FDRF651 Series

Certified according to ISO 9001:2008

05.2017 | version 20150121 Rev 3.1

Contents

1. Safety precautions

- Use supply voltage and interfaces indicated in the sensor specifications.
- In connection/disconnection of cables, the micrometer power must be switched off.
- Do not use micrometers in locations close to powerful light sources.
- To obtain stable results, wait about 20 minutes after micrometer activation to achieve uniform micrometer warm-up.

2. Electromagnetic compatibility

The micrometers have been developed for use in industry and meet the requirements of the following standards:

- EN 55022:2006 Information Technology Equipment. Radio disturbance characteristics. Limits and methods of measurement.
- EN 61000-6-2:2005 Electromagnetic compatibility (EMC). Generic standards. Immunity for industrial environments.
- EN 61326-1:2006 Electrical Equipment for Measurement, Control, and Laboratory Use. EMC Requirements. General requirements.

3. Laser safety

The micrometers make use LED or c.w. 660 nm wavelength semiconductor laser. Maximum laser output power is <0,2 mW. The micrometers belong to the 1 laser safety class. The following warning label is placed on the laser body.

CLASS 1 LASER PRODUCT

The following safety measures should be taken while operating the sensor:

- Avoid staring into the laser beam during a prolonged time period;
- Do not disassemble the micrometer

4. General information

The micrometers are intended for non-contact measuring and checking of diameters, gaps, displacement/position of the edges of technical objects

The series includes 5 models with the measurement range, from 10 to 98 mm.

5. Basic technical data

FDRF651-	25	50	75	100
Measurement range, mm	25	48	75	98
Minimum size of the object, mm	0.5	1	1.5	2
Accuracy [1], um	±5	±10	±15	±20
Max measurement frequency, Hz	2000	2000	2000	2000
Light source	LED or laser			
Laser safety class	1 (IEC60825-1)			
Output interface — digital	RS232 (max. 921,6 kbit/s) or RS485 (max. 921,6 kbit/s) or Ethernet & (RS32 or RS485)			

	analog	4…20 mA (load ≤ 500 Ohm) or 0…10 V			
Synchronization input		2,4 – 5 V (CMOS, TTL)			
Logic output		three outputs, NPN: 100 mA max; 40 V max			
Power supply, V		24 (9 …36)			
Power consumption, W		1,5..2			
Environ-ment re-sistance		IP67			
	Vibration	20g/10…1000Hz, 6 hours, for each of XYZ axes			
	Shock	30 g / 6 ms			
	Operation temperature, °C	-10…+60			
	Relative humidity	5-95% (no condensation)			
Housing material		aluminum			
Weight (without cable), gram		600	2000	2600	4000

[1] Typical data obtained when a knife edge was used to interrupt the beam and distance between transmitter and receiver is equal of two measurement range

6. Example of item designation when ordering

FDRF651-X/L-SERIAL-ANALOG- LOUT- IN-CC-M-AK

Symbol	Description
X	Measurement range, mm
L	The distance between the transmitter and receiver housings, fixed on the beam, mm
SERIAL	The type of serial interface: RS232 - 232 or RS485 - 485 or (Ethernet and RS232) – 232-ET or (Ethernet and RS485) – 485-ET
ANALOG	Attribute showing the presence of 4…20 mA (I) or 0…10V (U)
LOUT	Attribute showing the presence of 3 logical outputs
IN	Trigger input (input of synchronization) presence
CC	Cable gland - CG, or cable connector - CC
M	Cable length, m
AK	Micrometer with protect air knife for windows

Example: FDRF651-25/50-232-I-IN-CG-3 – measurement range – 25 mm, distance between transmitter and receiver – 50 mm, RS232 serial port, 4…20mA analog output, trigger input is available, cable connector, 3 m cable length.

Permitted modifications:

Model	Valid values for the parameters
FDRF651-25/L-SERIAL-ANALOG-LOUT-IN-CC-M-AK	L – 50mm..100mm (large base under the order) SERIAL – 232, 485, 232-ET, 485-ET ANALOG – no, I, U LOUT – no, LOUT IN - IN CC – CG, CC M – 0,1m..10m AK – no, AK

FDRF651-50/L-SERIAL-ANALOG-LOUT-IN-CC-M-AK	L – 50mm..150mm (large base under the order) SERIAL – 232, 485, 232-ET, 485-ET ANALOG – no, I, U LOUT – no, LOUT IN - IN CC – CG, CC M – 0,1m..10m AK – no, AK
FDRF651-75/L-SERIAL-ANALOG-LOUT-IN-CC-M-AK	L – 50mm..225mm (large base under the order) SERIAL – 232, 485, 232-ET, 485-ET ANALOG – no, I, U LOUT – no, LOUT IN - IN CC – CG, CC M – 0,1m..10m AK – no, AK
FDRF651-100/L-SERIAL-ANALOG-LOUT-IN-CC-M-AK	L – 50mm..300mm (large base under the order) SERIAL – 232, 485, 232-ET, 485-ET ANALOG – no, I, U LOUT – no, LOUT IN - IN CC – CG, CC M – 0,1m..10m AK – no, AK

7. Structure and operating principle

The micrometer operation is based on the so-called 'shadow' principle, Fig.1. The micrometer consists of two blocks – transmitter and receiver. Radiation of a semi-conductor laser or LED 1 is collimated by a lens 2. With an object placed in the collimated beam region, shadow image formed is scanned with a CCD photo-detector array 3. A processor 4 calculates the position (size) of the object from the position of shadow border (borders).

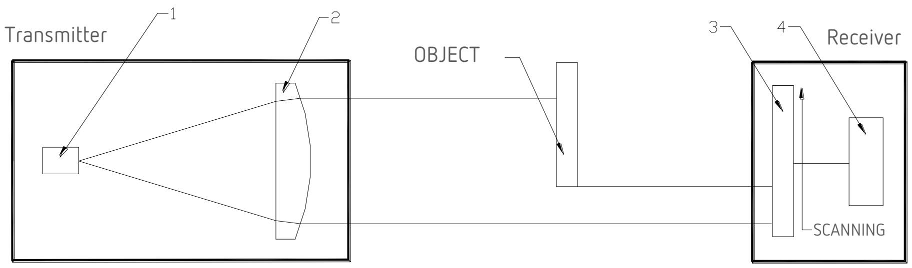


Figure 1

8. Options for use of the device

8.1. One-coordinate systems

Ways of using the micrometer for gauging of technological objects are shown in Fig. 2. Fig.2.1 – measuring of the edge position; Fig.2.2. – measuring of size or position; Fig.2.3. – measuring of the gap value or position; Fig.2.4. – measuring of internal or external dimension; Fig.2.5. – measuring of the size or position of large-size objects.

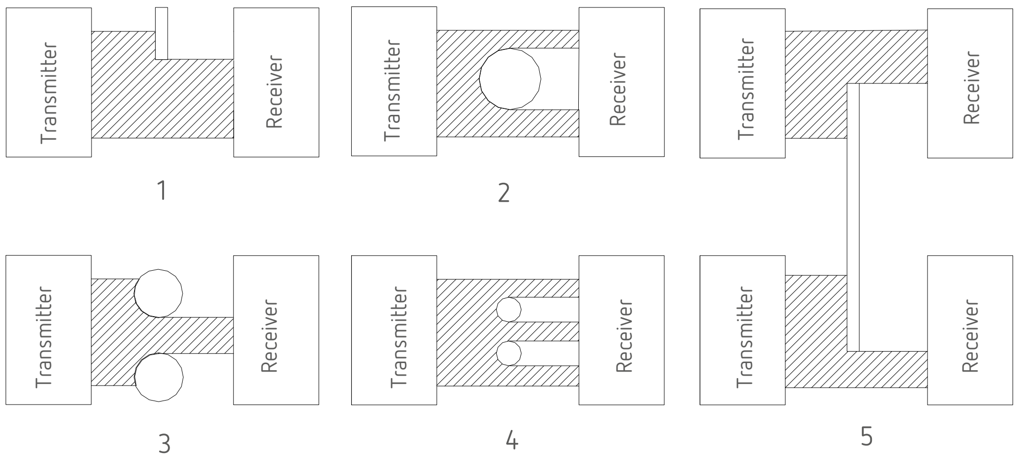


Figure 2

8.2. Multi-axis systems

Delivery of multi-axis measurement system (measured in several sections) is possible. Examples of the 2D and 3D-coordinate systems are shown in Figure 3.1 and 3.2, respectively. Ordering sample - FDRF651.NX / L ..., where N - number of coordinates in the system.

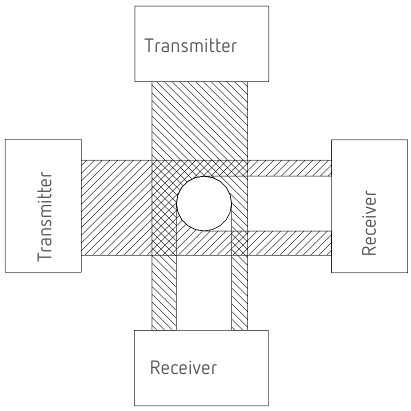


Figure 3.1.

