User Manual

LucidControl RT4

4 Channel RTD Input USB Module

1 Introduction

This document describes the functionality of the LucidControl RT4 USB module measuring 4 temperatures by using Pt100 / Pt1000 RTDs controllable via Universal Serial Bus.

A basic description of the complete LucidControl product family can be found in the document *LucidControl User Manual*.

This document concentrates on the topics that are specific to the RTD input module which is described here with all its details. In order to set up the module in a fast way please see the

LucidControl RT4 One Sheet Manual

which provides all information necessary to start working with the module out of the box without reading lots of documentation.

2 Setup and Installation

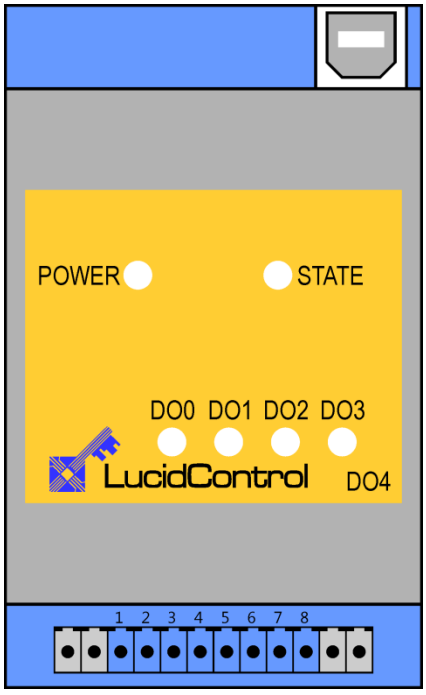

Fig. 1 RTD Input Module

Fig. 1 shows the sketch of the RTD Input RT4 module with 4 Pt100 / Pt1000 inputs (RT0 ~ RT3).

All LucidControl modules have two connectors, one USB connector and an IO- Connector which makes it easy to setup them.

While the upper USB connector is used for interconnection with the computer, the lower IO-Connector is used for inputs and outputs.

The IO-Connector provides 8 terminals in total - two for each input.

The intended use of the analog input module is the acquisition of temperatures. The module must only be used for the intended use.

For the analog input module it is explicitly stated that no potential of any external power source must be applied to any connector of the module. The modules must only be used within the specified conditions.

2.1 Interface and Interconnection

2.1.1 USB Connection

LucidControl USB modules are connected to the computer by using a standard USB cable which must not extend a length of 5 m. They are “bus powered” which means that the host computer supplies the module with power.

LucidControl RT4 is rated with a maximum current of 40 mA.

Note:
Supplying USB devices with power is not critical using a desktop computer or notebooks but it must be considered that the total power of one USB port is limited to 500 mA.

Note:
The USB ports of the Raspberry Pi® are limited to 100 mA. This means that maximum two devices can be connected to a port directly.

Note:
Using an active USB-Hub with its own power supply allows the connection of additional devices in the case that the host is not able to supply them.

2.1.2 IO Connection

The LucidControl RT4 module provides 4 channels measuring temperatures of Pt1000 RTDs.

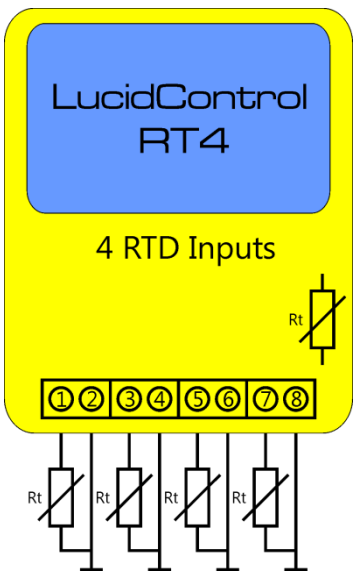

Fig. 2 RTD Module Connection

Fig. 2 shows the interconnection of the module with 4 RTDs connected to the IO-Connector.

The terminals 2, 4, 6 and 8 are internally connected to ground.

Only Resistors (RTDs) are allowed to be connected to the IO-Connector of the LucidControl RT4 module.

