

МЕТРА



МЕТРА

СЕТЬ ПРИБОРОВ

Версия 1.7.12.0

**РУКОВОДСТВО
СИСТЕМНОГО ПРОГРАММИСТА**

НППМ 435.001.РСП

Научно-производственное предприятие
“Метра”
г. Обнинск
2010

Аннотация

Данное руководство предназначается для системного программиста, обеспечивающего на предприятии настройку и надежную эксплуатацию поставляемого программного обеспечения.

СОДЕРЖАНИЕ

Аннотация	2
1 Общие сведения о программе	5
1.1 Назначение	5
1.2 Функциональные возможности	5
1.3 Подключение приборов	6
1.4 Защита от нелегального копирования	6
1.5 Технические и программные средства	6
1.5.1 Минимальная конфигурация компьютера для работы программы	6
1.6 Комплектность	6
2 Настройка программы	7
2.1 Настройка приборов и параметров программы	7
2.2 Работа под управлением сервера терминалов	9
2.2.1 Установка и настройка программы под Windows 2003-2008 Server	9
2.2.2 Настройка терминальной сессии	9
2.3 Протоколирование работы драйвера	11
2.4 Программный интерфейс	12
2.4.1 Данные, передаваемые в межзадачный буфер и/или на почтовый ящик "DevNet"	12
2.4.2 Команды, передаваемые в локальной сети на почтовый ящик "DevNetRpc"	17
2.4.3 Функции, реализованные в сервере автоматизации "DevNet.Drv"	23
3 Проверка программы	35
Приложение А. Примеры использования программы	36
А.1 Пример использования сервера автоматизации "DevNet.Drv" на языке программирования Delphi	36
А.2 Пример использования сервера автоматизации "DevNet.Drv" на языке программирования Visual Basic (VBA)	36
А.3 Пример использования сервера автоматизации "DevNet.Drv" на языке программирования "1С: Предприятие"	37
Приложение Б. Значения используемых констант	38

1 Общие сведения о программе

1.1 Назначение

Программа “Сеть приборов” позволяет следить за работой и управлять следующими приборами:

Модель прибора	Мин. версия прибора	Обозначение в тексте
M06A, M06B	3.1	M06A
M0600-A	3.7	M06A
M0601-A, M0601-B	4.3	M0601
M06Д	3.11	M06Д
M0600-Д1	3.71	M06Д
M0600-Д4	3.91	M06Д
M0600-Д6	4.11	M06Д6
M06КС	3.52	M06К
M0600-К1	3.72	M06К
M0600-К4	3.92	M06К
M0600-К6	4.02	M06К
M1600-Н	1.44	M1600
M1600-Н2	1.4.5	M1600
MT601Ц	2.1	M1900
M06АТ	2.2	M1900
M1900, M1901	2.4.2	M1900
M2606	1.0	M2606

Программа также позволяет транслировать показания приборов на цифровые дублирующие табло (M1900). Контроль и управление приборами возможно как непосредственно с компьютера, на котором запущена программа, так и через локальную сеть.

1.2 Функциональные возможности

Функциональные возможности программы:

- управление весами/дозаторами в режиме дистанционной клавиатуры;
- выдача управляющих сигналов через последовательный порт компьютера;
- имитация и визуальный контроль процесса дозирования и взвешивания;
- изменение значений уставок в дозаторах, выдача команд “ВКЛ”, “ВЫКЛ”, “СТАРТ”, “СТОП”, “ТАРА”;
- передача измеренных значений веса на цифровое табло, в буфер обмена и через локальную сеть;
- встраивание в АСУ заказчика в качестве драйвера с помощью технологий **MailSlot** (почтовый ящик), **FileMapping** (межзадачный буфер обмена) и **OLE Automation** (сервер OLE автоматизации).

1.3 Подключение приборов

Один весовой (M06A, M0601) прибор, дозирующий (M06Д, M06Д6, M06К) прибор или контроллер (M1600, M2606) подключается к последовательному порту компьютера через интерфейсный конвертор RS232/RS485; к другому последовательному порту можно подключить цифровое дублирующее табло (M1900) и/или внешние управляющие (например, процессом дозирования) устройства заказчика. При необходимости работы с несколькими приборами они соединяются в сеть RS-485 непосредственно или через интерфейсные конверторы RS232/RS485 INCON-S (только для приборов, не имеющих интерфейса RS485) и подключаются к последовательному порту компьютера через интерфейсный конвертор INCON-M, INCON-MT, M2107. Сеть RS-485 может быть выполнена 4-х проводным экранированным кабелем типа USB, FTP (с заземлением в одной точке около компьютера) или UTP. Полученная таким образом сеть RS-485 является надежным техническим решением для обеспечения связи приборов с компьютером в промышленных условиях. Схема подключения приборов и топология сети RS485 описана в документации приборов.

1.4 Защита от нелегального копирования

Программа защищена от нелегального копирования электронным ключом, который подключается к параллельному порту компьютера и является неотъемлемой частью программы. В ключе записано кол-во приборов (N), с которыми программа может работать. Утрата ключа равнозначна утрате самой программы. Если электронный ключ не подключен или по каким-либо причинам не обнаружен программой, то она переходит в демонстрационный режим работы. В демонстрационном режиме программа работает только с одним прибором (N=1); уникальный номер подключенного к порту прибора может быть любым; в диалоге **Список приборов** необходимо отметить прибор №1.

1.5 Технические и программные средства

Программа разработана для эксплуатации в операционной системе Microsoft Windows 95 OSR 2 / 98 SE / ME / NT 4.0 / 2000 / XP.

1.5.1 Минимальная конфигурация компьютера для работы программы

- процессор Pentium-100;
- оперативная память 32Mb;
- свободное дисковое пространство 10Mb;
- дисковод 1.44Mb или CD-ROM;
- два COM порта (без режима "локального эха");
- SVGA видео-плата с разрешением 800x600, 256 цветов;
- 15" SVGA монитор;
- клавиатура; мышь.

1.6 Комплектность

Диск с программой «Сеть приборов»	1 шт.
Руководство системного программиста НППМ.435.001.РСП	1 экз.
Электронный ключ (не входит в бесплатную версию программы DevCom, распространяемую в составе приборов)	1 шт.

2 Настройка программы

2.1 Настройка приборов и параметров программы

Перед подключением приборов к компьютеру в них необходимо установить следующие программные параметры:

- для приборов M06A, M0601, M06Д, M06Д6, M06К:

Pu6. = 6 (работа по протоколу RS-485),

Pu7. = N (N - уникальный номер прибора в сети RS-485, от 1 до 31),

PE0. = 0 (скорость обмена 9600 бод) (для прибора M0601 версии 4.7 и выше установить PE0. = 3).

- для приборов M1600:

Pu4. = 6 (работа по протоколу RS-485),

Pu5. = N (N - уникальный номер прибора в сети RS-485, от 1 до 31),

PE0. = 0 (скорость обмена 9600 бод).

- для приборов M1900:

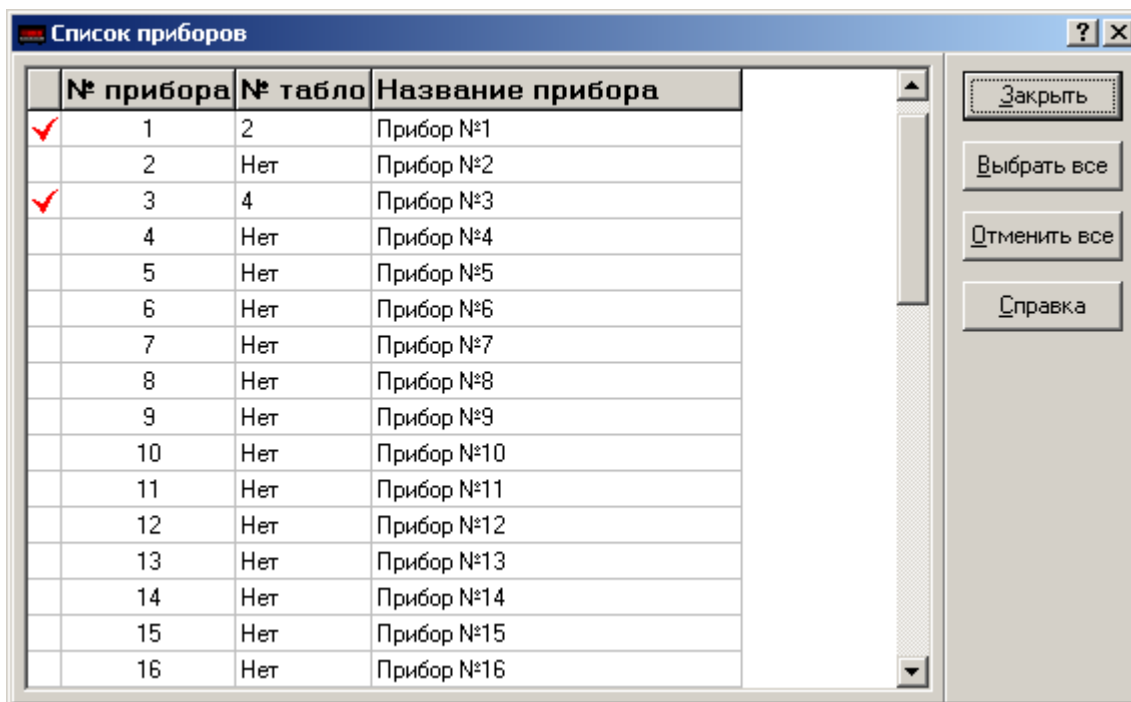
Pu0. = N (N - уникальный номер прибора в сети RS-485, от 1 до 31),

