

МЕТРА



МЕТРА

ДРАЙВЕР МНОГОПЛАТФОРМЕННЫХ ВЕСОВ MPSDrv

Версия 1.6.20.0

**РУКОВОДСТВО
СИСТЕМНОГО ПРОГРАММИСТА**

НППМ 435.002.РСП

Научно-производственное предприятие
“Метра”
г. Обнинск
2009

Содержание

1 Общие сведения о программе.....	5
1.1 Назначение.....	5
1.2 Функциональные возможности.....	5
1.3 Защита от нелегального копирования.....	5
1.4 Технические и программные средства.....	6
1.4.1 Минимальная конфигурация компьютера для работы программы.....	6
1.5 Комплектность.....	6
2 Настройка программы.....	7
2.1 Настройка прибора.....	7
2.2 Описание пользовательских параметров прибора M0801.....	12
2.2.1 Блок параметров пользователя "Usr".....	12
2.2.2 Блок параметров калибровки "Cib".....	13
2.2.3 Блок параметров АЦП "Adc".....	14
2.3 Порядок настройки весовых платформ.....	15
2.3 Порядок настройки весовых платформ.....	15
2.4 Программный интерфейс.....	17
2.4.1 Данные, передаваемые на почтовый ящик "DevNet".....	17
2.4.2 Команды, передаваемые в локальной сети на почтовый ящик "MPSDrvRps".....	18
2.4.3 Настройка имен почтовых ящиков.....	18
2.4.4 Функции, реализованные в сервере автоматизации "MPSDrv.Drv".....	19
Приложение А. Примеры использования программы.....	22
А.1 Пример использования сервера автоматизации "MPSDrv.Drv" на языке программирования Delphi.....	22
А.2 Пример использования сервера автоматизации "MPSDrv.Drv" на языке программирования Visual Basic (VBA).....	23
А.3 Пример использования сервера автоматизации "MPSDrv.Drv" на языке программирования "1С: Предприятие".....	23

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

1 Общие сведения о программе

1.1 Назначение

Программа “Драйвер многоплатформенных весов” (MPSDrv) предназначена для работы с приборами M0801 (версии 6 и выше), M0802 (версии 27 и выше), M0803 (версии 27 и выше). Программа может работать с одним или несколькими приборами, объединенными в сеть RS485 и подключенными через интерфейсный коннектор (RS232-RS485) INCON к последовательному порту компьютера типа IBM PC.

1.2 Функциональные возможности

Функциональные возможности программы:

- передача измеренных значений веса через локальную сеть;
- встраивание в АСУ заказчика в качестве драйвера с помощью технологий **MailSlot** (почтовый ящик) и **OLE Automation** (сервер OLE автоматизации).

1.3 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 весовых платформ, 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 одной весовой платформой; возможность передачи значений текущих параметров прибора по локальной сети отключена; ограничений на использование основных функций API нет.

1.4 Технические и программные средства

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

1.4.1 Минимальная конфигурация компьютера для работы программы

- процессор Pentium-150;
- оперативная память 32Mb;
- свободное дисковое пространство 10Mb;
- дисковод 1.44Mb;
- свободный COM порт (без режима "локального эха");
- SVGA видео-плата с разрешением 800x600, 256 цветов;
- 15" SVGA монитор;
- клавиатура;
- мышь.