2.2 Setup of Hard- and Software

Setting up LucidControl hardware is extremely easy:

1 Ensure that no signal is applied to the IO Connector
2 Connect LucidControl via USB with the computer
3 Applies for Microsoft windows only: The system asks for an installation file. This is not a driver but only an information file (INF). The file can be downloaded from our website www.lucid-control.com/downloads
4 That's all. LucidControl switches the green power LED on and the module is ready for usage.

2.2.1 Windows

As mentioned the installation under Microsoft Windows requires the information file.

After finished installation the Windows Device Manager contains a new serial port (COM). The module can be accessed using this port.

Note:
Even if more than one module is connected to a computer Windows ensures that the same serial port number is assigned to the module(s) after restart.

2.2.2 Linux

Despite to Windows installation under Linux the module is usable immediately after connection without any additional steps. Linux installs /dev/ttyACM devices for any module connected to the computer.

Note:
By default Linux cannot ensure that the same /dev/ttyACM device is assigned to the same module on restart. But as long as only one module is connected to the computer it is ensured that it is accessible via /dev/ttyACM0.

This problem can be solved by the LucidIoCtrl command line tool which can create static devices always pointing to a specific module. Moreover the device can be given useful names e.g. dev/digitalIoKitchen.

Please see the section … of the general LucidIo User Manual for more information.

2.2.3 Get command line LucidIoCtrl

LucidIoCtrl command line tool can be downloaded from our website:

www.lucid-control.com/downloads

This page provides the command line tool LucidIoCtrl for different architectures.

After downloading the program can be stored in a folder of choice.

Please see the section … of the general LucidControl User Manual for more information about this helpful tool.

2.2.4 Ready to Start

Once the module was installed successfully (if it was necessary at all) the green Power LED is switched on signaling that the module is ready for use.

Since the module was preconfigured for standard input mode (see …) it can be used without further configuration. The following examples demonstrate the functionality of the module by using the LucidIoCtrl command line tool.

Windows Examples
For all examples it is assumed that the module is connected to COM1.

Reading the values of all 4 input channels

```
      LucidIoCtrl -dCOM1 -tT -c0,1,2,3 -r [ENTER]
->    CH0:25.000  CH1:25.000  CH2:25.000  CH3:25.000
```

Linux Examples:
For all examples it is assumed that the module is connected to /dev/ttyACM0.

Reading the values of all 4 input channels

```
      LucidIoCtrl -d/dev/ttyACM0 -tT -c0,1,2,3 -r [ENTER]
->    CH0:25.000  CH1:25.000  CH2:25.000  CH3:25.000
```

Note:
Obviously the temperatures can only be read if RTDs are connected to the clamps. Otherwise the module returns:

```
->    CH0:ERR_OPEN  CH1:ERR_OPEN  CH2:ERR_OPEN  CH3:ERR_OPEN
```

3 Module Operation

The measurement circuit sources the RTD with a constant current and measures the voltage drop over the RTD which is proportional to the resistance and represents a temperature.

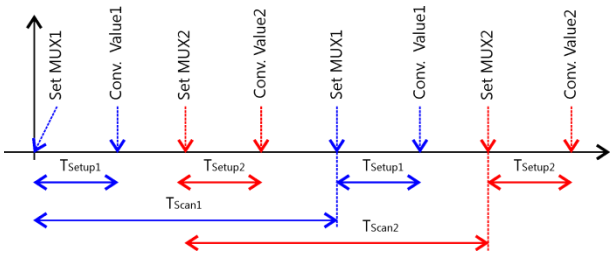

Fig. 3 Input Processing

Fig. 3 explains the measurement procedure in standard mode. The diagram shows two active RTD channels in order to keep the diagram simple.

The blue lines are related to measurement channel 1, the red lines to measurement channel 2.

It can be seen that the multiplexer is set to the appropriate input which sources the connected RTD with a constant current. Only one RTD is sourced with current at a given time. All active RTDs are processed subsequently by waiting the time T_{Scan}.

After an input channel was selected the input waits for a configurable time T_{Setup} until the resistance of the RTD is stable and ready for conversion.

T_{Setup} and T_{Scan} can be configured for each channel but it must be ensured that the times are chosen in a way that the scheduler is able to process each active channel.

Decreasing the values to not realistic timing may cause invalid values or may skip channels completely.

