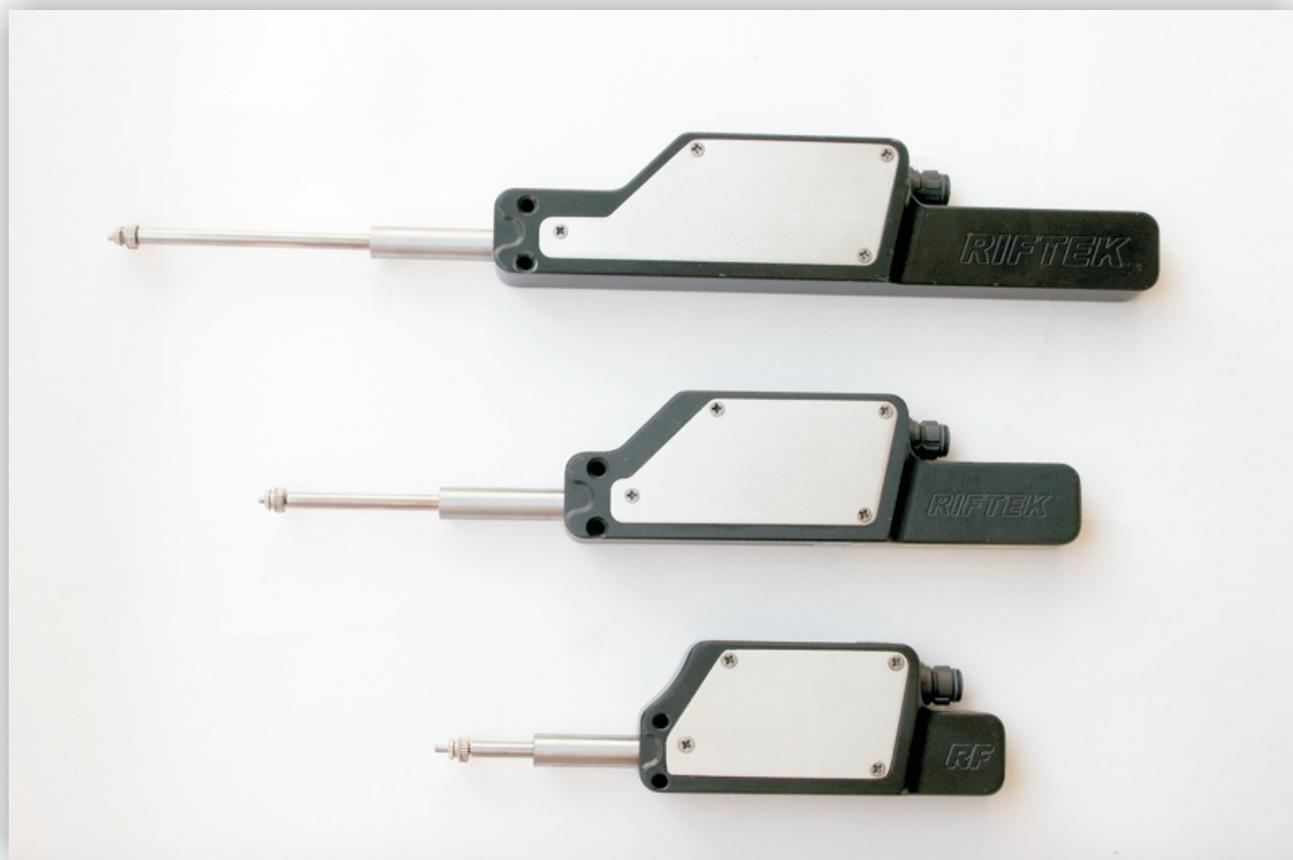


**RIFTEK**  
Sensors & Instruments



## АБСОЛЮТНЫЕ ЛИНЕЙНЫЕ ДАТЧИКИ ПОЛОЖЕНИЯ

**Серия РФ25х**

### 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.	Назначение .....	4
4.	Основные технические характеристики .....	4
5.	Пример обозначения при заказе.....	5
6.	Устройство и принцип работы.....	5
7.	Габариты и установка.....	6
8.	Подключение .....	7
9.	Схемотехника входов-выходов.....	8
9.1.	Вход синхронизации .....	8
9.2.	Логические выходы.....	8
10.	Режимы работы и конфигурационные параметры .....	9
10.1.	Режимы работы.....	9
10.2.	Конфигурационные параметры.....	9
10.2.1.	Режим выборки (Synchronization control byte) .....	9
10.2.2.	Период выборки (Sampling Period) .....	9
10.2.3.	Параметр линейной коррекции (Line Correction) .....	10
10.2.4.	Параметр задания начала системы координат (Zero Position).....	10
10.2.5.	Задание границ аналогового выхода (Analog output range, BEG and END) ...	10
10.2.6.	Масштабирование аналогового выхода (AnalogOutScope) .....	11
10.2.7.	Задание порогов срабатывания логических выходов (LowLimit, UpLimit) .....	11
10.2.8.	Полярность логических выходов (Out Logic Polarity).....	11
11.	Индикация состояния .....	11
12.	Описание последовательного интерфейса.....	12
12.1.	Порт RS232. ....	12
12.2.	Порт RS485. ....	12
12.3.	Конфигурационные параметры.....	12
12.3.1.	Скорость передачи данных через последовательный порт. ....	12
12.3.2.	Сетевой адрес.....	12
12.4.	Таблица заводских значений параметров.....	13
12.5.	Протокол обмена .....	13
12.5.1.	Формат последовательной посылки данных .....	13
12.5.2.	Типы сеансов связи.....	13
12.5.3.	Запрос.....	13
12.5.4.	Сообщение, MSG. ....	13
12.5.5.	Ответ.....	13
12.5.6.	Поток данных.....	14
13.	Коды запросов и список параметров.....	14
13.1.	Таблица кодов запросов.....	14
13.2.	Список параметров.....	14
13.3.	Примечания.....	15
14.	Программа настройки.....	16
14.1.	Назначение .....	16
14.2.	Установка соединения с датчиком.....	16
14.3.	Проверка работоспособности датчика .....	17
14.4.	Настройка и сохранение параметров датчика .....	18
14.4.1.	Настройка параметров.....	18
14.4.2.	Сохранение параметров.....	18
15.	Библиотека RF25X. Описание функций. ....	18
15.1.	Подключение к COM-порту (RF25x_OpenPort).....	19
15.2.	Отключение от COM-порта (RF25x_ClosePort) .....	19
15.3.	Идентификация устройства (RF25x_HelloCmd) .....	19
15.4.	Чтение параметров (RF25x_ReadParameter) .....	20

15.5.	Чтение неопределённых параметров (RF25x_ReadCustomParameter) .....	21
15.6.	Запись неопределённых параметров (RF25x_WriteCustomParameter).....	22
15.7.	Запись параметров (RF25x_WriteParameter).....	22
15.8.	Сохранение текущих параметров во FLASH-памяти (RF25x_FlushToFlash) .....	22
15.9.	Восстановление во FLASH-памяти значений параметров по умолчанию .....	23
15.10.	Защелкивание текущего результата (RF25x_LockResult).....	23
15.11.	Получение результата измерения (RF25x_Measure).....	23
15.12.	Запуск потока измерений (RF25x_StartStream) .....	24
15.13.	Останов потока измерений (RF25x_StopStream) .....	24
15.14.	Получение результатов измерений из потока (RF25x_GetStreamMeasure) .....	25
15.15.	Задание начала отсчета системы координат (RF25x_SetDatumPoint).....	25
15.16.	Передача пользовательских данных (RF25x_CustomCmd).....	25
15.17.	Функции для работы с датчиками, подключенными к USB с помощью FTDI.....	26
15.17.1.	Подключение к USB через FTDI (RF25x_FTDI_OpenPort).....	26
15.18.	Пример.....	27
16.	Принадлежности.....	27
16.1.	Устройство цифровой индикации .....	27
17.	Комплект поставки.....	28
18.	Гарантийные обязательства .....	28
19.	Приложение 1. Датчики производства РИФТЭК .....	28

## 1. Меры предосторожности

- Используйте напряжение питания и интерфейсы, указанные в спецификации на датчик.
- При подсоединении/отсоединении кабелей питания датчика должно быть отключено.
- 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 соответствует 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. Назначение

Абсолютные линейные датчики положения (абсолютные линейные энкодеры) предназначены для измерения и контроля размеров, положения, перемещения, биений, профиля и расположения поверхностей, деформаций технологических объектов.