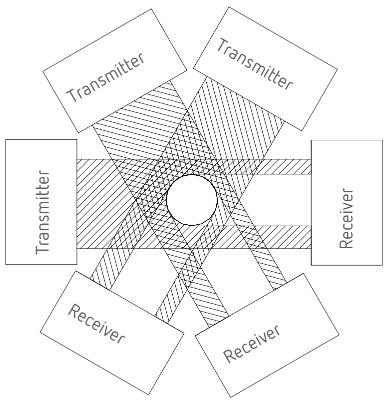
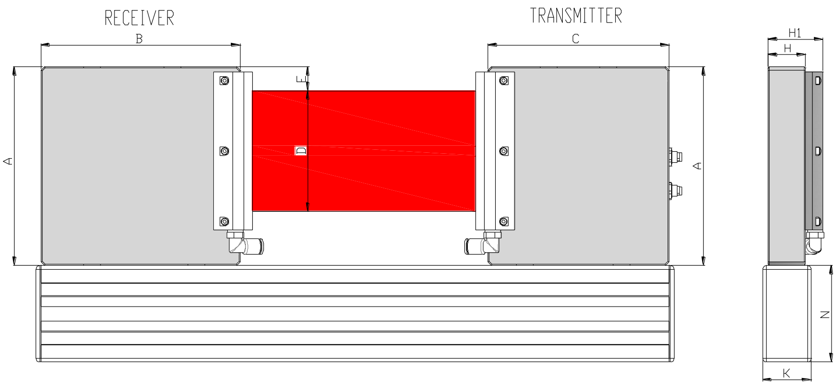


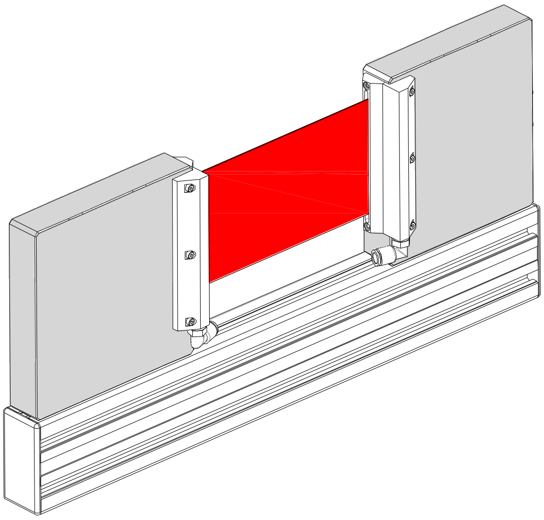
Figure 3.2.

9. Dimensions and mounting

9.1. Overall and mounting dimensions

Overall and mounting dimensions of micrometers are shown in Figure 4. Micrometer package is made of anodized aluminum.





	A, mm	B, mm	C, mm	D, mm	E, mm	H, mm	H1,mm	K, mm	N, mm
FDRF651-25	51	139	62	25	13	28	42,5	30	30
FDRF651-50	91	120	134	50	20	31	45,5	40	80
FDRF651-75	128	132	132	75	15	31	45,5	40	80
FDRF651-100	165	165	150	98	20	31	45,5	40	80

Figure 4

10. Connection

Micrometers are equipped with cable glands (CG option), or connectors (CC option).v
Micrometers with an Ethernet interface contains two cable glands or two connectors.

10.1. Micrometers without logical outputs

Micrometer is equipped by Binder 702-8 connector. Pin numbers and location of the installation is shown in Figure 5.

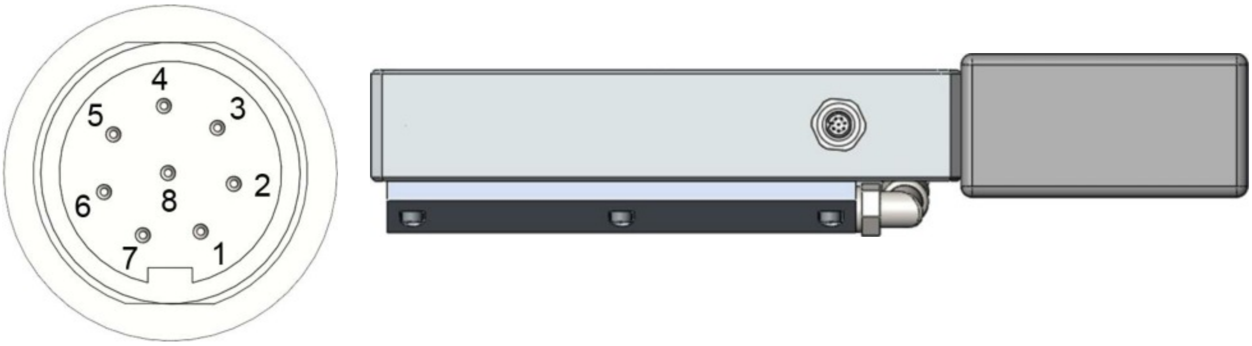


Figure 5

Designation of contacts is given in the following table:

Model of Micrometer	Pin number	Assignment
232 - U/I - IN-AL – CC	1	IN
	2	Gnd (power supply)
	3	TXD
	4	RXD
	5	Gnd (common for signals)
	6	AL
	7	U/I
	8	U+ (power supply)
485 - U/I - IN-AL - CC	1	IN
	2	Gnd (power supply)
	3	DATA+
	4	DATA-
	5	Gnd (common for signals)
	6	AL
	7	U/I
	8	U+ (power supply)

10.2. Micrometers with logical outputs

Micrometer is equipped by Binder 723-14 connector. Pin numbers and location of the installation is shown in Figure 6.

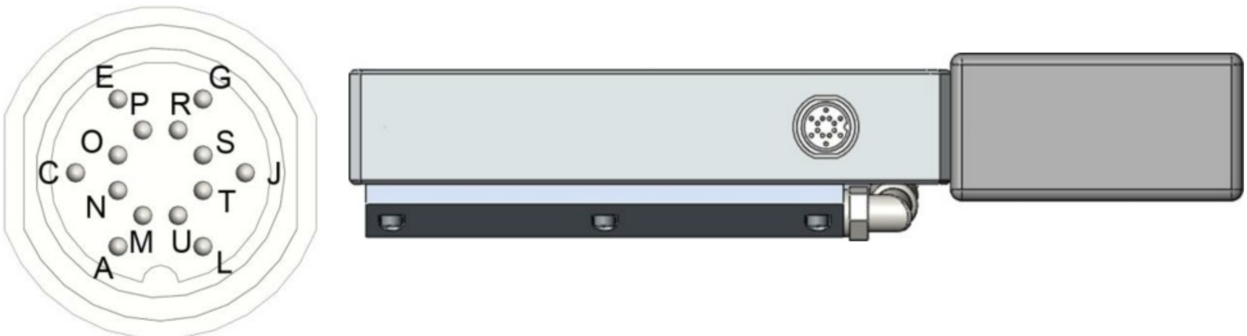


Figure 6

Designation of contacts is given in the following table:

Model of Micrometer	Pin number	Assignment
323 - U/I - IN-AL - TTL-OUT - CC	A	IN
	C	Gnd (power supply)
	E	TXD
	G	RXD
	J	Gnd (common for signals)
	L	AL
	M	U/I
	N	U+ (power supply)
	O	NormLimit

	P	UpLimit
	R	LowLimit
	S	N/C
	T	N/C
	U	N/C
485 - U/I - IN-AL - TTL-OUT - CC	A	IN
	C	Gnd (power supply)
	E	DATA+
	G	DATA-
	J	Gnd (common for signals)
	L	AL
	M	U/I
	N	U+ (power supply)
	O	NormLimit
	P	UpLimit
	R	LowLimit
	S	N/C
	T	N/C
	U	N/C

10.3. Micrometers with Ethernet interface

Micrometers are equipped by additional connector Binder 712-4. Pin numbers and location of the installation is shown in Figure 7.

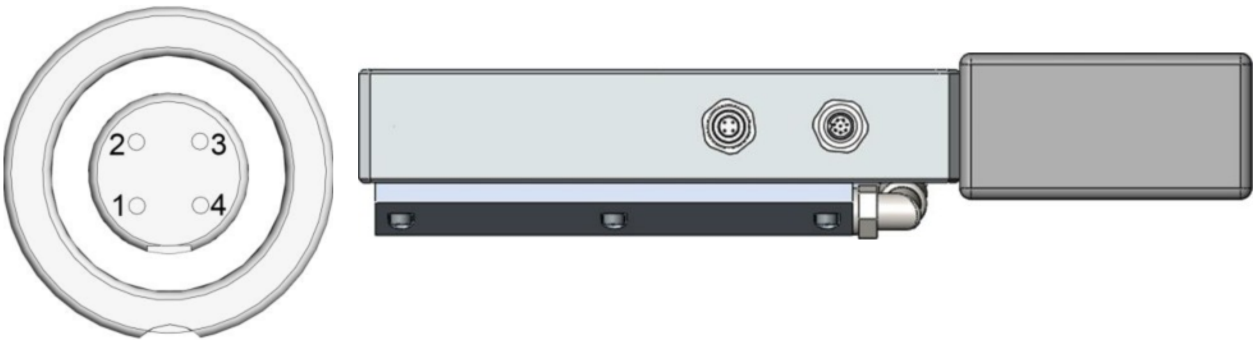


Figure 7

Designation of contacts is given in the following table:

Model of Micrometer	Pin number	Assignment
ET	1	TX+
	2	TX-
	3	RX+
	4	RX-

10.4. Micrometers with cable gland and cables

The place of the cable entry is shown in Figure 8.

