

**ТРИАНГУЛЯЦИОННЫЕ ЛАЗЕРНЫЕ ДАТЧИКИ. СЕРИЯ РФ603.**
**1. ОБЩИЕ СВЕДЕНИЯ**

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 24 модели датчиков с измерительным диапазоном от 2 до 1000 мм и базовым расстоянием от 10 до 245 мм. Возможны также заказные конфигурации с параметрами, отличными от указанных ниже.

**2. ОСНОВНЫЕ ТЕХНИЧЕСКИЕ ДАННЫЕ И ХАРАКТЕРИСТИКИ**

Модель РФ603-	X/2	X/5	X/10	X/15	X/25	X/30	X/50	X/100	X/250	X/500	X/750	X/1000
Базовое расстояние, X, мм	10	15	15, 25 55	15, 30 60	25, 45 80	35, 55 95	45, 65 105	60, 90 140	80	125	145	245
Диапазон, мм	2	5	10	15	25	30	50	100	250	500	750	1000
Линейность, %	±0.1 от диапазона										±0.2...0.3	
Разрешение, %	0.01 диапазона										0.03	
Макс. частота обновления данных, кГц	2 или 5 или 8											
Тип лазера	1...3 мВт, длина волны 660 нм									5 мВт, 660 нм		
Выходной интерфейс	Цифровой	RS232 (макс. 460,8 Кбит/с) или RS485 (макс. 460,8 Кбит/с) или RS232 и CAN V2.0B (макс. 1Мбит/с)										
	аналоговый	4...20 мА (нагрузка ≤ 500 Ом) или 0...10 В										
Вход синхронизации	2,4 – 5 В (CMOS, TTL)											
Логический выход	NPN: 100 мА max; 40 В max											
Напряжение питания, В	5 (4,5...9) или 12 (9...18) или 24 (18...36)											
Потребляемая мощность, Вт	1,5..2											
Класс защиты	IP67											
Рабочая температура, °С	-10...+60, (-30...+60 для датчиков со встроенным нагревателем), (-30...+120 для датчиков со встроенным нагревателем и защитным корпусом)											
Вес (без кабеля), г	100											

**3. ПРИМЕР ОБОЗНАЧЕНИЯ ПРИ ЗАКАЗЕ**

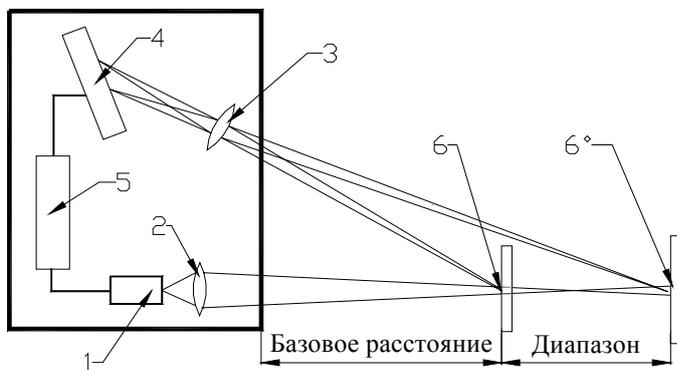
РФ603.F-X/L-SERIAL-OUT-IN-AL-VV-CC-M-N-P

Символ	Наименование
F	Максимальная частота обновления, кГц (2 или 5 или 8)
X	Базовое расстояние (начало диапазона), мм
L	Рабочий диапазон, мм
SERIAL	Тип последовательного интерфейса (RS232 или RS485 или RS232 и CAN)
OUT	Наличие аналогового выхода по току ( I ) или по напряжению ( U ) Примечание: только для датчиков с RS232 или RS485
IN	Наличие входа синхронизации
AL	Данный сигнал имеет двойное назначение. Может использоваться как 1) логический выход ("0" – объект вне диапазона (вне заданного окна диапазона), "1" – в диапазоне (в заданном окне диапазона)); 2) линия взаимной синхронизации двух и более датчиков
VV	Напряжение питания, В
CC	Кабельный ввод - CG, либо коннектор - CC (Binder 702, IP67)
M	Длина кабеля, м
N	Наличие встроенного нагревателя
P	Датчик в защитном корпусе с воздушным охлаждением (приложение 1)

**Пример.** РФ603.5-140/100-232-IN-I-12-CG-3 – максимальная частота 5 кГц, базовое расстояние – 140 мм, диапазон - 100мм, последовательный порт RS232, есть вход синхронизации, есть токовый выход 4...20мА, напряжение питания - 12В (9..18В), кабельный ввод, длина кабеля 3 м.

