



# TLE1 Sensor

Technical specification (rev.J, October 2020), sensor firmware: 111 3v5



# Table of contents

1. Introduction .....	6
2. Description .....	7
2.1 Dimensions (TLE1-35-65) .....	7
2.2 Dimensions (TLE1-58-122) .....	8
2.3 Dimensions (TLE1-100-300) .....	9
2.4 Interface .....	10
2.5 Measuring range .....	12
2.6 Framerate .....	13
2.7 Firmware .....	13
3. Command set .....	14
3.1 DATA .....	14
3.2 DATA_STREAM_start .....	14
3.3 DATA_STREAM_stop .....	14
3.4 DISTANCE_PROFILE_READ .....	14
3.5 FIRMWARE .....	13
3.6 READ_RAW .....	15
3.7 READ_RAW_1/4 .....	15
3.8 MODE .....	15
3.9 LASER_ON/OFF .....	15
3.10 EXTENDED_DATA_ON/OFF .....	15
3.11 SET_AUTO_EXPO .....	15
3.12 SET_BANK_PARAMETERS .....	15
3.13 USER_EEPROM_WRITE .....	16
3.14 USER_EEPROM_READ .....	16
3.15 REGISTER_WRITE .....	17
3.16 REGISTER_READ .....	17
4. Data format .....	18
5. EEPROM memory map .....	19
6. Registers map .....	20
7. Sensor parameters .....	21
7.1 Image .....	21
7.2 Profile .....	22
7.3 Profile processing .....	23
7.4 Other parameters .....	23
7.5 Register only .....	24
8. Measuring modes .....	26
8.1 MODE 0 - Mean Distance .....	27
8.2 MODE 1 - Closest Segment .....	28

- 8.3 MODE 2 - First Light Segment ..... 30
- 8.4 MODE 3 - Last Light Segment ..... 31
- 8.5 MODE 4 - Furthest Segment..... 31
- 8.6 MODE 5 - Gap Position..... 32
- 8.7 MODE 6 - Gap Height ..... 32
- 8.8 MODE 7 - Object Position ..... 32
- 8.9 MODE 8 - Object Height ..... 32
- 9. Digital I/O ..... 33
  - 9.1 Output configuration ..... 33
  - 9.2 Input configuration..... 34
  - 9.3 Read/write ..... 37
- 10. Installation..... 38
- 11. Code examples ..... 39
- 12. Package, warranty, contacts ..... 42

# List of figures

Image 1: TLE1-35-65 sensor dimensions (mm) .....	7
Image 2: TLE1-58-122 sensor dimensions (mm) .....	8
Image 3: TLE1-100-300 sensor dimensions (mm) .....	9
Image 4: Back panel with RJ45 .....	10
Image 5: Back panel with RJ45 and I/O/PWR .....	10
Image 6: Detail of I/O/PWR connector.....	11
Image 7: Measuring range visualization .....	12
Image 8: Sensor framerate .....	13
Image 9: Sensor memory structure .....	19
Image 10: Register map .....	20
Image 11: MODE 0 example.....	27
Image 12: MODE 1 example .....	28
Image 13: MODE 1 detailed view .....	29
Image 14: MODE 2 example .....	30
Image 15: MODE 3 example .....	30
Image 16: Ambient light optical filter transmittance .....	38
Image 17: Class 1M Laser safety label.....	38

# List of tables

Table 1: Sensor specifications .....	6
Table 2: Factory IP setting .....	10
Table 3: AUX_BYTE format .....	18
Table 4: Image sensor parameters .....	21
Table 5: Profile sensor parameters .....	22
Table 6: Profile processing parameters .....	23
Table 7: Other sensor parameters .....	23
Table 8: Register only parameters .....	24
Table 9: Input read/write .....	25
Table 10: Output configuration .....	33
Table 11: Output type configuration .....	33
Table 12: Output source selection .....	34
Table 13: Input configuration .....	35
Table 14: Input connections .....	35
Table 15: Inputs to user LED connection .....	36
Table 16: Inputs to laser on/off connection .....	36
Table 17: Position encoder channel 1 .....	36
Table 18: Position encoder channel 0 .....	36
Table 19: Input read/write .....	37

# 1. Introduction

The TLE1 sensor integrates laser line triangulation technology with Ethernet interface. It projects a laser line on the measured surface, instead of a single point as seen on standard triangulation sensors for distance measurement. The reflected light is captured by an area CMOS, which features an enhanced detail (for material profile measurement, i.e. hole size, gap, edge) and a much more accurate distance measurement (due to the high number of data points). All processing occurs in the sensor itself. No external controller or PC are required in its operation.

	TLE1-35-65	TLE1-58-122	TLE1-100-300
Distance measurement range	35 to 65 mm	58 to 122 mm	100 to 300 mm
Height measurement range	17 mm at 35 mm distance 24 mm at 65 mm distance	23 mm at 58 mm distance 40 mm at 122 mm distance	30 mm at 100 mm distance 90 mm at 300 mm distance
Line size *	40 $\mu\text{m}$ $\times$ 20.5 mm	60 $\mu\text{m}$ $\times$ 31.5 mm	140 $\mu\text{m}$ $\times$ 60 mm
Detector	CMOS (1280x1024 pixels)		
Distance Resolution	1 $\mu\text{m}$	3 $\mu\text{m}$	6 $\mu\text{m}$
Distance Nonlinearity	10 $\mu\text{m}$	25 $\mu\text{m}$	50 $\mu\text{m}$
Height Resolution	24 $\mu\text{m}$	40 $\mu\text{m}$	80 $\mu\text{m}$
Response Time	33.34 ms for full readout or faster for partial readout		
Interface	RJ45 (Ethernet) - 10/100 Mbit		
Communication	TCP protocol, ICMP (ping)		
Digital inputs (optional)	minimum pulse width 200 $\mu\text{s}$		
Power	POE (Power over Ethernet), from 12 (150mA) to 24 VDC (80mA)		
Weight	105 g		
Dimensions	100 $\times$ 70 $\times$ 25 mm		
Laser	650 nm, Class 1M, P<3mW	650 nm, Class 2M, P<10mW	650 nm, Class 2M, P<10mW

Table 1: Sensor specifications

\* at middle measuring range

## 2. Description

### 2.1 Dimensions (TLE1-35-65)

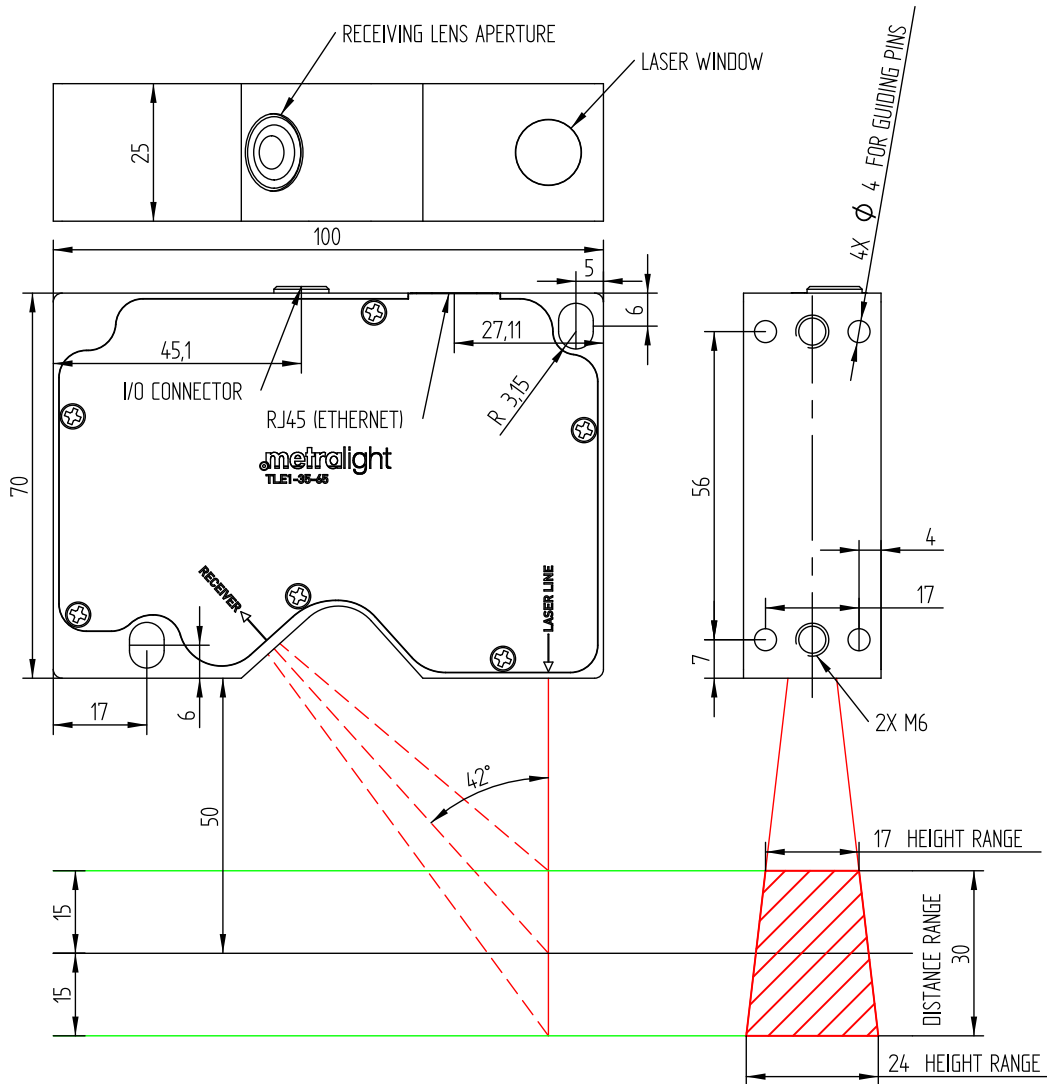


Image 1: TLE1-35-65 sensor dimensions (mm)

