OPTICKÉ SKENOVÁNÍ S ROBOTICKÝM MANIPULÁTOREM EPSON

SEMESTRÁLNÍ PRÁCE  
SEMESTRAL THESIS

AUTOR PRÁCE  
AUTHOR
Bc. ADAM CHROMÝ

VEDOUCÍ PRÁCE  
SUPERVISOR
doc. Ing. LUDĚK ŽALUD, Ph.D.
INTRODUCTION

The aim of this work is to prepare base for my future project, which is the construction of 3D scanner. The final result of this work should be complex scanning system for acquiring informations about scanned object such as dimensions, shape, surface profile and other visual values.

In order to reach this goal, the proper laser scanner must be chosen, which could be done only in case of knowledge of the most important parameters. First chapter of this project briefly summarizes these parameters and evaluates the properness of SICK LMS400 laser scanner.

Because the ability of real-time controlling of laser scanner's functions is essential when we want to create 3D scanner, the C# driver for this device has been created. Creating universal driver was the major aim of this work, which is described in chapter 2.
1 LASER SCANNERS

This chapter provides brief general informations about laser scanners, its principles and important parameters. It provides also more detailed specifications of particular product applied in this solution: laser scanner SICK LMS 400.

1.1 General Informations about Laser Scanners

Laser scanner is a contactless measuring system, which uses controlled steering of laser beam followed by distance measurement in order to construct profile of scene in device’s angle of view [1]. The output data from laser scanner is a set of distances between scanner and the nearest object in appropriate angle of view.

Basic laser scanner involves a laser range finder based on any principle described in section 1.1.1. Its laser beam is reflected by a rotating mirror gathering distance measurements at specified angle intervals.

Reason to use of laser instead of any other signal is because of its important advantages:

- Laser provides concentrated narrow beam with very low deviation. It allows measuring distance to very tiny objects, such a depth of very narrow holes, what is not possible by ultrasonic or radio beams.
- Laser beam is monochromatic, what simplifies measurement processing.
- Laser has a high intensity which is nearly not diffusing, so it is not as much attenuated by environment as other mediums. It allows performing of long-distance measurements.

1.1.1 Measuring Principles

There are 4 most common measuring principles implemented in laser range finders and laser scanners [1]. Each of these methods has its advantages and disadvantages, so they are used in different devices for different purposes.

Time Of Flight Method

This method is based on a very simple principle: there is a generator inside the laser scanner, which is able to generate very tiny pulses of laser beam. These pulses travel through an environment to the object, where are reflected and travel back to the detector. Time difference between pulse generation and its recognition at detector is proportional to the distance of object (fig. 1.1).
Although this principle is really simple, its realization is quite hard. Let’s imagine, that distance to the object is given by the equation:

\[ d_{\text{obj}} = \frac{1}{2} c \cdot t_{\text{of}} \]  

(1.1)

where \( c \) is speed of light \([m.s^{-1}]\) and \( t_{\text{of}} \) is time of laser beam’s flight \([s]\).

If we consider a distance of 1 mm, what is ordinary required resolution in many applications, by solving the equation (1.1) we can enumerate the required resolution of time-measuring unit of laser scanner:

\[ t_{\text{of}} = \frac{2 \cdot d_{\text{obj}}}{c} = \frac{2 \cdot 1 \cdot 10^{-3}}{2,99 \cdot 10^8} = 6,69 \cdot 10^{-12} = 6,69 \text{ ps} \]  

(1.2)

This implies, that we must have integrated a very precise time-measuring unit with resolution of picoseconds, if we want to reach this resolution of distance measuring.

The advantage of time-of-flight devices is that they are able to measure very long distances, on the order of kilometres, so they are suitable for scanning large objects like buildings or landscape profile. They are also independent on changes of temperature, because of compactness.

The disadvantage of time-of-flight devices is their accuracy, as was mentioned above. Due to the high speed of light, the time measuring is complicated and the accuracy of the distance measurement is relatively low, on the order of millimetres [1].