The default values given in Tab. 1 should only be changed if faster measurements are necessary or if the measured result is not stable.

Parameter	Time
T_{Scan}	500 ms
T_{Setup}	50 ms

Tab. 1 RTD Measurement Timing

This gives a periodical measurement of 500 ms for each RTD. After the RTD is sourced with current the input waits for 50 ms until the measured value is converted.

3.1 Operation Modes

3.1.1 Inactive Mode

In Inactive Mode the RTD measurement is disabled and the channel is skipped.

3.1.2 Standard Mode

In Standard Mode the RTD the inputs are measured as configured.

3.2 Offset Compensation

In order to compensate the offset of an input channel the Parameter *inRtOffset* can be used. (see 3.5.6)

This parameter allows configuring an offset resistance in 0.1 Ω steps for Pt1000 sensors and 0.01 Ω steps for Pt100 sensors. The value in Ω is added to the measured resistor value. Since *inRtOffset* can also be a negative value, compensation in both directions to higher or lower values is possible.

For the RT4 module a 0 offset compensation can be made by connecting a load resistor.

Measured Resistance: $R = R_M + R_{Offset}$

Since the temperature is approx. proportional $T[℃] \sim \frac{R}{10} * 0.256 \left[\frac{℃}{\Omega}\right]$ the value R_M changes the resulting temperature of a Pt1000 sensor by approx. $0.0256 \left[\frac{℃}{Digit}\right]$.

Example
When connecting a 1000 Ω resistor to the input channel 0 the resulting temperature should be exactly 0 °C. In this example a value of +0.5 °C is measured caused by outer influences e.g. the RTD itself. This offset value is compensated in the following procedure.

With the equations above it can be calculated that the deviation of +0.5 °C results in an offset correction value of -19.53 (rounded -20, which means -2.0 Ω)

The following function call adjusts the measured value to the load of 1000 Ω representing a value close to 0°C and stores the setting:

```
LucidIoCtrl –dCOM1 –c0 –sinRtOffset=-20 –p [ENTER]
```

3.3 Line Status Detection

The inputs are able to detect a broken or short cut sensor line.

In case that the measurement result is lower than a limit the module disregards the measured value and return a specified “line short cut” (ERR_SHORT) value.

In the case that the measurement result is higher than a limit the module disregards the measured value and return a specified “line broken” (ERR_OPEN) value.

Defined values for line status detection:

Value Type	Condition	Returned value
TMS4	$T < T_{NegLimit}$ (ERR_SHORT)	0x80000000
	$T > T_{PosLimit}$ (ERR_OPEN)	0x7FFFFFFF
TMS2	$T < T_{NegLimit}$ (ERR_SHORT)	0x8000
	$T > T_{PosLimit}$ (ERR_OPEN)	0x7FFF

3.4 Commands

Accessing inputs and outputs is a very common task which is mostly identical for all LucidControl modules. For this task input modules provide the commands GetIo for reading of a single value and GetIoGroup for reading of a group of values of the same type.

For more comprehensive information covering reading and writing of inputs and outputs please see the sections 3.2.1.1, 3.2.1.2 and 4.3 of the general LucidControl manual.

The following sections describe in detail the commands which are supported by the RT4 module.

3.4.1 GetIo

This command reads values of the RTD input channel.

Command	GetIo	**Access**	Read
Opcode	0x46		
LucidIoControl Command Line Tool			
Call (-tT)	`LucidIoCtrl -d[COMx] -c[Channel] -tT -r`		
Return	`CHn:tt` n – Input Channel tt – Temperature in °C		
Call (-tR)	`LucidIoCtrl -d[COMx] -c[Channel] -tR -r`		
Return	`CHn:rr` n – Input Channel rr – Resistance in Ω		

Note
When using the LucidIoCtrl command line tool the distinction between GetIo and GetIoGroup commands is not necessary since the program handles this automatically.

LucidIoCtrl Command Line Tool Example

Read temperature from input channel 0:

```
      LucidIoCtrl –dCOM4 –c0 –tT -r [ENTER]
->    CH0:100.200
```

Read the corresponding resistance of the same input

```
      LucidIoCtrl –dCOM4 –c0 –tR -r [ENTER]
->    CH0:1385.8
```

Line Status Detection:

The module is able to detect RTD shortcut and wire breaks in the connection cable (See 3.3). In the case of a wire break is detected the command returns `ERR_OPEN`. In the case that a shortcut is detected the command returns `ERR_SHORT`.

