User Manual

LucidControl DI4/DI8

4/8 Channel Digital Input USB Module

1 Introduction

This document describes the functionality of the LucidControl DI4/DI8 USB module with 4/8 digital inputs controllable via Universal Serial Bus.

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

This document explains the topics that are specific to the DI4/DI8 module.

2 Setup and Installation

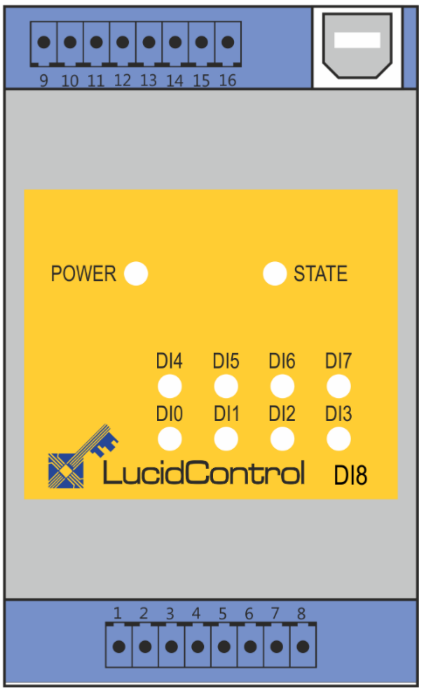

Fig. 1 Digital Input Module

Fig. 1 shows the sketch of the Digital Input DI8 module with 8 digital input channels.

The IO channels are split into the lower and the upper IO connectors.

The lower IO connector (IO1 to IO8) provides terminals for the channels DI0 to DI3.

The upper IO connector (IO9 to IO16) is mounted on DI8 module only. It provides terminals for the channels DI4 to DI7.

⚠ The intended use of the DI4/DI8 module is the acquisition of digital signals. The module must only be used for the intended use.

⚠ For this device it is explicitly stated that no potential (e.g. voltage) of more than 30V must be applied to any connector of the module. The modules must only be used within the specified conditions.

2.1 Configurations

Digital inputs have by nature only two states – they can be either LOW or HIGH. In order to be compatible with different voltage signals, the DI4/DI8 module is available with three threshold levels:

Threshold Level	V_{LowMax}	$V_{HighMin}$
5 V	2.5 V	3.5 V
10 V	6.0 V	8.5V
24 V	16.0 V	21.0 V

Tab. 1 Input Threshold Level

The table above shows the configurations and their input characteristics. All modules have a voltage range below V_{LowMax} which is treated as LOW state, and a voltage range above $V_{HighMin}$ representing a HIGH state.

Input level restrictions

The range between V_{LowMax} and $V_{HighMin}$ is forbidden by means that the state cannot be determined correctly.

Example

For typical automation applications, 24 V inputs are often used. When interfacing 24 V signals it needs to be ensured that in case of a LOW state a voltage equal or below 16 V is applied to the input. Detecting a HIGH state requires a voltage of equal or higher than 21 V. For the range between 16 V and 21 V is used as a hysteresis which means that a once set input is HIGH until the input voltages is lower than 17V.

2.2 Interface and Interconnection

2.2.1 USB Connection

Note

Please consider that the total power of one USB port is limited to 500 mA.

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.

LucidControl DI4/DI8 module is rated with a maximum current of 40 mA.

2.2.2 IO Connection

The LucidControl DI4/DI8 module is equipped with 4 opto-insulated and potential-insulated inputs.

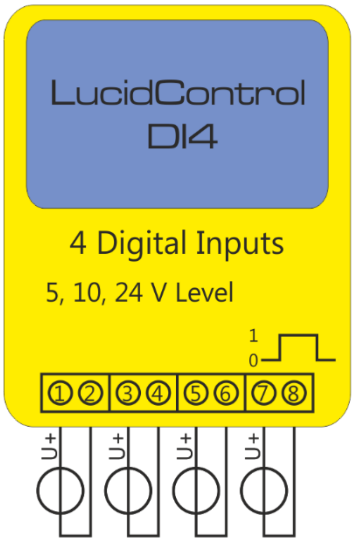

Fig. 2 Digital Input Module Connection

Fig. 2 shows the interconnection of the DI4 module in a typical application.

The input signals are represented by voltage sources applying a voltage within the valid threshold range to the inputs.

The terminals IO1, IO3, IO5, IO7 (and also IO9, IO11, IO13, IO15 for the DI8 module) are the positive voltage inputs.

The terminals IO2, IO4, IO6, IO8 (and IO10, IO12, IO14, IO16 for the DI8 module) are the negative voltage inputs.

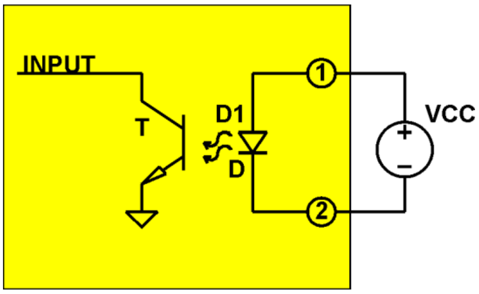

Fig. 3 Digital Input Principle

Fig. 3 illustrates the principle how the digital input works. The signal applied to the terminals 1 and 2 sources an opto-coupler which insulates the electronic of the module from the input signal.

All inputs are floating by means that they have no

common contacts (like ground) and are working independently.

