStateAsString   gtisas</b> Returns device state as string.	getTiltStateAsString >>	STOP RUN Running BREAK Detected a stop command STOPPING Stopping is imminent STOP Stopped and is locked at position MANUAL Manual mode for external control RAMP+ Accelerates RAMP- Decelerates RAMP+STOP Accelerates to stop RAMP-STOP Decelerates to stop ECOMODE ECO mode INIT Boot process running   tiltInit needs to be executed ERROR Error detected. Use getErrorList command. UNKNOWN Undefined status. Should not happen.
<b>setTiltAcceleration&lt;value&gt;   stia&lt;value&gt;</b> Sets the acceleration/deceleration value in seconds.	setTiltAcceleration4 >>	ok
<b>setTiltOscillationsPerMinuteTarget&lt;value&gt;   stiopmt&lt;value&gt;</b> Sets the target oscillations per minute (speed).	setTiltOscillationsPerMinuteTarget10 >>	ok
<b>setTiltPosition&lt;value&gt;   stip&lt;value&gt;</b> Device moves to given position and locks in place.	setTiltPosition2 >>	ok 2 Home position 3 East position 4 West Position
<b>setTiltPositionSwitch&lt;value&gt;   stips&lt;value&gt;</b> WEST and EAST position is switched (value=1) or reset (value=0). Switching the positions is not persistent. After each reset or switch off/on of the device the default value (value=0) is set.	setTiltPositionSwitch1 >>	ok
<b>tiltEmergencyOff   tieoff</b> Device stops immediately at an undefined position ignoring the defined deceleration time.	tiltEmergencyOff >>	ok
<b>tiltGoEast   tige</b> Device moves to East position and locks in place.	tiltGoEast >>	ok
<b>tiltGoHome   tigh</b> Device moves to Home position and locks in place.	tiltGoHome >>	ok
<b>tiltGoWest   tigw</b> Device moves to West position and locks in place.	tiltGoWest >>	ok

	Command<CR>   Short Form<CR> Command description	<Example Execution><CR>	<Example Return Value><CR><LF> Additional information
TILT	<b>tiltOff   tioff</b> Stops oscillating within the defined deceleration time, go to the home position and locks in place.	tiltOff >>	ok
	<b>tiltOn   tion</b> Starts oscillating with defined speed (oscil/min) with defined acceleration time.	tiltOn >>	ok
	<b>tiltOnWithOscillations&lt;value&gt;   tionwo&lt;value&gt;</b> Starts oscillating with defined speed with defined acceleration time for given amount of oscillations.	tiltOnWithOscillations10 >>	ok <i>Note: &lt;value&gt; range: 0 – 999999 (1 to 6-digits, without comma)</i>
	<b>tiltOnWithRuntime&lt;value&gt;   tionwr&lt;value&gt;</b> Starts oscillating with defined speed with defined acceleration time for given amount of seconds.	tiltOnWithRuntime15 >>	ok <i>Note: &lt;value&gt; range: 0 – 999999 (1 to 6-digits, without comma)</i>
	<b>getElmState   ges</b> Returns the ELM status.	getElmState >>	1 0 ELM is moving 1 ELM is locked 3 ELM is unlocked 9 ELM error occurred
ELM	<b>getElmStateAsString   gesas</b> Returns the ELM status as a string.	getElmStateAsString >>	1 ELMUndefined ELM is moving ELMLocked ELM is locked ELMUnlocked ELM is unlocked ELMError ELM error occurred
	<b>setElmLockPos   selp</b> Closes the ELM.	setElmLockPos >>	ok <i>Note: The runtime is less than 3 seconds.</i> <i>Note: This position is a current-free static state.</i>
	<b>setElmUnlockPos   seup</b> Opens the ELM	setElmUnlockPos >>	ok <i>Note: The runtime is less than 3 seconds.</i> <i>Note: This position is a current-free static state.</i>

### 3.4 ERROR CODES

In case of failure the device smart LED turns red and the device error code is accessible via the command: `getErrorList`



If an error occurred and not stated otherwise the device must be reset to clear its internal failure memory and continue working normally. If the error cannot be solved by restarting the device, please get in contact with our ["Support" on page 20](#)

