

# **ModBus Constructor**

## **Quick start Guide**

© 2006 by KurySoft, all rights reserved

Web: [www.kurysoft.com](http://www.kurysoft.com)

Date: 12-Nov-2006

## СОДЕРЖАНИЕ

1. ВВЕДЕНИЕ.....	3
2. СОЗДАНИЕ ОПИСАНИЯ MODBUS УСТРОЙСТВА.....	5
2.1. Coriolis Mass Flow Meter. ....	5
2.1.1. T1. Mass Flow Meter Holding Registers. ....	5
2.1.2. T2. Mass Flow Meter Coils.....	7
2.1.3. T3. Mass Flow Meter Command. ....	7
2.1.4. T4. Mass Flow Meter Exception Responses. ....	7
2.2. Описание основных параметров. ....	8
2.3. Размещение полей данных на форме. ....	13
2.4. Добавление запросов.....	14
2.5. Параметры симулятора. ....	19
2.5.1. T5. Discrete value simulation. ....	19
2.5.2. T6. Analogous value simulation. ....	19
3. РАБОТА С MODBUS УСТРОЙСТВОМ.....	21
4. ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ. ....	23
4.1. Объекты.....	23
4.2. Типы данных Modbus Constructor. ....	23
4.2.1. Numeric types .....	23
4.2.2. Discrete types .....	23
4.3. Data encoding. ....	23
4.3.1. Long.....	24
4.3.2. Float.....	24
4.3.3. Double.....	24
4.4. Правила использования функций Modbus.....	24
4.5. Format examples.....	25
4.5.1. Numeric types .....	25
4.5.2. Discrete types .....	25
4.5.3. Default for numeric types .....	26
4.5.4. Default for discrete types .....	26

## 1. ВВЕДЕНИЕ.

Спасибо, что вы используете Modbus Constructor!

Modbus Constructor создан для того, чтобы помочь разработчикам, программистам и пользователям Modbus устройств при тестировании и настройке.

Modbus Constructor позволяет быстро создавать модель организации данных Modbus устройства. На рынке существует множество подобных программ, но в них данные представляются как колонки абстрактных чисел одного типа, которые трудны для восприятия и анализа, особенно если устройство содержит в своих регистрах данные различных типов в произвольном порядке.

В Modbus Constructor можно задать для любого регистра или набора регистров все основные типы данных: integer, long (integer), float, double (float) и байт. Целые типы можно отображать в виде signed, unsigned, HEX или binary. Modbus Constructor допускает представление одного и того же регистра несколькими способами. Например, если два смежных регистра содержат число типа long, вы можете дополнительно определить для каждого из этих регистров поля данных, в которых будут представляться отдельные регистры в виде HEX или unsigned integer.

Поля данных, соответствующие элементам данных (регистров или дискретных), можно расположить на форме проекта в произвольном порядке. Можно задавать размер, цвет, шрифт и подпись для каждого поля независимо от других.

Дискретные элементы, занимающие непрерывную область адресов, можно отображать в одном поле для удобства восприятия и экономии места.

Modbus Constructor позволяет задать для каждого поля данных регистров линейное преобразование, которое позволяет переводить внутренние единицы устройства в общепринятые или в единицы другой системы. Например, если регистр содержит величину давления, выраженную в миллиамперах, другими словами, устройство принимает сигнал датчика давления с токовым выходом, можно отобразить эту величину в кгс/см<sup>2</sup> или фунт/дюйм<sup>2</sup> (*давление в империи, не помню как пишется*).

Modbus Constructor позволяет задать автоматические запросы для всех поддерживаемых функций (01, 02, 03, 04, 05, 06, 15, 16, 22, 23), которые будут выполняться в соответствии с заданным для каждого запроса индивидуально периодом.

Можно определить запросы, которые выполняются по желанию пользователя при помощи меню или диалогов. Это удобно для задания команд управления устройством, изменения параметров и тестовых запросов.

Проект может использоваться для опроса устройства в режиме Master, а так же для симуляции устройства в режиме Slave. Дополнительно для режима Slave можно задать изменение данных по определенному закону.

Modbus Constructor позволяет добавлять на форму проекта дополнительные надписи, рамки, а так же рисунки.

Вместе с Modbus Constructor поставляется специальная утилита Modbus Reader, которая использует модель, созданную при помощи Modbus Constructor для работы с устройством.

Modbus Reader это свободно распространяемая программа. Если вы уже создали проект для разработанного вами устройства, вашим клиентам нет необходимости

специально приобретать Modbus Constructor – достаточно скачать отдельный дистрибутив.

Modbus Reader позволяет работать с несколькими проектами одновременно, т.е. опрашивать или симулировать несколько устройств, используя один или несколько COM-портов.

Помимо запросов, определенных в проекте, возможно изменения любых элементов данных устройства простым кликом на связанное с ними поле данных как в режиме Master (путем отправки соответствующего запроса), так и в режиме Slave.

## 2. СОЗДАНИЕ ОПИСАНИЯ MODBUS УСТРОЙСТВА.

В этом разделе вы, шаг за шагом, узнаете как создать проект на примере устройства кориолисов массовый расходомер (Coriolis Mass Flow Meter).

- вы опишите Holding registers и Coils этого прибора
- преобразуете данные в стандартные величины
- примените форматирование данных
- разместите поля данных на форме
- определите автоматические и управляемые из меню запросы (команды)
- зададите параметры для симулятора в режиме Slave

### 2.1. Coriolis Mass Flow Meter.

Обратимся к описанию на прибор. Этот прибор измеряет массу жидкости или газа, прошедших через него. Массомер может работать как в прямом, так и в обратном направлении. При потоке жидкости в прямом направлении, счетчик массы инкрементируется, в обратном - декрементируется, и может принимать отрицательные значения.

Кроме того, он может работать как дозатор. Для этого у него имеются два дискретных выхода (открытый коллектор) для управления клапаном или двигателем, в прямом и обратном направлении. Величина порции устанавливается пользователем. Если установлена положительная величина порции – включается расход в прямом направлении, если отрицательная – в обратном направлении.

Прибор имеет интерфейс RS-485, протокол Modbus RTU.

Запустите Modbus Constructor:

**Start ⇒ Programs ⇒ ModbusConstructor ⇒ MBConstructor.exe**

Создайте новый проект: **File ⇒ New**

Сохраните проект в файле с именем “**Coriolis.mbc**”: **File ⇒ Save as...**

Давайте посмотрим, как реализован протокол обмена Modbus для данного массомера. Описание регистров массомера приводится в таблице T1.

#### 2.1.1. T1. Mass Flow Meter Holding Registers.

Назначение	Адрес (decimal)	Тип данных	Комментарий
Тип устройства	0000	unsigned integer	Read-only
Версия программного обеспечения	0001	unsigned integer	Read-only
Modbus адрес	0002	unsigned integer	заводская установка 1
Скорость передачи	0003	unsigned integer	может быть 4800, 9600 или 19200 бод заводская установка 9600 при записи любого значения не равного 4800, 9600 или 19200 принимает значение 9600

Назначение	Адрес (decimal)	Тип данных	Комментарий
Четность	0004	byte (MSB)	0-no parity 1-odd parity 2-even parity заводская установка 2 (even)
Стоп бит	0004	byte (LSB)	0 – 1 стоп бит 2 – два стоп бита
Величина порции	0005	long (1)	(3)
Счетчик массы	0007	long (1)	одна единица равна 0.1 кг может быть отрицательным при чтении отображает накопленную массу после сброса счетчика
Массовый расход	0009	integer	одна единица равна 0.1 кг/мин положительный при потоке в прямом направлении отрицательный при потоке в обратном направлении
Плотность	0010	unsigned integer	одна единица равна 1000/32768 кг/м <sup>3</sup>
Температура	0011	integer	одна единица равна 250/511 °C
Давление	0012	unsigned integer	сигнал 4-20 мА датчика избыточного давления 1023 соответствует 20 мА (40 кгс/см <sup>2</sup> ) 205 соответствует 4 мА (0 кгс/см <sup>2</sup> )
Время измерения	0013	unsigned long (1)	секунды
Контрольный регистр	0015	unsigned integer	(2)
Масса на один импульс	0016	unsigned integer	одна единица равна 0.1 кг
Калибровочные коэффициенты	0017- 0024		доступно для записи при установленной перемычке J7
Резерв	0025- 0031		не используется в данной версии

(1) Длинные целые размещаются в регистрах следующим образом:  
старшее слово в первом регистре, младшее слово во втором.  
Например, число 01020304h (16909060 decimal) размещается так:  
регистр A+0: 0102h  
регистр A+1: 0304h  
передается: 01 02 03 04

(2) При чтении отображает режим работы:  
0 – режим измерения (по умолчанию)  
1 – неактивный режим дозатора. Оба клапана закрыты.  
2 – активный режим дозатора. Открыт один из клапанов, отмеряется порция.