Applying a voltage above $V_{HighMin}$ between the input terminals 1 (resp. 3, 5, 7, …) and 2 (resp. 4, 6, 8, …) sources the LED of the opto-coupler and makes its transistor conductive resulting in a digital HIGH state level. If the voltage is below V_{LowMax} is applied this results in a LOW state level.

⚠ All inputs are protected against overvoltage. Applying a voltage higher than V_{InMax} = 30 V may damage the input.

⚠ All inputs are protected against reverse polarity. Applying a voltage lower than – V_{InMax} = -30V may damage the input.

2.2.3 Isolation of USB Interface (-ISO option)

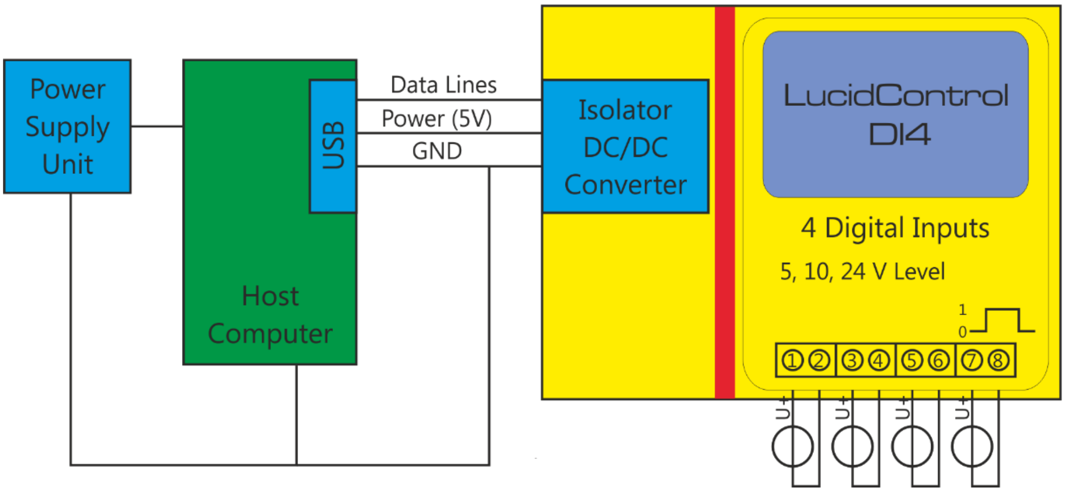

Fig. 4 DI4 Module with isolated USB Interface

DI4 and DI8 modules are optionally available with isolated USB interface (-ISO option). Fig. 4 shows the DI4 module with isolated USB interface.

The isolation consists of a galvanic barrier (red area) that isolates the IO module entirely from the USB data lines and power supply lines. An isolated DC/DC converter separates the power supply.

The main purpose of the isolated LucidControl module is the separation of the IO module from the data processing equipment (e.g. the host computer). Non-Isolated IO modules are conductive connected to the USB port also sharing a common ground line.

Harsh or noisy environments (e.g. with disturbances or long cables) may cause measurement errors or malfunction of the data processing equipment or the IO module caused by ground loops. This can be solved by the isolation of the USB port.

Another aspect is the protection of the data processing equipment from overvoltage. If for example a voltage above the limits of the module is applied to the terminals this can damage the module and the data processing equipment also.

The isolation limits the possible damage to the IO module itself.

USB isolation can be an option if a higher protection level required or if LucidControl IO modules are operating in harsh environments.

⚠ Even when the isolation protects the data processing equipment from overvoltage damage it does not protect from voltages > 50V!

Note:
All inputs of the DI4/DI8 module are isolated by an opto-coupler (Fig. 3).

2.3 Setup of Hard- and Software

Setting up LucidControl hardware is very 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 older than Windows 10 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
   That's all. LucidControl switches the green power LED on and the module is ready for usage.

2.3.1 Windows

As mentioned the installation under Microsoft Windows (older than Windows 10) 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.

2.3.2 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.

2.3.3 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 3 of the general LucidControl User Manual for more information about this helpful tool.

2.3.4 Ready for Take-Off

After 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, 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 -tL -c0,1,2,3 -r [ENTER]
->    CH0:00  CH1:00  CH2:00  CH3:00
```

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 -tL -c0,1,2,3 -r [ENTER]
->    CH0:00  CH1:00  CH2:00  CH3:00
```

3 Module Operation

3.1 Operation Modes

This section explains the operation of the different input modes and gives examples how to configure and use them.

Each of the inputs of the module can work in one of the following modes:

- Reflect Mode
- Rising Edge Mode
- Falling Edge Mode
- Count Mode

In all modes the input values are captured and evaluated after a stable signal has been detected.

Physical input value inversion:
Digital inputs distinguish between physical and logical input state. The physical state is represented by the voltage applied to the input. The logical state is calculated by the input handling and may be identical to the physical state.

In case of input inversion is enabled by setting *inDiInverted* to "on" the logical value is inverted in relation to the physical input. This means that applying a voltage above $V_{HighMin}$ results in a HIGH physical input value but a LOW logical input value.

All input modes support the inversion of physical input value, but in practice it is useful for Reflect Mode only.

3.1.1 Reflect Mode

Reflect Mode gives access to the logical input value directly.