Triangulation Method

The principle of triangulation method is shown on figure 1.2. There is a laser beam generator, which produces beam which is consequently reflected by object. Reflected
beam is concentrated by lens and reaches the optical position detector, for example PSD or CCD. There is a mark detected by position detector, which placing corresponds to distance of object.

![Triangulation measuring principle](image)

Fig. 1.2: Triangulation measuring principle

Triangulation-based devices are exactly the opposite according to time-of-flight type. They have a limited range of few meters, but their accuracy is relatively high. The accuracy of triangulation device is on the order of tens of micrometers, so they are suitable for scanning tiny structures like a bark of trees, etc.

Another disadvantage of this principle is a dependency on changes in temperature, what could be caused for example by strong sunlight on one side of the device. Link between transceiver and receiver will expand and slowly distort the scanned data. Some more developed laser scanners have level compensators to counteract any movement of the link during the scan process [1].

**Interferometric Method**

Principle of interferometric method is in interference of two beams which one of them passed longer way than second one. As shown on figure [1.3] generated laser beam reaches the semi-permeable mirror, where is split into two beams. First beam goes though the mirror, then is reflected by object and consequently by semi-permeable mirror and finally reaches the detector. Second beam is reflected by semi-permeable mirror, then goes through the compensator, consequently is reflected by mirror in reference distance, then goes once again through compensator, through mirror and finally reaches the detector.
The compensation block is placed into the way of second beam because the first beam goes through the glass 3 times and second one only once.

Depending on distance to the object, the time difference of two beams occurs, which is related to the difference of length of beam flight. According to this phenomenon, constructive or destructive interference appears and considering positions of minima and maxima we can define desired distance.

Interferometric principle is the method, which provides most precise measurements. Its accuracy is typically on the order of micrometres, rarely even on the order of nanometres. Use of these devices is unusual, because its accuracy is balanced by very high costs, hugeness and complexity of device.

**Phase Shift Method**

Phase Shift method is modified variation of Time of Flight method described above, which uses some features from Interferometric method. Compared to Time of Flight method, it differs in fact, that outputting laser beam has sinusoidally modulated optical power. Reflected light is monitored, and the phase of the power modulation is compared with that of the sent light. This phase shift is proportional to time of flight.
As for an interferometer, the phase shift method involves an ambiguity regarding the measured distance, because with increasing distance the phase will vary periodically. However, the periodicity is much larger than in an interferometer, since the modulation frequency is much lower than the optical frequency. Also, the ambiguity can be easily removed by measuring with two different modulation frequencies simultaneously [2].

Compared with interferometers, devices based on the phase shift technique are less accurate, but they allow unambiguous measurements over larger distances.

1.1.2 Important Parameters of Laser Scanners

There are many types of laser scanners, which are varying a lot from each other. As was mentioned above, for instance measuring principle significantly influences final parameters of laser scanner. The most important parameters, which should be considered when choosing the best fitting product, are:

- **Measuring range** – minimal and maximal distance from scanner to object that can be measured.
- **Field of view** – angular width of view. Together with measuring range defines working area.
- **Angular resolution** – the smallest angular difference between two measured points. Defines density of acquired points and consequently influences minimal distance between two measured points.
- **Scanning frequency** – updating frequency of point value. Defines how many measurements of distance to the one point are performed each second. Usually depends on field of view and angular resolution.
- **Resolution and accuracy** – resolution defines minimal difference between two distances which are noticeable by scanner and accuracy is defined as area around measured value in which manufacturer declares that the true value is located for sure.
- **Data interface** – important parameter which specifies how data can be transferred from measuring device to the processing device. Influences possible applications of scanner.
- **Weight** – important in cases when placed on some places where maximum load is limited, e.g. at the wrist of robotic manipulator.
1.2 Laser Scanner SICK LMS 400

Laser Scanner SICK LMS 400 is one of the most accurate laser scanners disposing with 3 m range. There are many of more accurate scanners, but no one has so wide measuring range as this model. This section presents important parameters of this model and summarizes if it is proper device according to the aim of this project.