При записи выполняет следующие действия:

- 0 – переходит в режим измерения из неактивного режима дозатора. Игнорируется, если дозатор в активном режиме.
- 1 – если находится в режиме измерения, переходит в неактивный режим дозатора, сбрасывает счетчики массы и времени, закрывает оба клапана.  
если находится в активном режиме дозатора, переходит в неактивный режим.
- 2 – начинает отмерять порцию в соответствии со значением регистра 5 (величина порции)
- 4 – сбрасывает счетчики массы и времени (только в режимах 0 и 1)
- 8 – проводит корректировку нуля расхода (только в режимах 0 и 1)

(3) используется для хранения величины порции при работе в режиме дозатора.

Одна единица равна 0.1 кг.

Описание Coils приводятся в таблице T2.

### 2.1.2. T2. Mass Flow Meter Coils.

Назначение	Адрес (decimal)	Тип данных	Комментарий
Выход для управления расходом в прямом направлении	0000		Read-Only в режиме дозатора 1-клапан открыт 0-клапан закрыт
Выход для управления расходом в обратном направлении	0001		Read-Only в режиме дозатора 1-клапан открыт 0-клапан закрыт
Резерв	0002-0031		не используется в данной версии

Массомер поддерживает следующие команды и исключения:

### 2.1.3. T3. Mass Flow Meter Command.

Function code	Command type	Name
01	Read	Read coil status
03	Read	Read holding registers
05	Write	Write single coil
06	Write	Write single register
15	Write	Write multiple coils
16	Write	Write multiple registers

### 2.1.4. T4. Mass Flow Meter Exception Responses.

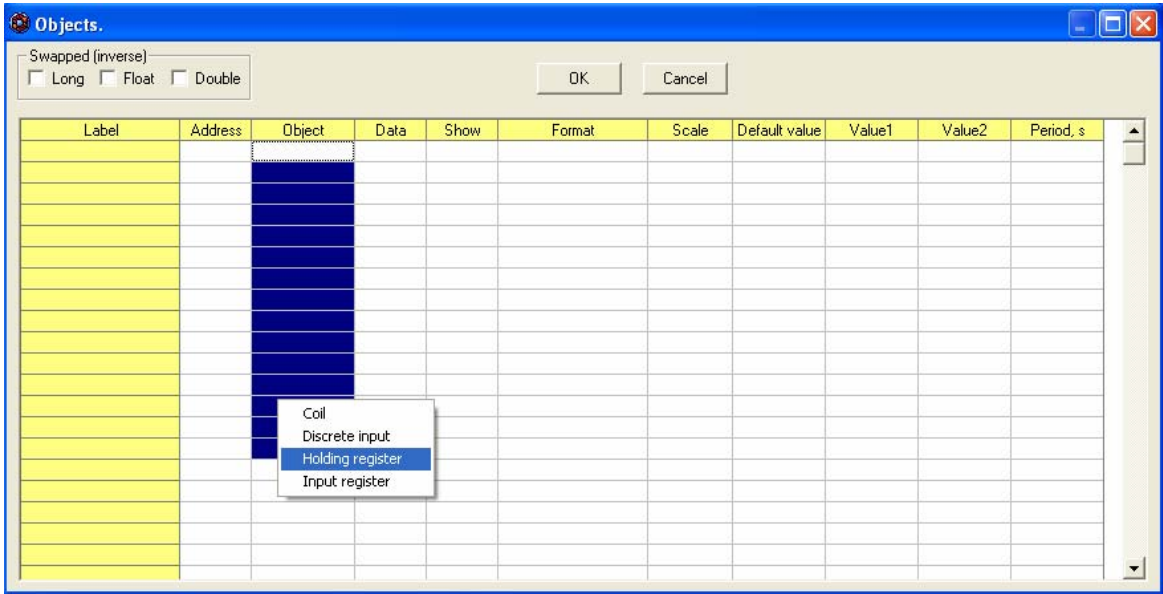
Code	Name	Description
01	Illegal function	Function code not supported.
02	Illegal data address	The data address in the query is not an allowable address for the device.
03	Illegal data value	A value contained in the query data field is not an

		allowable value for the device.
--	--	---------------------------------

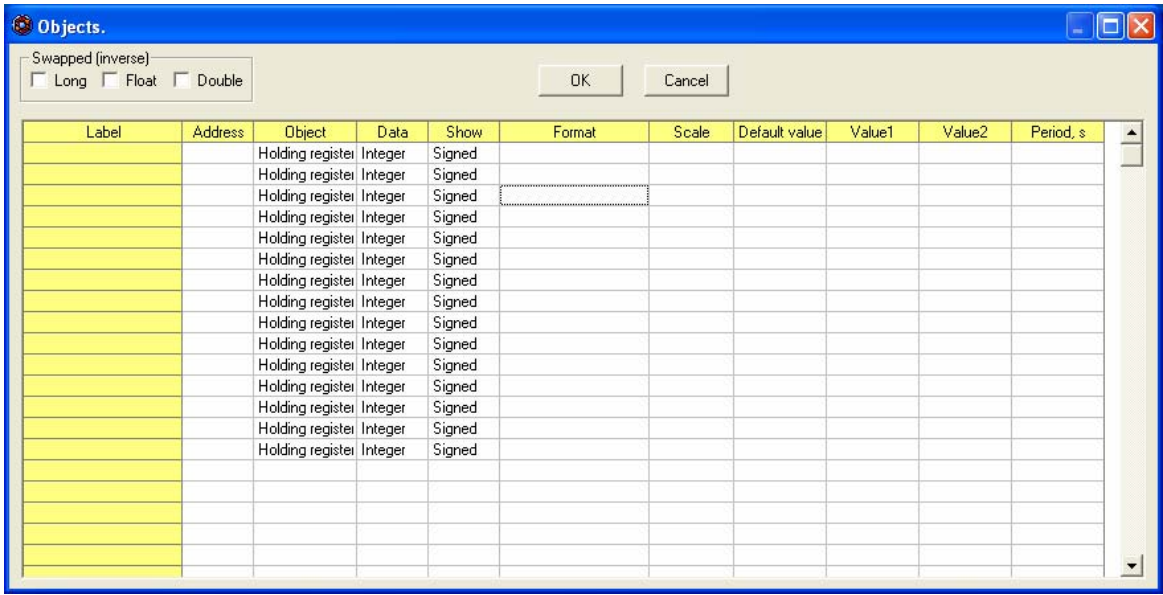
2.2. Описание основных параметров.

Опишите регистры и Coils массомера. Для этого в меню **Settings** выберите вкладку **Objects**:

**Settings⇒ Objects**



Появится форма похожая на лист электронной таблицы. Выделите в колонке **Object** пятнадцать клеточек по числу регистров массомера. Кликните правой кнопкой на выделенный участок, появится выпадающее меню, содержащее все типы используемых объектов. Кликните на пункте **Holding register**.



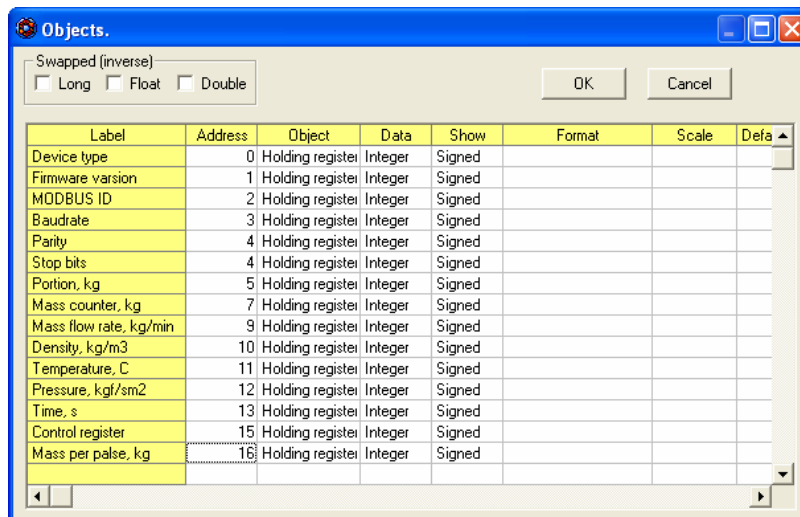
Теперь вам надо присвоить каждому регистру адрес и название в соответствии с описанием.