## 2.2 Dimensions (TLE1-58-122)

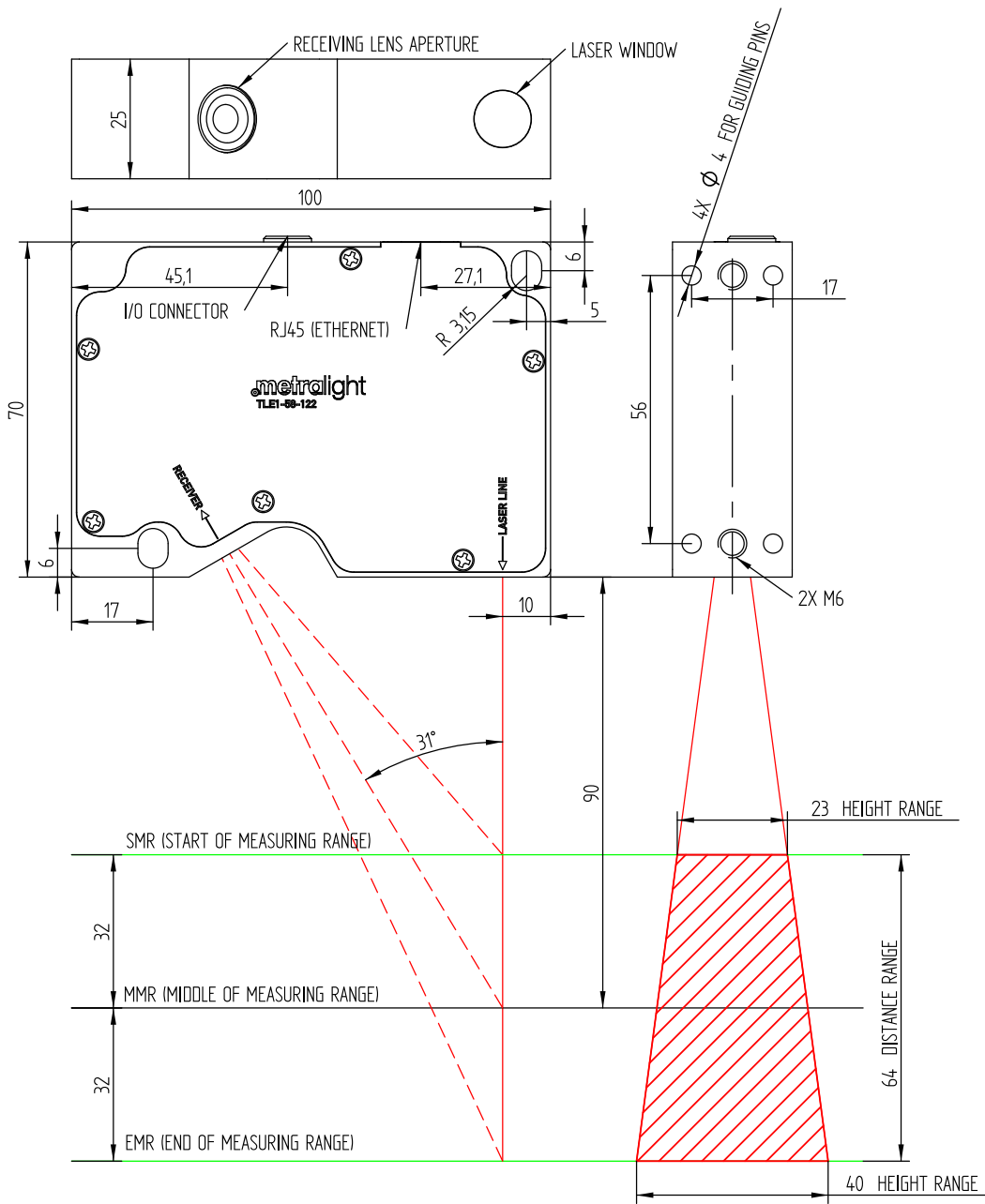


Image 2: TLE1-58-122 sensor dimensions (mm)



## 2.3 Dimensions (TLE1-100-300)

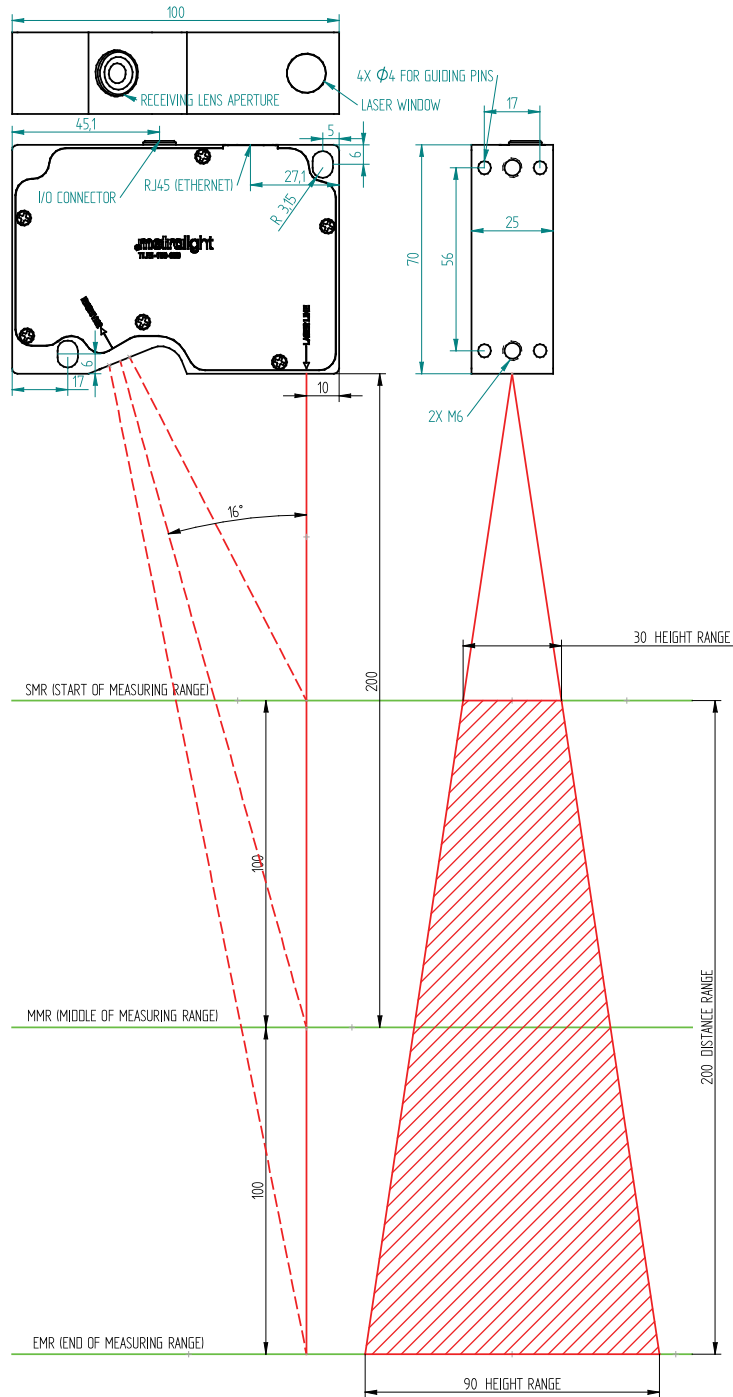


Image 3: TLE1-100-300 sensor dimensions (mm)

## 2.4 Interface

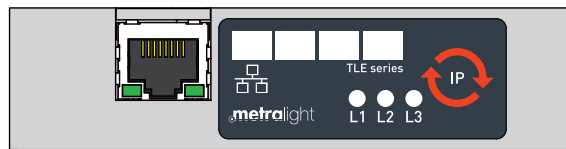
### 2.4.1 RJ45 (ethernet + power)

Ethernet communication is provided over RJ45 connector. This connector can be also used for powering of the sensor (Passive POE), so extra power cable can be eliminated.

User can change Ethernet setting from TL Studio SW. Sensor communicates via TCP protocol with two available sockets - Active (port 1024) and Passive (port 1028). Active socket is used as main communication channel, passive socket is used as monitor (changing of sensor parameters is not possible). TLE1 Sensor factory setting:

IP	192.168.0.16
Gateway	192.168.0.255
Subnet mask	255.255.255.0

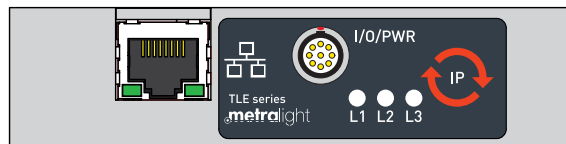
*Table 2: Factory IP setting*



*Image 4: Back panel with RJ45*

### 2.4.2 I/O/PWR connector (optional)

Sensor can be optionally equipped with I/O/PWR connector - digital inputs (measurement triggering or encoder signal) and push-pull or open collector outputs (synchronization signal, MIN/MAX limits comparison result) and Power connection.