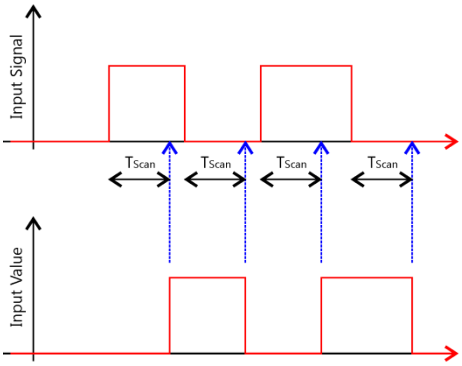

Fig. 5 Reflect Mode Input Processing

Fig. 5 illustrates the processing of the inputs in Reflect Mode.

As soon as the rising edge of the input signal is detected it must remain stable over the interval T_{Scan}. After T_{Scan} has passed the input signal is captured and the input value is set to the corresponding value.

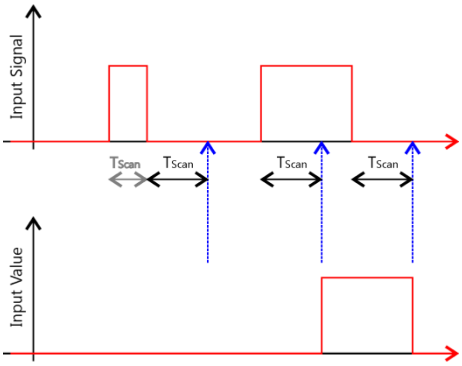

Fig. 6 Reflect Mode Pulse Width

In the case that a pulse is shorter than T_{Scan} the pulse is not valid. This scenario is shown in the first pulse of Fig. 6. While the first rising edge starts the scan timer the falling edge stops it (indicated by the gray T_{Scan} interval) resulting in canceling the pulse detection.

Since the second pulse is longer than T_{Scan} it is evaluated as valid value.

Filtering digital signals and validating their stability can be used to surpress errors and to make the recognition of digital inputs more relyable.

The scan interval T_{Scan} is configurable by changing the parameter *inDiScanTime* which is described in section 3.3.4 .

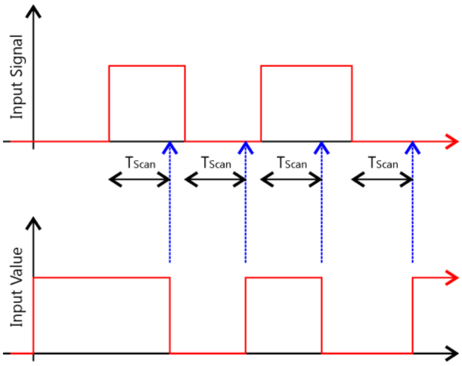

Fig. 7 Inverted Reflect Mode

Fig. 7 illustrates the same input signal as Fig. 5 but with *inDiInverted* set to “on” resulting in logical input value being inverted in relation to the physical input value.

Reflect Mode is simple to use and fits the requirement of many applications, but it is limited in detecting dynamic signals like pulses because the host computer is in charge of detecting them.

Since most operating systems and USB are not able to handle tasks in very short intervals (e.g. shorter than 10 ms) this is not reliable and pulses may be lost.

For such purpose edge detecting and counting modes provide more real time functionality without reading the input value permanently.

LucidIoCtrl Command Line Tool Example
Configure output channel 0 for Reflect Mode

```
      LucidIoCtrl -dCOM4 -c0 -sinDiMode=reflect [ENTER]
```

Read input channel 0

```
      LucidIoCtrl -dCOM4 -c0 -tL -r [ENTER]
->    CH0:00
```

3.1.2 Edge Detection

For applications with very short input signal pulses the module provides two edge detection modes. While in Rising Edge Mode the module is sensitive for input signal low-to-high transitions in Falling Edge Mode it recognizes high-to-low transitions.

Personal computers running Microsoft Windows® or Linux do not have a proper real-time behavior for detecting fast input signals correctly. Because of multitasking it cannot be ensured that a task will run within a specified interval. Assuming that short pulses (e.g. shorter than 1 ms) should be detected the computer has to read the input value at least 1000 times per second which is not realistic. Otherwise it is possible that a pulse is located between two readings of the computer and the pulse will be missed.

These two modes are useful in order to allow real-time detection of input signal transitions without the host computer being affected. Fig. 8 shows that a recognized pulse and the input value representing this pending event indicating that at least 1 transition occurred between two readings of the computer.

By changing parameter *diInScanTime* the length of the shortest detectable pulse can be adopted by changing scan interval time T_{Scan}.

Example
The input should be able to detect an input signal pulse of 100 µs length.

One would say it is sufficient to read the input in reflect mode every 50 µs (with safety time) in order to detect the transition by the computer. But probably the operating system of the computer would violate the timing restrictions and pulses are lost.

By using the input configured in one of the two edge detection modes this can be solved easily by setting T_{Scan} = 90 µs (including safety). A detected edge will be pending until it is read by the host computer. Assuming the host computer reads now every second it can check if one ore more transition(s) occurred within the last second. The 1 second interval can be realized by the host computer without problems.

Fehler! Kein gültiger Dateiname.

