
User Manual

# LucidControl DI4/DI8

## 4/8 Channel Digital Input USB Module

# Table of Content

1 Introduction ... 3
2 Setup and Installation ... 3
2.1 Safety Information ... 3
2.2 USB Connection ... 4
2.3 Software ... 4
2.3.1 Microsoft Windows ... 4
2.3.2 Linux ... 4
2.3.3 LucidIoCtrl Command Line Tool ... 5
2.3.4 First Steps ... 5
2.4 IO Configurations ... 6
3 Module Operation ... 9
3.1 Real-time Considerations ... 9
3.1.1 Input Signal Value Inversion ... 9
3.1.2 Filter ... 9
3.2 Operation Modes ... 9
3.2.1 Reflect Mode ... 10
3.2.2 Edge Detection ... 11
3.2.2.1 Rising Edge Detection Mode ... 12
3.2.2.2 Falling Edge Detection Mode ... 12
3.2.3 Count Mode ... 13
3.2.3.1 Count Mode Options ... 14
3.3 Commands ... 16
3.3.1 GetIo ... 16
3.3.2 GetIoGroup ... 18
3.4 Parameters ... 21
3.4.1 inDiValue ... 21
3.4.2 inDiMode ... 21
3.4.3 Bit Parameter inDiFlags ... 23
3.4.3.1 inDiInverted ... 23
3.4.3.2 inDiAddCounter ... 24
3.4.3.3 inDiResetCounterOnRead ... 24
3.4.4 inDiScanTime ... 25
3.4.5 inDiCountTime ... 25
4 Specification ... 27
5 Order Information ... 28
6 Document Revision ... 29

# 1 Introduction

This document describes the functionality of the LucidControl DI4/DI8 USB IO module with 4/8 digital input channels controllable by the Universal Serial Bus.

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

# 2 Setup and Installation

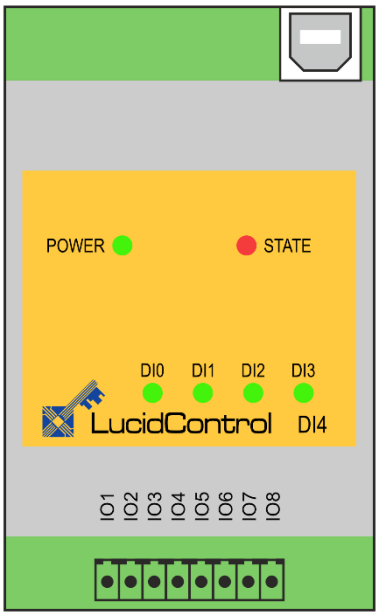


**Fig. 1 LucidControl DI4 Module**

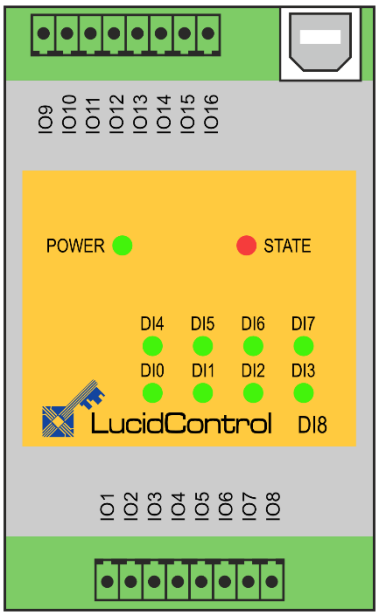


**Fig. 2 LucidControl DI8 Module**

Fig. 1 and Fig. 2 show drawings of the DI4 and DI8 digital input modules with 4/8 digital input channels (DI0 – DI3 and DI4 to DI7).

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 DI8 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

Never apply voltages higher than 30V to any IO terminal. This would damage the device.

All contacts of the modules are protected against ESD but not necessarily against overload, which is especially relevant for output modules.

## 2.2 USB Connection

LucidControl USB modules are connected to the computer by using a USB 2.0 cable which must not extend a length of 5 m. They are “bus powered” what means that the host computer supplies the module with power.

LucidControl DI4/DI8 module is rated with a maximum current of 100 mA.

## 2.3 Software

LucidControl modules are plug and play and a manual driver installation is not necessary.

First connection:

1. Ensure that no signal is applied to the IO terminals
2. Connect LucidControl via USB with the computer
3. Applies for Microsoft Windows before Windows 10 only: The system requests an installation file. This is not a driver but an information file (INF) only. It can be downloaded from our website https://lucid-control.com/downloads
4. LucidControl switches the green power LED on indicating that the module can be used.