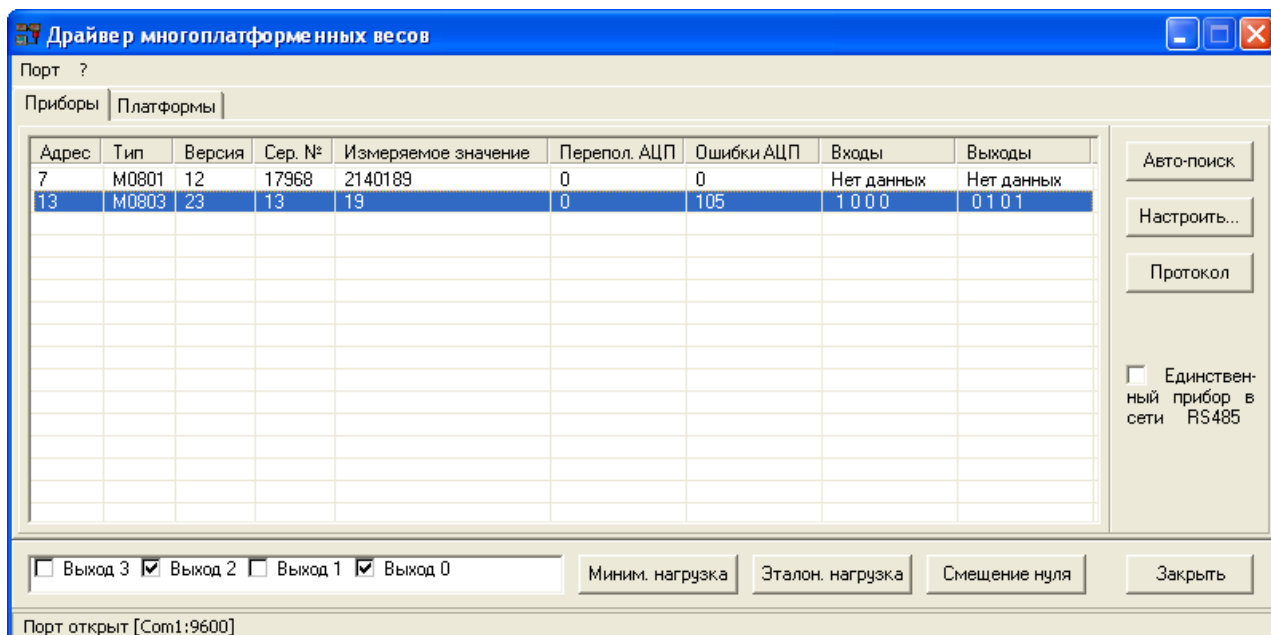
1.5 Комплектность

Диск с программой «Драйвер многоплатформенных весов» (MPSDrv)	1 шт.
Руководство системного программиста НППМ.435.002.РСП	1 экз.
Электронный ключ	1 шт.

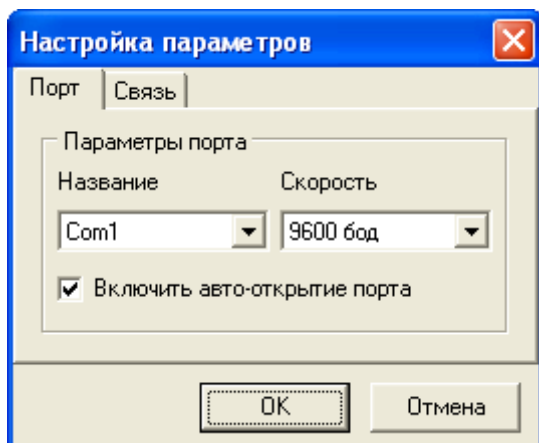
2 Настройка программы

2.1 Настройка прибора

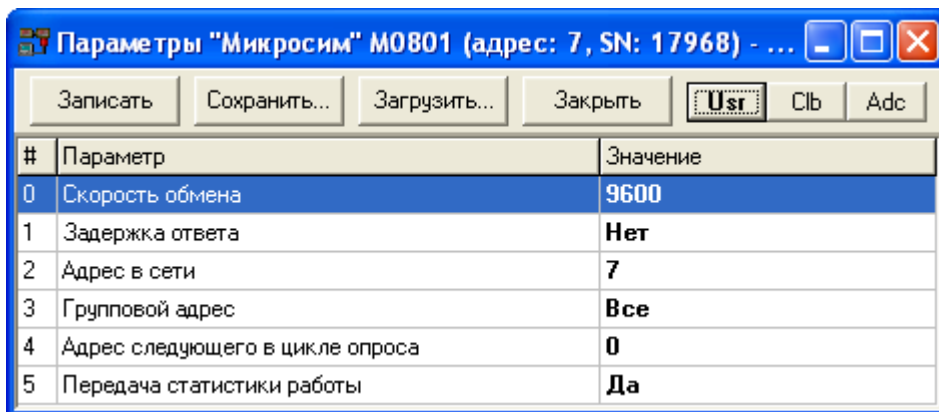
1. Запустить программу **MPSDrv.exe** с параметром **-CLB**.



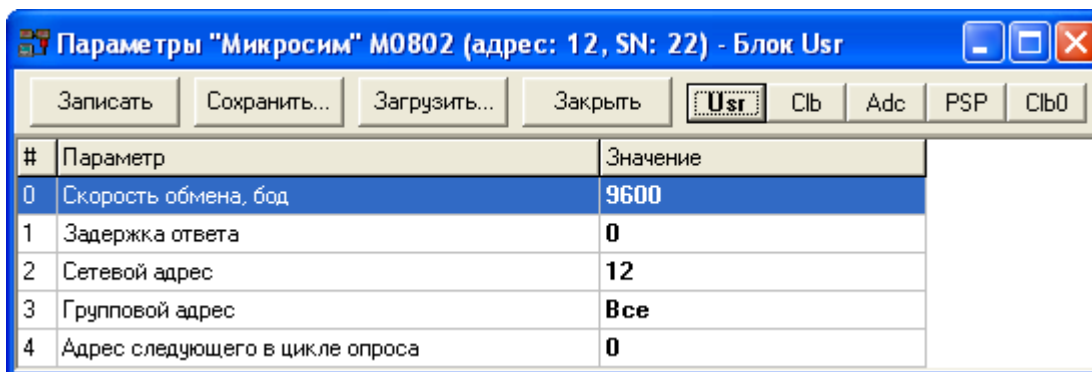
2. Открыть диалог **Настройка параметров** выбрав пункт меню **Порт | Настроить...** и настроить последовательный порт.