#### 4. УСТРОЙСТВО И ПРИНЦИП РАБОТЫ

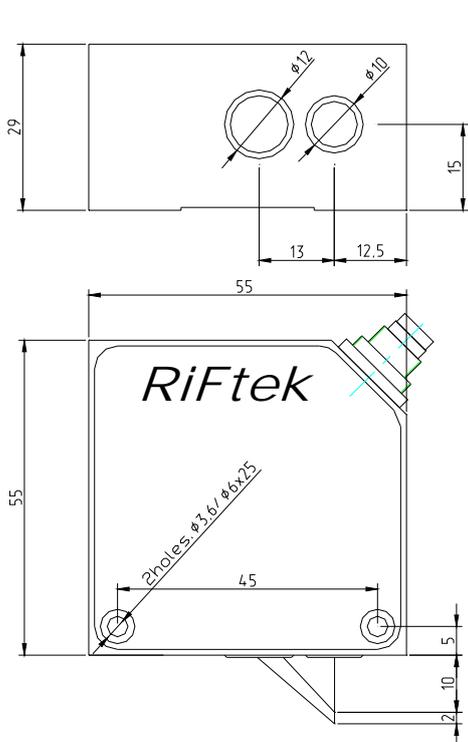
В основу работы датчика положен принцип оптической триангуляции, рис.1. Излучение полупроводникового лазера 1 фокусируется объективом 2 на объекте 6. Рассеянное на объекте излучение объективом 3 собирается на CMOS-линейке 4. Перемещение объекта 6 – 6' вызывает соответствующее перемещение изображения. Процессор сигналов 5 рассчитывает расстояние до объекта по положению изображения светового пятна на линейке 4.



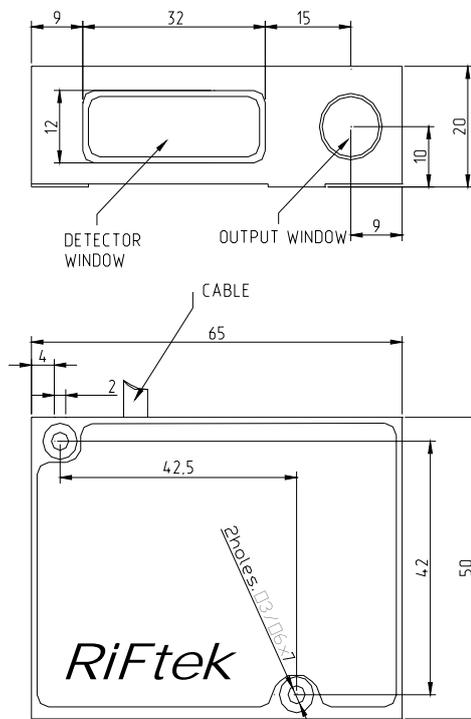
**Рисунок 1.**

#### 5. ГАБАРИТЫ И УСТАНОВКА.

5.1. Габаритные и установочные размеры датчика -10/2 показаны на рис.2, остальных датчиков – на рис.3. 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.



**Рисунок 2.**



**Рисунок 3.**

**ТРИАНГУЛЯЦИОННЫЕ ЛАЗЕРНЫЕ ДАТЧИКИ. СЕРИЯ РФ603.**

**5.2.** Датчик устанавливается таким образом, чтобы контролируемый объект располагался в зоне рабочего диапазона датчика. Кроме того, в области прохождения 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.

**6. ПОДКЛЮЧЕНИЕ**

Модель	Обозначение	Контакты DB9F	Цвет провода
232-U/I-IN-AL	Power U+	-	Красный
	Power U-	-	Коричневый
	TXD	2	Зеленый
	RXD	3	Желтый
	U/I	-	Синий
	IN	-	Белый/Фиолетовый
	AL	-	Розовый/Оранжевый
Gnd (Общий для сигналов)	5	Серый/Черный	
232-CAN-IN/AL	Power U+	-	Красный
	Power U-	-	Коричневый
	TXD	2	Зеленый
	RXD	3	Желтый
	CAN <sub>H</sub>	-	Розовый/Оранжевый
	CAN <sub>L</sub>	-	Белый/Фиолетовый
	IN/AL	-	Синий
Gnd (Общий для сигналов)	5	Серый/Черный	
485-U/I-IN-AL	Power U+	-	Красный
	Power U-	-	Коричневый
	DATA+	-	Зеленый
	DATA-	-	Желтый
	U/I	-	Синий
	IN	-	Белый/Фиолетовый
	AL	-	Розовый/Оранжевый
Gnd (Общий для сигналов)	-	Серый/Черный	

**7. РЕЖИМЫ РАБОТЫ И КОНФИГУРАЦИОННЫЕ ПАРАМЕТРЫ.**

**7.1.** Данные от датчика (результаты измерений) могут быть получены по последовательному интерфейсу и/или на аналоговом выходе. По последовательному интерфейсу результаты можно получить как по разовым запросам, так и автоматически потоком данных (раздел 7 “Описание последовательного интерфейса”). При использовании интерфейса стандарта RS485 или CAN к одному устройству сбора информации можно подключить несколько датчиков по схеме “общая шина” (сетевой вариант использования).

