User Manual

Lucid485 RI4/RI8

4/8 Channel RTD Input Serial Module

Table of Content

1 Introduction

This document describes the functionality of the Lucid485 RI4/RI8 serial module with 4/8 Pt100/Pt1000 RTD input channels.

A basic description of the Lucid485 product family can be found in the general *Lucid485 User Manual*.

2 Setup and Installation

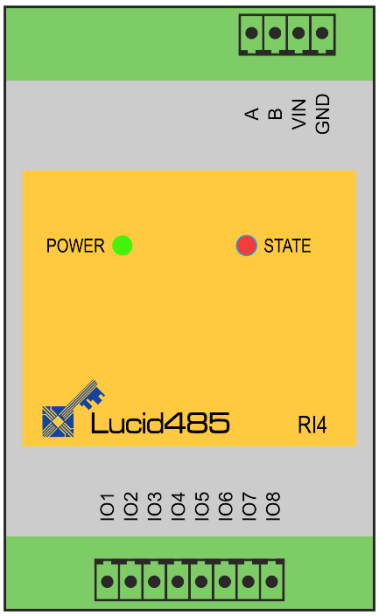

Fig. 1 Lucid485 RI4 Module

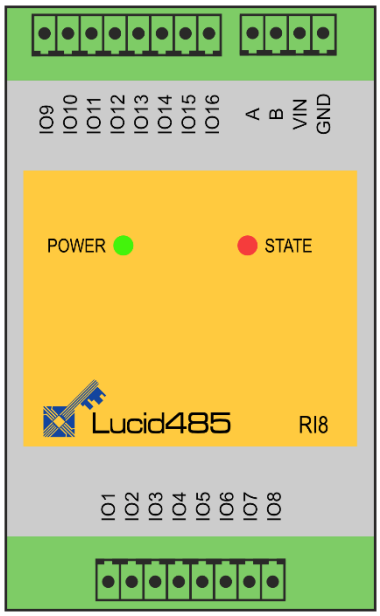

Fig. 2 Lucid485 RI8 Module

Fig. 1 and Fig. 2 show drawings of the RI4 and RI8 RTD input modules 4/8 Pt100/Pt1000 input channels.

The IO signals are connected to the lower (IO1 - IO8) and the upper (IO9 - IO16) IO terminals.

The upper IO terminal connector is present at the RI8 module only.

2.1 Safety Information

LucidControl complies with regulations and industrial standards active in the EU. To keep the device functional, the following safety and maintenance information must be adhered.

The device must only be used for the intended purpose.

The device must not be used under the following conditions:

- It is obviously damaged
- An error was detected
- Outside humidity and temperature limits
- Unauthorized personnel

The intended use of the RI4/RI8 module is the acquisition of temperatures. The module must only be used for the intended use.

For this device it is explicitly stated that no potential (e.g. voltage) of any external power source must be applied to any connector of the module. The modules must only be used within the specified conditions.

2.2 RS-485 bus and Power Connection

Please see the general *Lucid485 User Manual*.

2.3 LucidIoCtrl Command Line Tool

The LucidIoCtrl command line tool gives full access to all Lucid485 modules operating with Frame Protocol enabled. Executables for different architectures and can be downloaded from our website:

https://www.lucid-control.com/downloads

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

Please see the general *Lucid485 User Manual* for more information.

2.3.1 First Steps

When the module is powered, the green power LED is switched on, signaling that the module is ready.

The following examples demonstrate the functionality of the module by using the LucidIoCtrl command line tool.
It is assumed that the module is configured with default address 11, 9600 baud, 8N1.
The serial to RS-485 adapter is connected to COM1 of the host.