Request Frame

OPC	P1	P2	LEN
0x46	Channel	Value Type	0

Value	Description
Channel	Number of input or output channel (Range: 0 ~ 3)
Value Type	Supported Value Types

Value Type	Value Range	Size
Temperature Resolution : 1/100 °C (0x41)	-20,000 ~ 20,000 -200.00 °C ~ 200.00°C	4 Bytes
Temperature Resolution: 1/10 °C (0x40)	-2,000 ~ 2,000 -200.0 ~ 200.0 °C	2 Bytes
Resistance Resolution: 1/10 Ω (0x50)	0 ~ 65,535 0 ~ 6,553.5 Ω	2 Bytes
Resistance Resolution: 1/1000 Ω (0x51)	0 ~ 2^{32} -1 mΩ	4 Bytes

Tab. 2 GetIo Request

Response:

Status	LEN	Data
Status	Length	Value

In case of successful execution the command returns the value of the specified channel number.

In the case of an error the command returns Execution Status Code documented in section 4.4 of the LucidControl User Manual.

3.4.2 GetIoGroup

This command reads the input values of a group of inputs of the same Value Type. See also section 3.4.1.

<table>
<tr><td>Command</td><td>GetIoGroup</td><td>Access</td><td>Read</td></tr>
<tr><td>Opcode</td><td>0x48</td><td></td><td></td></tr>
<tr><td colspan="4">LucidIoControl Command Line Tool</td></tr>
<tr><td>Call (-tT)</td><td colspan="3"><code>LucidIoCtrl –d[COMx] –c[Channels] –tT –r</code>

<u>Channels:</u>
Comma separated list of channels e.g. –c0,1,3</td></tr>
<tr><td>Return</td><td colspan="3">List of values sorted from lower to higher channels
<code>CHn:tt</code><table><tr><td>n</td><td>Input Channel</td></tr><tr><td>tt</td><td>Temperature in °C</td></tr></table></td></tr>
<tr><td>Call (-tR)</td><td colspan="3"><code>LucidIoCtrl –d[COMx] –c[Channels] –tR –r</code>

<u>Channels:</u>
Comma separated list of channels e.g. –c0,1,3</td></tr>
<tr><td>Return</td><td colspan="3"><code>CHn:rr</code><table><tr><td>n</td><td>Input Channel</td></tr><tr><td>rr</td><td>Resistance in Ω</td></tr></table></td></tr>
</table>

<u>LucidIoCtrl Command Line Tool Example</u>
Read temperatures from all input channels:

```
      LucidIoCtrl –dCOM4 –c0,1,2,3 –tT –r [ENTER]
->    CH0:100.000  CH1:0.500  CH2:-100.300  CH3:78.250
```

Read temperatures form all input channels:

```
LucidIoCtrl –dCOM4 –c0,1,2,3 –tT –r [ENTER]
->    CH0:100.000  CH1:0.500  CH2:ERR_SHORT  CH3:ERR_OPEN
```

In this case the RTD connected to input channel 2 is shortcut and no RTD is connected to input channel 2.

<u>Line Status Detection:</u>
The module is able to detect RTD shortcut and wire breaks in the connection cable (See 3.3). In the case of a wire break is detected the command returns `ERR_OPEN`. In the case that a shortcut is detected the command returns `ERR_SHORT`.

Request Frame:

OPC	P1	P2	LEN
0x48	Ch-Mask	Value Type	0

<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>Channel Mask</td><td>Channel Mask
Specifies the output channels to access
<table>
<tr><th>Channel</th><th>Bit Position</th><th>Value</th></tr>
<tr><td>0</td><td>0</td><td>0x01</td></tr>
<tr><td>1</td><td>1</td><td>0x02</td></tr>
<tr><td>2</td><td>2</td><td>0x04</td></tr>
<tr><td>3</td><td>3</td><td>0x08</td></tr>
</table>
Values are bitwise or combined

