

**RIFTEK**  
Sensors & Instruments



## ТРИАНГУЛЯЦИОННЫЕ ЛАЗЕРНЫЕ ДАТЧИКИ

**Серия РФ600**

### Руководство по эксплуатации

Логойский тракт, 22, г. Минск  
220090, Республика Беларусь  
тел/факс: +375 17 281 35 13  
info@riftek.com  
www.riftek.com

## Содержание

1.	Меры предосторожности.....	4
2.	Электромагнитная совместимость .....	4
3.	Лазерная безопасность .....	4
3.1.	Датчики класса 3В.....	4
3.2.	Датчики класса 3R .....	5
4.	Назначение .....	5
5.	Основные технические данные .....	6
6.	Пример обозначения при заказе.....	6
7.	Устройство и принцип работы.....	7
8.	Габариты и установка.....	7
8.1.	Габаритные и установочные размеры.....	7
8.2.	Общие требования к установке.....	8
9.	Подключение .....	9
9.1.	Назначение контактов разъемов.....	9
9.2.	Кабеля .....	10
10.	Конфигурационные параметры .....	10
10.1.	Предельное время накопления.....	10
10.2.	Режим выборки .....	11
10.3.	Период выборки.....	11
10.4.	Точка нуля .....	12
10.5.	Режим работы линии AL.....	12
10.6.	Удержание результата .....	12
10.7.	Способ усреднение результата .....	13
10.8.	Количество усредняемых значений/время усреднения.....	13
10.9.	Таблица заводских значений параметров.....	13
11.	Описание интерфейсов RS232 и RS485 .....	13
11.1.	Порт RS232 .....	13
11.2.	Порт RS485 .....	13
11.3.	Режимы передачи данных.....	14
11.4.	Конфигурационные параметры.....	14
11.4.1.	Скорость передачи данных через последовательный порт .....	14
11.4.2.	Сетевой адрес.....	14
11.4.3.	Таблица заводских значений параметров .....	14
11.5.	Протокол обмена .....	14
11.5.1.	Формат последовательной посылки данных .....	14
11.5.2.	Типы сеансов связи.....	14
11.5.3.	Запрос.....	15
11.5.4.	Сообщение, MSG.....	15
11.5.5.	Ответ.....	15
11.5.6.	Поток данных.....	15
11.5.7.	Скорость передачи результата .....	16
11.5.8.	Коды запросов и список параметров .....	16
12.	Описание CAN интерфейса .....	16
12.1.	Режимы передачи данных.....	16
12.2.	Конфигурационные параметры.....	16
12.2.1.	Скорость передачи по CAN интерфейсу.....	16
12.2.2.	Идентификаторы .....	16
12.3.	Таблица заводских значений параметров.....	16
12.4.	Формат передаваемого кадра .....	17
13.	Описание Ethernet интерфейса .....	17
13.1.	Режимы передачи данных.....	17
13.2.	Таблица заводских значений параметров.....	17
13.3.	Формат пакета данных .....	17
13.4.	Структура данных .....	18
14.	Аналоговые выходы .....	18

14.1.	Токовый выход 4...20 мА.....	18
14.2.	Выход по напряжению 0...10В .....	18
14.3.	Конфигурационные параметры.....	19
14.3.1.	Диапазон аналогового выхода .....	19
14.3.2.	Режим работы аналогового выхода .....	19
14.4.	Таблица заводских значений параметров.....	19
15.	Коды запросов и список параметров.....	19
15.1.	Таблица кодов запросов.....	19
15.2.	Список параметров.....	19
15.3.	Примечания.....	21
15.4.	Примеры сеансов связи.....	22
16.	Программа параметризации .....	24
16.1.	Назначение .....	24
16.2.	Установка программы.....	24
16.3.	Установка соединения с датчиком (RS232/RS485).....	24
16.4.	Проверка работоспособности датчика .....	25
16.5.	Подключение по Ethernet интерфейсу.....	26
16.6.	Отображение, накопление и просмотр данных.....	27
16.7.	Настройка и сохранение параметров датчика .....	28
16.7.1.	Настройка параметров.....	28
16.7.2.	Сохранение параметров .....	29
16.7.3.	Сохранение и запись группы параметров.....	29
16.7.4.	Восстановление параметров по умолчанию .....	30
17.	Библиотека RF60X. Описание функций .....	30
17.1.	Подключение к COM-порту (RF60x_OpenPort).....	30
17.2.	Отключение от COM-порта (RF60x_ClosePort) .....	30
17.3.	Идентификация устройства (RF60x_HelloCmd) .....	31
17.4.	Чтение параметров (RF60x_ReadParameter) .....	31
17.5.	Сохранение текущих параметров во FLASH-памяти (RF60x_FlushToFlash).....	32
17.6.	Восстановление параметров по умолчанию (RF60x_RestoreFromFlash) .....	33
17.7.	Защелкивание текущего результата (RF60x_LockResult).....	33
17.8.	Получение результата измерения (RF60x_Measure).....	33
17.9.	Запуск потока измерений (RF60x_StartStream).....	34
17.10.	Остановка потока измерений (RF60x_StopStream).....	34
17.11.	Получение результатов измерений из потока (RF60x_GetStreamMeasure). .....	35
17.12.	Передача пользовательских данных (RF60x_CustomCmd).....	35
17.13.	Функции для работы с датчиками, подключенными к USB с помощью FTDI.....	36
17.14.	Функции для работы с датчиками с Ethernet интерфейсом.....	36
17.14.1.	Открытие порта для получения данных по Ethernet .....	36
17.14.2.	Закрытие порта для получения данных по Ethernet.....	36
17.14.3.	Получение 168-ми результатов измерений из потока.....	36
18.	Примеры.....	38
19.	Приложения .....	40
19.1.	Варианты установки разъема .....	40
20.	Гарантийные обязательства .....	40
21.	Датчики производства РИФТЭК.....	41

## 1. Меры предосторожности

- Используйте напряжение питания и интерфейсы, указанные в спецификации на датчик.
- При подсоединении/отсоединении кабелей питания датчика должно быть отключено.
- Не используйте датчики вблизи мощных источников света.
- Для получения стабильных результатов после включения питания необходимо выдержать порядка 20 минут для равномерного прогрева датчика.

## 2. Электромагнитная совместимость

Датчики разработаны для использования в промышленности и соответствуют следующим стандартам:

- EN 55022:2006 Оборудование информационных технологий. Характеристики радиопомех. Пределы и методы измерений.
- EN 61000-6-2:2005 Электромагнитная совместимость. Общие стандарты. Помехоустойчивость к промышленной окружающей среде.
- EN 61326-1:2006 Электрооборудование для измерения, управления и лабораторного использования. Требования к электромагнитной совместимости. Общие требования.

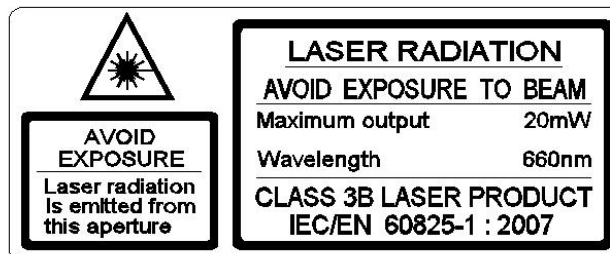
## 3. Лазерная безопасность

Датчики соответствуют следующим классам лазерной безопасности по IEC 60825-1:2007

Длина волны	660 нм	
Мощность излучения	≤4,8 мВт	≤20 мВт
Класс безопасности	3R	3B

### 3.1. Датчики класса 3B

В датчиках установлен полупроводниковый лазер с непрерывным излучением и длиной волны 660 нм. Максимальная выходная мощность 20 мВт. Датчики относятся к классу 3B лазерной безопасности. На корпусе датчиков размещена предупредительная этикетка:



При работе с датчиком необходимо соблюдать следующие меры безопасности:

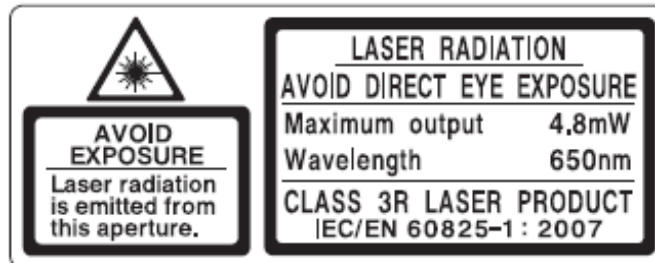
- не направляйте лазерный луч на людей;
- не смотрите на лазерный луч через оптические инструменты;
- устанавливайте датчик таким образом, чтобы лазерный луч располагался выше или ниже уровня глаз;

- устанавливайте датчик таким образом, чтобы лазерный луч не попадал на зеркальную поверхность;
- при работе с датчиком рекомендуется использовать защитные очки;
- не смотрите на лазерный луч, выходящий из датчика, и луч, отраженный от зеркальной поверхности;
- не разбирайте датчик;
- используйте защитный экран, установленный на датчике для блокирования выходящего пучка;
- используйте функцию отключения лазера в случае опасности.

**Примечание:** датчики класса 3B поставляются только как OEM продукт. Всю ответственность за соблюдение требований лазерной безопасности несет потребитель.

### 3.2. Датчики класса 3R

В датчиках установлен полупроводниковый лазер с непрерывным излучением и длиной волны 660 нм. Максимальная выходная мощность 5 мВт. Датчики относятся к классу 3R лазерной безопасности. На корпусе датчиков размещена предупреждающая этикетка:



При работе с датчиком необходимо соблюдать следующие меры безопасности:

- не направляйте лазерный луч на людей;
- не смотрите на лазерный луч через оптические инструменты;
- устанавливайте датчик таким образом, чтобы лазерный луч располагался выше или ниже уровня глаз;
- при работе с датчиком рекомендуется использовать защитные очки;
- не смотрите на лазерный луч
- не разбирайте датчик.

## 4. Назначение

Триангуляционные лазерные датчики предназначены для бесконтактного измерения и контроля положения, перемещения, размеров, профиля поверхности, деформаций, вибраций, сортировки, распознавания технологических объектов, измерения уровня жидкостей и сыпучих материалов.

Серия включает 13 моделей датчиков с измерительным диапазоном от 10 до 2500 мм и базовым расстоянием от 230 до 1000 мм.

Датчики разбиты на три группы. Первая группа включает датчики с увеличенным базовым расстоянием и небольшим измерительным диапазоном, вторая группа – датчики со средней базой и измерительным диапазоном, третья группа – датчики с большим измерительным диапазоном.

В датчиках доступны две опции - КРАСНЫЙ или СИНИЙ лазер. Использование синих лазеров вместо традиционных красных существенно расширяет воз-

возможности датчиков, в частности, при контроле высокотемпературных объектов, а также органических материалов.

Возможны также заказные конфигурации датчиков с параметрами, отличающимися от параметров, указанных ниже.

## 5. Основные технические данные

Модель РФ600-	X/10	X/30	X/40	X/100	X/250	X/500	X/600	X/1000	X/1000	X/1500	X/2000	X/2500
Базовое расстояние X, мм	230	300	330	500	230	300, 1000	230	1300	380	390	410	420
Диапазон, мм	10	30	40	100	250	500	600	1000	1000	1500	2000	2500
Линейность, %	±0.1 от диапазона									±0.2		
Разрешение, %	0.01 диапазона (только для цифрового выхода)									0.03		
Температурный дрейф	0,02% диапазона/°C											
Максимальная частота обновления данных	9,4 кГц											
Источник излучения	видимый красный полупроводниковый лазер, длина волны 660 нм или ультрафиолетовый полупроводниковый лазер, длина волны 405 нм (версия с СИНИМ лазером)											
мощность излучения	≤4,8 мВт									≤20 мВт		
класс безопасности	3R (IEC60825-1)									3B (IEC60825-1)		
Выходной интерфейс	цифровой	RS232 (макс. 460,8 Кбит/с) или RS485 (макс. 921,6 Кбит/с) или RS232 и CAN V2.0B (макс. 1Мбит/с) или Ethernet и (RS32 или RS485)										
	аналоговый	4...20 мА (нагрузка ≤ 500 Ом) или 0...10 В										
Вход синхронизации	2,4 – 5 В (CMOS, TTL)											
Логический выход	программируемые функции, NPN: 100 мА max; 40 В max											
Напряжение питания	9 ...36 В											
Потребляемая мощность	1,5..2 Вт											
Устойчивость к внешним воздействиям	Класс защиты	IP67										
	Уровень вибраций	20g/10...1000Гц, 6 часов для каждой из XYZ осей										
	Ударные нагрузки	30 g/6 мс										
	Окружающая температура, °C	-10...+60, (-30...+60 для датчиков со встроенным нагревателем)										
	Окружающая освещенность, люкс	30000										
	Относительная влажность	5-95% (без конденсации)										
	Температура хранения, °C	-20...+70										
Материал корпуса	алюминий											
Вес (без кабеля)	500 грамм											

## 6. Пример обозначения при заказе

РФ600(BLUE)-X/D-SERIAL-ANALOG-IN-AL-CC(90X)(R)-M-H

Символ	Наименование
(BLUE)	Опция лазера - синий (405 нм)
X	Базовое расстояние (начало диапазона), мм
D	Рабочий диапазон, мм
SERIAL	Тип последовательного интерфейса: RS232 - 232, или RS485 - 485, или (CAN и RS232) - CAN, или (Ethernet и RS232) – ET-232 или (Ethernet и RS485) - ET-485
ANALOG	Наличие аналогового выхода по току ( I ) или по напряжению ( U ) <b>Примечание:</b> 1) выход I - только для датчиков с RS232 или RS485 2) выход U – только для датчиков с RS232 или RS485 или CAN
IN	Наличие входа синхронизации
AL	Программируемый пользователем сигнал имеет несколько назначений. Может использоваться как: 1) логический выход (индикация наличия объекта в рабочем диапазоне); 2) линия взаимной синхронизации двух и более датчиков

	3) линия аппаратной установки начала отсчета 4) линия аппаратного выключения/включения лазера
CC(90X)(R)	Кабельный ввод - CG, либо разъем - CC (Binder 712, IP67). <b>Примечание 1:</b> датчики с CAN или Ethernet интерфейсом содержат 2 разъема или два кабельных ввода. <b>Примечание 2:</b> опция 90X – признак углового кабельного разъема <b>Примечание 3:</b> опция R – признак специального робототехнического кабеля
M	Длина кабеля, м
H	Наличие встроенного нагревателя

**Пример.** РФ600-380/1000-232-I-IN-AL-CCR90A-3 – датчик с базовым расстоянием – 380 мм, диапазоном - 1000мм, последовательный порт RS232, есть токовый выход 4...20мА, есть вход синхронизации и AL-выход, угловой кабельный разъем, вариант установки разъема "А", робот-кабель, длина кабеля 3 м.