### 2.3.1 Microsoft Windows

After the installation has finished, the Windows Device Manager lists a new serial port (COM) entry. The module can be accessed by using this port.

If more than one module is connected to a computer, the operating system ensures that the same serial port number is assigned to the module(s) after restart.

### 2.3.2 Linux

The module is immediately installed as /dev/ttyACM**n** device where n is a number referring to the index of the device.

Note:
If more than one module is connected to a computer, Linux does by default not ensure that a module is permanently linked to the same /dev/ttyACM**n** device.

### 2.3.3 LucidIoCtrl Command Line Tool

The LucidIoCtrl command line tool gives full access to all LucidControl modules. It is available 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 *LucidControl User Manual* for more information.

### 2.3.4 First Steps

After the module was successfully installed, 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.

Windows Examples:
Reading the values of the first 4 input channels from a device connected to COM1

```
      LucidIoCtrl -dCOM1 -tL -c0,1,2,3 -r
->    CH0:00  CH1:00  CH2:00  CH3:00
```

Windows requires a different argument for comport numbers 10 and above.

```
      LucidIoCtrl -d\\.\COM10 -tL -c0,1,2,3 -r
->    CH0:00  CH1:00  CH2:00  CH3:00
```

Linux Examples:
Reading the values of the first 4 input channels from a device connected to /dev/ttyACM0.

```
      LucidIoCtrl -d/dev/ttyACM0 -tL -c0,1,2,3 -r
->    CH0:00  CH1:00  CH2:00  CH3:00
```

## 2.4 IO Configurations

The DI4/DI8 module is available in different configurations, which are explained in this section.

| Function Class | Value | Channels |
|---|---|---|
| DI4 | 0x0000 | 4 |
| DI8 | 0x0010 | 8 |

**Tab. 1 Digital Input Function Classes**

| Function Class Type | Value | Threshold Level |
|---|---|---|
| 5 | 0x1000 | 5V |
| 10 | 0x1001 | 10V |
| 24 | 0x1005 | 24V |

**Tab. 2 Digital Input Function Class Types**

Tab. 1 and Tab. 2 list the Function Classes and their types.

The digital inputs are available with different threshold levels:

| Function Class Type | $V_{LowMax}$ | $V_{HighMin}$ | $V_{HighMax}$ |
|---|---|---|---|
| 5 | 2.5 V | 3.5 V | 7.5V |
| 10 | 6.0 V | 8.5V | 15V |
| 24 | 16.0 V | 21.0 V | 30V |

**Tab. 3 Digital Input Threshold Levels**

Tab. 3 shows the characteristic voltages for the different threshold levels.
Voltages below $V_{LowMax}$ result in a LOW value, voltage higher than $V_{HighMin}$ in a HIGH value. Because of the input hysteresis, voltages between $V_{LowMax}$ and $V_{HighMin}$ do not change the logic value.

In order to prevent excessive power dissipation of the digital inputs, the maximum applied voltage must not exceed $V_{HighMax}$.

A high input state is indicated by a green status LED.

Example
When interfacing a 24 V signal, the applied voltage of a HIGH state must be higher than 21.0 V. The voltage of a low state needs to be lower than 16.0 V. For the voltages between, the last detected stable value remains.

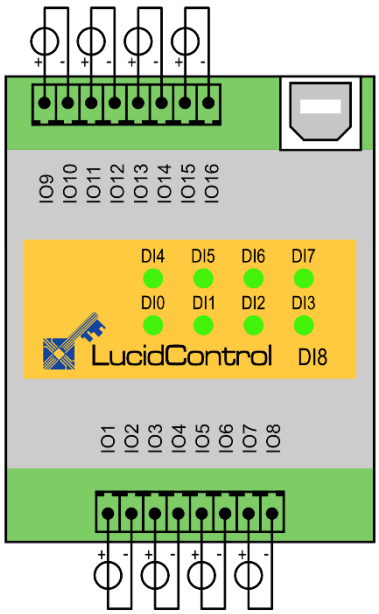


**Fig. 3 DI8 IO Connection**

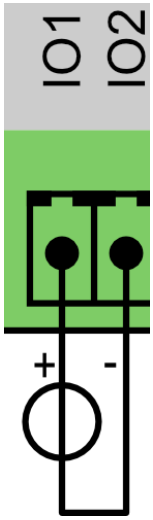


**Fig. 4 DI8 Signal**

Fig. 3 shows the connection of the DI8 module.

Voltage sources are connected to all IO terminals IO1 to IO16.

Fig. 4 shows in detail the voltage source connected to IO terminals IO1 and IO2 (DI0).