Examples:
Accessing channel 0 and 3 Value = 0x01 OR 0x08 = 0x09
Accessing channel 1 and 2 Value = 0x02 OR 0x04 = 0x06</td></tr>
<tr><td>Value Type</td><td>Supported Value Types
<table>
<tr><th>Value Type</th><th>Value Range</th><th>Size</th></tr>
<tr><td>Temperature
Resolution : 1/100 °C
(0x41)</td><td>-20,000 ~ 20,000
-200,00 °C ~
200,00°C</td><td>4 Bytes</td></tr>
<tr><td>Temperature
Resolution: 1/10 °C
(0x40)</td><td>-2,000 ~ 2000
-200,0 ~ 200,0 °C</td><td>2 Bytes</td></tr>
<tr><td>Resistance
Resolution: 1/10 Ω
(0x50)</td><td>0 ~ 65535
0 ~ 6553,5 Ω</td><td>2 Bytes</td></tr>
<tr><td>Resistance
Resolution: 1/1000 Ω
(0x51)</td><td>0 ~ 2^{32} -1 mΩ</td><td>4 Bytes</td></tr>
</table></td></tr>
</table>

Tab. 3 GetIoGroup Request

Response Frame:

Status	LEN	Data Field
Status	Length	Value(s)

In case of successful execution the command returns the read values of the channels specified in the Channel Mask.

In the case of an error the command returns Execution Status Code documented in section 4.4 of the LucidControl User Manual.

Example of GetIoGroup Request:
The following request frame reads temperatures from input channels 0 and 1

Opcode	P1	P2	Length
0x48	0x03	0x41	0x00

Channel Mask (P1): 0x01 OR 0x02 = 0x03

Response Frame:
For input 0 = 50°C , input 1 = -25°C
Values in Data Field are in ascending order Channel 0, Channel 1.

Header Field		Data Field							
Status	LEN	Value Channel 0				Value Channel 1			
0x00	0x08	0x88	0x13	0x00	0x00	0x3C	0xF6	0xFF	0xFF

3.5 Parameters

LucidControl modules allow configuration by a set of System Configuration Parameters and IO Configuration Parameters.

The Parameters are accessible via the SetParam and GetParam command which are described in sections 4.3.5 and 4.3.6 of the LucidControl User Manual.

3.5.1 inRtValue

This IO Configuration Parameter contains the measured resistance value with a resolution of 0.1 Ω.

Parameter	inRtValue	**Access**	Read
Address	0x1000		
Values	Measured Resistance in 0.1 Ω		
Default Value	0x00	**Parameter Type**	2 Bytes unsigned
LucidIoControl Command Line Tool			
Parameter Name	inRtValue	**Parameter Values**	Resistor [1/10 Ω] 0 ~ 6553,5 Ω
Call (Get)	`LucidIoCtrl –d[COMx] –c[Channel] –ginRtValue`		

LucidIoCtrl Command Line Tool Example
Read value parameter of input channel 0:

```
LucidIoCtrl –dCOM4 –c0 –ginRtValue [ENTER]
-> inDiValue=100
```

The measured value of 10.0 Ω is returned.

Note:
For normal operation is recommended to use the function GetIo (see 3.4.1) in order to read the input value. The parameter provides the resistance value only.

3.5.2 inRtMode

This IO Configuration Parameter configures the operation mode of the input.

<table>
<tr><td>Parameter</td><td>inRtMode</td><td>Access</td><td>Read / Write</td></tr>
<tr><td>Address</td><td>0x1100</td><td colspan="2"></td></tr>
<tr><td>Values</td><td colspan="3">Input Mode<table><tr><th>Byte</th><th>Mode</th></tr><tr><td>0x00</td><td>inactive</td></tr><tr><td>0x01</td><td>standard</td></tr></table></td></tr>
<tr><td>Default Value</td><td>standard</td><td>Parameter Type</td><td>1 Byte unsigned</td></tr>
<tr><td colspan="4">LucidIoControl Command Line Tool</td></tr>
<tr><td>Parameter Name</td><td>inRtMode</td><td>Parameter Values</td><td>inactive / standard</td></tr>
<tr><td>Call (Set)</td><td colspan="3"><code>LucidIoCtrl –d[COMx] –c[Channel] –sinRtMode=[Mode] {-p} {--default}</code></td></tr>
<tr><td>Call (Get)</td><td colspan="3"><code>LucidIoCtrl –d[COMx] –c[Channel] –ginRtMode</code></td></tr>
</table>