## 7

## 7. Устройство и принцип работы

В основу работы датчика положен принцип оптической триангуляции, рис.1.

Излучение полупроводникового лазера 1 фокусируется объективом 2 на объекте 6. Рассеянное на объекте излучение объективом 3 собирается на CMOS-линейке 4. Процессор сигналов 5 рассчитывает расстояние до объекта по положению изображения светового пятна на линейке 4.

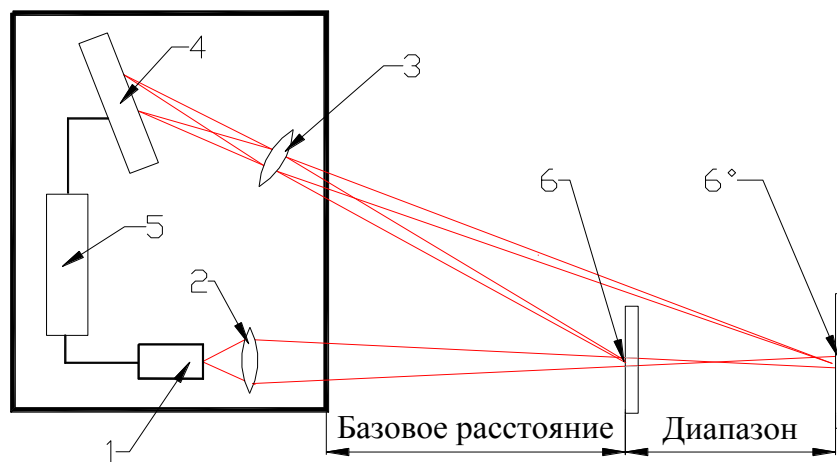


Рисунок 1

## 8. Габариты и установка

### 8.1. Габаритные и установочные размеры

Габаритные и установочные размеры датчика РФ600 показаны на рис. 2. Корпус датчика выполнен из анодированного алюминия. На передней панели корпуса расположены два окна: одно – выходное, другое – для приема излучения, отраженного от контролируемого объекта. Для установки в оборудование корпус датчика содержит крепежные отверстия.

Датчик содержит разъем или кабельный ввод. Датчики с CAN или Ethernet интерфейсом содержат два разъема или два кабельных ввода.

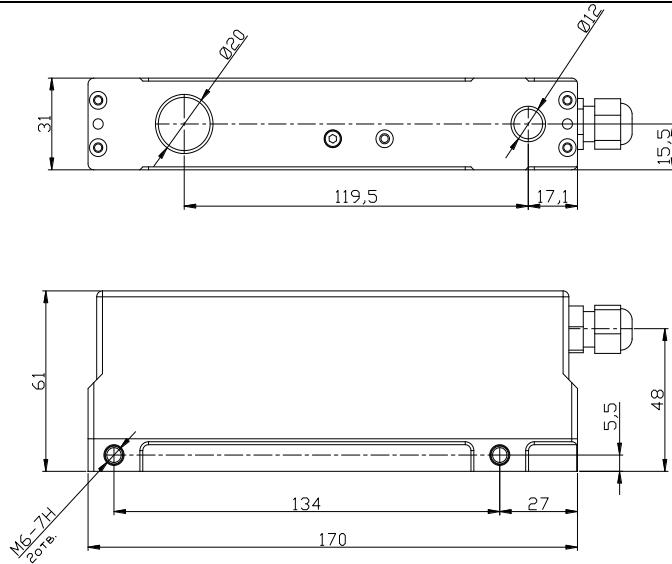


Рисунок 2. Датчик РФ600 с кабельным вводом

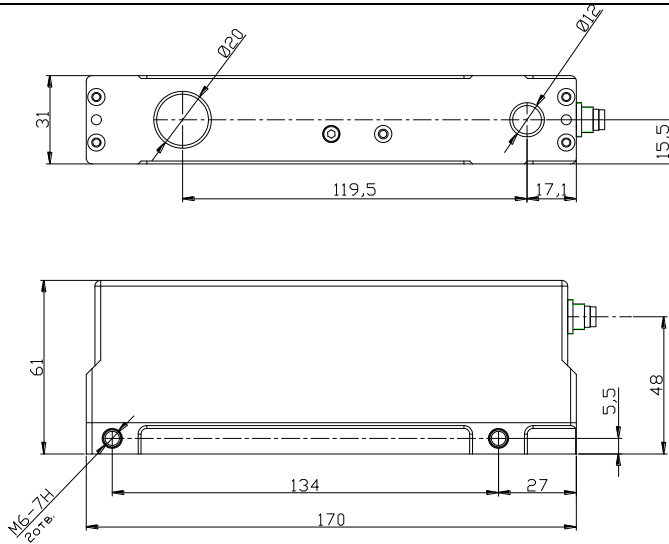


Рисунок 2.1. Датчик РФ600 с разъемом

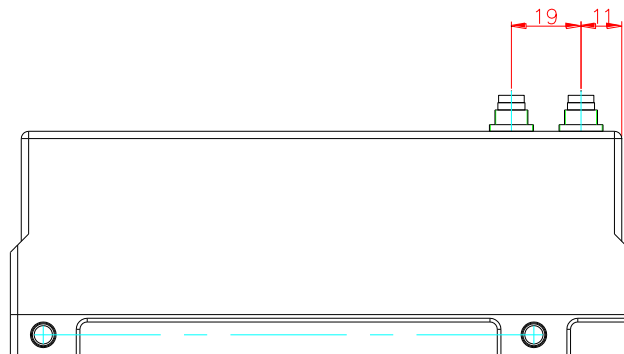


Рисунок 2.2. Датчик РФ600 с разъемом

## 8.2. Общие требования к установке

Датчик устанавливается таким образом, чтобы контролируемый объект располагался в зоне рабочего диапазона датчика. Кроме того, в области прохож-



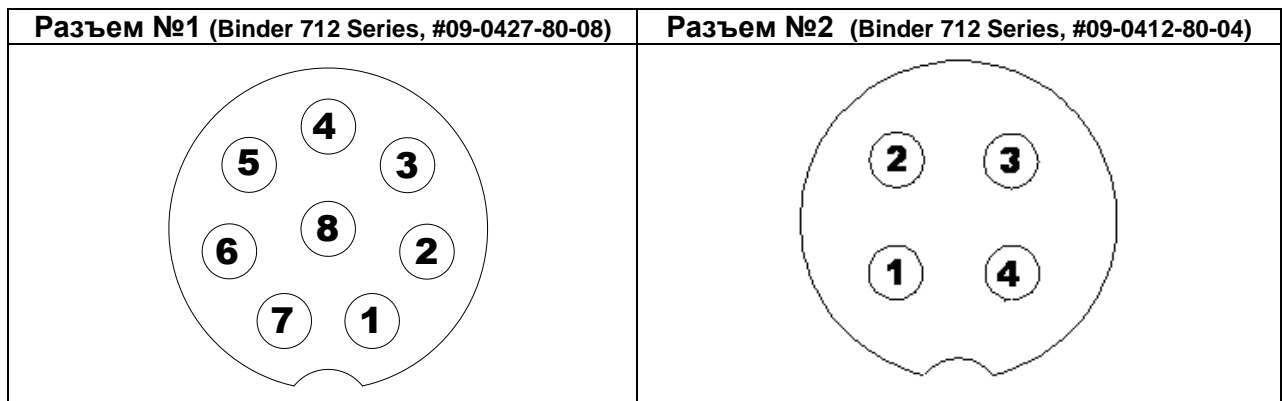
дения падающего на объект и отраженного от него излучения не должно находиться посторонних предметов.

При контроле объектов сложной формы и текстуры необходимо минимизировать попадание зеркальной составляющей отраженного излучения во входное окно датчика.

## 9. Подключение

### 9.1. Назначение контактов разъемов

Вид со стороны контактов разъемов, установленных на датчик, показан на рисунках (на датчике с одним разъемом установлен разъем №1):



Назначение контактов приведено в таблицах:

#### Разъем №1

Модель датчика	Номер контакта	Назначение
(CAN/ET)-232-U/I-IN-AL	1	IN
	2	Gnd (питание)
	3	TXD
	4	RXD
	5	Gnd (Общий для сигналов)
	6	AL
	7	U/I
	8	Питание U+
(ET)-485-U/I-IN-AL	1	IN
	2	Gnd (питание)
	3	DATA+
	4	DATA-
	5	Gnd (Общий для сигналов)
	6	AL
	7	U/I
	8	Питание U+

#### Разъем №2

Модель датчика	Номер контакта	Назначение
ET-	1	TX+
	2	TX-
	3	RX+
	4	RX-
CAN-	1	CAN_H
	2	CAN_L
	3	
	4	GND

## 9.2. Кабеля

Назначение проводников кабелей приведено в таблице:  
Кабель №1

Модель датчика	Номер контакта разъема		Назначение	Цвет провода
(CAN/ET)-232-U/I-IN-AL	свободный проводник	-	Power U+	Красный
	свободный проводник	-	Gnd (питание)	Коричневый
	DB9	2	2TXD	Зеленый
	DB9	3	3RXD	Желтый
	свободный проводник	-	U/I	Синий
	свободный проводник	-	IN	Белый
(ET)-485-U/I-IN-AL	свободный проводник	-	AL	Розовый
	свободный проводник	-	Gnd (Общий для сигналов)	Серый
	свободные провода	5	Power U+	Красный
			Gnd (питание)	Коричневый
			DATA+	Зеленый
			DATA-	Желтый
			U/I	Синий
			IN	Белый
			AL	Розовый
			Gnd (Общий для сигналов)	Серый

Кабель №2

Модель датчика	Номер контакта разъема		Назначение	Цвет провода
ET-	RJ-45	1	TX+	Бело-оранжевый
		2	TX-	Оранжевый
		3	RX+	Бело-зеленый
		4		
		5		
		6	RX-	Зеленый
		7		
		8		
CAN-	свободные провода		CAN_H	Бело-оранжевый
			CAN_L	Оранжевый
			Gnd	Зеленый

## 10. Конфигурационные параметры

Характер работы датчика определяют его конфигурационные параметры, изменение которых производится только путем передачи команд через последовательный порт RS232 или RS485. Основные параметры:

### 10.1. Предельное время накопления

Интенсивность отраженного излучения, поступающего в датчик, зависит от свойств поверхности контролируемого объекта, поэтому мощность излучения лазера и время накопления излучения, падающего на CMOS-линейку, автоматически регулируются с целью получения оптимального сигнала и достижения максимальной точности измерения.

Параметр "предельное время накопления" задает величину предельно допустимого времени накопления линейки. Если интенсивность принимаемого датчиком излучения настолько мала, что за время накопления, равное предельному времени, не получен результат, датчик передает нулевое значение.

**Примечание 1.** От величины времени накопления приемной линейки зависит частота обновления результата. Максимальная частота (9,4 кГц) достигается для времени накопления  $\leq 106$  мкс (минимально возможное время накопления – 0,1

мкс). При увеличении времени накопления свыше 106 мкс частота обновления результата пропорционально уменьшается.

**Примечание 2.** Увеличение данного параметра расширяет возможности контроля слабоотражающих (диффузная составляющая) поверхностей, однако уменьшает частоту обновления результата измерения и увеличивает влияние внешней засветки (фона) на точность измерения. Предельное время накопления – 3200 мкс

**Примечание 3.** Уменьшение данного параметра позволяет повысить результирующую частоту обновления результата, но может привести к снижению точности измерения.

## 10.2. Режим выборки

Данный параметр задает один из двух вариантов выборки результата при работе датчика в режиме потока данных:

- выборка по времени;
- выборка по внешнему входу.

При установке режима выборки по времени датчик автоматически по последовательному интерфейсу передает результат измерения в соответствии с заданным интервалом времени (периодом выборки).

При установке режима выборки по внешнему входу датчик передает результат при переключении входа внешней синхронизации (вход IN) с учетом установленного коэффициента деления.