	Code	Description
TILT	400 <sup>1</sup>	Error caused by over heating of driver.
	401 <sup>1</sup>	Error caused by under voltage of driver.
	402	Error caused by detected deviation in tilt position. A reason can be too high a load or an additional resistance in the movement of the tilt station.
	403	Error during dynamic correction of tilt position.
	404	Error caused by stall detection of motor.
	415	Error while executing TiltInit command. Error caused by error preceding the TiltInit command.
	416	Error while executing TiltInit command. Error caused by erroneous execution of ELM unlock command.
	417	Error while executing TiltInit command. Error caused by detected deviation in tilt position.
ELM	418	Error while executing TiltInit command. Error caused by stall detection of motor.
	300 <sup>1</sup>	General error.
	301 <sup>1</sup>	IC-Driver error.
	303	Verification error by the unlock position.
	304	Error caused by unsuccessful reach the lock position (timeout).
	305	Error caused by unsuccessful reach the unlock position (timeout).
	306	Error caused by unsuccessful reach the lock position (over current).
307	Error caused by unsuccessful reach the unlock position (over current).	

<sup>1</sup> - Please get in contact with the QINSTRUMENTS service team.

### 3.5 EXTENDED PROGRAMMING INFORMATION

This chapter presents additional information on programming while using the device commands.

#### Return value 'e'

**Information** It is important to know that there are several situations in which the return value of a command is `e`, although there is no error in the device error list. This is due to point that the return value `e` is also returned if the command does not fit to the current status of the device or other criteria.

Examples are:

- Sending `tion`, `tionwo` or `tionwr` command if speed is not set
- Sending `tion`, `tionwo` or `tionwr` command while device is already running.
- Sending `tion`, `tionwo` or `tionwr` command while the ELM is open
- Most commands while in ECO mode
- `seup` while ELM is already open and the other way around
- ...

**Tip** Use the get commands (`getTiltState`, `getElmState`) to evaluate the current device status before sending commands to prevent erroneous usage of commands.

#### `setElmLockPos` | `setElmUnlockPos` | `setEcoMode`

**Information** After sending these commands `setElmLockPos`, `setElmUnlockPos`, `setEcoMode` the device does not send an immediately response. The `ok` value is written to the serial input buffer after the ELM reached the (un)lock position, ECO mode. In the meantime, all commands that are send to the device will be buffered and executed after the device reached the new status.

**Tip** To determine the end of the `setElmLockPos`, `setElmUnlockPos`, `setEcoMode` command, read from the serial input buffer until the `ok` value is received. Do not send any commands in the meantime.

#### `leaveEcoMode` | `resetDevice` | `tiltGoHome` | `tiltGoEast` | `tiltGoWest`

**Information** After sending these commands the device immediately writes `ok` to the serial input buffer although the command execution has not finished. Commands that are send in the meantime will not be executed or return `e`.

**Tip** To determine the end of the `leaveEcoMode`, `resetDevice` command poll for the device status with the `getTiltState` command until it reaches the status 3 (Device stopped and is locked in home position).

## 4. HARDWARE INTERFACE

### 4.1 DEVICE INSTALLATION

All QINSTRUMENTS devices are mounted via the same physical interface and have one corner for the cable outlets, as can be seen in the schematic top view on the device, below.