1.2.1 Specifications

Working Area

The maximum measuring range of the LMS400 is 3 m and the smallest permitted distance of the measurement object is 700 mm. The field of view covers an angle of 70°[3]. Entire working area is shown on figure 1.4.

![Fig. 1.4: Working area of SICK LMS 400](image)

Angular Resolution

Angular resolution is configurable in range from 0.1333 to 1°. Maximal angular error of rotating mirror is ±0.1°[3]. In case of measuring in perpendicular plane relative to projection axis of scanner, given angular error corresponds to 5.2 mm error of laser beam positioning in 3 m distance and 1.2 mm error in 0.7 m distance.
Scanning Frequency

Scanning frequency is configurable in range from 360 Hz to 500 Hz [3]. Configured combination of scanning frequency and angular resolution influences measurement accuracy, so not all the combinations are allowed.

Resolution and Accuracy

The typical systematic measuring error of the LMS400 is ±4 mm. This information applies for the individual measurement point at an object remission in range from 10% to 100% at room temperature [3].

The statistical measuring error is dependent on the remission and distance of the object. Tab. 1.1 shows typical and maximal measuring errors when operating in permitted combination of scanning frequency and angular resolution, room temperature and maximum external light of 2000 Lux.

<table>
<thead>
<tr>
<th>Remission</th>
<th>Distance</th>
<th>Statistical error</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Typical</td>
</tr>
<tr>
<td>100%/200%</td>
<td>700 to 3000 mm</td>
<td>3 mm</td>
</tr>
<tr>
<td>78%</td>
<td>700 to 999 mm</td>
<td>3 mm</td>
</tr>
<tr>
<td></td>
<td>1000 to 2500 mm</td>
<td>3 mm</td>
</tr>
<tr>
<td></td>
<td>2501 to 3000 mm</td>
<td>3 mm</td>
</tr>
<tr>
<td>40%</td>
<td>700 to 999 mm</td>
<td>4 mm</td>
</tr>
<tr>
<td></td>
<td>1000 to 2500 mm</td>
<td>4 mm</td>
</tr>
<tr>
<td></td>
<td>2501 to 3000 mm</td>
<td>4 mm</td>
</tr>
<tr>
<td>10%</td>
<td>700 to 999 mm</td>
<td>9 mm</td>
</tr>
<tr>
<td></td>
<td>1000 to 2500 mm</td>
<td>9 mm</td>
</tr>
<tr>
<td></td>
<td>2501 to 3000 mm</td>
<td>9 mm</td>
</tr>
<tr>
<td>6,5%</td>
<td>700 to 3000 mm</td>
<td>10 mm</td>
</tr>
</tbody>
</table>

Data Interface

The LMS400 has three different interfaces which each of them could be used for device configuring and also for receiving measured values [3]:

- **Ethernet** – TCP/IP peer to peer interface, data transmission rate of 10 MBaud (10BaseET), only half duplex supported. IP address, TCP/IP port and subnet mask is predefined by manufacturer but can be changed by configuration message. Intended for measurements output because of its baudrate.
• **Host interface** – Intended for the same use as Ethernet, but because of its baudrate doesn’t allow real-time measurement output. It’s physical layer can be configured as both RS-232 or RS-422. The interface parameters are freely configurable. The factory setting for the host interface is as follows: RS-232, 9600 Baud, 8 data bits, 1 stop bit, no parity.

• **Terminal interface** – this feature can operate simultaneously with the Ethernet or Host interface and is primarily intended to provide a reliable data connection for configuration. Therefore, its interface parameters cannot be changed and are as follows: RS-232, 9600 Baud, 8 data bits, 1 stop bit, no parity. If real-time output of values is required, the only usable interface is Ethernet because of limited baudrate of other interfaces.

**Weight**

Weight of device is approximately 2.3 kg. [3]

### 1.2.2 Summary

The purpose of the future 3D scanner is not exactly specified. If scanned object is a large structure, the measuring range must be as wide as possible, on the other hand, if it is a tiny object, the accuracy is decisive factor. But we don’t know what type of objects will be scanned, so our solution must be universal.