LucidIoCtrl Command Line Tool Example
Set operation mode of input channel 0 to standard mode and make the setting persistent.

```
LucidIoCtrl –dCOM4 –c0 –sinRtMode=standard –p [ENTER]
```

Read the operation mode of input channel 0

```
LucidIoCtrl –dCOM4 –c0 –ginRtMode [ENTER]
->inRtMode=standard
```

3.5.3 Bit Parameter inRtFlags

This IO Configuration Parameter groups parameters which are represented by one bit e.g. having an “on” or “off” state only).

<table>
<tr><td>Parameter</td><td>inRtFlags</td><td>Access</td><td>Read / Write</td></tr>
<tr><td>Address</td><td>0x1101</td><td colspan="2"></td></tr>
<tr><td>Values</td><td colspan="3">Consists of the following Bit Parameters<table><tr><th>Bit Parameter</th><th>Bit Postion</th></tr><tr><td>inRtHighRes</td><td>Bit 0</td></tr></table></td></tr>
<tr><td>Default Value</td><td>0x00</td><td>Parameter Type</td><td>1 Byte unsigned</td></tr>
</table>

Note
The parameter *inRtFlags* cannot be accessed directly by using the Command Line Tool. The Bit Parameters can be used instead.

Note:
When *inRtFlags* is changed by the SetParam command which is described in section 4.3.5 of the LucidControl User Manual the current setting of *inRtFlags* must be read before updating it in order to prevent overwriting other Bit Parameters.

3.5.3.1 inRtHighRes

This Bit Parameter configures the resolution of the measurement. By enabling this Bit Parameter the accuracy of the measured result is increased by oversampling.

Parameter	inRtFlags	**Access**	Read / Write
Address	0x1101	Bit Parameter in inRtFlags	
Values	**Bit Parameter**: *inRtHighRes* — **Bit Postion**: Bit 0		
Default Value	Off	**Parameter Type**	1 Bit
LucidIoControl Command Line Tool			
Parameter Name	*inRtHighRes*	**Parameter Values**	on / off
Call (Set)	`LucidIoCtrl -d[COMx] -c[Channel] -sinRtHighRes=[Value] {-p} {--default}`		
Call (Get)	`LucidIoCtrl -d[COMx] -c[Channel] -ginRtHighRes`		

LucidIoCtrl Command Line Tool Example
Enable high resolution of input channel 0 and make the setting persistent.

```
LucidIoCtrl -dCOM4 -c0 -sinRtHighRes=on -p [ENTER]
```

Read inversion of physical input channel 0

```
LucidIoCtrl -dCOM4 -c0 -ginRtHighRes [ENTER]
->  inRtHighRes=on
```

3.5.4 inRtScanTime

This IO Configuration Parameter configures the scan time T_{Scan} of the input channel.

Parameter	inRtScanTime	**Access**	Read / Write
Address	0x1111		
Values	T_{Scan} in ms (milli seconds) 50 ms ≤ T_{Scan} ≤ 10 s		
Default Value	500 (500 ms)	**Parameter Type**	2 Bytes unsigned
LucidIoControl Command Line Tool			
Parameter Name	inRtScanTime	**Parameter Values**	Time [ms]
Call (Set)	`LucidIoCtrl –d[COMx] –c[Channel] –sinRtScanTime=[Time] {-p} {--default}`		
Call (Get)	`LucidIoCtrl –d[COMx] –c[Channel] –ginRtScanTime`		

LucidIoCtrl Command Line Tool Example

Set T_{Scan} of input channel 0 to 250 ms and make the setting persistent.

```
LucidIoCtrl –dCOM4 –c0 –sinRtScanTime=250 –p [ENTER]
```

Read T_{Scan} parameter of input channel 0

```
LucidIoCtrl –dCOM4 –c0 –ginRtScanTime [ENTER]
-> inRtScanTime=250
```

3.5.5 inRtSetupTime

This IO Configuration Parameter configures the input channel setup time T_{Setup}.