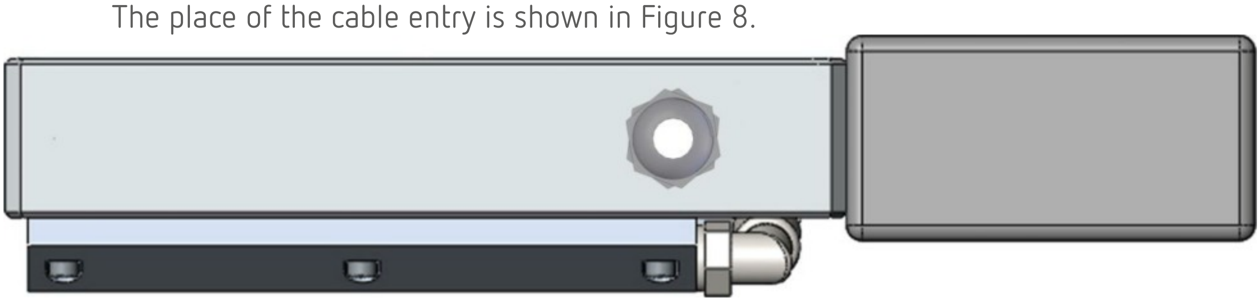


Figure 8

Designation of cable wires for micrometers with cable gland and cable of micrometers with cable connector is given in the following table:

Model of Micrometer	Pin number		Assignment	Wire color
232-U/I-IN - CG	free wire	-	IN	White
	free wire	-	Gnd (power supply)	Brown
	DB9	2	TXD	Green
	DB9	3	RXD	Yellow
	DB9	5	Gnd (common for signals)	Gray
	free wire	-	AL	Pink
	free wire	-	U/I	Blue
	free wire	-	U+ (power supply)	Red
485-U/I-IN – CG	free wire	-	IN	White
	free wire	-	Gnd (power supply)	Brown
	DB9	8	DATA+	Green
	DB9	7	DATA-	Yellow
	DB9	5	Gnd (common for signals)	Gray
	free wire	-	AL	Pink
	free wire	-	U/I	Blue
	free wire	-	U+ (power supply)	Red
232-U/I-IN-AL-LOUT-CG	free wire	-	IN	White
	free wire	-	Gnd (powe supply)	Brown
	DB9	2	TXD	Green
	DB9	3	RXD	Yellow
	DB9	5	Gnd (common for signals)	Gray
	free wire	-	AL	Pink
	free wire	-	U/I	Blue
	free wire	-	U+ (power supply)	Red
	free wire	-	NormLimit	White-Green
	free wire	-	UpLimit	Red-Blue
	free wire	-	LowLimit	Gray-Pink
485 - U/I - IN-AL - TTL-OUT - CG	free wire	-	IN	White
	free wire	-	Gnd (power supply)	Brown
	DB9	8	DATA+	Green
	DB9	7	DATA-	Yellow
	DB9	5	Gnd (common for signals)	Gray
	free wire	-	AL	Pink
	free wire	-	U/I	Blue
	free wire	-	U+ (power supply)	Red
	free wire	-	NormLimit	White-Green
	free wire	-	UpLimit	Red-Blue
	free wire	-	LowLimit	Gray-Pink

10.5. Ethernet cable

The place of the cable entry is shown in Figure 9.

Figure 9

Designation of cable wires is given in the following table:

Model of Micrometer	Pin number	Assignment
ET	TX+	Orange
	TX-	White-Orange
	RX+	Green
	RX-	White-Green

11. Configuration parameters

The nature of operation of the micrometer depends on its configuration parameters (operation modes), which can be changed only by transmission of commands through serial port RS232 or RS485. The basic parameters are as follows:

11.1. Parameter of Synchronization

This parameter specifies one of the three result sampling options in the case where the micrometer works in the data stream mode:

- Asynchronous Transmission
- Synchronous transmission, Time sampling;
- Synchronous transmission, Trigger sampling.

With **Asynchronous Transmission** selected, the micrometer automatically transmits the measurement result via serial interface as it is ready.
With **Time Sampling** selected, the micrometer automatically transmits the measurement result via serial interface in accordance with selected time interval (sampling period).
With Trigger sampling is selected, the micrometer transmits the measurement result when **external synchronization** input (IN input of the micrometer) is switched and taking the **division factor** set into account.

Note: The mode of operation of each of the interfaces can be set independently.

11.2. Sampling period

If the Time Sampling mode is selected, the 'sampling period' parameter determines the time interval in which the micrometer will automatically **transmit** the measurement result. The time interval value is set in increments of 0.1 ms.
For example, for the parameter value equal to 100, data are transmitted through bit-serial interface with a period of 0,1*100 = 10 ms.

If the Trigger Sampling mode is selected, the 'sampling period' parameter determines the division factor for the external synchronization input. For example, for the parameter value equal to 100, data are transmitted through bit-serial interface when each 100th synchronizing pulse arrives at IN input of the sensor.

Note 1. It should be noted that the 'sampling mode' and 'sampling period' parameters control only the transmission of data. The micrometer operation algorithm is so built that measurements are taken at a maximum possible rate determined by the integration time period, the measurement results is sent to buffer and stored therein until a new result arrives. The above-mentioned parameters determine the method of the readout of the result form the buffer.

Note 2. If the bit-serial interface is used to receive the result, the time required for data transmission at selected data transmission rate should be taken into account in the case where small sampling period intervals are used. If the transmission time exceeds the sampling period, it is this time that will determine the data transmission rate.

11.3. Method of results averaging

The averaging can operate in two modes:

- Off, no averaging
- Averaging over a number of results

When **averaging over a number of results** is selected, sliding average is calculated. The use of averaging makes it possible to reduce the output noise and increase the sensor resolution.

11.4. Number of averaged values/time of averaging

This parameter specifies the number of source results to be averaged for deriving the output value.
Averaging over a number of results does not affect the data update in the sensor output buffer.

Note. Maximum parameters value is 127.

11.5. Result type

The micrometer can provide the following types of results:

- Object dimension or
- Object positions or
- Deviation from definite dimension or position

11.6. Number of borders

A border means "light-shadow' transition or "shadow-light" transition which forms a shadow image of the object. Measurement is only conducted in the case where the number of borders detected by micrometer corresponds to a given parameter.

11.7. Numbers of borders under control

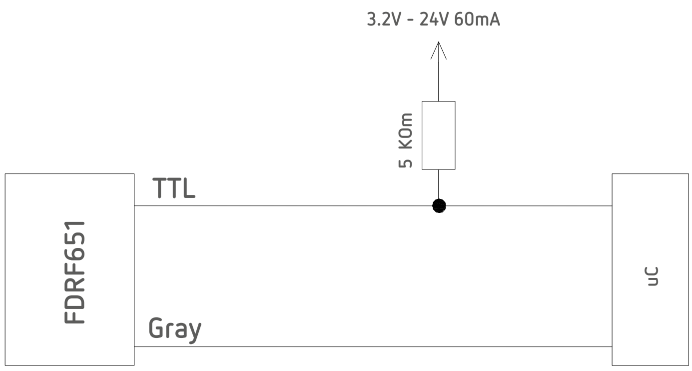
The measurement domain can include up to 8 borders, however, measurement can be made in relation to any two borders (hereinafter – borders A and B), whose numbers are specified by this parameter. Border numbers are counted in the direction of scanning. Direction of scanning is indicated on the body of receiver.

11.8. Nominal value and tolerances

The nominal value (dimension or position) can be transmitted as a parameter or preset by teaching. In the course of measurement, the micrometer controls sizes going beyond the permissible limits. Value of tolerances can be transmitted as parameters.

11.9. Logical outputs operation modes

Logical outputs of the micrometer are used to signal that the size under control is within or outside the tolerances selected. Logics of operation of the outputs can be changed, i.e. activate either low or high logical level. Wiring diagram of logical outputs are shown in the drawing:



11.10. Factory parameters by default

The parameters are stored in nonvolatile memory of the sensor. Correct changing of the parameters is carried out by using the parameterization program supplied with the sensor or a user program. The micrometers are supplied with the parameters shown in the table of point 15.10.

12. Description of RS232 and RS485 interfaces

12.1. RS232 port

The RS232 port ensures a "point-to-point" connection and allows the sensor to be connected directly to RS232 port of a computer or controller.

12.2. RS485 port

In accordance with the protocol accepted and hardware capability, the RS485 port makes it possible to connect up to 127 sensors to one data collection unit by a common bus circuit.

12.3. Modes of data transfer

Through these serial interfaces measurement data can be obtained by three methods:

- by single requests (inquiries);
- by automatic asynchronous data stream (results are transmitted as they become available);
- by automatic synchronous data stream (time sampling or trigger sampling)

12.4. Configuration parameters

12.4.1. Rate of data transfer through serial port

This parameter defines the rate of data transmission via the bit-serial interface in increments of 2400 bit/s. For example, the parameter value equal to 4 gives the transmission rate of 2400*4 = 9600 bit/s.

Note. The maximum transmission rate for RS232 interface is 460,8 kbit/s, and for RS485 interface the rate is 921,6 kbit/s

12.4.2. Net address

This parameter defines the network address of the sensor equipped with RS485 interface.

Note. Network data communications protocol assumes the presence of 'master' in the net, which can be a computer or other information-gathering device, and from 1 to 127 'slaves' (RF65x Series sensors) which support the protocol.

Each 'slave' is assigned a unique network identification code – a device address.

The address is used to form requests or inquiries all over the net. Each slave receive inquiries containing its unique address as well as '0' address which is broadcast-oriented and can be used for formation of generic commands, for example, for simultaneous latching of values of all sensors and for working with only one sensor (with both RS232 port and RS485 port).

12.4.3. Factory parameters table

Parameter	Value
Baud rate	230400 bit/s
Net address	1
Mode of data transfer	request

12.5. Interfacing protocol

12.5.1. Serial data transmission format

Data message has the following format:

1 start-bit	8 data bits	1 odd bit	1 stop-bit

12.5.2. Communication sessions types

The communications protocol is formed by communication sessions, which are only initiated by the 'master' (PC, controller). There are two kinds of sessions with such structures:

1) "request", ["message"] — ["answer"], **square brackets include optional elements**
2) "request" — "data stream" — ["request"].