If we look at the devices available on market, the best balance between accuracy and working area is just at SICK LMS400. The angular resolution is also sufficient as well as weight, which is inside the manipulator’s payload range. This device disposes of fast Ethernet interface, which is also required. So finally, we can declare, that for the purpose of a universal scanner, the laser scanner SICK LMS400 is proper device.
2 INTERFACING LASER SCANNER

This chapter describes solution of laser scanner on-line configuring and acquiring measured values. The result of this chapter is on-line scanner configuration and real-time output data acquired from laser scanner by Ethernet.

2.1 Communication Interface

As written in section 1.2.1, LMS400 disposes with various interfaces, but only one is capable to serve real-time transmission. Hence we chose the Ethernet interface for connecting the laser scanner.

Device is capable of communicating via standard TCP/IP protocol. Factory default settings of parameters are following [3]:

- IP address: 192.168.0.1
- TCP/IP port: 2111
- Subnet Mask: 255.255.255.0

These settings can be changed by Messages, which are described in following section.

2.1.1 Scanner Controlling

Remote device configuration as well as measure triggering and measured data outputting is realized using messages sent over any available interface. The message is in most cases ASCII-encoded string representing transmitted request or received parameters. Message could also be binary coded, eg. in case of returning output data from measurement.

In case of Ethernet interface, the message is framed as follows [3]:

![Fig. 2.1: Structure of Ethernet frame for SICK LMS400.](image)
Message block on fig. 2.1 contains request or output data. For detailed overview of messages see [3].

2.2 Driver for Remote Controlling of Scanner

Direct configuration by messages is not a friendly and flexible approach to laser scanner control. The right programmer’s approach is to create driver for this scanner, which encapsulates all available capabilities into one class, which offers methods for controlling them. Because no driver for this scanner is available, I programmed it by myself in C# programming language, which is also the language of future 3D scanner project.

2.2.1 Driver Basic Principles

Driver class implements its own thread, where all the inner mechanisms are performed. Because of it, driver is possible to provide asynchronous events, which occurs in cases of received output data from measurement or in cases of device configuring, measure starting, stopping, etc.

The main part of drivers interface consists of methods, which allows user to perform any possible scanner handling. Behind these methods, translation to message and its transmitting is realized.

Driver is designed as robust as I can, which means that error states are carefully handled and delivered to user by recommended way – using object derived from EventArgs.

2.2.2 Library Interface

Detailed documentation about LMS400 Driver, describing its using and provided interface, is attached as Appendix A. The driver itself in form of source codes is available on enclosed DVD with electronic attachments.

The fastest way to understand all the abilities of this driver is to look at the source codes of Demonstration Application, which are also included on enclosed DVD with electronic attachments.

If you don’t have an access to the LMS400 laser scanner, so you aren’t able to try it, you can see screenshot-video of Demonstration Application using which is also available on that DVD.
2.3 Demonstration Application of Scanner Remote Controlling

In order to show usage of the LMS400 Driver, the Demonstration Applications has been created. Using the Windows Form, it makes all the driver’s abilities available for try. Screenshot of this application is shown on fig. 2.2.

![Screenshot of LMS400 Driver Demonstration Application](image)

Fig. 2.2: Screenshot of LMS400 Driver Demonstration Application.

On the right side of window, the toolbar is displayed. It contains all features necessary to scanner control. Each ability of scanner can be controlled from here.

The main part of window is covered by visualization, which can be performed in two modes:

- **Angular View** – shows the position of object by dots in appropriate location within scanners range. Sequence of dots creates a curve relevant to the scanned profile. This view is easy comprehensible, but doesn’t allow to show all the details because of its scale.
• **Plain View** – displays scanned profile in form of graph, where X axis is the angle of view and Y axis means measured distance. Angular transformation is performed on output data, so the flat areas are shown as flat even if their distance in not equal. This view shows more details and also can display remission of material by the color varying from black to white according to remission.

The same profile shown in both modes is presented on figure 2.3.

![Plain View vs Angular View](image)

Fig. 2.3: Comparison of viewing modes of Demonstration Application.