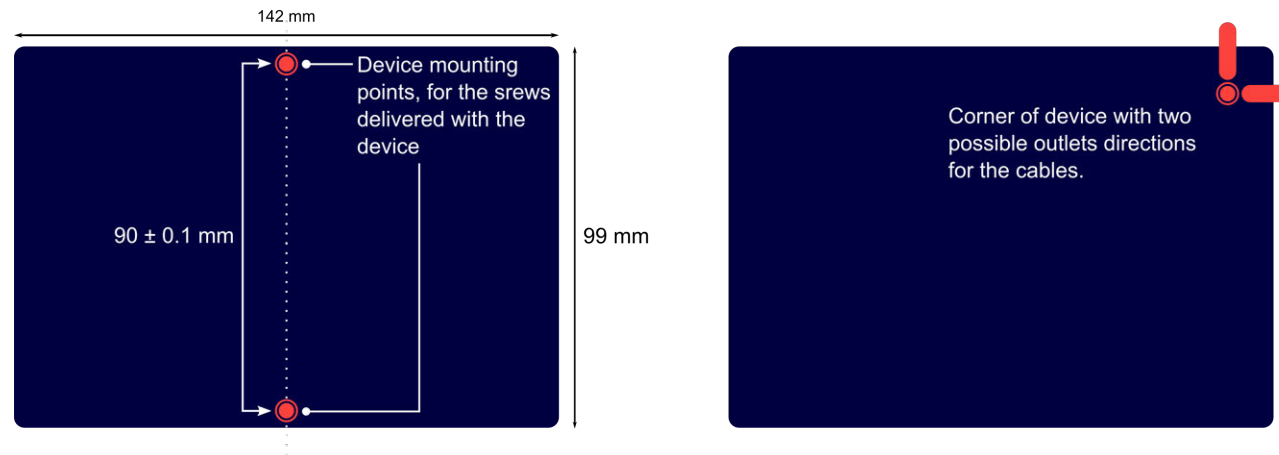


Figure 1 Schematic device top view showing the mounting points and cable outlet options

Besides mounting, other important aspects like the

- cable length (2 m by default)
- sufficient venting for devices with heat/cool feature
- deal with condensation water for devices with cooling feature
- space between devices if multiple devices are integrated (space for ELM movement, air inlet and outlet, other Labware nearby, ...)
- height of adapters
- robot specific requirements (device height, clearance, collision areas, ...)
- ...

need to be addressed when integrating one or multiple devices. A detailed description about the installation procedure, in depth device specifications and further device specific information is available in the device specific Operation manual.

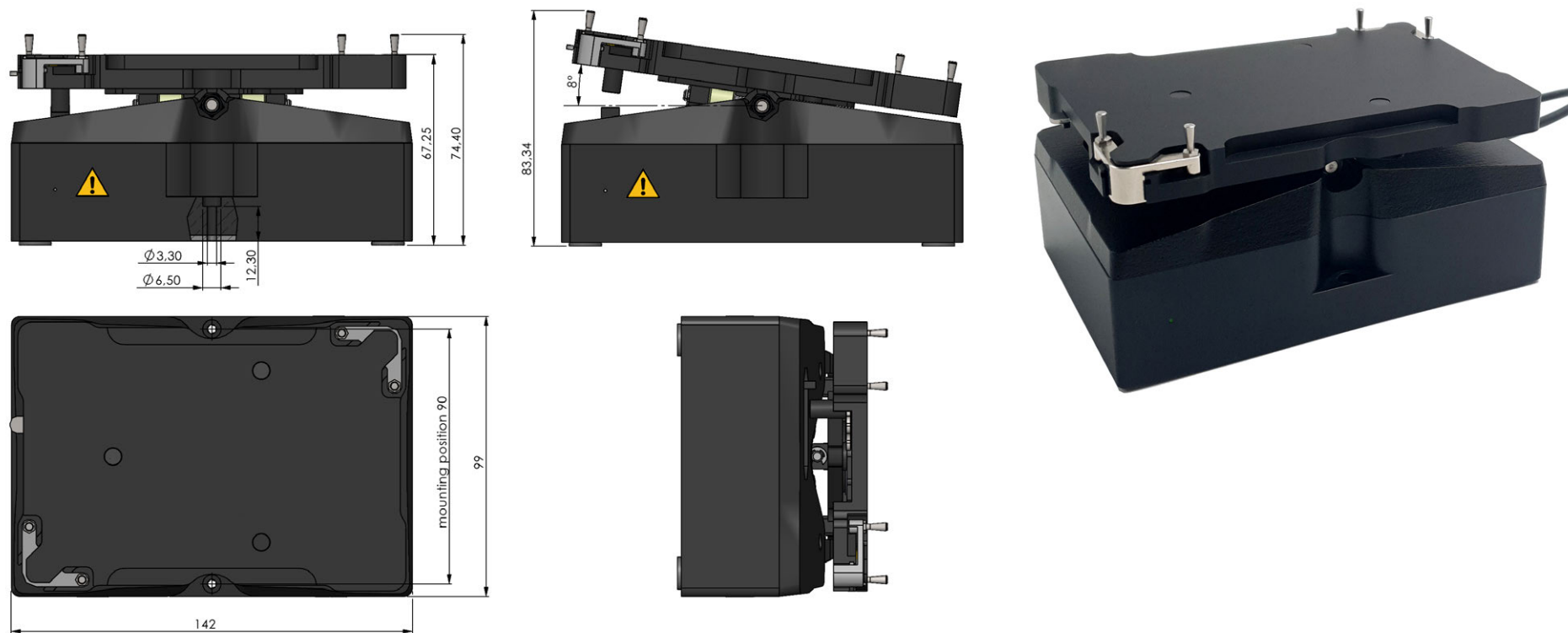


Required device installation information for each device is available online at: <https://www.qinstruments.com/service/downloads/>  
This information is mandatory required to successfully integrate a device.

