



v.6.0.

«Штрих-М: Драйвер электронных весов»

**Руководство программиста
(Версия 1)**

НТЦ «Штрих-М», 2006

Данное руководство программиста является описанием продукта «Штрих-М : Драйвер электронных весов» 6.0 , разработанного НТЦ «Штрих-М». При описании подразумевалось, что читатель имеет навыки программирования на одном или нескольких языках программирования для операционных систем: Windows NT, Windows 98 или Windows 95, а также знаком с используемым оборудованием (на уровне руководства оператора из его комплекта поставки).

**ПРАВО ТИРАЖИРОВАНИЯ
ПРОГРАММНЫХ СРЕДСТВ И ДОКУМЕНТАЦИИ
ПРИНАДЛЕЖИТ НТЦ «ШТРИХ-М»**

ОГЛАВЛЕНИЕ

Оглавление.....	3
Введение	6
Сокращения.....	6
Об электронных весах.....	6
Описание драйвера.....	7
Стратегия защиты драйверов	Ошибка! Закладка не определена.
Техническая поддержка пользователей	Ошибка! Закладка не определена.
Начало работы.....	8
Установка и регистрация драйвера.....	Ошибка! Закладка не определена.
Автоматическая установка.....	Ошибка! Закладка не определена.
Самостоятельная (вручную) установка	Ошибка! Закладка не определена.
Полезные советы.....	Ошибка! Закладка не определена.
Подключение драйвера	8
Приступая к работе	9
Поиск подключенного оборудования.....	9
Тестовая утилита Scale_T.exe.....	10
Назначение и возможности утилиты	10
Запуск утилиты	10
Работа с утилитой	11
Окно «О программе»	13
Окно для работы с ЛУ	14
Настройка параметров оборудования.....	15
Описание драйвера	16
Определения.....	16
О свойствах драйвера.....	17
Типы данных.....	17
Методы драйвера.....	18
Методы работы с логическими устройствами	20
AddDevice ДобавитьУстройство	20
DeleteDevice УдалитьУстройство	20
Методы работы с весами	20
ReadWeight ПолучитьВес.....	20
ZeroScale Ноль.....	21
SetTareWeight УстановитьТару	21
Tare Тара	21
Reset Сброс	21
Методы программирования весов	22
SetValue УстановитьПараметр	22
GetValue ПолучитьПараметр.....	22
Прочие методы	22
GetDeviceMetrics ПолучитьПараметрыУстройства	22
ShowProperties	23

DeleteEvent УдалитьСообщение	23
Свойства драйвера	24
Свойства общего назначения.....	26
ResultCode Результат	26
ResultDescription ОписаниеРезультата.....	27
Version Версия	27
Свойства для работы с ЛУ	27
DeviceCount КоличествоУстройств.....	27
CurrentDeviceIndex ИндексТекущегоУстройства	28
CurrentDeviceNumber НомерТекущегоУстройства	28
CurrentDeviceName НаименованиеТекущегоУстройства	28
LockDevices БлокироватьУстройства	29
Свойства ЛУ, параметры связи с весами.....	29
PortNumber НомерПорта	29
BaudRate СкоростьОбмена	30
Parity Четность.....	30
DeviceEnabled УстройствоВключено	30
Model Модель	31
Параметры, получаемые от весов	31
Weight Bec	31
TareWeight ВесТары.....	31
UpdatePrice ОбновлятьЦену	32
UnitPrice ЦенаЗаЕдиницу	32
SalesPrice ОбщаяЦена	32
NettoWeight ВесНетто.....	32
AutoZeroMode РежимАвтонуля.....	33
OutOfZero НольВнеДиапазонаНуля.....	33
NegWeight ОтрицательныйВес	33
BigWeight БольшойВес.....	33
NonStable ВесНеСтабилен.....	34
InvalidRange НеверныйДиапазонВеса.....	34
Свойства асинхронного режима.....	34
AsyncMode АсинхронныйРежим.....	34
DataEventEnabled ПосылкаДанных	34
AutoDisable АвтоВыключение.....	35
DataCount ЧислоДанныхВОчереди	35
EventNumber НомерСообщения	35
PollingInterval ПериодОпроса	36
WeightStep ШагВеса.....	36
Прочие свойства.....	36
DeviceDescription ОписаниеУстройства	36
ValuePurpose НазначениеПараметра	37
Value Параметр.....	37
UCodePage УКодоваяСтраница	37
UDescription УОписание.....	38
UMajorVersion УСтаршаяВерсия	38
UMinorVersion УМладшаяВерсия	38
UMode УРежим	39
UModel УМодель.....	39