левой кнопкой дважды кликните на верхнюю пустую клетку в колонке **Label**  
 введите название регистра “Device type”

левой кнопкой дважды кликните на соседнюю клетку в колонке **Address**,  
 введите “0”

введите названия и адреса для остальных элементов.

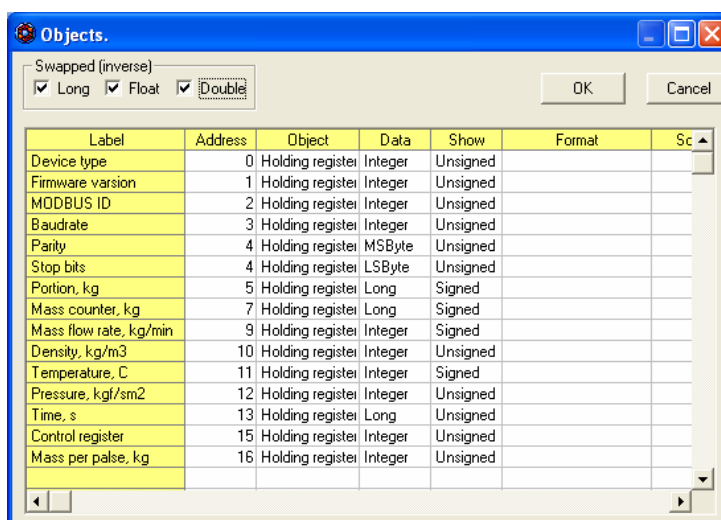


Числа (**long**, **float** и **double**), для представления которых используется несколько регистров, могут размещаться в регистрах и, соответственно, передаются по-разному. Как правило, производитель оборудования явно указывает, как представляются такие числа. Пользуйтесь разделом “**Data Encoding**” для определения способа размещения чисел в регистрах.

Из описания мы видим, что прибор использует разные типы данных. Установите нужные типы.

кликните правой кнопкой в колонке **Data** и выберите в выпадающем меню нужный пункт

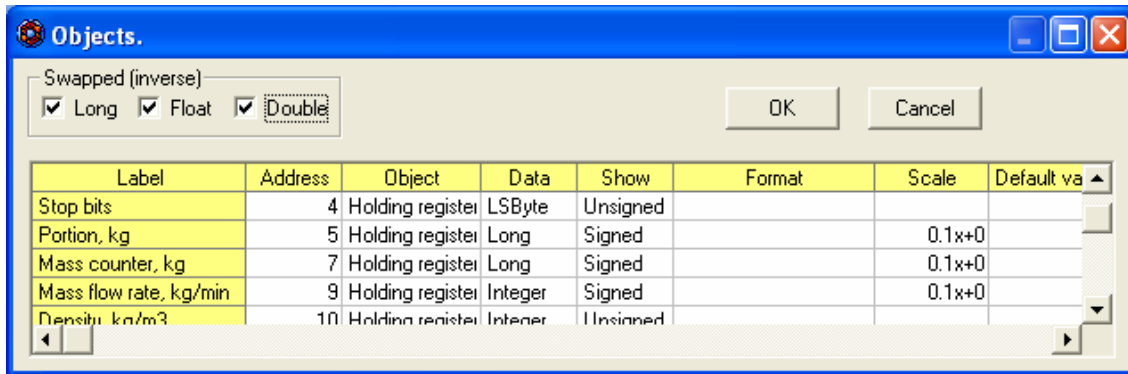
если нужно, установите свойство **Unsigned** в колонке **Show**.



Мы указали для массы единицы измерения в килограммах (кг), но ведь у счетчика массы одна единица равна 0.1 кг? В Modbus Constructor предусмотрен механизм для

каких случаев. Если в колонке **Scale** ввести выражение вида  $Ax+B$ , то над числом, содержащимся в данном регистре или группе регистров (если это число сложного типа **long**, **float** или **double**), будет произведено линейное преобразование. Это очень удобно, если необходимо использовать единицы измерения отличные от тех, что поддерживает прибор.

Запишите в колонке **Scale** для регистров 5, 7, 9 и 15 выражение "0.1x+0". Теперь вы перевели внутренние единицы массомера в килограммы.



Таким же образом вы можете преобразовать плотность:

$$D = x * 1000 / 32768 = 0.03052x$$

И температуру:

$$T = x * 250 / 511 = 0.489x$$

Несколько сложнее обстоит дело с давлением, придется решить уравнение:

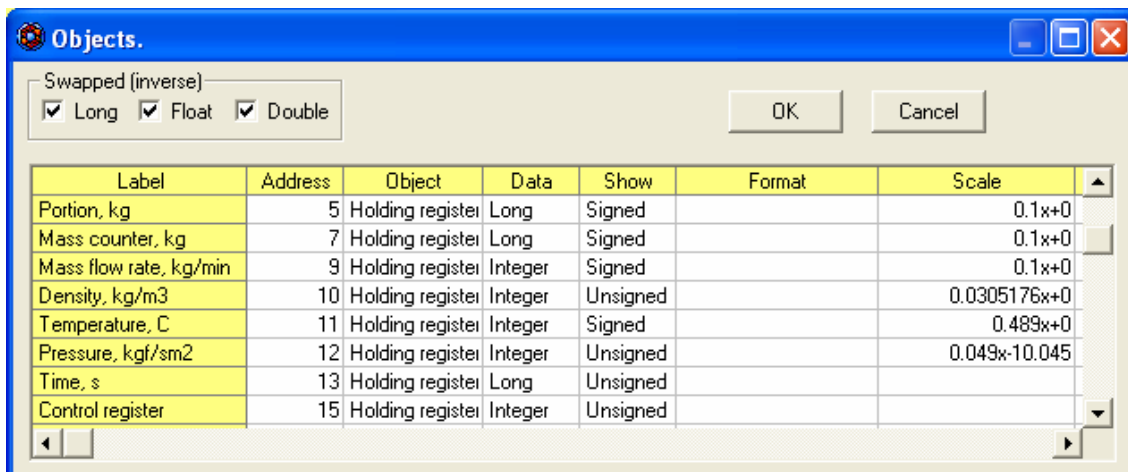
Давление  $P = Ax + B$

$$x_1 = 205, \quad P_1 = 0 \quad \Rightarrow \quad B = -205 * A$$

$$x_2 = 1023, \quad P_2 = 40 \quad \Rightarrow \quad 40 = A * 1023 + 205 * B$$

$$A = 40 / (1023 - 205) = 0.049$$

$$B = -205 * A = -205 * 0.049 = -10.045$$



Как видите, совсем не сложно преобразовывать величины в привычный вид. Но как быть со временем? Для коротких интервалов секунды вполне годятся, однако когда счет идет на сотни и тысячи секунд, это становится неудобным. Для решения этой проблемы мы применим форматирование.

Форматирование данных применяется:

во-первых, чтобы сделать числа более наглядными

во-вторых, избежать ситуации, когда лишние цифры переполняют поле вывода.

Для каждого типа данных определен формат по умолчанию. Нет необходимости указывать формат по умолчанию, если он вас устраивает – оставьте поле пустым. Используйте формат по умолчанию для создания собственного формата. Подробнее о правилах форматирования вы можете посмотреть в разделе **“Format examples”**.

Кликните правой кнопкой в строке **“Portion, kg”** в колонке **Format**.

Выберите пункт **Default**. В поле появится строка вида **“## ###”**.