Серия включает датчики с измерительным диапазоном от 3 до 55 мм и дискретностью отсчета до 0,1 мкм. Возможны также заказные конфигурации с параметрами, отличными от параметров, указанных ниже.

## 4. Основные технические характеристики

Модель	РФ251-3	РФ251-25	РФ256-15	РФ256-35	РФ256-55
Диапазон контроля, мм	3	25	15	35	55
Погрешность (при T=20 °C), мкм	± 2, ± 1 (для датчиков с индексом "P")				± 3, ± 2 (для датчиков с индексом "P")
Дискретность отсчета, мкм	0.1 или 0.5 или 1				
Выходной интерфейс	цифровой	RS485 или RS323			
	аналоговый	0...20 мА (нагрузка 500 Ом) или 0...10 В			
Вход внешней синхронизации	нет	опто-изолирован			
Логические выходы	два выхода, NPN: 100 мА max; 40 В max				
Индикатор работы	нет	двухцветный светодиод			
Напряжение питания, В	12...36				
Макс. потребляемая мощность, Вт	0,75				
Класс защиты	IP67	IP50			
Рабочая температура, °C	-40...+50	-10...+50			
Вес (без кабеля), г	70	110	110	150	180
Габариты, рисунок	2	3	4	5	6

## 5. Пример обозначения при заказе

### РФ25Х.Р-Л-D-UART-AN-IN-LOUT-M

Символ	Наименование
X	Исполнение (1 или 6)
P	Повышенная точность
L	Рабочий диапазон, мм
D	Дискретность, мкм
UART	Тип последовательного интерфейса (RS232 или RS485)
AN	Наличие аналогового выхода по току ( I ) или по напряжению ( U )
IN	Наличие входа внешней синхронизации (только для датчиков РФ256)
LOUT	Наличие пороговых логических выходов
M	Длина кабеля, м

**Пример.** РФ256-55-0.1-232-I-IN-CC-3, датчик РФ256, стандартная точность; диапазон 55 мм; дискретность 0,1 мкм; последовательный порт – RS232; наличие токового выхода 4...20мА; IN – наличие входа синхронизации; длина кабеля 3 м.

## 6. Устройство и принцип работы

В основу работы датчиков положен принцип фотоэлектрического сканирования специальной шкалы, связанной с измерительным наконечником датчика. Излучение светодиода 1 коллимируется объективом 2 и направляется на шкалу 3. Изображение участка шкалы 3 считывается CMOS линейкой 4 и анализируется процессором сигналов, который рассчитывает положение шкалы.

По сравнению с инкрементными датчиками абсолютные датчики обладают рядом преимуществ: абсолютное положение измерительного наконечника определяется немедленно после включения питания и исключает необходимость поиска референтной метки; исключена вероятность накопления или потери счетных импульсов при ударах, вибрациях и реверсе.

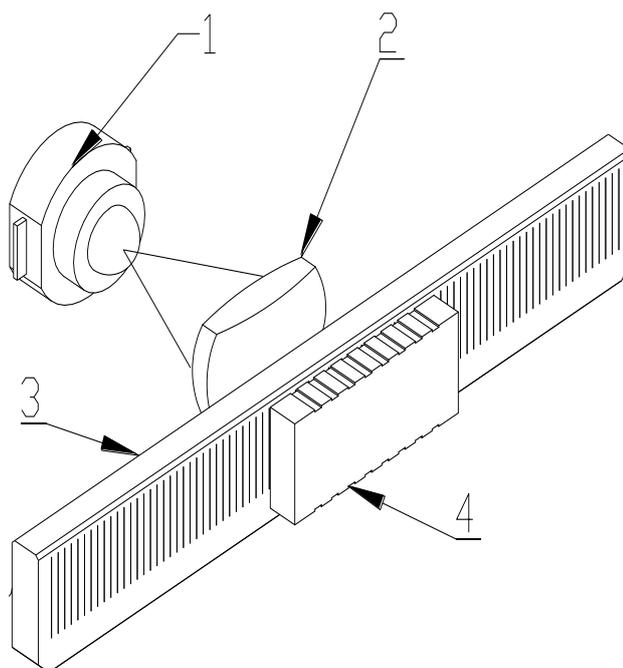


Рисунок 1

## 7. Габариты и установка

Габаритные и установочные размеры датчиков РФ25Х показаны на рис.2...6. Корпус датчика выполнен из анодированного алюминия. Датчики РФ251 снабжены кабельным вводом, датчики РФ256 – разъемом (Binder 620)