## 10.3. Период выборки

Если установлен режим выборки по времени, то параметр "период выборки" определяет интервал времени, через который датчик должен автоматически передавать результат измерения. Значение интервала времени задается в дискретах по 0.01мс. **Например**, для значения параметра, равного 100, данные по последовательному интерфейсу передаются с периодом  $0,01 \cdot 100 = 1$  мс.

Если установлен режим выборки по внешнему входу, то параметр "период выборки" определяет коэффициент деления для входа внешней синхронизации. **Например**, если параметр равен 100, данные по последовательному интерфейсу передаются с приходом на вход IN датчика каждого 100-го импульса синхронизации.

**Примечание 1.** Необходимо отметить, что параметры "режим выборки" и "период выборки" управляют только передачей данных. Алгоритм работы датчика построен таким образом, что собственно измерения выполняются постоянно с максимально возможным темпом, определяемым временем накопления, результат измерения заносится в буфер и хранится в нем до поступления нового результата. Указанные параметры определяют способ выдачи результата из этого буфера.

**Примечание 2.** Если для приема результата используется последовательный интерфейс, то при задании малых интервалов периода выборки следует учитывать время, необходимое для передачи данных на выбранной скорости передачи. Если время передачи превосходит период выборки, то именно оно будет определять темп передачи данных. Расчет времени, необходимого для передачи результата, представлен в п. [11.5.7.](#)

**Примечание 3.** Необходимо учитывать, что датчики отличаются некоторым разбросом параметров внутреннего генератора, что влияет на точность периода выборки по времени.

## 10.4. Точка нуля

Данный параметр задает начало отсчета в абсолютной системе координат в любой точке в пределах рабочего диапазона. Начало отсчета может устанавливаться как программно, путем передачи соответствующей команды, так и аппаратно, путем подачи нулевого потенциала на вход AL (предварительно данный вход должен быть установлен в режим 3). При изготовлении датчика базовое расстояние задается с некоторой неопределенностью и при необходимости возможно более точное задание начала координат.

## 10.5. Режим работы линии AL

Данная линия может работать в одном из четырех режимов, определяемых значением конфигурационного параметра:

- режим индикации выхода за диапазон;
- режим взаимной синхронизации;
- режим аппаратной установки начала отсчета;
- режим аппаратного выключения/включения лазера.

В режиме "Индикация выхода за диапазон" на линии AL устанавливается логическая "1", если контролируемый объект находится в пределах рабочего диапазона датчика (в пределах заданного окна в диапазоне), и логический "0", если в пределах рабочего диапазона (в пределах заданного окна) объект отсутствует. **Например**, в таком режиме данную линию можно использовать для управления исполнительным механизмом (реле), срабатывающим при нахождении (отсутствии) объекта в заданном диапазоне (рис. 3.1.).

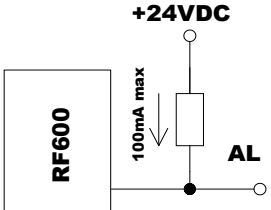
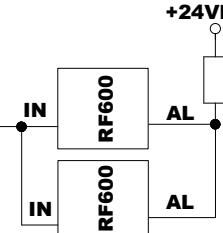
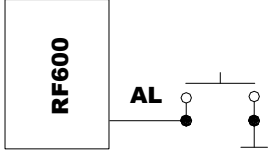
Режим "Взаимная синхронизация" позволяет синхронизировать моменты измерения двух и более датчиков. Режим удобно использовать при контроле одного объекта несколькими датчиками, например, при измерении толщины. На аппаратном уровне синхронизация датчика осуществляется путем объединения линий AL (рис. 3.2.).

**Примечание.** Необходимо учитывать, что в режиме "Взаимная синхронизация" частота измерения уменьшается пропорционально количеству синхронизируемых датчиков.

В режиме "Аппаратная установка начала отсчета" подача нулевого потенциала на линию AL устанавливает начало координат в текущую точку (рис.3.3.).

В режиме "Аппаратное выключение/включение лазера" подача нулевого потенциала на линию AL выключает/включает лазер (рис. 3.3.)

Примеры использования линии AL:

Индикация выхода за диапазон	Взаимная синхронизация	Установка начала отсчета. Включение/выключение лазера
		
Рисунок 3.1	Рисунок 3.2	Рисунок 3.3

## 10.6. Удержание результата

Если датчик не обнаруживает объект или если достоверный результат не может быть получен, то передается нулевое значение. Данный параметр задает

время, в течение которого передается не нулевое значение, а последний достоверный результат. Дискретность задания времени удержания – 5 мс.

## 10.7. Способ усреднение результата

Данный параметр определяет один из двух способов усреднения результатов измерений, выполняемых непосредственно в датчике:

- усреднение по количеству результатов
- усреднение по времени

При установке усреднения по количеству результатов вычисляется скользящее среднее.

При установке усреднения по времени получаемые результаты усредняются в течение заданного интервала времени

## 10.8. Количество усредняемых значений/время усреднения

Данный параметр определяет количество исходных результатов, по которым берется среднее для формирования выходного значения (усреднение по количеству результатов) или период времени усреднения (дискретность - 5 мс)

Применение усреднения позволяет уменьшить выходной шум и повысить разрешающую способность датчика.

Усреднение по количеству результатов не влияет на темп обновления данных в выходном буфере датчика.

При усреднении по времени данные в выходном буфере обновляются с темпом, равным периоду усреднения.

**Примечание.** Максимальное значение параметра -127.

## 10.9. Таблица заводских значений параметров

Датчики поставляются с параметрами, значения которых представлены в таблице:

Наименование параметра	Значение
Предельное время накопления	3200 (3,2 мс)
Режим выборки	по времени
Период выборки	500 (5 мс)
Точка нуля	начало диапазона
Режим линии AL	1
Время удержания результата	5 мс
Способ усреднения результата	по количеству
Количество усредняемых значений	1

Параметры хранятся в энергонезависимой памяти датчика. Корректное изменение параметров производится с помощью программы параметризации, поставляемой с датчиком, либо программой пользователя.

# 11. Описание интерфейсов RS232 и RS485

## 11.1. Порт RS232

Порт RS232 обеспечивает подключение “точка-точка” и позволяет подключать датчик непосредственно к RS232 порту компьютера, либо контроллера.

## 11.2. Порт RS485

Порт RS485 в соответствии с принятым сетевым протоколом и аппаратными возможностями позволяет подключить датчики к одному устройству сбора информации по схеме “общая шина”.

## 11.3. Режимы передачи данных

По данным интерфейсам результаты можно получить двумя способами:

- по разовым запросам;
- автоматически потоком данных.

## 11.4. Конфигурационные параметры

### 11.4.1. Скорость передачи данных через последовательный порт

Данный параметр определяет скорость передачи данных по последовательному интерфейсу в дискретах по 2400 бит/с. *Например*, значение параметра, равное 4, задает скорость передачи  $2400 \cdot 4 = 9600$  бит/с.

**Примечание.** Максимальная скорость передачи по интерфейсу RS232 – 460,8 кбит/с, по интерфейсу RS485 – 921,6 кбит/с.

### 11.4.2. Сетевой адрес.

Данный параметр определяет сетевой адрес датчика, оснащенного интерфейсом RS485.

**Примечание.** Сетевой протокол передачи данных предполагает наличие в сети одного “мастера”, которым может быть компьютер или другое устройство сбора информации, и от 1 до 127 “помощников” (датчик серии РФ600), поддерживающих этот протокол.

Каждому “помощнику” задается уникальный для данной сети идентификационный код — адрес устройства. Адрес устройства используется при формировании запросов по сети. Каждый из помощников принимает запросы, содержащие его личный адрес, а также адрес “0”, который является широковещательным и может быть использован для формирования групповых команд, например для одновременного защелкивания значений всех датчиков, а также при работе с одним датчиком (как с портом RS232, так и с портом RS485).

### 11.4.3. Таблица заводских значений параметров

Наименование параметра	Значение
Скорость передачи данных (интерфейс RS232 или RS485)	9600
Сетевой адрес	1
Режим передачи данных	по запросу

## 11.5. Протокол обмена

### 11.5.1. Формат последовательной посылки данных

Посылка данных имеет следующий формат:

1 старт-бит	8 бит данных	1 бит нечетности	1 стоп-бит
-------------	--------------	------------------	------------

Бит нечетности является дополнением 8-ми бит данных до четности.

### 11.5.2. Типы сеансов связи

Протокол обмена построен на сеансах связи, которые инициируются только внешним устройством, “мастером” (ПК, контроллер). Существуют сеансы связи двух видов, которые имеют следующую структуру:

- 1) “запрос”, [“сообщение”] — [“ответ”], в квадратных скобках указаны необязательные элементы
- 2) “запрос” — “поток данных” — [“запрос”].

### 11.5.3. Запрос

“Запрос” (INC) — это двухбайтная посылка, полностью определяющая сеанс обмена. Посылка “запроса” - единственная из всех посылок сеанса связи, в которой в первом посылаемом байте старший бит установлен в 0, поэтому она служит для синхронизации начала сеанса. Кроме того, она содержит адрес устройства (ADR), код запроса (COD) и, возможно, сообщение [MSG].

Формат “запроса”:

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

### 11.5.4. Сообщение, MSG.

“Сообщение” — это пакет данных, который может передаваться в сеансе связи “мастером”.

Все посылки пакета сообщения содержат 1 в старшем разряде. Данные в посылках передаются потетрадно. При передаче байта сначала передается младшая тетрада, затем старшая. При передаче многобайтных значений передача начинается с младшего байта.

Формат двух посылок данных “сообщения” для передачи байта DAT(7:0):

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

### 11.5.5. Ответ

“Ответ” — это пакеты данных, которые могут передаваться в сеансе связи “помощником”.

Все посылки пакета сообщения содержат 1 в старшем разряде. Данные в посылках передаются потетрадно. При передаче байта сначала передается младшая тетрада, затем старшая. При передаче многобайтных значений передача начинается с младшего байта.

При передаче “ответа” в посылку данных добавляются:

- бит (SB), характеризующий обновление результата. Если бит равен "1" это означает, что результат в буфере передачи обновлен, если бит равен "0" - передается не обновленный результат (см. Примечание 1, п. [10.3.](#)). При передаче параметров бит SB равен "0";
- два бита циклического двоичного счетчика пакетов (CNT). Значения битов счетчика пакетов одинаковы для всех посылок одного пакета. Значение счетчика пакетов инкрементируется при передаче каждого пакета и используется для формирования (сборки) пакета, а также контроля потери пакетов при приеме потока данных.

Формат двух посылок данных “ответа” для передачи байта DAT(7:0):

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

### 11.5.6. Поток данных

“Поток данных” — это бесконечная последовательность пакетов данных, передаваемая “помощником” “мастеру”, которая может быть прервана новым запросом. При передаче “потока данных” один из “помощников” полностью захватывает канал передачи данных, однако при выдаче “мастером” любого нового запро-

са по любому адресу передача потока прекращается. Существует и специальный запрос прекращения потока.

### 11.5.7. Скорость передачи результата

Скорость передачи результата (Output rate, "OR") зависит от установленной скорости интерфейса (Baud rate, "BR") и рассчитывается следующим образом:

$$OR = 1 / (44/BR + 1 * 10^{-5}) \text{ Гц.}$$

Например, при BR=460800 бит/с OR = 9,4 кГц.

### 11.5.8. Коды запросов и список параметров

Коды запросов и список параметров представлены в главе [15](#).

## 16

## 12. Описание CAN интерфейса

CAN интерфейс используется только для получения данных. Параметризация датчиков осуществляется по интерфейсу RS232.

### 12.1. Режимы передачи данных

Датчик может работать в режимах:

- Нет передачи
- Режим по запросу. В этом режиме каждый из датчиков принимает кадр удаленного запроса данных (Remote Frame), содержащий его идентификатор, и отвечает кадром данных (Data Frame) с тем же самым идентификатором.
- Режим потока. При работе в режиме потока каждый из датчиков автоматически передает кадр данных (Data Frame) со своим идентификатором в соответствии с установленным режимом выборки по времени или по внешнему входу (см. п. [10.2](#)) и соответствующим периодом выборки (см. п. [10.3](#)).

### 12.2. Конфигурационные параметры

#### 12.2.1. Скорость передачи по CAN интерфейсу

Данный параметр определяет скорость передачи данных по CAN интерфейсу в дискретах по 5000 бит/с. *Например*, значение параметра, равное 50, задает скорость передачи 5000\*50 = 250000 бит/с.

**Примечание.** Максимальная скорость передачи по CAN интерфейсу – 1 Мбит/с.

#### 12.2.2. Идентификаторы

Датчик, оборудованный портом CAN 2.0B, поддерживает обмен данными стандартными кадрами (с 11-разрядным идентификаторами) и расширенными кадрами (с 29-разрядным идентификаторами). Каждому датчику задается стандартный или расширенный идентификатор, уникальный для данной сети. Количество датчиков в сети – до 112.