Массомер измеряет массу и массовый расход с точностью 0.1 кг и 0.1 кг/мин соответственно. Если мы оставим формат по умолчанию, то не увидим дробную часть. Чтобы показать ее надо добавить к строке **“.0”**. Получится:

**“# ### ### ##0.0”** для массы и отмеряемой порции

**“## ##0.0”** для расхода

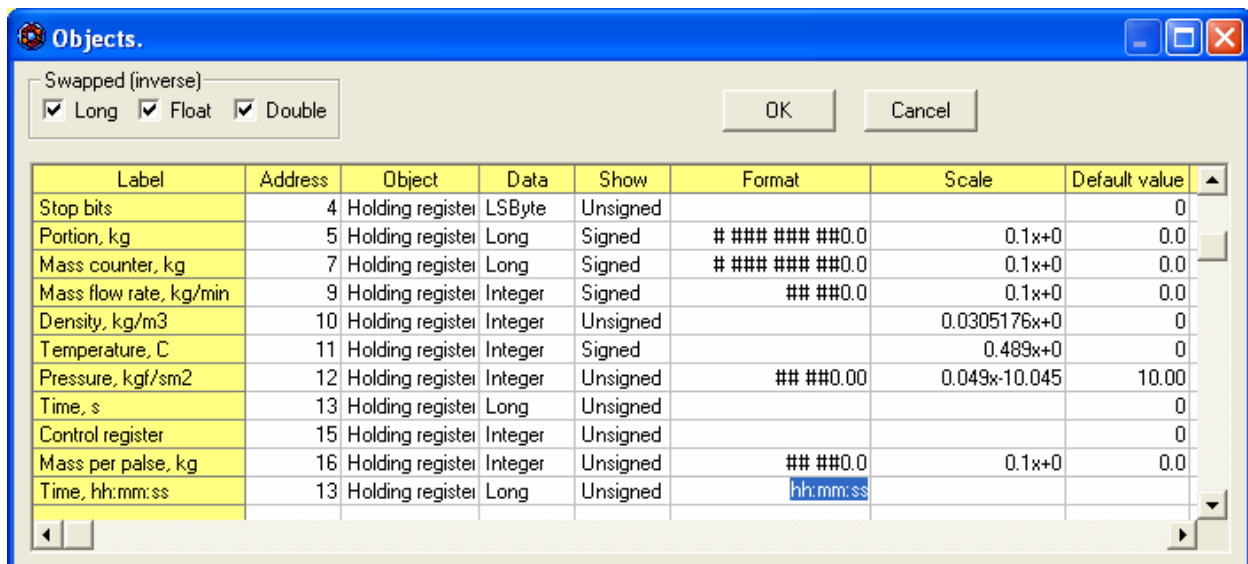
Так как аналого-цифровой преобразователь сигнала датчика давления обеспечивает разрешение 0.05 кг/см<sup>2</sup>, имеет смысл показать два знака после запятой:

**“## ##0.00”**

Введите в поле **Default value** для регистра **“Pressure, kgs/cm<sup>2</sup>”** число **“10”**.

Нажмите Enter. Вы увидите, что число показано в виде **“10.00”**.

Температуру и плотность оставьте целыми.



Сделайте отображение времени измерения в двух режимах. Пусть первый режим останется в секундах, а для второго примените форматирование вида **“hh:mm:ss”**, чтобы показать часы : минуты : секунды соответственно.

дважды кликните на пустом поле в колонке **Object**

выберите **Holding register**

дважды кликните в этой же строке в колонке **Data**

выберите тип **Long**

установите свойство **Unsigned** в колонке **Show**

введите в поле **Label** "Time, hh:mm:ss"

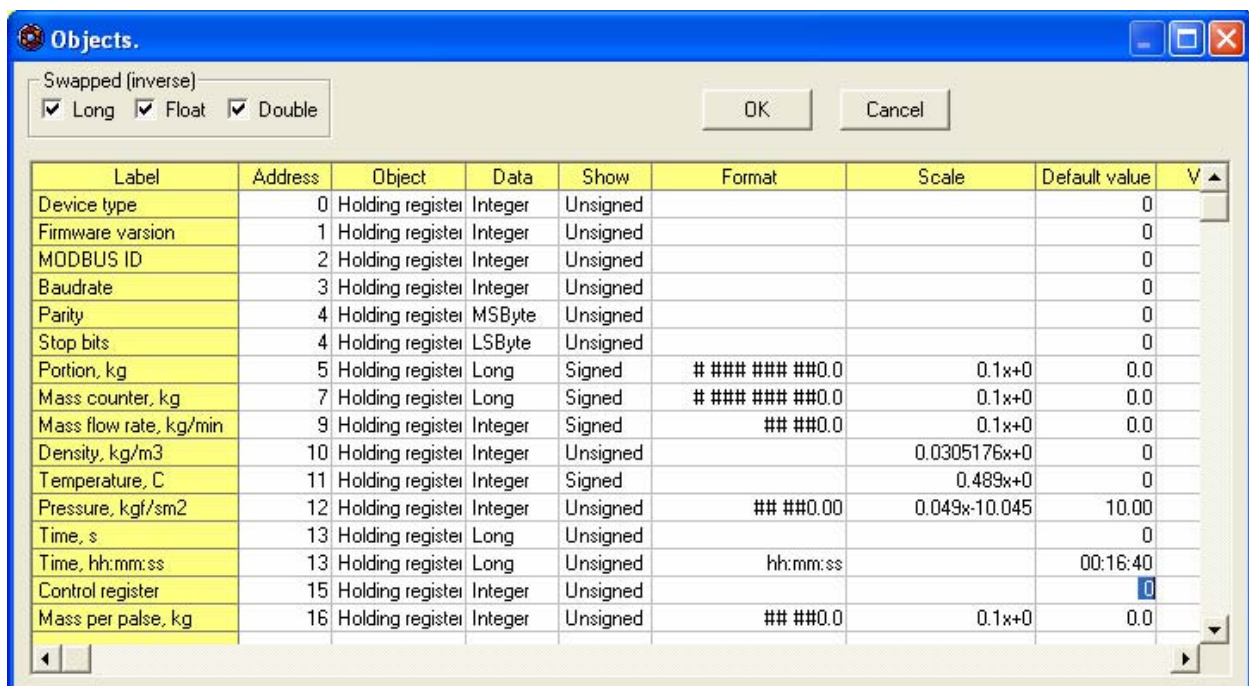
введите в поле адреса "13"

введите в поле **Format** "hh:mm:ss"

введите в поле **Default value** "1000" (время в секундах)

кликните по названию колонки **Address**, чтобы отсортировать регистры по порядку

1000 секунд теперь отображаются в виде "00:16:40"



И наконец, опишите Coils массомера:

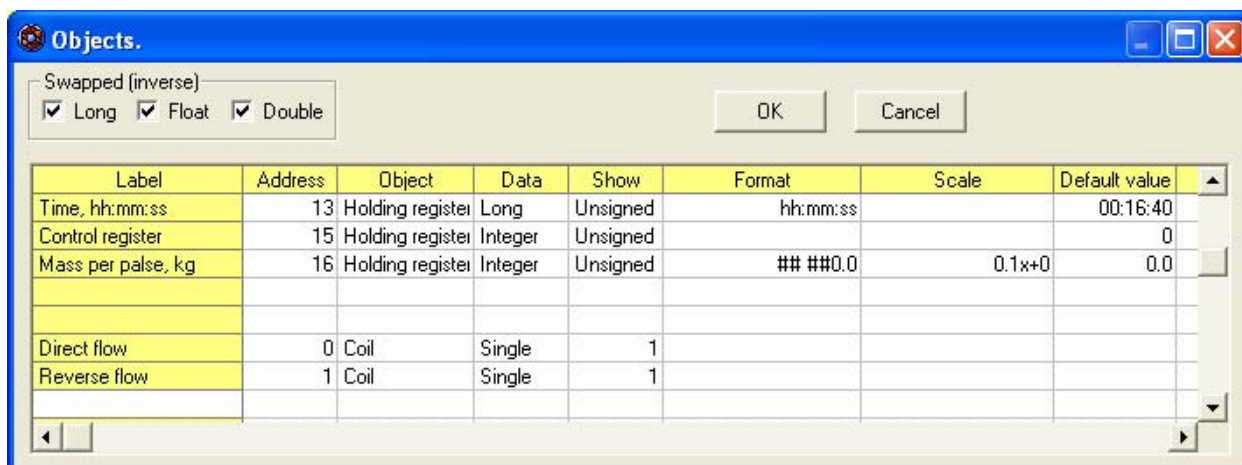
выделите мышкой две пустых ячейки в колонке Object

кликните на них правой кнопкой

в выпадающем меню выберите вкладку Coils

назовите одну Coil "Прямой поток", присвойте ей адрес "0"