12.5.3. Request

"Request" (INC) — is a **two-byte message**, which fully controls communication session. The 'request' message is the only one of all messages in a session where **most significant bit is set at 0**, therefore, it serves to synchronize the beginning of the session. In addition, it contains the device address (ADR), code of request (COD) and, optional, the message [MSG].

"Request" format:

Byte 0		Byte 1					[Bites 2...N]
INC0(7:0)		INC1(7:0)					MSG
0	ADR(6:0)	1	0	0	0	COD(3:0)	

12.5.4. Message

"Message" is data burst that can be transmitted by 'master' in the course of the session.

All messages with a "message" burst contain 1 in the most significant digit. Data in a message are transferred in tetrads. When byte is transmitted, lower tetrad goes first, and then follows higher tetrad. When multi-byte values are transferred, the transmission begins with lower byte.

The following is the format of two 'message' data bursts for transmission of byte:

DAT(7:0)									
Byte 0					Byte 1				
1	0	0	0	DAT(3:0)	1	0	0	0	DAT(7:4)

12.5.5. Answer

"Answer" is data burst that can be transmitted by 'slave' in the course of the session.

All messages with a message burst contain 1 in the most significant digit. Data in a message are transferred in tetrads. When byte is transmitted, lower tetrad goes first, and then follows higher tetrad. When multi-byte values are transferred, the transmission begins with lower byte.

When 'answer' is transmitted, the message contains:

- SB-bit, characterizes the updating of the result. If SB is equal to "1" this means that the sensor has updated the measurement result in the buffer, if SB is equal to "0" - then non-updated result has been transmitted (see. Note 1, p.10.3.). SB=0 when parameters transmit;
- two additional bits of cyclic binary batch counter (CNT). Bit values in the batch counter are identical for all sendings of one batch. The value of batch counter is incremented by the sending of each burst and is used for formation

(assembly) of batches or bursts as well as for control of batch losses in receiving data streams.
The following is the format of two 'answer' data bursts for transmission of byte:

DAT(7:0)							
Byte 0				Byte 1			
1	SB	CNT(1:0)	DAT(3:0)	1	SB	CNT(1:0)	DAT(7:4)

12.5.6. Data stream

'Data stream' is an infinite sequence of data bursts or batches transmitted from 'slave' to 'master', which can be interrupted by a new request.
In transmission of 'data stream' one of the 'slaves' fully holds data transfer channel, therefore, when 'master' produces any new request sent to any address, data streaming process is stopped. Also, there is a special request to stop data streaming.

12.5.7. Request codes and list of parameters

Request codes and list of parameters are presented in Chapter 13 and 14.

13. Description of Ethernet interface

Ethernet interface is used only for the reception of data from the sensor. Parameterization of sensors is carried out via RS232 or RS485 interface.

13.1. Modes of data transfer

The sensor can be operated in the following modes:

- No transmission.
- Automatic asynchronous data stream (results are transmitted as they become available);
- Automatic synchronous data stream (time sampling or trigger sampling)

13.2. Protocols

There are two types of protocols:

- MAC level of the OSI. The packets are transmitted in accordance with the protocol of 802.3
- IP/UDP protocol

13.3. Data packet format, MAC level

The micrometer sends MAC-packet of 28 byte length:

- bytes 0-5	:	MAC address of receiver
- bytes 6-11	:	MAC address of sender: : 0x00; 0x20; 0xED; 0x03; serial_number_H; serial_number_L;
- bytes 12-13	:	packet length: 0x00; 0x1D
- byte 14	:	sensor type – 65
- byte 15	:	firmware version
- bytes 16-17	:	serial number
- bytes 18-19	:	transmitter-receiver distance
- bytes 20-21	:	measurement range
- byte 22	:	cyclic counter of the packet, the value of counter is incremented by the sending of each packet
- byte 23	:	measurement result, byte 3
- byte 24	:	measurement result, byte 2
- byte 25	:	measurement result, byte 1
- byte 26	:	measurement result, byte 0

- byte 27	:	flag, which characterizes the update of the result
- byte 28	:	packet checksum

In this mode, the micrometer transmits packet with the result of one measurement in accordance with the established mode of sampling.

13.4. Data packet format, IP/UDP

The micrometer sends IP/UDP-packet of variable length. Packet length depends on parameter {0x52h}, which determines quantity of measurement results in the packet. Packet consists of a header and field of data:

- bytes 0-5	:	MAC address of receiver
- bytes 6-11	:	MAC address of sender: : 0x00; 0x20; 0xED; 0x03; serial_number_H; serial_number_L;
- bytes 12-13	:	Ethernet type: 0x08; 0x00;
- byte 14	:	IP version & header: 0x45;
- byte 15	:	0x00;
- bytes 16-17	:	total length of the packet including the IP header
- bytes 18-19	:	packet ID: 0x08; 0x7F;
- bytes 20-21	:	packet flags;
- byte 22	:	packet TTL: 0x80
- byte 23	:	protocol: 0x11;
- bytes 24-25	:	IP checksum;
- bytes 26-29	:	IP address of the sender;
- bytes 30-33	:	IP address of recipient;
- bytes 34-35	:	sender port: 0x13; 0x88;
- bytes 36-37	:	recipient port: 0x02; 0x5D;
- bytes 38-39	:	UDP packet length;
- bytes 40-41	:	UDP checksum;
- byte 42	:	sensor type
- byte 43	:	firmware version
- bytes 44-45	:	serial number
- bytes 46-47	:	transmitter-receiver distance
- bytes 48-49	:	measurement range
- byte 50	:	number of measurement results in the packet
- byte 51	:	cyclic counter of the packet
- bytes 52-53	:	measurement result_0
- byte 54	:	flag of the result_0 (0/1)
....		
- bytes 56+N	:	measurement result_N
- byte 57+N	:	flag of the result_N (0/1)

In this mode the internal data transmission buffer of the micrometer is filled with measurement data in accordance with a selected sampling mode of Time or Trigger (see p.10.2.) and corresponding sampling period (see p.10.3.). After the internal buffer has been filled (buffer size is determined by parameter), the sensor transmits data packet accumulated in the transmission buffer to UDP network.

14. Analog outputs

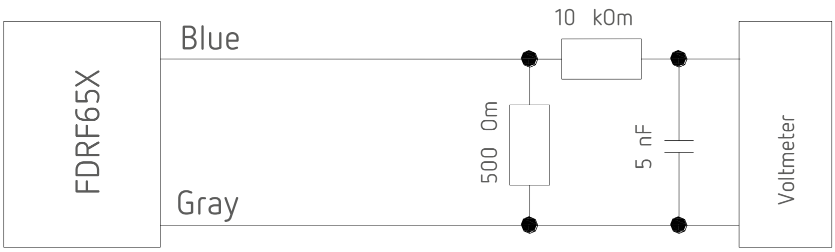
14.1. Modes of data transfer

The micrometer can be operated in the following modes:

- No transmission.
- Automatic asynchronous data stream (results are transmitted as they become available);
- Automatic synchronous data stream (time sampling or trigger sampling)

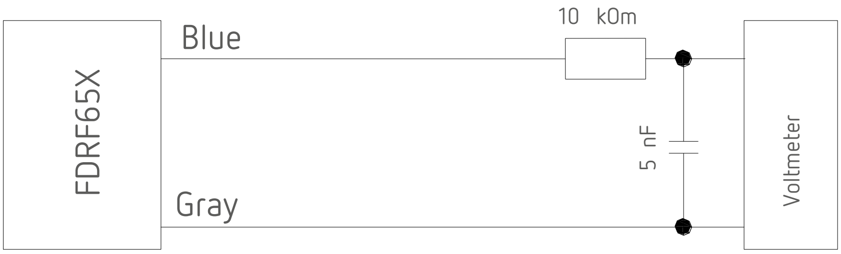
14.2. Current output 4...20 mA

The connection scheme is shown in the figure. The value of load resistor should not be higher than 500 Om. To reduce noise, it is recommended to install RC filter before the measuring instrument. The filter capacitor value is indicated for maximum sampling frequency of the micrometer (2 kHz) and this value increases in proportion to the frequency reduction.



14.3. Voltage output

The connection scheme is shown in the figure. To reduce noise, it is recommended to install RC filter before the measuring instrument. The filter capacitor value is indicated for maximum sampling frequency of the micrometer (2 kHz) and this value increases in proportion to the frequency reduction.



14.4. Configuration parameters

14.4.1. Range of the analog output

While working with the analog output, resolution can be increased by using the 'Window in the operating range' function which makes it possible to select a window of required size and position in the operating range of the sensor within which the whole range of analog output signal will be scaled.

If the beginning of the range of the analog signal is set at a higher value than the end value of the range, this will change the direction of rise of the analog signal.

Note. If the beginning of the range of the analog signal is set at a higher value than the end value of the range, this will change the direction of rise of the analog signal.

14.4.2. Analog output operation mode

Analog output can be:

- in the window mode or
- in the deviation mode.

"Window mode". The entire range of the analog output is scaled within the selected window. Under the window we mean the whole range of micrometer or any area in the measuring range specified in "Top of the range of the analog output and the" End of the range of the analog output area. The analog output is "0" outside the window,.

"Deviation mode". For 'deviation'- type result, the window boundaries must be defined so that the value corresponding to zero deviation is **located in the middle of the window**. In this case, the middle part of the analog output range (12mA or 5V) will correspond to zero deviation.