Оглавление.

UProtocolVersion УВерсияПротокола.....	39
UType УТип.....	40
ApplicationHandle	40
Асинхронный режим	41
Введение	41
Прием событий от драйвера.....	43
Система программ 1С:ПРЕДПРИЯТИЕ v.7.x.....	43
Borland Delphi 3 и Delphi 4.....	43

ВВЕДЕНИЕ

Сокращения

При описании драйвера использовались сокращения:

ККМ	Контрольно-кассовая машина.
POS	Point of sale (рабочее место кассира).
ПК	Персональный компьютер.
ОС	Операционная система.
ПО	Программное обеспечение.
ВК	Внешняя компонента для системы программ «1С:ПРЕДПРИЯТИЕ» v. 7.5 и 7.7.
ЛУ	Логическое устройство.
АРМК	Автоматизированное рабочее место кассира

Об электронных весах

Весы электронные, о которых далее пойдет речь, должны иметь возможность связи с ПК (по интерфейсу RS-232).

Такие весы (далее возможно использование синонима «оборудование»), обычно, используются при создании АРМК на базе ПК (или POS-терминала). Подключенные к ПК весы служат для определения веса отпускаемого товара. Такие АРМК позволяют повысить (по сравнению с традиционным комплексом «ККМ и весы отдельно») производительность работы кассира. Продавцу не приходится:

- взвешивать товар;
- умножать массу на цену за 1 кг;
- вводить значение стоимости в ККМ (зарегистрировать продажу).

Подключенные к ПК весы и ККМ (в режиме «Пассивный On-Line») позволяют сократить последовательность действий кассира:

- поместить товар на чашу весов;
- указать код товара (сканировать штрих-код сканером, ввести PLU или выбрать из списка).

ПК самостоятельно получит вес товара, вычислит стоимость и напечатает чек на ККМ.

Примечание: в описанном комплексе можно заменить «ПК» на «POS-терминал», «ККМ в режиме «Пассивный On-Line» на «фискальный регистратор».

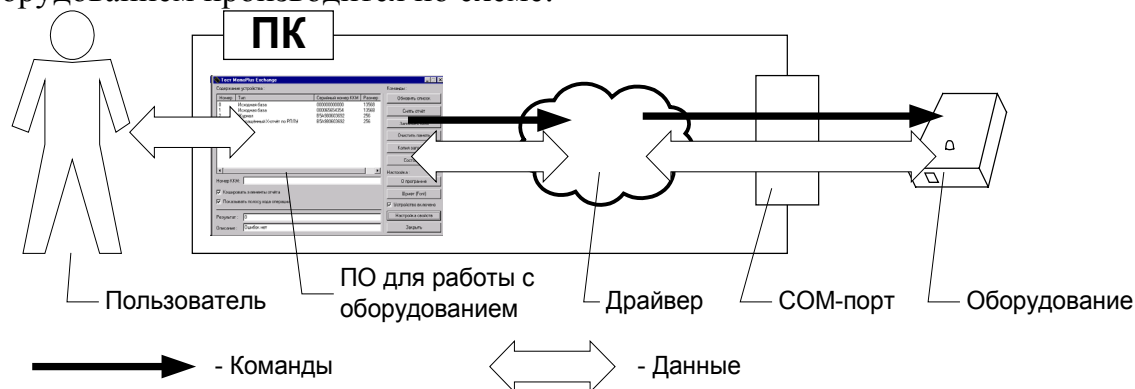
Мы описали наиболее часто встречающийся способ использования весов, подключенных к ПК, но область применения этого тандема значительно обширнее.