назовите другую Coil "Обратный поток", присвойте ей адрес "1"

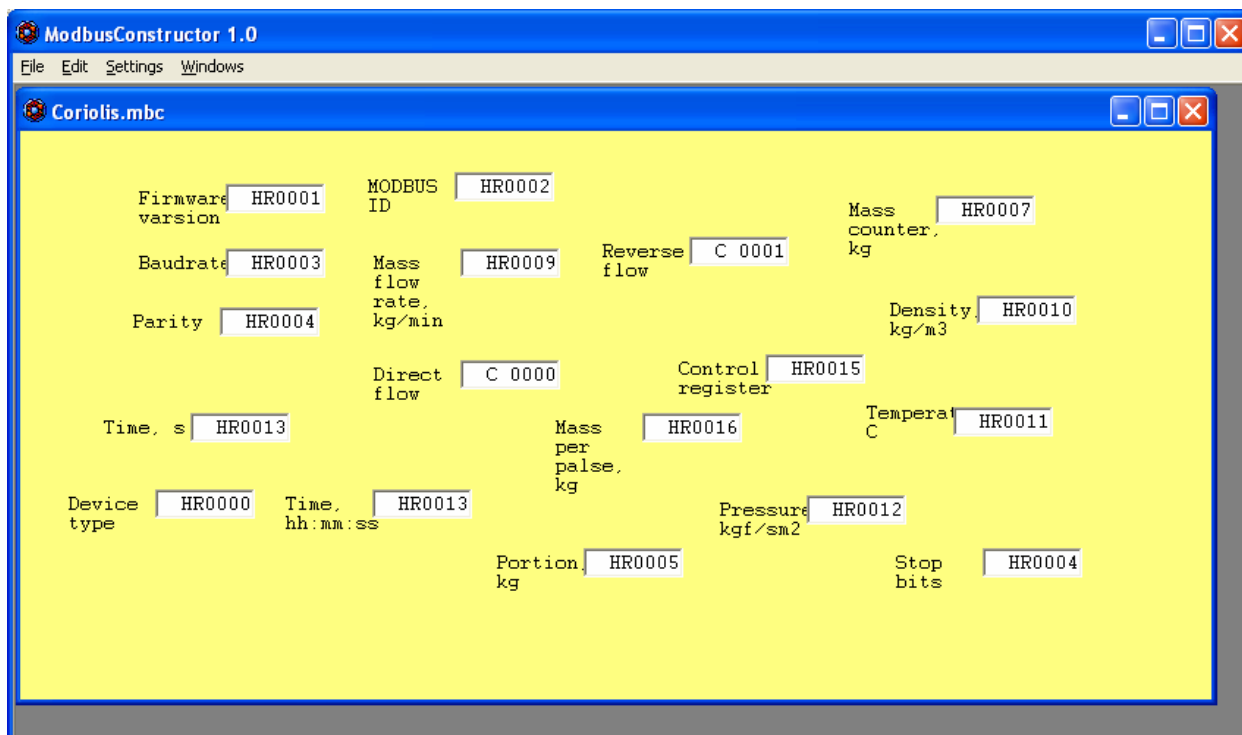


Вы описали все основные свойства объектов массомера. Нажмите кнопку “OK”.  
Сохраните проект: **File⇒Save**.

### 2.3. Размещение полей данных на форме.

В правом верхнем углу формы вы увидите поля данных и метки созданных нами объекты. Кликните левой кнопкой мыши на белой клетке. Она выделится темно-синим цветом. Нажмите левой кнопкой на выделенную клетку и, не отпуская кнопки, переместите ее в сторону. Для множественного выделения нажмите клавишу **Ctrl**, и, удерживая ее, выделите нужные объекты.

То же самое проделайте с остальными клетками так, чтобы видеть все подписи. Обратите внимание, когда вы перемещаете поле, вместе с ним перемещается и его метка, но не наоборот – когда вы перемещаете метку, поле остается на месте.

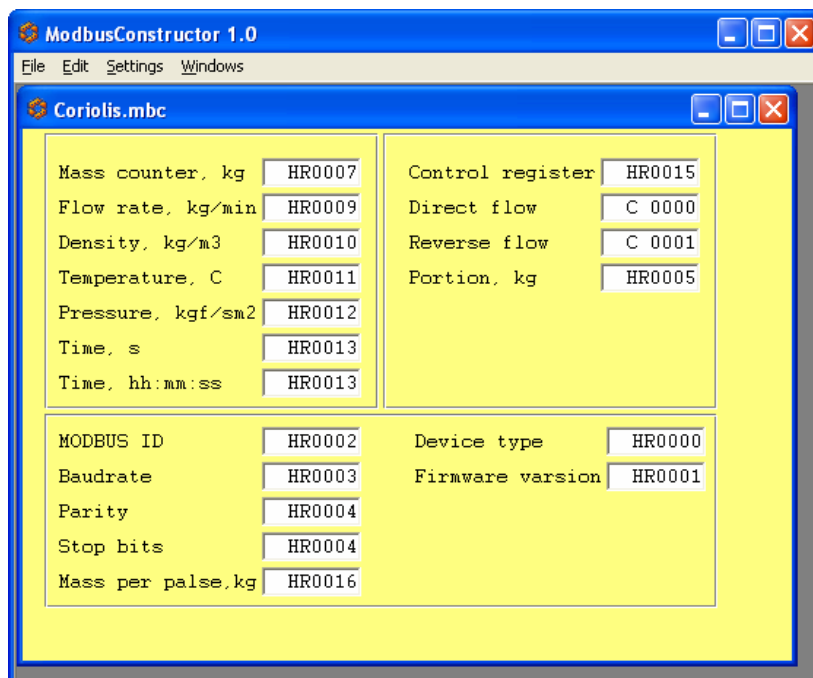


Выделите метку “Mass counter, kg”. Растяните метку влево, так чтобы надпись занимала одну строку. Тоже проделайте с остальными метками.

Для удобства восприятия расположите поля измеряемых параметров отдельно от остальных полей. Начните с левого верхнего угла формы.

Чтобы визуально отделить группы полей, используйте рамки. Кликните правой кнопкой мыши на форме. В открывшемся меню выберите **Add ⇒ Frame**. Изменяя положение и размер рамки, разместите ее по границе группы.

Тоже сделайте с полями, относящимися к управлению и параметрам связи.



Нажмите **Ctrl+S** чтобы сохранить проект.

## 2.4. Добавление запросов.

Для того чтобы удаленное устройство сообщало данные и выполняло необходимые действия, в протоколе Modbus предусмотрены стандартные запросы. Modbus Constructor позволяет определить управление запросами тремя способами:

- автоматически
- через меню
- через диалог

Откройте окно настроек запросов: **Settings⇒Requests**.

В нижней части окна вы увидите таблицу, содержащую все элементы структуры данных. В верхней части окна находится таблица, в которую заносятся параметры запросов.

Сначала создадим автоматические запросы, которые читают содержание регистров и Coils массомера.

дважды кликните на пустую клетку в колонке **Function**

в появившемся меню выберите пункт **03.Read holding registers**. В таблице появится строка с параметрами запроса по умолчанию.

щелкните еще раз по этой строке

в нижней таблице выделите мышкой все ячейки в колонке **Address**, относящиеся к регистрам



кликните правой кнопкой

в появившемся меню нажмите **Apply**

тоже сделайте для Coils. Используйте функцию **01.Read coils**

Label	Function	Read address	Read quantity	Write address	Write quantity	Control	Min.period, ms
	03.Read holding registers	0	17			Auto	300
	01.Read coils	0	2			Auto	300

Label	Read address	Write address	Write	Object	Data	Show	Format	Scale
Device type	0	0	0	Holding register	Integer	Unsigned		
Firmware version	1	1	0	Holding register	Integer	Unsigned		
MODBUS ID	2	2	0	Holding register	Integer	Unsigned		
Baudrate	3	3	0	Holding register	Integer	Unsigned		
Parity	4	4	0	Holding register	MSByte	Unsigned		
Stop bits	4	4	0	Holding register	LSByte	Unsigned		
Portion, kg	5	5	0.0	Holding register	Long	Signed	### ###0.0	0.1x+0
Mass counter, kg	7	7	0.0	Holding register	Long	Signed	### ###0.0	0.1x+0
Flow rate, kg/min	9	9	0.0	Holding register	Integer	Signed	## ###0.0	0.1x+0
Density, kg/m3	10	10	0	Holding register	Integer	Unsigned		0.0305176x+0
Temperature, C	11	11	0	Holding register	Integer	Signed		0.489x+0
Pressure, kgf/sm2	12	12	10.00	Holding register	Integer	Unsigned	## ###0.00	0.049x-10.045
Time, s	13	13	0	Holding register	Long	Unsigned		
Time, hh:mm:ss	13	13	00:16:40	Holding register	Long	Unsigned	hh:mm:ss	
Control register	15	15	0	Holding register	Integer	Unsigned		
Mass per pulse,kg	16	16	0.0	Holding register	Integer	Unsigned	## ###0.0	0.1x+0
Direct flow	0	0	0	Coil	Single	1		
Reverse flow	1	1	0	Coil	Single	1		

Вы можете изменить минимальный период опроса. Для этого дважды кликните на нужную клетку в колонке **Min.period, ms**. Введите значение 300 миллисекунд для обоих запросов.