**7.2.** Характер работы датчика определяют его конфигурационные параметры (режимы работы), изменение которых производится передачей команд через последовательный порт. Основные параметры:

**Период выборки** — определяет интервал времени (внутренняя синхронизация) или коэффициент деления для входа внешней синхронизации, через который датчик должен автоматически обновлять результат измерения. Значение интервала времени задается в дискретах по 0.01мс. Если для приема результата используется последовательный интерфейс, то при задании малых интервалов 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.

**Режим выборки** — задает режим выборки

- по времени или
- по внешнему входу

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

При установке *выборки по входу* датчик передает результат при переключении входа внешней синхронизации и с учетом установленного *коэффициента деления*.

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

**Режим работы аналогового выхода.** При работе с функцией "окно в рабочем диапазоне" 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).

**Режим работы логического выхода и режим взаимной синхронизации.**

*Логический выход* может использоваться для

- индикации выхода за диапазон ("0" – объект вне диапазона (вне заданного окна диапазона), "1" – в диапазоне (в заданном окне диапазона))
- взаимной синхронизации двух и более датчиков.

Установка режима взаимной синхронизации 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.

**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.

**Количество усредняемых значений** — определяет количество исходных результатов, по которым берется среднее для формирования выходного значения. Исходные результаты помещаются в кольцевой буфер заданного размера, и новое значение среднего вычисляется каждый раз после поступления нового результата, в этом смысле выходная величина является скользящим средним.

Обновление результата на аналоговом выходе также управляется вышеописанными параметрами.

**Уровень выходной мощности лазера.** Изменение этого параметра позволяет настроить датчик на работу с минимальным предельным временем накопления (максимальное быстрое действие) для конкретных поверхностей.

**Предельное время накопления.** Интенсивность отраженного излучения зависит от качества поверхности контролируемых объектов, поэтому время накопления излучения, падающего на CMOS-линейку, автоматически подстраивается с целью достижения максимальной точности измерения. Данный параметр задает величину предельно допустимого времени на

копления. Если интенсивность принимаемого датчиком излучения настолько мала, что за время накопления, равное предельному, не получен приемлемый результат, датчик передает нулевое значение. Увеличение данного параметра расширяет возможности контроля слабоотражающих (диффузная составляющая) поверхностей, однако уменьшает частоту обновления данных и увеличивает влияние внешней засветки (фона) на точность измерения. Заводская установка предельного времени накопления – 1300 мкс.

**Точка нуля** — задает начало отсчета в абсолютной системе координат в любой точке в пределах рабочего диапазона. При изготовлении датчика базовое расстояние задается с некоторой неопределенностью и при необходимости возможно более точное задание начала координат.

**Зарезервированные параметры используются для настройки датчика. Изменение этих параметров может привести к нарушению калибровки датчика. Корректное изменение параметров производится с помощью программы параметризации, поставляемой с датчиком.**

## **8. ПРОСЛЕДОВАТЕЛЬНЫЙ ИНТЕРФЕЙС (RS232 или RS485)**

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

**8.2.** Порт RS485 в соответствии с принятым протоколом и аппаратными возможностями позволяет подключить до 127 датчиков к одному устройству сбора информации по схеме “общая шина”.

**8.3.** 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 РФ603), 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).

**8.4.** Формат последовательной посылки данных:

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

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

**8.5.** Протокол обмена построен на сеансах связи, которые иницируются только “мастером”. Существуют сеансы связи двух видов:

- 1) “запрос”, [“сообщение”] — [“ответ”], в квадратных скобках указаны необязательные элементы
- 2) “запрос” — “поток данных” — [“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”: INC0(7:0), INC1(7:0) = 0, ADR(6:0), 1, 0, 0, 0, COD(3:0), [MSG].