15. Request codes

15.1. Request codes table

Request code	Description	Message (size in bytes)	Answer (size in bytes)
{0x01h}	Device identification	—	– device type (1) – firmware release (1) – serial number (2) – distance "trns-receiv" (2) – range (2)
{0x02h}	Reading of parameter	- code of parameter (1)	- value of parameter (1)
{0x03h}	Writing of parameter	- code of parameter (1) - value of parameter (1)	—
{0x04h}	Storing current parameters to FLASH-memory	- constant AAh (1)	- constant AAh (1)
{0x04h}	Recovery of parameter default values in FLASH-memory	- constant 69h (1)	- constant 69h (1)
{0x05h}	Latching of current result	—	—
{0x06h}	Inquiring of result	-code of sychronization-source (1) 0x01h – internal timer 0x02h – trigger	- result in micrometers (4)
{0x07h}	Inquiring of a stream of results	—	- stream of results (4)
{0x08h}	Stop data streaming	—	—
{0x0Ch}	Nominal value setting		- constant 0Ch

15. List of parameters

15.1. Switching parameters

Code	Name	Values	Action
0x20h	Sensor On/OFF	0	micrometer is OFF and in power save mode
		1 (by default)	Micrometer is ON

15.2. Synchronization parameters

Code	Name	Values	Action
0x00h	Synchronization source	0 (by default)	OFF, asynchronous transmitting
		1	Time sampling, source – internal timer with 100 us period
		2	Trigger sampling, source - trigger
0x01h - 0x02h	Multiplier of internal timer/divider of input trigger	0-65535 (by default - 0x64h)	Sets the number of impacts timing signal to the system response

15.3. Averaging parameters

Code	Name	Values	Action
0x21h	Averaging	0 (by default)	OFF
		1	ON
0x22h – 0x23h	Quantity of measurement values	1-4096 (by default 0x04h)	Number of averaged values.

15.4. Type of measurement parameters

Code	Name	Values	Action
0x24h	Type of measurement	0 (by default)	One border (knife);
		1	Distance between and borders (size of the object). Result = B – A. (Borders number are set by 0x25h and 0x26 h parameters).
		2	Object position – (B+A)/2.
		3	Border A position.
		4	Border B position.
0x25h	Number of A border	0-127 (by default 0x00h)	Index number of border A.
0x26h	Number of B border	1-127 (by default 0x01h)	Index number of border B.

15.5. Nominal value and tolerances parameters

Code	Name	Values	Action
0x40h - 0x43h	Nominal value	0 – micrometer range (by default 0x00h)	Sets nominal value for deviation calculations
0x45h - 0x48h	Minimal tolerance	In microns 0 – micrometer range (by default 0x00h)	Sets the allowable value is less than the nominal.
0x49h - 0x4Ch	Maximal tolerance	In microns 0 – micrometer range (by default - micrometer range)	Sets the allowable value is more than the nomi-nal.

15.6. Logical outputs control parameters

Code	Name	Values	Action
0x44h	Control byte of output logic polarity	By default 0x00h	byte x,x,x,x,x,X,Y,Z bit Z - LowLimit bit control: 0 – LowLimit low active level, 1 – LowLimit high active level; bit Y - HighLimit bit control: 0 – HighLimit low active level, 1 – HighLimit high active level; bit X - PASS bit control: 0 – Normal low

			active level, 1 – Normal high active level;

15.7. Serial interface parameters

Code	Name	Values	Action
0x10h	Data transfer mode	0 (by default) 1 2	OFF. Asynchronous Synchronous
0x11h - 0x12h	Rate of data transfer through serial port	X*2400 (by default 0x60h)	X = 48; 48*2400=115200
0x13h	Net address	1 (by default)	Net address

15.8. Analog output parameters

Code	Name	Values	Action
0x30h	Data transfer mode	0 (by default) 1 2	OFF. Asynchronous Synchronous
0x31h - 0x34h	The beginning of analog output range	In microns (by default - 0)	Sets point in the range where analog output has minimal value
0x35h - 0x38h	The end of analog output range	In microns (by default – the range of micrometer)	Sets point in the range where analog output has maximal value
0x39h	Analog output mode	0 (by default)	Window
		1	Deviation

15.9. Ethernet interface parameters

Code	Name	Values	Action
0x50h	Data transfer mode	0 (by default)	OFF.
		1	Asynchronous
		2	Synchronous
0x51h	Packet type	0	MAC
		1 (by default)	IP/UDP
0x52h	Measurements quantity in packet	0-255 (by default 0x05h)	
0x53h - 0x58h	MAC address of recipient	By default 00h-00h-00h-00h-00h-00h	
0x59h - 0x5Ch	Subnet mask	By default FFh-FFh-FFh-00h	
0x5Dh - 0x60h	Sender IP address	By default C0h-A8h-00h-02h	
0x61h - 0x64h	Recipient IP address	By default C0h-A8h-00h-01h	

15.10. Factory parameters by default

Address	Parameter	Value
0x00h	Synchronization source	00h
0x01h - 0x02h	Multiplier of internal timer/divider of input trigger	64h - 00h

0x10h	Serial interface mode - stream	00h
0x11h - 0x12h	Baud rate	60h - 00h
0x13h	Net address	01h
0x20h	Micrometers switching	01h
0x21h	Averaging	00h
0x22h – 0x23h	Number of averaging values	04h - 00h
0x24h	Type of measurement	00h
0x25h	Number of border A	00h
0x26h	Number of border B	01h
0x30h	Analog output mode	01h
0x31h - 0x34h	Beginning of analog output	00h 00h 00h 00h
0x35h - 0x38h	The end of analog output	Micrometer range
0x40h - 0x43h	Nominal value	00h 00h 00h 00h
0x44h	Control byte of output logic	00h
0x45h - 0x48h	Minimal tolerance	00h
0x49h - 0x4Ch	Maximal tolerance	Micrometer range
0x50h	ETHERNET mode - stream	01h
0x51h	Packet type	01h
0x52h	The number of measurements	05h
0x53h - 0x58h	Recipient MAC address	00h 00h 00h 00h 00h 00h
0x59h - 0x5Ch	Subnet mask	FFh FFh FFh 00h
0x5Dh - 0x60h	Sender IP address	02h 00h A8h C0h
0x61h - 0x64h	Recipient IP address	01h 00h A8h C0h

15.11. Notes

- All values are given in binary form.
- The range is given in millimeters.

- The value of the result transmitted by micrometer is represented by 4 bytes and is given in micrometers
- On special request (05h), the current result can be latched in the output buffer where it will be stored unchanged up to the moment of arrival of request for data transfer. This request can be sent simultaneously to all micrometers in the net in the broadcast mode in order to synchronize data pickup from all sensors.
- When working with the parameters, it should be borne in mind that when power is OFF the parameter values are stored in nonvolatile FLASH-memory of the sensor. When power is ON, the parameter values are read out to RAM of the sensor. In order to retain these changes for the next power-up state, a special command for saving current parameter values in the FLASH-memory (04h) must be run.
- Parameters with the size of more than one byte should be saved starting from the high-order byte and finishing with the low-order byte.
- WARNING! It is forbidden to carry out the configuration of sensors included in the RS485 network

15.12. Examples of parameters setting

- Border position measurement. Number of borders – 1, A = 0, B = 1; Parameters [0x24h] = 00h, [0x25h] = 00h;
- Diameter measurement. Number of borders – 2, A = 0, B = 1; Parameters [0x24h] = 01h, [0x25h] = 00h, [0x26h] = 01h;
- Gap measurement. Number of borders – 2, A = 0, B = 1; Parameters [0x24h] = 01h, [0x25h] = 00h, [0x26h] = 01h;
- Shaft position measurement. Number of borders – 2, A = 0, B = 1; Parameters [0x24h] = 02h, [0x25h] = 00h, [0x26h] = 01h;
- Ring inner diameter measurement. Number of borders – 4, A = 1, B = 2; Parameters [0x24h] = 01h, [0x25h] = 01h, [0x26h] = 02h

15.13. Examples of communication sessions

1) Request "Device identification".

Condition: device address —1, request code – {0x01h}, device type — 61, firmware release — 88 (58h), serial number — 0402 (0192h), transmitter-receiver distance — 80mm (0050h), measurement range — 50 (0032h), packet number — 1.

The request format:

Byte 0		Byte 1					[Bytes 2...N]
INC0(7:0)		INC1(7:0)					MSG
0	ADR(6:0)	1	0	0	0	COD(3:0)	

Request from "Master"

Byte 0								Byte 1								
INC0(7:0)								INC1(7:0)								
0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1	
01h								81h								

The following is the format of two 'answer' data bursts for transmission of byte DAT(7:0):

DAT(7:0)							
Byte 0				Byte 1			
1	0	CNT(1:0)	DAT(3:0)	1	0	CNT(1:0)	DAT(7:4)

Answer of "Slave":

Device type:

DAT(7:0)															
Byte 0								Byte 1							
1	0	0	1	0	0	0	1	1	0	0	1	0	1	1	0
91h								96h							

Firmware release

DAT(7:0)															
Byte 0								Byte 1							
1	0	0	1	1	0	0	0	1	0	0	1	0	1	0	1
98h								95h							

Serial Number

DAT(7:0)															
Byte 0								Byte 1							
1	0	0	1	0	0	1	0	1	0	0	1	1	0	0	1
92h								96h							
DAT(7:0)															
Byte 0								Byte 1							
1	0	0	1	0	0	0	1	1	0	0	1	0	0	0	0
91h								90h							

Transmitter-receiver distance

DAT(7:0)															
Byte 0								Byte 1							
1	0	0	1	0	0	0	0	1	0	0	1	0	1	0	1
90h								95h							
DAT(7:0)															
Byte 0								Byte 1							
1	0	0	1	0	0	0	0	1	0	0	1	0	0	0	0
90h								90h							