6

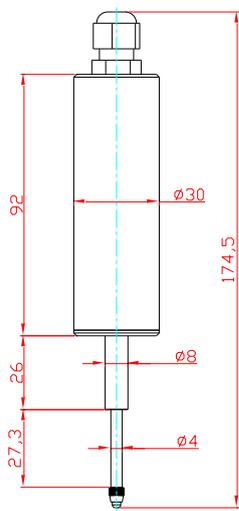


Рисунок 2

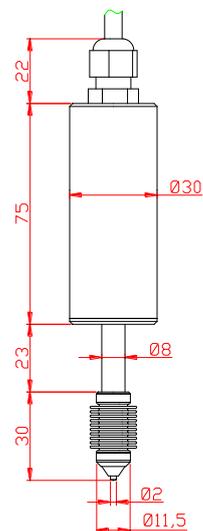


Рисунок 3

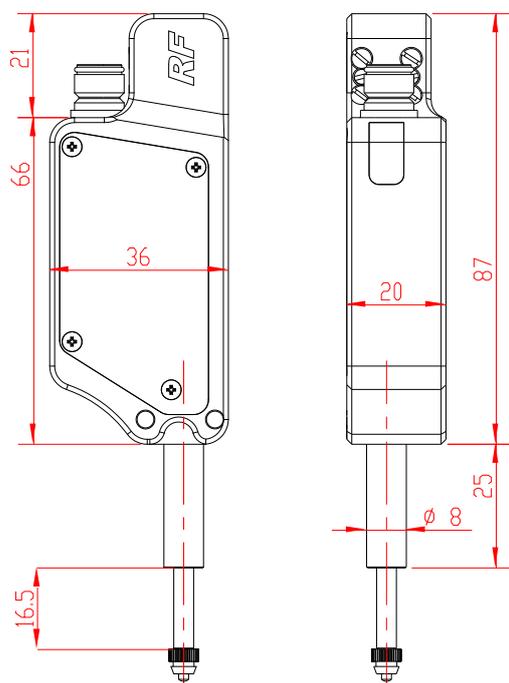


Рисунок 4

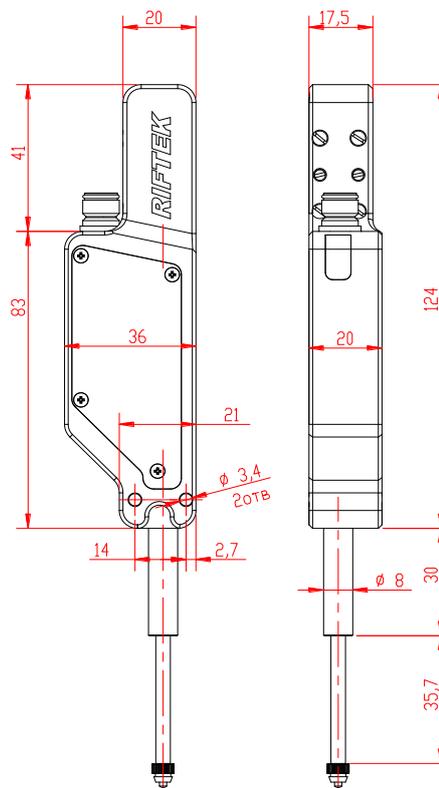


Рисунок 5

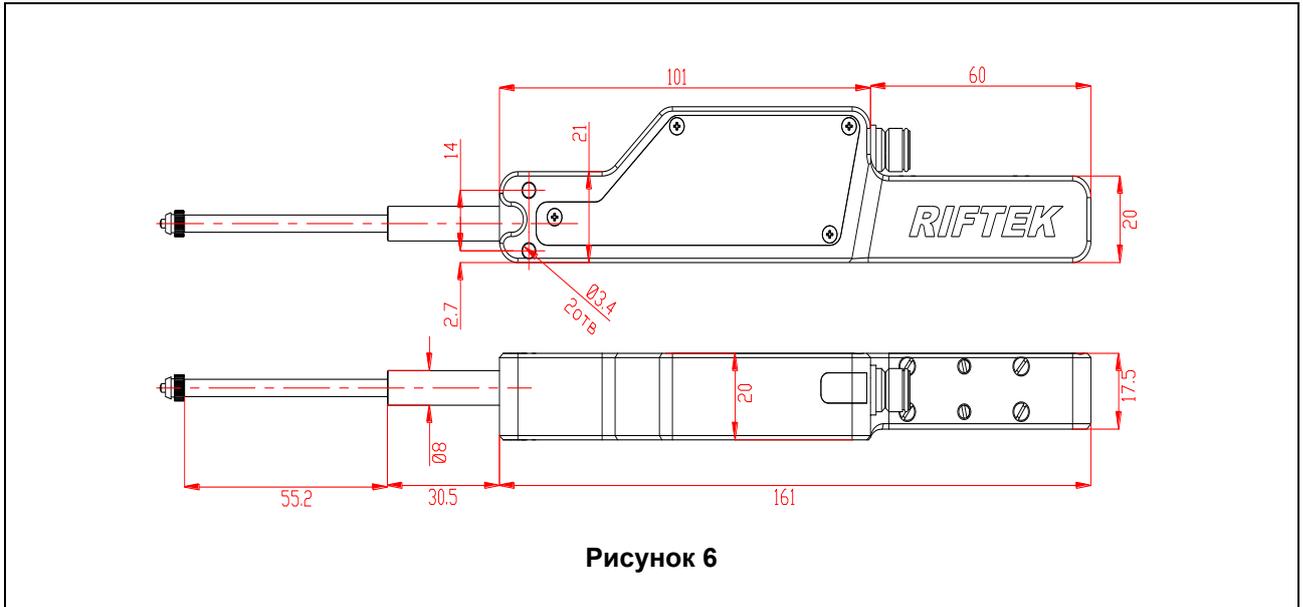


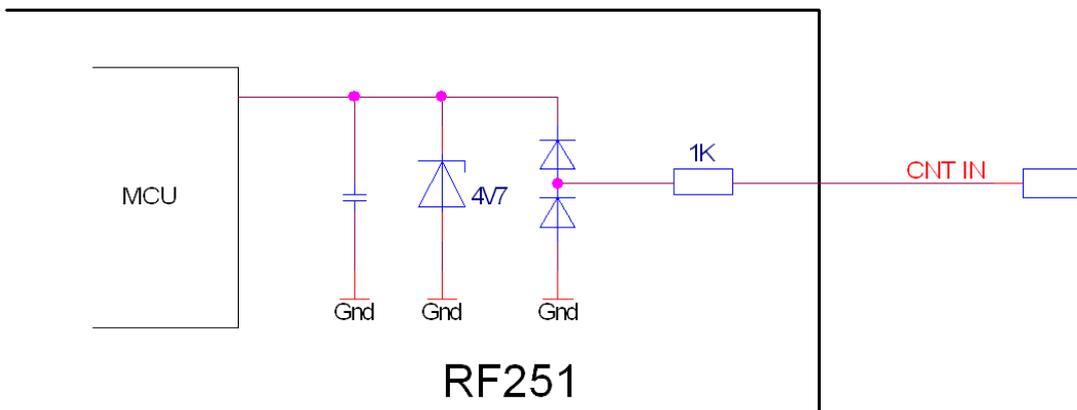
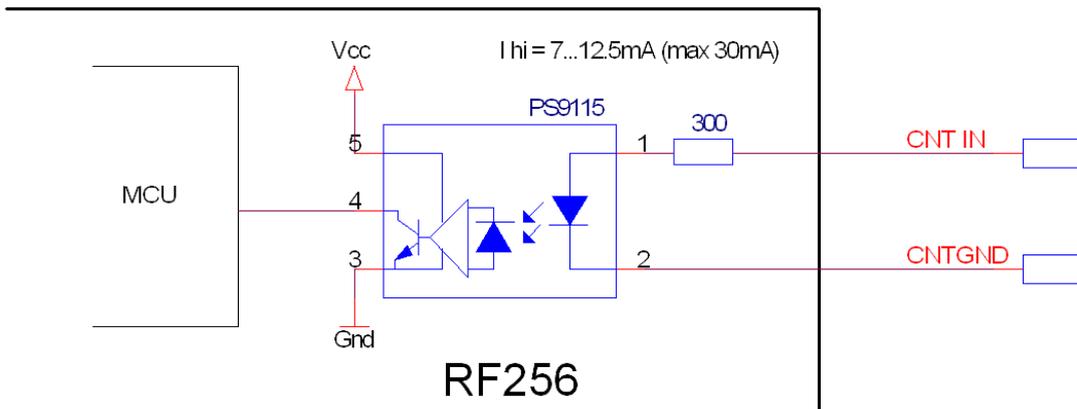
Рисунок 6

## 8. Подключение

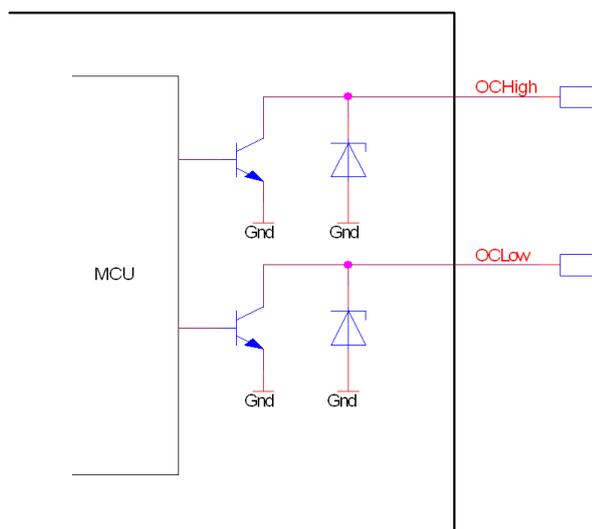
Модель	Обозначение	Контакты DB9F	Цвет провода
232-U/I-IN	Power U+	-	Красный
	Power U-	-	Коричневый
	TXD	2	Зеленый
	RXD	3	Желтый
	IN	-	Белый
	Gnd (Общий для сигналов)	5	Серый
	U/I (UpLimitOC) AGND (LowLimitOC)	- -	Синий Розовый
485-U/I-IN	Power U+	-	Красный
	Power U-	-	Коричневый
	DATA+	8	Зеленый
	DATA-	7	Желтый
	IN	-	Белый
	Gnd (Общий для сигналов)	5	Серый
	U/I (UpLimitOC) AGND (LowLimitOC)	- -	Синий Розовый

## 9. Схемотехника входов-выходов

### 9.1. Вход синхронизации



### 9.2. Логические выходы



## 10. Режимы работы и конфигурационные параметры

### 10.1. Режимы работы.

Данные от датчика (результаты измерений) могут быть получены по последовательному интерфейсу и/или на аналоговом выходе. По последовательному интерфейсу результаты можно получить двумя способами:

- по разовым запросам;
- автоматически потоком данных.

Изменение сигнала на аналоговом выходе происходит синхронно с изменением результата, передаваемого по последовательному интерфейсу. При использовании интерфейса стандарта RS485 к одному устройству сбора информации можно подключить несколько датчиков (сетевой вариант использования).