## 4.2 DEVICE DIMENSIONS AND IMAGES

### TiltStation

2016-0710



## 5. CHANGELOG

---

<b>Version</b>	<b>Information</b>
010.0	Initial creation

---



## 6. NOTES

**NOTES**

## NOTES

## 7. SUPPORT

We provide a range of technical material (e.g., application notes, poster, bulletin, data sheets) that describes our products and key applications in detail. All of our technical documents are available on our web page in the corresponding product pages. Technical Data Sheet, Operation manual, 2D and 3D drawings and Software can be found in the download area of each product.

Please use the following link to our support form (<https://www.qinstruments.com/service/support/>) in case service or support for your product is needed. Please ensure to provide the serial number, as it is an important information for our support team. Direct contact via phone or email is also possible.

**Online**                    QINSTRUMENTS.com  
**E-Mail**                 support@qinstruments.com  
**Phone**                    +49 3641 55430  
**Address**                QINSTRUMENTS GmbH  
                               **A BICO COMPANY**  
                               Loebstedter Strasse 101  
                               07749 Jena  
                               Germany

### DISCLAIMER, LEGAL NOTICES AND TRADEMARKS

All document design, text, graphics, the selection and arrangement thereof and all other materials in this document are copyright by QINSTRUMENTS. QINSTRUMENTS GmbH reserves the right to modify their products for quality improvement and such modifications may not be documented in this manual. This manual and the information herein have been assembled with due diligence. QINSTRUMENTS does not assume liability for misprints or cases of damage resulting from misprints in this manual. If there are any uncertainties, please contact [info@qinstruments.com](mailto:info@qinstruments.com).

QINSTRUMENTS is owner of numerous patents worldwide. Please respect our intellectual property.

**WO2008135565, US8323588, EP2144716:** Sample handling device for and methods of handling a sample | **WO2011113858, US9126162, EP2547431:** Positioning unit for a functional unit

**WO2013113847, US10052598, EP2809436:** Cog-based mechanism for generating an orbital shaking motion

**WO2013113849, US9371889, EP2809435:** Mechanism for generating an orbital motion or a rotation motion by inverting a drive direction of a drive unit

**WO2014207243, US20160368003, EP3013480:** Application-specific sample processing by modules surrounding a rotor mechanism for sample mixing and sample separation

**WO002022128814A1:** Laboratory apparatus comprising a fixing mechanism for fixing a slide | **WO002022128809A2:** Laboratory apparatus comprising a mixing mechanism for mixing a medium of a slide

Please notify us in writing, by email or mail to our designated agent, if you believe that a user has infringed our intellectual property rights. QINSTRUMENTS trademarks are recognised worldwide. Please respect our trademarks as we will vigorously protect their proper usage.

**QINSTRUMENTS®**, **BioShake®**, **ColdPlate®**, **HeatPlate®** (QINSTRUMENTS GmbH)

Trademarks of third parties may appear on this site when referring to those entities or their products or services. All registered names, trademarks, etc. used on this site, even when not specifically marked as such, are not to be considered unprotected by law. Any names and trademarks not specifically marked or listed are property of the respective owner.

Further trademarks used in this website and catalogs: Brand® (BRAND GmbH + Co KG), Corning® (Corning, Inc.), Eppendorf® (Eppendorf AG), Thermomixer® (Eppendorf AG), Eppendorf Tubes® (Eppendorf AG), Eppendorf twin.tec® (Eppendorf AG), Falcon® (Becton, Dickinson And Company), Greiner® (Greiner Labortechnik GmbH), MOXA® (Moxa, Inc.), NUNC® (Nunc NS Corporation), SILA Rapid Integration® (Association Consortium Standardization in Lab Automation), TECAN® (TECAN Group AG), Windows® (Microsoft Corporation).

Technical specifications are subject to change without notice. All rights reserved.