Tab. 4 lists the IO terminals of the DI4/DI8 module and their IO channel numbers.

| IO Terminal | Signal | IO Channel Number |
|---|---|---|
| 1 | DI0 + | 0 |
| 2 | DI0 - | |
| 3 | DI1 + | 1 |
| 4 | DI1 - | |
| 5 | DI2 + | 2 |
| 6 | DI2 - | |
| 7 | DI3 + | 3 |
| 8 | DI3 - | |
| 9 | DI4 + | 4 |
| 10 | DI4 - | |
| 11 | DI5 + | 5 |
| 12 | DI5 - | |
| 13 | DI6 + | 6 |
| 14 | DI6 - | |
| 15 | DI7 + | 7 |
| 16 | DI7 - | |

**Tab. 4 DI4 / DI8 IO Terminal Connector**

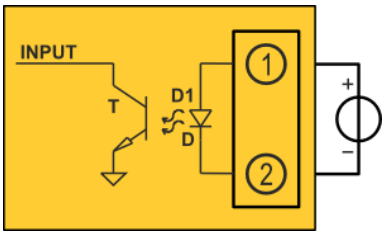


**Fig. 5 Digital Input Principle**

Fig. 5 illustrates the principle of the digital input channels. The input signal applied to the IO terminals 1 and 2 (DI0) powers the LED of the opto-coupler, which insulates the input signal and the acquisition hardware.

All inputs are floating and do not have any common connections e.g. to ground signal.

# 3 Module Operation

## 3.1 Real-time Considerations

Operating systems for personal computers are not made for deterministic real-time operation. Because of multitasking it cannot be ensured that a task will continue to run within a specified interval.

USB is also no real-time bus and limits the timing.

Assuming that short pulses (e.g. shorter than 10 ms) should be detected, the computer has to read the input value more than 100 times per second what is not realistic. A pulse located between two readings and the pulse would be lost.

Edge detection and count modes improve the real-time characteristics.

### 3.1.1 Input Signal Value Inversion

Digital input channels have an input signal value and a logical input value. The input signal value is represented by the voltage applied to the input channel. The logical value is evaluated by the input processing.

In case of inversion is disabled, the input signal values and logical values are identical.

In case of input inversion is enabled by configuration parameter *inDiInverted* (→ 3.4.3.1) set to “on”, the logical value is the inverted input signal value. This means that a voltage higher than $V_{HighMin}$ results in a HIGH input signal value but a LOW logical input value.

All input modes support input signal value inversion.

### 3.1.2 Filter

The digital input signal is filtered and only stable high or low signals are evaluated. This function allows the input channel to debounce the signal.

Signals are only considered valid if they are stable during the time $T_{Scan}$ (→ 3.4.4 → Fig. 7).

## 3.2 Operation Modes

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

Digital input channels can operate in one of the 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.

### 3.2.1 Reflect Mode

The reflect mode acquires the logical input value.

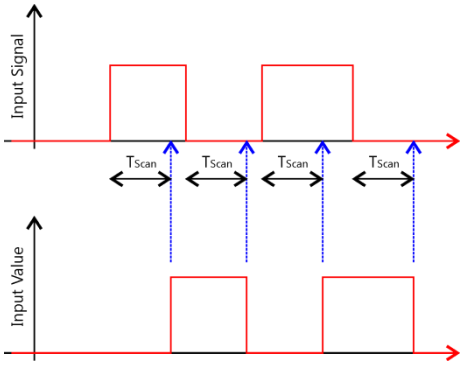


**Fig. 6 Digital Input Channel Processing in Reflect Mode**

Fig. 6 illustrates the processing of the digital input channels in reflect mode.

After the rising edge of the input signal was detected and it remained stable for the interval $T_{Scan}$ the input value is updated.

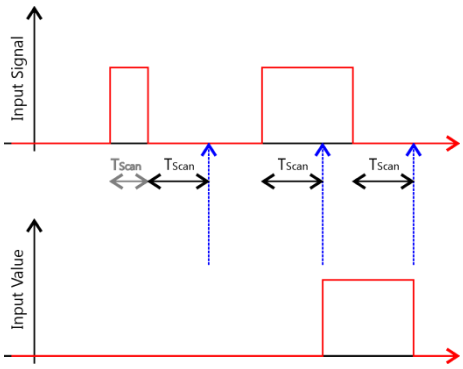


**Fig. 7 Reflect Mode Pulse Width**

In the case that a pulse of the input signal is shorter than $T_{Scan}$, it is ignored and the input value does not change. Fig. 7 shows this at the first pulse. The rising edge of the input signal starts the scan timer and the falling edge stops it (indicated by the gray $T_{Scan}$ interval).