*Image 5: Back panel with RJ45 and I/O/PWR*

## Connector specification:

Number of pins	9
Manufacturer	Touch Technology (www.touch-sz.com)
Mnf. Part #	0BTZN09P
Mating connector Part #	0BTTN09P60 (Straight), 0BWTN09P60 (Right angle)
Cable	LAPP KABEL 0028210 CABLE, LIYY, 10CORE, 0.14mm

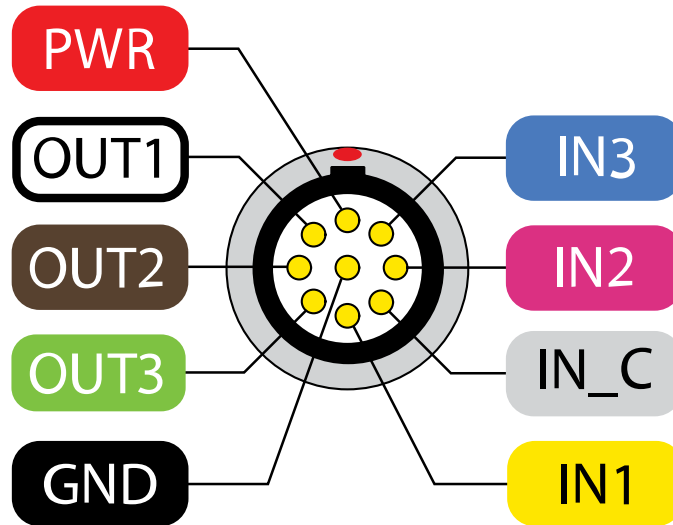


Image 6: Detail of I/O/PWR connector

### 2.4.3 IP reset button

When IP reset button is pressed during power-up sequence IP address is temporarily set to 192.168.0.15. Previous IP setting saved in EEPROM will not be changed and will be used in next power-up if IP button is not pressed. During normal operation IP reset button has additional functionality (e.g. measurement trigger).

### 2.4.4 L1 | L2 | L3 LEADS

Leds L1, L2, L3 have different functions:

- L1 indicates sensor power
- L2 indicates object in sensor measuring range
- L3 user defined LED (see [I/O Config \(page 33\)](#))

## 2.5 Measuring range

See image below to understand how the sensor captures and handles the measured range. Please note, that the actual dimensions and measuring range is different for TLE1-35-65 and TLE1-58-122.

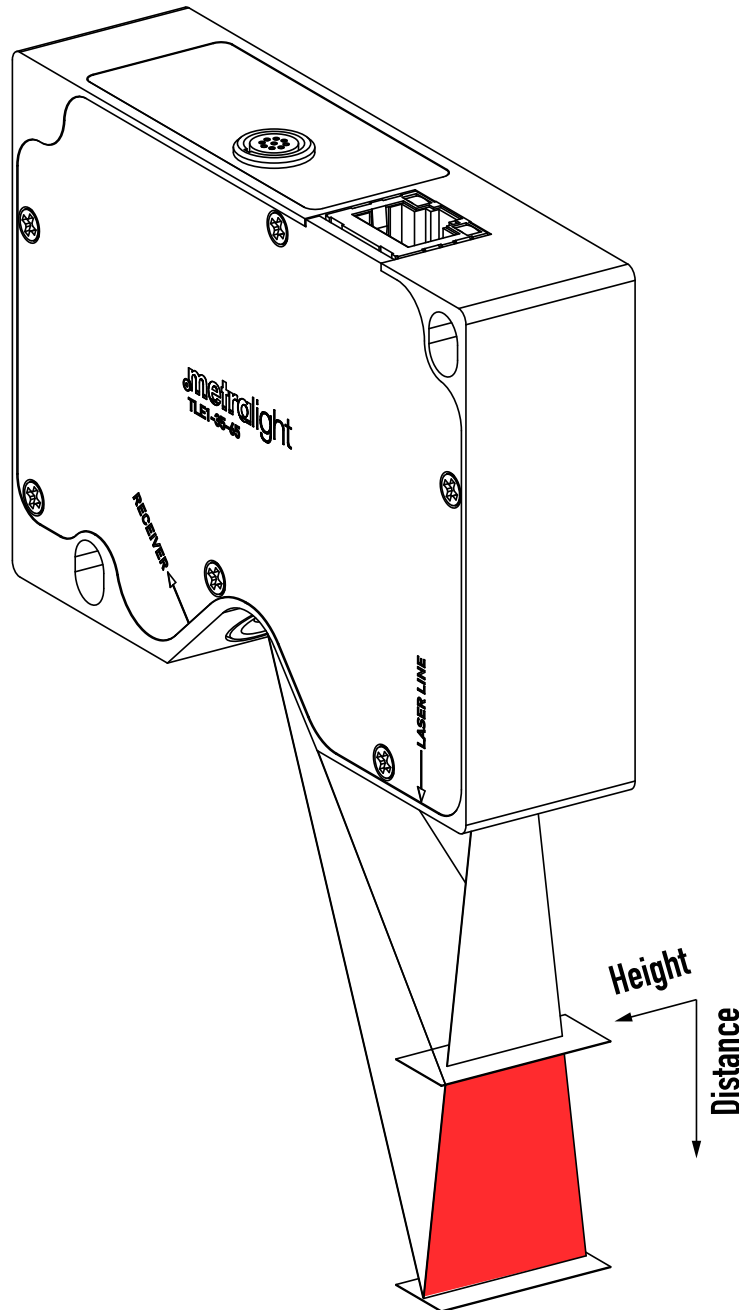


Image 7: Measuring range visualization

## 2.6 Framerate

Sensor framerate depends on the size of active window on image sensor (see [Image parameters \(page 21\)](#))

$$FPS = \frac{48 \cdot 10^6}{(\text{NUMRW\_WIN} + 26) \cdot (\text{NUMCL\_WIN} + 234)}$$

*Formula 1: Sensor framerate*

## 2.7 Firmware

Firmware of the TLE1 sensor can be uploaded via TL Studio SW.

## 3. Command set

### 3.1 DATA

Hex: <0x1X>, where X specifies amount of requested data.

Sensor response:  $2^X \times \text{DATA}$  - see chapter [Data format \(page 18\)](#) for more details.

**Example 1 (valid when extended data format is off - see chapter [Data format \(page 18\)](#))**

```
PC request:          <0x10> (request for 1 data)
Sensor response:    <0x13> <0xDF> <0x00> <0xF9> <0x85>
                   // X Data = 0x13DF = 5087 μm (distance)
                   // Y Data = 0x00F9 = 249 μm (height)
```

**Example 2 (valid when extended data format is off - see chapter [Data format \(page 18\)](#))**

```
PC request:          <0x12> (request for 4 data)
Sensor response:    4×5 bytes (see above)
```

### 3.2 DATA\_STREAM\_start

Hex: <0x21>, continuous DATA stream start

Sensor response: measured data stream

See chapter [Data format \(page 18\)](#) for info about returned data.

### 3.3 DATA\_STREAM\_stop

Hex: <0x20>, continuous DATA stream stop

Sensor response: no response

See chapter [Data format \(page 18\)](#) for info about returned data.

### 3.4 DISTANCE\_PROFILE\_READ

Hex: <0x6X>, where X specifies number of requested profiles

Sensor response:  $2^X \times \text{DISTANCE PROFILE}$  - multiple of  $(4 \times \text{NUM\_LINES} + 2)$  bytes packet

PROFILE FORMAT:

```
<DIST_1_HIGH> <DIST_1_LOW> <HEIGHT_1_HIGH> <HEIGHT_1_LOW> <DIST_2_HIGH> <DIST_2_LOW>
<HEIGHT_2_HIGH> <HEIGHT_2_LOW> ... <DIST_N_HIGH> <DIST_N_LOW> <HEIGHT_N_HIGH> <HEIGHT_N_LOW>
<PROFNUM_HIGH> <PROFNUM_LOW>
```

When multiple profile command is sent and no digital input is connected to encoder channel A, then PROFNUM sequentially

increases (+1). When encoder channel A is connected (see chapter [Input connections \(page 35\)](#)), PROFNUM has meaning of 16 bit position counter. Channel A rising edge changes counter value by 1, channel B select counter direction up/down.

## 3.5 FIRMWARE

Hex: <0xF0>, reads firmware version

Sensor response: two bytes

## 3.6 READ\_RAW

Hex: <0xD0>, read full image (1280 × 1024)

Sensor response: 1310720 pixels (1 byte per pixel - grayscale image)

## 3.7 READ\_RAW\_1/4

Hex: <0xD1>, read downsized image (640 × 480)

Sensor response: 327680 pixels (1 byte per pixel - grayscale image)

## 3.8 MODE

Hex: <0x3X>, to switch to mode X

Sensor response: <0x3X>

## 3.9 LASER\_ON/OFF

Hex: <0x91> for Laser ON, <0x90> for Laser OFF

Sensor response: <0x91> / <0x90>

## 3.10 EXTENDED\_DATA\_ON/OFF

Hex: <0x99> for extended data ON, <0x98> for extended data OFF

Sensor response: <0x99> / <0x98>

## 3.11 SET\_AUTO\_EXPO

Hex: <0x93> set optimal exposure (parameter TINT), sensor will return calculated value in two bytes in 300 ms at most

Sensor response: <RESULT\_HIGH\_BYTE> <RESULT\_LOW\_BYTE>

## 3.12 SET\_BANK\_PARAMETERS

Hex: <0x4X> to load parameters from EEPROM bank number X (cached). Data from EEPROM are cached only during sensor start-up sequence

Sensor response: <0x4X>

## 3.13 USER\_EEPROM\_WRITE

Hex: <0xB0> <ADDR\_HIGH> <ADDR\_LOW> <N\_BYTES-1> <BYTE\_1> ... <BYTE\_N>

Sensor response: <N\_BYTES-1>

ADDR range: 0 ÷ 65535

N\_BYTES range: 1 ÷ 256

This command can write data in one memory page only. It's not possible to write data to more pages at once.

