

# INTEGRATION MANUAL

## ColdPlate

### Easy integration of ColdPlate modules in lab automation systems

Using with firmware version  $\geq 2.0$



## Table of contents

1. Device Control.....	3
2. Specifications .....	3
3. Chronological order to install the ColdPlate .....	3
4. Settings for RS-232 control via PC or robot .....	4
5. Important basics for programming.....	4
6. Command overview.....	5
7. Error control.....	8
8. Changelog .....	9
9. Geometric dimensions, drawings & images .....	10
10. Support.....	11

## 1. Device Control

The ColdPlate instruments are completely equipped for hands-free operation with an extensive remote command set for easy software integration.

Through the integrated microelectronics, no other external components and control devices are necessary.

An automatic initialization will start after turn on.

The ColdPlate can be operated via RS232/USB<sup>1</sup> user interface.

A simple command set allows you to easily control all parameters for initialization, starting-/stopping temperature control.

All process parameters are controlled and read out.

<sup>1</sup> USB control via a separate USB/RS232 converter

## 2. Specifications









Operation	Remote controlled
User Interface	RS232 interface
Power supply	100-240 V, 50-60 Hz (input), 24 V DC with ripple and noise < 1% or < 50mV (output)
Housing material	Aluminum anodized
Ambient conditions	+5 up to 45 °C at 80 % humidity, non-condensing

## 3. Chronological order to install the ColdPlate




1. Install the ColdPlate into the robot system. Use 2 screws for the adjustment and fixation.
2. Connect with the RS-232 port from your PC.
3. Connect the 24 VDC power supply. The smart LED should be turn blue.
4. Automatic initialization and execution of a self-test for approx. 5 sec. The smart LED turns green.
5. Now you can send and receive commands.

## 4. Settings for RS-232 control via PC or robot

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

-  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
-  All requests terminated with `<CR>`  
*<CR> is the term for the control character "carriage return" in ASCII code (decimal 13, hexadecimal 0x0D)*
-  All responds terminated with `<CR><LF>`  
*<LF> is the term for the control character "line feed" in ASCII code (decimal 10, hexadecimal 0x0A)*
-  An unknown command has the return value `u -> <text of the unknown command><CR><LF>`
-  All temperature values in degree Celsius (°C).
-  Is no communication cable (RS232 cable) connected, the smart LED turns yellow.
-  Between two sent signs may be a maximum of 5 seconds. After 5 seconds and without receiving a termination character `<CR>`, the internal receiver unit is reset and waits for the first character of the next instruction.
-  If an error has occurred (smart LED turns red), the device must be reset, to clear its internal failure memory and continue working normally.  
Exception: error 33020 (see Chapter 7. Error control)

## 5. Important basics for programming

-  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 5 sec.
-  After setting of parameters with `"set"` commands, please recheck the status with the related `"get"` commands.  
This saves the control of a correct operation of hardware and software.
-  The waiting time for status requests with `"get"` command is minimum 100 msec.

## 6. Command overview

Commands		Return Value	Error Value	Sample xxx = input / xxx = output	Description
Long Form	short Form				
<b>Initialization</b>					
getVersion<CR>		Version number <CR><LF>	e<CR><LF>	getVersion<CR> 2.0.0<CR><LF>	Returns the current firmware version number.
getDescription<CR>		Model information <CR><LF>	e<CR><LF>	getDescription<CR> Q.MTP-ColdPlate<CR><LF>	Returns the current model information.
version<CR>	v<CR>	Model information + Version number <CR><LF>	e<CR><LF>	version<CR> Q.MTP-ColdPlate v2.0.0<CR><LF>	Returns the current model information followed by the version number.
info<CR>		Boot screen text <CR><LF>	e<CR><LF>	info<CR> <boot screen text><CR><LF>	Returns the boot screen text (without device self-test information).
resetDevice<CR>	reset<CR>	ok<CR><LF>	e<CR><LF>	reset<CR> ok<CR><LF>	Restarts the controller. This takes about 5 sec.
getErrorList<CR>	gel<CR>	{e1; .. ;en}<CR><LF>	e<CR><LF>	getErrorList<CR> {10002}<CR><LF>	Returns a semicolon-separated list with errors and warnings that occurred during processing. <i>Notes: Please see also section "Error control" in this document</i>
enableCLEd<CR>		ok<CR><LF>	e<CR><LF>	enableCLEd<CR> ok<CR><LF>	Permanent activation of the LED indication lights. The instrument will reset after this command.
disableCLEd<CR>		ok<CR><LF>	e<CR><LF>	disableCLEd<CR> ok<CR><LF>	Permanent deactivation of the LED indication lights. The instrument will reset after this command.
getCLEd<CR>		<value><CR><LF>	e<CR><LF>	getCLEd<CR> 1<CR><LF>	Returns the status LED state: 1: LED is enabled 2: LED is disabled
flashLed<CR>	fid<CR>	ok<CR><LF>	e<CR><LF>	flashLed<CR> ok<CR><LF>	User device LED flashes five times.
setBuzzer<value><CR>		ok<CR><LF>	e<CR><LF>	setBuzzer500<CR> ok<CR><LF>	Lets the buzzer beep for the given time in milliseconds. <i>Notes: &lt;value&gt; range: 50 – 999</i>
enableBootScreen<CR>		ok<CR><LF>	e<CR><LF>	enableBootScreen<CR> ok<CR><LF>	Permanent activation of the boot screen startup text
disableBootScreen<CR>		ok<CR><LF>	e<CR><LF>	disableBootScreen<CR> ok<CR><LF>	Permanent deactivation of the boot screen startup text.