Windows Examples

Reading the values of the first 4 input channels

```
      LucidIoCtrl -drs485:COM4:11 –tT –c0,1,2,3 –r
->    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 the first 4 input channels

```
      LucidIoCtrl -drs485:/dev/ttyACM0:11 –tT –c0,1,2,3 –r [ENTER]
->    CH0:25.000  CH1:25.000  CH2:25.000  CH3:25.000
```

2.4 IO Configurations

The LucidControl RI4/RI8 module is available in different configurations:

Function Class	Value	Channels
RI4	0x8A00	4
RI8	0x8A10	8

Tab. 1 RTD Input Function Class

Function Class Type	Value	Sensor Type	Temp. Range
Pt1000	0x1000	Pt1000	±180 °C
Pt1000C360	0x1001	Pt1000	0 – 360 °C
Pt100	0x1010	Pt100	±180 °C
Pt100C360	0x1011	Pt100	0-360 °C

Tab. 2 RTD Input Function Class Types

Tab. 1 and Tab. 2 list the Function Classes and Function Class Types.

The RTD sensor types and temperature ranges are listed in Tab. 2.

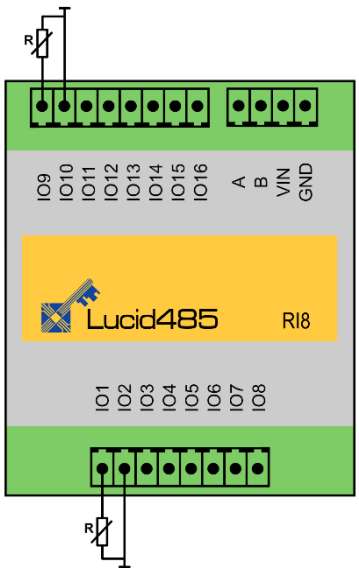

Fig. 3 RI8 IO Connection

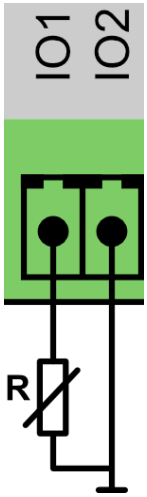

Fig. 4 RI Signal

Fig. 3 shows 2 RTD sensors connected to the IO terminals of the channels 0 and 4 of the RI8 module.

Fig. 4 shows in detail the RTD sensor connected to IO terminals IO1 and IO2 (input channel 0).

IO Terminal	Signal	IO Channel Number
1	IN0	0
2	GND	
3	IN1	1
4	GND	
5	IN2	2
6	GND	
7	IN3	3
8	GND	
9	IN4	4
10	GND	
11	IN5	5
12	GND	
13	IN6	6
14	GND	
15	IN7	7
16	GND	

Tab. 3 RI4 / RI8 IO Terminal Connector

Tab. 3 lists the IO terminals and the RTD Input signals.

Only resistors (RTDs) must be connected to the IO-terminals.

3 Module Operation

The Lucid485 RI4/RI8 RTD Input Module measures voltages or currents applied to an RTD sensor and calculates resistance and temperature values.

3.1 Channel Processing

The data acquisition of the RI4/RI8 selects the active input channels subsequently, sourcing them with a current for a configurable time T_{Setup} and measuring the current and voltage at the input channel.

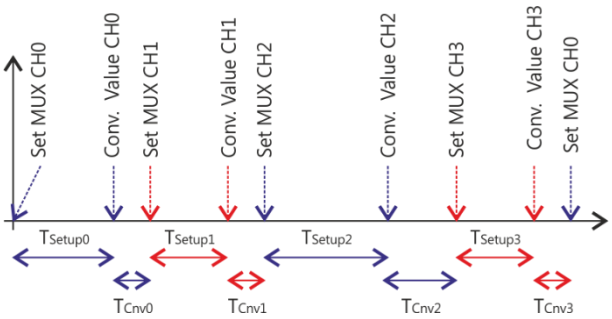

Fig. 5 explains the measurement procedure. The diagram shows the subsequent measurement of the input channels CH0 to CH3.

Fig. 5 Input Processing

At a given time, the algorithm selects one input channel and all others are deselected.

At first, channel CH0 is selected and the multiplexer is set to channel CH0, sourcing the RTD with the measurement current. After the time T_{Setup} has passed the data conversion starts. Depending on the conversion settings the result is ready after the conversion time T_{Conv}. When the result is ready, it is stored and the next channel is selected.

This procedure continues for channels CH1, CH2 and CH3, and for the RI8 module also for the channels CH4, CH5, CH6 and CH7. After the last channel was processed, the first channel is selected again and the conversion loop continues from beginning.