### Example 1

Parameters to write:	START_LINE	0x0180	= 384
	NUMBER_LINES	0x0100	= 256
	SUBSEGMENT,SEGMENT	0x0020	= 32
	THRESHOLD	0x0060	= 90
	TINT	0x0419	= 1049
	LSRPWR	0x0080	= 128

PC request: <0xB0> <0xFE> <0x00> <0x0B> <0x01> <0x80> <0x01> <0x00>  
<0x00> <0x20> <0x00> <0x60> <0x04> <0x19> <0x00> <0x80>

Sensor response: <0x0B>

### Example 2

IP parameters to write:	IP	192.168.0.16
	GATEWAY	192.168.0.255
	SUBNET MASK	255.255.255.0

PC request: <0xB0> <0xFF> <0x00> <0x0B> <0xC0> <0xA8> <0x00> <0x10>  
<0xFF> <0xFF> <0xFF> <0x00> <0xC0> <0xA8> <0x00> <0xFF>

Sensor response: <0x0B>

## 3.14 USER\_EEPROM\_READ

Hex: <0xA0> <ADDR\_HIGH> <ADDR\_LOW> <N\_BYTES-1>

Sensor response: <N\_BYTES-1> requested bytes

ADDR range: 0 ÷ 65535

N\_BYTES range: 1 ÷ 256



**Example 1**

Read sensor parameters:

PC request: <0xA0> <0xFE> <0x00> <0x0B>

Sensor response: <0x01> <0x80> <0x01> <0x00> <0x00> <0x20> <0x00> <0x60>  
<0x04> <0x19> <0x00> <0x80>

**Example 2**

Read IP parameters:

PC request: <0xA0> <0xFF> <0x00> <0x0B>

Sensor response: <0xC0> <0xA8> <0x00> <0x10> <0xFF> <0xFF> <0xFF> <0x00>  
<0xC0> <0xA8> <0x00> <0xFF>

## 3.15 REGISTER\_WRITE

For writing volatile sensor parameters - time of integration, laser power, etc.

Hex: <0x0D> <ADDR\_HIGH> <ADDR\_LOW> <PARAM\_HIGH\_BYTE> <PARAM\_LOW\_BYTE>

Sensor response: <0x0D>

Parameters with initial value stored in EEPROM have ADDR\_HIGH=0x00 and ADDR\_LOW is same as for EEPROM.

**Example 1**

Parameters to write: TINT - address <0x09> - value 0x012C (300)

PC request: <0x0D> <0x00> <0x09> <0x01> <0x2C>

Sensor response: <0x0D>

## 3.16 REGISTER\_READ

For reading volatile sensor parameters - time of integration, laser power, etc. Addresses of parameters are the same as for eeprom - use low byte of eeprom address.

Hex: <0x0C> <ADDR\_HIGH> <ADDR\_LOW>

Sensor response: <PARAM\_HIGH\_BYTE> <PARAM\_LOW\_BYTE>

**Example 1**

Parameters to read: TINT - address 0x09

PC request: <0x0C> <0x00> <0x09>

Sensor response: <0x01> <0x2C> (TINT is 300)

## 4. Data format

Sensor is able to return data in two formats - standard and extended. This was made to remain possible to use existing client software with newer versions of sensor, which is - compared to older versions - equipped with more measuring modes where some of them return more points at once (for example Mode 7 - Object position returns top edge, bottom edge, center point). Older versions of sensor were equipped with measuring modes returning only one point.

Currently used data format depends on whether sensor have extended format turned on using [EXTENDED\\_DATA\\_ON \(page 15\)](#) commad.

### Standard data format contains one point and aux byte:

<POINT1> <AUX BYTE>

### Extended DATA contains four points:

<POINT1> <POINT2> <POINT3> <POINT4> <EXTAUX\_BYTE> <AUX\_BYTE>

### Each point consists of two coordinates, where each of them occupies two bytes:

<POINT\_X\_HIGH> <POINT\_X\_LOW> <POINT\_Y\_HIGH> <POINT\_Y\_LOW>

AUX_BYTE format		
Bit #	Bit name	Description
7 (MSB)	OIN	Object detected in measurement range
6	ZERO_CNT	Indicates Line in processing window which has 0 pixels over threshold
5	OVER410_CNT	Indicates Line in processing window with signal width>410 pixels
4	0	unused
3	USER_PAR_CHG	User's parameters changed after power-up
2...0	MODES	Measuring mode, see chapter <a href="#">Measuring modes (page 26)</a>

Table 3: AUX\_BYTE format

Meaning of specific points in returned data is available in [Measuring modes \(page 26\)](#) chapter.

# 5. EEPROM memory map

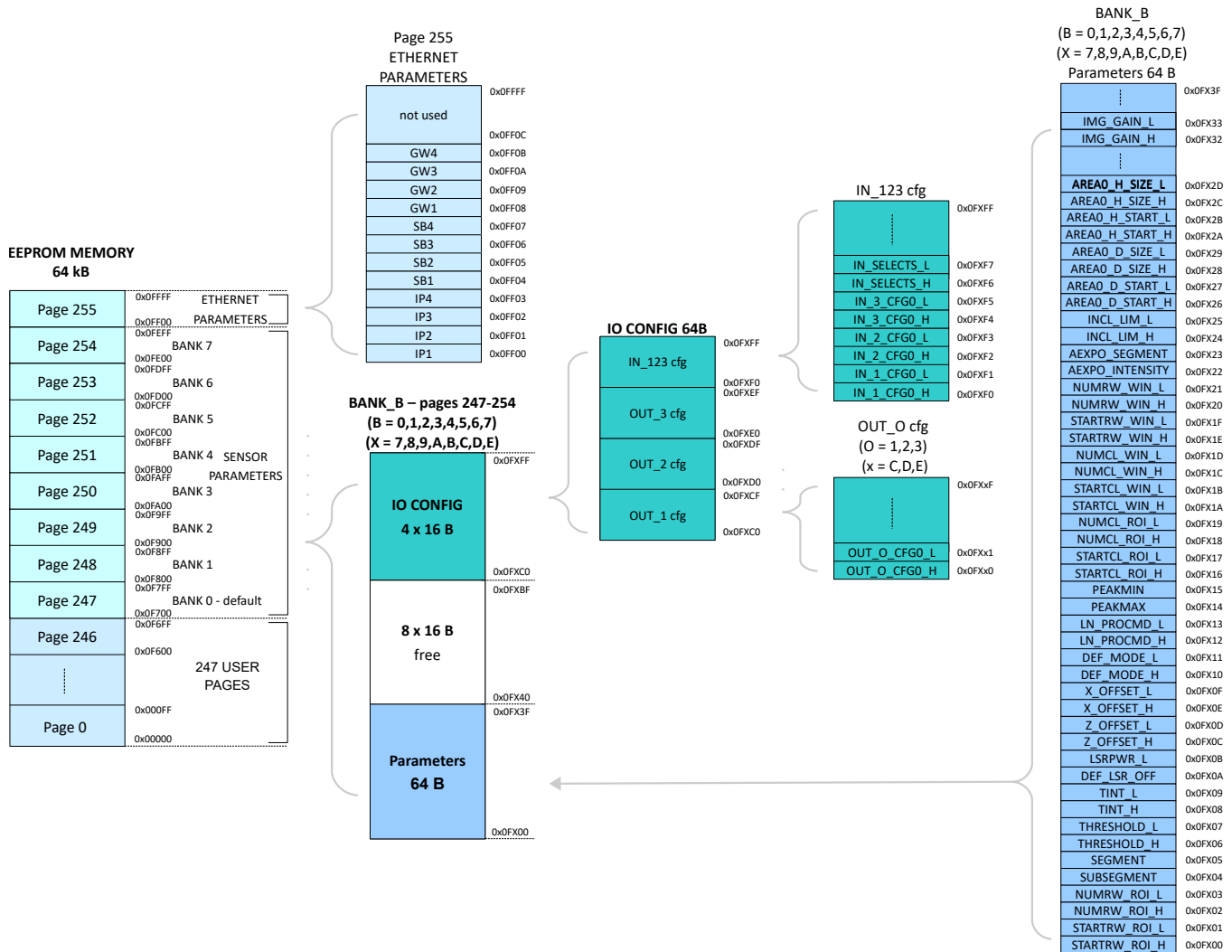


Image 8: Sensor memory structure

Sensor is equipped with 64kb EEPROM memory. Memory is divided into 256 pages (numbered 0-255) with size 256B. Pages 0-246 are user pages. Pages 247-254 are sensor parameter banks (IO config + parameters). Page 255 is used for ethernet interface settings.

Each sensor parameter bank (247-254) consists of IO config, sensor imaging parameters and currently unused space.

IO configuration occupies 64 bytes. Inputs are configured in 16 bytes. Remaining 48 bytes are used for Output config.

Sensor parameters occupies 64 bytes. Most of parameters consists of two bytes, a few occupies only 1 byte. Complete list of parameters is visible in the image above and is also described in chapter [Sensor parameters \(page 21\)](#).

# 6. Registers map

Besides sensor parameters which are present both in volatile registers and EEPROM map (volatile parameters, which has its initialization value in eeprom - same addressing for volatile registers and eeprom), there are some volatile registers which are not present in EEPROM banks.

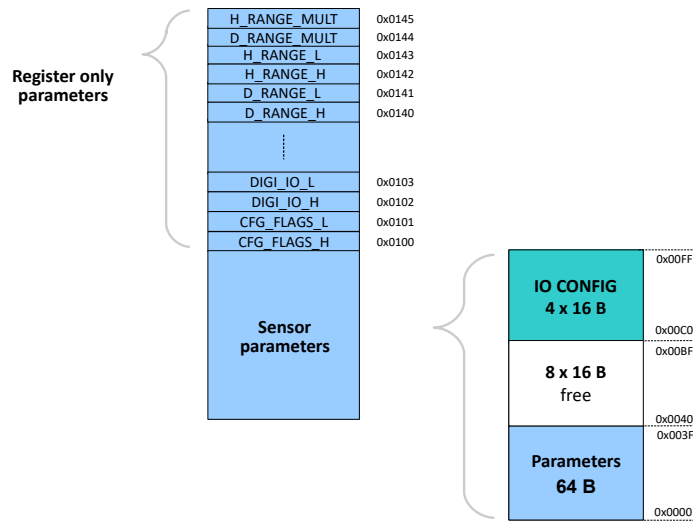


Image 9: Register map

## 7. Sensor parameters

This chapter describes sensor parameters which can adjust measuring, image processing and workflow. Changes of volatile parameters (see chapter [Command set \(page 14\)](#)) have immediate effect to sensor behavior. Set of sensor parameters can be saved in one EEPROM bank. Total 8 EEPROM banks are available (BANK 0-7). BANK\_0 contains default setting used on power-up. User can switch between banks to quickly change sensor parameters - see [EEPROM memory map \(page 19\)](#).