Parameter	inRtSetupTime	**Access**	Read / Write
Address	0x1112		
Values	T_{Scan} in ms (milli seconds) 5 ms ≤ T_{Scan} ≤ 1 s		
Default Value	50 (50 ms)	**Parameter Type**	2 Bytes unsigned
LucidIoControl Command Line Tool			
Parameter Name	inRtSetupTime	**Parameter Values**	Time [ms]
Call (Set)	`LucidIoCtrl –d[COMx] –c[Channel] –sinRtSetupTime=[Time] {-p} {--default}`		
Call (Get)	`LucidIoCtrl –d[COMx] –c[Channel] –ginRtSetupTime`		

LucidIoCtrl Command Line Tool Example

Set T_{Setup} of input channel 0 to 25 ms and make the setting persistent.

```
LucidIoCtrl –dCOM4 –c0 –sinRtSetupTime=25 –p [ENTER]
```

Read T_{Scan} parameter of input channel 0

```
LucidIoCtrl –dCOM4 –c0 –ginRtSetupTime [ENTER]
-> inRtSetupTime=25
```

3.5.6 inRtOffset

This IO Configuration Parameter configures the Input Offset Compensation value which is described in section 3.2.

Parameter	inRtOffset	**Access**	Read / Write
Address	0x1120		
Values	Pt1000: Offset Compensation in 0.1 Ω steps (-1,000 Ω ~ 1,000 Ω) -10,000 ~ 10,000 Pt100: Offset Compensation in 0.01 Ω steps (-100 Ω ~ 100 Ω) -10,000 ~ 10,000		
Default Value	0	**Parameter Type**	2 Bytes signed
LucidIoControl Command Line Tool			
Parameter Name	inRtOffset	**Parameter Values**	Resistance [0.1 Ω]
Call (Set)	`LucidIoCtrl –d[COMx] –c[Channel] –sinRtOffset=[Offset] {-p} {--default}`		
Call (Get)	`LucidIoCtrl –d[COMx] –c[Channel] –ginRtOffset`		

LucidIoCtrl Command Line Tool Example

Set Input Offset Compensation value of the Rt1000 input channel 0 to -2 Ω and make the setting persistent.

```
LucidIoCtrl –dCOM4 –c0 –sinRtOffset=-20 –p [ENTER]
```

Read Input Offset Compensation value

```
LucidIoCtrl –dCOM4 –c0 –ginRtOffset [ENTER]
-> inRtOffset=20
```

4 Specification

Parameter		Condition	Value	
Inputs				
	No of Input Channels		4	
Input - Electrical Characteristics				
	Measurement Method		RTD Two wire measurement	
	Resolution		0.1°C	
	RTD Type		Pt100 / Pt1000 DIN IEC 751	
	Measurement Error		typ. +/- 0.5°C	
	Constant Measurement Current	Pt100	1mA	
		Pt1000	0.5 mA	
Input – Timing Characteristic				
	Measurement Interval		T_{Scan}	50 ms ≤ t ≤ 10 s
	Setup Time		T_{Setup}	5 ms ≤ t ≤ 1 s
Module – Communication				
	USB		2.0 Full Speed CDC Profile	
Module – Electrical Characteristics				
	Power Supply		USB Bus Powered with +5V No additional Power Supply needed.	
	Maximum Rated Supply Current			40 mA
Module – Environment				
	Temperature	Storage	-20 °C ~ +70 °C	
		Operation	0 °C ~ +55 °C	
	Humidity		< 85 % RH, non-condensing	
Module – Housing				
	Dimension L x W x H		90 x 54 x 62 mm	
	Weight (in total)		120 g	
	Assembly		Rail-Mount (EN 50022, TS35)	
	Protection Class (DIN 40050)		IP20	
Module - Indicators				
	• Operation and Error Indicator • Communication Indicator			

5 Order Information and Accessories

Digital Input Product Family

Order Code	Product
LCTR-RT4-1000	LucidControl RTD Input USB Module with 4 channels for Pt1000 Sensors
LCTR-RT4-100	LucidControl RTD Input USB Module with 4 channels for Pt100 Sensors

The following accessories are available:

Order Code	Product
LCTR-AK1710-8	Plug-In Terminal 8-way 1,5 mm^2 wire