3.1.1 Measurement Timing

The procedure explained in Fig. 5 takes some time until it completes and the new result is ready USB.

The acquisition time of the measurement can be changed for each channel by the parameters *inRtSetupTime* (→ 3.7.5) and *inRtNrSamples* (→ 3.7.3).

The parameter *inRtSetupTime* specifies the time T_{Setup}. After a channel is selected the conversion will be started after T_{Setup} has passed.

The parameter *inRtNrSamples* specifies the number of oversampling cycles. Oversampling can give additional accuracy by sampling the analog values multiple times. In theory, this can also give additional resolution.

Channels that are not used can be set can be set to Inactive Mode (→ 3.2.1, → 3.7.2). Inactive channels are skipped and not processed, improving the timing of active channels.

Typical Measurement Timing Examples

Nr. of active Channels	Nr. of Samples	T_{Setup} [ms]	T_{Cycle} [ms]	Remarks
4	16	25	160	Default for RI4 module
4	16	10	100	
4	16	5	80	
2	16	25	80	
2	16	5	40	
4	4	25	116	
4	4	5	36	
8	16	25	330	Default for RI8 module
8	16	5	170	
8	4	5	72	

Tab. 4 Measurement Timing

Tab. 4 shows the time T_{Cycle} for some parameter settings of *inRtNrSamples* and *inRtSetupTime*. T_{Cycle} is the acquisition time interval and after T_{Cycle} all channels have been updated.

Using the default settings, the value of the channels is updated every 160ms for the RI4 module and every 330ms for the RI8 module.

Note
There are many combinations of *inRtSetupTime* and *inRtNrSamples* possible but we recommend only using the settings from the table above.

3.2 Operation Modes

3.2.1 Inactive Mode

In Inactive Mode the RTD measurement is disabled and the channel is skipped, increasing the conversion speed of active input channels.

3.2.2 Standard Mode

Input channel is active and signals are converted in a round robin sequence.

3.3 Environment Temperature Compensation

The measurement of Pt100 sensor signals is more sensitive than Pt1000 sensor signals. Components of the measurement circuit (e.g. resistances) vary slightly over environment temperature, causing errors which are compensated by this function.

The environment temperature compensation is configured by the parameter *inRtTempComp* (→ 3.7.3.3). The parameter is only effective for Pt100 input channels.

3.4 Offset Compensation

In order to compensate the offset of an input channel the Parameter *inRtOffset* can be used. (see 3.7.6)

This parameter allows setting up 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 RI4/RI8 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 -drs485:COM4:11 –c0 –sinRtOffset=-20 –p
```