Commands		Return Value	Error Value	Sample xxx = input / xxx = output	Description
Long Form	short Form				
getBootScreenState<CR>		<value><CR><LF>	e<CR><LF>	getBootScreenState<CR> 0<CR><LF>	Returns the boot screen state: 0: screen is disabled 1: screen is enabled
setLedBrightness<value><CR>		ok<CR><LF>	e<CR><LF>	setLedBrightness50<CR> ok<CR><LF>	Sets the device LED brightness. <i>Notes: &lt;value&gt; range: 0 – 127</i>
setLedPulseModeEnable<CR>		ok<CR><LF>	e<CR><LF>	setLedPulseModeEnable<CR> ok<CR><LF>	Enables the device LED pulse mode. Device LED pulsates during heating or cooling.
setLedPulseModeDisable<CR>		ok<CR><LF>	e<CR><LF>	setLedPulseModeDisable<CR> ok<CR><LF>	Disables the device LED pulse mode.

Temperature control					
tempOn<CR>	ton<CR>	ok<CR><LF>	e<CR><LF>	tempOn<CR> ok<CR><LF>	Activates the temperature control and starts heating/cooling <i>Notes: it is strongly recommended to control if desired temperature is reached (see command getTempActual)</i>
tempOff<CR>	toff<CR>	ok<CR><LF>	e<CR><LF>	tempOff<CR> ok<CR><LF>	Switches off the temperature control and stops heating/cooling.
getTempState<CR>	gts<CR>	<value><CR><LF>	e<CR><LF>	getTempState<CR> 1<CR><LF>	Returns the state of the temperature function. value 0 Temperature control is disabled. value 1 Temperature control is enabled.
getTempStateAsString<CR>	gtsas<CR>	<value><CR><LF>	e<CR><LF>	getTempStateAsString<CR> on<CR><LF>	Return the state of the temperature function as string. value value = off Temperature control is disabled. value value = on Temperature control is enabled.
getTempTarget<CR>	gtt<CR>	<value><CR><LF>	e<CR><LF>	getTempTarget<CR> 80.000000<CR><LF>	Returns the target temperature. <i>Notes: deprecated command: getTargetTemp&lt;CR&gt;</i>
setTempTarget<value><CR>	stt<value><CR>	ok<CR><LF>	e<CR><LF>	setTempTarget370<CR> ok<CR><LF>	Sets the target temperature in 1/10 °C between -10 and 99.9°C. <i>Notes: &lt;value&gt; range: -100 – 999 (3-digit value without comma)</i> <i>Notes: deprecated command: setTargetTemp&lt;value&gt;&lt;CR&gt;</i> <i>Notes: temperature value are automatically limited to the min/max value</i>
getTempActual<CR>	gta<CR>	<value><CR><LF>	e<CR><LF>	getTempActual<CR> 74.200000<CR><LF>	Returns the current temperature. <i>Notes: deprecated command: getActualTemp&lt;CR&gt;, gat&lt;CR&gt;</i>
getTempMin<CR>	gtmin<CR>	<value><CR><LF>	e<CR><LF>	getTempMin<CR> -10.000000<CR><LF>	Returns the minimum of the possible temperature set point. <i>Notes: command is used only for information</i>