The second pulse is longer than $T_{Scan}$ and considered valid. The input value is updated.

Filtering digital signals can be used in order to surpress instable signals (debouncing). It makes the detection of digital input signals more relyable.

The scan interval $T_{Scan}$ is configured by the parameter *inDiScanTime* (→ 3.4.4).

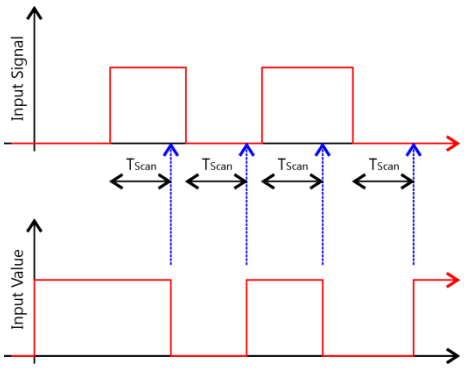


**Fig. 8 Inverted Reflect Mode**

Fig. 8 illustrates the digital input signal and the inverted input value with parameter *inDiInverted* set to "on".

LucidIoCtrl Command Line Tool Example
Set digital input channel 0 to Reflect Mode

```
LucidIoCtrl -dCOM4 -c0 -sinDiMode=reflect
```

## 3.2.2 Edge Detection

Digital input channels can operate in edge detection modes. In rising edge mode, the channel is sensitive for low-to-high transitions, in falling edge mode it recognizes high-to-low transitions.

Edge detection modes improve the real-time processing of input signal transitions without the host computer being involved.

The length of the shortest valid pulse is configured by the scan interval time $T_{Scan}$.

Example
The input should be able to detect an input signal pulse 10 ms length.

In theory it would be sufficient to read the digital input in reflect mode every 5 ms (with safety time) in order to detect the transition by the computer. But probably the operating system and communication overhead would violate the timing restrictions and pulses are lost.

In edge detection mode this can be solved by setting $T_{Scan}$ = 9 ms (including safety).

LucidIoCtrl Command Line Tool Example
Configure input channel 0 for rising edge detection mode

```
LucidIoCtrl -dCOM4 -c0 -sinDiMode=risingEdge
```

Set $T_{Scan}$ to 9 ms

```
LucidIoCtrl -dCOM4 -c0 -sinDiScanTime=9000
```

Read input channel 0

```
      LucidIoCtrl -dCOM4 -c0 -tL -r
->    CH0:01
```

### 3.2.2.1 Rising Edge Detection Mode

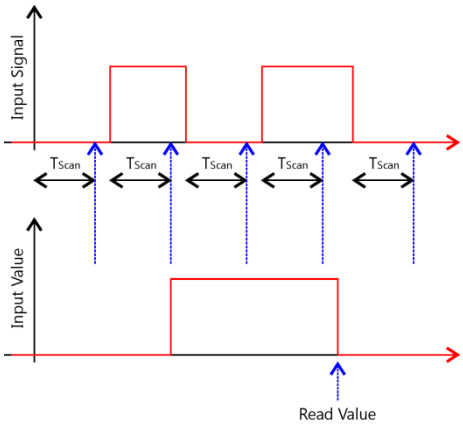


**Fig. 9 Rising Edge Detection Mode**

Fig. 9 shows a digital input signal and the corresponding input value in rising edge detection mode. After the HIGH input signal was detected as being valid, the input value remains pending until it was read by the host computer.

The rising edge detection mode allows detecting low-to-high transitions of the input signal without the host computer being involved.

### 3.2.2.2 Falling Edge Detection Mode

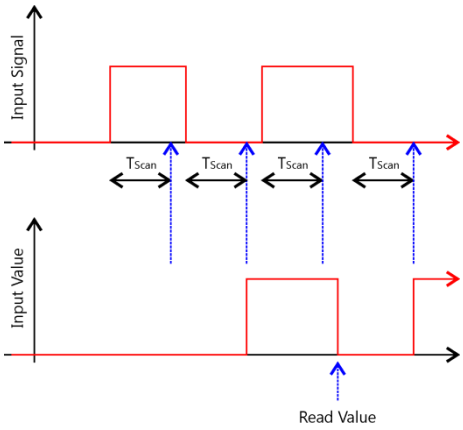


**Fig. 10 Falling Edge Detection Mode**

The falling edge detection mode (Fig. 10) operates similar to the rising edge detection mode (→ 3.2.2.1) but triggers at high-to-low transition of the input signal.

In falling edge detection mode, a high-to-low transition of the input signal sets the input value to “1”.