### 10.2. Конфигурационные параметры

Характер работы датчика определяют его конфигурационные параметры, изменение которых производится передачей команд через последовательный порт. Основные параметры:

#### 10.2.1. Режим выборки (Synchronization control byte)

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 положения в режиме потока (вход в режим по команде "Запрос потока результатов") автоматически по последовательному интерфейсу передает результат измерения в соответствии с заданным временным интервалом (параметр "Период выборки"). При установке режима выборки по внешнему входу датчик передает результат при каждом переключении внешнего входа с учетом установленного коэффициента деления (параметр "Период выборки")

#### 10.2.2. Период выборки (Sampling Period)

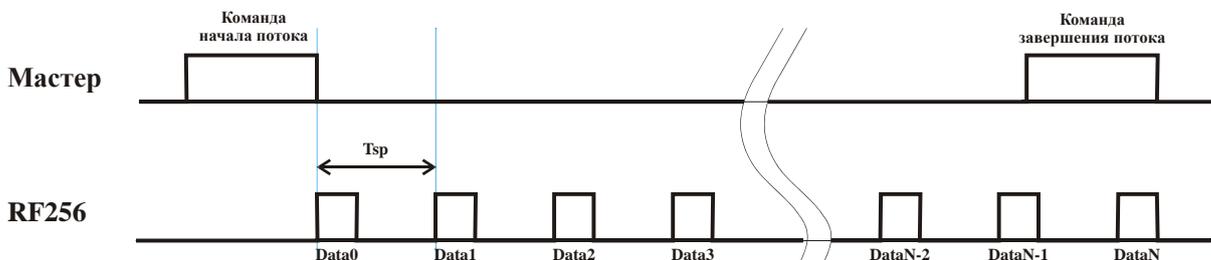
Если установлен режим выборки по времени, то параметр "период выборки" определяет интервал времени, через который датчик должен автоматически передать результат измерения. Значение интервала времени задается в дискретах по 0.01мс. **Например**, для значения параметра, равного 100, данные по последовательному 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, результат измерения заносится в буфер и хранится в нем до поступления нового результата. Указанные параметры определяют способ выдачи результата из этого буфера.

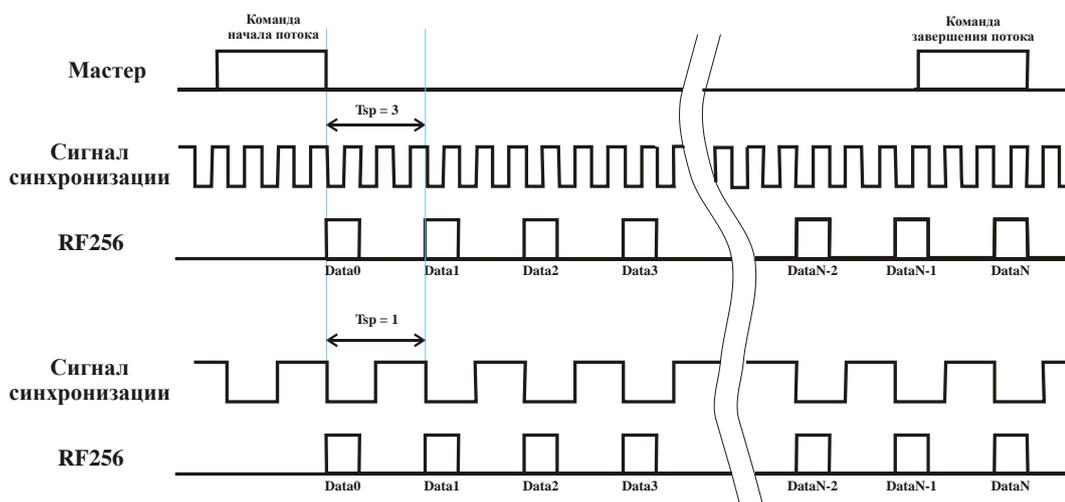
**Примечание 2.** Если для приема результата используется последовательный интерфейс, то при задании малых интервалов периода выборки следует учитывать время, необходимое для передачи данных на выбранной скорости передачи. Если время передачи превосходит период выборки, то именно оно будет определять темп передачи данных.

Принцип передачи данных поясняется диаграммами:



$T_{sp} = \text{Sampling Period} * 0.01\text{mS}$  - период выборки      Data0, Data1 ... DataN - синхронизированные пакеты данных

**Передача данных в потоке с синхронизацией по времени**



$T_{sp} = \text{Sampling Period}$  - период выборки      Data0, Data1 ... DataN - синхронизированные пакеты данных

**Передача данных в потоке с внешней синхронизацией**

**10.2.3. Параметр линейной коррекции (Line Correction)**

используется для настройки линейности датчика (*изменение параметра может привести к ухудшению точности датчика*).

**10.2.4. Параметр задания начала системы координат (Zero Position)**

используется для задания точки нуля датчика

**10.2.5. Задание границ аналогового выхода (Analog output range, BEG and END)**

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.

### 10.2.6. Масштабирование аналогового выхода (AnalogOutScore)

Параметр для калибровки размаха аналогового выхода (*изменение параметра может привести к ухудшению точности аналогового выхода*)

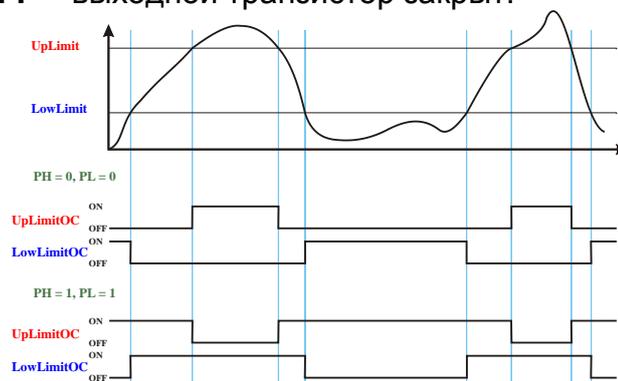
### 10.2.7. Задание порогов срабатывания логических выходов (LowLimit, UpLimit)

Параметр **LowLimit** задает порог переключения логического выхода **LowLimitOC**. Если измеренное значение меньше чем **LowLimit**, выход **LowLimitOC** устанавливается в активный логический уровень, иначе на **LowLimitOC** устанавливается неактивный логический уровень (полярность активного и неактивного логических уровней задается параметром **Out Logic Polarity**). Параметр **UpLimit** задает порог переключения логического выхода **UpLimitOC**. Если измеренное значение больше чем **UpLimit**, выход **UpLimitOC** устанавливается в активный логический уровень, иначе на **UpLimitOC** устанавливается неактивный логический уровень (полярность активного и неактивного логических уровней задается параметром **Out Logic Polarity**).

### 10.2.8. Полярность логических выходов (Out Logic Polarity)

Задаёт полярность активного и не активного состояния логических выходов **LowLimitOC** и **UpLimitOC**.

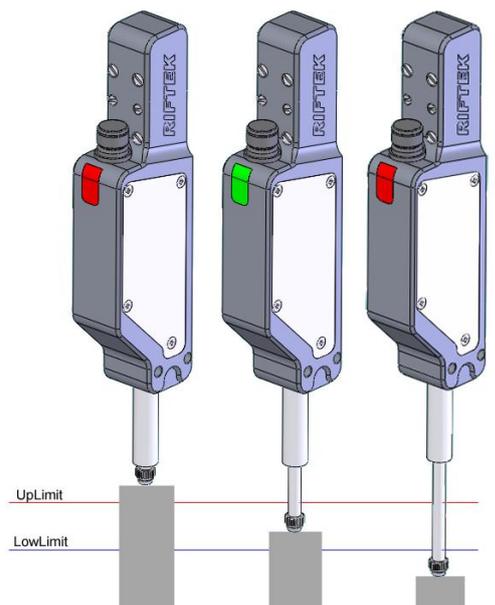
На рисунке показаны диаграммы переключения логических выходов в зависимости от состояния битов полярности **HP** – бит полярности выхода **UpLimitOC** и **LP** – бит полярности выхода **LowLimitOC**. Состояние **ON** – выходной транзистор открыт, состояние **OFF** – выходной транзистор закрыт.