### 12.3. Таблица заводских значений параметров

Наименование параметра	Значение
Скорость передачи данных, CAN интерфейс	125 кб/с
Стандартный идентификатор	7FFh
Расширенный идентификатор	1FFFFFFFh
Состояние интерфейса	включен
Режим работы	по запросу



## 12.4. Формат передаваемого кадра

Датчик передает кадр длиной 8 байт:

- байт 0: тип устройства
- байт 1: = 0 - зарезервирован
- байт 2: младший байт серийного номера
- байт 3: старший байт серийного номера
- байт 4: младший байт рабочего диапазона
- байт 5: старший байт рабочего диапазона
- байт 6: младший байт результата
- байт 7: старший байт результата

Расчет результата производится по формуле (1), (см. п. [15.3](#))

## 13. Описание Ethernet интерфейса

Ethernet интерфейс используется только для передачи данных. Параметризация датчиков осуществляется по интерфейсу RS232 или RS485

### 13.1. Режимы передачи данных

Датчик может работать в режимах:

- Нет передачи.
- Режим потока. Сначала производится заполнение внутреннего буфера передачи датчика измеренными данными в соответствии с установленным режимом выборки по времени или по внешнему входу (см. п. [10.2](#)) и соответствующим периодом выборки (см. п. [10.3](#)). После заполнения буфера (размер буфера – 168 измерений) датчик автоматически передает в сеть UDP пакет с данными, накопленными в этом буфере передачи.

### 13.2. Таблица заводских значений параметров

Наименование параметра	Значение
IP-адрес получателя	255.255.255.255
IP-адрес шлюза	192.168.0.1
Маска подсети	255.255.255.255
IP-адрес источника	192.168.0.3
Состояние интерфейса	включен
Режим передачи данных	выборка по времени

### 13.3. Формат пакета данных

Датчик передает IP порту назначения 603 пакет длиной 512 байт:

- байт 0, байт 1: 1-е измерение
- байт 2, : слово состояния для 1-го измерения
- байт 3, байт 4: 2-е измерение
- байт 5, : слово состояния для 2-го измерения
- .....
- байт 501, байт 502: 168-е измерение
- байт 503, : слово состояния для 168-го измерения
- байт 504, байт 505: серийный номер датчика
- байт 506, байт 507: базовое расстояние датчика
- байт 508, байт 509: диапазон измерения датчика
- байт 510, : циклический счетчик номера пакета
- байт 511, : контрольная сумма пакета

## 13.4. Структура данных

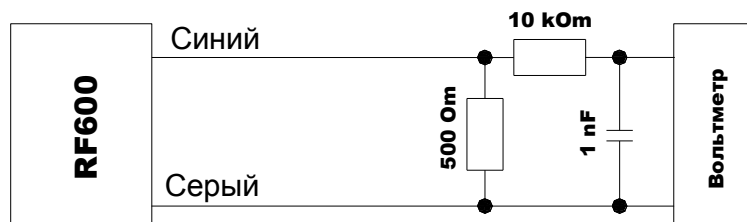
- результат измерения передается 16-ти разрядным словом, расчет результата производится по формуле (1), см. п. [15.3](#).
- размер слова состояния – 1 байт. Состояние бита 0 характеризует обновление результата. Если бит равен "1", то это означает, что к моменту прихода импульса внешней синхронизации (начала нового периода выборки) датчик обновил результат измерений. Если бит равен "0", то передан не обновленный результат. Биты 7...1 слова состояния зарезервированы и равны "0";
- базовое расстояние датчика передается 16-ти разрядным словом с дискретностью 1 мм
- диапазон измерения датчика передается 16-ти разрядным словом с дискретностью 1 мм;
- циклический счетчик номера пакета имеет размерность один байт. Значение счетчика инкрементируется при передаче каждого пакета и используется для контроля потери пакетов при приеме данных;
- контрольная сумма пакета имеет размерность один байт и вычисляется как исключающее ИЛИ всех байт в пакете.

## 14. Аналоговые выходы

Изменение сигнала на аналоговом выходе происходит синхронно с изменением результата, передаваемого по последовательному интерфейсу.

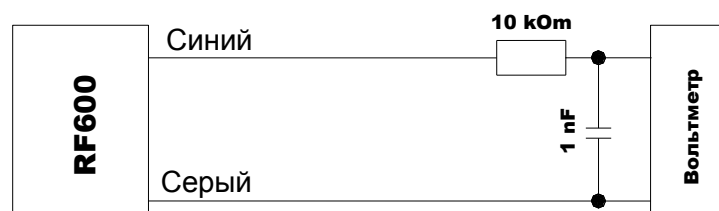
### 14.1. Точковый выход 4...20 мА

Схема подключения показана на рисунке. Значение нагрузочного резистора не должно превышать 500 Ом. Для уменьшения шума перед измерительным прибором рекомендуется установить RC фильтр. Величина конденсатора фильтра указана для максимальной частоты выборки датчика (9,4 кГц) и пропорционально увеличивается при уменьшении частоты.



### 14.2. Выход по напряжению 0...10В

Схема подключения показана на рисунке. Для уменьшения шума перед измерительным прибором рекомендуется установить RC фильтр. Величина конденсатора фильтра указана для максимальной частоты выборки датчика (9,4 кГц) и пропорционально увеличивается при уменьшении частоты.



## 14.3. Конфигурационные параметры

### 14.3.1. Диапазон аналогового выхода

При работе с аналоговым выходом для повышения разрешения можно воспользоваться функцией "окно в рабочем диапазоне", которая позволяет выбрать в рабочем диапазоне датчика окно требуемых размеров и положения, в пределах которого будет масштабироваться весь диапазон аналогового выходного сигнала.

**Примечание.** В случае если начало диапазона аналогового сигнала задать большим по величине, чем конец этого диапазона, то это изменит направление нарастания аналогового сигнала.

### 14.3.2. Режим работы аналогового выхода

При работе с функцией "окно в рабочем диапазоне" данный параметр задает режим работы аналогового выхода.

Аналоговый выход может находиться:

- в оконном режиме или
- в полном режиме.

**"Оконный режим".** Весь диапазон аналогового выхода масштабируется в заданном окне. Вне окна на аналоговом выходе "0".

**"Полный режим".** Весь диапазон аналогового выхода масштабируется в заданном окне (рабочий диапазон). Вне заданного окна весь диапазон аналогового выхода автоматически масштабируется на весь рабочий диапазон датчика (диапазон чувствительности).

## 14.4. Таблица заводских значений параметров

Наименование параметра	Значение
Диапазон аналогового выхода	рабочий диапазон датчика
Режим работы аналогового выхода	оконный

## 15. Коды запросов и список параметров

### 15.1. Таблица кодов запросов

Код запроса	Описание	Сообщение (размер в байтах)	Ответ (размер в байтах)
01h	Идентификация устройства	—	-тип устройства (1) -версия ПО (1) -серийный номер (2) -базовое расстояние (2) -диапазон (2)
02h	Чтение параметра	-код параметра (1)	-значение параметра (1)
03h	Запись параметра	-код параметра (1) -значение параметра (1)	—
04h	Сохранение текущих параметров во FLASH-памяти	-константа AAh (1)	-константа AAh (1)
04h	Восстановление во FLASH-памяти значений параметров по умолчанию	-константа 69h (1)	-константа 69h (1)
05h	Защелкивание текущего результата	—	—
06h	Запрос результата	—	-результат (2)
07h	Запрос потока результатов	—	-поток результатов (2)
08h	Прекратить передачу потока	—	—

### 15.2. Список параметров

Код параметра	Наименование	Значения
---------------	--------------	----------

00h	Включение датчика	1 — лазер включен, выполняются измерения (по умолчанию); 0 — лазер отключен, датчик находится в энергосберегающем режиме.
01h	Включение аналогового выхода	1 — аналоговый выход включен; 0 — аналоговый выход отключен. В случае если модификация датчика не оснащена аналоговым выходом, при попытке записать в этот бит 1 он останется в состоянии 0.
02h	Управление усреднением, выборкой, режимами AL - выхода	x,x,M,C,M1,M0,R,S – контрольный байт, задающий режим работы усреднения – бит M, CAN интерфейса - бит C, логического выхода - биты M0 и M1, аналогового выхода - бит R, и режим выборки - бит S; биты x – не используются; бит M: 0 — режим усреднения измеренных значений по количеству (по умолчанию); 1 — режим усреднения измеренных значений по времени (по 5 мс); бит C: 0 – режим CAN интерфейса по запросу (по умолчанию); 1 – режим CAN интерфейса с синхронизацией по времени или по внешнему входу. биты M1:M0: 00 – режим индикации выхода за диапазон (по умолчанию); 01 – режим взаимной синхронизации; 10 - режим обнуления результата. 11- режим выключения/включения лазера бит R: 0 – оконный режим (по умолчанию); 1 – полный режим. бит S: 0 — режим выборки по времени (по умолчанию); 1 — режим выборки по внешнему входу.
03h	Сетевой адрес	1...127 (по умолчанию — 1)
04h	Скорость передачи данных через последовательный порт	1...192, (по умолчанию — 4) задает скорость передачи данных в дискретах по 2400бод, например значение 4 задает скорость 4*2400=9600бод.
05h	<b>Зарезервировано</b>	
06h	Количество усредняемых значений	1...128, (по умолчанию — 1)
07h	<b>Зарезервировано</b>	
08h	Младший байт периода выборки	1) 10...65535, (по умолчанию — 500)
09h	Старший байт периода выборки	задает временной интервал в дискретах по 0.01мс, через который датчик автоматически передает результаты по запросу потока данных (приоритет выборки = 0); 2) 1...65535, (по умолчанию — 500) коэффициент деления для входа синхронизации (приоритет выборки = 1)
0Ah	Младший байт максимального времени накопления	2...65535 (по умолчанию – 3200), задает предельное время накопления CMOS-линейки в дискретах по 1 мкс
0Bh	Старший байт максимального времени накопления	
0Ch	Младший байт начала диапазона аналогового выхода	0...4000h (по умолчанию – 0) определяет точку внутри диапазона датчика, в которой аналоговый выход принимает минимальное значение (4 мА или 0В).
0Dh	Старший байт начала диапазона аналогового выхода	
0Eh	Младший байт конца диапазона аналогового выхода	0...4000h (по умолчанию – 0) определяет точку внутри диапазона датчика, в которой аналоговый выход принимает максимальное значение (4 мА или 0В).
0Fh	Старший байт конца диапазона аналогового выхода	
10h	Время задержки результата	0...255, определяет задержку в инкрементах по 5 мс.
11...16h	<b>Зарезервировано</b>	
17h	Младший байт точки нуля	0...4000h, (по умолчанию — 0) задает начало отсчета в абсолютной системе координат.
18h	Старший байт точки нуля	
19...1Ch	<b>Зарезервировано</b>	

20h	Скорость передачи данных по CAN интерфейсу	10...200, (по умолчанию — 25) задает скорость передачи данных в дискретах по 5 000 бод, например значение 50 задает скорость 50*5 000= 250 000 бод.
22h	Младший байт стандартного идентификатора	0...7FFh, (по умолчанию — 7FFh) задает стандартный идентификатор CAN
23h	Старший байт стандартного идентификатора	
24h	0-й байт расширенного идентификатора	
25h	1-й байт расширенного идентификатора	0...1FFFFFFFh, (по умолчанию — 1FFFFFFFh) задает расширенный идентификатор CAN
26h	2 -й байт расширенного идентификатора	
27h	3-й байт расширенного идентификатора	
28h	Идентификатор CAN интерфейса	1 — расширенный идентификатор CAN; 0 — стандартный идентификатор CAN.
29h	Включение CAN интерфейса	1 — CAN интерфейс включен; 0 — CAN интерфейс отключен.
6Ch	0-й байт IP-адреса назначения	По умолчанию — FFFFFFFFh = 255.255.255.255
6Dh	1-й байт IP-адреса назначения	
6Eh	2 -й байт IP-адреса назначения	
6Fh	3-й байт IP-адреса назначения	
70h	0-й байт IP-адреса шлюза	По умолчанию — C0A80001h = 192.168.0.1
71h	1-й байт IP-адреса шлюза	
72h	2 -й байт IP-адреса шлюза	
73h	3-й байт IP-адреса шлюза	
74h	0-й байт маски подсети	По умолчанию — FFFFFFF00h = 255.255.255.0
75h	1-й байт маски подсети	
76h	2 -й байт маски подсети	
77h	3-й байт маски подсети	
78h	0-й байт IP-адреса источника	По умолчанию — C0A80003h = 192.168.0.3
79h	1-й байт IP-адреса источника	
7Ah	2 -й байт IP-адреса источника	
7Bh	3-й байт IP-адреса источника	
88h	Включение ETHERNET интерфейса	0 — ETHERNET интерфейс отключен; 1 — ETHERNET интерфейс включен в режиме UDP протокола.

### 15.3. Примечания

- Все значения представлены в двоичном виде.
- Базовое расстояние и диапазон задаются в миллиметрах.
- Значение передаваемого датчиком результата (D) нормировано таким образом, чтобы полному диапазону датчика (S в мм) соответствовала величина 4000h (16384), поэтому результат в миллиметрах получают по следующей формуле:

$$X=D*S/4000h \text{ (мм)} \quad (1).$$

- По специальному запросу (05h) текущий результат, может быть, защелкнут в выходном буфере, где он будет оставаться в неизменном виде до прихода запроса передачи данных. Этот запрос может быть передан всем датчикам в сети одновременно в широкоэмитальном режиме для синхронизации момента съема данных со всех датчиков.

- При работе с параметрами следует иметь в виду, что при выключенном питании параметры хранятся в энергонезависимой FLASH-памяти датчика. При включении питания они считываются в оперативную память контроллера датчика. Команда записи новых параметров меняет только их текущие значения в оперативной памяти. Для того чтобы эти изменения сохранились при следующем включении питания, необходимо выполнить специальную команду сохранения текущих значений параметров во FLASH-памяти.
- Параметры, которые имеют размерность более одного байта, должны сохраняться, начиная со старшего байта и заканчивая младшим.
- **ВНИМАНИЕ!** Запрещено выполнять конфигурирование датчиков, включенных в сеть.