### 3.2.3 Count Mode

The count mode accumulates valid pulses of the input signal within a count interval specified by $T_{Count}$.

The counter is 16 bit wide. When 65535 is reached it rolls over to 0 with the next increment.

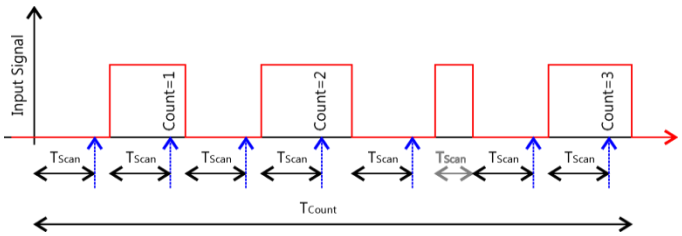


Fig. 11 illustrates a typical periodical input signal. In count mode all valid pulses are accumulated until the count interval $T_{Count}$ finishes.

**Fig. 11 Count Mode**

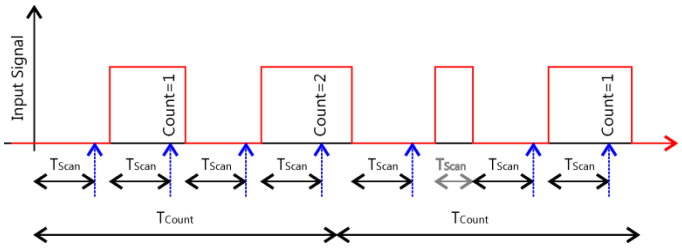


Fig. 12 shows the same input signal, but with a shorter count interval.

After count interval $T_{Count}$ has passed, the count value is accessible and a new measurement cycle starts.

**Fig. 12 Count Mode Interval**

The input signal is considered as valid after it was stable for at least the scan time $T_{Scan}$. Only valid pulses are accumulated (see the gray interrupted pulse).

$T_{Scan}$ time starts when the rising edge of the input signal was detected. In the case that the parameter *inDiInverted* is set to "on", $T_{Scan}$ time starts when the falling edge of the input signal is detected.

Reading the input value in count mode returns the count value of the last finished count interval cycle.

LucidIoCtrl Command Line Tool Example

Configure input channel 0 for Count Mode

```
LucidIoCtrl -dCOM4 -c0 -sinDiMode=count
```

Set Scan Time $T_{Scan}$ to 1ms

```
LucidIoCtrl -dCOM4 -c0 -sinDiScanTime=1000
```

Set Count Time $T_{Count}$ to 1s

```
LucidIoCtrl -dCOM4 -c0 -sinDiCountTime=1000000
```

Read count value (number of pulses)

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

In this example 100 pulses of at least 1 millisecond length have been acquired within a count interval of 1 second.

Since the input value is updated after count interval has passed it takes 1 second to update the value. Decreasing the count interval results in a faster update of the input value.

In Count Mode the value type N in supported. The count value is returned in hexadecimal and decimal format.

#### 3.2.3.1 Count Mode Options

The functionality of the count mode can be controlled by the parameters *inDiAddCounts* and *inDiResetCountsOnRead*.

The number counted pulses becomes valid after count interval $T_{Count}$ has passed:

- In the case that *inDiAddCounter* is set to "off", the current counter value is overwritten by the new counter value and previous counter value might be lost.
- In the case that *inDiAddCounter* is set to "on", the new counter value is added to the current counter value.

When the count number is read by the host:

- If *inDiResetCounterOnRead is set to* "*off*", reading the counter value does not affect the counter value.
- If *inDiResetCounterOnRead* is set to "on", the counter value is reset after reading it.

The parameter *inDiResetCounterOnRead* has only an effect when *inDiAddCounter* set to "on".

In order to avoid counter overflows parameter *inDiAddCounter* "on" should be combined with *inDiResetCounterOnRead* "on" parameter.

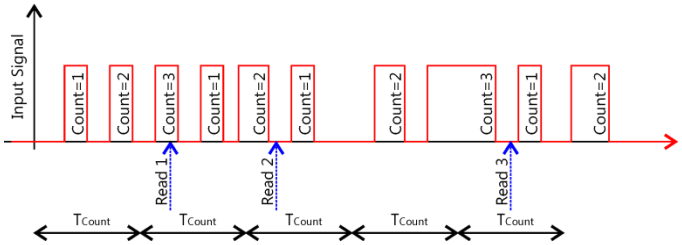


**Fig. 13 Count Value Add Mode**

Fig. 13 shows a non-periodic input signal with 10 pulses in total. The counter value is read at 3 times. In the following it is explained how the options affect the count value.