Описание драйвера

Описываемый ниже драйвер позволяет работать с весами марок:

- Штрих МI;
- Штрих МII;
- Штрих МIII;
- Штрих М5;
- Весовая ячейка BM100;
- Штрих ВП.

Вся работа оборудования с ПК производится под управлением программы, выполняемой на ПК. В комплект поставки входит дискета с ПО, позволяющим работать с оборудованием. Вся работа пользователя с оборудованием производится по схеме:



В данном разделе будет описан элемент схемы «Драйвер». Данное описание предназначено, в первую очередь, программистам, создающим собственное ПО для работы с оборудованием.

Внутренняя организация драйвера такова, что позволяет использовать его как Server Automation или внешнюю компоненту для системы программ 1С:Предприятие версий 7.5 и 7.7.

Ниже рассматривается интерфейс «ПО для работы с оборудованием – Драйвер». Интерфейсы «Драйвер – Оборудование» не описываются.

НАЧАЛО РАБОТЫ

Подключение драйвера

В разных языках программирования используется различный синтаксис, мы приведем здесь только пример для «1С:ТОРГОВЛЯ И СКЛАД» v. 7.7. В комплект поставки так же входит пример для Borland Delphi 5.0

Имя объекта драйвера – «AddIn.Scale45».

Пример глобального модуля конфигурации 1С:ПРЕДПРИЯТИЕ:

```
Перем Scl Экспорт; // Глоб.перем. для работы с драйвером
Перем Результ; // Служебная переменная
. . . // Раздел описания глобальных процедур
    //Загрузка внешней компоненты
    Результ = ЗагрузитьВнешнююКомпоненту ("Scale1C.dll");

    // Инициализация глоб. переменной, если компонента найдена
    Если Результ = 1 Тогда Scl = СоздатьОбъект ("AddIn.Scale45");
        Иначе Сообщить ("Не найдена внешняя компонента.");
    КонецЕсли;

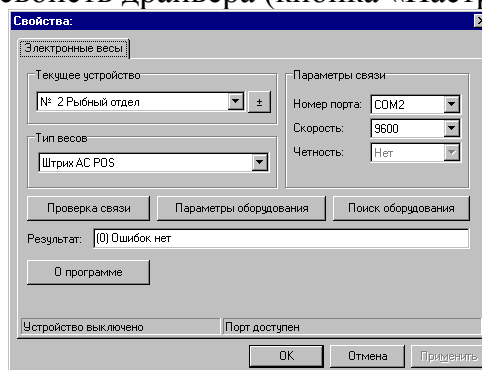
    //После этого возможен доступ к свойствам и методам драйвера
    // во всех модулях :
    // Вызов метода драйвера
    Если Scl.AddDevice () < 0
        // Обращение к свойству
        Тогда Сообщить ("Ошибка:" + Строка(Scl.ResultDescription));
    КонецЕсли;
```


Приступая к работе

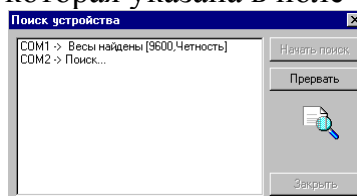
Поиск подключенного оборудования

Для начала работы необходимо настроить драйвер. Это легко сделать с помощью тестовой утилиты, поставляемой вместе с драйвером:

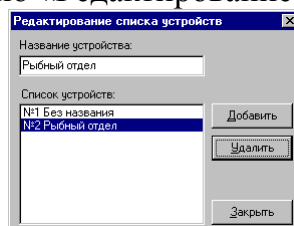
1. Подключите весы к ПК.
2. Запустите тестовую утилиту.
3. Откройте страницу свойств драйвера (кнопка «Настройка свойств»).