3. Подключить к выбранному порту **один** прибор.
4. Установить флажок **Единственный прибор в сети RS485** или нажать кнопку **Авто-поиск**.
5. Настроить прибор нажав кнопку **Настроить...** На экране появится окно настройки параметров прибора.
6. В каждом приборе, работающем под управлением программы “Драйвер многоплатформенных весов” (MPSDrv), рекомендуется установить следующие программные параметры.

В блоке *Usr*:

Для параметра **Задержка ответа** не рекомендуется выбирать значение, меньшее чем **"3 байта"** для высоких скоростей обмена.



#	Параметр	Значение
0	Скорость обмена, бод	9600
1	Задержка ответа	0
2	Сетевой адрес	3
3	Групповой адрес	Все
4	Адрес следующего в цикле опроса	0
5	Время синхронизации с MASTER, с	0
6	Логическая функция	Удаленный ввод/вывод
7	Параметр_1	0
8	Параметр_2	0
9	DO0 при аварии	Выкл.
10	DO1 при аварии	Выкл.
11	DO2 при аварии	Выкл.
12	DO3 при аварии	Выкл.
13	Время устранения дребезга DI0, мс	10
14	Время устранения дребезга DI1, мс	10
15	Время устранения дребезга DI2, мс	10
16	Время устранения дребезга DI3, мс	10
17	Параметр_3	0
18	Параметр_4	0
19	Параметр_5	0
20	Параметр_6	0

Параметр **Скорость обмена** должен быть одинаковым для каждого прибора в сети RS485. Рекомендуется проверить работоспособность настроенных приборов в сети RS485 на более низких скоростях обмена, прежде чем устанавливать максимальную.

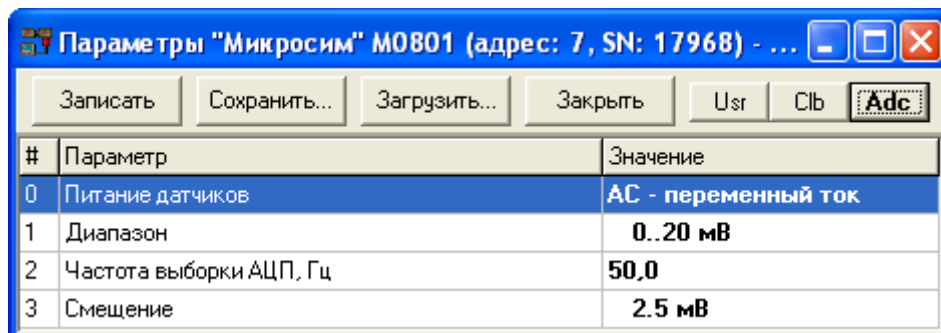
Параметр **Адрес в сети** должен быть уникальным для каждого прибора и отличаться от адреса MASTER-станции. Не рекомендуется выбирать значение адреса в сети, равное "0", т. к. оно зарезервировано за адресом MASTER-станции (компьютера).

В блоке Clb:

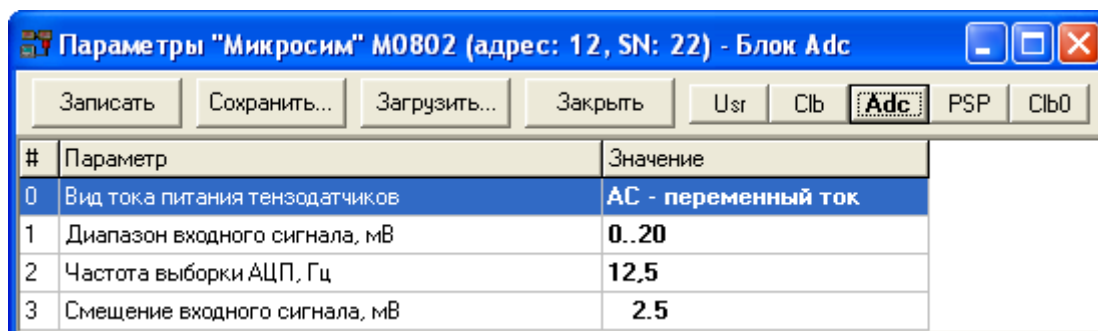
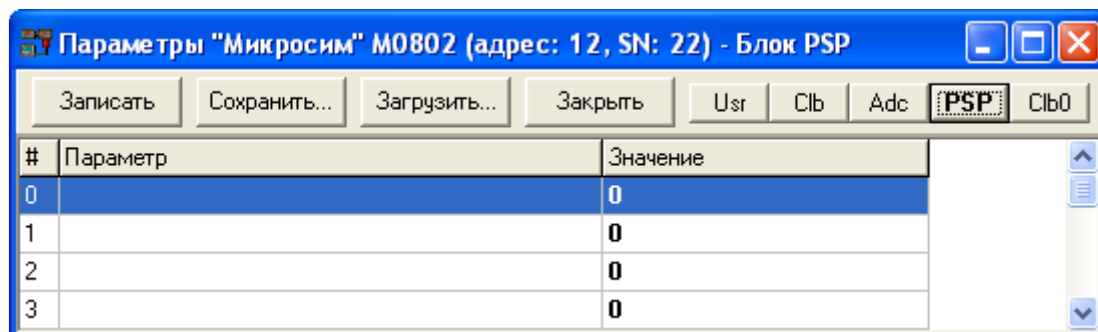
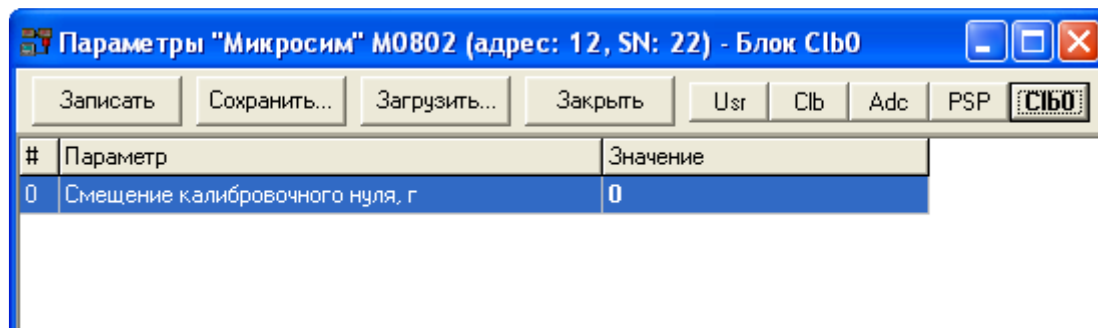
#	Параметр	Значение
0	Суммирование	16
1	Нормализация в 16 разрядов	Без нормализации
2	Статус передавать (v0.09)	Да
3	Шкала в единицах	кода АЦП
4	Внутренняя калибровка при включении питания	Нет
5	Режим постоянной передачи после включения	Нет
6	Авто-поиск скорости обмена	Нет
7	Игнорирование FRAME-ERROR	Да
8	Авто-перезапуск АЦП	Да
9	Код АЦП при минимальной нагрузке	0
10	Код АЦП при эталонной нагрузке	16777215
11	Эталонная нагрузка	1000000
12	Дискрета	1

Параметр **Режим постоянной передачи после включения** необходимо установить в значение **“Нет”**.

#	Параметр	Значение
0	Суммирование фильтра	8
1	Длина фильтра	32
2	Порог отключения фильтра, %	0,10
3	Граница весового диапазона, дискрет	9
4	Авто-восстановление АЦП	Вкл.
5	Код АЦП при минимальной нагрузке	2182088
6	Код АЦП при эталонной нагрузке	14705728
7	Эталонная нагрузка	1000000
8	Дискрета	1
9	НПВ, г	1000000
10	Период стабилизации веса, с	2

В блоке Adc:

Не рекомендуется устанавливать значение "DC – постоянный ток" параметру **Питание датчиков** при использовании тензодатчиков.

**В блоке PSP:****В блоке Clb0:**

2.2 Описание пользовательских параметров прибора M0801

2.2.1 Блок параметров пользователя “Usr”

1. **Скорость обмена.** Скорость обмена в бит/сек по сети RS485. Выбирается из списка в зависимости от длины линии связи, уровня помех и качества кабеля.
2. **Задержка ответа.** Выбирается из списка. Не рекомендуется выбирать значение, меньшее чем “**3 байта**” для высоких скоростей обмена.
3. **Адрес в сети.** Уникальный адрес прибора в сети RS485. Выбирается из списка.
4. **Групповой адрес.** Адрес группы, в которую входит прибор. Используется для выполнения операций над группой приборов. Выбирается из списка.
5. **Адрес следующего в цикле опроса.** Необходим для организации автоматического циклического опроса приборов в сети RS485. Выбирается из списка.
6. **Передача статистики работы.** Добавляет статистику работы прибора в каждый передаваемый прибором пакетом. Рекомендуется установить значение “**Нет**”.