Configure output channel 0 for Rising Edge detection mode

```
      LucidIoCtrl -dCOM4 -c0 -sinDiMode=risingEdge [ENTER]
```

Set T_{Scan} to 90 µs

```
      LucidIoCtrl -dCOM4 -c0 -sinDiScanTime=90000 [ENTER]
```

Read input channel 0

```
      LucidIoCtrl -dCOM4 -c0 -tL -r [ENTER]
->    CH0:01
```

3.1.2.1 Rising Edge Mode

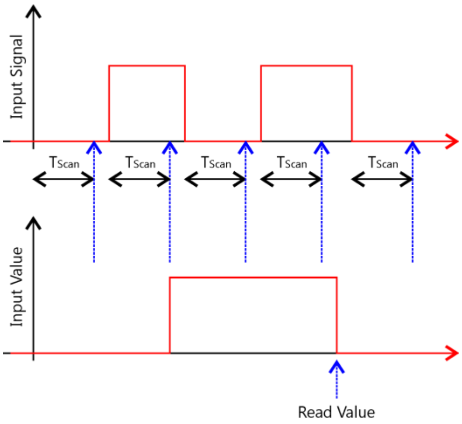

Fig. 8 Rising Edge Mode

Fig. 8 shows a digital input signal and the corresponding input value in Rising Edge Mode. After the high signal was detected as valid it is noticeable that the set input value is pending until the value is read by the host computer.

Rising Edge Mode allows detecting low-to-high transitions of the input signal without the host computer being involved.

3.1.2.2 Falling Edge Mode

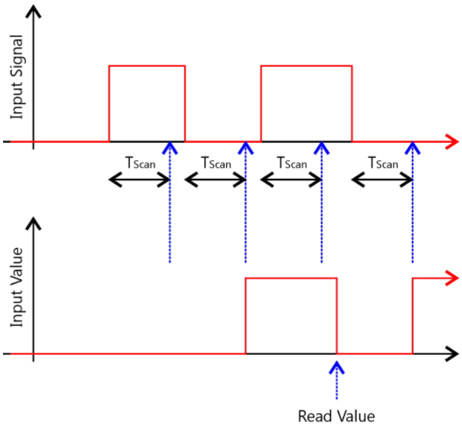

Fig. 9 Falling Edge Mode

For the Falling Edge Mode applies the same functionality as for Rising Edge mode (see section 3.1.2.1) with the one exception that not a rising but a falling transition sets the input value.

While in Rising Edge Mode it is obvious that a low-to-high transition sets the input value to “1” it must be mentioned that in Falling Edge Mode a transition from high-to-low sets the input value to “1”.

This shows that the input value in edge detection modes does not refer the input signal rather than it must be seen as a pending event. When the event (which can be either a low-to-high or high-to-low input signal transition) arises the input value is set to “1”.

3.1.3 Count Mode

Compared to other modes the Count Mode is different by means that it does not generate an input value, but calculates the number of valid pulses accumulated within a time specified by T_{Count}. This mode allows to count pulses and to calculate frequencies of periodical input signals.

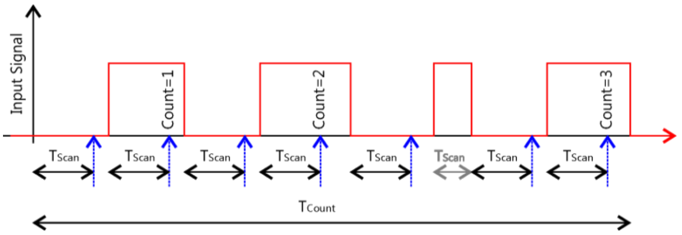

Fig. 10 Count Mode

Fig. 10 illustrates a typical periodical input signal. In Count Mode all detected pulses are accumulated until the counting interval T_{Count} finishes.

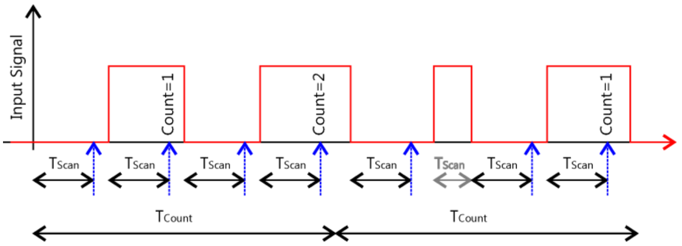

Fig. 11 Count Mode Short Count Interval

Fig. 11 shows the identical input signal, but with a shorter counting interval.

After T_{Count} has passed the valid count value is accessible and the module starts with a new measurement cycle.

Like in the other modes an input signal is only regarded as valid when it was stable for at least the scan time T_{Scan}. In Count mode only stable pulses are accumulated (see the gray interrupted T_{Scan} intervals).

Normally, T_{Scan} timer is started on the rising edge of the input signal as it is shown in the two pictures above. In the case that the parameter *inDiInverted* is set to "on" T_{Scan} timer is started on the falling edge of the input signal.

Accessing the input value in Count Mode returns the number of pulses of the last finished scan interval cycle.

Fehler! Kein gültiger Dateiname.

Configure input channel 0 for Count Mode

```
      LucidIoCtrl –dCOM4 –c0 –sinDiMode=count [ENTER]
```

Set Scan Time T_{Scan} to 1ms

```
      LucidIoCtrl –dCOM4 –c0 –sinDiScanTime=1000 [ENTER]
```

Set Count Time TCount to 1s

```
      LucidIoCtrl –dCOM4 –c0 –sinDiCountTime=1000000 [ENTER]
```

Read number of pulses

```
      LucidIoCtrl –dCOM4 –c0 –tN -r [ENTER]
->    CH0:0x0064 (100)
```

In this example 100 pulses of at least 1 millisecond length occured within a measurement interval of 1 second.

Since the input value (number of pulses) is updated after count time has passed it takes 1 second to update the value. Decreasing count time results in a faster update of the input value.

Note

While in other modes the value type "L" is specified, in Count Mode the value type must be set to "N". LucidIoCtrl returns the value in hexadecimal and decimal format.