Pu1. = 1 (работа по протоколу RS-485),

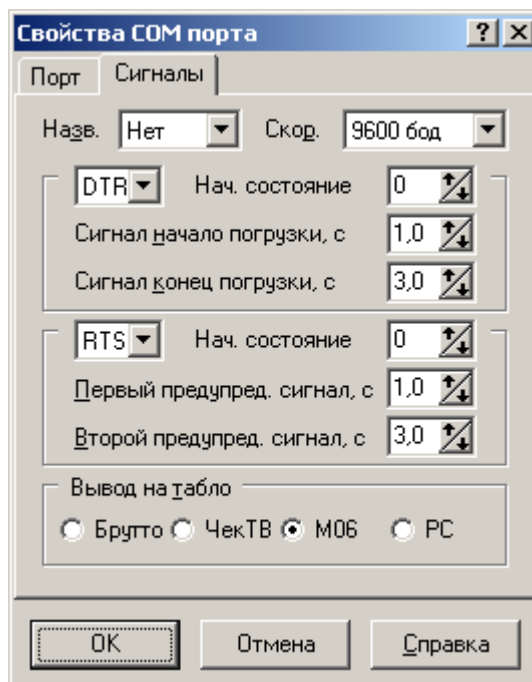
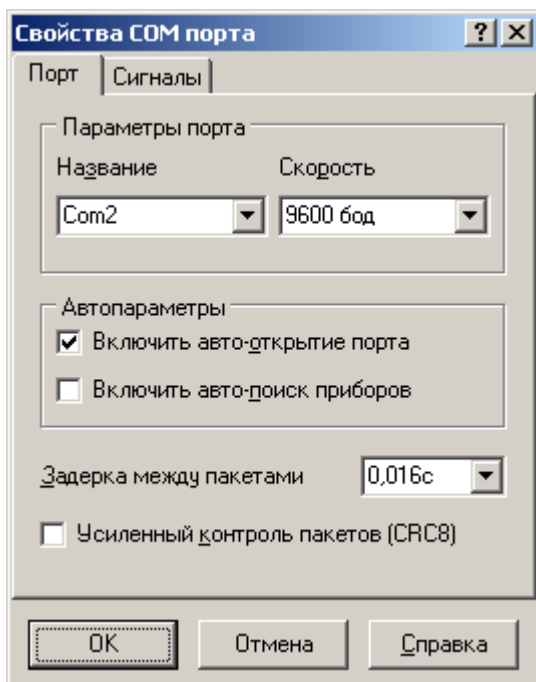
Pu2. = 3 (скорость обмена 9600 бод).

Используемый для связи последовательный порт компьютера не должен работать в режиме “локального эха”. Все подключаемые к порту приборы должны иметь одинаковую скорость обмена и уникальные номера от 1 до N, где N – кол-во приборов, с которыми программа может работать.

В программе в диалоге **Список приборов** необходимо отметить номера подключенных к сети RS-485 приборов (M06A, M0601, M06Д, M06Д6, M06K, M1600) и выбрать номера дублирующих табло (M1900), подсоединяемых к приборам. Если дублирующие табло подключаются к дополнительному COM порту, то их номера должны совпадать с номерами приборов, к которым они подсоединяются. При этом допускается подсоединение нескольких табло с одинаковым номером к одному прибору.



В диалоге **Свойства COM порта** необходимо задать название порта, к которому подключены приборы, и соответствующую скорость обмена.



По умолчанию программа настроена на подключение одного прибора с сетевым номером 1 к последовательному порту COM1 на скорости 9600 бод.

2.2 Работа под управлением сервера терминалов.

2.2.1 Установка и настройка программы под Windows 2003-2008 Server.

Установку программы рекомендуется выполнять локально, т.е. непосредственно на сервере, т.к. при удаленной установке может не хватить прав, необходимых для регистрации программы как COM-сервера. После установки:

- 1) Установить электронный ключ (если имеется).
- 2) После установки ключа может появиться сообщение, что найдено новое оборудование и необходимо установить драйвера. Выполните установку требуемых драйверов.
- 3) Пользователю, который будет работать с программой, необходимо добавить **разрешение на запись** в папку, где установлена программа.

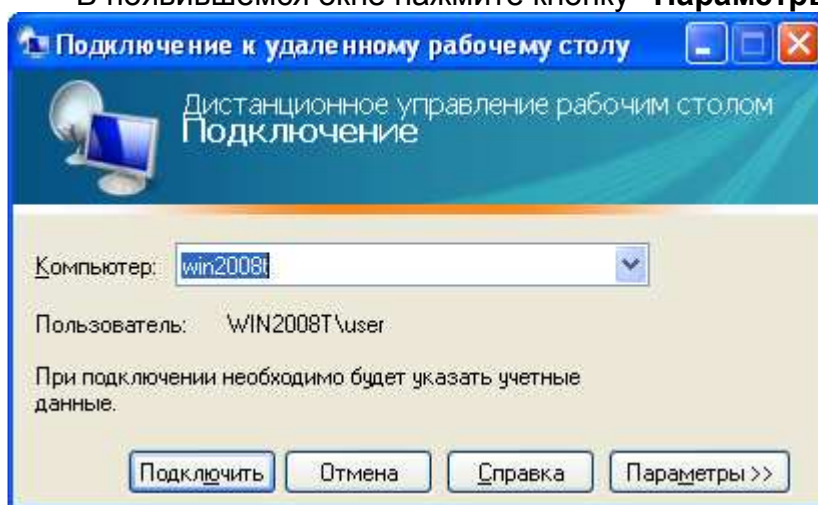
2.2.2 Настройка терминальной сессии

При работе программы под управлением сервера терминалов (например Windows 2003 Server, Windows 2008 Server и т.п.) возможны два варианта подключения приборов:

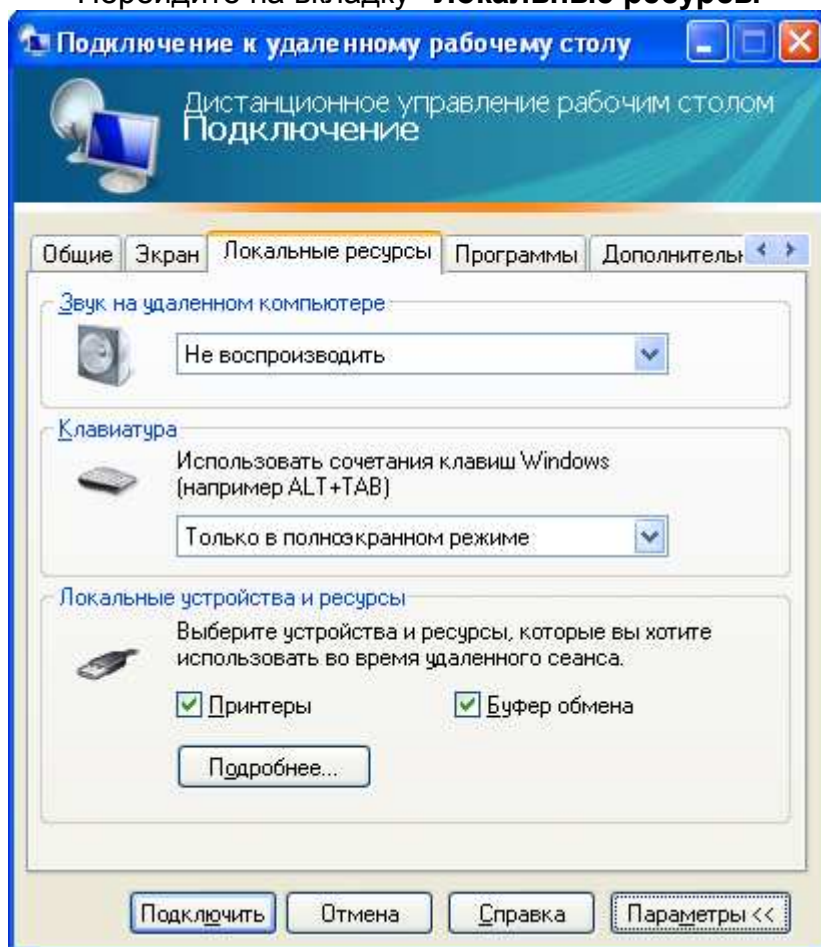
- 1) Приборы подключаются к локальному COM-порту самого сервера.
- 2) Приборы подключаются к COM-порту терминального клиента.

В зависимости от варианта подключения приборов при запуске терминальной сессии необходимо соответствующим образом настроить параметры подключения. Например, если в качестве терминального клиента используется программа "Удаленный рабочий стол" (Remote Desktop), входящая в состав операционных систем Windows, то настройка подключения выполняется так:

- 1) Из меню кнопки **«Пуск»** выберите **«Все программы\Стандартные\Подключение к удаленному рабочему столу»**. Будет запущен терминальный клиент.
- 2) В появившемся окне нажмите кнопку **"Параметры"**



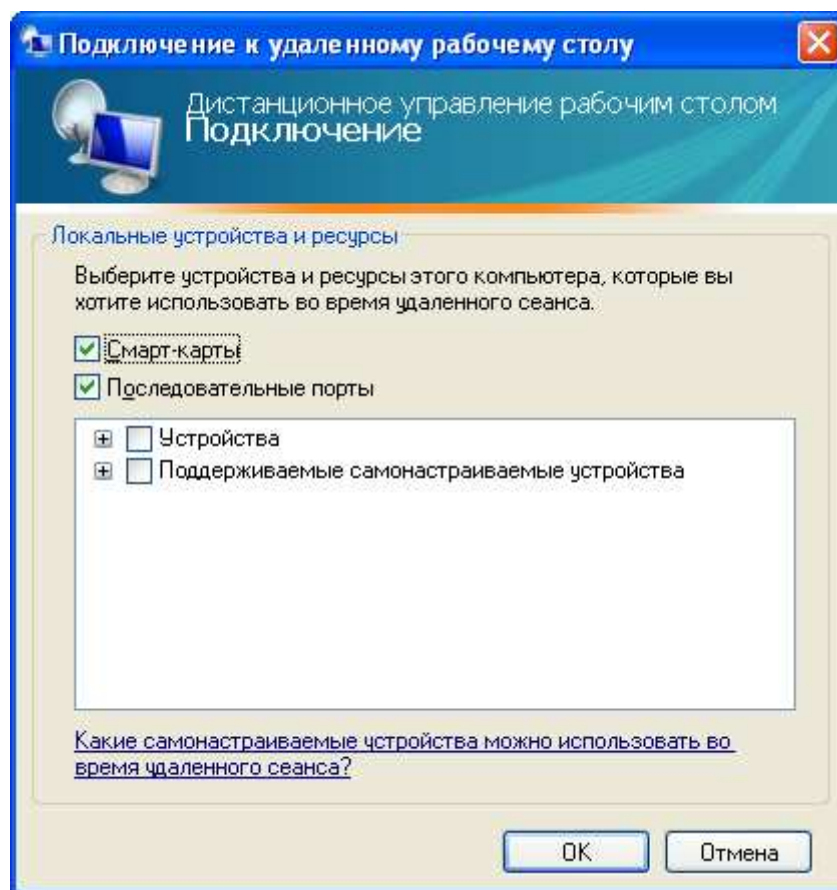
3) Перейдите на вкладку "Локальные ресурсы"



4) В разделе "Локальные устройства и ресурсы" нажмите кнопку "Подробнее".