### 7.1 Image

Name	Description	Unit	Range
TINT	time of integration	-	1-1049
IMG_GAIN	image gain	-	1-15
LSRPWR	laser power	-	0-255
DEF_LSR_OFF	laser off	-	0   1
<b>Active window on image sensor (affect sensor frame rate, see <a href="#">Framerate (page 13)</a>)</b>			
STARTRW_WIN	first row of window	pixel	0-1020
NUMRW_WIN	number of rows of window	pixel	3-1024
STARTCL_WIN	first column of window	pixel	0-680, only even
NUMCL_WIN	number of columns of window	pixel	600-1280, only even

*Table 4: Image sensor parameters*

## 7.2 Profile

Name	Description	Unit	Range
THRESHOLD	minimal pixel value for peak detection on the line	light intensity	1-255
LN_PROC_MODE	Mode of peak selection on the line: 0 - ALL OVER THR 1 - LAST PEAK (CLOSEST TO SENSOR) 2 - FIRST PEAK (FURTHEST FROM SENSOR) 3 - BIGGEST PEAK	-	0   1   2   3
PEAK_MIN	minimal valid peak width	pixel	1-255
PEAK_MAX	maximal valid peak width	pixel	1-255
<b>Region Of Interest (ROI) definition (ROI specifies area in raw image for Profile computation), ROI has to be inside active window of image sensor</b>			
STARTRW_ROI	first row of ROI	pixel	1-1022
NUMRW_ROI	number of rows of ROI	pixel	1-1022
STARTCL_ROI	first column of ROI	pixel	1-1278
NUMCL_ROI	number of columns of ROI	pixel	1-1278

*Table 5: Profile sensor parameters*

## 7.3 Profile processing

Name	Description	Unit	Range
SEGMENT	segment size for profile filtering	pixel	1-255
SUBSEGMENT	required continuity of profile, higher number = lower tolerance for missing profile points	pixel	SUBSEGMENT<SEGMENT
DEF_MODE	measuring mode (see <a href="#">Measuring modes (page 26)</a> )	-	0-8
INCL_LIM	inclination limit for object search	-	0-255
<b>Profile processing area definition. *</b>			
AREA0_D_START	first row of area	µm	0-D_RANGE
AREA0_D_SIZE	number of rows of area	µm	0-D_RANGE
AREA0_H_START	first column of area	µm	0-H_RANGE
AREA0_H_SIZE	number of columns of area	µm	0-H_RANGE

Table 6: Profile processing parameters

\* When using long-range sensor TLE1-100-300, parameters need to be divided using height and distance range multipliers. More info in chapter [Register only parameters \(page 24\)](#).

## 7.4 Other parameters

Name	Description	Unit	Range
AEXPO_SEGMENT	flatness of measured surface (recommended value is 64)	-	2-255
AEXPO_INTENSITY	required exposure level for SET_AUTO_EXPO command (recommended value is 16)	-	1-64
USER_PAR1	user value, which doesn't affect sensor behaviour (can be also used for firmware customization)		
USER_PAR2	user value, which doesn't affect sensor behaviour (can be also used for firmware customization)		

Table 7: Other sensor parameters

## 7.5 Register only

Following parameters are available only as volatile registers.

Name	Description	Read/write	Unit	Range
H_RANGE_MULT	Sensor height range multiplier	Read only	-	-
D_RANGE_MULT	Sensor distance range	Read only	-	-
H_RANGE	Sensor height range (real height measuring range is always smaller than H_RANGE (to avoid negative numbers)). Multiply by H_RANGE_MULT to get value in $\mu\text{m}$ .	Read only	$\mu\text{m}$	-
D_RANGE	Sensor distance range. Multiply by D_RANGE_MULT to get value in $\mu\text{m}$ .	Read only	$\mu\text{m}$	-
DIGI_IO	inputs/outputs state (read/write)	R/W	-	0   1
CFG_FLAGS	configuration flags	Read only	-	0   1

*Table 8: Register only parameters*

H\_RANGE\_MULT and D\_RANGE\_MULT multipliers are primarily intended for use with long-range sensor TLE1-100-300, where numbers can overflow two-byte fields. Values read using DATA and DISTANCE\_PROFILE\_READ have to be multiplied by height or distance range multiplier respectively to upscale to  $\mu\text{m}$ . Also coordinates of profile processing area have to be divided by respective multiplier before writing area parameters into the sensor.

More information about DIGI\_IO register can be found in chapter [I/O read/write \(page 37\)](#).



CFG\_FLAGS register has following structure:

Address	0x0100							0x0101								
Param	CFG_FLAGS_H							CFG_FLAGS_L								
Bit #	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit usage	unused						CFG_CHNG	EXT_DAT	unused					RST_IP	PROF_ON_IMG	LSR

*Table 9: Input read/write*

CFG\_CHNG bit: '1' if some sensor parameter was changed, '0' otherwise

EXT\_DAT bit: '1' if extended data format is turned on, '0' otherwise

RST\_IP bit: '1' if the sensor was turned on with IP button pressed, '0' otherwise

PROF\_ON\_IMG: '1' if profile type is set to be displayed on image (values are recalculated), '0' otherwise

LSR bit: '1' laser on, '0' laser off

## 8. Measuring modes

TLE1 sensor can measure distance, edge position, profile, gap size and other shape profiles. TLE1 sensor has up to 3 inputs and 3 outputs (push-pull or NPN/PNP open collector). Inputs can be configured for motor encoder connection. Sensor data can be matched with encoder position for precise 3d shape scanning (please contact Metralight for details). Custom modes can be implemented upon customer request.

Standard DATA command response gives one DISTANCE number and one HEIGHT number. Each of them in two bytes (numbers in range 0 to 65535). These two numbers represents point in sensor measuring plane given by the laser line beam (one axis is distance from the sensor, second axis is along the laser line - see image [Measuring range \(page 12\)](#)). Measuring mode specifies what point to find on measured object (and send as response on data command).

In standard data format sensor returns only Point 1 as a response. In extended data format sensor returns all points. See [Data format \(page 18\)](#) for more information about data formats.

## 8.1 MODE 0 - Mean Distance

MODE 0 is intended for measuring of distance from basic plane. Plane can be straight or curved. Result data is average distance of the number of points - user can set the length of processed line (parameters ①② in figure - user defined ROI - see parameters STARTRW, NUMRW). Measurement range is between SMR (start of measuring range) and EMR (end of measuring range). SMR position is represented by sensor as 0, EMR is represented as 30000 (30mm) for TLE1-35-65 or 64000 (64mm) for TLE1-58-122. SMR at 35mm for TLE1-35-65 and 58mm for TLE1-58-122. Measured area can be reduced by ROI or PROFILE\_AREA settings (see above).

**Points meaning:**

Point 1 - Distance - average distance (computed from laser line points on object selected by the ROI)

Point 1 - Height - center of selected ROI (just for reference, for M0 only distance value is important)

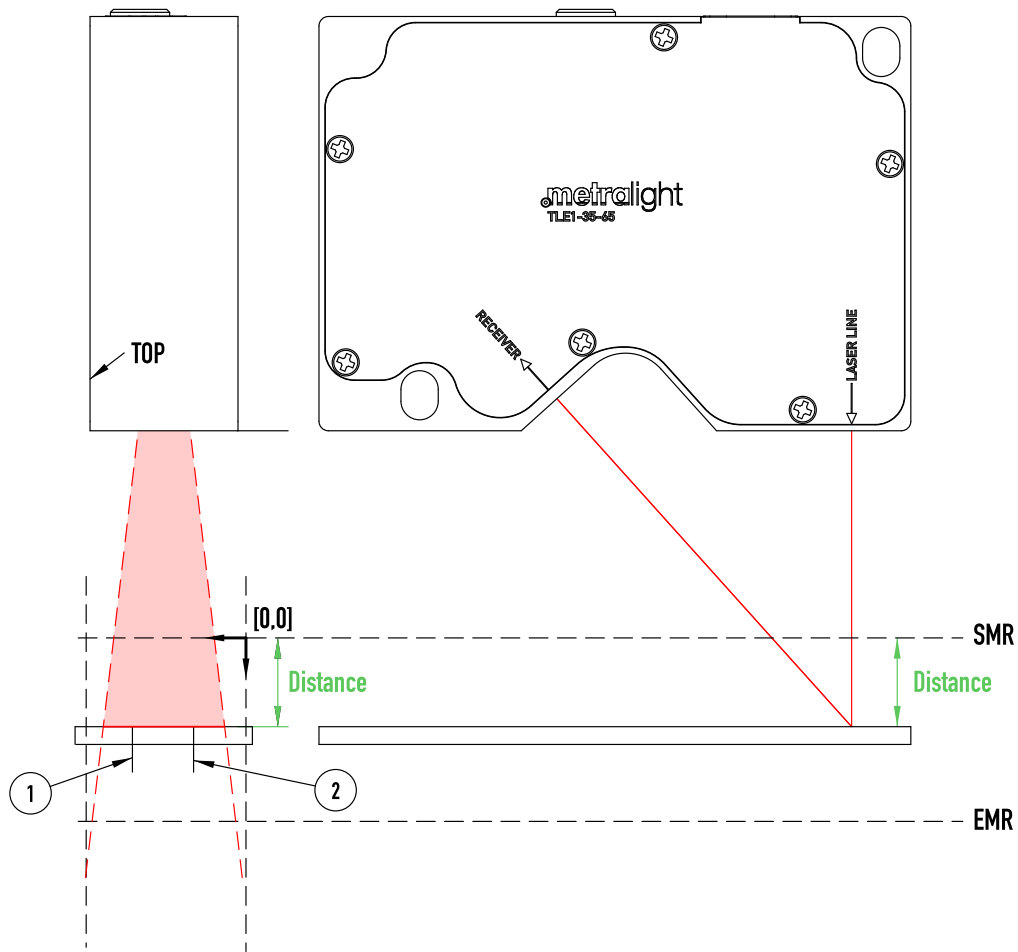


Image 10: MODE 0 example

## 8.2 MODE 1 - Closest Segment

In this mode, sensor finds closest part of the object to the sensor. Size of this object (part of the laser line) is given by the SEGMENT parameter.