Measurement range

DAT(7:0)															
Byte 0								Byte 1							
1	0	0	1	0	0	1	0	1	0	0	1	0	0	1	1
92h								93h							
DAT(7:0)															
Byte 0								Byte 1							
1	0	0	1	0	0	0	0	1	0	0	1	0	0	0	0
90h								90h							

```
//C# code example
        out_data[0] = 0x01;
        out_data[1] = 0x81;
        SerialPort.Write(out_data, 0, 2);

        Thread.Sleep(50);
        SerialPort.Read(in_data, 0, 16);
        SensorType = ((in_data[0] & 0x0F) + ((in_data[1] & 0x0F) << 4));
        Release = ((in_data[2] & 0x0F) + ((in_data[3] & 0x0F) << 4));
        Serial = ((in_data[4] & 0x0F) + ((in_data[5] & 0x0F) << 4) + ((in_data[6] & 0x0F)
<< 8) + ((in_data[7] & 0x0F) << 12));
        Base_distance = ((in_data[8] & 0x0F) + ((in_data[9] & 0x0F) << 4) + ((in_data[10]
& 0x0F) << 8) + ((in_data[11] & 0x0F) << 12));
        Measure_Range = ((in_data[12] & 0x0F) + ((in_data[13] & 0x0F) << 4) +
((in_data[14] & 0x0F) << 8) + ((in_data[15] & 0x0F) << 12));
```

Note: as bust number =1, then CNT=1

2) Request "Reading of parameter".
Condition: device address —1, request code – 02h, code of parameter — 05h, value of parameter — 04h, packet number — 2.
Request ("Master") — 01h;82h;
Message ("Master") — 85h, 80h;
Answer ("Slave") — A4h, A0h

//C# code example
```
paramADR = 0x05;
out_data[0] = 0x01;
out_data[1] = 0x82;
out_data[2] = (byte)((paramADR & 0x0F) + 0x80); //low
out_data[3] = (byte)((((paramADR & 0xF0) >> 4)) + 0x80); //high
SerialPort.Write(out_data, 0, 4);

Thread.Sleep(50);
SerialPort.Read(in_data, 0, 2);
paramVALUE = (long)((in_data[0] & 0x0F) + ((in_data[1] & 0x0F) << 4));
```

3) Request "Inquiring of result".
Condition: device address — 1, result — 02A5h, packet number — 3.
Request ("Master") — 01h;86h;
Answer ("Slave") — B0h, B0h, B0h, B0h, B0h, B2h, BAh, B5h
Measured displacement: X=0x000002A5h= 677 um

//C# code example
```
out_data[0] = 0x01;
out_data[1] = 0x86;
SerialPort.Write(out_data, 0, 2);

Thread.Sleep(50);
SerialPort.Read(in_data, 0, 8);
UInt32 temp1 = (UInt32)((in_data[0] & 0x0F) + ((in_data[1] & 0x0F) << 4) +
((in_data[2] & 0x0F) << 8) + ((in_data[3] & 0x0F) << 12));
UInt32 temp2 = (UInt32)((in_data[4] & 0x0F) + ((in_data[5] & 0x0F) << 4) +
((in_data[6] & 0x0F) << 8) + ((in_data[7] & 0x0F) << 12));
Int32 Measure = (Int32)(temp1 + (temp2 << 16));
```

4) Request: "writing the divider ration"
Condition: divider ration – 1234=3039h, device address – 1, request code – 03h, code of parameter – [0x01h,0x02h], value of parameter – 0x11FFh
Request ("Master") – 01h, 83h
Message ("Master") – 82h, 80h, 81h, 81h;
Request ("Master") – 01h, 83h
Message ("Master") – 81h, 81h, 8Fh, 8Fh;
//C# code example
```
NewValue = 0x11FF;
Param_Size=2;
Param_Adress = 0x01;
while (Param_Size > 0)
{
        out_data[0] = 0x01;
        out_data[1] = 0x83;
        out_data[2] =(byte)((((byte)(Param_Adress + Param_Size-1))& 0x0F)+ 0x80);
        out_data[3] =(byte)((((byte)(Param_Adress + Param_Size-1)&0xF0)>>4)+0x80);
        out_data[4] =(byte)(((NewValue >> (8 * Param_Size - 8)) & 0x0F) + 0x80);
        out_data[5] =(byte)(((NewValue >> (8 * Param_Size - 4)) & 0x0F) + 0x80);
        SerialPort.Write(out_data, 0, 6);
        Param_Size--;
}
```

5) Request "writing sampling regime (internal timer sampling)".
Condition: device address – 1, request code – 07h,
Request ("Master") – 01h, 87h;
Message ("Master") – 82h, 80h;

//C# code example
```
        out_data[0] = 0x01;
        out_data[1] = 0x87;
        out_data[2] = 0x81;
        out_data[3] = 0x80;
        SerialPort.Write(out_data, 0, 4);

        while(true)
        {
                SerialPort.Read(in_data, 0, 8);
                UInt32 temp1 = (UInt32)((in_data[0] & 0x0F) + ((in_data[1] & 0x0F) << 4) +
                ((in_data[2] & 0x0F) << 8) + ((in_data[3] & 0x0F) << 12));
                UInt32 temp2 = (UInt32)((in_data[4] & 0x0F) + ((in_data[5] & 0x0F) << 4) +
                ((in_data[6] & 0x0F) << 8) + ((in_data[7] & 0x0F) << 12));
                Int32 Measure = (Int32)(temp1 + (temp2 << 16));
        }
```

16. Ethernet packets examples

16.1. MAC packet

- Parameter [0x51h] = 0 (MAC level).
- Recipient MAC address [0x53h – 0x58h]
- Stream mode
- Mode – "Results" (parameter [0x00h], request {0x07h})

The format of received packet:

	0-7	8-15	16-23	24-31	32-39	40-47
0	Recipient MAC address.					
1	Sender MAC address: 0x00; 0x20; 0xED; 0x03; serial_number_H; serial_number_L;					
2	Packet length: 0x00; 0x1C;		Sensor type	Firmware version	Serial number	
3	"Transmitter-receiver" distance		Measurement range		Cyclic counter of the packet	Result, byte 3
4	Result, byte 2	Result, byte 1	Result, byte 0	Flag of result (0/1)	Checksum	Not used

16.2. IP/UDP

- Parameter [0x51h] = 1 (IP level).
- Recipient MAC address [0x53h – 0x58h]
- Stream mode
- Mode – "Results" (parameter [0x00h], request {0x07h})

The format of received packet:

	0-7	8-15	16-23	24-31	32-39	40-47
0	Recipient MAC address.					
1	Sender MAC address: 0x00; 0x20; 0xED; 0x03; serial_number_H; serial_number_L;					
2	Ethernet type: 0x08; 0x00;		IP version & header: 0x45	0x00	Total packet length	

3	ID: Cyclic counter of the packet;		Flags: 0x40; 0x00;		TTL: 0x80	Protocol: 0x11
4	Checksum		Sender IP address:			
5	Recipient IP address:				Sender port: 0x13; 0x88;	
6	Recipient port: 0x02; 0x5D;		UDP packet length		Checksum	
7	Sensor type	Firmware version	Serial number		"Transmitter-receiver" distance	
8	Measurement range		Number of measurements in packet	Cyclic counter of the packet	Result_0, byte 3	Result_0, byte 2
9	Result_0, byte 1	Result_0, byte 0	Flag of result_0 (0/1)	…	…	…
10	…	…	Result_N, byte 3	Result _N, byte 2	Result_N, byte 1	Result_N, byte 0
11	Flag of result_N (0/1)	Not used	Not used	Not used	Not used	Not used

17. Parameterization program

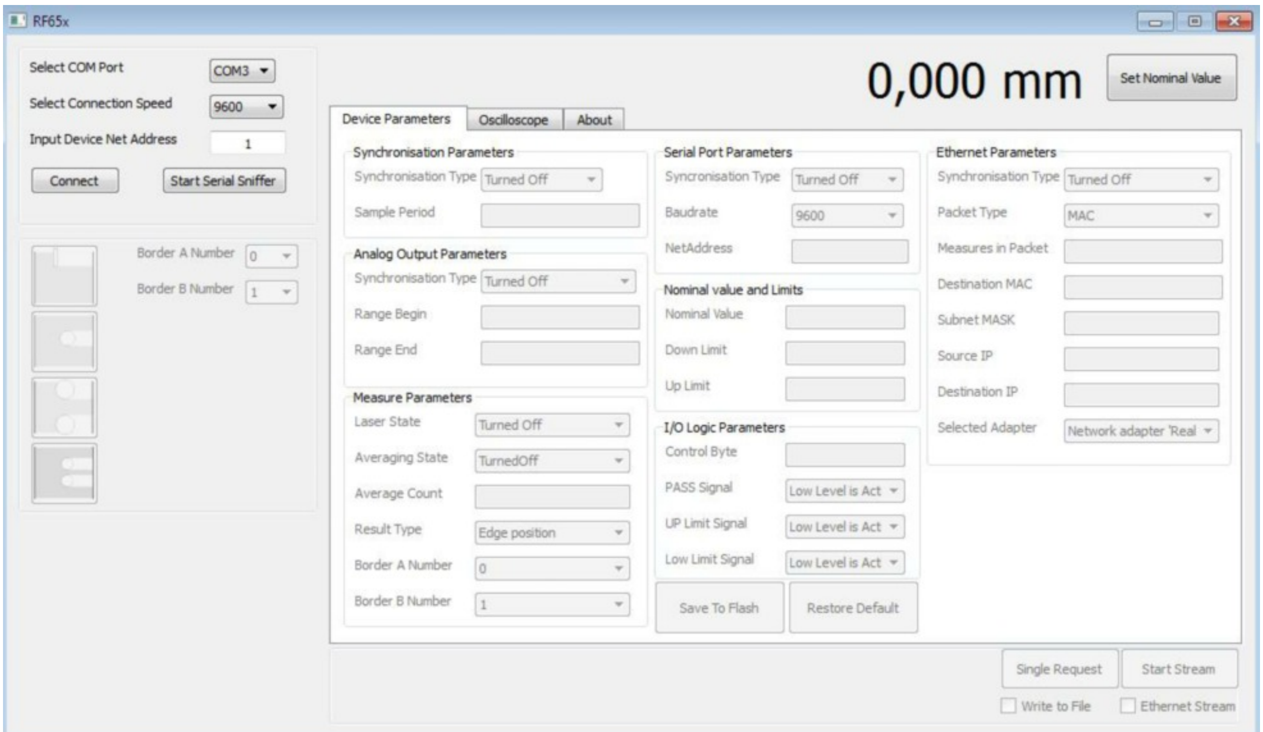
17.1. Function

The FDRF65X-SP software is intended for:

1) Testing and demonstration of work of RF651 series micrometers;
2) Setting of the micrometer parameters;
3) Reception and gathering of the micrometer data signals

17.2. Obtaining connection to micrometer

Once the program is started, the main window emerges:

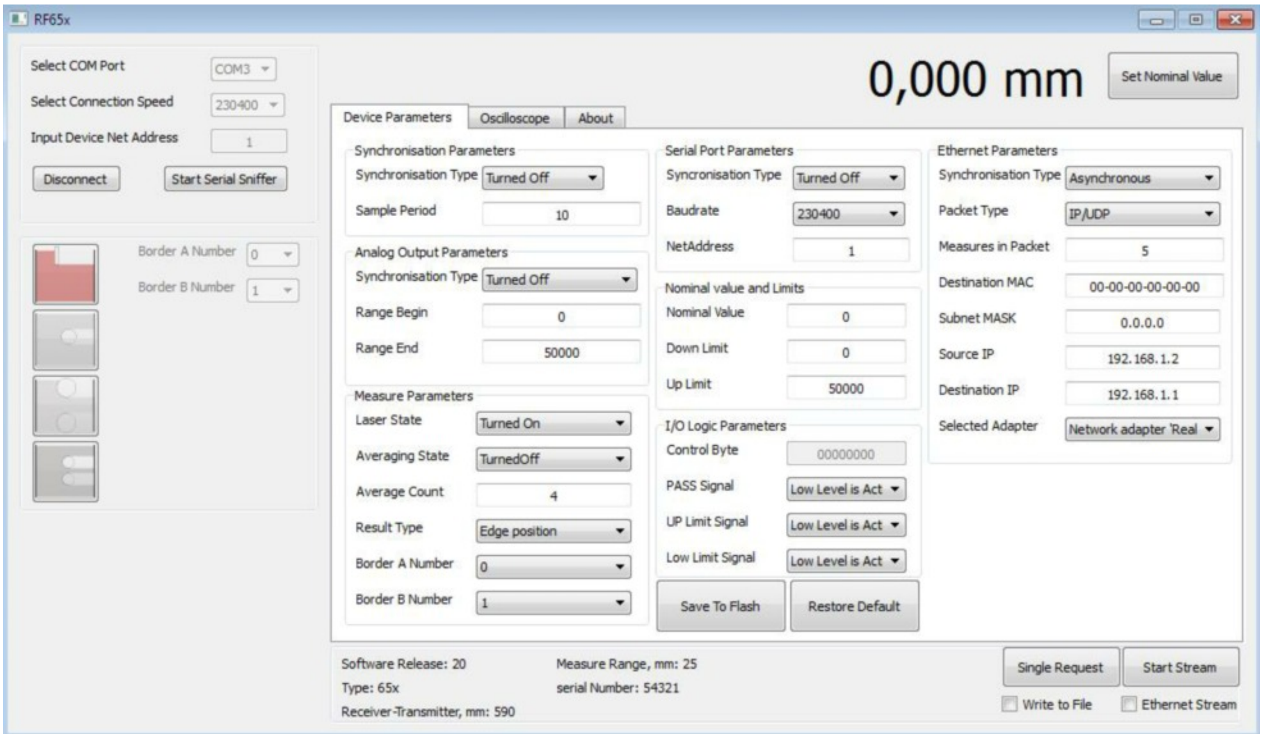


To obtain connection via RS232/RS485 interfaces:

- select COM-port whereto the sensor is connected (logical port if the sensor is connected via USB-adapter)

- select transmission rate (Baud rate) at which the sensor will work (230400 by default)
- select the sensor network address, if necessary
- press Connect button.

If the selected parameters correspond to the parameters of the micrometer interface, the program will identify the micrometer, read and display its configuration parameters:



17.3. Setting and saving parameters of the sensor

The part of FDRF65x application, which has become an active, allows to edit and to put in RAM and FLASH memory of micrometer appropriate parameters.

The parameters table of the micrometer is divided into several groups, which are identical of logical groups of parameters described in par 15.

Configuring the micrometer is done by selecting the appropriate item from the proposed drop-down menu, or by entering the absolute value of the desired parameter (all parameters are entered in decimal form, the user must follow the correct input of a specific parameter). After selecting the desired value from the drop down menu system will automatically load it into the RAM. If you enter the absolute value of the parameter the record in the RAM is made after hitting Enter key.

In the left part of the window there are buttons that let carries out rapid switching between operating modes of the micrometer, that is, to carry out selection of the type of measurement results:

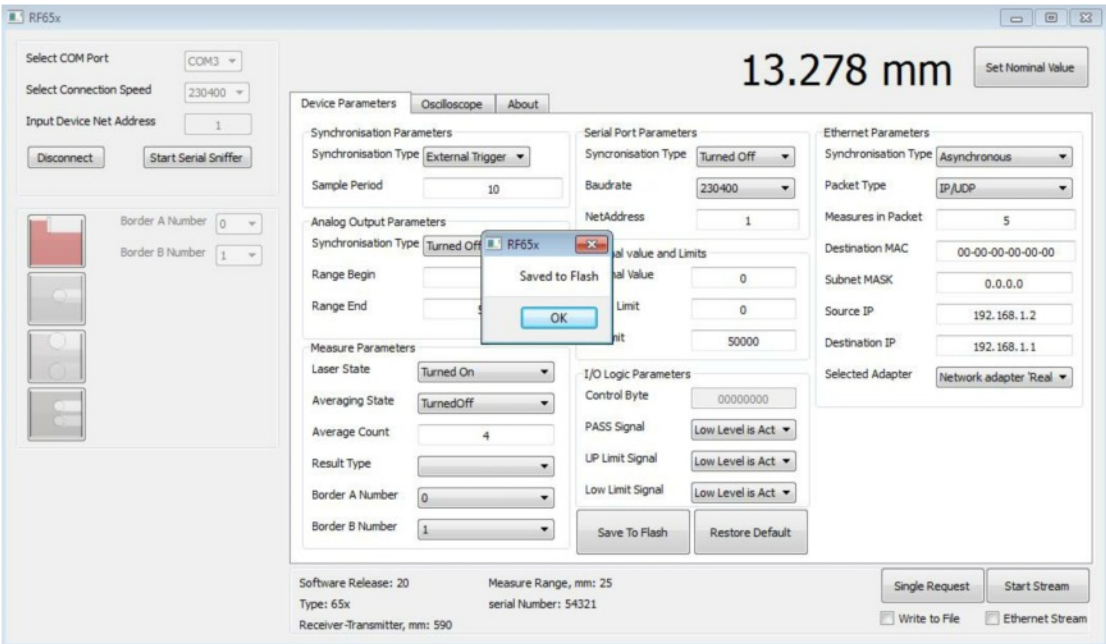
- one border measurement
- diameter or position measurement
- gap or it's position measurement
- complex object measurement

In the same part of the screen there are two drop-down menu. Here you can set the number of measured boundaries. The buttons and drop-down lists are a tool to quickly set up the device. Described buttons and drop-down lists are directly related to

The fields Result Type, Border A Number, Border B Number which permit a more detailed configuration of the micrometer

17.4. Saving parameters and factory parameters restore

In the region of parameter setting there are two 2 buttons SaveTo-Flash and RestoreDefault, they allow to save the current parameters from the RAM of micrometer in non-volatile memory and recovery factory micrometer settings respectively.

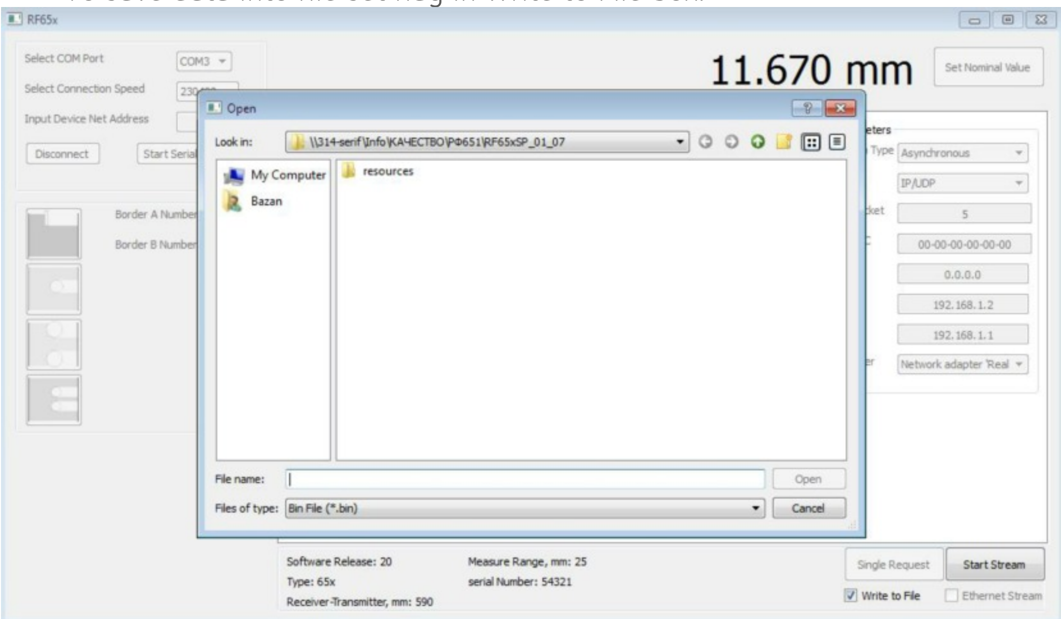


For changes to take effect, you must end the connection session and reset micrometer by turning off the micrometer power

18. Working with micrometer

- Place object into working range of micrometer
- To get single result press button SingleRequest
- To get continuous data stream it is necessary to set synchronization mode and press button StartStream. The result of measurement is indicated on the display.
- To get result through Ethernet interface set flag in Ethernet Stream box in the right corner of the window.
- To see the history of measurement go to the tab Oscillocope. This tab displays the last 50 measurements.