5) В открывшемся окне:

- выключите флаг "Последовательные порты", если приборы подключены к локальному COM-порту самого сервера (**вариант подключения 1**).
- включите флаг "Последовательные порты", если приборы подключены к COM-порту терминального клиента (**вариант подключения 2**).



2.3 Протоколирование работы драйвера

Программа драйвера может протолировать свою работу. Получаемые протоколы могут помочь разработчикам выяснить причину некорректной работы программы и/или прибора. По умолчанию протоколирование выключено. Для того чтобы включить протоколирование надо выбрать пункт меню «Настройка»|«Параметры». В открывшемся окне диалога перейти на страницу «Протоколы» и отметить:

- флаг «**Протолировать текущие показания прибора**», если необходимо получить историю показаний прибора за все время работы программы, пока включен этот флаг.
- флаг «**Вести протокол обмена с прибором**», если необходимо получить протокол обмена программы и прибора по COM порту.

В поле «**Максимальный размер протоколов в процентах от свободного места на диске**» необходимо задать максимальный размер протокола. Например, если на данный момент размер свободного места на диске, где установлена программа, равен 20 Гб, а размер протокола задан равным 10%, то суммарный объем протоколов не может быть больше $20/100 \cdot 10 = 2$ Гб. Как только размер протоколов превысит это значение, программа автоматически удалит наиболее старые протоколы.

Если необходимо переслать накопленные за определенный промежуток времени протоколы в НПП «Метра», то для формирования файла с архивом протоколов можно воспользоваться программой «**Подготовка протоколов для НПП 'Метра'**» (LogPrepare.exe), поставляемой совместно с драйвером «DevNet».

2.4 Программный интерфейс

2.4.1 Данные, передаваемые в межзадачный буфер и/или на почтовый ящик “DevNet”

Значения используемых констант указаны в Приложении Б.

Структура данных *TPlatformRec*:

```
TDeviceRec = record
    NumDev      : Byte;      // номер прибора
    TypeDev     : Byte;      // тип прибора (см. ниже)
    VersionDev  : Byte;      // версия прибора (см. ниже)
    Variable1   : Double;    // вес брутто платформы (M06A, M06D, M06D6),
                               // счетчик-интегратор (M06K),
                               // измеренная производительность (M1600)
    Variable2   : Double;    // вес «пустой» платформы,
                               // «рабочий» ноль (M06A, M06D, M06D6),
                               // мгновенная производительность или
                               // мгновенная лин. плотность (M06K),
                               // заданная производительность (M1600)
    PntPos      : Byte;      // число знаков после запятой
    ErrState    : Byte;      // код ошибки (см. ниже)
    Flags0      : Byte;      // бит-флаги состояния платформы (см. ниже)
    Flags1      : Byte;      // бит-флаги состояния прибора (см. ниже)
    DFlags      : Byte;      // бит-флаги состояния дозатора (см. ниже)
    DState      : Byte;      // состояние дозатора (см. ниже)
end;
```

Поле *TypeDev*:

TypeDev_M06A	– M06A, M0601
TypeDev_M06D	– M06D
TypeDev_M06D6	– M06D6
TypeDev_M06K	– M06K
TypeDev_M1600	– M1600
TypeDev_M1900	– M1900
TypeDev_M2606	– M2606
TypeDev_Unknown	– Значение не определено

Поле *VersionDev*:

VersionDev_31X	– 3.1X
VersionDev_32X	– 3.2X
VersionDev_34X	– 3.4X
VersionDev_35X	– 3.5X
VersionDev_37X	– 3.7X
VersionDev_41X	– 4.1X
VersionDev_43X	– 4.3X
VersionDev_49X	– 4.9X
VersionDev_492	– 4.92
VersionDev_50X	– 5.0X
VersionDev_14X	– 1.4X
VersionDev_2X	– 2X
VersionDev_Unknown	– Значение не определено

Поле **ErrState** (для приборов M06A, M0601):

Значение	Комментарий
10	Ошибка записи в EEPROM
20	Вес меньше нижнего предела, заданного параметром "Pc1"
21	Вес больше НПВ + 9*D
22	Ошибка подключения датчика (входной сигнал меньше нижней границы рабочего диапазона АЦП)
23	Ошибка подключения датчика (входной сигнал больше верхней границы рабочего диапазона АЦП)
24	АЦП не функционирует
25	Питание датчика отключено из-за перегрузки по току (возможно короткое замыкание в линии питания датчика)
31	Вычислительная ошибка. Значение брутто/нетто < -31768 Возможно, из-за некорректной калибровки.
32	Вычислительная ошибка. Значение брутто/нетто > 31767 Возможно, из-за некорректной калибровки.
40	Попытка взять за ноль вес < -1% от M
41	Попытка взять за ноль вес > +3% от M
42	Попытка выполнить тарирование или "заморозку" при нестабильных весах
45	Попытка вторично просуммировать уже учтенные замороженные показания
46	Попытка "заморозить" вес < 20 дискрет
47	Попытка "заморозить" без разгрузки после предыдущей заморозки. Только при установленном параметре "Pu.8 1"
77	Часы реального времени не установлены (отсутствуют)
78	Часы реального времени не функционируют
100	Нет опроса со стороны компьютера (при установленном параметре "u.St ", не равном 0)
101	Нет чтения и очистки сообщения со стороны компьютера (при установленном параметре "u.Pt ", не равном 0)

Поле **ErrState** (для приборов M06Д, M06Д6, M06К):

Значение	Комментарий
10	Ошибка записи в EEPROM
20	Вес меньше нижнего предела, заданного параметром "Pc.1"
21	Вес больше НПВ + 9*D
22	Ошибка подключения датчика (входной сигнал меньше нижней или больше верхней границы рабочего диапазона АЦП)
30	Вычислительная ошибка. Значение брутто/нетто < -31768 Возможно, из-за некорректной калибровки.
31	Вычислительная ошибка. Значение брутто/нетто > 31767 Возможно, из-за некорректной калибровки.
40	Попытка взять за ноль вес < -1% от М
41	Попытка взять за ноль вес > +3% от М
42	Попытка выполнить тарирование при нестабильных весах

Поле **ErrState** (для приборов M1600):

Значение	Комментарий
110	Ошибка записи в EEPROM
120	Сигнал с аналогового входа меньше нижнего предела
121	Сигнал с аналогового входа больше верхнего предела
122	Отключение (обрыв) задания с аналогового входа
125	Отключение (обрыв) связи с M06К
170	Отключение (обрыв) питания
190	Отключение (обрыв) связи с управляющим компьютером

Поле **Flags0** (для приборов M06A, M0601, M06Д, M06Д6, M06К):

Бит	Название	Комментарий
0	fClb	прибор в режиме “калибровка”
1	fClbZero	прибор в режиме “калибровка нуля”
2	fStable	значения веса стабильны
3	fNearZero	значения веса в диапазоне $-0.5d..+0.5d$
4	fIsMin20d	значение веса меньше 20d
5	fUnderLoad	значение веса существенно меньше нуля
6	fOverLoad	значение веса больше НПВ+9d
7	fNoLoadCell	ошибка подключения (обрыв, повреждение) тензодатчика

Поле **Flags0 = 0** (для приборов M1600)

Поле **Flags1** (для приборов M06A, M0601):

Бит	Название	Комментарий
0	fHandTare	значение веса тары прибора задано в ручную
1	fHoldSignal	показания прибора “заморожены”
2	fHoldPrinted	“замороженные” показания прибора распечатаны на принтере
3	fStableHold	“замороженные” показания прибора стабильны
4	fAutoHoldUse	используется “авто-заморозка” показаний прибора
5	fWaitUnload	ожидается разгрузка весовой платформы
6	fHoldCounted	“замороженные” показания прибора добавлены к счетчику
7	fLock485	интерфейс RS485 заблокирован

Поле **Flags1** (для приборов M06Д, M06Д6):

Бит	Название	Комментарий
0	fBadDosator Data	прибору задан неверный набор уставок
1	fDMode	прибор в режиме “дозатор”
2	fViewFull Portion	прибор в режиме просмотра полной дозы в режиме “порционного дозирования”
3	fLastPortion	прибор дозирует последнюю дозу в режиме “порционного дозирования”
7	fLock485	интерфейс RS485 заблокирован

Поле **Flags1** (для приборов M06К):

Бит	Название	Комментарий
0	fBadDosator Data	прибору задан неверный набор уставок
1	fDMode	прибор в режиме “дозатор”
7	fLock485	интерфейс RS485 заблокирован

Поле **Flags1** (для приборов M1600):

Бит	Название	Комментарий
0	fIndNormal	прибор в режиме просмотра измеренной производительности

Поле **DFlags** = 0 (для приборов M06A, M0601, M1600)

Поле **DFlags** (для приборов M06Д, M06К):

Бит	Название	Комментарий
DO.0	fMeandr	МЕАНДР
DO.1	fProcess	ЦИКЛ
DO.2	fStable	СТАБИЛЬНО
DO.3	fLimit0	ВЕС > L.00
DO.4	fLimit1	ВЕС > L.01
DO.5	fLimit2	ВЕС > L.02
DO.6	fAnswer	ОТВЕТ
DO.7	fError	ОШИБКА

Поле **DFlags** (для приборов M06Д6):

Бит	Название	Комментарий		
DO.0	fMeandr	"L.05 OF"	"L.08 OF"	МЕАНДР
			"L.08 Su"	WrkTotal > L8_dOSA
		"L.05 br/nE"	"L.08 OF"	ВЕС > L.05
			"L.08 Su"	WrkTotal > L8_dOSA
DO.1	fProcess	ЦИКЛ		
DO.2	fStable	"L.06 OF"	"L.08 OF"	СТАБИЛЬНО
			"L.08 Su"	WrkTotal > (L8_dOSA – L8_PrE)
		"L.06 br/nE"	"L.08 OF"	ВЕС > L.06
			"L.08 Su"	WrkTotal > (L8_dOSA – L8_PrE)
DO.3	fLimit0	ВЕС > L.00		
DO.4	fLimit1	ВЕС > L.01		
DO.5	fLimit2	ВЕС > L.02		
DO.6	fAnswer	"L.03 OF"	ОТВЕТ	
		"L.03 br/nE"	ВЕС > L.03	
DO.7	fError	"L.04 OF"	"L.07 OF"	ОШИБКА
			"L.07 rA"	Импульсный выход UNLOAD
		"L.04 br/nE"	"L.07 OF"	ВЕС > L.04
			"L.07 rA"	Импульсный выход UNLOAD

Поле **DState** (для приборов M06A, M0601):

0 – режим отображения брутто

1 – режим отображения нетто

2 – режим отображения тара

3 – режим выключения прибора (только для M0601 версии 4.90 и выше)

4 – режим удаленного управления (только для M0601 версии 5.00 и выше)

Поле **DState** (для приборов M06Д, M06Д6, M06К):

0 – инициализация

1 – ожидание команды "Старт"

2 – команда "Старт" отработана

3 – значение "Уставки-1" достигнуто

4 – ожидание команды "Стоп"

Поле **DState** (для приборов M1600):

0 – режим управления с аналогового входа 4-20мА

1 – режим управления с кнопок прибора

2 – режим управления от компьютера

2.4.2 Команды, передаваемые в локальной сети на почтовый ящик “DevNetRpc”

Для управления программой, как драйвером по технологии **MailSlot** используются два почтовых ящика (порта). На порт “**DevNetRpc**” подаются управляющие работой программы команды (ASCII строки), начинающиеся с символа @. Если команда распознана, то на порт “**DevNetDrv**” выдается символ #, распознанная часть команды и результат ее выполнения.