The value "Count" in the diagram refers to the internal counter which is accumulated during count interval $T_{Count}$.

| Mode | Value Read 1 | Value Read 2 | Value Read 3 |
|---|---|---|---|
| *inDiAddCounter* = "on"<br>*inDiResetCounterOnRead* = "on" | 2 | 3 | 3 |
| *inDiAddCounter* = "on"<br>*inDiResetCounterOnRead* = "off" | 2 | 5 | 8 |
| *inDiAddCounter* = "off"<br>*inDiResetCounterOnRead* = "off" (Default) | 2 | 3 | 2 |

**Tab. 5 Count Mode Options Value Read Results**

Tab. 5 lists the count values at the 3 reading positions for the *inDiAddCounter* and *inDiResetCounterOnRead* options.

The options *inDiAddCounter* and *inDiResetCounterOnRead* are set to "on" which is the most useful combination. (→ Tab. 5, first row)

On Read 1 the returned counter value is 2 since the internal counter value when $T_{Count}$ ended 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 returned counter value is 3 because of the internal counter value 2 plus the carried over counter 1 from the last count interval. The counter value is reset on read.

At the reading 3 the read value is 3. The counter value is reset on read.

Count mode with *inDiAddCounter* = "on"

In the case that *inDiAddCounter* is "on" and *inDiResetCounterOnRead* is "off" the pulses are accumulated but the counter is not reset on reading the value. This causes the counter value will be 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.

## 3.3 Commands

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

This section describes in detail the commands, which are supported by the DI4/DI8 modules.

### 3.3.1 GetIo

This command reads the logic value of an input.

| Mode | Value |
|---|---|
| Reflect | Logic input value 0 or 1 |
| Edge Detection | Pending edge event detected value 0 or 1 |
| Count | Counter value 0 to 65535 |

**Tab. 6 Value of Digital Input Channel**

Tab. 6 lists the digital input channel modes and how the IO value is interpreted.

| **Command** | GetIo | **Access** | Read |
|---|---|---|---|
| **Opcode** | 0x46 | | |
| **LucidIoCtrl Command Line Tool** | | | |
| **Call (-tL)** | `LucidIoCtrl –d[COMx] –c[Channel] –tL –r` | | |
| **Return** | CHn:L<br>n – Input Channel<br>L – Input Digital Value | | |
| **Call (-tN)** | `LucidIoCtrl –d[COMx] –c[Channel] –tN –r` | | |
| **Return** | CHn:V<br>n – Input Channel<br>V – Input Count Value | | |

**Tab. 7 GetIo Command**

LucidIoCtrl Command Line Tool Example

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

```
      LucidIoCtrl –dCOM4 –c0 –tL –r
->    CH0:01
```

Read input channel 0 operating in Count Mode

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

Request Frame

| OPC | P1 | P2 | LEN |
|---|---|---|---|
| 0x46 | Channel | Value Type | 0 |

**Tab. 8 GetIo Request Frame**

<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>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>0 / 1</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. 9 GetIo Request**

Response Frame:

| Status | LEN | Data Field |
|---|---|---|
| Status | Size | Value |

**Tab. 10 GetIo Response Frame**

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

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

### 3.3.2 GetIoGroup

This command reads the logic input values of a group of input 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">LucidIoCtrl Command Line Tool</td></tr>
<tr><td>Call (-tL)</td><td colspan="3"><code>LucidIoCtrl –d[COMx] –c[Channels] –tL –r</code><br><br><u>Channels:</u><br>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<br>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><td>Call (-tN)</td><td colspan="3"><code>LucidIoCtrl –d[COMx] –c[Channels] –tN –r</code><br><br><u>Channels:</u><br>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<br>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>

**Tab. 11 GetIoGroup Command**

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

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

```
      LucidIoCtrl –dCOM4 –c0,1,3 –tL –r
->    CH0:00  CH1:01  CH3:01
```

<u>Request Frame</u>

| OPC | P1 | P2 | LEN |
|---|---|---|---|
| 0x48 | Channel Mask | Value Type | 0 |

**Tab. 12 GetIoGroup Request Frame**

<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>Channel Mask</td><td>Channel Mask<br>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<br>P1A = 0x01</td></tr>
</table>
Values are bitwise or combined<br>Size of P1 is 1 or 2 bytes. If Bit 7 of P1 is set, a subsequent P1A is expected.<br><br><u>Examples:</u><br>Accessing channel numbers:<br>0 and 3 Value = 0x01 OR 0x08 = 0x09<br>1 and 2 Value = 0x02 OR 0x04 = 0x06<br>1, 2 and 7 Value P1 = 0x02 OR 0x04 = 0x86<br>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>0 / 1</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. 13 GetIoGroup Command**