3.1.3.1 Count Mode Options

In standard Count Mode the counter value is valid after count interval T_{Count} has passed and it overwrites previously detected pulses.

In order to count pulses of non-periodical input signals the module supports two additional options *inDiAddCounts* and *inDiResetCountsOnRead*.

In the case *inDiAddCounter* is set to "off" the number of currently counted pulses is used for update after T_{Cycle} has finished. Previously counted pulses are overwritten and get lost. In the case *inDiAddCounter* is "on" the number of currently counted pulses is added to the counter containing the pulses of previous measurements.

In the case of *inDiResetCounterOnRead* set to “off” reading of the counter value does not affect the counter value itself. In the case of *inDiResetCounterOnRead* is “on” the counter value is reset after reading it.

In order to avoid overflows *inDiAddCounter*=”on” should be combined with *inDiResetCounterOnRead* parameter.

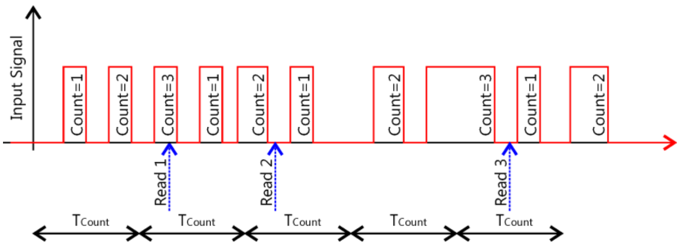

Fig. 12 Add Counter Mode

Fig. 12 shows a typical input signal with 10 pulses in total which is not periodic. This diagram shows the behavior and the 3 read values for both, *inDiAddCounter* and *inDiResetCounterOnRead* switched “on” which is the most useful mode.

The number of counts in the diagram refers to the internal counter which may differ from the read number of counts by means that the read value is updated only after T_{Count} has finished.

On Read 1 the read value is 2 since the internal counter value after T_{Count} has finished was 2. The counter value is reset on reading and 1 pulse is carried over to the next T_{Count} inverval.

On Read 2 the read value is 3 because of the current pulse counter of 2 plus the carried over counter value of 1 from the previous count interval. The counter value is reset on read.

At the reading 3 the read value is 3. And the counter value is reset again

This example explains that there are no pulses lost even if there is a longer time in between of subsequent readings.

Count mode with *inDiAddCounter* = “on”

In the case that *inDiAddCounter* is “on” and *inDiResetCounterOnRead* is “off” the behavior changes that the pulses are accumulated but the counter is not reset on reading the value. This causes the counter value is updated when count interval has finished and the current counter is added to the last counter value. This may result in an overflow when the counter value rolls over its maximum value of 65535.

The parameter *inDiResetCounterOnRead* has only an effect together with *inDiAddCounter* set to “on”.

Tab. 2 explains the different count value results of the previous example for all options.

Mode	Value Read 1	Value Read 2	Value Read 3
inDiAddCounter = "on" *inDiResetCounterOnRead* = "on"	2	3	3
inDiAddCounter = "on" *inDiResetCounterOnRead* = "off"	2	5	8
inDiAddCounter = "off" *inDiResetCounterOnRead* = "off" (Standard Count mode)	2	3	2

Tab. 2 Add Counter and Reset Counter on Read Results

3.2 Commands

After an input was set up correctly and configured it is possible to read the input value by using the commands GetIo for a single value or GetIoGroup in order to read a group of input values of the same type.

Accessing inputs and outputs is a very common task which is mostly identical for all Lucid Control modules. Please refer to the section 3.2.1.1, 3.2.1.2 and 4.3 of the general LucidControl manual for comprehensive information covering reading and writing of inputs and outputs in general.

The following sections describe in detail the commands which are supported by the DI4/DI8 module.

3.2.1 GetIo

This command reads the logic value of an input.

Tab. 3 shows the input values depending on the configured input mode.

Mode	Value
Reflect	Contains the logic input value as “00” or “01”
Edge Detection	Represents pending edge detected event with values “00” or “01”
Count	Represents the number of valid pulses within count interval time within the range of 0 … 65535

Tab. 3 GetIo Values

<table>
<tr><th>Command</th><td>GetIo</td><th>Access</th><td>Read</td></tr>
<tr><th>Opcode</th><td>0x46</td><td></td><td></td></tr>
<tr><th colspan="4">LucidIoControl Command Line Tool</th></tr>
<tr><th>Call (-tL)</th><td colspan="3"><code>LucidIoCtrl -d[COMx] -c[Channel] -tL -r</code></td></tr>
<tr><th>Return</th><td colspan="3">CHn:ll<table><tr><td>n</td><td>Input Channel</td></tr><tr><td>ll</td><td>Input Digital Value</td></tr></table></td></tr>
<tr><th>Call (-tN)</th><td colspan="3">LucidIoCtrll –d[COMx] –c[Channel] –tN –r</td></tr>
<tr><th>Return</th><td colspan="3">CHn:vv<table><tr><td>n</td><td>Input Channel</td></tr><tr><td>vv</td><td>Input Count Value</td></tr></table></td></tr>
</table>

<u>Note</u>

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

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

Read input channel 0. The module is operating in Reflect or Edge Detection Mode

```
      LucidIoCtrl -dCOM4 -c0 -tL -r [ENTER]
->    CH0:01
```

Read input channel 0 in the case that the module is operating in Count Mode

```
      LucidIoCtrl -dCOM4 -c0 -tN -r [ENTER]
->    CH0:0x0064 (100)
```