2.2.2 Блок параметров калибровки “С1b”

1. **Суммирование.** Среднее арифметическое из выбранного количества измерений (скользящее среднее). Выбирается из списка.
2. **Нормализация в 16 разрядов.** Результат измерения подготовленных к передаче данных нормализуется в 16 из 24 разрядов.
3. **Статус передавать.** Передача служебной информации с АЦП прибора в пакете. Рекомендуется установить значение “Да”.
4. **Шкала (Ацп/мВв).** Шкала АЦП: передаются данные с АЦП после суммирования и нормализации. Шкала мВ/В (пользовательская шкала): позволяет определить масштаб измерений в размерных единицах. Не рекомендуется выбирать мВ/В до того, как будут установлены правильные непротиворечивые значения для **Код АЦП при минимальной нагрузке**, **Код АЦП при эталонной нагрузке**, **Эталонная нагрузка**, **Дискрета**.
5. **Внутренняя калибровка при включении питания.** Позволяет производить внутреннюю аппаратную калибровку АЦП при включении питания. Рекомендуется установить значение “Нет”.
6. **Режим постоянной передачи после включения.** Переводит прибор в режим постоянной автоматической передачи пакетов данных. Для возможности работы нескольких приборов в сети RS485 необходимо установить значение “Нет”.
7. **Авто-поиск скорости обмена.** Рекомендуется включать на этапе настройки прибора. Для работы прибора в сети RS485 рекомендуется установить значение “Нет”.
8. **Игнорирование FRAME-ERROR.** Игнорирование аппаратной ошибки связи FRAME-ERROR. Рекомендуется установить значение “Да”.
9. **Авто-перезапуск АЦП.** Автоматически восстанавливает калибровку и перезапускает АЦП при сбоях его работы. Рекомендуется установить значение “Да”.
10. **Код АЦП при минимальной нагрузке.** Величина кода АЦП, устанавливаемая пользователем, соответствующая нулю пользовательской шкалы. Используется при работе в режиме “Шкала мВ/В”.
11. **Код АЦП при эталонной нагрузке.** Величина кода АЦП, устанавливаемая пользователем, соответствующая **Эталонной нагрузке**. Используется при работе в режиме “Шкала мВ/В”.
12. **Эталонная нагрузка.** Числовое значение НПВ, устанавливаемое пользователем. Используется при работе в режиме “Шкала мВ/В”.
13. **Дискрета.** Величина дискретности. Значения измеряемой величины округляются до соответствующей дискретности. Используется при работе в режиме “Шкала мВ/В”. Выбирается из списка.

2.2.3 Блок параметров АЦП “Аdc”

1. **Питание датчиков.** Определяет тип напряжения питания датчиков. Выбирается из списка. Рекомендуется устанавливать значение “**АС – переменный ток**”.
2. **Диапазон.** Максимальная амплитуда входного сигнала. Выбирается из списка. Реальная амплитуда входного сигнала определяется как граничные значения диапазона плюс значение смещения.
3. **Частота выборки АЦП.** Частота измерений АЦП, устанавливаемая пользователем. Введенное значение автоматически корректируется до ближайшего допустимого значения частоты АЦП после сохранения параметров.
4. **Смещение.** Смещение нуля АЦП. Выбирается из списка.

2.3 Порядок настройки весовых платформ

1. Запустить программу **MPSDrv.exe** с параметром **-CLB**.
2. Открыть диалог **Настройка параметров** выбрав пункт меню **Порт | Настроить...** и настроить последовательный порт.
3. Подключить к выбранному порту **один** прибор и настроить его (см. раздел **Настройка прибора**)
4. Повторить п.3 для каждого прибора, используемого в весах.
5. Соединить все настроенные приборы в сеть RS485 и подключить к выбранному последовательному порту.
6. Выбрать вкладку **Платформы**.
7. Для настройки новой платформы нажать кнопку **Добавить...**; для настройки существующей платформы нажать кнопку **Настроить...**

Калибровка весовой платформы

Номер платформы: 5

Границы весового диапазона: Нижняя: -1%НПВ, Верхняя: НПВ+9d, Аварийная: НПВ+10%

Название платформы: Новая платформа

НПВ: 30000

Дискретность индикации (d): 10

Позиция десятичной точки: 3

Единицы измерения: кг

Глубина фильтра (кол-во изм.): Нет фильт

Порог отключения фильтра: Без отклк

Период стабилизации: 0.5 с

Авто ноль Ноль при включении

НПВ = 30,000 кг X 0,010 кг

Номера каналов (цифровых датчиков):

№ канала	Коэффициент	Сигнал	Вес брутто
7	1	2139460	0,523

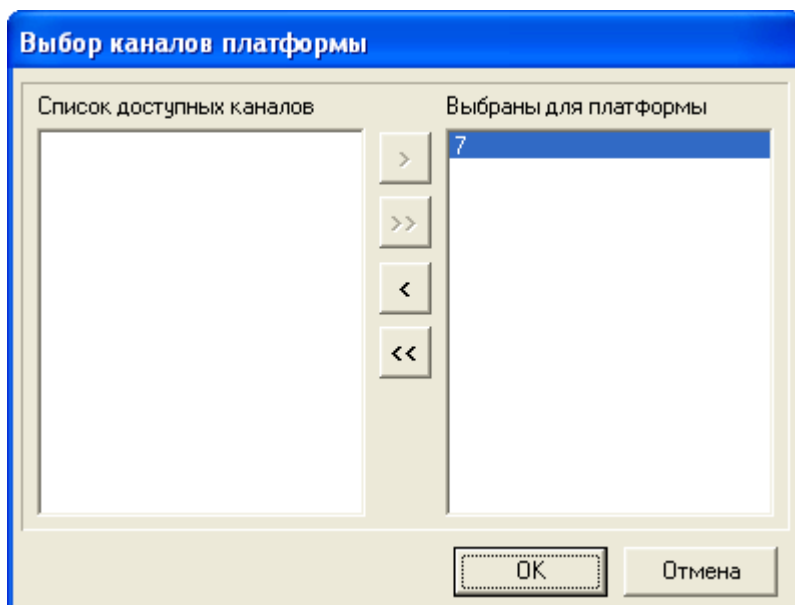
Суммарный сигнал/вес брутто платформы: 2139460 / 0,520 кг