**Примечание.** Корректное изменение параметров производится с помощью программы настройки "RF25x-SP", поставляемой с датчиком.

## 11. Индикация состояния

Датчики модели **РФ256** оборудованы светодиодной индикацией с двумя состояниями: "красный" и "зелёный". Работа индикации напрямую связана со значениями параметров **LowLimit** и **UpLimit**. Светодиодная индикация находится в состоянии "зелёный", если результат измерения находится в диапазоне **LowLimit < Результат < UpLimit**, т.е. результат измерения не выходит за установленные пределы. Светодиодная индикация находится в состоянии "красный", если результат измерения находится в диапазоне **Результат < LowLimit**, или **UpLimit < Результат** т.е. результат измерения выходит за какой-либо установленный предел.



## 12. Описание последовательного интерфейса

### 12.1. Порт RS232.

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

### 12.2. 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”.

### 12.3. Конфигурационные параметры

#### 12.3.1. 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.

#### 12.3.2. 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 РФ25Х), 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).

## 12.4. Таблица заводских значений параметров.

Наименование параметра	Значение
Скорость передачи данных (интерфейс RS232 или RS485)	115200
Сетевой адрес	1

## 12.5. Протокол обмена.

### 12.5.1. 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.

### 12.5.2. 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"].

### 12.5.3. 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)

### 12.5.4. 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)

### 12.5.5. 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 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):

DAT(7:0)					
Байт 0			Байт 1		
1	CNT(2:0)	DAT(3:0)	1	CNT(2:0)	DAT(7:4)

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

## 13. Коды запросов и список параметров

### 13.1. Таблица кодов запросов

Код запроса	Описание	Сообщение (размер в байтах)	Ответ (размер в байтах)
01h	Идентификация устройства	—	-тип устройства (1) -модификация (1) -серийный номер (2) -зарезервирован (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 (4)
07h	Запрос потока результатов	—	-поток результатов (4)
08h	Прекратить передачу потока	—	—
0Ch	Задание начала системы координат	—	-константа 0Ch (1)

### 13.2. Список параметров

Код параметра	Наименование	Значения
00h	Включение датчика	1 — датчик включен, выполняются измерения (по умолчанию); 0 — датчик отключен, находится в энергосберегающем режиме.
01h	Контрольный байт режима выборки	x,x,x,x,x,x,x,S – контрольный байт, задающий режим выборки бит S, биты x – не используются; бит S: 0 — режим выборки по времени (по умолчанию); 1 — режим выборки по входу синхронизации;
02h	Сетевой адрес	1...127 (по умолчанию — 1)
03h	Скорость передачи данных через последовательный порт	1...192, (по умолчанию — 48) задает скорость передачи данных в дискретах по 2400бод, например значение 4 задает скорость 192*2400=460800бод.

		Доступные скорости: 9600, 19200, 57600, 115200, 230400, 460800
04h...06h	Зарезервированы	
07h	Младший байт параметра начала системы координат	1...167777215, (по умолчанию — 0) задает значение параметра начала системы координат
08h	Средний байт параметра начала системы координат	
09h	Старший байт параметра начала системы координат	
0Ah	Младший байт периода выборки	1) 1...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)
0Bh	Старший байт периода выборки	
0Ch	Младший байт начала диапазона аналогового выхода	0...65535 (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...65535 (по умолчанию – 4000h) определяет точку внутри диапазона датчика, в которой аналоговый выход принимает максимальное значение (20 мА или 10В).
0Fh	Старший байт конца диапазона аналогового выхода	
10h	Младший байт коэффициента масштаба аналогового выхода	1...65535, параметр настройки масштаба аналогового выхода
11h	Старший байт коэффициента масштаба аналогового выхода	
12h	Младший байт порога срабатывания выхода <b>LowLimitOC</b>	1...167777215, (по умолчанию — 0) задает значение порога срабатывания логического выхода <b>LowLimitOC</b>
13h	Средний байт порога срабатывания выхода <b>LowLimitOC</b>	
14h	Старший байт порога срабатывания выхода <b>LowLimitOC</b>	
15h	Младший байт порога срабатывания выхода <b>UpLimitOC</b>	1...167777215, (по умолчанию — 0) задает значение порога срабатывания логического выхода <b>UpLimitOC</b>
16h	Средний байт порога срабатывания выхода <b>UpLimitOC</b>	
17h	Старший байт порога срабатывания выхода <b>UpLimitOC</b>	
18h	Полярность логических выходов <b>LowLimitOC</b> и <b>UpLimitOC</b>	<p>x,x,x,x,x,x,PH,PL – байт, задающий полярность логических выходов PH – полярность выхода <b>UpLimitOC</b>, PL - полярность выхода <b>LowLimitOC</b>, биты x – не используются;</p> <p>бит PL:                      0 — активный логический уровень 0 (выходной транзистор n-p-n открыт) (по умолчанию);                      1 — активный логический уровень 1 (выходной транзистор n-p-n закрыт);</p> <p>бит PH:                      0 — активный логический уровень 0 (выходной транзистор n-p-n открыт) (по умолчанию);                      1 — активный логический уровень 1 (выходной транзистор n-p-n закрыт);</p>

### 13.3. Примечания

- Все значения представлены в двоичном виде.
- По специальному запросу (05h) текущий результат, может быть, зацелкнут 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 широковещательном 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 кон-

троллера 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.

## 14. Программа настройки

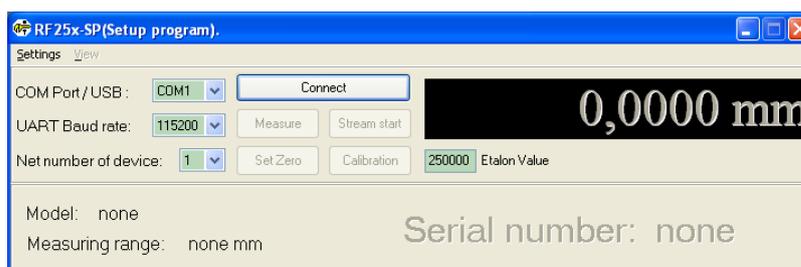
### 14.1. Назначение

Программное обеспечение **RF25X-SP** ([www.riftek.com/resource/files/rf25x-sp.zip](http://www.riftek.com/resource/files/rf25x-sp.zip)) предназначено для:

- тестирования и демонстрации работы датчиков серии RF25х;
- настройки параметров датчика;
- приема и накопления данных.

### 14.2. Установка соединения с датчиком

После запуска программы появляется рабочее окно:



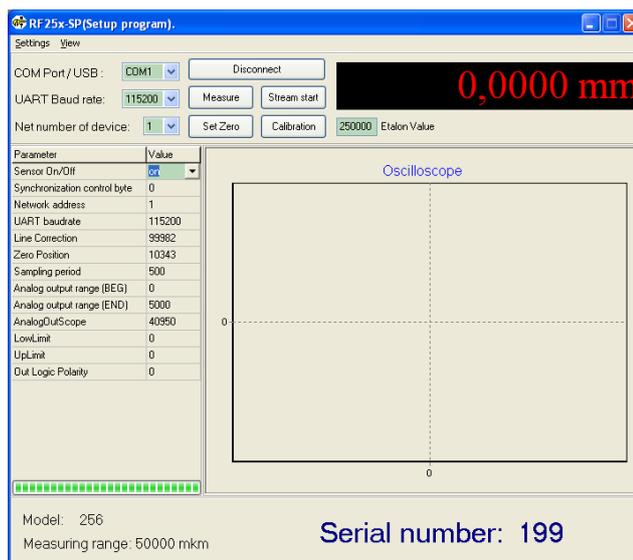
- в строке **UART Baud rate** выбрать скорость, на которой работает датчик (заводская установка – 115200 бит/с),
- в строке **COM Port/USB** выбрать номер порта **RS232** персонального компьютера, к которому подключён датчик, или **USB**, если датчик подключен к компьютеру через **FTDI** преобразователь.
- строка **Net number of device** определяет сетевой адрес датчика (заводская установка для всех датчиков – "1")
- нажать кнопку **Connect**. RF25X-SP попытается наладить связь с датчиком с параметрами, заданными в выше перечисленных строках. Если связь не установлена, выдается сообщение об отсутствии связи.