<u>Request Frame</u>

OPC	P1	P2	LEN
0x46	Channel	Value Type	0

<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>Channel</td><td>Number of input or output channel (Range: 0 ~ 7)</td></tr>
<tr><td>Value Type</td><td>Value Type
Supported Value Types
<table>
<tr><th>Value Type</th><th>Value Range</th><th>Size</th></tr>
<tr><td>Digital Logic Value (0x00)</td><td>0x00 or 0x01</td><td>1 Byte</td></tr>
<tr><td>Digital Counter Value (0x0A)</td><td>0 ~ 65,535</td><td>2 Bytes</td></tr>
</table></td></tr>
</table>

Tab. 4 GetIo Request

Response Frame:

Status	LEN	Data Field
Status	Size	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.

3.2.2 GetIoGroup

This command reads the logic input values of a group of input of the same Value Type. See also section 3.2.1.

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

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

<u>Channels:</u>
Comma separated list of channels e.g. –c0,1,3</td></tr>
<tr><th>Return</th><td colspan="3">List of values sorted from lower to higher channels
CHn:vv<table><tr><td>n</td><td>Input Channel</td></tr><tr><td>vv</td><td>Input Counter Value</td></tr></table></td></tr>
</table>

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

Read input values of input channel 0, 1 and 3:

```
LucidIoCtrl -dCOM4 -c0,1,3 -tL -r [ENTER]
CH0:00  CH1:01  CH3:01
```

<u>Request Frame</u>

OPC	P1	P2	LEN
0x48	Channel 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>
<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 are bitwise or 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 = 0x01 OR 0x08 = 0x09
1 and 2 Value = 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>Response Len</th></tr>
<tr><td>Digital Logic Value (0x00)</td><td>0x00 oder 0x01</td><td>1 Byte</td></tr>
<tr><td>Digital Counter Value (0x0A)</td><td>0 ~ 65,535</td><td>2 Bytes</td></tr>
</table></td></tr>
</table>

Tab. 5 GetIoGroup Request

Response Frame:

Status	LEN	Data Field
Status	Length	Value

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.

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 inputs 0, 1, 3, and 7

Opcode	P1	P1A	P2	Length
0x48	0x8B	0x01	0x00	0x00

Response Frame:

For input 0 = "0", input 1 = "1", input 3 = "1" and input 7 = "0"

Values in Data Field are in ascending order Channel 0, Channel 1, Channel3, and Channel7.

Header Field		Data Field			
Status	LEN	Value Channel 0	Value Channel 1	Value Channel 3	Value Channel 7
0x00	0x04	0x00	0x01	0x01	0x00

3.3 Parameters

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

The Parameters are accessible by the commands SetParam and GetParam. The sections 4.3.5 and 4.3.6 of the LucidControl User Manual describe them in detail.

The relevance of some parameters of the DI4/DI8 module may depend on the active mode described in section 3.1.

3.3.1 inDiValue

This IO Configuration Parameter reflects the value of the input.

In Reflect Mode and Edge Detection Mode the parameter contains the input value as it can also be read by GetIo or GetIoGroup command.

If the input is configured in Count Mode this parameter is "0".

Parameter	inDiValue	**Access**	Read
Address	0x1000		
Values	Input Value		
Default Value	0x00	**Parameter Type**	1 Byte unsigned
LucidIoControl Command Line Tool			
Parameter Name	inDiValue	**Parameter Values**	0x00 or 0x01
Call (Get)	`LucidIoCtrl –d[COMx] –c[Channel] –ginDiValue`		

LucidIoCtrl Command Line Tool Example

Read value of input channel 0:

```
LucidIoCtrl –dCOM4 –c0 –ginDiValue [ENTER]
```

```
->      inDiValue=0
```

Note:
For normal operation it is recommended to use the function GetIo (3.2.1) in order to read the input value.

3.3.2 inDiMode

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

<table>
<tr><td>Parameter</td><td>inDiMode</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>reflect</td></tr>
<tr><td>0x10</td><td>risingEdge</td></tr>
<tr><td>0x11</td><td>fallingEdge</td></tr>
<tr><td>0x20</td><td>count</td></tr>
</table></td></tr>
<tr><td>Default Value</td><td>0x00</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>inDiMode</td><td>Parameter Values</td><td>0x00 or 0x01</td></tr>
<tr><td>Call (Set)</td><td colspan="3"><code>LucidIoCtrl -d[COMx] -c[Channel] -sinDiMode=[Value] {-p} {--default}</code></td></tr>
<tr><td>Call (Get)</td><td colspan="3"><code>LucidIoCtrl -d[COMx] -c[Channel] -ginDiMode</code></td></tr>
</table>

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

```
        LucidIoCtrl -dCOM4 -c0 -sinDiMode=count -p [ENTER]
```

Read the operation mode of input channel 0

```
        LucidIoCtrl -dCOM4 -c0 -ginDiMode [ENTER]
->      inDiMode=count
```

3.3.3 Bit Parameter inDiFlags

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

<table>
<tr><td>Parameter</td><td>inDiFlags</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>inDiAddCounter</td><td>Bit 0</td></tr>
<tr><td>inDiResetCounterOnRead</td><td>Bit 1</td></tr>
<tr><td>inDiInverted</td><td>Bit 2</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 *inDiFlags* cannot be accessed directly by using the Command Line Tool. The Bit Parameters can be used instead.