4. Установите используемый тип весов в поле «Тип весов».
5. Нажмите кнопку «Поиск оборудования».
6. Произведите поиск всех подключенных к ПК весов (поиск ведется только весов той марки, которая указана в поле «Тип весов»).



7. Создайте структуру ЛУ так, чтобы Вам было удобно с ней работать:
 - кнопка  открывает окно «Редактирование списка устройств»:



- текущее ЛУ выделяется цветом (подсвечивается);
 - «Добавить» – добавить новое ЛУ, «Удалить» – удалить текущее ЛУ;
 - изменить имя текущего ЛУ можно в поле «Название устройства» (задавайте ЛУ «говорящие» имена).
8. Настройте каждое из ЛУ на требуемые «Тип весов», «Порт» и параметры связи.
 9. Проверить корректность настройки любого ЛУ можно последовательностью действий:
 - сделайте текущим ЛУ, настроенное на данные весы;

- нажмите кнопку «Проверка связи»;
- в поле «Результат» должен быть выведена масса, установленная на чаше весов.

В процессе работы для «переключения» драйвера с одних весов на другие просто выбирайте соответствующее ЛУ (на странице свойств драйвера поле «Текущее устройство»). Если установлено одно ЛУ, то это поле заблокировано.

Для работы утилиты с оборудованием надо занять порт – установить флажок (галочка есть) «Устройство включено». Если устройство не включено, то работа с оборудованием вестись не будет.

Примечание: если флажок установить не удастся, проверьте, существует ли порт с таким номером и не занят ли он другим ПО, например драйвером «мыши».

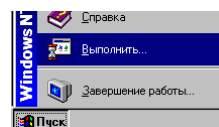
Тестовая утилита Scale_T.exe

Назначение и возможности утилиты

1. Зарегистрировать драйвер.
2. Настроить драйвер.
3. Познакомить пользователя с приемами работы с оборудованием.
4. Познакомить разработчиков с принципами использования драйвера.

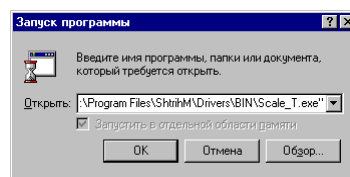
Запуск утилиты

Выберите пункт «Выполнить...» из меню «Пуск» («Start» / «Run...»)



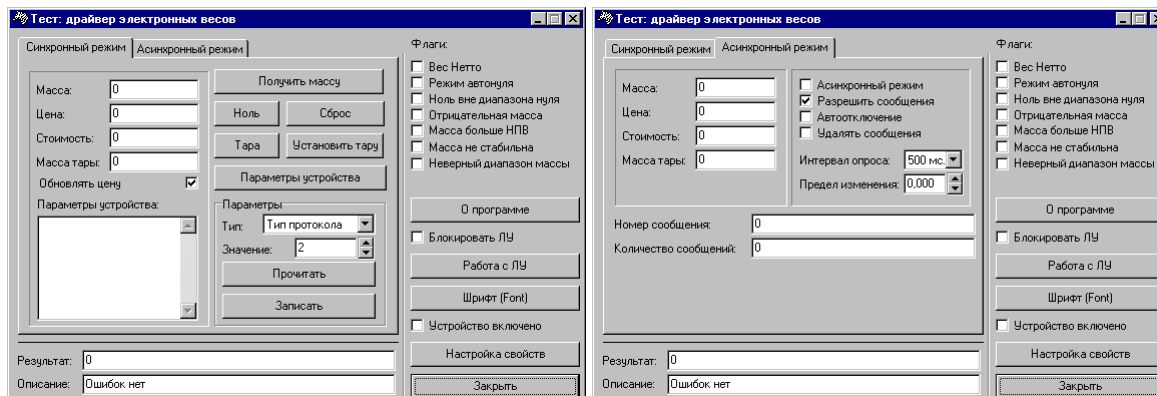
Наберите «<путь>\SCALE_T.EXE» и нажмите клавишу Enter.