## 15.4. Примеры сеансов связи

1) Запрос "идентификация устройства".

Условия: адрес устройства — 1, код запроса — 01h, тип устройства — 61, версия ПО — 88 (58h), серийный номер — 0402 (0192h), базовое расстояние — 80мм (0050h), диапазон — 50мм (0032h), номер пакета — 1.

Формат запроса:

Байт 0				Байт 1				[ Байты 2...N ]			
INC0(7:0)				INC1(7:0)				MSG			
0				1	0	0	0	COD(3:0)			

Запрос "мастера"

Байт 0								Байт 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							

Формат двух посылок данных "ответа" для передачи байта DAT(7:0):

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

Ответ "помощника":

Тип устройства:

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

Версия ПО

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

Серийный номер

DAT(7:0)														
Байт 0							Байт 1							
1	0	0	1	0	0	1	1	0	0	1	1	0	0	1
92h							99h							
DAT(7:0)														
Байт 2							Байт 3							

1	0	0	1	0	0	0	1	1	0	0	1	0	0	0	0
91h								90h							

**Базовое расстояние**

DAT(7:0)															
Байт 0								Байт 1							
1	0	0	1	0	0	0	0	1	0	0	1	0	1	0	1
90h								95h							
DAT(7:0)															
Байт 2								Байт 3							
1	0	0	1	0	0	0	0	1	0	0	1	0	0	0	0
90h								90h							

**Диапазон**

DAT(7:0)															
Байт 0								Байт 1							
1	0	0	1	0	0	1	0	1	0	0	1	0	0	1	1
92h								93h							
DAT(7:0)															
Байт 2								Байт 3							
1	0	0	1	0	0	0	0	1	0	0	1	0	0	0	0
90h								90h							

**Примечание:** так как номер пакета = 1, CNT=1

2) Запрос "чтения параметра".

Условия: адрес устройства — 1, код запроса — 02h, код параметра — 05h, значение параметра — 04h, номер пакета — 2.

Запрос ("мастер") — 01h;82h;

Сообщение ("мастер") — 85h, 80h;

Ответ ("помощник") — A4h, A0h

3) Запрос "запрос результата".

Условия: адрес устройства — 1, значение результата — 02A5h, номер пакета — 3.

Запрос ("мастер") — 01h;86h;

Ответ ("помощник") — B5h, BAh, B2h, B0h

Измеренное смещение (мм) (например, для датчика с диапазоном 50 мм):

$$X=677(02A5h)*50/16384 = 2.066 \text{ мм}$$

4) Запрос: "запись режима выборки "синхронизация по внешнему входу".

Условия: адрес устройства — 1, код запроса — 03h, код параметра — 02h, значение параметра — 01h.

Запрос ("мастер") — 01h, 83h;

Сообщение ("мастер") — 82h, 80h, 81h, 80h;

5) Запрос: "запись периода выборки"

Условия: период выборки — 1234=3039h, адрес устройства — 1, код запроса — 03h, код параметра — 09h (первый или старший байт), значение параметра — 30h

Запрос ("мастер") — 01h, 83h

Сообщение ("мастер") — 89h, 80h, 80h, 83h

и для младшего байта, код параметра — 08h, значение параметра — 39h

Запрос ("мастер") — 01h, 83h

Сообщение ("мастер") — 88h, 80h, 89h, 83h

## 16. Программа параметризации

### 16.1. Назначение

Программное обеспечение **RF\_Sensor\_60x**

([www.riftek.com/resource/files/RFSensor\\_60x.zip](http://www.riftek.com/resource/files/RFSensor_60x.zip)) предназначено для:

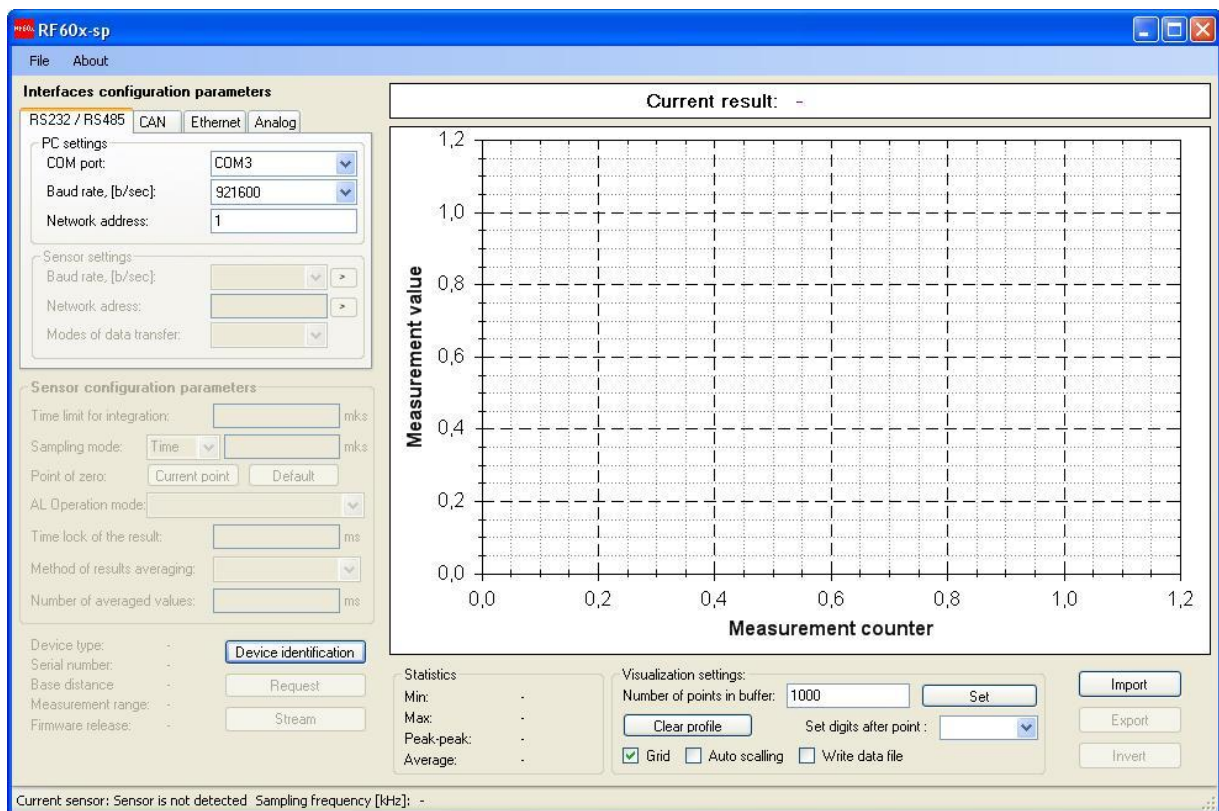
- 1) тестирования и демонстрации работы датчиков серии РФ600;
- 2) настройки параметров датчиков;
- 3) приема и накопления данных с датчика

### 16.2. Установка программы

Запустить файл RF60xsetup.exe и следовать инструкциям мастера установки.

### 16.3. Установка соединения с датчиком (RS232/RS485)

После запуска программы появляется рабочее окно:

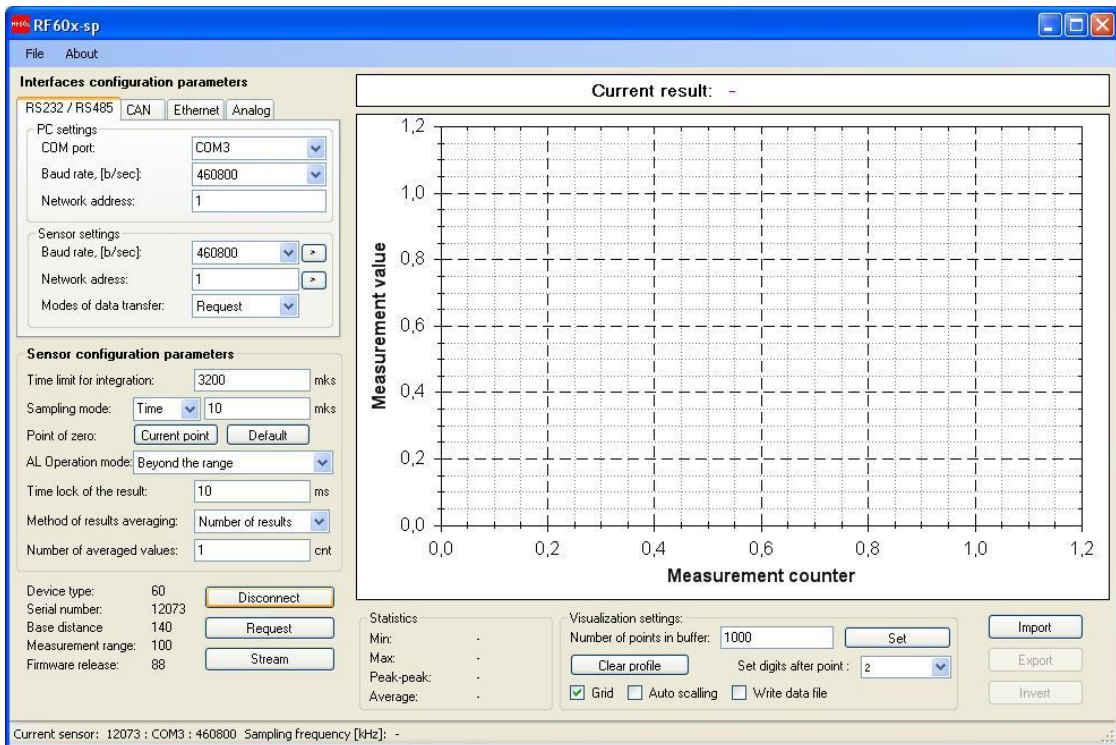


Для установки соединения по RS232/RS485 интерфейсам необходимо во вкладке **RS232/RS485 PC settings** панели **Interface configuration parameters**:

- выбрать COM-порт, к которому подключен датчик (виртуальный порт, в случае подключения датчика через USB-адаптер)
- выбрать скорость передачи (Baud rate), на которой работает датчик
- выбрать, при необходимости, сетевой адрес датчика
- нажать кнопку **Device identification**.

Если установленные параметры соответствуют параметрам интерфейса датчика, программа выполнит идентификацию датчика, считает и отобразит его конфигурационные параметры:

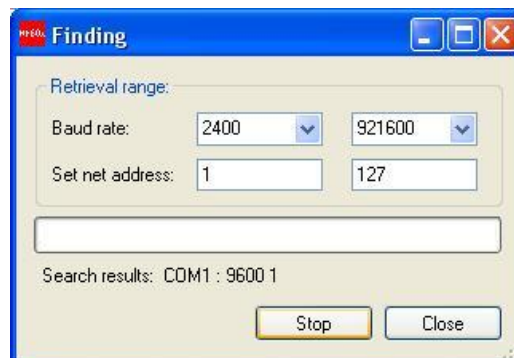




Если связь не установлена, выдается сообщение с предложением провести автоматический поиск датчика:



Для проведения поиска нажать клавишу **Yes**



- в строке Baud rate установить диапазон поиска скорости передачи
- в строке Net address установить диапазон поиска сетевого адреса
- нажать клавишу **Search**

Программа выполнит автоматический поиск датчика путем перебора возможных скоростей, сетевых адресов и COM-портов ПК.

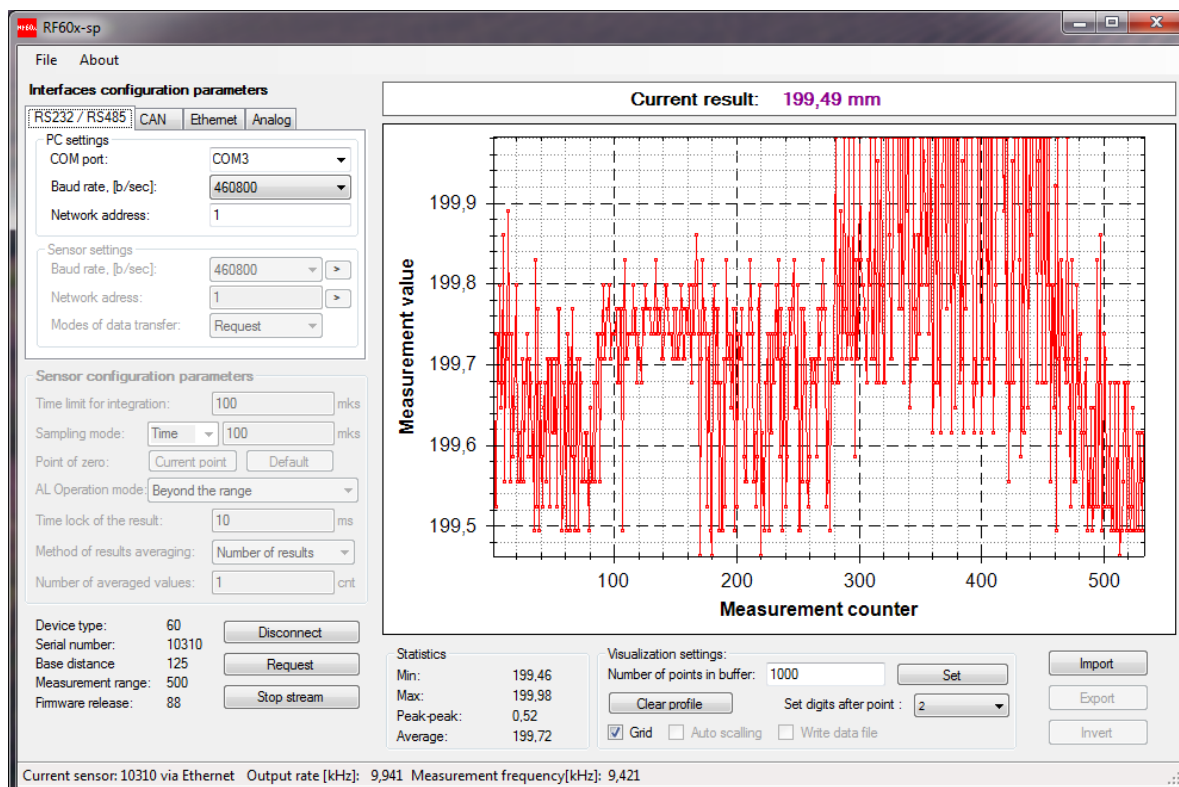
#### 16.4. Проверка работоспособности датчика

После успешной идентификации проверяем работоспособность датчика.