3.5 Line Status Detection

The inputs channel can detect a broken or short-cut sensor line. The detection can be activated with *inRtTestOpen* (→ 3.7.3.1) and *inRtTestShort* (→ 3.7.3.2) parameters.

If the parameter *inRtTestOpen* is enabled and a broken line is detected, the IO value is set to ERR_OPEN.

If the parameter *inRtTestShort* is enabled and a shortcut is detected, the IO value is set to ERR_SHORT.

Value Type	Condition	Returned value
TMS4	ERR_SHORT	0x80000000
	ERR_OPEN	0x7FFFFFFF
TMS2	ERR_SHORT	0x8000
	ERR_OPEN	0x7FFF
RMU2	ERR_SHORT	0
	ERR_OPEN	0xFFFF
RMU4	ERR_SHORT	0
	ERR_OPEN	0xFFFFFFFF

Tab. 5 RTD Input Line State Errors

3.6 Commands

Lucid485 IO Modules can be accessed by the Network Frame Protocol, which is documented in the general *Lucid485 User Manual*.

This section describes in detail the commands which are supported by the RI4/RI8 modules.

3.6.1 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:addr] -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		

Tab. 6 GetIo Command

LucidIoCtrl Command Line Tool Example
Read temperature from input channel 0.

```
      LucidIoCtrl -drs485:COM4:11 -c0 -tT -r
->    CH0:100.200
```

Read the corresponding resistance of the same input.

```
      LucidIoCtrl -drs485:COM4:11 -c0 -tR -r
->    CH0:1385.8
```

Request Frame

DST	SRC	OPC	P1	P2	LEN	CRC
DST	SRC	0x46	Channel	Value Type	0	CRC-16

Tab. 7 GetIo Network Requet Frame

Value	Description
Channel	Number of input or output channel (Range: 0 to 7)
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. 8 GetIo Request

Response

Status	LEN	Data
Status	Length	Value

Tab. 9 GetIo Network Response Frame

Returns Execution Status Code, documented in the general *Lucid485 User Manual*.

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

3.6.2 GetIoGroup

This command reads the values of a group of input channels of the same Value Type.

<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:addr] –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:addr] –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>

Tab. 10 GetIoGroup Command

<u>LucidIoCtrl Command Line Tool Example</u>

Read temperatures from input channels 0, 1, 2 and 7.

```
      LucidIoCtrl -drs485:COM4:11 –c0,1,2,7 –tT –r
->    CH0:100.000  CH1:0.500  CH2:-100.300  CH7:78.250
```

Read temperatures form input channels 0, 1, 2 and 7.

```
      LucidIoCtrl -drs485:COM4:11 –c0,1,2,7 –tT –r
->    CH0:100.000  CH1:0.500  CH2:ERR_SHORT  CH7:ERR_OPEN
```

In this case the RTD connected to input channel 2 is shortcut and no RTD is connected to input channel 7.

<u>Request Frame</u>

OPC	P1	P2	LEN
0x48	Channel Mask	Value Type	0

Tab. 11 GetIoGroup Network Request Frame

<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>Channel Mask</td><td>Channel Bit Mask specifying the channel number(s)
<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>
<tr><td>4</td><td>4</td><td>0x10</td></tr>
<tr><td>5</td><td>5</td><td>0x20</td></tr>
<tr><td>6</td><td>6</td><td>0x40</td></tr>
<tr><td>7</td><td>P1A 0</td><td>P1=0x80
P1A = 0x01</td></tr>
</table>
Values can be bitwise combined.

Size of P1 is 1 or 2 bytes. If Bit 7 of P1 is set, a subsequent P1A is expected.

<u>Examples</u>
Accessing channel numbers:
0 and 3 Value P1 = 0x01 OR 0x08 = 0x09
1 and 2 Value P1 = 0x02 OR 0x04 = 0x06
1, 2 and 7 Value P1 = 0x02 OR 0x04 = 0x86
Value P1A = 0x01 (for channel 7)</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. 12 GetIoGroup Request

<u>Response Frame</u>

DST	SRC	Status	LEN	Data Field	CRC
DST	SRC	Status	Length	Value(s)	CRC-16

Tab. 13 GetIoGroup Network Response Frame

Returns Execution Status Code, documented in the general *Lucid485 User Manual*.

In case of successful execution, the command returns the read values of the channels specified in the Channel Mask in channel number ascending order.

Example of GetIoGroup Request
Read temperatures from input channels 0 and 1

DST	SRC	OPC	P1	P2	LEN	CRC
0x0B	0x0A	0x48	0x03	0x41	0x00	0x8A4E

Tab. 14 GetIoGroup Network Request Example

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.

DST	SRC	Status	LEN	Data Field		CRC
0x0A	0x0B	0x00	0x08	CH 0	CH 1	0x299C
				...	...	

Tab. 15 GetIoGroup Network Response Example

CH 0				CH1			
0	1	2	3	4	5	6	7
0x88	0x13	0x00	0x00	0x3C	0xF6	0xFF	0xFF

Tab. 16 GetIoGroup Example Channel Values

3.7 Parameters

Lucid485 modules are configured by a set of System Configuration Parameters and IO Configuration Parameters.

The parameters are accessible by the SetParam and GetParam commands, which are described in the general *Lucid485 User Manual*.