**Points meaning:**

Point 1 - coordination of closest found segment

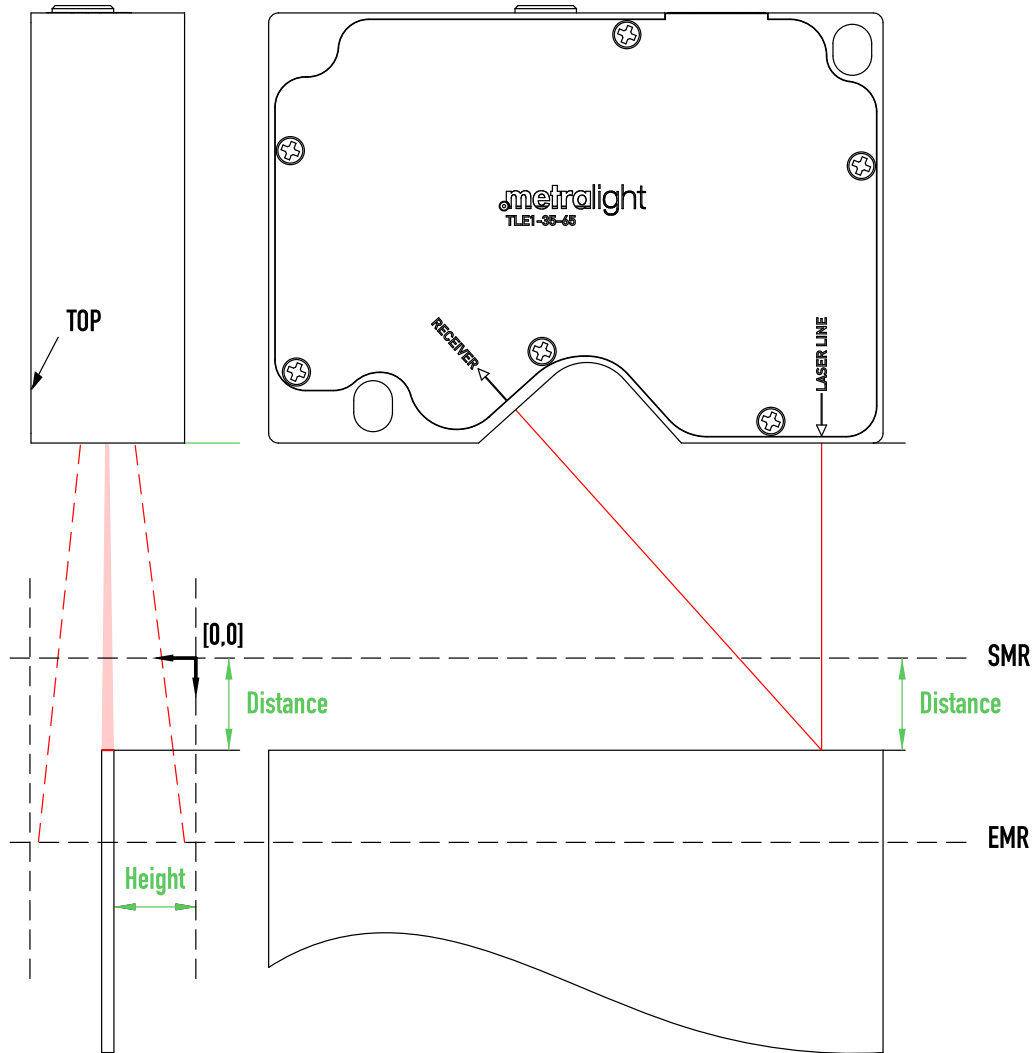


Image 11: MODE 1 example

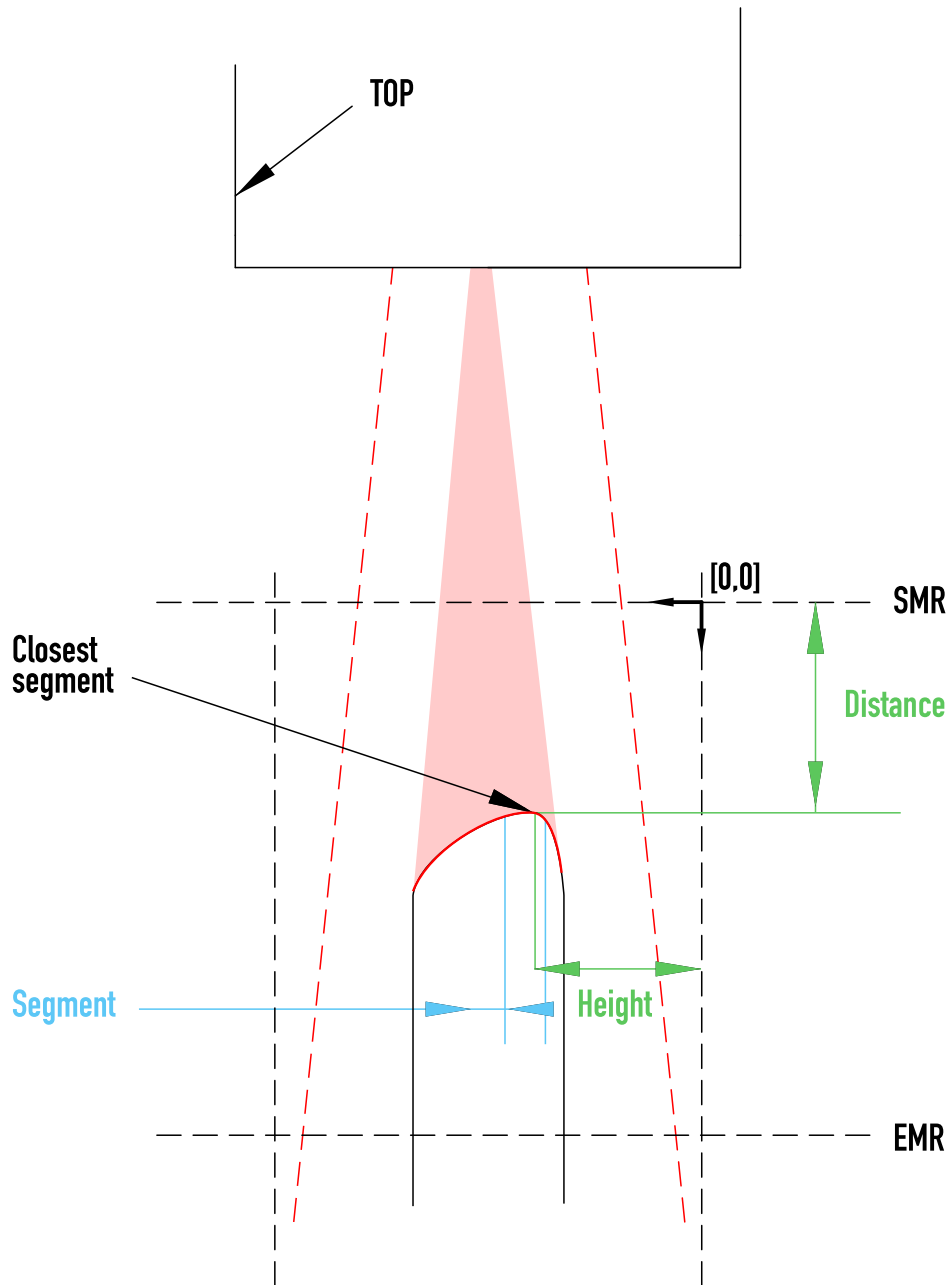


Image 12: MODE 1 detailed view

## 8.3 MODE 2 - First Light Segment

MODE 2 is primary intended for measuring of object EDGE. Measured object has to be presented from sensor top side. SEGMENT parameter specifies the portion of projected line, used for calculating.