- устанавливаем объект в области рабочего диапазона датчика

- нажатие кнопки **Request** выводит на панель индикации (**Current result**) результат единичного измерения. При этом реализуется тип запроса 06h (см. п. [15.1](#))
- нажатие кнопки **Stream** переводит датчик в режим передачи потока данных. При этом реализуется тип запроса 07h (см. п. [15.1](#)).
- перемещая объект, наблюдаем изменение показаний.
- в статусной строке в нижней части окна отображаются текущие скорость передачи и скорость обновления данных

Нажатие кнопки **Stop stream** останавливает передачу данных.



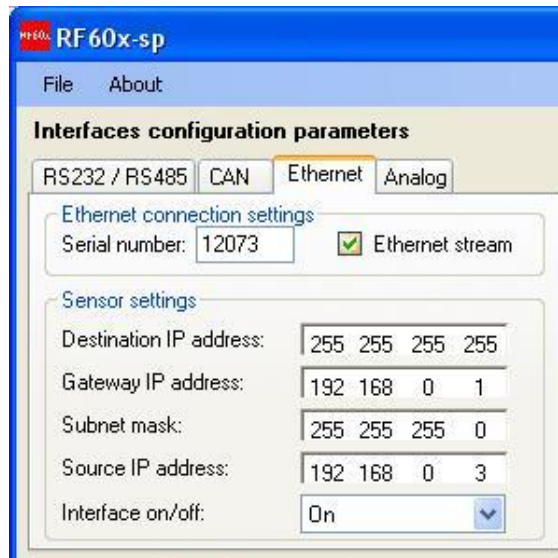
## 16.5. Подключение по Ethernet интерфейсу

Для приема данных по Ethernet-интерфейсу:

- во вкладке Ethernet отметить **Ethernet stream**
- если в сети несколько датчиков, в поле **Serial number** записать серийный номер датчика, от которого необходимо принимать данные
- нажать кнопку **Stream**

**Примечание 1.** Если поле **Serial number** пустое, программа будет работать с датчиком, данные от которого пришли первыми

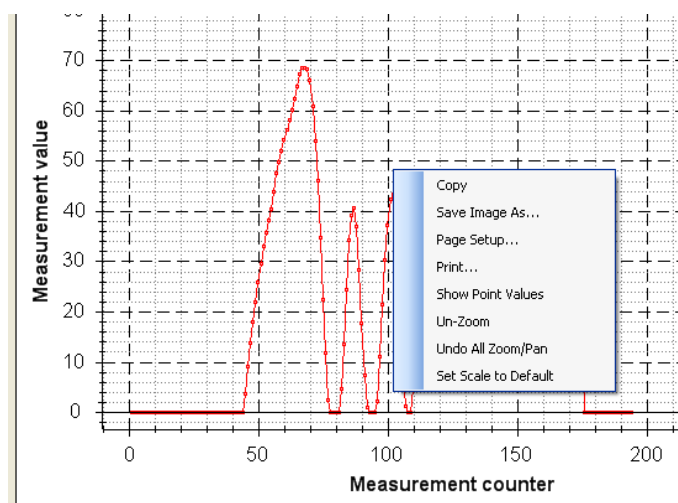
**Примечание 2.** Если **Ethernet stream** не отмечен, а датчик подключен и по RS232/RS485, то данные будут приниматься по этому интерфейсу.



## 16.6. Отображение, накопление и просмотр данных

Результат измерения отображаются в цифровом виде и в виде осциллограммы, и накапливается в памяти ПК.

- количество отображаемых точек по координате X можно задать в окне **Number of points in buffer**;
  - способ масштабирования по координате Y можно задать функцией **Auto scaling**;
  - включение/отключение масштабной сетки производится функцией **Grid**;
  - количество отображаемых после запятой знаков в результате можно установить в окне **Set**;
  - для сохранения поступаемых данных в файл отметить **Write data file**;
- Примечание:** количество точек, отображаемых на графике, зависит от быстродействия ПК и уменьшается пропорционально скорости передачи. После остановки потока, кнопка **Stop Stream**, на графике отображаются все принятые данные
- для работы с изображением щелкнуть правой кнопкой мыши по графику, вызвав соответствующее меню:



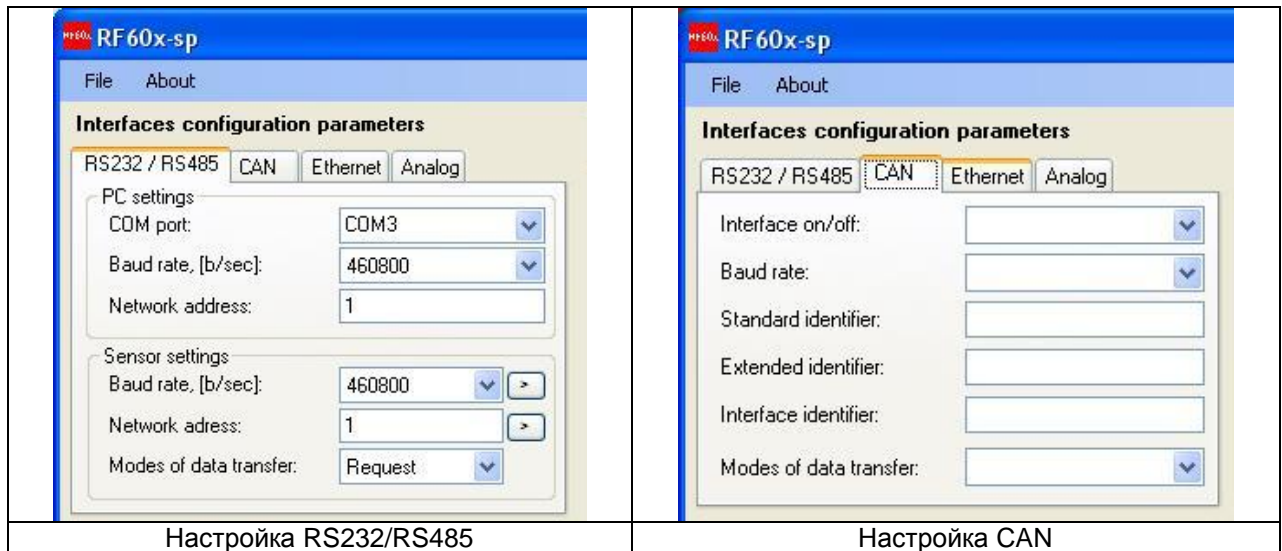
- перемещать изображение можно, нажав колесо мыши
- для зума вращайте колесо мыши

- для сохранения данных в файл нажать кнопку **Export**. Программа предложит сохранить данные в двух возможных форматах: внутреннем и Excel.
- для просмотра ранее сохраненных данных нажать кнопку **Import** и выбрать соответствующий файл.

## 16.7. Настройка и сохранение параметров датчика

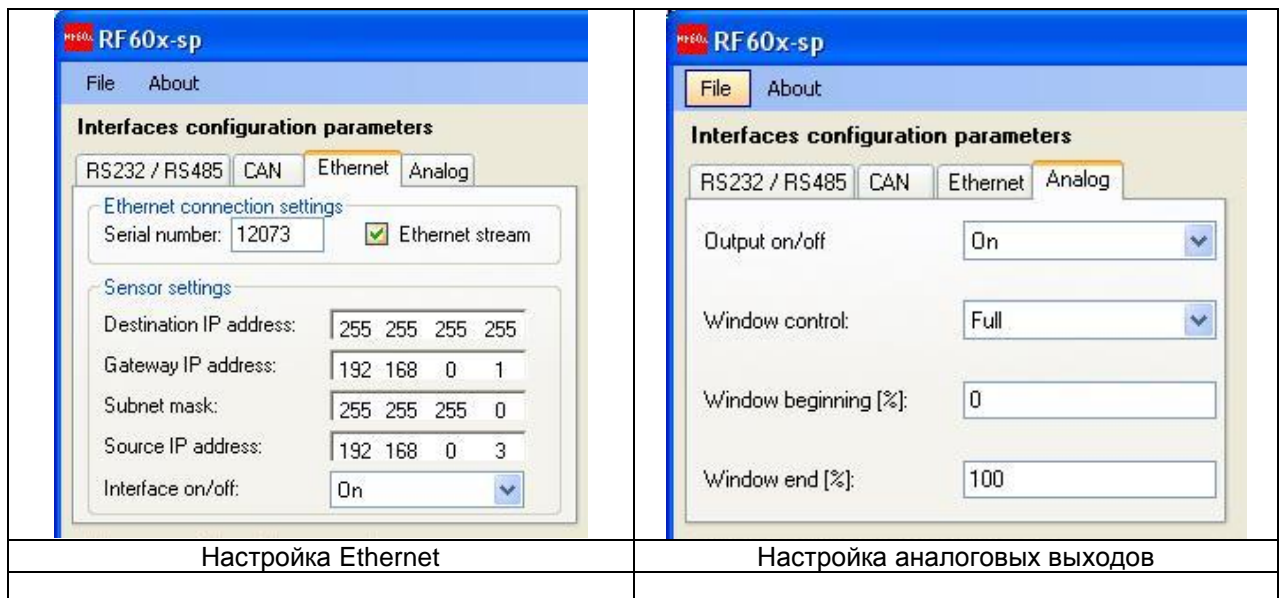
### 16.7.1. Настройка параметров

Параметризация датчика осуществляется только через RS232 или RS485 интерфейсы. Настроить параметры всех интерфейсов можно в соответствующих вкладках панели **Interfaces configuration parameters**:



Настройка RS232/RS485

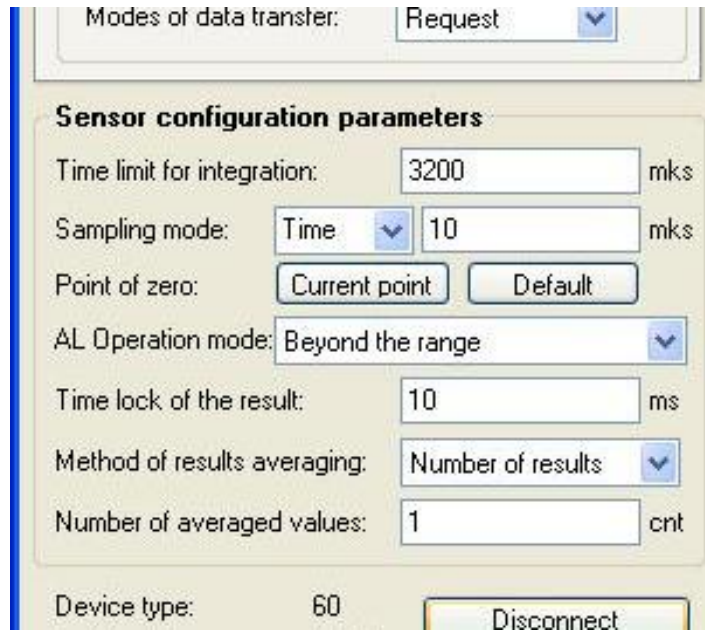
Настройка CAN



Настройка Ethernet


Настройка аналоговых выходов

Настроить все конфигурационные параметры датчика можно в соответствующей панели (**Sensor configuration parameters**):



### 16.7.2. Сохранение параметров

- после установки одного или нескольких требуемых параметров необходимо записать их в память датчика, для чего выполнить **File>Write parameters**.

**Примечание:** для быстрой записи параметров интерфейсов RS232/RS485 предлагается специальная клавиша ;

- провести тестирование работы датчика с новыми параметрами;
- для сохранения новых параметров в энергонезависимой памяти датчика выполнить **File>Write to flash**. Теперь при любом последующем включении датчика он будет работать с установленной Вами конфигурацией.



### 16.7.3. Сохранение и запись группы параметров

Параметры датчика можно сохранить в файл, для чего выбрать **File>Write parameters set**, сохранить файл в предложенном окне.

Для вызова группы параметров из файла выбрать **File>Sensor parameters sets...**, выбрать требуемый файл. **Примечание:** данными функциями удобно пользоваться, если необходимо записать одинаковые параметры в несколько датчиков.

#### 16.7.4. Восстановление параметров по умолчанию

Для восстановления параметров датчика, заданных по умолчанию, выполнить **File>Restore defaults**.