Калибровочные данные платформы

Эталонная нагрузка	Суммарный сигнал
0	2000000
30	10000000

8. В открывшемся диалоге **Калибровка весовой платформы** выбрать номер платформы, ввести ее название, установить наибольший предел взвешивания, дискретность индикации, позицию десятичной точки, единицу измерения и т.д.

9. Нажать кнопку **Выбрать каналы...** После нажатия кнопки программа попытается установить связь со всеми приборами, подключенными к сети RS485 и определить их тип.



В появившемся диалоге **Выбор каналов платформы** выбрать номера каналов (номера приборов), установленных в платформе. Для каждой весовой платформы следует выбрать **уникальный** список номеров каналов, т.е. один и тот же канал не может входить одновременно в списки для разных платформ. Опрос каналов будет производиться в соответствии с заданным порядком их номеров и номером платформы. Порядок каналов задается перетаскиванием мышкой их номеров в списке **Выбраны для платформы**.

10. Произвести калибровку платформы по контрольным точкам (до 4 точек). Для этого необходимо для каждой j -ой контрольной точки ввести значение эталонной нагрузки в колонку **Эталонная нагрузка** (M_j), установить на платформу груз массой, равной значению эталонной нагрузки, дождаться успокоения показаний веса платформы и нажать кнопку **Измерить**. Каждое следующее значение эталонной нагрузки и суммарного сигнала должно быть больше или равно предыдущему ($M_j \leq M_{j+1}$, $Z_j \leq Z_{j+1}$). **Суммарный сигнал** каналов (Z_j) рассчитывается по формуле $Z_j = \sum K_i \cdot Y_i$, где K_i - коэффициент суммирования сигнала i -того канала, Y_i - значение сигнала i -того канала. При необходимости корректировки Z_j можно изменить значения коэффициентов в колонках K_i или непосредственно значение Z_j . Пары значений (M_j , Z_j) будут использоваться для расчета веса по методу кусочно-линейной аппроксимации.
11. Нажать кнопку **ОК** для окончания настройки платформы или кнопку **Отмена** для отказа от внесенных изменений. После нажатия кнопки **ОК** в выбранные цифровые датчики произойдет запись последовательности номеров каналов (открытая цепочка). В каждый выбранный цифровой датчик будет записан адрес следующего по списку канала. В последний цифровой датчик будет записан адрес MASTER-станции (компьютера). При ошибке записи повторить п. 11.
12. Повторить п.п.7-11 для других весовых платформ.

13. Для сохранения настроечных и калибровочных данных весовых платформ выйти из программы, выбрав пункт меню **Порт | Выход** или нажав кнопку **Заккрыть**.

После успешной калибровки весовых платформ можно запустить на исполнение файл **MPSDrv.exe** и проверить точность калибровки платформ, установив на них эталонные грузы и сравнив их значения с показаниями программы.

2.4 Программный интерфейс

2.4.1 Данные, передаваемые на почтовый ящик "DevNet"

Структура данных *TPlatformRec*:

```
TPlatformRec = record
    PlatformNum : Byte;      // номер весовой платформы
    TypeDev     : Byte;      // тип прибора (см. ниже)
    VersionDev  : Byte;      // версия прибора (см. ниже)
    Weight      : Double;    // вес брутто платформы
    ClbZero     : Double;    // вес "пустой" платформы ("рабочий" ноль)
    PntPos      : Byte;      // число знаков после запятой
    ErrState    : Byte;      // код ошибки (см. ниже)
    Flags0      : Byte;      // бит-флаги состояния платформы (см. ниже)
end;
```

Поле TypeDev:

1 – M0801, M0803

Поле VersionDev:

1 – 0.06

2 – 0.09

3 – 0.10

4 – 0.20

Поле ErrState:

Значение	Комментарий
1	Нарушены калибровочные данные весов
20	Вес меньше нижнего предела
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)

Поле Flags0:

fStable = 4 – значения веса стабильны

fNearZero = 8 – значения веса в диапазоне $-0.5d..+0.5d$

fIsMin20d = 16 – значение веса меньше 20d

fUnderLoad = 32 – значение веса существенно меньше нуля

fOverLoad = 64 – значение веса больше НПВ+9d

fNoLoadCell = 128 – ошибка подключения (обрыв, повреждение) тензодатчика

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

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

Команда на порт “**MPSDrvRpc**”:

@<Тело команды>

Ответ на порт “**MPSDrvDrv**”:

#<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.