If you don’t have an access to the LMS400 laser scanner and so you aren’t able to try this Demonstration Application, you can see screenshot-video of it in use, which is available on enclosed DVD with attachments.

### 2.4 Summary

In this chapter, we created driver for interfacing SICK LMS400 laser scanner in order to configure it and perform measuring. This driver was programmed in C# language, the same programming language as the future project will be created in.

Using my driver, we established connection with laser scanner. By using the Demonstration Application, we showed, that we are able to acquire measured values. This ability will be necessary in the 3D scanner construction.
3 CONCLUSION

The main scope of this project was to create driver library for configuring SICK LMS400 laser scanner and performing on-line measurements by this device. This was successfully done, and universal driver for this scanner has been released. It has implemented almost entire interface, which is normally provided by its messages, sent over TCP/IP. This driver class allows to manage it by much more simple way – only by using methods of the control object.

The minor part of this work was evaluating the properness of this device for our purposes. Finally, we decided that this device is the best one for our aim, because it has a wide measuring range with also sufficient accuracy. These two parameters are fundamental for laser scanner, which should be used in universal 3D scanner.
BIBLIOGRAPHY


LIST OF SYMBOLS, PHYSICAL CONSTANTS AND ABBREVIATIONS

c  The speed of light \([m \cdot s^{-1}]\)

\(d_{\text{obj}}\)  Distance from laser scanner to the object \([m]\)

\(t_{\text{of}}\)  Time difference between laser beam’s generation and detection of itself reflected back \([s]\)
LIST OF APPENDICES

A  LMS400 Driver Operating Manual 20
   A.1 Description of Public Methods 20
   A.2 Description of Public Parameters 25
   A.3 Description of Events 26
A LMS400 DRIVER OPERATING MANUAL

<table>
<thead>
<tr>
<th>Author</th>
<th>Bc. Adam Chromý</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version</td>
<td>1.0.0</td>
</tr>
<tr>
<td>Release date</td>
<td>26. 3. 2012</td>
</tr>
<tr>
<td>Description</td>
<td>C# driver for configuring and controlling operation of SICK LMS400 laser scanner.</td>
</tr>
<tr>
<td>Licence</td>
<td>GNU GPL</td>
</tr>
</tbody>
</table>

The best way to explore all the functions and find out how to use this driver is to see the demonstrating program, which is also attached.

For brief overview of driver’s and demonstrating program’s features see the attached demonstrating video.

A.1 Description of Public Methods

Open

Syntax:

```csharp
void Open()
```

Requires power-supplied LMS400 device connected to the driver by Ethernet. This method opens TCP/IP communication between computer and device. If device is successfully connected, `OnConnected` event occurs.

Close

Syntax:

```csharp
void Close()
```

Terminates connection with device. If successfully disconnected, `OnClosed` event occurs.

StartMeas

Syntax:

```csharp
void StartMeas(EMeasuredData whatToMeasure = EMeasuredData.DistanceOnly)
```
Starts continuous measuring process. If it is successful, **OnMeasureStart** event occurs. Data are available in **RawData** property and each time when they have been refreshed, **OnMeasured** event occurs. Continuous measurement can be stopped by **StopMeas**. Parameter **whatToMeasure** defines which value is measured (distance, remission). If omitted, only distance is measured.

**TriggeredMeas**

Syntax:

```c
void TriggeredMeas(EMeasuredData whatToMeasure = EMeasuredData.DistanceOnly)
```

Starts measurement in mode where measuring is triggered by selected trigger. You can select one of available triggers by **SelectTrigger** method. For software triggering use **SingleMeas** method instead this one. Data are available in **RawData** property and each time when they have been refreshed, **OnMeasured** event occurs. Parameter **whatToMeasure** defines which value is measured (distance, remission). If omitted, only distance is measured.

**SingleMeas**

Syntax:

```c
void SingleMeas(EMeasuredData whatToMeasure)
void SingleMeas(int count = 1)
void SingleMeas(EMeasuredData whatToMeasure, int count)
```

Performs **count** measurements of value defined by **whatToMeasure**. If **count** is omitted, one measurement is performed. Data are available in **RawData** property and each time when they have been refreshed, **OnMeasured** event occurs. If **whatToMeasure** is omitted, only distance is measured.