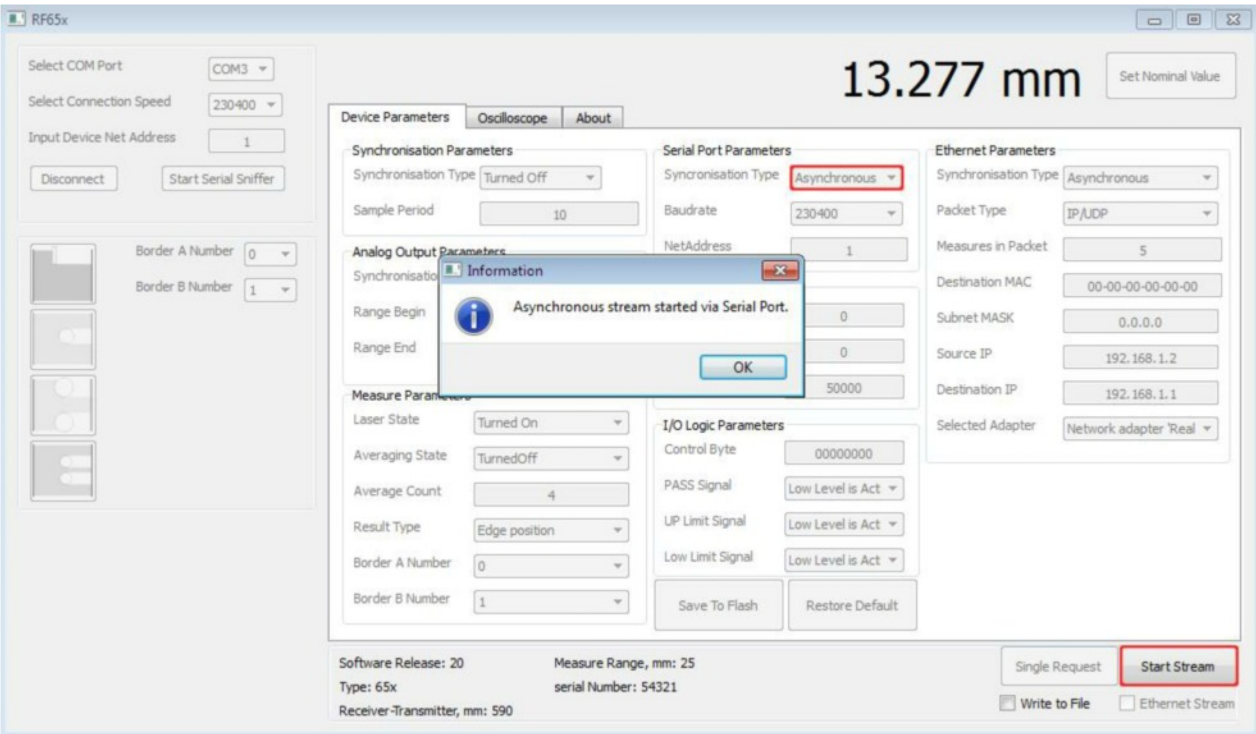
- To save data into file set flag in Write to File box.



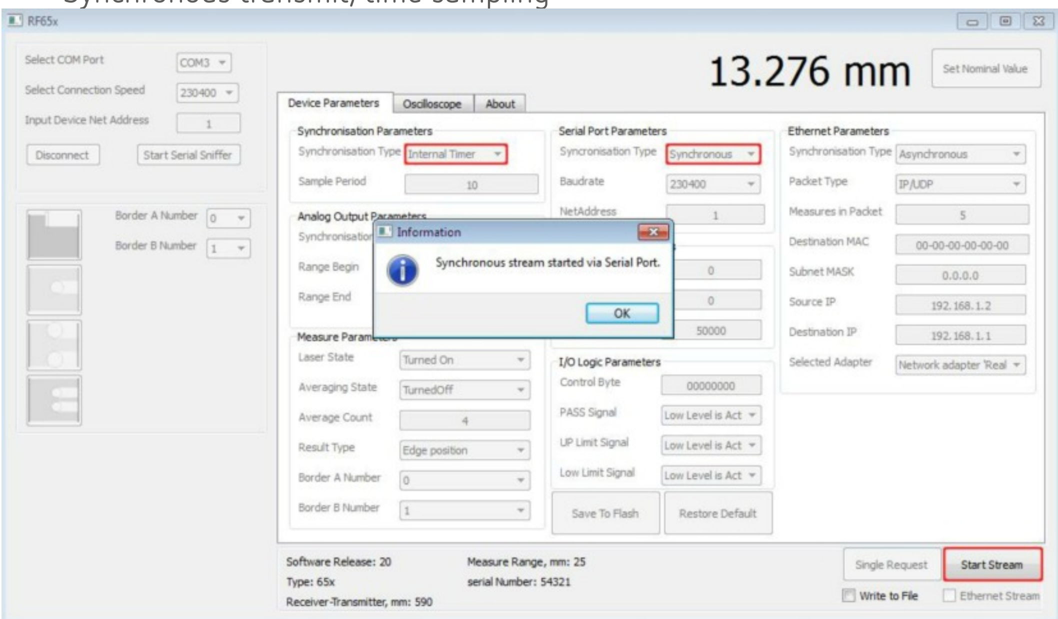
- It is possible to specify the position and zoom of the graph: pressing of the Auto button puts it into the active (passive) state that allows you to position and scale the graph automatically (manually).
- To store the most recently received result as nominal value it is necessary to stop stream and press button Set Nominal Value.

19. Examples of stream setting

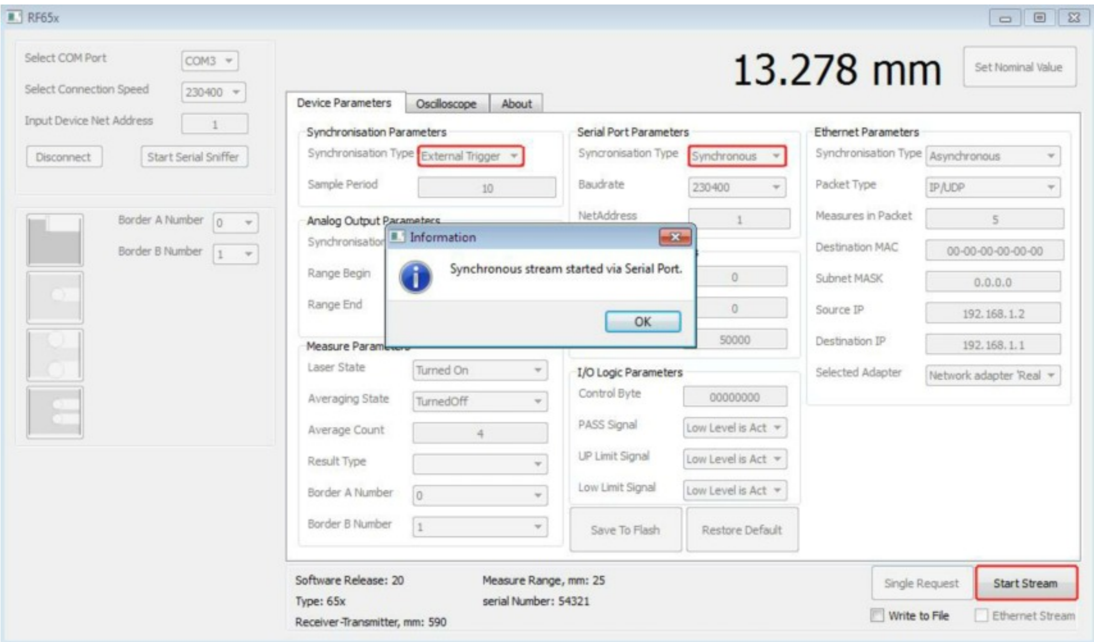
- Asynchronous transmit



- Synchronous transmit, time sampling



· Synchronous transmit, trigger sampling



· If you need to get results stream automatically after turn on the micrometer, make it's configuration and press SaveToFlash button.

20. FDRF65X-SDK.

Laser sensor is supplied together with SDK. The SDK allows user to develop his own software products without going into details of the micrometer communications protocol.

21. Warranty policy

Warranty assurance for the Laser triangulation sensors FDRF603 - 24 months from the date of putting in operation; warranty shelf-life - 12 months

22. Changes

Date	Version	Description
21/01/2015	3.1	1. Section 5 of the table of parameters in the model range FDRF651-50 changed from 50mm to 48mm 2. Section 6 fixed methodology model designation: ET-232 corrected to 232-ET, input IN is placed after all outputs 3. In Section 6, added a table of allowable modificationtions