Воспользуйтесь клавишей «Обзор» («Browse...») для более простого указания файла.



Работа с утилитой

После запуска на экране должно появиться главное окно утилиты (справа показана вторая закладка тестовой утилиты):



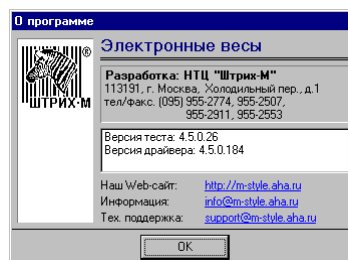
Элемент	Функция
Кнопка «Получить массу»	Прочитать массу, массу тары и статус из весов (см. метод ReadWeight на стр. 20).
Флаг «Обновлять цену»	Если «галочка» есть, то по нажатию кнопки «Получить массу» в весы буде передана цена товара за 1 кг (из поля «Цена»), иначе цена передаваться в весы не будет (будет считано текущее значение цены из весов). Допустимо для «Штрих AC POS» и «Штрих AC».
Кнопка «Ноль»	Установить ноль на весах. Допустимо для «Штрих POS2» и «Штрих AC POS».
Кнопка «Сброс»	Перезапустить весы (весы перейдут в состояние, аналогичное состоянию после включения питания). Допустимо для «Штрих AC POS» и «Штрих POS2».
Кнопка «Тара»	Установить массу тары равной той, что установлена в текущий момент на чаше весов. Допустимо для «Штрих POS2» и «Штрих AC POS».
Кнопка «Установить тару»	Установить массу тары равной той, что задана в поле «Масса тары». Допустимо для «Штрих POS2» и «Штрих AC POS».
Кнопка «Параметры устройства»	Прочитать параметры весов и вывести их в поле «Данные». Допустимо для «Штрих POS2» и «Штрих AC POS».
Кнопка «Прочитать»	Получить параметр (см. поле «Тип параметра») из весов и занести полученное значение в поле «Значение параметра». Допустимо для «Штрих AC POS».
Кнопка «Записать»	Записать в параметр весов (см. поле «Тип параметра») значение из поля «Значение параметра». Допустимо для «Штрих POS2» и «Штрих AC POS».

Продолжение таблицы.

Элемент	Функция
Флаг «Вес Нетто»	Если «галочка» есть, то значение поля «Масса» – нетто, иначе – брутто. Обновляется по нажатию кнопки «Получить массу». Допустимо для «Штрих POS2» и «Штрих AC POS».
Флаг «Режим автонуля»	Если «галочка» есть, то на чаше весов груза нет. Обновляется по нажатию кнопки «Получить массу». Допустимо для «Штрих POS2» и «Штрих AC POS».
Флаг «Ноль вне диапазона нуля»	Если «галочка» есть, то весы либо сломаны, либо перешли в нерабочее состояние – выключите и включите весы. Обновляется по нажатию кнопки «Получить массу». Допустимо для «Штрих POS2» и «Штрих AC POS».
Флаг «Отрицательная масса»	Если «галочка» есть, то на весах показания индикатора «вес» содержит отрицательное значение, а поле «Масса» отображает это значение по модулю. Обновляется по нажатию кнопки «Получить массу». Допустимо для «Штрих POS2» и «Штрих AC POS».
Флаг «Масса больше НПВ»	Если «галочка» есть, то на чаше весов находится груз с массой, превышающей «Наибольший Предел Взвешивания» данных весов. Обновляется по нажатию кнопки «Получить массу». Допустимо для «Штрих POS2» и «Штрих AC POS».
Флаг «Масса не стабильна»	Если «галочка» есть, то весы не могут определить взвешиваемую массу, так как она нестабильна. Допустимо для «Штрих POS2» и «Штрих AC POS».
Флаг «Неверный диапазон массы»	Если «галочка» есть, то на чаше весов находится груз с массой, превышающей «Наибольший Предел Взвешивания» или меньший (по модулю), чем «Наименьшего Предела Взвешивания» весов. Допустимо для «Штрих POS2» и «Штрих AC POS».
Флаг «Асинхронный режим»	Если «галочка» есть, то драйвер переходит в асинхронный режим – драйвер самостоятельно оповещает клиента об изменении массы на чаше весов. Опрос весов производится с периодичностью, заданной в «Интервал опроса», и при изменении более, чем на «Предел измерения». См. свойство DataEventEnabled на стр. 34. Допустимо для «Штрих AC POS».
Флаг «Разрешить сообщения»	Если «галочка» есть, то драйвер будет посылать асинхронные сообщения тестовой утилите. См. свойство DataEventEnabled на стр. 34. Допустимо для «Штрих AC POS».