**StopMeas**

Syntax:

```c
void StopMeas()
```

Stop continuous measurement (if it is actually performed) and if this action is successful, **OnMeasureStop** event occurs.
ConfigByFreqAndAng

Syntax:

```c
void ConfigByFreqAndAng(int Frequency, float AngularResolution,
    float StartingAngle = 55.0F, float ScanWidth = 70.0F)
```

Configures angular resolution and scanning frequency. Not every combination of these parameters is possible, so device computes nearest possible combination and set parameters to these values. Then `OnParametersChanged` event occurs. Values of parameters which were really used in configuration of device can be obtained from `InfoLMS` property.

**Warning:** Device allows to configure value, which are out of range guaranteed by vendor. Check `ValueQuality` at `InfoLMS`, which refers to actual device precision (more at `InfoLMS` section).

ConfigByFreq

Syntax:

```c
void ConfigByFreq(int Frequency, ERoughAngularValue
    AngularResolution, float StartingAngle = 55.0F, float
    ScanWidth = 70.0F)
```

Configures angular resolution and scanning frequency. Device tries to set up exact value of scanning frequency, appropriate angular resolution is counted by device. Then `OnParametersChanged` event occurs. Values of parameters which were really used in configuration of device can be obtained from `InfoLMS` property.

**Warning:** Device allows to configure value, which are out of range guaranteed by vendor. Check `ValueQuality` at `InfoLMS`, which refers to actual device precision (more at `InfoLMS` section).

ConfigByAng

Syntax:

```c
ConfigByAng(ERoughFrequency Frequency, float AngularResolution,
    float StartingAngle = 55.0F, float ScanWidth = 70.0F)
```

Configures angle resolution and scanning frequency. Device tries to set up exact value of angular resolution, appropriate scanning frequency is counted by device. Then `OnParametersChanged` event occurs. Values of parameters which were really used in configuration of device can be obtained from `InfoLMS` property.
**Warning:** Device allows to configure value, which are out of range guaranteed by vendor. Check `ValueQuality` at `InfoLMS`, which refers to actual device precision (more at InfoLMS section).

**SaveParameters**

Syntax:

```c
void SaveParameters()
```

Saves actual configuration parameters as default settings, which are applied to device after start up (power on). These values are saved in device’s EEPROM, so no power supply is required to store them.

**Run**

Syntax:

```c
void Run()
```

Exits configuration mode of device. This step is performed automatically when any measuring command is received.

**Reset**

Syntax:

```c
void reset()
```

Switch off device and switch it on again. Then default values are applied to the configuration parameters. During this process, TCP/IP connection with driver is lost because of lost of power supply. It leads to device disconnecting, so it is necessary to reconnect by `Open` method after device has been restarted.

**Synchronize**

Syntax:

```c
void Synchronize(ESyncMode SyncMode, int PhaseOffset = 0)
```

Set up synchronizing when working of two devices in pair is required. First device is configured as Master and second as Slave. Devices must be connected by System connection. See Operating Instructions for details.
MedianFilter

Syntax:

```c
void EnableMedianFilter()
void DisableMedianFilter()
```

Enables/disables Median Filter. Median value is chosen from group of 9 values:
- actual value of appropriate point
- actual value of first point left
- actual value of first point right
- previous value of appropriate point
- previous value of first point left
- previous value of first point right
- second-last value of appropriate point
- second-last value of first point left
- second-last value of first point right

EdgeFilter

Syntax:

```c
void EnableEdgeFilter()
void DisableEdgeFilter()
```

Enables/disables Edge Filters. The edge filter prevents incorrect/extreme distance values at edges that result from it not being possible to determine a distance value for the previous or next point (e.g. if the previous/next measured point was too dark or outside the measuring range of 3 metres).

RangeFilter

Syntax:

```c
void EnableRangeFilter(float BottomLimit, float TopLimit)
void DisableRangeFilter()
```

Enables/disables Range Filter. Only values bigger than `BottomLimit` and smaller than `TopLimit` are provided. Other values are set to 0.