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

3.3.3.1 inDiInverted

This Bit Parameter configures the inversion of the physical input value.

See input modes descriptions in section 3.1 for more information.

<table>
<tr><td>Parameter</td><td>inDiFlags</td><td>Access</td><td>Read / Write</td></tr>
<tr><td>Address</td><td>0x1101</td><td colspan="2">Bit Parameter in inDiFlags</td></tr>
<tr><td>Values</td><td colspan="3">
<table>
<tr><th>Bit Parameter</th><th>Bit Postion</th></tr>
<tr><td>inDiInverted</td><td>Bit 2</td></tr>
</table></td></tr>
<tr><td>Default Value</td><td>Off</td><td>Parameter Type</td><td>1 Bit</td></tr>
<tr><td colspan="4">LucidIoControl Command Line Tool</td></tr>
<tr><td>Parameter Name</td><td>inDiInverted</td><td>Parameter Values</td><td>on / off</td></tr>
<tr><td>Call (Set)</td><td colspan="3"><code>LucidIoCtrl –d[COMx] –c[Channel] –sinDiInverted=[Value] {-p} {--default}</code></td></tr>
<tr><td>Call (Get)</td><td colspan="3"><code>LucidIoCtrl –d[COMx] –c[Channel] –ginDiInverted</code></td></tr>
</table>

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

```
LucidIoCtrl –dCOM4 –c0 –sinDiInverted=on –p [ENTER]
```

Read inversion of physical input channel 0

```
      LucidIoCtrl -dCOM4 -c0 -ginDiInverted [ENTER]
->    inDiInverted=on
```

3.3.3.2 inDiAddCounter

This Bit Parameter controls how the counter value is updated after count interval time T_{Count} has finished. It is relevant in Count Mode only. (see section 3.1.3)

In the case *inDiAddCounter* is “off” the number of currently counted pulses is used for update after count interval time T_{Count} has finished. Previously counted pulses are overwritten.

In the case *inDiAddCounter* is “on” the number of currently counted pulses is added to the counter value of previous measurements.

In order to avoid overflows *inDiAddCounter*=”on” should be combined with *inDiResetCounterOnRead* parameter.

<table>
<tr><td>Parameter</td><td>inDiFlags</td><td>Access</td><td>Read / Write</td></tr>
<tr><td>Address</td><td>0x1101</td><td colspan="2">Shared bit in Parameter <i>inDiFlags</i></td></tr>
<tr><td>Values</td><td colspan="3"><table><tr><th>Bit Parameter</th><th>Bit Postion</th></tr><tr><td>inDiAddCounter</td><td>Bit 0</td></tr></table></td></tr>
<tr><td>Default Value</td><td>Off</td><td>Parameter Type</td><td>1 Bit</td></tr>
<tr><td colspan="4">LucidIoControl Command Line Tool</td></tr>
<tr><td>Parameter Name</td><td>inDiAddCounter</td><td>Parameter Values</td><td>on / off</td></tr>
<tr><td>Call (Set)</td><td colspan="3"><code>LucidIoCtrl -d[COMx] -c[Channel] -sinDiAddCounter=[Value] {-p} {--default}</code></td></tr>
<tr><td>Call (Get)</td><td colspan="3"><code>LucidIoCtrl -d[COMx] -c[Channel] -ginDiAddCounter</code></td></tr>
</table>

LucidIoCtrl Command Line Tool Example

Enable counter add on update for input channel 0 and make the setting persistent.

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

Read counter add setting of input channel 0

```
      LucidIoCtrl -dCOM4 -c0 -ginDiAddCounter [ENTER]
      inDiAddCounter=on
```

3.3.3.3 inDiResetCounterOnRead

This Bit Parameter controls how to update the counter value after reading it. It is relevant in Count mode only. (see section 3.1.3)

In the case of *inDiResetCounterOnRead* is “off” reading of the counter value does not affect the counter value itself.

In the case of *inDiResetCounterOnRead* is “on” the counter value is reset after reading it.

<table>
<tr><td>Parameter</td><td>inDiFlags</td><td>Access</td><td>Read / Write</td></tr>
<tr><td>Address</td><td>0x1101</td><td colspan="2">Shared bit in Parameter <i>inDiFlags</i></td></tr>
<tr><td>Values</td><td colspan="3"><table><tr><th>Bit Parameter</th><th>Bit Postion</th></tr><tr><td>inDiResetCounterOnRead</td><td>Bit 1</td></tr></table></td></tr>
<tr><td>Default Value</td><td>Off</td><td>Parameter Type</td><td>1 Bit</td></tr>
<tr><td colspan="4">LucidIoControl Command Line Tool</td></tr>
<tr><td>Parameter Name</td><td>inDiResetCounterOnRead</td><td>Parameter Values</td><td>on / off</td></tr>
<tr><td>Call (Set)</td><td colspan="3"><code>LucidIoCtrl -d[COMx] -c[Channel]</code>
<code>-sinDiResetCounterOnRead=[Value] {-p} {--default}</code></td></tr>
<tr><td>Call (Get)</td><td colspan="3"><code>LucidIoCtrl -d[COMx] -c[Channel] -ginDiResetCounterOnRead</code></td></tr>
</table>

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

Enable counter reset on read for input channel 0 and make the setting persistent.

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

Read counter reset setting of input channel 0

```
      LucidIoCtrl -dCOM4 -c0 -ginDiResetCounterOnRead [ENTER]
->    inDiResetCounterOnRead=on
```

3.3.4 inDiScanTime

This IO Configuration Parameter specifies the scan time T_{Scan} of the digital input.