Команды:

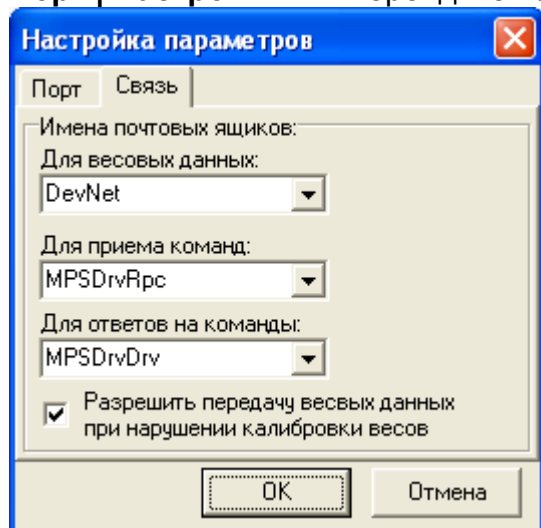
```
@SV {  
    0 |           – Скрыть окно DevNet.  
    1           – Показать окно DevNet.  
}
```

@DXXSZ – установить «рабочий ноль» весов с номером **XX**

@DXXSIYY - вывести на табло с номером **XX** сообщение **YY** (сообщение задается в виде ASCII строки)

2.4.3 Настройка имен почтовых ящиков

Для настройки имен почтовых ящиков выберите пункт меню **Порт | Настроить...** и перейдите на вкладку «**Связь**».



Имена почтовых ящиков можно выбрать из списка или задать произвольное имя.

Флаг «**Разрешить передачу весовых данных при нарушении калибровки весов**». Если драйвер обнаружил нарушение калибровочных данных и при

этом данный флаг **выключен**, то весовые данные на ящик «**DevNet**» передаваться не будут. Если флаг включен, то данные будут передаваться, но поле «**ErrState**» структуры «**TPlatformRec**» **будет равно 1**.

2.4.4 Функции, реализованные в сервере автоматизации “**MPSDrv.Drv**”

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

Все функции (за исключением **TestPort**, **OpenPort**, **ClosePort**, **SetPortDlg**, **SelectPlatformDlg**, **GetPlatformList**, **GetDevList**, **GetVersion**) имеют обязательный параметр **PlatformNum** – номер платформы. При успешном выполнении функции возвращают результат **True**, иначе – **False**.

Таблица 1. Типы данных:

Pascal тип	IDL тип	Комментарий
Byte	unsigned char	1 byte unsigned integer
Double	Double	8-byte real
WordBool	VARIANT_BOOL	True = -1, False = 0
OleVariant	VARIANT	Ole Variant
WideString	BSTR	binary string

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

– Проверить состояние последовательного порта.

Параметры:

PortNum – номер последовательного порта

(1 – Com1, 2 – Com2, 3 – Com3, 4 – Com4),

BaudRate – скорость обмена

(0 – 1200, 1 – 2400, 2 – 4800, 3 – 9600, 4 – 19200, 5 – 38400, 6 – 57600,

7 – 115200),

AutoOpen – режим автоматического открытия порта

(True – вкл., False – выкл.).

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

– Открыть последовательный порт.

Параметры:

PortNum – номер последовательного порта

(1 – Com1, 2 – Com2, 3 – Com3, 4 – Com4),

BaudRate – скорость обмена

(0 – 1200, 1 – 2400, 2 – 4800, 3 – 9600, 4 – 19200, 5 – 38400, 6 – 57600,

7 – 115200),

AutoOpen – режим автоматического открытия порта

(True – вкл., False – выкл.).

function **ClosePort**: WordBool;

– Закрыть последовательный порт.

function **SetPortDlg**: WordBool;

– Вызвать диалог настройки последовательного порта.

function **SelectPlatformDlg**: WordBool;

– Вызвать диалог выбора весовых платформ для работы.