getTempMax<CR>	gtmax<CR>	<value><CR><LF>	e<CR><LF>	getTempMax<CR> 99.900000<CR><LF>	Returns the maximum of the possible temperature set point. <i>Notes: command is used only for information</i>
getTempLimiterMin<CR>	gtlmin<CR>	<value><CR><LF>	e<CR><LF>	getTempLimiterMin<CR> 25.000000<CR><LF>	Returns the minimum of the temperature set point. <i>Notes: can be adjust with setTempLimiterMin</i>
setTempLimiterMin<value><CR>	gtlmin<CR>	ok<CR><LF>	e<CR><LF>	setTempLimiterMin250<CR> ok<CR><LF>	Limits the minimum temperature set point. <i>Notes: &lt;value&gt; range: -100 – 999 (3-digit value without comma)</i> <i>Notes: value must not be greater than the maximum limit</i>
getTempLimiterMax<CR>	gtlmin<CR>	<value><CR><LF>	e<CR><LF>	getTempLimiterMax<CR> 60.000000<CR><LF>	Returns the maximum of the temperature set point. <i>Notes: can be adjust with setTempLimiterMax</i>
setTempLimiterMax<value><CR>	gtlmin<CR>	ok<CR><LF>	e<CR><LF>	setTempLimiterMax600<CR> ok<CR><LF>	Limits the maximum temperature set point. <i>Notes: &lt;value&gt; range: -100 – 999 (3-digit value without comma)</i> <i>Notes: value must not be lower than the maximum limit</i>
setTempLoggingOn<CR>		ok<CR><LF>	e<CR><LF>	setTempLoggingOn<CR> ok<CR><LF>	Enables the automatically logging of the actual temperature. <i>Notes: the sample time is 1 second</i> <i>Notes: return string includes device serial number, device time in msec and actual temperature value separated by &lt;/&gt; character</i>
setTempLoggingOff<CR>		ok<CR><LF>	e<CR><LF>	setTempLoggingOff<CR> ok<CR><LF>	Disables the automatically logging of the actual temperature.

## 7. Error control

Error Value	Description
<b>General</b>	
10002 or 10003	Instruction sent with an invalid parameter.
<b>Temperature control</b>	
2xxxx	Internal MCU periphery error.
310xx	EEPROM data verification failed.
320xx	Communication with internal temperature sensors failed.
33010	Device internal temperature too hot. <sup>1</sup>
33020	Emergency shutdown of the temperature fuse has triggered. <sup>1, 2</sup>
33030	Emergency temperature sensor validation has failed.
34010	FAN power supply invalid.
34020	FAN has stalled.
34030	FAN airway clogged.
35010	TEC power supply invalid.
35020	TEC power supply short circuit detected.
35030	TEC power supply open circuit detected.

x = Don't care

### Note:

- 1 - Please let the device cool down, before performing a reset.
- 2 - If an internal over-temperature has occurred, the power of the device must be turned off, to clear the fault.



If an error has occurred (smart LED turns red), 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 contact the service.