Команда на порт “**DevNetRpc**”:

@<Тело команды>

Ответ на порт “**DevNetDrv**”:

#<Распознанная часть команды><Результат выполнения>

Символ **E** в <результате выполнения> команды означает невозможность ее выполнения по каким-либо причинам. Символы **?** в <результате выполнения> команды означают синтаксическую ошибку в теле команды. Для некоторых команд <результат выполнения> может отсутствовать.

- @PT – Проверить состояние последовательного порта.
{1 (порт открыт) | 0 (порт закрыт)};<название порта>;
<скорость обмена>.
- @PO[X][Y][A] – Открыть последовательный порт, где
X – номер порта (Com_1, ..., Com_8),
Y – скорость обмена (BaudRate_1200, ..., BaudRate_57600),
A – включить режим авто-поиска подключенных к сети RS-485 приборов.
- @PC – Закрыть последовательный порт.
- @SV {
0 | – Скрыть окно DevNet.
1 | – Показать окно DevNet.
}
- @DL – Получить список подключенных приборов к сети RS-485 в формате: <Адрес1><Тип1><Версия1>;<Адрес2>... – см. выше [TypeDev](#), [VersionDev](#).
- @DXX{
+[YY] | – добавить прибор XX в список подключенных приборов и назначить ему табло YY.
– | – удалить прибор XX из списка подключенных приборов вместе с назначенным ему табло.
G {
I | – Получить тип [TypeDev](#) и версию [VersionDev](#) прибора XX – см. выше.
S | – Получить бит-флаги [ErrState](#), [Flags0](#), [Flags1](#), [DFlags](#), [DState](#) состояния прибора XX – см. выше. (кроме приборов M1900).
V { – Получить из прибора XX значение переменной
Для приборов M06A, M0601:
M06A_Brutto | – вес брутто,
M06A_Netto | – вес нетто,
M06A_Tare | – вес тара,
M06A_Zero | – “рабочий” ноль,
M06A_ErrState | – **ErrState** (см. [выше](#)),
M06A_Flags0 | – **Flags0** (см. [выше](#)),
M06A_Flags1 | – **Flags1** (см. [выше](#)),
M06A_DState | – **DState** (см. [выше](#)),
M06A_SumTotal | – сумма нетто,
M06A_SumCounter | – счетчик,

M06A_SBrutto	– исходный вес брутто,
M06A_ADC	– код АЦП.
<i>Для приборов M06Д, M06Д6:</i>	
M06D_Brutto	– вес брутто,
M06D_Netto	– вес нетто,
M06D_Tare	– вес тара,
M06D_Zero	– “рабочий” ноль,
M06D_ErrState	– ErrState (см. выше),
M06D_Flags0	– Flags0 (см. выше),
M06D_Flags1	– Flags1 (см. выше),
M06D_DFlags	– DFlags (см. выше),
M06D_DState	– DState (см. выше),
M06D_Limits	– Limits (см. ниже Таблицы 2, 3),
M06D_SumTotal	– счетчик загрузки/выгрузки (Pro 0), общая набранная доза (Pro 1),
M06D_SumCounter	– номер дозы (Pro 0), общий счетчик порций (Pro 1),
M06D_WrkTotal	– текущая набранная доза (Pro 1),
M06D_WrkCounter	– текущий счетчик порций (Pro 1),
M06D_ADC	– код АЦП.
<i>Для приборов M06К:</i>	
M06K_Brutto	– вес брутто,
M06K_Netto	– вес нетто,
M06K_Tare	– вес тара,
M06K_Zero	– “рабочий” ноль,
M06K_ErrState	– ErrState (см. выше),
M06K_Flags0	– Flags0 (см. выше),
M06K_Flags1	– Flags1 (см. выше),
M06K_DFlags	– DFlags (см. выше),
M06K_DState	– DState (см. выше),
M06K_Limits	– Limits (см. ниже Таблица 4),
M06K_Integral	– счетчик-интегратор,
M06K_xRate	– мгн. производительность,
M06K_IntTime	– время интегрирования,
M06K_xSpeed	– мгн. скорость,
M06K_xDensity	– мгн. лин. плотность,
M06K_LIntegral	– длина ленты,
M06K_ADC	– код АЦП.
<i>Для приборов M1600:</i>	
M1600_FlowRateMeasured	– измеренная производительность,
M1600_FlowRateCurrent	– заданная производительность,
M1600_ErrState	– ErrState (см. выше),
M1600_Flags1	– Flags1 (см. выше),
M1600_DState	– DState (см. выше),
M1600_Limits	– Limits (см. ниже Таблица 5),
M1600_PLC_Labels	– внутренняя переменная,
M1600_PLC_LabelsExt	– внутренняя переменная,
M1600_PLC_Ext_Outputs	– дискретные выходы M1600 на исполнительные механизмы,
M1600_PLC_Ext_Inputs	– дискретные входы M1600 с элементов управления,
M1600_PLC_KC_Outputs	– из M1600 в M06К по шине MS-Bus,
M1600_PLC_KC_Inputs	– из M06К в M1600 по шине MS-Bus.

- Для приборов M2606:*
- M2606_Inputs | – состояние входов,
M2606_PLCNum | – счетчик изменений состояния PLC модуля.
- } | (значение выдается в формате ASCII строки)
- C** { – Получить из прибора XX значение константы
- Для приборов M06A, M0601:*
- M06A_TypeDev | – **TypeDev** (см. [выше](#)),
M06A_VersionDev | – **VersionDev** (см. [выше](#)),
M06A_MaxRange | – максимальное количество весовых диапазонов - 1,
M06A_PntPos | – число знаков после запятой,
M06A_MaxBrutto | – массив НПВ, в формате:
<НПВ диапазона1>;<НПВ диапазона2>;...,
M06A_Discret | – массив дискрет, в формате:
<Дискрета диапазона1>; <Дискрета диапазона2>;...,
M06A_EEPROM_BlockCounter | – максимальное количество блоков EEPROM,
M06A_EEPROM_ValidMask | – бит-флаг контроля чтения CRC блоков EEPROM,
M06A_EEPROM_CRC – CRC блоков EEPROM, в формате: <CRC блока1>;<CRC блока2>;...
- Для приборов M06D, M06D6:*
- M06D_TypeDev | – **TypeDev** (см. [выше](#)),
M06D_VersionDev | – **VersionDev** (см. [выше](#)),
M06D_PntPos | – число знаков после запятой,
M06D_MaxBrutto | – НПВ,
M06D_Discret – дискрета.
- Для приборов M06K:*
- M06K_TypeDev | – **TypeDev** (см. [выше](#)),
M06K_VersionDev | – **VersionDev** (см. [выше](#)),
M06K_PntPos | – число знаков после запятой,
M06K_MaxBrutto | – НПВ,
M06K_Discret | – дискрета,
M06K_xRatePos | – число знаков после запятой для мн. производительности,
M06K_xDensityPos | – число знаков после запятой для мн. лин. плотности,
M06K_xSpeedPos | – число знаков после запятой для мн. скорости,
M06K_IntegralPos | – число знаков после запятой для счетчика-интегратора,
M06K_LIntegralPos | – число знаков после запятой для длины ленты,
M06K_MaxRate | – наибольшая производительность,
M06K_MaxDensity – наибольшая лин. плотность.
- Для приборов M1600:*
- M1600_TypeDev | – **TypeDev** (см. [выше](#)),
M1600_VersionDev | – **VersionDev** (см. [выше](#)),
M1600_PntPos | – число знаков после запятой,
M1600_MaxFlowRate | – наибольшая заданная производительность,
M1600_Discret | – дискрета,
M1600_EEPROM_BlockCounter | – максимальное количество блоков EEPROM,
M1600_EEPROM_ValidMask | – бит-флаг контроля чтения CRC блоков EEPROM,

	M1600_EEPROM_CRC	– CRC блоков EEPROM, в формате: <CRC блока1>;<CRC блока2>;...
	<i>Для приборов M2606:</i>	
	M2606_TypeDev	– TypeDev (см. выше),
	M2606_VersionDev	– VersionDev (см. выше),
	M2606_Params	– значения параметров PLC модуля.
	} (значение выдается в формате ASCII строки)	
L	– Получить значения уставок Limits (см. ниже Таблицы 2..5) из прибора XX (только для приборов M06Д, M06Д6, M06К, M1600).	
K	– Получить код нажатых на приборе XX кнопок (только для приборов M1900, M0601 версии 5.00 и выше).	
D	– Получить из прибора XX значение маски опрашиваемых параметров данных (см. ниже). (кроме приборов M1900).	
N	– Получить из прибора XX значение маски опрашиваемых параметров переменных (см. ниже). (кроме приборов M1900).	
	}	
S	{	
	Z	– Нажать на приборе XX кнопку Ноль . (кроме приборов M1900).
	z	– Отменить нажатие на приборе XX кнопки Ноль . (кроме приборов M1900).
	T	– Нажать на приборе XX кнопку Тара . (кроме приборов M1900).
	t	– Отменить нажатие на приборе XX кнопки Тара . (кроме приборов M1900).
	W	– Нажать на приборе XX кнопку Вес . (кроме приборов M1900).
	X	{ – Нажать на приборе XX кнопку
	0	– кнопку Выкл. ,
	1	– кнопку Вкл.
	} (только для приборов M06Д, M06Д6, M06К).	
	S	{ – Нажать на приборе XX кнопку
	0	– кнопку Стоп ,
	1	– кнопку Старт .
	} (только для приборов M06Д, M06Д6, M06К, M1600).	
L	{	– Записать значения уставок Limits (см. ниже Таблицы 2..5) в прибор XX.
	Limits	
	} (только для приборов M06Д, M06Д6, M06К, M1600).	
M	– Нажать на приборе XX кнопку Режим . (кроме приборов M1900).	
F	{	– Установить на приборе XX новое значение заданной производительности FlowRate.
	FlowRate	
	(значение задается в формате ASCII строки).	
	} (только для приборов M1600).	
I	{	– Установить сообщение LedBuf для вывода на индикатор прибора XX.
	LedBuf	
	(значение задается в формате ASCII строки).	
	} (только для приборов M1900, M0601 версии 5.00 и выше).	
D	{	– Установить на приборе XX новое значение

маски опрашиваемых параметров данных
(см. [ниже](#)).