- function **GetPlatformList**(out PlatformList: WideString): WordBool;
– Получить список номеров подключенных платформ в формате:
<Платформа1>;<Платформа2>...
- function **GetStatePlatform**(PlatformNum: Byte; out ErrState, Flags0, Inputs, Outputs: Byte): WordBool;
– Получить бит-флаги состояния платформы PlatformNum.
Параметры:
ErrState, Flags0 – см. выше,
Inputs – бит-маска входов (только для прибора M0803),
Outputs – бит-маска выходов (только для прибора M0803).
- function **GetWeight**(PlatformNum, TypeWeight: Byte; out Weight: Double; out ErrState, Flags0, Inputs, Outputs, Reserve: Byte): WordBool;
– Получить значение веса платформы PlatformNum.
Параметры:
TypeWeight – тип значения веса
(0 – брутто, 3 – “рабочий” ноль),
Weight – значение веса,
ErrState, Flags0, Inputs, Outputs – см. выше,
Reserve – зарезервирован.
- function **SetOutputs**(PlatformNum, Outputs: Byte): WordBool;
– Установить новые состояния выходов прибора платформы PlatformNum.
Параметры:
Outputs – см. выше.
- function **GetVariable**(PlatformNum, TypeVariable: Byte; out Variable: OleVariant): WordBool;
– Получить значение переменной платформы PlatformNum.
Параметры:
TypeVariable – тип значения переменной
(0 – брутто, 3 – “рабочий” ноль, 4 – ErrState, 5 – Flags0,
15 – Inputs, 16 – Outputs),
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 ‘;’
VarList – список значений переменных в формате ASCII строки:
<Значение переменной1>;<Значение переменной2>... (Param = 0)
<Имя переменной1>;<Имя переменной2>... (Param = 1)
- function **GetConstant**(PlatformNum, TypeConstant: Byte; out Constant: OleVariant): WordBool;
– Получить значение константы платформы PlatformNum.
Параметры:
TypeConstant – тип значения константы
(0 – код типа прибора – см. выше (TypeDev),
1 – код версии прибора – см. выше (VersionDev),
2 – максимальное количество весовых диапазонов - 1,
3 – число знаков после запятой,
4..6 – массив НПВ,

7..9 – массив дискрет),
Constant – значение константы.

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

– Проверка готовности платформы PlatformNum к взвешиванию.

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

– Установить новый уровень "рабочего" нуля платформы PlatformNum.

function **ConnectPlatform**(PlatformNum: Byte; Connect: WordBool): WordBool;

– Подключить/отключить платформу PlatformNum.

Параметры:

Connect – подключить(True)/отключить(False) платформу.

function **GetDevList**(out DevList: WideString): WordBool;

– Получить список адресов подключенных к сети RS-485 приборов в формате:
<Адрес1><Тип1><Версия1>;<Адрес2>... – см. выше.

function **GetVersion**: WideString;

– Получить строку с версией драйвера.

property **Visible**: WordBool;

– Показать(True)/скрыть(False) окно MPSDrv.

function **GetVariable1C**(PlatformNum, TypeVariable: Byte): OleVariant;

function **GetVarByList1C**(DevNum: Byte; const List: WideString; Param: Byte): OleVariant;

function **GetConstant1C**(PlatformNum, TypeConstant: Byte): OleVariant;

function **GetPlatformList1C**: OleVariant;

function **GetDevList1C**: OleVariant;

– Аналоги функций GetVariable, GetVarByList, GetConstant, GetPlatformList, GetDevList для системы 1С:Предприятие и языков программирования, не поддерживающих передачу значений по ссылке в параметрах функции. Функции возвращают запрошенный параметр или False – при не успешном выполнении.

Приложение А. Примеры использования программы

В комплект поставки программы входят:

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

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

```
procedure SomeProcedure; // Какая-то процедура
var
  MPSDrv : OleVariant;
begin
  MPSDrv := CreateOleObject('MPSDrv.Drv'); // Установить связь с сервером
                                             // автоматизации “MPSDrv.Drv”
  if not MPSDrv.SetToZero(1) then // Установить на платформе с номером
                                  // 1 новый уровень «рабочего» нуля
    ShowMessage('Невозможно установить ноль на выбранной платформе.'#13#10+
      'Возможные причины: показания веса нестабильны, '#13#10+
      'платформа не выбрана для работы, нет связи с приборами платформы');
  MPSDrv := UnAssigned; // Дальнейшей работы с сервером
                        // автоматизации “MPSDrv.Drv” не будет
end;
```

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

```

Dim objMPSDrv
Private Sub SomeProcedure()
    Set objMPSDrv = CreateObject("MPSDrv.Drv")
    If Not objMPSDrv.SetToZero(1) Then
        MsgBox ("Невозможно установить ноль на выбранной платформе." + _
            "Возможные причины: показания веса нестабильны, платформа не выбрана" + _
            "для работы, нет связи с приборами платформы")
    End If
    Set objMPSDrv = Nothing
End Sub

```

‘ Какая-то процедура
‘ Установить связь с сервером
‘ автоматизации “MPSDrv.Drv”
‘ Установить на платформе с номером 1
‘ новый уровень «рабочего» нуля
‘ Дальнейшей работы с сервером
‘ автоматизации “MPSDrv.Drv” не будет

А.3 Пример использования сервера автоматизации “MPSDrv.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 MPSDrv Экспорт;

А в процедуре **ПриНачалеРаботыСистемы** создать объект:

MPSDrv = СоздатьОбъект("MPSDrv.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(PlatformNum, TypeWeight)

Вес = MPSDrv.GetVariable1C(PlatformNum, TypeWeight)

КонецПроцедуры.