Обратите внимание, что этот параметр определяет минимально возможный период с которым опрашиваются определенные в запросе данные. Это время может увеличиться за счет времени, требующегося для выполнения других запросов. Если установить значение 0 для этого параметра, запрос будет выполняться с максимально возможной частотой.

Для управления массомером требуется определить команды. Команда осуществляется путем записи определенного значения в контрольный регистр с адресом 15.

создайте запрос **"06.Write single register"**

в нижней таблице в колонке Write address выделите клетку с адресом 15 (контрольный регистр)

кликните правой кнопкой мышки

в появившемся меню нажмите **Apply**

дважды кликните на клетку в колонке **Write**

введите "0" – это значение, которое будет записано в контрольный регистр при выполнении этой команды

в верхней таблице в колонке **Label** введите название команды «Режим измерения». Это название будет служить пунктом пользовательского меню.

кликните дважды по клетке в колонке **Control**

в появившемся меню выберите пункт **Menu 1**

аналогично создайте команды:

«Режим дозатора» (значение 1)

«Отмерить порцию» (значение 2)

«Сброс счетчиков» (значение 4)

«Корректировка нуля» (значение 8)

**Requests.**

Menu caption: Menu 1, Menu 2

Message length: Request length, byte (RTU: 8, ASCII: 17), Response length, byte (RTU: 8, ASCII: 17)

Label	Function	Read address	Read quantity	Write address	Write quantity	Control	Min. period, ms
	03. Read holding registers	0	17			Auto	300
	01. Read coils	0	2			Auto	300
Measure mode	06. Write single register			15	1	Menu 1	1000
Dose-meter mode	06. Write single register			15	1	Menu 1	1000
Measure portion	06. Write single register			15	1	Menu 1	1000
Counters reset	06. Write single register			15	1	Menu 1	1000
Zero adjustment	06. Write single register			15	1	Menu 1	1000

Label	Read address	Write address	Write	Object	Data	Show	Format	Scale
Pressure, kgf/sm2	12	12	10.00	Holding register	Integer	Unsigned	## ##0.00	0.049x-10.045
Time, s	13	13	0	Holding register	Long	Unsigned		
Time, hh:mm:ss	13	13	00:16:40	Holding register	Long	Unsigned	hh:mm:ss	
Control register	15	15	8	Holding register	Integer	Unsigned		
Mass per pulse, kg	16	16	0.0	Holding register	Integer	Unsigned	## ##0.0	0.1x+0
Direct flow	0	0	0	Coil	Single	1		
Reverse flow	1	1	0	Coil	Single	1		

Создайте команды для управления клапанами:

создайте запрос **“15. Write multiple coils”**

в нижней таблице в колонке **Write address** выделите клетки с адресами Coils 0 и 1

кликните правой кнопкой мышки

в появившемся меню нажмите **Apply**

дважды кликните на клетку в колонке **Write** для адреса 0, введите «1»

дважды кликните на клетку в колонке **Write** для адреса 1, введите «0»

при выполнении этой команды будет открываться клапан в прямом направлении, а в обратном будет закрываться

назовите команду «Открыть в прямом направлении»

Создайте команду «Открыть в обратном направлении»:

для Coil с адресом 0 установите значение «0»

для Coil с адресом 1 установите значение «1»

И, наконец, создайте команду «Закреть поток», которая закрывает оба клапана: установите для обеих Coils значение «0»



**Requests.**

Menu caption: Menu 1, Menu 2

Message length: Request length, byte (RTU: 10, ASCII: 21), Response length, byte (RTU: 8, ASCII: 17)

OK Cancel

Label	Function	Read address	Read quantity	Write address	Write quantity	Control	Min.period, ms
	03.Read holding registers	0	17			Auto	300
	01.Read coils	0	2			Auto	300
Measure mode	06.Write single register			15	1	Menu 1	1000
Dose-meter mode	06.Write single register			15	1	Menu 1	1000
Measure portion	06.Write single register			15	1	Menu 1	1000
Counters reset	06.Write single register			15	1	Menu 1	1000
Zero adjustment	06.Write single register			15	1	Menu 1	1000
Forward flow open	15.Write multiple coils			0	2	Menu 1	1000
Reverse flow open	15.Write multiple coils			0	2	Menu 1	1000
Shut flow	15.Write multiple coils			0	2	Menu 1	1000

Label	Read address	Write address	Write	Object	Data	Show	Format	Scale
Time, s	13	13	0	Holding register	Long	Unsigned		
Time, hh:mm:ss	13	13	00:16:40	Holding register	Long	Unsigned	hh:mm:ss	
Control register	15	15	0	Holding register	Integer	Unsigned		
Mass per pulse,kg	16	16	0.0	Holding register	Integer	Unsigned	## ##0.0	0.1x+0
Direct flow	0	0	0	Coil	Single	1		
Reverse flow	1	1	0	Coil	Single	1		

Одна часть команд относится к управлению в режиме дозатор, а другая в режиме измерения, так как клапанами нельзя управлять в режиме дозатора. Есть две команды общие для обоих режимов: «Сброс нуля» и «Коррекция нуля». Разместите их в меню отдельно друг от друга. Для этого:

создайте два запроса, например, **“01.Read coils”**

установите для них параметр контроль **“Menu 1”**

вместо названия введите символ “-” (дефис)

Эти запросы не будут доступны из меню, но послужат разделителями между пунктами.

нажмите левую кнопку на клетке в колонке **Function**.

не отпуская кнопки перетащите серую рамку между командами «Отмерить порцию» и «Сброс счетчиков»

действуя таким же образом, разместите пункты как показано на рисунке

**Requests.**

Menu caption: Menu 1, Menu 2

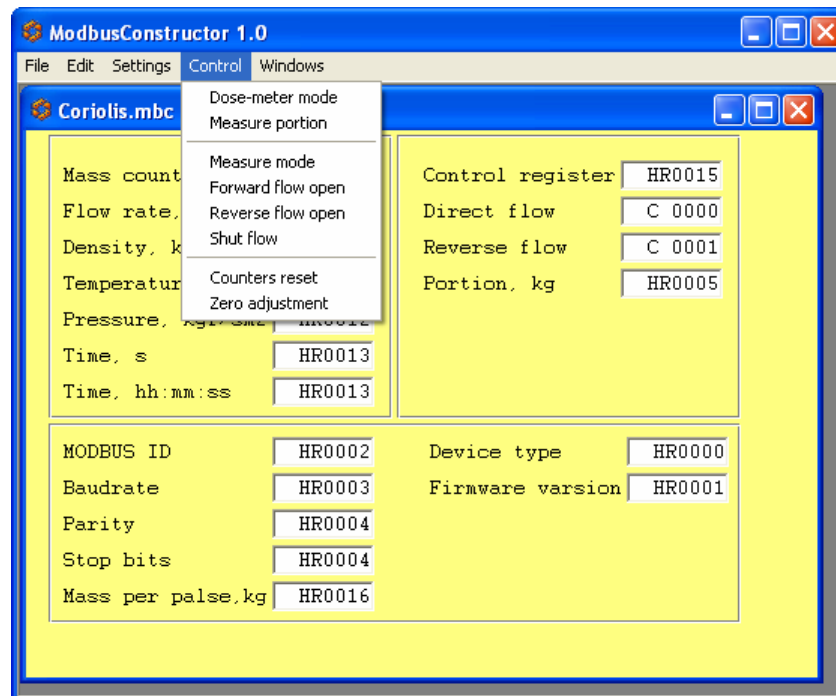
Message length: Request length, byte (RTU: 8, ASCII: 17), Response length, byte (RTU: 6, ASCII: 13)

OK Cancel

Label	Function	Read address	Read quantity	Write address	Write quantity	Control	Min.period, ms
	03.Read holding registers	0	17			Auto	300
	01.Read coils	0	2			Auto	300
Dose-meter mode	06.Write single register			15	1	Menu 1	1000
Measure portion	06.Write single register			15	1	Menu 1	1000
-	01.Read coils	0	1			Menu 1	1000
Measure mode	06.Write single register			15	1	Menu 1	1000
Forward flow open	15.Write multiple coils			0	2	Menu 1	1000
Reverse flow open	15.Write multiple coils			0	2	Menu 1	1000
Shut flow	15.Write multiple coils			0	2	Menu 1	1000
-	01.Read coils	0	1			Menu 1	1000
Counters reset	06.Write single register			15	1	Menu 1	1000
Zero adjustment	06.Write single register			15	1	Menu 1	1000

На вкладке **Menu caption** замените название “Menu 1” на “Control”. Нажмите **OK**. Сохраните проект **Ctrl+S**.