Parameter values can be made persistent in the non-volatile memory of the microcontroller e.g., by adding the -p argument to LucidIoCtrl or by setting the persistent parameter in the API function to true. Values of persistent parameters are restored when Lucid485 is powered on.

The number of write cycles to the non-volatile memory is limited. Write operations wear out the non-volatile memory and periodical updates of persistent parameters should be avoided in order not to destroy the device over time.

3.7.1 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:addr] -c[Channel] -ginRtValue`		

Tab. 17 IO Configuration Parameter inRtValue

LucidIoCtrl Command Line Tool Example

Read value parameter of input channel 0.

```
LucidIoCtrl -drs485:COM4:11 -c0 -ginRtValue
-> inRtValue=100
```

The measured value of 10.0 Ω is returned.

3.7.2 inRtMode

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

Parameter	*inRtMode*	**Access**	Read / Write
Address	0x1100		
Values	Input Mode **Byte** 0x00 = inactive 0x01 = standard		
Default Value	standard	**Parameter Type**	1 Byte unsigned
LucidIoControl Command Line Tool			
Parameter Name	*inRtMode*	**Parameter Values**	inactive / standard
Call (Set)	`LucidIoCtrl -d[COMx:addr] -c[Channel] -sinRtMode=[Mode] {-p} {--default}`		
Call (Get)	`LucidIoCtrl -d[COMx:addr] -c[Channel] -ginRtMode`		

Tab. 18 IO Configuration Parameter inRtMode

LucidIoCtrl Command Line Tool Example

Wet operation mode of input channel 0 to standard mode and make the setting persistent.

```
LucidIoCtrl -drs485:COM4:11 –c0 –sinRtMode=standard –p
```

Read the operation mode of input channel 0.

```
LucidIoCtrl -drs485:COM4:11 –c0 –ginRtMode
->inRtMode=standard
```

3.7.3 Bit Parameter inRtFlags

This IO Configuration Parameter groups configuration settings which are represented by one bit.

Parameter	*inRtFlags*	**Access**	Read / Write
Address	0x1101		
Values	Consists of the following Bit Parameters **Bit Parameter** / **Bit Position** *inRtTestOpen* / Bit 0 *inRtTestShort* / Bit 1 *inRtTempComp* / Bit 4		
Default Value	0x00	**Parameter Type**	1 Byte unsigned

Tab. 19 IO Configuration Parameter inRtFlags

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

When *inRtFlags* is changed by the SetParam command, it must be done in a read-modify-write sequence in order to prevent overwriting other parameter bits.

3.7.3.1 inRtTestOpen

This Bit Parameter configures open line test.

Parameter	*inRtFlags*	**Access**	Read / Write
Address	0x1101	Parameter bit *inRtFlags*	
Values	**Bit Parameter** / **Bit Position** *inRtTestOpen* / Bit 0		
Default Value	Off	**Parameter Type**	1 Bit
LucidIoCtrl Command Line Tool			
Parameter Name	*inRtTestOpen*	**Parameter Values**	on/off
Call (Set)	`LucidIoCtrl -d[COMx:addr] –c[Channel] –sinRtTestOpen=[Value] {-p} {--default}`		
Call (Get)	`LucidIoCtrl -d[COMx:addr] –c[Channel] -ginRtTestOpen`		

Tab. 20 IO Configuration Parameter Bit inRtTestOpen

3.7.3.2 inRtTestShort

This Bit Parameter configures short line test.

Parameter	*inRtFlags*	**Access**	Read / Write
Address	0x1101	Parameter bit *inRtFlags*	
Values	**Bit Parameter** / **Bit Position**: *inRtTestShort* / Bit 1		
Default Value	Off	**Parameter Type**	1 Bit
LucidIoCtrl Command Line Tool			
Parameter Name	*inRtTestShort*	**Parameter Values**	on/off
Call (Set)	`LucidIoCtrl -d[COMx:addr] -c[Channel] -sinRtTestShort=[Value] {-p} {--default}`		
Call (Get)	`LucidIoCtrl -d[COMx:addr] -c[Channel] -ginRtTestShort`		

Tab. 21 IO Configuration Parameter Bit inRtTestShort

3.7.3.3 inRtTempComp

This Bit Parameter configures the environment temperature compensation.