## 17. Библиотека RF60X. Описание функций

С лазерным датчиком поставляется SDK ([www.riftek.com/resource/files/rf60x-sdk.zip](http://www.riftek.com/resource/files/rf60x-sdk.zip)), содержащий:

- динамическую библиотеку RF60x.dll,
- файл для статического подключения DLL к проекту RF60x.lib,
- файл определений RF60x.h.

SDK позволяет пользователю разрабатывать собственные программные продукты, не вдаваясь в подробности протокола обмена данными с датчиком. SDK для 64-разрядных машин можно найти здесь: [www.riftek.com/resource/files/rf60x-sdk64\\_eng.zip](http://www.riftek.com/resource/files/rf60x-sdk64_eng.zip)

### 17.1. Подключение к COM-порту (RF60x\_OpenPort)

Функция **RF60x\_OpenPort** открывает COM-порт с заданным символьным именем, заполняет указатель на дескриптор устройства и возвращает результат операции:

```

BOOL RF60x_OpenPort(
    LPCSTR      lpPort_Name,
    DWORD      dwSpeed,
    HANDLE *    lpHandle
);

```

#### Параметры:

*lpPort\_Name* – название COM-порта (например “COM1:”), полный синтаксис при задании имени COM-порта см. в MSDN, функция CreateFile;  
*dwSpeed* – скорость работы через COM-порт. Параметр идентичен полю BaudRate в структуре DCB, описанной в MSDN;  
*lpHandle* – указатель на дескриптор устройства;

#### Возвращаемое значение:

Если COM-порт открыть и настроить не удалось, функция вернёт FALSE иначе, если COM-порт открыт и настроен успешно – функция вернёт TRUE. Более детальные сведения об возвращаемых ошибках можно получить с помощью API функции GetLastError, описанной в MSDN.

### 17.2. Отключение от COM-порта (RF60x\_ClosePort)

Функция **RF60x\_ClosePort** закрывает COM-порт и возвращает результат операции:

```

BOOL RF60x_ClosePort(
    HANDLE      hHandle
);

```

**Параметры:**

*hHandle* – дескриптор устройства, полученного от функции RF60x\_OpenPort либо CreateFile;

**Возвращаемое значение:**

Если COM-порт закрыть не удалось, функция вернёт FALSE иначе, если COM-порт был успешно закрыт – функция вернёт TRUE.

**17.3. Идентификация устройства (RF60x\_HelloCmd)**

Функция **RF60x\_HelloCmd** производит идентификацию устройства RF60x сети по сетевому адресу и заполняет структуру **RF60xHELLOANSWER**:

```
typedef struct _RF60x_HELLO_ANSWER_ {
    BYTE    bDeviceType;
    BYTE    bcDeviceModificaton;
    WORD    wDeviceSerial;
    WORD    wDeviceMaxDistance;
    WORD    wDeviceRange;
```

Где:

- bDeviceType* – однобайтная величина показывающая тип устройства (для RF60x данная величина равна 60) (тип **BYTE**);
- bDeviceModificaton* – однобайтная величина показывающая модификацию программного обеспечения (тип **BYTE**);
- wDeviceSerial* – двубайтная величина содержащая серийный номер устройства устройства (тип **WORD**);
- wDeviceMaxDistance* – двубайтная величина содержащая значение базового расстояния для датчика РФ60X (тип **WORD**);
- wDeviceRange* – двубайтная величина, содержащая значение диапазона для датчика РФ60X (тип **WORD**).

```
BOOL RF60x_He11oCmd (
    HANDLE          hCOM,
    BYTE            bAddress,
    LPRF60XHELLOANSWER lpHelloAnswer
);
```

**Параметры:**

*hCOM* – дескриптор устройства, полученного от функции RF60x\_OpenPort либо CreateFile;

*bAddress* – адрес устройства;

*lpHelloAnswer* - указатель на структуру RF60xHELLOANSWER.

**Возвращаемое значение:**

Если устройство не ответило на запрос идентификации, функция возвращает FALSE, иначе функция возвращает TRUE и заполняет структуру **RF60xHELLOANSWER**.

**17.4. Чтение параметров (RF60x\_ReadParameter)**

Функция **RF60x\_ReadParameter** читает внутренние параметры датчика РФ600 и возвращает текущее значение по адресу параметра:

```

BOOL RF60x_ReadParameter (
    HANDLE          hCOM,
    BYTE            bAddress,
    WORD            wParameter,
    DWORD *        lpdwValue
);

```

**Параметры:**

*hCOM* – дескриптор устройства, полученного от функции  
 RF60x\_OpenPort, либо CreateFile;  
*bAddress* – адрес устройства;  
*wParameter* – номер параметра, см. табл. 1,

Табл. 1

Параметр	Описание
RF60x_PARAMETER_POWER_STATE	Состояние питания датчика
RF60x_PARAMETER_ANALOG_OUT	Подключение аналогового выхода
RF60x_PARAMETER_SAMPLE_AND_SYNC	Управление выборкой и синхронизацией
RF60x_PARAMETER_NETWORK_ADDRESS	Сетевой адрес
RF60x_PARAMETER_BAUDRATE	Скорость передачи данных через последовательный порт
RF60x_PARAMETER_AVERAGE_COUNT	Количество усредняемых значений
RF60x_PARAMETER_SAMPLING_PERIOD	Период выборки
RF60x_PARAMETER_ACCUMULATION_TIME	Максимальное время накопления
RF60x_PARAMETER_BEGIN_ANALOG_RANGE	Начало диапазона аналогового выхода
RF60x_PARAMETER_END_ANALOG_RANGE	Конец диапазона аналогового выхода
RF60x_PARAMETER_RESULT_DELAY_TIME	Время задержки результата
RF60x_PARAMETER_ZERO_POINT_VALUE	Точка нуля
RF60x_PARAMETER_CAN_SPEED	Скорость передачи данных по CAN интерфейсу
RF60x_PARAMETER_CAN_STANDARD_ID	Стандартный идентификатор CAN
RF60x_PARAMETER_CAN_EXTENDED_ID	Задаёт расширенный идентификатор CAN
RF60x_PARAMETER_CAN_ID	Идентификатор CAN интерфейса

*lpdwValue* – указатель на переменную типа DWORD, в которую будет сохранено текущее значение параметра.

**Возвращаемое значение:**

Если устройства не ответило на запрос чтения параметра, функция возвращает FALSE, иначе функция возвращает TRUE и заполняет переменную *lpdwValue*

## 17.5. Сохранение текущих параметров во FLASH-памяти (RF60x\_FlushToFlash)

Функция RF60x\_FlushToFlash сохраняет все параметры во FLASH-память датчика РФ600:

```

BOOL RF60x_FlushToFlash(
    HANDLE          hCOM,
    BYTE            bAddress
);

```



**Параметры:**

*hCOM* – дескриптор устройства, полученного от функции RF60x\_OpenPort либо CreateFile;  
*bAddress* - адрес устройства.

**Возвращаемое значение:**

Если устройства не ответило на запрос восстановления всех параметров во FLASH-памяти, функция возвращает FALSE, иначе, если от датчика получено подтверждение о восстановлении, функция возвращает TRUE.

## 17.6. Восстановление параметров по умолчанию (RF60x\_RestoreFromFlash)

Функция **RF60x\_RestoreFromFlash** восстанавливает значения всех параметров во FLASH по умолчанию:

```

BOOL RF60x_RestoreFromFlash(
    HANDLE hCOM,
    BYTE bAddress
);

```

**Параметры:**

*hCOM* – дескриптор устройства, полученного от функции RF60x\_OpenPort либо CreateFile;  
*bAddress* - адрес устройства.

**Возвращаемое значение:**

Если устройства не ответило на запрос восстановления всех параметров во FLASH-памяти, функция возвращает FALSE, иначе, если от датчика получено подтверждение о восстановлении, функция возвращает TRUE.

## 17.7. Защелкивание текущего результата (RF60x\_LockResult)

Функция **RF60x\_LockResult** восстанавливает значения всех параметров во FLASH по умолчанию:

```

BOOL RF60x_LockResult(
    HANDLE hCOM,
    BYTE bAddress
);

```

**Параметры:**

*hCOM* – дескриптор устройства, полученного от функции RF60x\_OpenPort либо CreateFile;  
*bAddress* - адрес устройства.

**Возвращаемое значение:**

Если устройства не ответило на запрос защелкивания результата, функция возвращает FALSE, иначе функция возвращает TRUE

## 17.8. Получение результата измерения (RF60x\_Measure)

Функция **RF60x\_Measure** читает из датчика РФ600 текущее измеренное значение. Значение передаваемого датчиком результата (D) нормировано таким образом, чтобы полному диапазону датчика (S в мм) соответствовала величина 4000h (16384), поэтому результат в миллиметрах получают по следующей формуле:  $X=D*S/4000h$  (мм):

```

BOOL RF60x_Measure(
    HANDLE          hCOM,
    BYTE            bAddress,
    USHORT *       lpusValue
);

```

**Параметры:**

*hCOM* – дескриптор устройства, полученного от функции RF60x\_OpenPort либо CreateFile;  
*bAddress* - адрес устройства.  
*lpusValue* - указатель на переменную типа USHORT/WORD, содержащую результат D.

**Возвращаемое значение:**

Если устройства не ответило на запрос результата, функция возвращает FALSE, иначе, если от датчика получено подтверждение о восстановлении, функция возвращает TRUE.

**17.9. Запуск потока измерений (RF60x\_StartStream)**

Функция **RF60x\_StartStream** переводит датчик РФ600 в режим непрерывной передачи результатов измерений:

```

BOOL RF60x_StartStream(
    HANDLE          hCOM,
    BYTE            bAddress
);

```

**Параметры:**

*hCOM* – дескриптор устройства, полученного от функции RF60x\_OpenPort либо CreateFile;  
*bAddress* - адрес устройства.

**Возвращаемое значение:**

Если устройства не удалось перевести в режим непрерывной передачи результатов измерений, функция возвращает FALSE, иначе функция возвращает TRUE.

**17.10. Остановка потока измерений (RF60x\_StopStream)**

Функция **RF60x\_StopStream** переводит датчик из режима непрерывной передачи результатов измерений в режим «запрос-ответ»:

```

BOOL RF60x_StartStream(
    HANDLE          hCOM,
    BYTE            bAddress
);

```

**Параметры:**

*hCOM* – дескриптор устройства, полученного от функции RF60x\_OpenPort либо CreateFile;  
*bAddress* - адрес устройства.

**Возвращаемое значение:**

Если устройства не удалось остановить непрерывную передачу данных, функция возвращает FALSE, иначе функция возвращает TRUE.

### 17.11. Получение результатов измерений из потока (RF60x\_GetStreamMeasure).

Функция **RF60x\_GetStreamMeasure** читает из входного буфера СОМ-порта данные, полученные от датчика РФ600 после успешного выполнения функции **RF60x\_StartStream**. В буфер данные приходят со скоростью, установленной в параметрах датчика РФ600. Т.к. глубина входного буфера ограничена 1024 байтами, то желательно вычитывать данные с периодичностью, равной установленной в параметрах датчика РФ600. Параметр *lpusValue* идентичен параметру *lpusValue* в функции **RF60x\_Measure**.

```

BOOL RF60x_GetStreamMeasure(
    HANDLE          hCOM,
    USHORT *       lpusValue
);

```

#### Параметры:

*hCOM* – дескриптор устройства, полученного от функции **RF60x\_OpenPort** либо **CreateFile**;  
*lpusValue* - указатель на переменную типа **USHORT/WORD**, содержащую результат D.

#### Возвращаемое значение:

Если в буфере данные отсутствуют, то функция возвращает **FALSE**, иначе функция возвращает **TRUE** и заполняет значение *lpusValue*.

### 17.12. Передача пользовательских данных (RF60x\_CustomCmd)

Функция **RF60x\_CustomCmd** используется для передачи и/или приёма данных от датчика РФ600.

```

BOOL RF60x_CustomCmd(
    HANDLE          hCOM,
    char *         pcInData,
    DWORD          dwInSize,
    char *         pcOutData,
    DWORD *        pdwOutSize
);

```

#### Параметры:

*hCOM* – дескриптор устройства, полученного от функции **RF60x\_OpenPort** либо **CreateFile**;  
*pcInData* - указатель на массив данных, который будет передан в датчик РФ600. Если передавать данные не требуется, *pcInData* должен быть **NULL** и *dwInSize* должен быть 0.  
*dwInSize* - размер передаваемых данных. Если данные передавать не требуется, данный параметр должен быть 0.  
*pcOutData* - указатель на массив данных, в который будет сохранены данные полученные от датчика РФ600. Если получать данные не требуется, *pcOutData* должен быть **NULL**.  
*pdwOutSize* - указатель на переменную содержащую размер получаемых данных. Если данные принимать не требуется, данный параметр должен быть **NULL**. После успешного получения данных в переменную, на которую указывает данный параметр, будет записано количество прочитанных байт.

**Возвращаемое значение:**

Если передача либо приём байт не удался, то функция возвращает FALSE, иначе функция возвращает TRUE.

**17.13. Функции для работы с датчиками, подключенными к USB с помощью FTDI**

При работе с USB устройствами на FTDI в данной библиотеке реализована поддержка функций, работающих через D2XX библиотеку FTDI. Работа функций идентична функциям для работы с последовательным портом, основное отличие это присутствие префикса **FTDI\_** в имени функции, например: функция получения результата **RF60x\_Measure** для последовательного порта и **RF60x\_FTDI\_Measure** для устройств с FTDI USB.