В меню появился новый пункт **Control**:



Точно такое же меню добавится в Modbus Reader когда вы откроете файл этого проекта, и вы сможете использовать его для выполнения созданных вами команд. В Modbus Constructor это меню неактивно.

Теперь определите диалоги для изменения параметров настройки удаленного прибора.

Снова откройте форму **Requests**.

создайте запрос “**16. Write multiple register**”

назовите запрос «Величина порции»

установите адрес «5»

создайте запрос “**16. Write multiple register**”

назовите запрос «Параметры связи»

установите адреса от 2 до 4

создайте запрос “**06. Write single register**”

назовите запрос «Масса на один импульс, кг»

установите адрес «16»

установите параметр контроля для всех трех команд “**Dialog (menu 2)**”

замените название “Menu 1” на “MFM Settings”

нажмите “OK”

сохраните проект: **Ctrl+S**

В меню появился еще один пункт “MFM Settings”. Откройте меню и выберите «Параметры связи». Появится диалог, позволяющий изменять параметры связи для

массомера. При работе с Modbus Reader поля данных для таких диалогов будут заполнены либо текущими значениями, прочитанными непосредственно из устройства с помощью команд чтения, либо значениями по-умолчанию. Значения по-умолчанию используются если соответствующие данные ни разу не были прочитаны до момента открытия диалога, например, если были отключены автоматические запросы.

## 2.5. Параметры симулятора.

Проект, созданный с помощью Modbus Constructor, Modbus Reader может использовать как для опроса устройства в режиме Master, так и для симуляции устройства в режиме Slave. Вы можете менять значения регистров и дискретных данных в режиме Slave вручную, а так же автоматически по заданной функции. Функция задается с помощью следующих параметров: **Default value**, **Value1**, **Value2**, **Period**.

Для дискретных элементов и регистров, отображаемых в виде **Hex** и **Binary**, действует следующее правило:

### 2.5.1. T5. Discrete value simulation.

Время	Значение элемента данных
$0 < t < 1/3 \text{ Period}$	Default value
$1/3 \text{ Period} < t < 2/3 \text{ Period}$	Value1
$2/3 \text{ Period} < t < \text{Period}$	Value2
$\text{Period} < t < 4/3 \text{ Period}$	Default value
$4/3 \text{ Period} < t < 5/3 \text{ Period}$	Value1
...	...

Для регистров содержащих данные других типов **Float**, **Double**, а так же целые, отображаемые в виде **Signed** или **Unsigned**, значения меняются следующим образом:

### 2.5.2. T6. Analogous value simulation.

Время	Значение элемента данных
$0 < t < 1/3 \text{ Period}$	линейно меняется от Default value до Value1
$1/3 \text{ Period} < t < 2/3 \text{ Period}$	линейно меняется от Value1 до Value2
$2/3 \text{ Period} < t < \text{Period}$	линейно меняется от Value2 до Default value
$\text{Period} < t < 4/3 \text{ Period}$	линейно меняется от Default value до Value1
$4/3 \text{ Period} < t < 5/3 \text{ Period}$	линейно меняется от Value1 до Value2
...	...

Если Period равен нулю, используется **Default value**.

Откройте окно **Objects: Settings**⇒ **Objects** и установите параметры симуляции как показано на рисунке.

Обратите внимание, что значение **Value1** для элемента Time равно одной трети от величины периода, в значение **Value2** равно двум третям. В этом случае значение Time будет строго равно времени, прошедшему с момента соединения.

нажмите "OK"

сохраните проект: **Ctrl+S**

**Objects.**

Swapped (inverse)  
☒ Long ☒ Float ☒ Double

OK Cancel

Label	Address	Object	Data	Show	Format	Scale	Default value	Value1	Value2	Period, s
Device type	0	Holding register	Integer	Unsigned			0	0	0	0
Firmware version	1	Holding register	Integer	Unsigned			0	0	0	0
MODBUS ID	2	Holding register	Integer	Unsigned			1	0	0	0
Baudrate	3	Holding register	Integer	Unsigned			9 600	0	0	0
Parity	4	Holding register	MSByte	Unsigned			2	0	0	0
Stop bits	4	Holding register	LSByte	Unsigned			0	0	0	0
Portion, kg	5	Holding register	Long	Signed	# ###0.0	0.1x+0	1 000.0	0.0	0.0	0
Mass counter, kg	7	Holding register	Long	Signed	# ###0.0	0.1x+0	0.0	1 000.0	2 000.0	100
Flow rate, kg/min	9	Holding register	Integer	Signed	## ###0.0	0.1x+0	0.0	- 100.0	100.0	100
Density, kg/m3	10	Holding register	Integer	Unsigned		0305176x+0	900	1 000	800	100
Temperature, C	11	Holding register	Integer	Signed		0.489x+0	20	50	30	100
Pressure, kgf/sm2	12	Holding register	Integer	Unsigned	## ###0.00	049x-10.045	10.00	20.00	15.00	100
Time, hh:mm:ss	13	Holding register	Long	Unsigned	hh:mm:ss		00:16:40	00:00:00	00:00:00	0
Time, s	13	Holding register	Long	Unsigned			0	100 000	200 000	300000
Control register	15	Holding register	Integer	Unsigned			0	0	0	0
Mass per pulse,kg	16	Holding register	Integer	Unsigned	## ###0.0	0.1x+0	0.0	0.0	0.0	0

### 3. РАБОТА С MODBUS УСТРОЙСТВОМ.

В этом разделе вы узнаете как работать с устройством, описанным при помощи Modbus Constructor.

Вместо реального устройства вы будете использовать симулятор. Если на вашем компьютере установлено два COM-порта, соедините их нуль-модемным кабелем. Один порт будет использоваться мастером, другой слейвом. Если на вашем компьютере имеется только один порт, установите Modbus Reader на какой ни будь другой компьютер, имеющий, по крайней мере, один COM-порт, и соедините их нуль-модемным кабелем. Вы можете скачать отдельный дистрибутив для Modbus Reader здесь: <http://www.kurysoft.com/products.shtml>.

Запустите Modbus Reader: **Start** ⇒ **Programs** ⇒ **ModbusConstructor** ⇒ **MBReader.exe**

откройте файл **Coriolos.mbc**.

в меню **Mode** поставьте флажок **Master**

в меню **Connection** выберите первый COM-порт (например COM1)

откройте еще один файл **Coriolos.mbc**.