<u>Response Frame:</u>

| Status | LEN | Data Field |
|---|---|---|
| Status | Length | Value |

**Tab. 14 GetIoGroup Response Frame**

Returns Execution Status Code, documented in the general *LucidControl 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:

Request frame reads input channels 0, 1, and 7

| Opcode | P1 | P1A | P2 | Length |
|---|---|---|---|---|
| 0x48 | 0x83 | 0x01 | 0x00 | 0x00 |

**Tab. 15 GetIoGroup Request Example**

Response Frame:

Channel 0 = "0", channel 1 = "1", channel 7 = "1"

| Header Field | | Data Field | | |
|---|---|---|---|---|
| Status | LEN | CH0 | CH1 | CH7 |
| 0x00 | 0x03 | 0x00 | 0x01 | 0x01 |

**Tab. 16 GetIoGroup Response Example**

## 3.4 Parameters

LucidControl 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 *LucidControl 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 LucidControl 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 parameter values should be avoided in order not to destroy the device over time.

### 3.4.1 inDiValue

This IO Configuration Parameter reflects the value of the input channel.

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 |
| **LucidIoCtrl Command Line Tool** | | | |
| **Parameter Name** | *inDiValue* | **Parameter Values** | 0 / 1 |
| **Call (Get)** | `LucidIoCtrl -d[COMx] -c[Channel] -ginDiValue` | | |

**Tab. 17 IO Configuration Parameter inDiValue**

LucidIoCtrl Command Line Tool Example

Read value of input channel 0:

```
      LucidIoCtrl -dCOM4 -c0 -ginDiValue
->    inDiValue=0
```

### 3.4.2 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">LucidIoCtrl Command Line Tool</td></tr>
<tr><td>Parameter Name</td><td>inDiMode</td><td>Parameter Values</td><td>Mode</td></tr>
<tr><td>Call (Set)</td><td colspan="3"><code>LucidIoCtrl –d[COMx] –c[Channel] –sinDiMode=[Mode] {-p} {--default}</code></td></tr>
<tr><td>Call (Get)</td><td colspan="3"><code>LucidIoCtrl –d[COMx] –c[Channel] –ginDiMode</code></td></tr>
</table>

**Tab. 18 IO Configuration Parameter inDiMode**

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
```

Read the operation mode of input channel 0

```
LucidIoCtrl –dCOM4 –c0 –ginDiMode
->  inDiMode=count
```

### 3.4.3 Bit Parameter inDiFlags

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

| **Parameter** | *inDiFlags* | **Access** | Read / Write |
|---|---|---|---|
| **Address** | 0x1101 | | |
| **Values** | Bit Parameters<br>**Bit Parameter** / **Bit Position**<br>*inDiAddCounter* / Bit 0<br>*inDiResetCounterOnRead* / Bit 1<br>*inDiInverted* / Bit 2 | | |
| **Default Value** | 0x00 | **Parameter Type** | 1 Byte unsigned |

**Tab. 19 IO Configuration Parameter inDiFlags**

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

If *inDiFlags* is changed by the SetParam command the update must be performed in a read-modify-write sequence in order to prevent overwriting other bit parameters.

#### 3.4.3.1 inDiInverted

This Bit Parameter configures the input signal value inversion.

| **Parameter** | *inDiFlags* | **Access** | Read / Write |
|---|---|---|---|
| **Address** | 0x1101 | Parameter bit *inDiFlags* | |
| **Values** | **Bit Parameter** / **Bit Position**<br>*inDiInverted* / Bit 2 | | |
| **Default Value** | Off | **Parameter Type** | 1 Bit |
| **LucidIoCtrl Command Line Tool** | | | |
| **Parameter Name** | *inDiInverted* | **Parameter Values** | on / off |
| **Call (Set)** | `LucidIoCtrl -d[COMx] -c[Channel] -sinDiInverted=[Value] {-p} {--default}` | | |
| **Call (Get)** | `LucidIoCtrl -d[COMx] -c[Channel] -ginDiInverted` | | |

**Tab. 20 IO Configuration Parameter Bit inDiInverted**

LucidIoCtrl Command Line Tool Example

Enable input signal value inversion of channel 0 and make the setting persistent.

```
LucidIoCtrl -dCOM4 -c0 -sinDiInverted=on -p
```

Read input signal value inversion flag of channel 0.