DataParam
 } | (кроме приборов M1900).
N { – Установить на приборе XX новое значение
 маски опрашиваемых параметров переменных.
 (см. [ниже](#)).

VarParam
 } (кроме приборов M1900).

} |
R { – Установить/отключить режим дистанционного
 управления индикатором прибора XX.
0 | – отключить режим,
1 | – включить режим с ранее установленными
 параметрами,
2 { – включить режим с заданными
 параметрами.
SyncTimeout – максимальное время ожидания
 опроса со стороны компьютера до выдачи ошибки
 «Err.100» (1..60 сек),
ReturnMode – возврат в весовой режим после
 возникновения ошибки «Err.100» (0 – запрещен,
 1 – разрешен по нажатию кнопки **Ноль**),
ErrorMode – Индикация ошибок прибора в
 режиме дистанционного управления индикатором
 (0 – нет, 1 – есть).
 }

} | (только для приборов M0601 версии 5.00 и выше).
i – Инициализировать прибор XX.
O { – Состояние выходов Outputs.
Outputs
 } | (только для приборов M2606).

}, где

- {...} – обязательный параметр,
- [...] – необязательный параметр,
- ... | – вариантный параметр,
- XX** – уникальный номер прибора в сети RS-485,
- YY** – уникальный номер табло в сети RS-485, подключаемое к прибору XX.

Примеры:

1. Добавить прибор №1 в список подключенных приборов и назначить ему табло №2.

Команда на порт “**DevNetRpc**”: @D01+02

Ответ на порт “**DevNetDrv**”: #D01+02

2. Получить из прибора №2 значение веса брутто.

Команда на порт “**DevNetRpc**”: @D02GV0

Ответ на порт “**DevNetDrv**”: #D02GV0**28,375**

3. Получить из прибора №32 значение веса нетто.

Команда на порт “**DevNetRpc**”: @D32GV1

Ответ на порт “**DevNetDrv**”: #D?? – недопустимый номер прибора

4. Нажать на приборе №3 кнопку **Тара**.

Команда на порт “**DevNetRpc**”: @D03ST

Ответ на порт “**DevNetDrv**”: #@D03ST**E** – невозможно выполнить команду по каким-либо причинам (например, нет стабильности веса).

5. Подать в прибор №1 несуществующую команду.

Команда на порт “**DevNetRpc**”: @D01GG

Ответ на порт “**DevNetDrv**”: #D01G?

2.4.3 Функции, реализованные в сервере автоматизации “DevNet.Drv”

Для управления программой, как драйвером по технологии **OLE Automation** необходимо зарегистрировать сервер автоматизации “**DevNet.Drv**” в реестре операционной системы Windows. Для этого достаточно запустить на исполнение файл **DevNet.exe**, настроить параметры программы и закрыть ее.

Все функции (за исключением TestPort, OpenPort, ClosePort, SetPortDlg, SetParamDlg, SelectDevDlg, GetDevList, GetVersion) имеют обязательный параметр DevNum – номер прибора. При успешном выполнении функции возвращают результат True, иначе – False.

Используемые типы данных перечислены в [Таблице 1](#).

Таблица 1. Типы данных

Pascal-тип	IDL-тип	Комментарий
Byte	unsigned char	1 byte unsigned integer
Word	unsigned short	2 byte unsigned integer
Smallint	short	2 byte signed integer
DWord	unsigned int	4 byte unsigned integer
WordBool	VARIANT_BOOL	True = -1, False = 0
Double	Double	8-byte real
WideString	BSTR	binary string
OleVariant	VARIANT	Ole Variant

2.4.3.1 Функции, используемые для всех приборов

function **TestPort**(out PortNum, BaudRate: Byte; out AutoConnect: WordBool): WordBool;

– Проверить состояние последовательного порта.

Параметры:

PortNum – номер последовательного порта
(Com1, ..., Com8),

BaudRate – скорость обмена
(BaudRate_1200, ..., BaudRate_57600),

AutoConnect – режим авто-поиска подключенных к сети RS-485 приборов:
включен (True), выключен (False).

function **OpenPort**[(PortNum, BaudRate: Byte; AutoConnect: WordBool)]: WordBool;

– Открыть последовательный порт.

Параметры:

PortNum – номер последовательного порта
(Com_1, ..., Com_8),

BaudRate – скорость обмена
(BaudRate_1200, ..., BaudRate_57600),

AutoConnect – режим авто-поиска подключенных к сети RS-485 приборов:
включен (True), выключен (False).

function **ClosePort**: WordBool;

– Закрыть последовательный порт.

function **SetPortDlg**: WordBool;

– Вызвать диалог настройки последовательного порта.

function **SetParamDlg**: WordBool;

– Вызвать диалог настройки параметров программы DevNet.

- function **SelectDevDlg**: WordBool;
– Вызвать диалог выбора списка приборов, подключаемых к сети RS-485.
- function **GetVersion**: WideString;
– Получить строку с версией драйвера.
- property **Visible**: WordBool;
– Показать (True), скрыть (False) окно DevNet.
- function **GetDevList**(out DevList: WideString): WordBool;
– Получить список адресов подключенных к сети RS-485 приборов в формате: <Адрес1><Тип1><Версия1>;<Адрес2>... – см. выше [TypeDev](#), [VersionDev](#).
- function **ConnectDev**(DevNum, SubDevNum: Byte; Connect: WordBool): WordBool;
– Подключить/отключить прибор DevNum к списку приборов, подключить/отключить дублирующее табло SubDevNum к прибору DevNum. Если дублирующее табло не используется, то SubDevNum = 0.
Параметры:
Connect – подключить (True), отключить (False) прибор.
- function **InitDev**(DevNum: Byte): WordBool;
– Инициализировать прибор DevNum.
- function **GetTypeDev**(DevNum: Byte; out TypeDev: Byte): WordBool;
– Получить код типа **TypeDev** прибора DevNum – см. [ВЫШЕ](#).
- function **GetVersionDev**(DevNum: Byte; out VersionDev: Byte): WordBool;
– Получить код версии **VersionDev** прибора DevNum – см. [ВЫШЕ](#).
- function **GetStateDev**(DevNum: Byte; out ErrState, Flags0, Flags1, DFlags, DState: Byte): WordBool;
– Получить бит-флаги состояния прибора DevNum
Параметры:
[ErrState](#), [Flags0](#), [Flags1](#), [DFlags](#), [DState](#) – см. выше.
- function **GetWeight**(DevNum: Byte; TypeWeight: Byte; out Weight: Double; out ErrState, Flags0, Flags1, DFlags, DState: Byte): WordBool;
– Получить значение веса из прибора DevNum.
Параметры:
TypeWeight – тип значения веса:
 брутто (M06A_Brutto), нетто (M06A_Netto), тапа (M06A_Tare),
 “рабочий” ноль (M06A_Zero),
Weight – значение веса,
[ErrState](#), [Flags0](#), [Flags1](#), [DFlags](#), [DState](#) – см. выше.
- function **GetVariable**(DevNum: Byte; TypeVariable: Byte; out Variable: OleVariant): WordBool;
– Получить значение переменной из прибора DevNum.
Параметры:
TypeVariable – тип значения переменной
(см. выше – параметры команды [@DXXGV](#)),
Variable – значение переменной.
- function **GetVarByList**(DevNum: Byte; const List: WideString; Param: Byte; out VarList: WideString): WordBool;
– Получить список значений (Param = 0) или имен (Param = 1) переменных

из прибора DevNum.

Параметры:

List – список типов переменных, разделенных ‘,’

(см. выше – параметры команды @DXXGV).

VarList – список значений переменных в формате ASCII строки:

<Значение переменной1>;<Значение переменной2>... (Param = 0)

<Имя переменной1>;<Имя переменной2>... (Param = 1)

function **GetConstant**(DevNum: Byte; TypeConstant: Byte; out Constant: OleVariant): WordBool;

– Получить значение константы из прибора DevNum.

Параметры:

TypeConstant – тип значения константы

(см. выше – параметры команды @DXXGC),

Constant – значение константы.

function **IsReadyDev**(DevNum: Byte): WordBool;

– Проверить, что все параметры прибора DevNum получены и прибор не занят выполнением каких-либо функций.

function **SetToZero**(DevNum: Byte): WordBool;

– Нажать на приборе DevNum кнопку **Ноль**.

function **UndoZero**(DevNum: Byte): WordBool;

– Отменить нажатие на приборе DevNum кнопки **Ноль**.

function **TakeToTare**(DevNum: Byte): WordBool;

– Нажать на приборе DevNum кнопку **Тара**.

function **UndoTare**(DevNum: Byte): WordBool;

– Отменить нажатие на приборе DevNum кнопки **Тара**.

function **SaveWeight**(DevNum: Byte): WordBool;

– Нажать на приборе DevNum кнопку **Вес**.

function **SelectMode**(DevNum: Byte): WordBool;

– Нажать на приборе DevNum кнопку **Режим**.

function **LockKeysDev**(DevNum: Byte; Lock: WordBool): WordBool;

– Заблокировать/Разблокировать кнопки прибора DevNum.

Параметры:

Lock – заблокировать (True), разблокировать (False) кнопки прибора.

property **DataParam**[DevNum: Byte]: Byte;

– Получить/установить значение маски опрашиваемых параметров данных из прибора DevNum (см. [ниже](#)).

Поле DataParam (для приборов M06A, M06Д, M06Д6, M06К):

Бит	Название	Комментарий
0	fAdcCode	код АЦП
1	fBrutto	вес брутто
2	fNetto	вес нетто
3	fTare	вес тары
4	fZero	«рабочий» ноль
5	fStateBits	бит-флаги состояния прибора
6	fIndicator	копия индикатора прибора

property **VarParam**[DevNum: Byte]: Byte;

– Получить/установить значение маски опрашиваемых параметров переменных из прибора DevNum (см. [ниже](#)).