При успешном соединении:

в строке **Model** индицируется модель датчика;

в строке **Serial number** - серийный номер датчика;

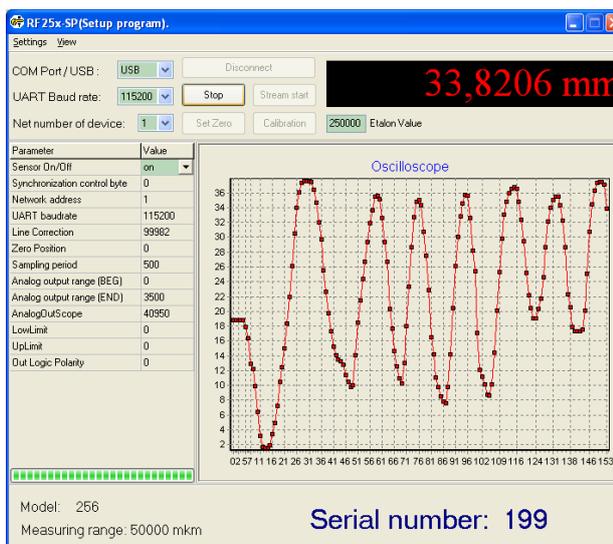
в строке **Measuring range** индицируется рабочий диапазон датчика, и окно трансформируется к следующему виду:



### 14.3. Проверка работоспособности датчика

После успешной установки связи проверяем работоспособность датчика.

- нажатие кнопки **Measure** выводит на панель индикации и панель **Oscilloscope** результаты измерения. (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. [13.1](#)).



- 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. [13.1](#)).
- Перемещая шток датчика, наблюдаем изменение показаний.
- Нажатие кнопки **Stop/Stream stop** останавливает передачу данных.

**Примечание 1.** Данные, поступающие от датчика, накапливаются и сохраняются в кольцевом буфере размером 10000 измерений.

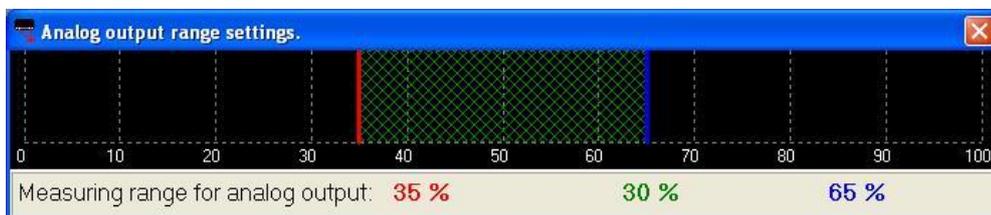
**Примечание 2.**левой клавишей мыши можно изменять масштаб графика, правой – перемещать график в области просмотра. Нажатие правой клавиши вызывает меню **Сохранение данных в файл**.

## 14.4. Настройка и сохранение параметров датчика

Открывшаяся часть приложения RF25X-SP (Таблица **Parameter – Value**) позволяет редактировать и заносить как в RAM, так и во FLASH датчика соответствующие параметры.

### 14.4.1. Настройка параметров

- включение/выключение датчика: дважды щелкните левой клавишей мыши в поле **Value** параметра **Sensor On/Off**;
- установка режима выборки: нажмите клавишу в поле **Value** параметра **Synchronization Control Byte**, вызвав меню редактирования битов приоритета;
- установка скорости обмена: щелкните левой клавишей мыши в поле **Value** строки **UART Baud rate**, вызвав список допустимых скоростей;
- установка границ окна аналогового выхода: в строках **Analog output range** (BEG) и (END). Можно также, дважды щелкнув в поле **Value**, вызвать 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.

Для типа результата "отклонение" границы окна необходимо задать так, чтобы значение, соответствующее нулевому отклонению, располагалось в середине окна. В этом случае, нулевому отклонению будет соответствовать середина диапазона аналогового выхода (12мА или 5В).

### 14.4.2. Сохранение параметров

- После установки одного или нескольких требуемых параметров нажати- ем правой клавиши вызываем меню записи параметров в датчик. Выби- раем **Load** (запись одного параметра) либо **Load All** (запись всех пара- метров).
- Проводим тестирование работы датчика с новыми параметрами.
- Для сохранения новых параметров в энергонезависимой памяти датчика нажимаем **Write to FLASH**. Теперь при любом последующем включении датчика он будет работать с установленной Вами конфигурацией.

## 15. Библиотека RF25X. Описание функций.

С датчиком поставляется, SDK ([www.riftek.com/resource/files/rf25x-sdk.zip](http://www.riftek.com/resource/files/rf25x-sdk.zip)).

SDK позволяет пользователю разрабатывать собственные программные продукты, не вдаваясь в подробности протокола обмена данными с датчиком. SDK содержит:

- ✓ Заголовочный файл RF25x.h с константами и прототипами функций;
- ✓ Подключаемая библиотека RF25x.lib, необходимая для работы с VC++/VS2003/VS2005/Borland C++;
- ✓ Исполняемая библиотека RF25x.dll с набором готовых функций;
- ✓ Примеры для работы с библиотекой RF25x;
- ✓ Настоящее описание

**Примечание:** библиотека RF25x предназначена для ОС Windows 98/Me/2000/XP

### 15.1. Подключение к COM-порту (RF25x\_OpenPort)

Функция **RF25x\_OpenPort** открывает COM-порт с заданным символьным именем, заполняет указатель на дескриптор устройства и возвращает результат операции:

```

BOOL RF25x_OpenPort(
    LPCSTR      lpPort_Name,
    DWORD      dwSpeed,
    HANDLE *    lpHandle
);

```

#### Параметры:

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

#### Возвращаемое значение:

Если COM-порт не открыт, и/или настроить его 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

### 15.2. Отключение от COM-порта (RF25x\_ClosePort)

Функция **RF25x\_ClosePort** закрывает COM-порт и возвращает результат операции:

```

BOOL RF25x_ClosePort(
    HANDLE      hHandle
);

```

#### Параметры:

*hHandle* – дескриптор устройства, полученного от функции RF25x\_OpenPort либо CreateFile;

#### Возвращаемое значение:

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

### 15.3. Идентификация устройства (RF25x\_HelloCmd)

Функция **RF25x\_HelloCmd** производит идентификацию устройства RF25x сети по сетевому адресу и заполняет структуру **RF25xHELLOANSWER**:

```
typedef struct _RF25x_HELLO_ANSWER_ {
    BYTE      bDeviceType;
    BYTE      bcDeviceModificaton;
    WORD      wDeviceSerial;
    WORD      wDeviceMaxDistance;
    WORD      wDeviceRange;
```

Где:

- bDeviceType – однобайтная величина, показывающая тип устройства (для RF25x данная величина равна 65) (тип BYTE);
- bDeviceModificaton – однобайтная величина показывающая модификацию устройства (тип BYTE);
- wDeviceSerial – двубайтная величина, содержащая серийный номер устройства устройства (тип WORD);
- wDeviceMaxDistance – двубайтная величина, содержащая значение максимального расстояния для датчика РФ25x (тип WORD);
- wDeviceRange – двубайтная величина, содержащая значение рабочего диапазона для датчика РФ25x (тип WORD).

```
BOOL RF25x>HelloCmd (
    HANDLE          hCOM,
    BYTE            bAddress,
    LPRF25xHELLOANSWER lprfHelloAnswer
);
```

#### Параметры:

- hCOM* – дескриптор устройства, полученного от функции RF25x\_OpenPort либо CreateFile;
- bAddress* - адрес устройства;
- lprfHelloAnswer*- указатель на структуру RF25xHELLOANSWER.