T_{Scan} defines the scan time interval within the input signal must be stable in order to detect it as valid.

<table>
<tr><td>Parameter</td><td>inDiScanTime</td><td>Access</td><td>Read / Write</td></tr>
<tr><td>Address</td><td>0x1111</td><td colspan="2"></td></tr>
<tr><td>Values</td><td colspan="3">T_{Scan} in µs (micro seconds)
80 µs ≤ T_{Scan} ≤ 1 s</td></tr>
<tr><td>Default Value</td><td>50,000 (50 ms)</td><td>Parameter Type</td><td>4 Bytes unsigned</td></tr>
<tr><td colspan="4">LucidIoControl Command Line Tool</td></tr>
<tr><td>Parameter Name</td><td>inDiScanTime</td><td>Parameter Values</td><td>Time [µs]</td></tr>
<tr><td>Call (Set)</td><td colspan="3"><code>LucidIoCtrl -d[COMx] -c[Channel] -sinDiScanTime=[Time]</code>
<code>{-p} {--default}</code></td></tr>
<tr><td>Call (Get)</td><td colspan="3"><code>LucidIoCtrl -d[COMx] -c[Channel] -ginDiScanTime</code></td></tr>
</table>

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

Set T_{Scan} of input channel 0 to 1.5 s and make the setting persistent.

```
      LucidIoCtrl -dCOM4 -c0 -sinDiScanTime=1500000 -p [ENTER]
```

Read T_{Scan} parameter of input channel 0

```
      LucidIoCtrl -dCOM4 -c0 -ginDiScanTime [ENTER]
->    inDiScanTime=1500000
```

3.3.5 inDiCountTime

This IO Configuration Parameter specifies the count time T_{Count} of the digital input in Count Mode.

The count interval specifies the time the pulses of the input signal are accumulated.

Parameter	inDiCountTime	**Access**	Read / Write
Address	0x1112		
Values	T_{Count} in µs (micro seconds) 1 ms ≤ T_{Scan} ≤ 1 h		
Default Value	5,000,000 (5 s)	**Parameter Type**	4 Bytes unsigned
LucidIoControl Command Line Tool			
Parameter Name	inDiCountTime	**Parameter Values**	Time [µs]
Call (Set)	`LucidIoCtrl -d[COMx] -c[Channel] -sinDiCountTime=[Time] {-p} {--default}`		
Call (Get)	`LucidIoCtrl -d[COMx] -c[Channel] -ginDiCountTime`		

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

Set T_{Count} of input channel 0 to 10 s and make the setting persistent.

```
LucidIoCtrl -dCOM4 -c0 -sinDiScanTime=10000000 -p [ENTER]
```

Read T_{Count} parameter of input channel 0

```
LucidIoCtrl -dCOM4 -c0 -ginDiCountTime [ENTER]
->inDiCountTime=10000000
```

4 Specification

Parameter		Condition	Value	
Inputs				
	No of Input Channels			4/8
Inputs - Electrical Characteristics				
	Input Signal Maximum Low Level	5V	$U_{05MaxLow}$	2.5 V
		10V	$U_{10MaxLow}$	6.0 V
		24V	$U_{24MaxLow}$	16.0 V
	Input Signal Minimum High Level	5V	$U_{05MinHigh}$	3.5 V
		10V	$U_{10MinHigh}$	8.5 V
		24V	$U_{24MinHigh}$	21.0 V
	Maximum Input Voltage		U_{Max}	30 V
	Maximum Reverse Input Voltage		U_{RMax}	-30 V
	Input Impedance		R_{In}	> 1 kΩ
Inputs – Timing Characteristic				
	T_{Scan}			$t_{Min} < T_{Scan} < 1$ s
	T_{Count}			1 ms $< T_{Count} < 1$ h
	Minimum pulse length		t_{Min}	100 µs
	Maximum Frequency of Input Signal in Count Mode	DC 50 %		2,000 Hz
		20% > DC < 80%		500 Hz
Module – Host Interface				
	USB			2.0 Full Speed CDC Profil
	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				
	Dimensions 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 Status Indicator • Communication Indicator • Indicator for Input State (Enabled / Disabled)			

5 Order Information and Accessories

Digital Input Devices

General format of order code

LCTR-DIn-I-voltage(-ISO)

Order Code	**Product**
LCTR-DI4-I-5	LucidControl Digital Input USB Module with 4 insulated Channels for 5 Volt Signals
LCTR-DI4-I-10	LucidControl Digital Input USB Module with 4 insulated Channels for 10 Volt Signals
LCTR-DI4-I-24	LucidControl Digital Input USB Module with 4 insulated Channels for 24 Volt Signals
LCTR-DI8-I-5	LucidControl Digital Input USB Module with 8 insulated Channels for 5 Volt Signals
LCTR-DI8-I-10	LucidControl Digital Input USB Module with 8 insulated Channels for 10 Volt Signals
LCTR-DI8-I-24	LucidControl Digital Input USB Module with 8 insulated Channels for 24 Volt Signals

Order Code (ISO)	**Product**
-ISO	With galvanic isolation of USB Interface

The following accessories are available:

Order Code	**Product**
64.200.0005	Plug-In Terminal 8-way 1,5 mm^2 wire

6 Document Revision

Date	Rev.	
2018/08/04	2.0	• Added documentation of DI8 module • Added documentation of USB Isolation

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