Поле VarParam (для приборов M06A):

Бит	Название	Комментарий
0	fHold	сумма нетто
1	fCounter	счетчик
2	fSBrutto	исходный вес брутто

Поле VarParam (для приборов M06Д, M06Д6):

Бит	Название	Комментарий
0	fTotal	счетчик загрузки/выгрузки (Pro 0) или общая набранная доза (Pro 1)
1	fCounter	номер дозы (Pro 0) или общий счетчик порций (Pro 1)
3	fStateBits	бит-флаги состояния прибора
4	fWrkTotal	текущая набранная доза и текущий счетчик порций (Pro 1)

Поле VarParam (для приборов M06К):

Бит	Название	Комментарий
0	fIntegral	счетчик-интегратор
1	fRate	мгн. производительность
2	fFillTime	время интегрирования
3	fSpeed	мгн. скорость
4	fDensity	мгн. лин. плотность
5	fLIntegral	длина ленты
6	fStateBits	бит-флаги состояния прибора

Поле VarParam (для приборов M1600):

Бит	Название	Комментарий
0	fWState	режим управления прибором
1	fFlowRate Measured	измеренная производительность
2	fFlowRate Current	заданная производительность
3	fErrState	код ошибки
4	fPLC_Labels	внутренние переменные PLC
5	fPLC_EXT_IO	дискретные входы/выходы PLC
6	fPLC_KC_IO	входы/выходы шины MS-Bus
7	fStFlags1	бит-флаги состояния прибора

2.4.3.2 Функции, используемые только для приборов M0601 версии 4.90 и выше

function **CheckMailBox**(DevNum: Byte; out MailType, MailNum, MailTotalParts: Byte): WordBool;

– Проверить наличие сообщения от прибора DevNum.

Параметры:

MailType – тип сообщения:

перезагрузка прибора (Mail_NewCPU),
вес зафиксирован, а коды пользователя не вводились (Mail_HoldCounted),
вес не фиксировался, а коды пользователя введены (Mail_UserCodesOnly),
вес зафиксирован, и коды пользователя введены (Mail_UserCodesAfterFix),

MailNum – ключ сообщения,

MailTotalParts – количество частей в сообщении.

function **ReadMailBox**(DevNum, MailNum: Byte; ClearAfterRead: WordBool; out UserCode1, UserCode2, UserCode3: Byte; out OrderNum: Integer; out Range: Byte; out Brutto, Netto, Tare: Double; out Flags, ErrState: Byte; out SumTotal: Double; out SumCounter: Integer): WordBool;

– Прочитать полученное сообщение от прибора DevNum.

Параметры:

MailNum – ключ сообщения из функции CheckMailBox,
ClearAfterRead – очистить (True), оставить (False) сообщение после прочтения,
UserCode1, UserCode2, UserCode3 – введенные коды пользователя,
OrderNum – порядковый номер ввода (сообщения),
Range – текущий весовой диапазон,
Brutto, Netto, Tare, **ErrState** – см. [выше](#),
Flags – бит-флаг параметров сообщения (см. [ниже](#)),
SumTotal – сумма нетто,
SumCounter – счетчик.

function **ClearMailBox**(DevNum, MailNum: Byte): WordBool;

– Очистить сообщение MailNum от прибора DevNum.

Параметры:

MailNum – ключ сообщения из функции CheckMailBox.

Поле **Flags**:

Бит	Название	Комментарий
0	fStableFix	зафиксированы стабильные вес и тара
1	fAutoFix	фиксирование произошло в режиме авто-заморозки, иначе вручную пользователем по нажатию кнопки {ПЕЧАТЬ}
2	fManualTare	тара была введена пользователем вручную, иначе взвешена по кнопке {ТАРА}
3	fNoUnload	не было разгрузки весов после предыдущей фиксации
4	fNoCodes	коды пользователя не вводились

2.4.3.3 Функции, используемые только для приборов M06Д, M06Д6, M06К

function **TurnDosator**(DevNum: Byte; Turn: WordBool): WordBool;

– Нажать на приборе DevNum кнопку **Вкл./Выкл.**

Параметры:

Turn – включить (True), выключить (False) режим “дозатор”.

2.4.3.4 Функции, используемые только для приборов M06Д, M06Д6, M06К, M1600

function **SwitchDosator**(DevNum: Byte; Switch: WordBool): WordBool;

– Нажать на приборе DevNum кнопку **Старт/Стоп.**

Параметры:

Switch – включить (True), выключить (False) цикл дозирования в режиме “дозатор”.

function **SetLimitsDlg**(DevNum: Byte): WordBool;

– Нажать на приборе DevNum кнопку **Уставки** и вызвать диалог настройки уставок.

function **SetLimits**(DevNum: Byte; const Limits: WideString): WordBool;

– Записать в прибор DevNum значения уставок **Limits** (см. [Таблицы 2..5](#)).

function **GetLimits**(DevNum: Byte; out Limits: WideString): WordBool;

– Получить из прибора DevNum значения уставок **Limits** (см. [Таблицы 2..5](#)).

2.4.3.5 Функции, используемые только для прибора M06Д6

function **ClearDozaCounter** (DevNum: Byte; StartAfterClear: WordBool): WordBool;

– Обнулить рабочий счетчик (WrkCounter) дозатора

Параметры:

DevNum – сетевой адрес прибора

StartAfterClear – если равно «True», то после обнуления включить цикл дозирования в режиме “дозатор”.

function **SetNeedToSaveLimits** (DevNum: Byte; DoSave: WordBool): WordBool.

- Функция указывает, следует ли сохранять новую уставку в EEPROM прибора. Флаг сохранения, заданный этой функцией, используется при выполнении функции **SetLimits** и при записи уставки непосредственно из программы DevNet.()

Параметры:

DevNum – сетевой адрес прибора

DoSave – сохранять (True), не сохранять (False).

Таблица 2. Структура таблицы уставок Limits (для приборов М06Д):

Название поля	Размер	Тип	Комментарий
Флаги дозатора	1	Byte	Бит-флаги таблицы уставок (см. ниже)
Уставка-0	2	Word	Значение уставки 0
Уставка-1	2	Word	Значение уставки 1
Уставка-2	2	Word	Значение уставки 2 при работе по алгоритму "Pro 0"
Задержка уставки-0	1	Byte	Задержка срабатывания уставки 0
Задержка уставки-1	1	Byte	Задержка срабатывания уставки 1
Задержка уставки-2	1	Byte	Задержка срабатывания уставки 2
Полная доза	4	DWord	Значение уставки 2 при работе по алгоритму "Pro 1"
Позиция дес. точки	1	Byte	Позиция дес. точки для полной дозы

Таблица 3. Структура таблицы уставок Limits (для приборов М06Д6):

Название поля	Размер	Тип	Комментарий
Флаги дозатора	1	Byte	Бит-флаги таблицы уставок (см. ниже)
Доп. флаги дозатора	1	Byte	Доп. бит-флаги таблицы уставок (см. ниже)
Уставка-0	2	Word	Значение уставки 0
Уставка-1	2	Word	Значение уставки 1
Уставка-2	2	Word	Значение уставки 2 при работе по алгоритму "Pro 0"
Уставка-3	2	Word	Значение уставки 3
Уставка-4	2	Word	Значение уставки 4
Уставка-5	2	Word	Значение уставки 5
Уставка-6	2	Word	Значение уставки 6
Задержка уставки-0	1	Byte	Задержка срабатывания уставки 0
Задержка уставки-1	1	Byte	Задержка срабатывания уставки 1
Задержка уставки-2	1	Byte	Задержка срабатывания уставки 2
Задержка уставки-3	1	Byte	Задержка срабатывания уставки 3
Задержка уставки-4	1	Byte	Задержка срабатывания уставки 4
Задержка уставки-5	1	Byte	Задержка срабатывания уставки 5
Задержка уставки-6	1	Byte	Задержка срабатывания уставки 6
Полная доза	4	DWord	Значение уставки 2 при работе по алгоритму "Pro 1"
Позиция дес. точки	1	Byte	Позиция дес. точки для полной дозы
Доп. флаги2 дозатора	1	Byte	Доп. бит-флаги2 таблицы уставок (см. ниже)
Уставка-7	4	DWord	Значение уставки 7 (Производительность)
Позиция дес. точки	1	Byte	Позиция дес. точки для уставки 7 (Производительность)
Ед. измерения	1	Byte	Ед. изм. производительности: 0 – кг/сек, 1 – кг/мин, 2 – кг/час, 3 – тонн/сек, 4 – тонн/мин, 5 – тонн/час

Уставка-8	4	DWord	Значение уставки 8 (Суммарная доза)
Позиция дес. точки	1	Byte	Позиция дес. точки для уставки 8 (Суммарная доза)
Уставка-8	4	DWord	Значение уставки 8 (Упреждение)
Позиция дес. точки	1	Byte	Позиция дес. точки для уставки 8 (Упреждение)

Таблица 4. Структура таблицы уставок Limits (для приборов M06K):

Название поля	Размер	Тип	Комментарий
Флаги дозатора	1	Byte	Бит-флаги таблицы уставок (см. ниже)
Уставка-0	2	Word	Значение уставки 0
Уставка-1	4	DWord	Значение уставки 1
Задержка уставки-0	1	Byte	Задержка срабатывания уставки 0
Уставка-2	2	Word	Значение уставки 2

Таблица 5. Структура таблицы уставок Limits (для приборов M1600):

Название поля	Размер	Тип	Комментарий
Коеф. усиления (Cp)	2	Word	Значение коеф. усиления
Коеф. интегральной части (Ci)	2	Word	Значение коеф. интегральной части
Коеф. дифференциальной части (Cd)	2	Word	Значение коеф. дифференциальной части
Шаг регулирования (td)	1	Byte	Значение шага регулирования
Задержка начального включения (Pd)	1	Byte	Значение задержки начального включения, в dT
Порог начального включения (PL)	1	Byte	Значение порога начального включения, в % от заданной производительности

Поле Флаги дозатора (для приборов М06Д, М06Д6):