#### Возвращаемое значение:

Если устройство не ответило на запрос идентификации, функция возвращает FALSE, иначе функция возвращает TRUE и заполняет структуру RF25xHELLOANSWER

### 15.4. Чтение параметров (RF25x\_ReadParameter)

Функция RF25x\_ReadParameter читает внутренние параметры датчика РФ25x и возвращает текущее значение по адресу параметра:

```
BOOL RF25x_ReadParameter (
    HANDLE          hCOM,
    BYTE            bAddress,
    WORD            wParameter,
    DWORD *        lpdwValue
```

**Параметры:**

- hCOM* – дескриптор устройства, полученного от функции RF25x\_OpenPort либо CreateFile;
- bAddress* - адрес устройства;
- wParameter* - номер параметра, см. табл. 1.

Табл. 1

Параметр	Описание
RF25x_PARAMETER_POWER_STATE	Питание датчика
RF25x_PARAMETER_PRIORITY_AND_SYNC	Приоритет и взаимная синхронизация
RF25x_PARAMETER_NETWORK_ADDRESS	Сетевой адрес
RF25x_PARAMETER_BAUDRATE	Скорость передачи данных через последовательный порт
RF25x_PARAMETER_PIXEL_SIZE	Размер пикселя
RF25x_PARAMETER_DATUM_POINT	Начало системы координат
RF25x_PARAMETER_SAMPLING_PERIOD	Период выборки
RF25x_PARAMETER_BEGIN_ANALOG_RANGE	Начало диапазона аналогового выхода
RF25x_PARAMETER_END_ANALOG_RANGE	Конец диапазона аналогового выхода

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

**Возвращаемое значение:**

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

**15.5. Чтение неопределённых параметров (RF25x\_ReadCustomParameter)**

Функция **RF25x\_ReadCustomParameter** читает внутренние параметры датчика РФ25х и возвращает текущее значение по адресу параметра:

```

BOOL RF25x_ReadCustomParameter(
    HANDLE hCOM,
    BYTE bAddress,
    BYTE bParameterAddress,
    BYTE bParametersSize,
    void * lpValue
    )
    
```

**Параметры:**

- hCOM* – дескриптор устройства, полученного от функции RF25x\_OpenPort либо CreateFile;
- bAddress* - адрес устройства;
- bParameterAddress* - адрес внутреннего параметра в датчике РФ25х;
- bParameterSize* - размер внутреннего параметра в датчике РФ25х;
- lpValue* - указатель на массив, в который будут сохранены прочитанные данные;

**Возвращаемое значение:**

Если устройство не ответило на запрос чтения параметра, функция возвращает FALSE, иначе функция возвращает TRUE и заполняет массив переданный через *lpValue*.

## 15.6. Запись неопределённых параметров (RF25x\_WriteCustomParameter)

Функция **RF25x\_WriteCustomParameter** записывает внутренние параметры датчика РФ25х:

```

BOOL RF25x_writeCustomParameter(
    HANDLE          hCOM,
    BYTE            bAddress,
    BYTE            bParameterAddress,
    BYTE            bParameterSize,
    void *          lpValue

```

### Параметры:

*hCOM* – дескриптор устройства, полученного от функции **RF25x\_OpenPort** либо **CreateFile**;  
*bAddress* - адрес устройства;  
*bParameterAddress* - адрес внутреннего параметра в датчике РФ25х;  
*bParameterSize* - размер внутреннего параметра в датчике РФ25х;  
*lpValue* - указатель на массив, который будет сохранён, начиная с адреса *bParameterAddress*;

### Возвращаемое значение:

Если запись параметра не произведена, функция возвращает FALSE, иначе функция возвращает TRUE

## 15.7. Запись параметров (RF25x\_WriteParameter)

Функция **RF25x\_WriteParameter** записывает внутренние параметры датчика РФ25х:

```

BOOL RF25x_writeParameter (
    HANDLE          hCOM,
    BYTE            bAddress,
    WORD            wParameter,
    DWORD           dwValue

```

### Параметры:

*hCOM* – дескриптор устройства, полученного от функции **RF25x\_OpenPort** либо **CreateFile**;  
*bAddress* - адрес устройства;  
*wParameter* - номер параметра, см. табл. 1, детальные значения параметров смотрите в технической документации на датчик РФ25х;  
*dwValue* - значение параметра, которое будет сохранено в датчик РФ25х.

### Возвращаемое значение:

Если запись параметра не произведена, функция возвращает FALSE, иначе функция возвращает TRUE

## 15.8. Сохранение текущих параметров во FLASH-памяти (RF25x\_FlushToFlash)

Функция **RF25x\_FlushToFlash** сохраняет все параметры во FLASH-память датчика РФ25х:

```

BOOL RF25x_FlushToFlash(
    HANDLE hCOM,
    BYTE bAddress
);

```

**Параметры:**

*hCOM* – дескриптор устройства, полученного от функции RF25x\_OpenPort либо CreateFile;  
*bAddress* - адрес устройства.

**Возвращаемое значение:**

Если устройство не ответило на запрос сохранения всех параметров во FLASH-память, функция возвращает FALSE, иначе, если от датчика получено подтверждение о записи, функция возвращает TRUE

## 15.9. Восстановление во FLASH-памяти значений параметров по умолчанию

Функция **RF25x\_RestoreFromFlash** восстанавливает значения всех параметров во FLASH по умолчанию:

```

BOOL RF25x_RestoreFromFlash(
    HANDLE hCOM,
    BYTE bAddress
);

```

**Параметры:**

*hCOM* – дескриптор устройства, полученного от функции RF25x\_OpenPort либо CreateFile;  
*bAddress* - адрес устройства.

**Возвращаемое значение:**

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

## 15.10. Защелкивание текущего результата (RF25x\_LockResult)

Функция **RF25x\_LockResult** защелкивает текущее измеренное значение до следующего вызова функции **RF25x\_Measure**:

```

BOOL RF25x_LockResult(
    HANDLE hCOM,
    BYTE bAddress
);

```

**Параметры:**

*hCOM* – дескриптор устройства, полученного от функции RF25x\_OpenPort либо CreateFile;  
*bAddress* - адрес устройства.

**Возвращаемое значение:**

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

## 15.11. Получение результата измерения (RF25x\_Measure)

Функция **RF25x\_Measure** читает из датчика РФ25х текущее измеренное значение. Значение *lpdwValue* в отсчётах по 0.1мкм:

```

BOOL RF25x_Measure(
    HANDLE          hCOM,
    BYTE           bAddress,
    DWORD *        lpdwValue
);

```

**Параметры:**

*hCOM* – дескриптор устройства, полученного от функции RF25x\_OpenPort либо CreateFile;  
*bAddress* - адрес устройства.  
*lpdwValue* - указатель на переменную типа ULONG/DWORD, содержащую результат D.

**Возвращаемое значение:**

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

**15.12. Запуск потока измерений (RF25x\_StartStream)**

Функция **RF25x\_StartStream** переводит датчик РФ25х в режим непрерывной передачи результатов измерений:

```

BOOL RF25x_StartStream(
    HANDLE          hCOM,
    BYTE           bAddress
);

```

**Параметры:**

*hCOM* – дескриптор устройства, полученного от функции RF25x\_OpenPort либо CreateFile;  
*bAddress* - адрес устройства.

**Возвращаемое значение:**

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

**15.13. Останов потока измерений (RF25x\_StopStream)**

Функция **RF25x\_StopStream** переводит датчик из режима непрерывной передачи результатов измерений в режим «запрос-ответ»:

```

BOOL RF25x_StopStream(
    HANDLE          hCOM,
    BYTE           bAddress
);

```

**Параметры:**

*hCOM* – дескриптор устройства, полученного от функции RF25x\_OpenPort либо CreateFile;  
*bAddress* - адрес устройства.