Начало работы.**Продолжение таблицы.**

Элемент	Функция
Флаг «Автоотключение»	Если «галочка» есть, то драйвер будет автоматически скидывать флаг «Разрешить сообщения», посылая асинхронное сообщение тестовой утилите. См. свойство AutoDisable на стр. 35. Допустимо для «Штрих АС POS».
Флаг «Удалять сообщения»	Если «галочка» есть, то тестовая утилита, получив асинхронное сообщение от драйвера будет удалять его. См. метод DeleteEvent на стр. 23. Допустимо для «Штрих АС POS».
Поле «Интервал опроса»	Интервал опроса весов в асинхронном режиме (мс). Допустимо для «Штрих АС POS».
Поле «Предел изменения»	Минимальное значение изменения показаний весов, при превышении которого возникает событие «Изменение веса». Используется в асинхронном режиме. Допустимо для «Штрих АС POS».
Кнопка «О программе»	Открывает окно, в котором можно просмотреть версии тестовой утилиты и драйвера, адреса электронной почты разработчиков и WWW фирмы «Штрих-М».
Флаг «Блокировать ЛУ»	см. свойство LockDevices на стр. 29.
Кнопка «Работа с ЛУ»	Открывает окно для теста работы с ЛУ. Подробнее см. ниже.
Кнопка «Шрифт (Font)»	Кнопка позволяет настроить шрифт, используемый тестовой утилитой и драйвером.
Флаг «Устройство включено»	Включить текущее ЛУ (см. свойство DeviceEnabled на стр. 30).
Кнопка «Настройка свойств»	Открытие визуальной страницы свойств драйвера (см. метод ShowProperties на стр. 22).
Кнопка «Заккрыть»	Заккрытие утилиты.
Поля «Результат» и «Описание результата»	В эти поля выводятся сообщения об ошибках.

Окно «О программе»

Если в системе установлен браузер интернета, то щелкните по <http://www.shtrih-m.ru> для открытия www страницы «Штрих-М». Если в

Начало работы.

системе установлена система для работы с электронной почтой, то щелкните по support@shtrih-m.ru для создания письма в службу технической поддержки и info@shtrih-m.ru для создания письма в службу информации «Штрих-М».

Окно для работы с ЛУ

Окно открывается по нажатию кнопки «Работа с ЛУ» на главной форме утилиты.

Тест свойств устройств

Список логических устройств:

Индекс: 0 Общее кол-во: 1
 Номер: 1

Команды:
 Добавить
 Удалить

Параметры активного устройства:

Название: Без названия
 Номер порта: COM 1
 Скорость: 9600
 Четность: Нет
 Модель: Штрих AC POS