MeanFilter

Syntax:
void EnableMeanFilter(int NumberOfMeans)
void DisableMeanFilter()

Enables/disables Mean Filter. Mean is counted from \texttt{NumberOfMeans} last values.

Note: applying mean filter slows the measurement, the scanning frequency is then \texttt{NumberOfMeans}-times smaller then configured.

SelectTrigger

Syntax:
\texttt{SelectTrigger(ETriggerSource StartSource, ETriggerSource StopSource)}

Selects trigger to begin and end measurement. For more information see Operating Instructions.

\subsection*{A.2 Description of Public Parameters}

RawData

Structure containing set of measured points. Each point contains values of distance, remission and angle of scanning. If measuring of parameter is not performed (depends on configuration) or measured value is incorrect (out of range, etc.), the value is 0. For recommended use see demonstrating program.

InfoLMS

Contains information about configuration of device. \texttt{InfoLMS} is a structure containing following properties:

- \texttt{UserLevel} – user level of actually connected user
- \texttt{StartTrigger, StopTrigger} – actually selected triggers of measuring
- \texttt{Name} – name of the device, same as \texttt{Name} property.
- \texttt{FirmwareVersion} – version of device firmware
- \texttt{ScanningFrequency} – actual scanning frequency. If frequency is not acquired yet, the value is 0.
- \texttt{AngularResolution} – actual scanning frequency. If frequency is not acquired yet, the value is 0.
- \texttt{ValueQuality} – quality index of actual combination of angular resolution and scanning frequency. If 10, device is operating with best quality, if 7, device is operating with precision defined in datasheet, if less than 7, device is operating out of the range.
Name

Returns name of device. In most cases the return value will be “LMS400_XX00” depending on the device version.

Connected

Readonly parameter which returns true if device is actually connected to driver, otherwise it returns false.

A.3 Description of Events

OnConnected

Syntax:

```csharp
public event ChangedEventHandler OnConnected;
```

Occurs when driver become connected with device by TCP/IP. When link is not successfully established, an exception is thrown.

OnClosed

Syntax:

```csharp
public event ChangedEventHandler OnClosed;
```

Occurs when driver is successfully disconnected from the device. This event occurs also in case when device is not disconnected properly, but also exception is thrown.

OnMeasureStart

Syntax:

```csharp
public event ChangedEventHandler OnMeasureStart;
```

Occurs when continuous measurement is successfully started by `StartMeas` method.

OnMeasureStop

Syntax:

```csharp
public event ChangedEventHandler OnMeasureStop;
```

Occurs when continuous measurement is successfully stopped by `StopMeas` method.
OnMeasured

Syntax:

```csharp
public event ChangedEventHandler OnConnected;
```

Occurs every time when cyclic message with measured data is received and data are processed and prepared for reading. Is recommended to use this event to further processing of received data. Event handling method receives measured data as a parameter in form of `RawDataSet` object.

OnParametersChanged

Syntax:

```csharp
public event ParametersEventHandler OnParametersChanged;
```

Occurs when scanning frequency, angular resolution or quality index has been changed. Is recommended to use this event to acquire real values of these settings decided by device (which may vary from desired values).

OnParametersSaved

Syntax:

```csharp
public event ChangedEventHandler OnParametersSaved;
```

Occurs when actual configuration has been set as default configuration and saved into the device memory.

OnSynchronizingChanged

Syntax:

```csharp
public event ChangedEventHandler OnSynchronizingChanged;
```

Occurs when synchronizing has been switched on, or synchronizing role has been changed from one role to another (master <> slave).

OnFilterChanged

Syntax:

```csharp
public event ChangedEventHandler OnFilterChanged;
```

Occurs when at least one of possible filters is switched on or off, or its parameters have been changed.
OnException

Syntax:

```csharp
public event ExceptionEventHandler OnException;
```

Occurs when exception is thrown. Event carries information about exception stored in object of `EventArgs` type. Object contains text message describing exception and date and time when exception occurred.