**Возвращаемое значение:**

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

## 15.14. Получение результатов измерений из потока (RF25x\_GetStreamMeasure)

Функция **RF25x\_GetStreamMeasure** читает из входного буфера СОМ-порта данные, полученные от датчика РФ25х, после успешного выполнения функции **RF25x\_StartStream**. В буфер данные приходят со скоростью, установленной в параметрах датчика РФ25х, т.к. глубина входного буфера ограничена 1024 байтами, то желательно считывать данные с периодичностью, равной установленной в параметрах датчика РФ25х. Параметр *lpdwValue* идентичен параметру *lpdwValue* в функции **RF25x\_Measure**.

```

BOOL RF25x_GetStreamMeasure(
    HANDLE          hCOM,
    DWORD *        lpdwValue
);

```

### Параметры:

*hCOM* – дескриптор устройства, полученного от функции **RF25x\_OpenPort** либо **CreateFile**;  
*lpdwValue* - указатель на переменную типа ULONG/DWORD, содержащую результат D.

### Возвращаемое значение:

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

### Примечание:

Для стабильной работы функции **RF25x\_GetStreamMeasure** необходимо использовать её в отдельном потоке, с приоритетом, не ниже **THREAD\_PRIORITY\_NORMAL**, иначе происходит переполнение входного буфера последовательного порта, что приводит к непредсказуемым результатам

## 15.15. Задание начала отсчета системы координат (RF25x\_SetDatumPoint)

Функция **RF25x\_SetDatumPoint** задаёт записанное в параметрах значение начала отсчёта системы координат:

```

BOOL RF25x_SetDatumPoint (
    HANDLE          hCOM,
    BYTE           bAddress
);

```

### Параметры:

*hCOM* – дескриптор устройства, полученного от функции **RF25x\_OpenPort** либо **CreateFile**;  
*bAddress* - адрес устройства.

### Возвращаемое значение:

Если устройству не удалось установить начало системы координат, функция возвращает FALSE, иначе функция возвращает TRUE

## 15.16. Передача пользовательских данных (RF25x\_CustomCmd)

Функция **RF25x\_CustomCmd** используется для передачи и/или приёма данных от датчика РФ25х.

```

BOOL RF25x_CustomCmd(
    HANDLE          hCOM,
    char *         pcInData,
    DWORD          dwInSize,
    char *         pcOutData,
    DWORD *        pdwOutSize
);

```

**Параметры:**

*hCOM* – дескриптор устройства, полученного от функции RF25x\_OpenPort либо CreateFile;

*pcInData* - указатель на массив данных, который будет передан в датчик РФ25х. 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 РФ25х. 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 данных не удался, то функция возвращает FALSE, иначе функция возвращает TRUE

## 15.17. Функции для работы с датчиками, подключенными к USB с помощью FTDI

При работы с USB устройствами на FTDI-микросхемах, 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 **RF25x\_Measure** для последовательного порта и **RF25x\_FTDI\_Measure** для устройств с FTDI USB. Единственное отличие лишь при работе с функцией **RF25x\_FTDI\_OpenPort** заключается в передаче в качестве имени порта - символьную ссылку

### 15.17.1. Подключение к USB через FTDI (RF25x\_FTDI\_OpenPort)

Функция **RF25x\_FTDI\_OpenPort** открывает соединение через USB с заданным символьным именем, заполняет указатель на дескриптор устройства и возвращает результат операции:

```

BOOL RF25x_FTDI_OpenPort(
    LPCSTR      lpPort_Name,
    DWORD      dwSpeed,
    FT_HANDLE * lpftHandle
);

```

**Пара-****метры:**

*lpPort\_Name* – название USB-устройства (например "D2XX Recovery PID for XP"), полный синтаксис при задании имени USB-устройства можно посмотреть в D2XX Programmers Guide, функция **FT\_W32\_CreateFile** выполняемая с параметром FT\_OPEN\_BY\_DESCRIPTION;

*dwSpeed* - скорость работы через USB-устройство. Параметр идентичен полю BaudRate в структуре FTDCB подробно описанной в **D2XX Programmers Guide**;

*lpftHandle* - указатель на дескриптор устройства;

**Возвращаемое значение:**

Если соединение через USB-порт не открыто и/или настроить его не удалось, функция вернёт FALSE иначе, если соединение через USB-порт открыто и настроено успешно – функция вернёт TRUE. Более детальные сведения об возвращаемых ошибках можно получить с помощью функции FT\_W32\_GetLastError, описанной в **D2XX Programmers Guide**

## 15.18. Пример

```

HANDLE          hRF25x          = INVALID_HANDLE_VALUE;
DWORD           dwValue;
DWORD           dwMeasured;
RF25XHELLOANSWER hAns;

// Чистим структуру RF25XHELLOANSWER
memset(&hAns, 0x00, sizeof(RF25XHELLOANSWER));

// Открываем COM-порт
if (!RF25x_OpenPort("COM1:", CBR_115200, &hRF25x)
    return (FALSE);

// Опрашиваем устройство
if (RF25x_helloCmd( hRF25x, 1, &hAns ))
{

    ////////////////////////////////////////
    //
    // После успешного выполнения RF25x_helloCmd
    // в структуре hAns содержится информация о
    // датчике РФ25x, ответившем на запрос
    //
    ////////////////////////////////////////

    //читаем параметр: Период выборки
    RF25x_ReadParameter(
        hRF25x,
        1,
        RF25X_PARAMETER_SAMPLING_PERIOD
        ,
        &dwValue
    );

    /* В dwValue содержится значение периода выборки
    */

    //Получаем измерение из датчика РФ25x

```

### Примечание:

Для работы с DLL можно также использовать функции **LoadLibrary** для загрузки библиотеки и функцию **GetProcAddress** для получения указателя на функцию.

## 16. Принадлежности

### 16.1. Устройство цифровой индикации

Устройство индикации РФ301 предназначено для работы с датчиками РФ256 и обеспечивает автономную работу датчика благодаря встроенному аккумулятору. Габаритные размеры устройства показаны на рисунке 7. На

передней панели устройства индикации расположен цифровой дисплей, кнопки сброса показаний 1, а также кнопки "Перебор" 2, "Сдвиг" 3 и "Ввод" 4. На нижней панели – разъем для подключения зарядного устройства. В нижней части размещен разъем для подключения к датчику.

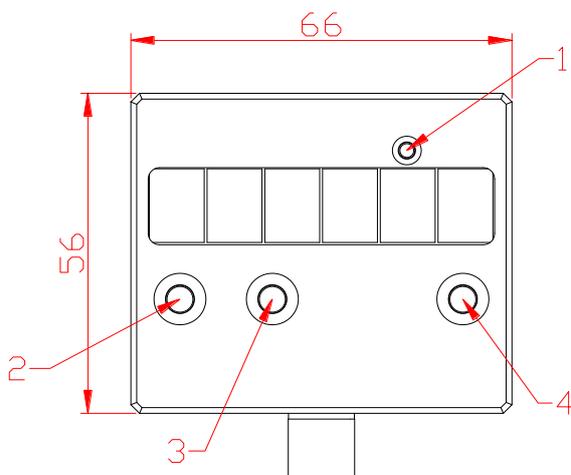


Рисунок 7

## 17. Комплект поставки

Датчик Серии РФ25Х	1 шт
CD с ПО (RF25X-SP и SDK)	1 шт

## 18. Гарантийные обязательства

Гарантийный срок эксплуатации датчика РФ25Х - 24 месяца со дня ввода в эксплуатацию, гарантийный срок хранения - 18 месяцев

## 19. Приложение 1. 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;



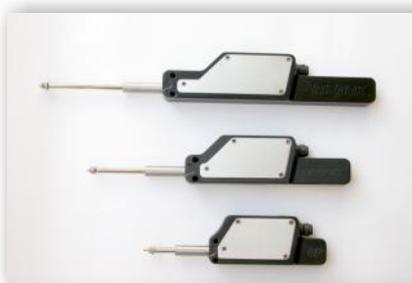
### Оптические микрометры. Серии РФ65х

- 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.

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