“Сообщение” и “ответ” — это пакеты данных, которые могут передаваться в сеансе связи “мастером” и “помощником”, соответственно. “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, поэтому при выдаче “мастером” любого нового запроса

**ТРИАНГУЛЯЦИОННЫЕ ЛАЗЕРНЫЕ ДАТЧИКИ. СЕРИЯ РФ603.**

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

**8.6. Передача сообщения**

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):

$$Dt0(7:0);Dt1(7:0) = 1,0,0,0,DAT(3:0);1,0,0,0,DAT(7:4).$$

**8.7. Передача ответа (для кодов запроса 01h...04h).**

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 еще три 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):

$$Dt0(7:0);Dt1(7:0) = 1,CNT(2:0),DAT(3:0);1,CNT(2:0),DAT(7:4).$$

**8.8. Передача результата измерения. Ответ формируется аналогично п.8.7.**
**8.9. Типы запросов.**

Код запроса	Описание	Сообщение (размер в байтах)	Ответ (размер в байтах)
01h	Идентификация устройства	—	-тип устройства (1) -модификация (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	—	—

**8.10. 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 выборкой и синхронизацией	x,x,x,C,x,M,R,S – контрольный байт, задающий режим работы CAN интерфейса, бит C, логического выхода, бит M, аналогового выхода, бит R, и режим выборки, бит S;

**ТРИАНГУЛЯЦИОННЫЕ ЛАЗЕРНЫЕ ДАТЧИКИ. СЕРИЯ РФ603.**

		биты x – не используются; бит C: 0 – режим CAN интерфейса по запросу (по умолчанию); 1 – режим CAN интерфейса с синхронизацией по времени или по внешнему входу. бит M: 0 – режим индикации выхода за диапазон (по умолчанию); 1 – режим взаимной синхронизации. 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	Уровень яркости лазера	0...31
06h	Количество усредняемых значений	1...128, (по умолчанию — 1)
07h		<b>Зарезервировано</b>
08h	Младший байт периода выборки	1) 10...65535, (по умолчанию — 500) 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)
09h	Старший байт периода выборки	
0Ah	Младший байт максимального времени накопления	2...65535 (по умолчанию – 200), 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 — 200) 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	0...1FFFFFFFh, (по умолчанию — 1FFFFFFFh) задает расширенный идентификатор CAN
25h	1-й байт расширенного идентификатора	
26h	2 -й байт расширенного идентификатора	
27h	3-й байт расширенного идентификатора	
28h	Идентификатор CAN интерфейса	1 — расширенный идентификатор CAN; 0 — стандартный идентификатор CAN.

**8.11. Примечание:**

Page 7/21, Rev F, 10.01.2008г.

**ТРИАНГУЛЯЦИОННЫЕ ЛАЗЕРНЫЕ ДАТЧИКИ. СЕРИЯ РФ603.**

- 1) Все значения представлены в двоичном виде.
- 2) Базовое расстояние и диапазон задаются в миллиметрах.
- 3) 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:

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

4) 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 широкополосном режиме для синхронизации момента съема данных со всех датчиков.

5) 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.

**6) Параметры, которые имеют размерность более одного байта, должны сохраняться, начиная со старшего байта и заканчивая младшим.**

**8.12. Примеры сеансов связи:**

- 1) Условие: запрос идентификатора устройства. Адрес устройства — 1, код запроса — 01h, тип устройства — 61, модификация — 00h, серийный номер — 0402 (0192h), базовое расстояние — 80мм (0050h), диапазон — 50мм (0032h), номер пакета — 1.  
Формат запроса:

**INC0(7:0), INC1(7:0) = 0,ADR(6:0),1,0,0,0,COD(3:0),[MSG]**

Запрос ("мастер") — 01h;81h; (INC0(7:0)=0,ADR=0000001,INC1(7:0)=1,0,0,0,COD=0001).

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

**Dt0(7:0);Dt1(7:0) = 1,CNT(2:0),DAT(3:0);1,CNT(2:0),DAT(7:4)**

Ответ ("помощник") — 91h, 96h (тип устройства), 90h, 90h (модификация), 92h, 99h, 91h, 90h (серийный номер), 90h, 95h, 90h, 90h (базовое расстояние), 92h, 93h, 90h, 90h (диапазон).

(замечание: так как номер пакета = 1, CNT=1)

- 2) Условие: запрос чтения параметра. 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) Условие: запрос результата. 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) Условие: запись режима выборки (синхронизация по внешнему входу). Адрес устройства — 1, код запроса — 03h, код параметра — 02h, значение параметра — 01h.

Запрос ("мастер") — 01h, 83h;

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

- 5) Условие: запись периода выборки, например, 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

## ТРИАНГУЛЯЦИОННЫЕ ЛАЗЕРНЫЕ ДАТЧИКИ. СЕРИЯ РФ603.

и для младшего байта, код параметра – 08h, значение параметра – 39h  
Запрос ("мастер") – 01h, 83h  
Сообщение ("мастер") – 88h, 80h, 89h, 83h

### 9. ОПИСАНИЕ CAN ИНТЕРФЕЙСА.

**9.1.** Датчик, оборудованный портом CAN 2.0В, 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.

Датчик может работать в двух режимах:

- 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 синхронизации. При работе в режиме синхронизации каждый из датчиков, автоматически передает кадр данных (Data Frame) со своим идентификатором в соответствии с заданным временным интервалом (периодом выборки), либо при переключении входа внешней синхронизации и с учетом установленного коэффициента деления.