### Points meaning:

Point 1 - coordination of first light point

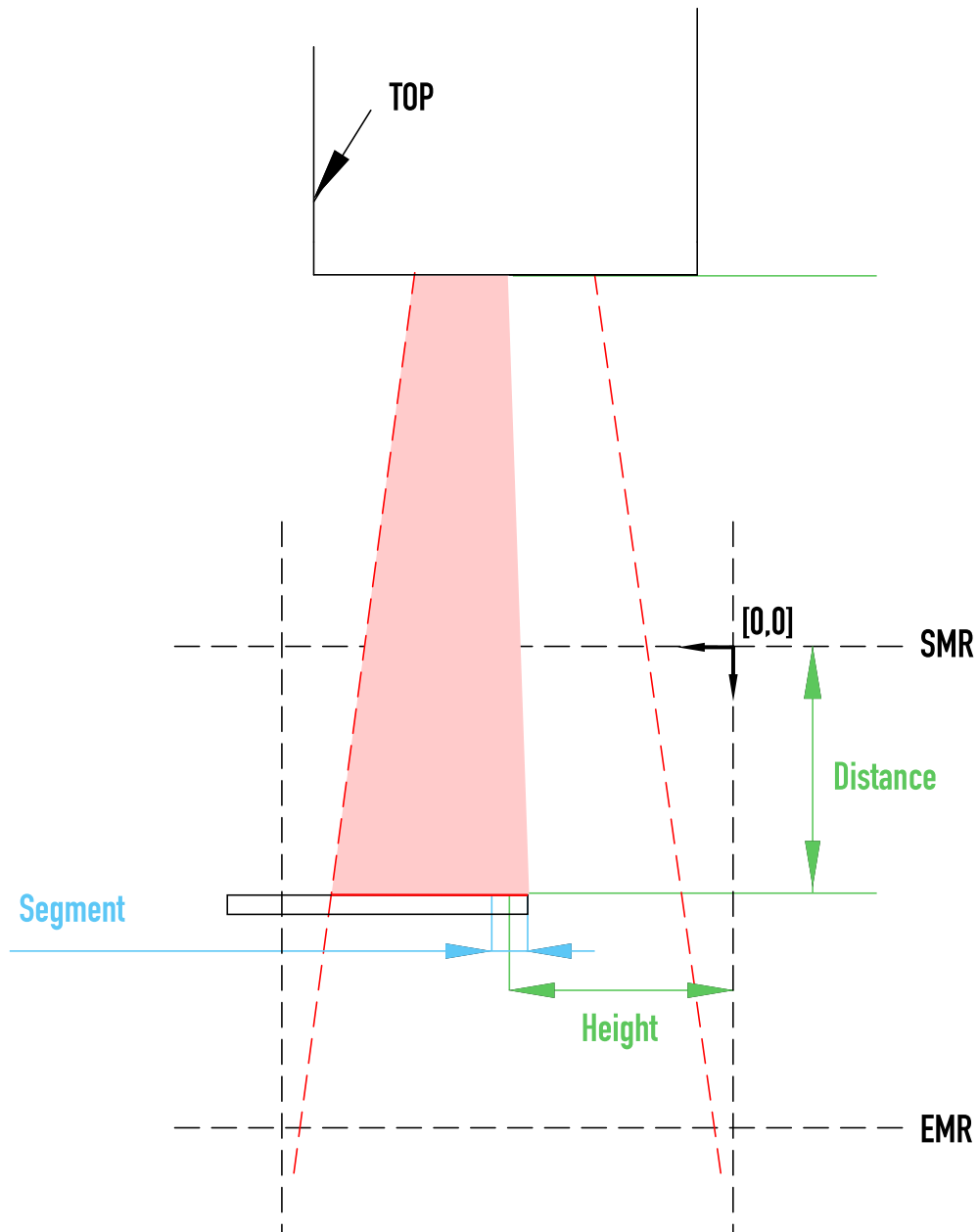


Image 13: MODE 2 example

## 8.4 MODE 3 - Last Light Segment

Same as MODE 2, but object is inserted from the bottom side of the sensor.

### Points meaning:

Point 1 - coordination of last light point

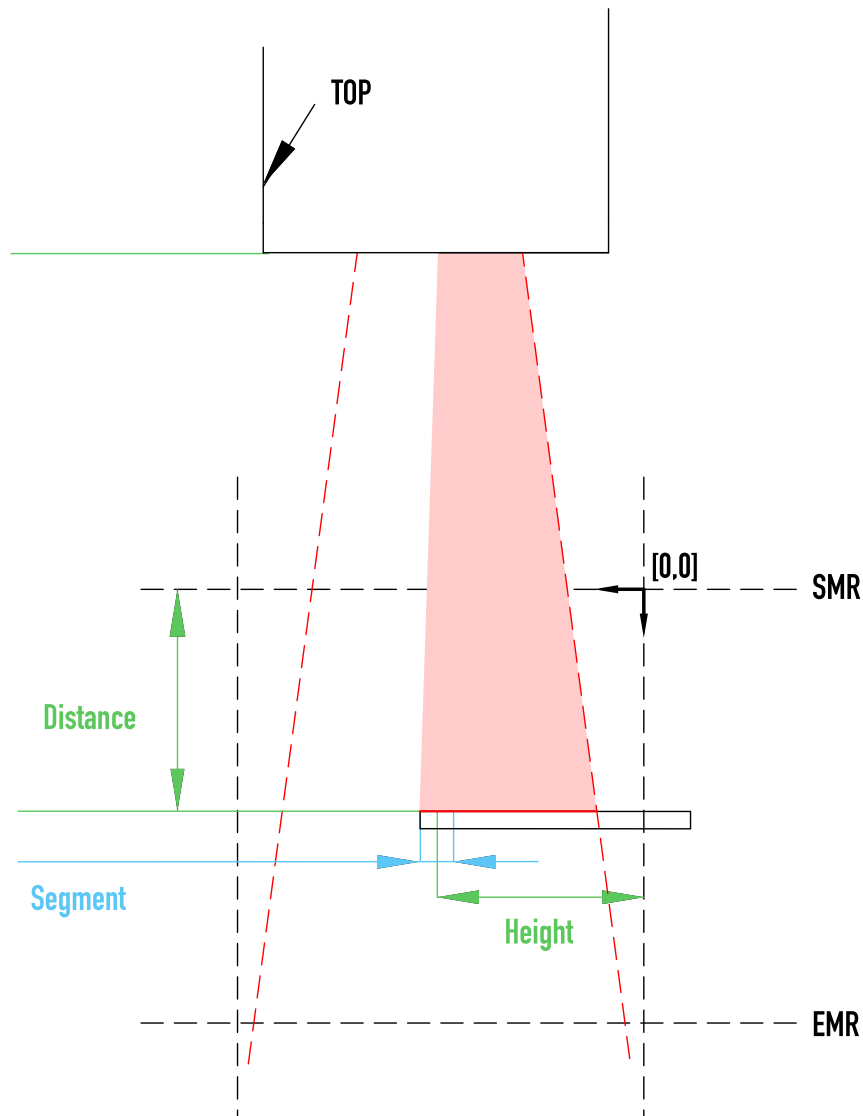


Image 14: MODE 3 example

## 8.5 MODE 4 - Furthest Segment

Same as mode M1, but algorithm finds furthest part of laser line from the front side of the sensor (with biggest distance from the

sensor).

**Points meaning:**

Point 1 - coordination of furthest found segment

## 8.6 MODE 5 - Gap Position

Biggest GAP between objects.

**Points meaning:**

Point 1 - coordination of center of biggest continuous gap (computed as average from edge points)

Point 2 - coordination of first edge

Point 3 - coordination of second edge

Point 4 - height of found gap (distance = 0)

## 8.7 MODE 6 - Gap Height

Biggest GAP between objects. Same as mode 5, but has different order of returned points. This matters mostly in standard data mode, where only Point 1 is returned.

**Points meaning:**

Point 1 - height of found gap (distance = 0)

Point 2 - coordination of first edge

Point 3 - coordination of second edge

Point 4 - coordination of center of biggest continuous gap (computed as average from edge points)

## 8.8 MODE 7 - Object Position

Biggest object, which means biggest continuous segment (number of lines) with light intensity exceeding required threshold.

**Points meaning:**

Point 1 - coordination of center of biggest continuous object (computed as average from edge points)

Point 2 - coordination of first edge

Point 3 - coordination of second edge

Point 4 - height of object (distance = 0)

## 8.9 MODE 8 - Object Height

Biggest object, which means biggest continuous segment (number of lines) with light intensity exceeding required threshold. The mode is the same as mode 7, but has different order of returned points. This matters mostly in standard data mode, where only Point 1 is returned.

**Points meaning:**

Point 1 - height of object (distance = 0)

Point 2 - coordination of first edge

Point 3 - coordination of second edge

Point 4 - coordination of center of biggest continuous object (computed as average from edge points)



# 9. Digital I/O

## 9.1 Output configuration

Output configuration occupies two bytes and is present in every memory bank (0-7). In the tables below X represents the address of the bank (7, 8, 9, A, ... E): For BANK\_0 X = 7, for BANK\_1 X = 8, ... , for BANK\_7 X = E.

There are three outputs available. In the tables below, x represents the output address which is C for input 1, D for input 2 and E for input 3.

Address	0x0FXx0 (X = 7, 8, 9, A, B, C, D, E; x = C, D, E)								0x0FXx1 (X = 7, 8, 9, A, B, C, D, E; x = C, D, E)							
Param	OUT_X_CFG0_H (X = 1, 2, 3)								OUT_X_CFG0_L (X = 1, 2, 3)							
Bit #	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name	INV	0	0	0	0	0	0	0	TYPE		0	0	SRC_SEL			
Meaning	enable output inverting	unused						output type config		unused		output source selection				

Table 10: Output configuration

TYPE1 and TYPE0 bits defines output type. See table below.

TYPE1	TYPE0	OUT type
0	0	NC *
0	1	NPN
1	0	PNP
1	1	PUSH/PULL

Table 11: Output type configuration

\* NC = not connected

SRC\_SEL3, SRC\_SEL2, SRC\_SEL1 and SRC\_SEL0 bits defines output source. See table below.