Бит	Название	Комментарий
0	fNettoLimit0	Уставка 0 по Нетто (1 - "L.00 nE") или по Брутто (0 - "L.00 br")
1	fUsedLimit0	Уставка 0 используется (1 - "L.00 nE/br") или не используется (0 - "L.00 OF")
2	fNettoLimit1	Уставка 1 по Нетто (1 - "L.01 nE") или по Брутто (0 - "L.01 br")
3	fUsedLimit1	Уставка 1 используется (1 - "L.01 nE/br/rE/Ро") или не используется (0 - "L.01 OF")
4	fNettoLimit2	Уставка 2 по Нетто (1 - "L.02 nE") или по Брутто (0 - "L.02 br")
5	fUsedLimit2	Уставка 2 используется (1 - "L.02 nE/br/Fu") или не используется (0 - "L.02 OF")
6	fRelativeLimit1	Уставка 1 в процентах относительно уставки 2 (1 - "L.01 rE"), при этом флаг fNettoLimit1 не используется
7	fPortionMode	Порционный алгоритм (1 - "Pro 1"), при этом флаги fNettoLimit1, fNettoLimit2, fRelativeLimit1 не используются

Поле Доп. флаги дозатора (для приборов М06Д6):

Бит	Название	Комментарий
0	fNettoLimit3	Уставка 3 по Нетто (1 - "L.03 nE") или по Брутто (0 - "L.03 br")
1	fUsedLimit3	Уставка 3 используется (1 - "L.03 nE/br") или не используется (0 - "L.03 OF")
2	fNettoLimit4	Уставка 4 по Нетто (1 - "L.04 nE") или по Брутто (0 - "L.04 br")
3	fUsedLimit4	Уставка 4 используется (1 - "L.04 nE/br") или не используется (0 - "L.04 OF")
4	fNettoLimit5	Уставка 5 по Нетто (1 - "L.05 nE") или по Брутто (0 - "L.05 br")
5	fUsedLimit5	Уставка 5 используется (1 - "L.05 nE/br") или не используется (0 - "L.05 OF")
6	fNettoLimit6	Уставка 6 по Нетто (1 - "L.06 nE") или по Брутто (0 - "L.06 br")
7	fUsedLimit6	Уставка 6 используется (1 - "L.06 nE/br") или не используется (0 - "L.06 OF")

Поле Доп. флаги2 дозатора (для приборов М06Д6):

Бит	Название	Комментарий
1	fUsedLimit7	Уставка 7 используется (1 - "L.07 rAt") или не используется (0 - "L.07 OF")
3	fUsedLimit8	Уставка 8 используется (1 - "L.08 Su") или не используется (0 - "L.08 OF")

Поле Флаги дозатора (для приборов М06К):

Бит	Название	Комментарий
0	fRateLimit0	Уставка 0 по Производит. (1 - "L.00 rA") или по Лин. плотн. (0 - "L.00 dE")
1	fUsedLimit0	Уставка 0 используется (1 - "L.00 rA/dE") или не используется (0 - "L.00 OF")
3	fUsedLimit1	Уставка 1 используется (1 - "L.01 On") или не используется (0 - "L.01 OF")
5	fUsedLimit2	Уставка 2 используется (1 - "L.02 On") или не используется (0 - "L.02 OF")

2.4.3.5 Функции, используемые только для приборов M1600

function **SetFlowRate**(DevNum: Byte; FlowRate: Double): WordBool;
– Установить на приборе DevNum новую заданную производительность FlowRate.

2.4.3.6 Функции, используемые только для приборов M1900

function **SetTabloLed**(DevNum: Byte; const LedBuf: WideString): WordBool;
– Установить сообщение LedBuf для вывода на индикатор прибора DevNum.
В настройках программы необходимо установить “Вывод на табло – РС”.

function **GetTabloKey**(DevNum: Byte; out KeyPressed: Byte): WordBool;
– Получить код KeyPressed нажатых на приборе DevNum кнопок.

2.4.3.7 Функции, используемые только для приборов M2606

function **GetParams**(DevNum: Byte; out Params: WideString): WordBool;
– Получить значения параметров прибора DevNum.

function **ReadInputs**(DevNum: Byte; out Inputs: WideString;
out PLCNum: Byte): WordBool;
– Прочитать состояния входов Inputs прибора DevNum.
Параметры:
PLCNum – счетчик изменений состояния PLC модуля. Циклически изменяет значение от 1 до 255 при каждом изменении состояния PLC модуля.

function **WriteOutputs**(DevNum: Byte; const Outputs: WideString): WordBool;
– Установить состояния выходов Outputs прибора DevNum.

2.4.3.8 Функции, используемые только для приборов M0601 версии 5.00 и выше

function **SetLedMask**(DevNum: Byte; const LedBuf: WideString): WordBool;

– Установить сообщение LedBuf для вывода на индикатор прибора DevNum в режиме дистанционного управления индикатором.

function **SetLedMaskStr**(DevNum: Byte; const LedBuf, BlinkLed, BlinkDiod: WideString): WordBool;

– Установить сообщение LedBuf для вывода на индикатор прибора DevNum в режиме дистанционного управления индикатором.

Параметры:

LedBuf – строка вида «-t.E.S.t.-» (6 байт без учета символа «.») в формате ASCII для вывода на индикатор,

BlinkLed – строка вида «00100111» (8 байт) в формате ASCII, задает мигание символа в соответствующей позиции индикатора

(0 – горит постоянно, 1 – горит ½ сек. (bit7 = 0) или ¼ сек. (bit7 = 1)),

BlinkDiod – строка вида «01230123» (8 байт) в формате ASCII, задает мигание соответствующих светодиодов

(0 – не горит, 1 – горит ¼ сек., 2 – горит ¾ сек., 3 – горит постоянно).

function **GetTabloKey**(DevNum: Byte; out KeyPressed: Byte): WordBool;

– Получить код KeyPressed нажатых на приборе DevNum кнопок в режиме дистанционного управления индикатором.

function **SetRemoteControlDev**(DevNum[, InputMode, SyncTimeout, ReturnMode, ErrorMode]: Byte): WordBool;

– Установить/отключить режим дистанционного управления индикатором на приборе DevNum.

Параметры:

InputMode – отключить режим (0), включить режим с ранее установленными параметрами (1), включить режим с заданными параметрами (2),

SyncTimeout – максимальное время ожидания опроса со стороны компьютера до выдачи ошибки «Err.100» (1..60) сек.,

ReturnMode – возврат в весовой режим после возникновения ошибки «Err.100»: запрещен (0), разрешен по нажатию кнопки {НОЛЬ} (1),

ErrorMode – Индикация ошибок прибора в режиме дистанционного управления индикатором:

нет (0), есть(1).

function **IsRemoteControlDev**(DevNum: Byte): WordBool;

– Проверить, что прибор DevNum находится в режиме дистанционного управления индикатором.

function **RemoteControlParam**(DevNum: Byte; out InputMode, SyncTimeout, ReturnMode, ErrorMode: Byte): WordBool;

– Получить параметры (см. выше) режима дистанционного управления индикатором из прибора DevNum.

2.4.3.9 Функции для системы “1С:Предприятие” и языков программирования, не поддерживающих передачу значений по ссылке в параметрах функции

Примечание: функции возвращают запрошенный параметр или Null – при не успешном выполнении.

function **GetVariable1C**(DevNum, TypeVariable: Byte): OleVariant;
– Аналог функции GetVariable.

function **GetVarByList1C**(DevNum: Byte; const List: WideString; Param: Byte): OleVariant;
– Аналог функции GetVarByList.

function **GetConstant1C**(DevNum, TypeConstant: Byte): OleVariant;
– Аналог функции GetConstant.

function **CheckMailBox1C**(DevNum, TypeParam: Byte): OleVariant;
– Аналог функции CheckMailBox.

Параметры:

TypeParam – тип параметра:

тип сообщения (Mail_Type) – см. выше (MailType),
ключ сообщения (Mail_Num),
количество частей в сообщении (Mail_TotalParts).

function **ReadMailBox1C**(DevNum, MailNum, TypeParam: Byte): OleVariant;
– Аналог функции ReadMailBox.

Параметры:

MailNum – ключ сообщения из функции CheckMailBox1C,

TypeParam – тип параметра:

введенные коды пользователя (Mail_UserCode1.. Mail_UserCode3),
порядковый номер сообщения (Mail_OrderNum),
текущий весовой диапазон (Mail_Range),
брутто (Mail_Brutto),
нетто (Mail_Netto),
тара (Mail_Tare),
бит-флаг параметров сообщения (Mail_Flags) – см. выше **Flags**,
код ошибки (Mail_ErrState) – см. выше **ErrState**,
сумма нетто (Mail_SumTotal),
счетчик (Mail_SumCounter).

function **GetTabloKey1C**(DevNum: Byte): OleVariant;
– Аналог функции GetTabloKey.

function **GetDevList1C**: OleVariant;
– Аналог функции GetDevList.

3 Проверка программы

Для проверки работоспособности программы необходимо выполнить следующие действия:

- 1) Запустите на выполнение с помощью **Проводника** Windows или любым другим способом файл **Samples\Delphi\ControlScales.exe**. На экране появится окно программы «Контроль весов».
- 2) В группе переключателей **Удаленное управление весами по** выберите значение **Ole Automation**.
- 3) Несколько раз нажмите кнопку **Драйвер весов**. При этом на экране должно появляться и исчезать окно программы «Сеть приборов».
- 4) В группе переключателей **Удаленное управление весами по** выберите значение **MailSlot**.
- 5) В поле **Команда:** введите одну из команд, описанных в разделе 2.2.2, и нажмите клавишу «Enter».
- 6) В поле **Ответ:** появится результат выполнения введенной команды.
- 7) Для завершения работы с программой «Контроль весов» нажмите кнопку **Выход**.

Проверка работоспособности программы «Сеть весов» может быть так же осуществлена из прикладной программы пользователя с помощью вызова описанных методов. Порядок вызова методов показан в примерах.

Приложение А. Примеры использования программы

В комплект поставки программы входят:

- модули на языке программирования Delphi для использования технологий **MailSlot** и **FileMapping**;
- пример **Весы.ert**, демонстрирующий использование DevNet в качестве драйвера для управления приборами и получения от них данных в 1С: Предприятие по технологии **OLE Automation**;
- пример **Весы.mdb**, демонстрирующий использование DevNet, как драйвера для управления приборами и получения от них данных в MS Access 2000 по технологии **OLE Automation**;
- пример **ControlScales** на языке программирования Delphi, демонстрирующий использование DevNet в качестве драйвера для управления приборами и получения от них данных по технологиям **MailSlot** и **OLE Automation**;
- программа **WeightImitator**, имитирующая работу DevNet без подключения приборов к компьютеру. Используется вместо DevNet для отладки приложений пользователя по технологиям **MailSlot** и **FileMapping**.