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

**9.2.** 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.8.11).

### 10. ПРОГРАММА ПАРАМЕТРИЗАЦИИ.

**10.1.** Программное обеспечение "RF60X-SP" предназначено для:

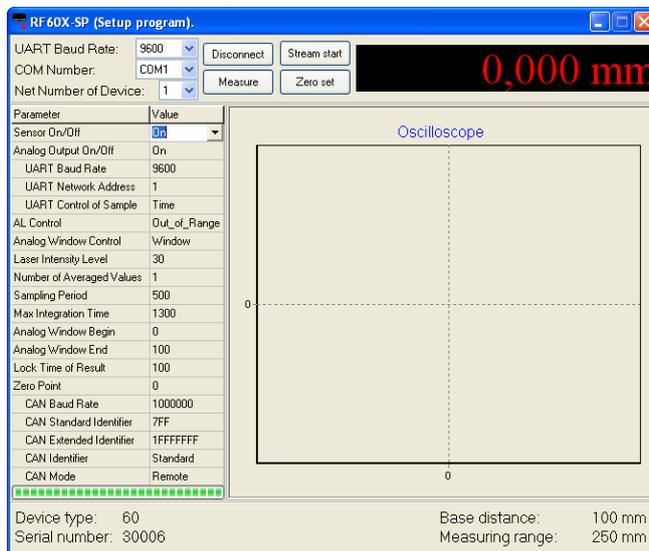
- 1) тестирования и демонстрации работы датчиков серии РФ603;
- 2) настройки параметров датчиков;
- 3) приема и накопления данных;

**10.2.** После запуска программы появляется рабочее окно:



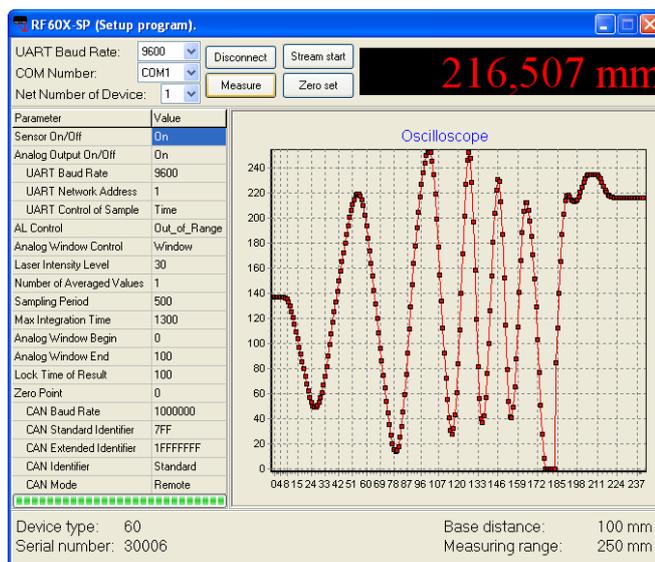
- 1) В строке "UART Baud Rate" выбираем скорость, на которой работает датчик (заводская установка для датчика – 9600 бит/с),
- 2) В строке "COM Number" выбираем номер порта RS232 персонального компьютера, к которому датчик подключён.
- 3) Строка "Net Number of Device" определяет сетевой адрес датчика (заводская установка для всех датчиков – "1")
- 4) При нажатии на кнопку "Connect" RF60X-SP попытается наладить связь с датчиком с параметрами, заданными в выше перечисленных строках. Если связь не установлена, выдается сообщение об отсутствии связи.

- 5) При успешном соединении:
- 1) в строке "Device Type" индицируется тип датчика;
  - 2) в строке "Serial Number" - серийный номер датчика;
  - 3) в строке "Base Distance" - базовое расстояние датчика;
  - 4) в строке "Measuring Range" индицируется рабочий диапазон датчика, и окно трансформируется к следующему виду:



### 10.3. После успешной установки связи проверяем работоспособность датчика:

- 1) Устанавливаем объект в области рабочего диапазона датчика.
- 2) Нажатие кнопки "Measure" выводит на панель индикации и панель "Oscilloscope" результаты измерения положения объекта. (Ось X – время (режим выборки по времени), либо номер отсчета (режим выборки по внешнему входу), Y – координата). При этом реализуется тип запроса 06h (см. п.8.9).
- 3) Нажатие кнопки "Stream start" включает измерения в режиме выборки по времени в соответствии с установленным параметром Sampling Period. При этом реализуется тип запроса 07h (см. п. 8.9).
- 4) Перемещая объект в области рабочего диапазона, наблюдаем изменение показаний.