**17.14. Функции для работы с датчиками с Ethernet интерфейсом**

Данные функции позволяют получать данные от датчика в поточном режиме (используется протокол UDP) и синхронизироваться по времени либо по внешним сигналам синхронизации.

**17.14.1. Открытие порта для получения данных по Ethernet**

Функция **RF60x\_Ethernet\_OpenPort** открывает сетевой порт, заполняет указатель на дескриптор устройства и возвращает результат операции:

```
BOOL RF60x_Ethernet_OpenPort (
    HANDLE * lpHandle
);
```

**Параметры:**

*lpHandle* - указатель на дескриптор устройства;

**Возвращаемое значение:**

Если сетевой порт не открыт, и/или настроить его не удалось, функция вернёт FALSE иначе, если порт открыт и настроен успешно – функция вернёт TRUE. Более детальные сведения об возвращаемых ошибках можно получить с помощью API функции **WSAGetLastError**, описанной в MSDN.

**17.14.2. Закрытие порта для получения данных по Ethernet**

Функция **RF60x\_Ethernet\_ClosePort** закрывает сетевой порт и возвращает результат операции:

```
BOOL RF60x_Ethernet_ClosePort(
    HANDLE hHandle
);
```

**Параметры:**

*hHandle* – дескриптор устройства, полученного от функции **RF60x\_Ethernet\_OpenPort**;

**Возвращаемое значение:**

Если сетевой порт закрыть не удалось (не был открыт ранее), функция вернёт FALSE, иначе, если порт был успешно закрыт – функция вернёт TRUE.

**17.14.3. Получение 168-ми результатов измерений из потока**

Функция **RF600\_Ethernet\_GetStreamMeasure** получает от датчика 168 измерений и заполняет следующую структуру:

```
typedef struct _RF60x_UDP_PACKET_ {
    RF60xUDPVALUE    rf60xValArray[168];
    WORD              wDeviceSerial;
    WORD              wDeviceBaseDistance;
    WORD              wDeviceMeasureRange;
    BYTE              bPackCount;
    DWORD             dwReserved;
    BYTE              bPacketControlSumm;
} RF60xUDPPACKET, *LPRF60xUDPPACKET;
```

**Где:**

*rf60xValArray* –168 структур (измерений) **RF60xUDPVALUE**, в которых сохранены измерения и статус;

*wDeviceSerial* –двубайтная величина содержащая серийный номер устройства устройства (тип WORD);

*wDeviceBaseDistance* –двубайтная величина содержащая значение базового расстояния для датчика РФ600 (тип WORD);

*wDeviceMeasureRange* –двубайтная величина содержащая значение диапазона для датчика РФ600 (тип WORD);

*bPackCount* –однобайтная величина содержащая значение счётчика пакетов (тип BYTE);

*dwReserved* –четырёхбайтная зарезервированная величина (тип DWORD);

*bPacketControlSumm* –однобайтная величина содержащая значение контрольной суммы (тип BYTE).

**Структура RF60xUDPVALUE:**

```
typedef struct _RF60x_UDP_VALUE_ {
    WORD              wMeasure;
    BYTE              bStatus;
} RF60xUDPVALUE, *LPRF60xUDPVALUE;
```

**Где:**

*wMeasure* –указатель на переменную типа USHORT/WORD, содержащую результат D;

*bStatus* –однобайтная величина, содержащая значение статуса измерения (тип BYTE);

Как и в случае с функцией **RF60x\_Measure**, значение передаваемого датчиком результата (D) нормировано таким образом, чтобы полному диапазону датчика (S в мм) соответствовала величина 4000h (16384), поэтому результат в миллиметрах получают по следующей формуле:  $X=D*S/4000h$  (мм)

**Функция RF603\_Ethernet\_GetStreamMeasure:**

```
BOOL RF603_Ethernet_GetStreamMeasure (
    HANDLE           hHandle,
    LPRF60xUDPPACKET lprf60xUDPPacket
);
```

**Параметры:**

*hHandle* – дескриптор устройства, полученного от функции `RF60x_Ethernet_OpenPort`;  
*lprf60xUDPPacket* - указатель на структуру типа **RF60xUDPPACKET**, содержащую результат D.

**Возвращаемое значение:**

Если в буфере данные отсутствуют, то функция возвращает FALSE, иначе функция возвращает TRUE и заполняет структуру `lprf60xUDPPacket`.

**Примечание:** для стабильной работы функции **RF603\_Ethernet\_GetStreamMeasure** ее необходимо использовать в отдельном потоке с приоритетом, не ниже **THREAD\_PRIORITY\_NORMAL**, иначе происходит переполнение входного буфера, что приводит к непредсказуемым результатам

## 18. Примеры

**ПРИМЕР 1**

```

HANDLE          hRF60x      = INVALID_HANDLE_VALUE;
DWORD           dwValue;
USHORT         usMeasured;
RF60xHELLOANSWER hAns;

// Чистим структуру RF60xHELLOANSWER
memset(&hAns, 0x00, sizeof(RF60xHELLOANSWER));

// Открываем COM-порт
if (!RF60x_OpenPort("COM2:", CBR_9600, &hRF60x))
    return (FALSE);

// Опрашиваем устройство
if (RF60x_helloCmd( hRF60x, 1, &hAns ))
{

    ////////////////////////////////////////
    //
    // После успешного выполнения RF60x_helloCmd
    // в структуре hAns содержится информация о
    // датчике РФ600, ответившем на запрос
    //
    ////////////////////////////////////////

    //читаем параметр: яркость лазера
    RF60x_ReadParameter(
        hRF60x,
        1,
        RF60x_PARAMETER_LASER_BRIGHT,
        &dwValue
    );

    /* В dwValue содержится значения яркости лазера */

    //Получаем значения расстояния из датчика РФ600
    RF60x_Measure( hRF60x, 1, &usMeasured );

    /* В usMeasured содержится измеренный результат */

}

RF60x_ClosePort( hRF60x );

```

**ПРИМЕР 2 (прием потока)**

```

HANDLE                                hRF60x                                =
INVALID_HANDLE_VALUE;
USHORT                                usMeasured;
RF60xHELLOANSWER                       hlans;

memset(&hlans, 0x00, sizeof(RF60xHELLOANSWER));

RF60x_OpenPort("COM2:", CBR_9600, &hRF60x);

if (RF60x_HelloCmd( hRF60x, 1, &hlans ))
{
    printf("Dev modify\t: %d\r\nDev type\t: %d\r\nDev max dist\t:
    %d\r\nDev range\t: %d\r\nDev serial\t: %d\r\n",
           hlans.bDeviceModificaton,
           hlans.bDeviceType,
           hlans.wDeviceMaxDistance,
           hlans.wDeviceRange,
           hlans.wDeviceSerial
    );

    if (!RF60x_WriteParameter( hRF60x, 1,
RF60x_PARAMETER_SAMPLING_PERIOD, 500 ))
        return (-1);

    if (!RF60x_StartStream(hRF60x, 1))
        return (-1);

    RF60x_GetStreamMeasure(hRF60x, &usMeasured);
    printf("Measure \t: %d\r\n", usMeasured);

    RF60x_GetStreamMeasure(hRF60x, &usMeasured);
    printf("Measure \t: %d\r\n", usMeasured);

    RF60x_StopStream(hRF60x, 1);
} else printf("rs232 error!\r\n");

RF60x_ClosePort( hRF60x );

```

**ПРИМЕР 3 (прием с защелкиванием результата)**

```

for (int i=0;i<100;i++)
{
    // Защёлкиваем результат
    RF60x_LockResult(hRF60x, 1);

    /* Перемещаем объект */

    // Получаем результат
    RF60x_Measure( hRF60x, 1, &usMeasured);

    printf("Measure-1 \t: %d\r\n", usMeasured);
}

```

Примеры программ для работы с датчиками в среде LabView можно найти здесь: [www.riftek.com/resource/files/rf60x\\_labview\\_example.zip](http://www.riftek.com/resource/files/rf60x_labview_example.zip)

Примеры программ для работы с датчиками в среде Visual Basic можно найти здесь: [www.riftek.com/resource/files/rf60x-vb.zip](http://www.riftek.com/resource/files/rf60x-vb.zip)

## 19. Приложения

### 19.1. Варианты установки разъема

Габаритные размеры датчика с кабельным разъемом показаны на рис. 4, размеры датчика с коннектором Binder - на рис.5.

Варианты установки углового разъема, вид сбоку, отображены на рисунке 6, угловые разъемы, вид сверху, - на рис.7.

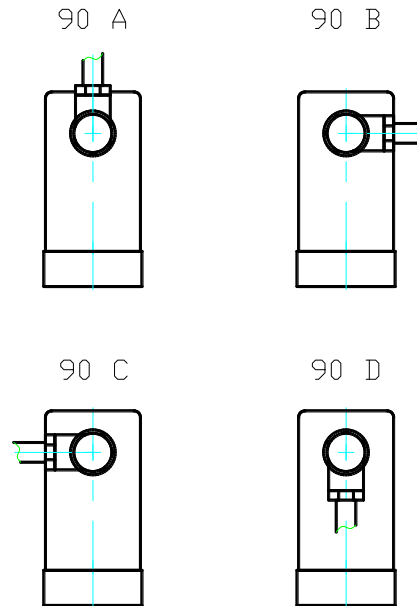


Рисунок 4

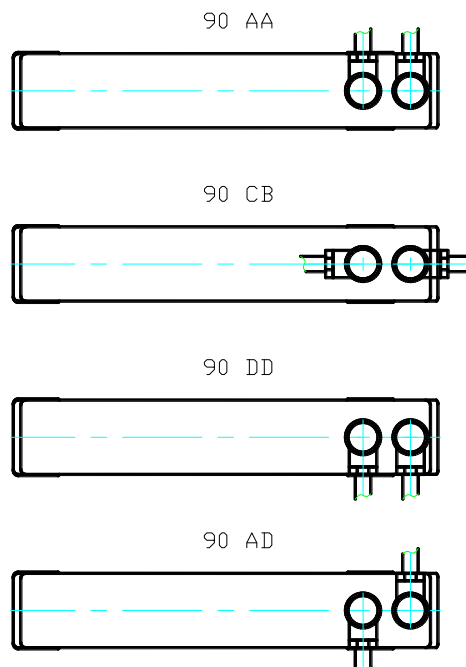


Рисунок 5

## 20. Гарантийные обязательства

Гарантийный срок эксплуатации Триангуляционных лазерных датчиков РФ600 - 24 месяца со дня ввода в эксплуатацию, гарантийный срок хранения - 12 месяцев.



## 21. Датчики производства РИФТЭК



### Лазерные триангуляционные датчики. Серия РФ60х

- измерение размеров и перемещений;
- диапазоны измерения от 2 мм до 2,5 м;
- точность измерения  $\pm 1$  мкм;
- частота обновления данных 180 кГц;
- датчики на базе **СИНЕГО** и **Инфракрасного** лазеров;
- быстродействующие датчики с индексом HS (High Speed);



Серия включает четыре модельных ряда:

**РФ603** – универсальные датчики с рабочим диапазоном от 2 до 1250 мм;

**РФ603HS** – быстродействующие датчики;

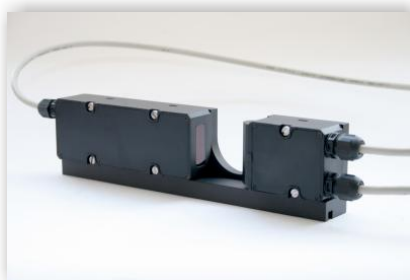
**РФ600** – датчики с увеличенным базовым расстоянием и большим диапазоном измерений;

**РФ605** – недорогие, малогабаритные датчики.



### Лазерные 2D сканеры. Серия РФ620HS (DHS)

- 2D/3D измерения;
- рабочие диапазоны от 5 мм до 1500 мм;
- линейность 0,05% от диапазона;
- быстродействие до 1000 профилей/с;
- сканеры на базе **СИНЕГО** и **Инфракрасного** лазеров;



### Оптические микрометры. Серии РФ65х

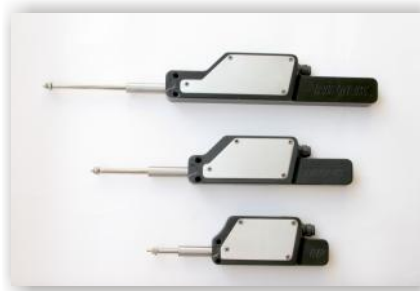
- измерение диаметров, зазоров и положения;
- рабочие диапазоны от 6 мм до 60 мм;
- точность  $\pm 0.5$  мкм;
- частота обновления данных 1000 Гц;

Серия включает два модельных ряда:

**РФ651** – теневые микрометры с диапазоном измерений 25 и 59 мм и точностью  $\pm 5$  мкм;

**РФ656** – высокоточные теневые микрометры с телецентрической оптикой, диапазоном измерений 5 и 25 мм и точностью  $\pm 0,5$  мкм.

### Абсолютные линейные энкодеры. Серия РФ25х



- измерение размеров и перемещений;
- инновационная технология абсолютного отсчета;
- диапазоны измерения от 3 до 55 мм;
- разрешающая способность 0,1 мкм;

*Серия включает две модели:*

**РФ251** – датчики для работы в тяжелых условиях;

**РФ256** – широкодиапазонные датчики.

42

Все подробности об измерительном оборудовании РИФТЭК на сайте [www.riftek.com](http://www.riftek.com).