Parameter	*inRtFlags*	**Access**	Read / Write
Address	0x1101	Parameter bit *inRtFlags*	
Values	**Bit Parameter** / **Bit Position**: *inRtTempComp* / Bit 4		
Default Value	Off	**Parameter Type**	1 Bit
LucidIoCtrl Command Line Tool			
Parameter Name	*inRtTempComp*	**Parameter Values**	on/off
Call (Set)	`LucidIoCtrl -d[COMx:addr] -c[Channel] -sinRtTempComp=[Value] {-p} {--default}`		
Call (Get)	`LucidIoCtrl -d[COMx:addr] -c[Channel] -ginRtTempComp`		

Tab. 22 IO Configuration Parameter Bit inRtTempComp

3.7.4 inRtNrSamples

This IO Configuration Parameter configures the number of oversampling cycles. See also section 3.1.1.

Parameter	*inRtNrSamples*	**Access**	Read / Write
Address	0x1113		
Values	1, 2, 4, 8, 16, 32, 64, 128, 256 oversampling cycles		
Default Value	16	**Parameter Type**	2 Bytes unsigned
LucidIoControl Command Line Tool			
Parameter Name	*inRtNrSamples*	**Parameter Values**	Cycles
Call (Set)	`LucidIoCtrl -d[COMx:addr] -c[Channel] -sinRtNrSamples=[cycles] {-p} {--default}`		
Call (Get)	`LucidIoCtrl -d[COMx:addr] -c[Channel] -ginRtNrCycles`		

Tab. 23 IO Configuration Parameter inRtNrSamples

LucidIoCtrl Command Line Tool Example

Set number or oversampling cycles for channel 0 to 8 and make the setting persistent.

```
LucidIoCtrl -drs485:COM4:11 -c0 -sinRtNrSamples=8 -p
```

Read number of oversampling cycles of channel 0

```
LucidIoCtrl -drs485:COM4:11 -c0 -ginRtNrSamples
-> inRtNrSamples=8
```

3.7.5 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	25 (25 ms)	**Parameter Type**	2 Bytes unsigned
LucidIoCtrl Command Line Tool			
Parameter Name	*inRtSetupTime*	**Parameter Values**	Time [ms]
Call (Set)	`LucidIoCtrl -d[COMx:addr] -c[Channel] -sinRtSetupTime=[Time] {-p} {--default}`		
Call (Get)	`LucidIoCtrl -d[COMx:addr] -c[Channel] -ginRtSetupTime`		

Tab. 24 IO Configuration Parameter inRtSetupTime

LucidIoCtrl Command Line Tool Example

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

```
LucidIoCtrl -drs485:COM4:11 -c0 -sinRtSetupTime=25 -p
```

Read T_{Scan} parameter of input channel 0

```
LucidIoCtrl -drs485:COM4:11 -c0 -ginRtSetupTime
-> inRtSetupTime=25
```

3.7.6 inRtOffset

This IO Configuration Parameter configures the Input Offset Compensation Value

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
LucidIoCtrl Command Line Tool			
Parameter Name	*inRtOffset*	**Parameter Values**	Resistance [0.1 Ω]
Call (Set)	`LucidIoCtrl -d[COMx:addr] –c[Channel] –sinRtOffset=[Offset] {-p} {--default}`		
Call (Get)	`LucidIoCtrl –-d[COMx:addr] –c[Channel] –ginRtOffset`		

Tab. 25 IO Configuration Parameter inRtOffset

LucidIoCtrl Command Line Tool Example

Set Input Offset Compensation value of the Pt1000 input channel 0 to -2 Ω and make the setting persistent.

```
LucidIoCtrl -drs485:COM4:11 –c0 –sinRtOffset=-20 –p
```

Read Input Offset Compensation value

```
LucidIoCtrl -drs485:COM4:11 –c0 –ginRtOffset
-> inRtOffset=20
```

4 Modbus RTU Protocol

The IO values can be accessed by Modbus holding registers listed in Tab. 26.