☐ Устройство включено

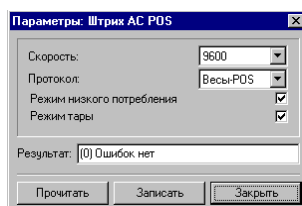
Результат: 0
 Описание: Ошибка нет

Закреть

Элемент	Функция
Кнопка «Добавить»	Добавляет одно ЛУ (см. метод AddDevice на стр. 20).
Кнопка «Удалить»	Удаляет текущее ЛУ (см. метод DeleteDevice на стр. 20).
Кнопка «Закреть»	Закрывает окно работы с ЛУ и возвращает управление главной форме утилиты.
Поля «Индекс», «Номер» и «Общее кол-во»	Эти поля позволяют проводить переключение ЛУ (по индексу и номеру). См. свойства CurrentDeviceIndex (стр. 28), CurrentDeviceNumber (стр. 28) и DeviceCount (стр. 27).
Поля «Название», «Номер порта», «Скорость», «Четность», «Модель» и «Устройство включено»	Эти поля отображают свойства ЛУ. См. свойства CurrentDeviceName (стр. 28), PortNumber (стр. 29), BaudRate (стр. 30), Parity (стр. 30), Model (стр. 31) и DeviceEnabled (стр. 30).
Поля «Результат» и «Описание результата»	В эти поля выводятся сообщения об ошибках.

Настройка параметров оборудования

Драйвер позволяет визуально настроить параметры весов марки «Штрих АС POS». Для других марок данная возможность отсутствует (кнопка «Параметры оборудования» на странице свойств недоступна). При нажатии кнопки «Параметры оборудования» на странице свойств открывается окно:



Примечание: перед открытием окна драйвер считывает настройки из весов. Если прочитать данные не удалось, то окно не открывается.

Элемент	Функция
Поле «Скорость»	Скорость обмена весов с компьютером. После удачной записи параметров в весы драйвер изменяет значение свойства BaudRate (см. стр. 30), чтобы связь не прерывалась.
Поле «Протокол»	Тип протокола, используемый весами для обмена. Данный драйвер поддерживает обмен только по протоколу «Штрих АС POS».
Флаг «Режим низкого потребления»	Если «галочка» есть, то весы будут переходить в режим низкого потребления энергии (Sleep-режим) через несколько минут бездействия.
Флаг «Режим тары»	Если «галочка» есть, то драйвер позволяет устанавливать новое значение тары, не сбрасывая текущего.
Поле «Результат»	В поле выводятся сообщения об ошибках.
Кнопка «Прочитать»	Прочитать текущие настройки из весов.
Кнопка «Записать»	Записать текущие настройки в весы.
Кнопка «Закрыть»	Закрыть окно настроек. При закрытии окна автоматическая запись параметров в весы не производится.

ОПИСАНИЕ ДРАЙВЕРА

Определения

Логическое устройство – набор свойств драйвера, имеющий уникальный номер и необязательное имя. Подобных наборов (устройств) одновременно может быть от 1 до 99 штук. Это позволяет, однажды настроив несколько наборов свойств, быстро применять необходимые параметры просто переключая устройства. Номер устройству присваивается автоматически при его создании и изменяться не может. Имя устройства доступно для изменения в любой момент времени.

Текущее устройство – то устройство, свойства которого доступны в текущий момент для чтения и редактирования. Все методы драйвера работают со свойствами именно этого устройства. Чтобы изменить свойства другого устройства, его необходимо предварительно сделать текущим.

Индекс логического устройства – порядковый номер устройства в списке существующих логических устройств драйвера. То есть при создании нового логического устройства ему присваивается минимальный свободный номер, а индексы пересчитываются так, чтобы номера шли по порядку. При удалении логического устройства из «середины» списка устройств номера остальных устройств не меняются, а индексы изменяются таким образом, чтобы опять получился непрерывный ряд значений.

Например, существовали три логических устройства с номерами: 1, 2 и 3 с индексами 0, 1 и 2 соответственно; после удаления логического устройства с номером 2 появится «дыра», т.е. можно сделать текущим логическое устройство с номером 1 или 3, но не 2. Однако индекс устройства № 3 изменился: был «2», а стал «1». При создании нового логического устройства, ему будет присвоен номер 2 и индекс 1 (у устройства №3 индекс поменяется с 1 на 2). Такие «провалы в номерах» создают определенные проблемы при работе с устройствами, по этой причине и было введено понятие **индекс логического устройства**.