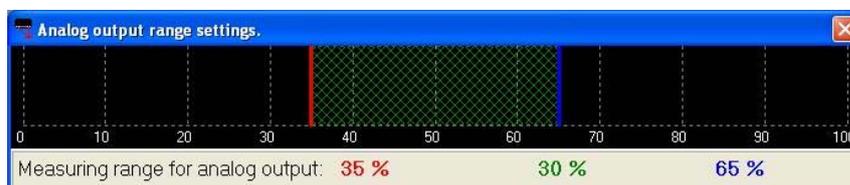
- 5) Нажатие кнопки "Stop"/"Stream stop" останавливает передачу данных.
- 6) Данные, поступающие от датчика, накапливаются и сохраняются в кольцевом буфере размером 10000 измерений.

- 7)левой клавишей мыши можно изменять масштаб графика, правой – перемещать график в области просмотра. Нажатие правой клавиши вызывает меню "Сохранение данных в файл".

#### 10.4. Настройка параметров датчика

Открывшаяся часть приложения "RF60X-SP" (Таблица "Parameter – Value") позволяет редактировать и заносить как в RAM, так и во FLASH датчика соответствующие параметры.

- для включения/выключения датчика дважды щелкните левой клавишей мыши в поле "Value" параметра "Sensor On/Off";
- для включения/отключения аналогового выхода дважды щелкните левой клавишей мыши в поле "Value" параметра "Analog Output On/Off";
- для установки режима выборки "UART Control of Sample" нажмите клавишу в поле "Value" и выберете режим;
- для установки скорости обмена щелкните левой клавишей мыши в поле "Value" строки "UART Baud Rate", вызвав список допустимых скоростей;
- в строке "UART Network Address" устанавливаем сетевой адрес датчика;
- в строке "AL Control" устанавливаем режим работы выхода AL;
- в строке "Laser Intensity Level" можно установить уровень выходной мощности лазера в милливаттах;
- в строке "Averaged Values Counter" выбираем число измерений, подвергающихся усреднению непосредственно в датчике. Заводская установка – "0";
- в строке "Sampling Period" устанавливаем период выборки в дискретах по 0.01 мс;
- в поле "Max Integration Time" можно установить предельное время накопления линейки (в долях по 1 мкс);
- в строках "Analog Range Begin" и "Analog Range End" можно установить границы окна аналогового выхода, в дискретах по 1% рабочего диапазона. Дважды щелкнув в поле "Value", вызываем панель настройки:



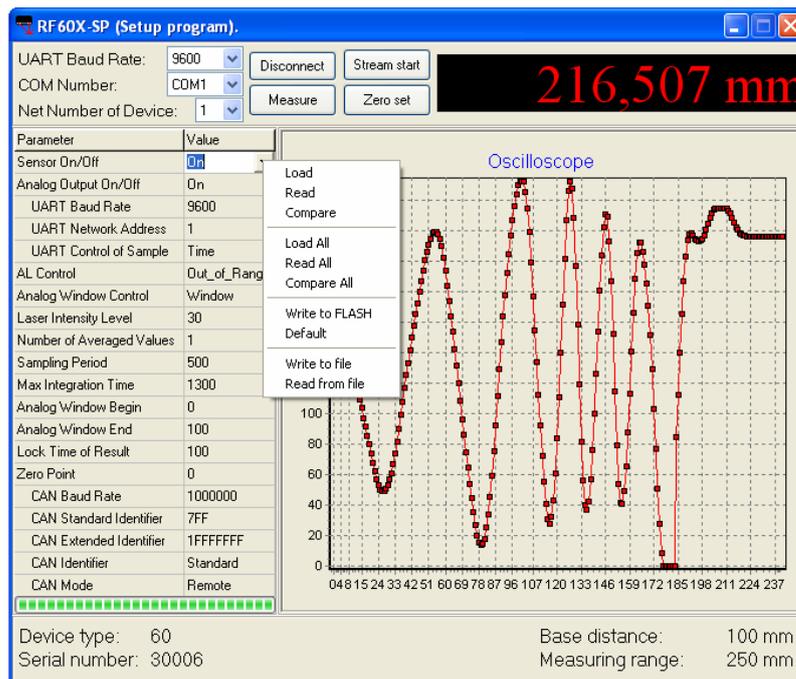
При нажатии левой кнопки мыши активируется красный курсор, указывающий начало масштабного диапазона, при нажатии правой кнопки мыши активируется синий курсор, указывающий конец масштабного диапазона. Для установки границ рабочего окна датчика нажать соответствующую кнопку и, не отпуская ее, произвести перемещение курсора в области измерительного диапазона датчика. В нижней строке индицируются значения границ выбранного окна в % от диапазона.