А.1 Пример использования сервера автоматизации “DevNet.Drv” на языке программирования Delphi

```

procedure SomeProcedure;                                // Какая-то процедура
var
    DevNet: OleVariant;
begin
    DevNet := CreateOleObject('DevNet.Drv');             // Установить связь с сервером
                                                         // автоматизации "DevNet.Drv"
    if not DevNet.SetToZero(1) then                     // Нажать на приборе с номером 1
                                                         // кнопку Ноль
        ShowMessage('Невозможно установить ноль на выбранном приборе.'#13#10+
            'Возможные причины: показания веса нестабильны, прибор не выбран'#13#10+
            'для работы, нет связи с прибором');
    DevNet := UnAssigned;                                // Дальнейшей работы с сервером
                                                         // автоматизации "DevNet.Drv" не будет
end;

```

А.2 Пример использования сервера автоматизации “DevNet.Drv” на языке программирования Visual Basic (VBA)

```

Dim objDevNet
Private Sub SomeProcedure()                             ' Какая-то процедура
    Set objDevNet = CreateObject("DevNet.Drv")         ' Установить связь с сервером
                                                         ' автоматизации "DevNet.Drv"
    If Not objDevNet.SetToZero(1) Then                 ' Нажать на приборе с номером 1
                                                         ' кнопку Ноль
        MsgBox ("Невозможно установить ноль на выбранном приборе." + _
            "Возможные причины: показания веса нестабильны, прибор не выбран" + _
            "для работы, нет связи с прибором")
    End If
    Set objDevNet = Nothing                             ' Дальнейшей работы с сервером
                                                         ' автоматизации "DevNet.Drv" не будет
End Sub

```

End Sub

А.3 Пример использования сервера автоматизации “DevNet.Drv” на языке программирования “1С: Предприятие”

В **Глобальный модуль** поместить код создания сервера OLE автоматизации. Для этого в секции объявления глобальных переменных необходимо объявить объект.

Перем DevNet Экспорт;

А в процедуре **ПриНачалеРаботыСистемы** создать объект:

```
DevNet = СоздатьОбъект("DevNet.Drv");
```

Создать новый документ или взять существующий и в его реквизиты шапки добавить параметры “Вес” с типом значения “*Число*”. Затем вывести форму этого документа и через меню “Вставить”/“Реквизиты...” разместить на форме окно, куда будет записываться полученный вес. Далее на форме размещается кнопка “Получить вес”, вызывается окно свойств для этой кнопки, и на закладке “Дополнительно” в окне “Формула” указывается функция “**ПолучитьВес()**”, которая будет выполняться при нажатии кнопки. Осталось только, щелкнув по закладке “Модуль” формы, определить функцию **ПолучитьВес ()**.

Процедура ПолучитьВес(DevNum, TypeWeight)

```
Вес = DevNet.GetVariable1C(DevNum, TypeWeight)
```

КонецПроцедуры.

Приложение Б. Значения используемых констант

```
// Последовательные порты
No_Com = 0;
Com_1 = 1;
Com_2 = 2;
Com_3 = 3;
Com_4 = 4;
Com_5 = 5;
Com_6 = 6;
Com_7 = 7;
Com_8 = 8;

// Скорости обмена
BaudRate_1200 = 0;
BaudRate_2400 = 1;
BaudRate_4800 = 2;
BaudRate_9600 = 3;
BaudRate_19200 = 4;
BaudRate_38400 = 5;
BaudRate_57600 = 6;

// Типы приборов
TypeDev_M06A = 0;
TypeDev_M06D = 1;
TypeDev_M06D6 = 2;
TypeDev_M06K = 3;
TypeDev_M1600 = 4;
TypeDev_M1900 = 5;
TypeDev_M2606 = 6;
TypeDev_Unknown = 255;

// Версии приборов
VersionDev_31X = 0;
VersionDev_32X = 1;
VersionDev_34X = 2;
VersionDev_35X = 3;
VersionDev_37X = 4;
VersionDev_41X = 5;
VersionDev_43X = 6;
VersionDev_49X = 7;
VersionDev_492 = 8;
VersionDev_50X = 9;
VersionDev_14X = 10;
VersionDev_2X = 11;
VersionDev_Unknown = 255;

// Переменные M06A
M06A_Brutto = 0;
M06A_Netto = 1;
M06A_Tare = 2;
M06A_Zero = 3;
M06A_ErrState = 4;
M06A_Flags0 = 5;
M06A_Flags1 = 6;
M06A_DState = 8;
M06A_SumTotal = 10;
M06A_SumCounter = 11;
M06A_SBrutto = 12;
M06A_ADC = 255;

// Переменные M06Д, M06Д6
M06D_Brutto = 0;
M06D_Netto = 1;
```

```
M06D_Tare      = 2;
M06D_Zero     = 3;
M06D_ErrState = 4;
M06D_Flags0   = 5;
M06D_Flags1   = 6;
M06D_DFlags   = 7;
M06D_DState   = 8;
M06D_Limits   = 9;
M06D_SumTotal = 10;
M06D_SumCounter = 11;
M06D_WrkTotal = 12;
M06D_WrkCounter = 13;
M06D_CtrlWeight = 14;
M06D_ADC      = 255;

// Переменные M06K
M06K_Brutto    = 0;
M06K_Netto     = 1;
M06K_Tare      = 2;
M06K_Zero     = 3;
M06K_ErrState = 4;
M06K_Flags0   = 5;
M06K_Flags1   = 6;
M06K_DFlags   = 7;
M06K_DState   = 8;
M06K_Limits   = 9;
M06K_Integral = 10;
M06K_xRate     = 11;
M06K_xRateTarget = 12;
M06K_IntTime   = 13;
M06K_xSpeed    = 14;
M06K_xDensity  = 15;
M06K_LIntegral = 16;
M06K_ADC      = 255;

// Переменные M1600
M1600_FlowRateMeasured = 0;
M1600_FlowRateCurrent  = 1;
M1600_ErrState         = 4;
M1600_Flags1          = 6;
M1600_DState          = 8;
M1600_Limits          = 9;
M1600_PLC_Labels      = 10;
M1600_PLC_LabelsExt   = 11;
M1600_PLC_Ext_Outputs = 12;
M1600_PLC_Ext_Inputs  = 13;
M1600_PLC_KC_Outputs  = 14;
M1600_PLC_KC_Inputs   = 15;

// Переменные M2606
M2606_Inputs = 0;
M2606_PLCNum = 1;

// Константы M06A
M06A_TypeDev    = 0;
M06A_VersionDev = 1;
M06A_MaxRange   = 2;
M06A_PntPos     = 3;
M06A_MaxBrutto  = 4;
M06A_MaxBrutto0 = 4;
M06A_MaxBrutto1 = M06A_MaxBrutto0 + 1;
M06A_MaxBrutto2 = M06A_MaxBrutto0 + 2;
M06A_Discret    = 7;
M06A_Discret0   = 7;
```

```
M06A_Discret1 = M06A_Discret0 + 1;
M06A_Discret2 = M06A_Discret0 + 2;
M06A_EEPROM_BlockCounter = 10;
M06A_EEPROM_ValidMask = 11;
M06A_EEPROM_CRC = 12;
M06A_EEPROM_CRC0 = 12;
M06A_EEPROM_CRC1 = M06A_EEPROM_CRC0 + 1;
M06A_EEPROM_CRC2 = M06A_EEPROM_CRC0 + 2;
M06A_EEPROM_CRC3 = M06A_EEPROM_CRC0 + 3;
M06A_EEPROM_CRC4 = M06A_EEPROM_CRC0 + 4;
M06A_EEPROM_CRC5 = M06A_EEPROM_CRC0 + 5;
M06A_EEPROM_CRC6 = M06A_EEPROM_CRC0 + 6;
M06A_EEPROM_CRC7 = M06A_EEPROM_CRC0 + 7;

// Константы M06Д, M06Д6
M06D_TypeDev = 0;
M06D_VersionDev = 1;
M06D_PntPos = 3;
M06D_MaxBrutto = 4;
M06D_Discret = 7;

// Константы M06К
M06K_TypeDev = 0;
M06K_VersionDev = 1;
M06K_PntPos = 3;
M06K_MaxBrutto = 4;
M06K_Discret = 7;
M06K_xRatePos = 10;
M06K_xDensityPos = 11;
M06K_xSpeedPos = 12;
M06K_IntegralPos = 13;
M06K_LIntegralPos = 14;
M06K_MaxRate = 15;
M06K_MaxDensity = 16;

// Константы M1600
M1600_TypeDev = 0;
M1600_VersionDev = 1;
M1600_PntPos = 3;
M1600_MaxFlowRate = 4;
M1600_Discret = 7;
M1600_EEPROM_BlockCounter = 10;
M1600_EEPROM_ValidMask = 11;
M1600_EEPROM_CRC = 12;
M1600_EEPROM_CRC0 = 12;
M1600_EEPROM_CRC1 = M1600_EEPROM_CRC0 + 1;
M1600_EEPROM_CRC2 = M1600_EEPROM_CRC0 + 2;
M1600_EEPROM_CRC3 = M1600_EEPROM_CRC0 + 3;
M1600_EEPROM_CRC4 = M1600_EEPROM_CRC0 + 4;
M1600_EEPROM_CRC5 = M1600_EEPROM_CRC0 + 5;
M1600_EEPROM_CRC6 = M1600_EEPROM_CRC0 + 6;
M1600_EEPROM_CRC7 = M1600_EEPROM_CRC0 + 7;

// Константы M2606
M2606_TypeDev = 0;
M2606_VersionDev = 1;
M2606_Params = 2;

// Константы системы сообщений
Mail_Type = 0;
Mail_Num = 1;
Mail_TotalParts = 2;

Mail_NewCPU = 1;
```



```
// Только для прибора M0601
Mail_HoldCounted      = 10;
Mail_UserCodesOnly    = 11;
Mail_UserCodesAfterFix = 12;

Mail_UserCode1 = 0;
Mail_UserCode2 = 1;
Mail_UserCode3 = 2;
Mail_OrderNum  = 3;
Mail_Range     = 4;
Mail_Brutto    = 5;
Mail_Netto     = 6;
Mail_Tare      = 7;
Mail_Flags     = 8;
Mail_ErrState  = 9;
Mail_SumTotal  = 10;
Mail_SumCounter = 11;
Mail_RTC       = 12;
No_RTC         = 255;
```