```
LucidIoCtrl -dCOM4 -c0 -ginDiInverted
->  inDiInverted=on
```

### 3.4.3.2 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 (→ 3.2.3).

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

**Tab. 21 IO Configuration Parameter Bit inDiAddCounter**

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
```

Read counter add setting of input channel 0

```
LucidIoCtrl -dCOM4 -c0 -ginDiAddCounter
->  inDiAddCounter=on
```

### 3.4.3.3 inDiResetCounterOnRead

This Bit Parameter controls how to update the counter value after it was read. It is relevant in Count Mode only (→ 3.2.3).

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

**Tab. 22 IO Configuration Parameter Bit inDiResetCounterOnRead**

LucidIoCtrl Command Line Tool Example

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

```
LucidIoCtrl -dCOM4 -c0 -sinDiResetCounterOnRead=on -p
```

Read counter reset setting of input channel 0

```
      LucidIoCtrl –dCOM4 –c0 –ginDiResetCounterOnRead
->    inDiResetCounterOnRead=on
```

## 3.4.4 inDiScanTime

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

| **Parameter** | *inDiScanTime* | **Access** | Read / Write |
|---|---|---|---|
| **Address** | 0x1111 | | |
| **Values** | $T_{Scan}$ in µs (micro seconds)<br>$t_{Min} \leq T_{Scan} \leq 1$ s | | |
| **Default Value** | 50,000 (50 ms) | **Parameter Type** | 4 Bytes unsigned |
| **LucidIoCtrl Command Line Tool** | | | |
| **Parameter Name** | *inDiScanTime* | **Parameter Values** | Time [µs] |
| **Call (Set)** | `LucidIoCtrl –d[COMx] –c[Channel] –sinDiScanTime=[Time] {-p} {--default}` | | |
| **Call (Get)** | `LucidIoCtrl –d[COMx] –c[Channel] –ginDiScanTime` | | |

**Tab. 23 IO Configuration Parameter inDiScanTime**

LucidIoCtrl Command Line Tool Example

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

```
      LucidIoCtrl –dCOM4 –c0 –sinDiScanTime=1500000 –p
```

Read $T_{Scan}$ parameter of input channel 0

```
      LucidIoCtrl –dCOM4 –c0 –ginDiScanTime
->    inDiScanTime=1500000
```

## 3.4.5 inDiCountTime

This IO Configuration Parameter specifies the count time $T_{Count}$ of the digital input in Count Mode (→ 3.2.3).

| **Parameter** | *inDiCountTime* | **Access** | Read / Write |
|---|---|---|---|
| **Address** | 0x1112 | | |
| **Values** | $T_{Count}$ in µs (micro seconds)<br>$t_{Min} \leq T_{Count} \leq 1$ h | | |
| **Default Value** | 5,000,000 (5 s) | **Parameter Type** | 4 Bytes unsigned |
| **LucidIoCtrl 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` | | |

**Tab. 24 IO Configuration Parameter inDiCountTime**

LucidIoCtrl Command Line Tool Example

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

```
      LucidIoCtrl –dCOM4 –c0 –sinDiCountTime=10000000 –p
```

Read $T_{Count}$ parameter of input channel 0

```
      LucidIoCtrl –dCOM4 –c0 –ginDiCountTime
->    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 |
| | Input Signal Maximum Voltage Note1 | 5V | $U_{5Max}$ | 7.5 V |
| | | 10V | $U_{10Max}$ | 15 V |
| | | 24V | $U_{24Max}$ | 30 V |
| | Input Impedance | | $R_{In}$ | > 1 kΩ |
| **Inputs – Timing Characteristic** | | | | |
| | $T_{Scan}$ | | $t_{Min} < T_{Scan} < 1$ s | |
| | $T_{Count}$ | | $t_{Min} < T_{Count} < 1$ h | |
| | Minimum pulse length | | $t_{Min}$ | 1 ms |
| | Timer Resolution | | $t_{Res}$ | 10 µs |
| | Maximum Frequency of Input Signal in Count Mode | | | 1000 Hz |
| **Module – Host Interface** | | | | |
| | USB | | | 2.0 Full Speed CDC Profile |
| | Power Supply | | | USB Power supplied |
| | 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 |

**Tab. 25 DI4/DI8 Specification**

Note1 Maximum forward and reverse voltage

Module is specified at environmental temperature of 25°C.

# 5 Order Information

| 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 |

**Tab. 26 Digital Input Module Order Codes**

# 6 Document Revision

| Date | Rev. | |
|---|---|---|
| 2018/08/04 | 2.0 | • Added documentation of DI8 module<br>• Added documentation of USB Isolation |
| 2022/02/04 | 2.1 | • Applied some corrections |
| 2024/08/22 | 2.2 | • Minor Updates and corrections |

**Tab. 27 Document Revision**

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