- в строке "Lock Time of Result" задаем временной интервал в дискретах по 5 мс, через который датчик формирует результат при вхождении объекта в рабочий диапазон и удержания последнего результата при уходе объекта из рабочего диапазона (потере объекта);
- в строке "Zero Point" можно установить начало координат в долях по 0,1% от диапазона, либо, нажав кнопку "Measure", разместите объект в требуемой точке рабочего диапазона и нажмите клавишу "Zero set". Теперь начало координат будет соответствовать выбранной вами точке;
- для установки скорости обмена по CAN интерфейсу щелкните левой клавишей мыши в поле "Value" строки "CAN Baud Rate", вызвав список допустимых скоростей.

При работе с CAN-интерфейсом

## ТРИАНГУЛЯЦИОННЫЕ ЛАЗЕРНЫЕ ДАТЧИКИ. СЕРИЯ РФ603.

- в строке "CAN Standard Identifier" устанавливаем стандартный идентификатор CAN;
  - в строке "CAN Extended Identifier" устанавливаем расширенный идентификатор CAN;
  - в строке "CAN Identifier" устанавливаем вид идентификатора (соответственно кадра) с которым работает датчик;
  - в строке "CAN Mode" устанавливаем режим работы CAN интерфейса;
- 1) После установки одного или нескольких требуемых параметров нажмем правой клавиши вызываем меню записи параметров в датчик. Выбираем "Load" (запись одного параметра) либо "Load All" (запись всех параметров).



- 2) Проводим тестирование работы датчика с новыми параметрами.
- 3) Для сохранения новых параметров в памяти датчика нажимаем "Write to FLASH". Теперь при любом последующем включении датчика он будет работать с установленной Вами конфигурацией.
- 4) Для установки параметров датчиков по умолчанию выбираем "Default".
- 5) Для сохранения параметров датчика на диске выбираем "Write to file".
- 6) Для восстановления параметров датчика с диска выбираем "Read from file".

## 11. БИБЛИОТЕКА RF60X. ОПИСАНИЕ ФУНКЦИЙ

С лазерным датчиком поставляется, SDK (software development kit), 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.

### 11.1. 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.

**11.2. 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.

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

**ТРИАНГУЛЯЦИОННЫЕ ЛАЗЕРНЫЕ ДАТЧИКИ. СЕРИЯ РФ603.**

- 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_helloCmd (
    HANDLE          hCOM,
    BYTE           bAddress,
    LPRF60xHELLOANSWER lprfHelloAnswer
);

```

**Параметры:**

- hCOM* – дескриптор устройства, полученного от функции RF60x\_OpenPort либо CreateFile;
- bAddress* – адрес устройства;
- lprfHelloAnswer* – указатель на структуру RF60xHELLOANSWER.

**Возвращаемое значение:**

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

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

Функция **RF60x\_ReadParameter** читает внутренние параметры датчика РФ603 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_LASER_BRIGHT	Уровень яркости лазера
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

## ТРИАНГУЛЯЦИОННЫЕ ЛАЗЕРНЫЕ ДАТЧИКИ. СЕРИЯ РФ603.

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*.

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

Функция **RF60x\_FlushToFlash** сохраняет все параметры во FLASH-память датчика РФ603:

```

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 сохранения всех параметров во FLASH-память, функция возвращает FALSE, иначе, если от датчика получено подтверждение о записи, функция возвращает TRUE.

### 11.6. Восстановление во FLASH-памяти значений 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.

### 11.7. 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 результат, функция возвращает FALSE, иначе функция возвращает TRUE.

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

Функция **RF60x\_Measure** читает из датчика РФ603 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 *pResult  
);
```

**Параметры:**

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

### 11.9. Запуск потока измерений (RF60x\_StartStream)

Функция **RF60x\_StartStream** переводит датчик РФ603 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.

**11.10. 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.

**11.11. Получение результатов измерений из потока (RF60x\_GetStreamMeasure)**

Функция **RF60x\_GetStreamMeasure** читает из входного буфера COM-порта данные, полученные от датчика РФ603 после успешного выполнения функции RF60x\_StartStream. В буфер данные приходят со скоростью, установленной в параметрах датчика РФ603. Т.к. глубина входного буфера ограничена 1024 байтами, то желательно вычитывать данные с периодичностью, равной установленной в параметрах датчика РФ603. 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*.

**11.12. Передача пользовательских данных (RF60x\_CustomCmd)**

Функция **RF60x\_CustomCmd** используется для передачи и/или приёма данных от датчика РФ603.