IO Channel Values contain the Temperature in 1/10 °C (signed)
Alternative IO Channel Values contain the Resistor value:

- 1/10 Ω for Pt1000 Sensor Types
- 1/100 Ω for Pt100 Sensor Types

Address	Type	Width	Description
0x2000	Holding	16	IO Channel Number 0 Value (RI0)
0x2001	Holding	16	IO Channel Number 1 Value (RI1)
0x2002	Holding	16	IO Channel Number 2 Value (RI2)
0x2003	Holding	16	IO Channel Number 3 Value (RI3)
0x2004	Holding	16	IO Channel Number 4 Value (RI4)
0x2005	Holding	16	IO Channel Number 5 Value (RI5)
0x2006	Holding	16	IO Channel Number 6 Value (RI6)
0x2007	Holding	16	IO Channel Number 7 Value (RI7)
0x2080	Holding	16	Alternative IO Channel Number 0 Value (RI0)
0x2081	Holding	16	Alternative IO Channel Number 1 Value (RI1)
0x2082	Holding	16	Alternative IO Channel Number 2 Value (RI2)
0x2083	Holding	16	Alternative IO Channel Number 3 Value (RI3)
0x2084	Holding	16	Alternative IO Channel Number 4 Value (RI4)
0x2085	Holding	16	Alternative IO Channel Number 5 Value (RI5)
0x2086	Holding	16	Alternative IO Channel Number 6 Value (RI6)
0x2087	Holding	16	Alternative IO Channel Number 7 Value (RI7)

Tab. 26 Modbus RTU Registers

5 Specification

Parameter		Condition	Value	
Inputs				
	No of Input Channels			4/8
Input - Electrical Characteristics				
	Measurement Method			RTD Two wire measurement
	Resolution			typ. 0.1°C
	RTD Type			Pt100 / Pt1000 DIN IEC 751
	Measurement Error	Pt100		typ. +/- 1.0°C
		Pt1000		typ. +/- 0.5°C
	Constant Measurement Current	Pt100		1mA
		Pt1000		0.5 mA
Input – Timing Characteristic				
	Setup Time		T_{Setup}	5 ms ≤ t ≤ 1 s
Module – Communication				
	Baudrate			1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200 bps
	Parity			NONE, EVEN, ODD
	Stopbits			1 or 2
	Databits			8
	Bus Termination (If enabled)			120 Ω
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			

Tab. 27 RI4/RI8 Specification

Module is specified at environmental temperature of 25°C.

6 Order Information

Order Code	Product
L485-RI4-1000	LucidControl RTD Input Serial Module with 4 Channels for Pt1000 Sensors Measurement Range ±180 °C
L485-RI4-1000C360	LucidControl RTD Input Serial Module with 4 Channels for Pt1000 Sensors Measurement Range 0-360 °C
L485-RI4-100	LucidControl RTD Input Serial Module with 4 Channels for Pt100 Sensors Measurement Range ±180 °C
L485-RI4-100C360	LucidControl RTD Input Serial Module with 4 Channels for Pt100 Sensors Measurement Range 0-360 °C
L485-RI8-1000	LucidControl RTD Input Serial Module with 8 Channels for Pt1000 Sensors Measurement Range ±180 °C
L485-RI8-1000C360	LucidControl RTD Input Serial Module with 8 Channels for Pt1000 Sensors Measurement Range 0-360 °C
L485-RI8-100	LucidControl RTD Input Serial Module with 8 Channels for Pt100 Sensors Measurement Range ±180 °C
L485-RI8-100C360	LucidControl RTD Input Serial Module with 8 Channels for Pt100 Sensors Measurement Range 0-360 °C

Tab. 28 RTD Input Module Order Codes

7 Document Revision

Date	Rev.	
2024/10/08	1.0	Initial Documentation

Tab. 29 Document Revision

deciphe it GmbH
Schäferstr. 5
87600 Kaufbeuren / Germany
www.lucid-control.com