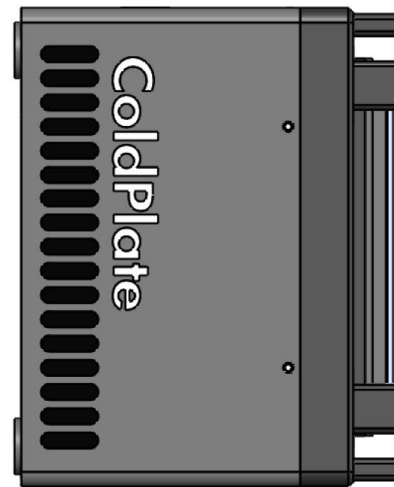
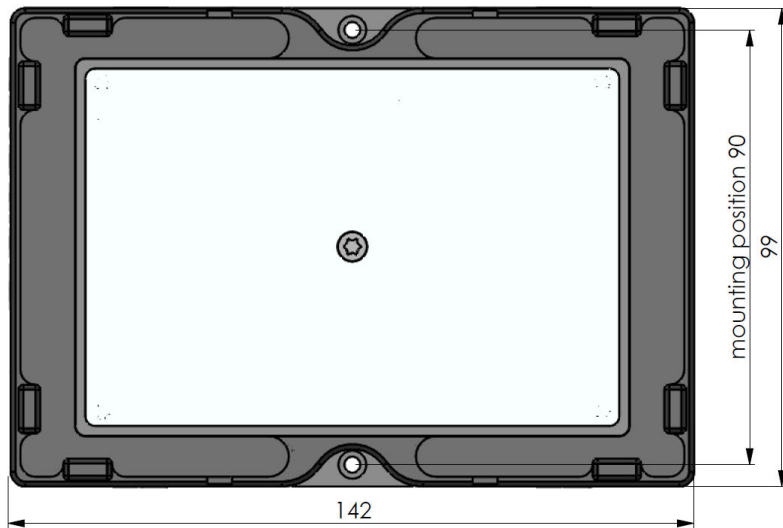
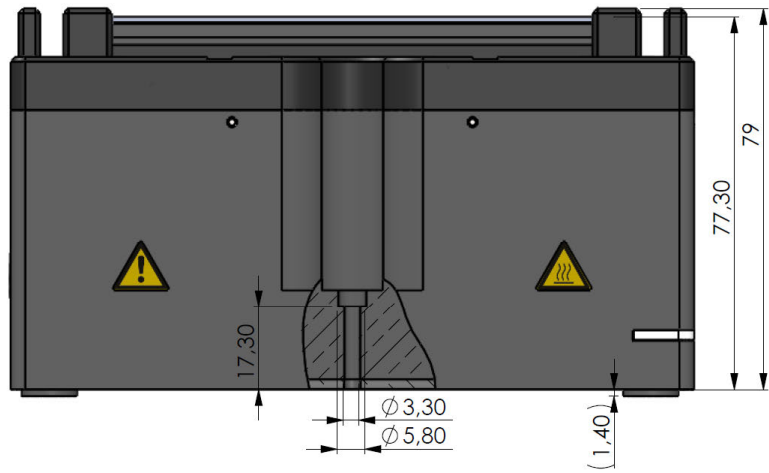


## 8. Changelog

<b>Version 2.0.0 v01</b> (Serial ≥ 12250)	<ul style="list-style-type: none"><li>• Chapter 4. Settings for RS-232 control via PC or robot adjusted</li><li>• Chapter 6. Command overview adjusted</li><li>• Commands added: <i>setBuzzer&lt;CR&gt;</i>, <i>setLedBrightness &lt;CR&gt;</i>, <i>setLedPulseModeEnable &lt;CR&gt;</i>, <i>setLedPulseModeDisable &lt;CR&gt;</i></li><li>• Chapter 7. Error control adjusted</li></ul> Error code 33030 added
<b>Version 1.0.2 v002</b> (Serial ≥ 11614)	Only for internal evaluation processes
<b>Version 1.0.2 v001</b> (Serial ≥ 11614)	<ul style="list-style-type: none"><li>• Chapter Error control added</li><li>• Commands added: <i>info&lt;CR&gt;</i>, <i>version&lt;CR&gt;</i>, <i>v&lt;CR&gt;</i>, <i>fld&lt;CR&gt;</i>, <i>getTempLimiterMin&lt;CR&gt;</i>, <i>setTempLimiterMin&lt;CR&gt;</i>, <i>getTempLimiterMax&lt;CR&gt;</i>, <i>setTempLimiterMax&lt;CR&gt;</i></li><li>• Chapter 1. Device control adjusted</li><li>• Chapter 4. Settings for RS-232 control via PC or robot adjusted</li></ul>
<b>Version 1.0.1 v001</b> (Serial ≥ 11613)	<ul style="list-style-type: none"><li>• Commands added: <i>getDescription&lt;CR&gt;</i></li></ul>
<b>Version 1.0.0 v001</b>	Initial Creation

## 9. Geometric dimensions, drawings & images

**ColdPlate** (art. no. 2016-0110)



## 10. Support

We provide a range of technical material (e.g. application notes, bulletins, instruction manuals, and selection and use guides) that support our products and key applications.

All of our technical documents can be viewed and printed. Many documents are available as pdf files, which can be downloaded from our homepage.

Please contact QInstruments for additional information and availability about the BioShake.  
For this please use our online contact form or contact us directly via phone or email.

**Online:** [QInstruments.com](http://QInstruments.com)  
**Email:** [support@QInstruments.com](mailto:support@QInstruments.com)  
**Phone:** +49 3641 876120  
**Address:** Quantifoil Instruments GmbH  
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 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

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.

**BioShake®** (Quantifoil Instruments 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.

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

Notes