SRC_SEL3	SRC_SEL2	SRC_SEL1	SRC_SEL0	output	description
0	0	0	0	0	
0	0	0	1	SYNC_OUT	synchronization with internal sensor clock
0	0	1	0	OBJ_IN	object in signal
0	0	1	1	USER_OUT	command defined value
0	1	0	0	IN1	digital IN connected to output
0	1	0	1	IN2	digital IN connected to output
0	1	1	0	IN3	digital IN connected to output
0	1	1	1	0	unused
1	0	0	0	0	unused
1	0	0	1	0	unused
1	0	1	0	0	unused
1	0	1	1	0	unused
1	1	0	0	0	unused
1	1	0	1	0	unused

Table 12: Output source selection

## 9.2 Input configuration

Input configuration occupies two bytes and is present in every memory bank (0-7). In the tables below X represents the address of the bank (7, 8, 9, A, ... E): For BANK\_0 X = 7, for BANK\_1 X = 8, ... , for BANK\_7 X = E.

There are three inputs available, each occupies two bytes. For IN\_1, x = 0 (LOW\_BYTE), 1 (HIGH\_BYTE); for IN\_2, x = 2, 3; for IN\_3, x = 4, 5.

Address	0x0FXFx (X = 7, 8, 9, A, B, C, D, E; x = 0, 2, 4)								0x0FXFx (X = 7, 8, 9, A, B, C, D, E; x = 1, 3, 5)							
Param	IN_O_CFG0_H (O = 1, 2, 3)								IN_O_CFG0_L (O = 1, 2, 3)							
Bit #	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name	0	0	HOLD	INV	0	0	0	0	FILTER							
Meaning	0	0	enable hold function	enable input inverting	0	0	0	0	Time constant of digital input filter							

*Table 13: Input configuration*

**FILTER:** Time constant of digital input filter (signal has to be stable for  $(Filter+1) \times 5 \mu s$ ).

**HOLD:** Enable hold '1' function of input until reading.

**INV:** Enable input inverting.

Connection of inputs:

Address	0x0FXF6								0x0FXF7								
Param	IN_SELECTS_H								IN_SELECTS_L								
Bit #	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Bit name	0	0	0	0	0	0	0	0	LSRoff_SEL	LED3_SEL	ECH_A_SEL			ECH_B_SEL			
Meaning	unused											ECH_A - edge counting ECH_B - direction (0 - up, 1 - down)					

*Table 14: Input connections*

Connecting inputs to white user led on sensor back panel (LED3):

LED3_SEL1	LED3_SEL0	
0	0	NC*
0	1	IN1
1	0	IN2
1	1	IN3

*Table 15: Inputs to user LED connection*

Connecting inputs to switch laser on/off:

LSRoff_SEL1	LSRoff_SEL0	
0	0	NC*
0	1	IN1
1	0	IN2
1	1	IN3

*Table 16: Inputs to laser on/off connection*

Position encoder channel 1 and channel 0:

ECH_A_SEL1	ECH_A_SEL0	
0	0	NC*
0	1	IN1
1	0	IN2
1	1	IN3

*Table 17: Position encoder channel 1*

ECH_B_SEL1	ECH_B_SEL0	
0	0	NC*
0	1	IN1
1	0	IN2
1	1	IN3

*Table 18: Position encoder channel 0*

\* NC = not connected

## 9.3 Read/write

State of inputs/outputs can be read/written using parameter in register DIGI\_IO. Register has following structure:

Address	0x0102								0x0103							
Param	DIGI_IO_H								DIGI_IO_L							
Bit #	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read/write	-	-	-	-	-	R	R	R	-	-	-	-	-	RW	RW	RW
Bit usage	unused					IN3	IN2	IN1	unused					OUT3	OUT2	OUT1

*Table 19: Input read/write*

States of the inputs is available for reading in bits IN1, IN2, IN3. Output state is accessible in bits OUT1, OUT2, OUT3. If outputs are configured as USER\_OUT (see table [Output configuration \(page 33\)](#)), their value can be changed by writing value in those bits.

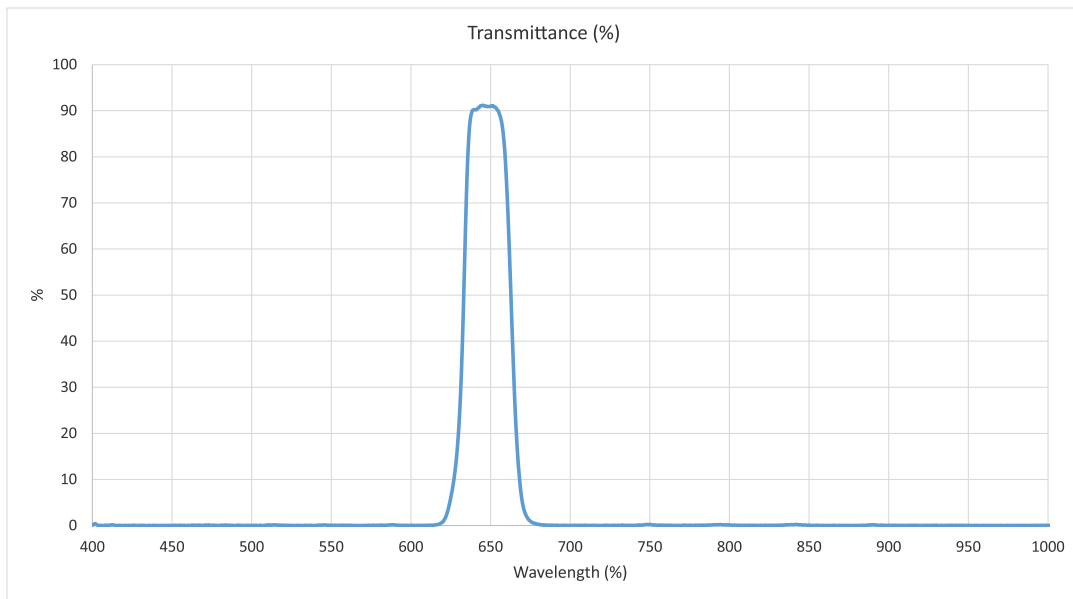
# 10. Installation

USE APPROPRIATE MOUNTING SCREWS (SEE MECHANICAL DRAWING)

AVOID ESPECIALLY DIRECT SUNLIGHT AND ALL OTHER LIGHT SOURCES WITH WAVELENGTH CLOSE TO 650nm (see Optical filter transmittance on figure below).

ALWAYS KEEP OPTICAL WINDOWS CLEAN, FREE FROM DUST AND FINGERPRINTS, AVOID SCRATCHES ON THE OPTICAL WINDOWS.

USE CORRECT VOLTAGE - SEE ELECTRICAL SPECIFICATION



*Image 15: Ambient light optical filter transmittance*

## Laser Safety

TLE1 Sensor is classified as Class 1M Laser device (Laser power < 3mW according to IEC 60825-1 or ANSI Z136.1). A Class 1M laser is safe for all conditions of use except when passed through magnifying optics such as microscopes and telescopes. Class 1M lasers produce large-diameter beams, or beams that are divergent.



*Image 16: Class 1M Laser safety label*

# 11. Code examples

Microsoft C# code example:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Net.Sockets;
using System.IO;
namespace TCPexample
{
    class Program
    {
        static void Main(string[] args)
        {
            TcpClient tcpClient = new TcpClient();
            tcpClient.Connect("192.168.0.15", 1024);
            NetworkStream stream = tcpClient.GetStream();
            stream.Write(new byte[] { 0x10 }, 0, 1); //send data request
            while (tcpClient.Available < 5) { }; //wait for sensor response
            byte[] recievedData = new byte[5]; //read sensor response
            stream.Read(recievedData, 0, 5);
            Console.WriteLine(recievedData[0] * 256 + recievedData[1]); //display result
            tcpClient.Close();
        }
    }
}
```

Java code example:

```
import java.io.BufferedInputStream;
import java.io.BufferedOutputStream;
import java.io.IOException;
import java.net.Socket;
import java.net.UnknownHostException;
public class MainClass {
public static void main(String[] args) throws UnknownHostException, IOException {
    try {
        Socket socket = new Socket("10.226.164.150", 1024);
        BufferedOutputStream out = new BufferedOutputStream(socket.getOutputStream());
        BufferedInputStream in = new BufferedInputStream(socket.getInputStream());
        out.write(new byte[]{16}, 0, 1); //send data request
        out.flush();
        byte[] recievedData = new byte[5];
        in.read(recievedData, 0, 5); //read sensor response
        System.out.println(recievedData[0] * 256 + recievedData[1]); //display result
    }
    catch (Exception e) {
        System.out.println(e.toString());
    }
}
```



Node.js code example:

```
//required modules
var net = require('net');
var Buffer = require('buffer').Buffer;

// init socket and prepare asynchronous data handler
// (close connection on data receive)
var sensorSocket = new net.Socket();
sensorSocket.on('data', function(data){
  var distance = (data[0] * 256 + data[1]);
  var height = (data[2] * 256 + data[3]);
  console.log(data);
  console.log('Distance: ' + distance + ' µm');
  console.log('Height: ' + height + ' µm');
  sensorSocket.destroy();
});

//prepare command
var command = new Buffer(1, 'hex');
command[0] = 0x10;

//connect and write command
sensorSocket.connect(1024, '192.168.0.17');
sensorSocket.write(command);

//output
//<Buffer 4b af 39 1a 80>
//Distance: 19375 µm
//Height: 14618 µm
```

# 12. Package, warranty, contacts

Package components:

- 1x Laser sensor
- Optional:
  - RJ45 CAT5 cable straight or right angle
  - Power injector
  - I/O cable

## Warranty

METRALIGHT provides a ONE YEAR manufacturer's limited warranty against defective materials and workmanship. Please do not attempt to open the unit, as this will void all warranties.

## Contacts

METRALIGHT, Inc.

1670 S. Amphlett Blvd., Unit # 214-M

Mailstop # 1008

San Mateo, CA 94402

phone: (650) 581 3088

fax: (650) 808 9830

email: [sales@metralight.com](mailto:sales@metralight.com)

technical support: [support@metralight.com](mailto:support@metralight.com)

web site: <http://www.metralight.com>