```
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 РФ603. 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 будут сохранены данные полученные от датчика РФ603. 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.

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

При работы 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.

## 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 РФ603, 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 РФ603
    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 */

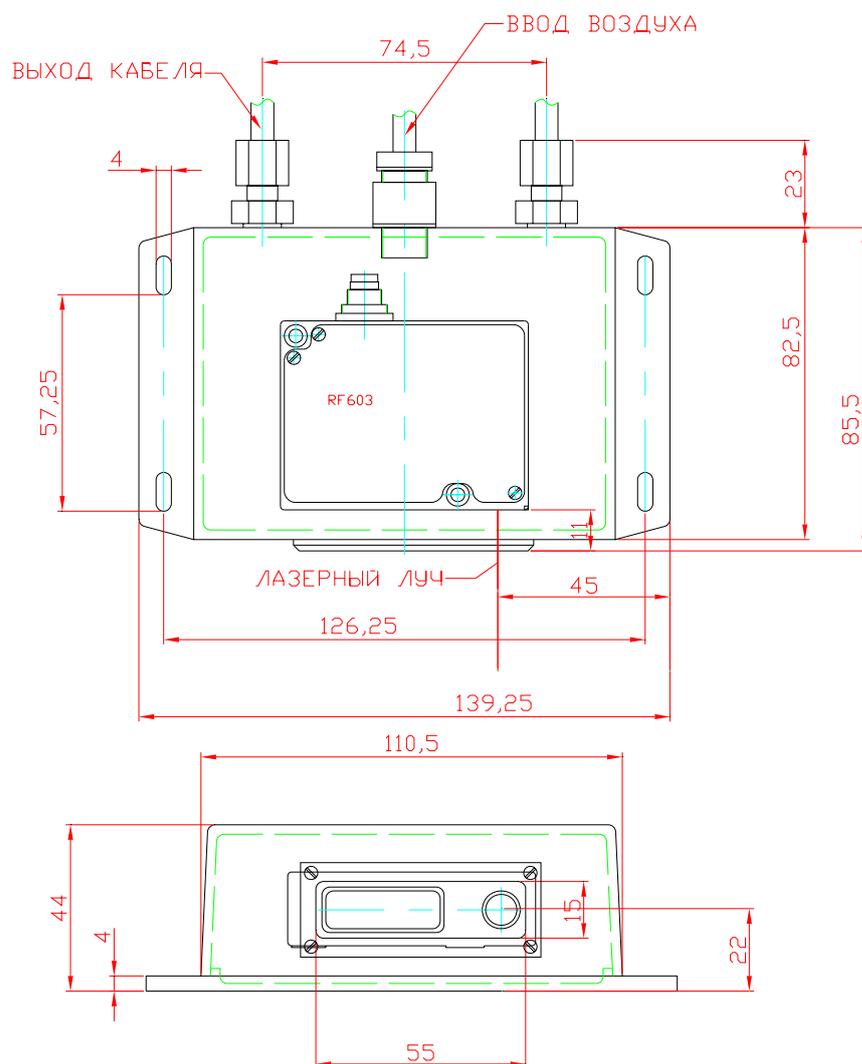
    // Получаем результат
    RF60x_Measure( hRF60x, 1, &usMeasured);

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

**ПРИЛОЖЕНИЕ.**
**Защитный корпус**

Защитный корпус с воздушным охлаждением может быть использован при работе датчика в условиях повышенных температур или высокого загрязнения. Габаритные и установочные размеры корпуса показаны на рисунке 1П. Основные требования:

- 1) Температура сжатого воздуха на входе датчика  $< 25^{\circ}\text{C}$
- 2) Воздух должен быть очищен от масла и влаги
- 3) Максимальная допустимая окружающая температура  $120^{\circ}\text{C}$  для давления воздуха 6 атм.
- 4) Датчик калибруется непосредственно в корпусе, поэтому при использовании датчика без корпуса линейность характеристики нарушается.


**12. КОМПЛЕКТ ПОСТАВКИ**

Датчик Серии РФ603	1 шт
CD с данным описанием, RF60X-SP (исполнимый модуль) и SDK	1 шт

**13. ГАРАНТИЙНЫЕ ОБЯЗАТЕЛЬСТВА**

Гарантийный срок эксплуатации датчика РФ603 - 18 месяцев со дня ввода в эксплуатацию, гарантийный срок хранения - 12 месяцев.