в меню **Mode** поставьте флажок **Slave**

в меню **Connection** выберите другой COM-порт (например COM2)

расположите обе формы рядом так, чтобы были все поля данных. При необходимости измените размеры

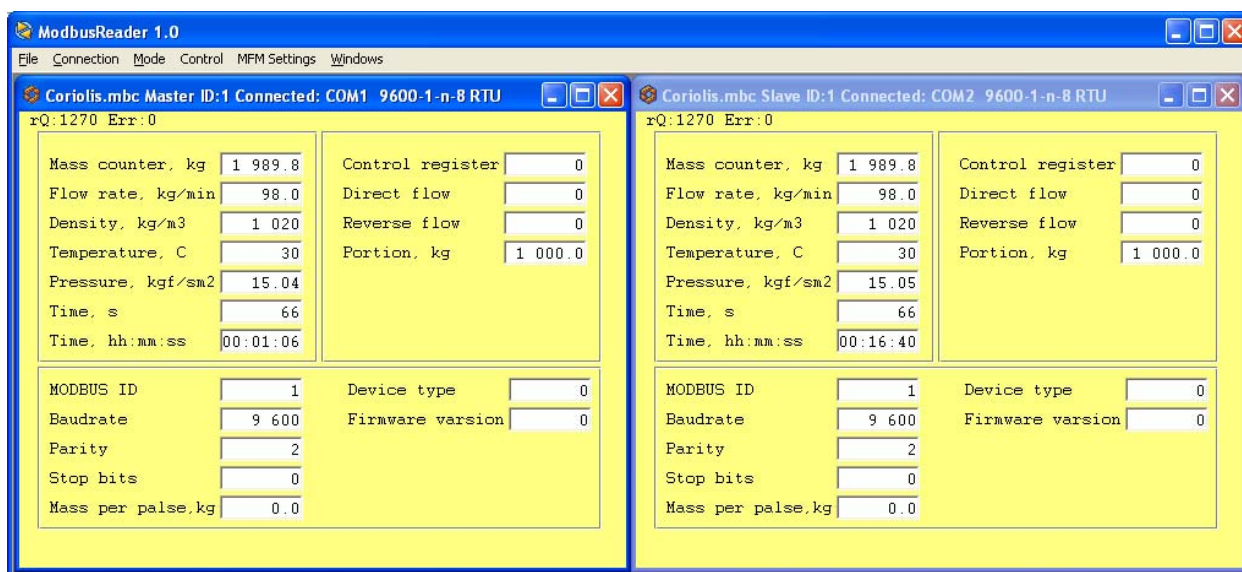
убедитесь, что параметры связи обоих COM-портов совпадают. Для вызова диалога изменения этих параметров нажмите **Ctrl+P**

убедитесь, что Master и Slave имеют одинаковый ID. Для вызова диалога изменения ID нажмите **Ctrl+I**

убедитесь, что в меню **Mode** для формы Master установлен флажок **Auto request enable**, а для формы Slave установлен флажок **Simulation enable**

Обратите внимание, что поля данных на форме Master пусты, а на форме Slave заполнены значениями по-умолчанию.

Нажмите кнопку **F3** чтобы установить соединение. Вы увидите, что данные на форме Master меняются синхронно с данными формы Slave.



Когда активна форма Master, становятся доступными два дополнительных меню **Control** и **MFM Settings**, которые вы создали в разделе «Добавление запросов». Щелкните по форме Master, чтобы сделать ее активной. Воспользуйтесь пунктами этих меню и посмотрите, как меняются данные при выполнении команд.

Вы можете непосредственно менять данные элементов на форме Slave при отключенном режиме симуляции:

- щелкните по форме Slave

- снимите флажок в пункте **Simulation Enable** меню **Mode**

Данные перестанут меняться.

- дважды щелкните мышкой в поле “Density” формы Slave

- введите какое ни будь число

- нажмите **Enter**

В соответствующем поле на форме Master появится такое же число, как только Master выполнит очередной запрос чтения регистров.

Таким же образом можно осуществлять запись отдельных величин в регистры и дискретные элементы устройства.

- дважды щелкните мышкой в поле “Density” формы Master

- введите какое ни будь число

- нажмите **Enter**

По окончании работы не забудьте сохранить текущие настройки, если они могут вам понадобиться. Для этого в меню **File** выберите пункт **Save session**. Сохраните файл под именем **Coriolis.mbs**.

Теперь, когда вы знаете основные приемы работы с Modbus Constructor и Modbus Reader, вы сможете создавать проекты для работы с реальными Modbus-устройствами гораздо быстрее, чем вы смогли выполнить этот пример.



## 4. ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ.

### 4.1. Объекты.

В спецификации Modbus объектом называется элемент данных, который можно прочитать или записать независимо от других данных. Определены четыре типа объектов:

1. Coil	single bit	Read-Write
2. Discrete input	single bit	Read-Only
3. Holding register	16-bits word	Read-Write
4. Input register	16-bits word	Read-Only

### 4.2. Типы данных Modbus Constructor.

Modbus Constructor поддерживает следующие типы данных:

#### 4.2.1. Numeric types

Type	Quantity of bits	Quantity of registers
integer (signed or unsigned)	16	1
long (signed or unsigned)	32	2
float	32	2
double	64	4
MSByte (signed or unsigned)*	8	½ (most significant byte - MSB)
LSByte (signed or unsigned)*	8	½ (least significant byte - LSB)

\*Байт (MSB or LSB) не может быть прочитан или записан независимо от другого байта регистра.

#### 4.2.2. Discrete types

Type	Quantity of bits	
single	1	
multiple	2-16	

### 4.3. Data encoding.

Для представления длинных типов **long**, **float** и **double** используется несколько подряд расположенных регистров.

Используется два способа размещения таких чисел в регистрах: прямой и swapped (inverse).

**4.3.1. Long.**

B3	B2	B1	B0
MSB			LSB

	Register A		Register A+1	
Transmitted	first			last
Default	B1	B0 (LSB)	B3 (MSB)	B2
Swapped	B3 (MSB)	B2	B1	B0 (LSB)

**4.3.2. Float.**

B3	B2	B1	B0
S E E E E E E E	E M M M M M M M	M M M M M M M M	M M M M M M M M

Where floating point value =  $S1.M * 2^{E-127}$

	Register A		Register A+1	
Transmitted	first			last
Default	B1	B0 (LSB)	B3 (MSB)	B2
Swapped	B3 (MSB)	B2	B1	B0 (LSB)

**4.3.3. Double.**

B7	B6	B5-B1	B0
S E E E E E E E	E E E E M M M M M	M M M M M M M M	M M M M M M M M

Where double precision floating point value =  $S1.M * 2^{E-1023}$

	Register A		Register A+1		Register A+2		Register A+3	
Transmitted	first							last
Default	B1	B0	B3	B2	B5	B4	B7	B6
Swapped	B7	B6	B5	B4	B3	B2	B1	B0

**4.4. Правила использования функций Modbus.**

Вы можете создать запрос для записи или чтения какого либо элемента данных устройства только, если этот элемент определен в проекте.

Например, если определен хотя бы один **Holding register**, всегда можно использовать функции 03, 06.



В диалоге **Options** (меню **Settings**) вы можете разрешить или отменить использование дополнительных функции для **Holding register** 16, 22, 23.

Если использование какой либо из этих функций отменено, при работе в режиме симулятора Modbus Reader будет отвечать Exception 1 (Illegal Function), если пришел запрос, использующий такую функцию.

Элемент	Основные функции*	Дополнительные функции
Coil	01 – Read coils	15– Read multiple coils
Discrete input	02 – Read discrete inputs	-
Holding register	03 – Read holding registers 06 – Write single register	16 – Write multiple registers 22 – Write mask register 23 – Read/write registers
Input register	04 – Read input registers	-

\*Должен быть определен, по крайней мере, один элемент.

## 4.5. Format examples.

Для представления чисел в удобном виде используется форматирование.

Форматирование позволяет избежать появления в поле данных чисел со слишком большим количеством цифр после десятичной точки.

### 4.5.1. Numeric types

Число	Формат	Вид
12345	### ##0	12 345
12345	0 000 000	0 012 345
123.45678	# ##0.000	123.457
123.45678	0.00 E+00	1.23 E+02
12	+## 000	+ 012
0.45678	.000 000 000	.456 780 000
12345	hh:mm:ss	03:25:45

Если в поле данных представлено несколько дискретных элементов, использование форматирования позволяет визуально разделить эти элементы.

### 4.5.2. Discrete types

Bits	Формат	Вид
11110000	0000 0000	1111 0000
11110000	0000 0 0 0 0	0000 0 0 0 0
11110000	0	11110000
11110000	0 0	1111000 0
11110000	000 00 000	111 10 000

**4.5.3. Default for numeric types**

Type	Формат
integer	## ##0
long	# ### ##0
byte	##0
float	0.00
double	0.000 E+00

**4.5.4. Default for discrete types**

Type	Формат
single (1)	0
multiple 2	00
multiple 3	000
multiple 4	0000
multiple 5	0 0000
multiple 9	0 0000 0000
multiple 16	0000 0000 0000 0000