О свойствах драйвера

Все данные драйверу передаются через его свойства. Это означает, что всю информацию для выполнения действий драйвер извлекает из соответствующих свойств, предварительно заполненных пользователем.

Типы данных

В данной главе для описания данных используются типы:

- Целое – целое 32-битное число со знаком; диапазон значений: от -2147483648 до 2147483647
- Денежное* – 64-битное число с 4 дробными разрядами; диапазон: от -922337203685477,5808 до 922337203685477,5807
- Дробное* – знаковое дробное 64-битное число из диапазона: от $5,0 \times 10^{-324}$ до $1,7 \times 10^{+308}$ (точность 15÷16 знаков после дес. запятой)
- Строка – строка символов
- Логическое – целое число, интерпретируемое как «ЛОЖЬ (FALSE)» при значении 0 и «ИСТИНА (TRUE)» в остальных случаях.

* – для дробных величин допускается погрешность менее 0,0001 от указанной точности. Значения с недопустимой погрешностью (отклонение более 0,0001 от допустимой точности) рассматриваются как «недопустимое значение» (ошибка -6). Допустимая точность описана для каждого из свойств отдельно.

Пример: свойство **Price** имеет точность 0,01 руб. Погрешность для **Price** $0,01 \times 0,0001 = 0,000001$ руб. Значение 5,670001 считается недопустимым, а 5,6700009999999 – допустимо.

Методы драйвера

Все методы драйвера являются функциями, то есть возвращают значение – код ошибки, возникшей при выполнении данного метода (код ошибки также заносится в свойство **ResultCode**, а ее описание – в **ResultDescription**).

Методы не требуют параметров, но не забывайте указывать пустые скобки после имени функций (если этого требует синтаксис языка программирования).

Англ. название	Русское название	Использует	Модифицирует*	Стр.
AddDevice	Добавить Устройство		DeviceCount CurrentDeviceIndex CurrentDeviceNumber CurrentDeviceName Все свойства с признаком «L»	20
DeleteDevice	Удалить Устройство		DeviceCount CurrentDeviceIndex CurrentDeviceNumber CurrentDeviceName Все свойства с признаком «L»	20
DeleteEvent	Удалить Сообщение	EventNumber	EventNumber, DataCount, Weight, TareWeight, UnitPrice, SalesPrice, NettoWeight, AutoZeroMode, OutOfZero, NegWeight, BigWeight, NonStable, InvalidRange, CurrentDeviceIndex, CurrentDeviceNumber, CurrentDeviceName Все свойства с признаком «L»	23
GetDeviceMetrics	Получить Параметры-Устройства		UProtocolVersion UType UModel UMode UMajorVersion UMinorVersion UCodePage UDescription	22
GetValue	Получить Параметр	ValuePurpose	Value	22

Описание драйвера (синхронный режим).**Продолжение таблицы.**

Англ. название	Русское название	Использует	Модифицирует*	Стр.
ReadWeight	ПолучитьВес	UnitPrice, UpdatePrice	Weight, TareWeight, UnitPrice, SalesPrice, NettoWeight, AutoZeroMode, OutOfZero, NegWeight, BigWeight, NonStable, InvalidRange	20
Reset	Сброс			21
SetTareWeight	УстановитьТару	TareWeight	Weight, TareWeight, UnitPrice, SalesPrice, NettoWeight, AutoZeroMode, OutOfZero, NegWeight, BigWeight, NonStable, InvalidRange	21
SetValue	УстановитьПараметр	ValuePurpose, Value	BaudRate	22
ShowProperties		LockDevices		22
Tare	Тара		Weight, TareWeight, UnitPrice, SalesPrice, NettoWeight, AutoZeroMode, OutOfZero, NegWeight, BigWeight, NonStable, InvalidRange	21
ZeroScale	Ноль		Weight, TareWeight, UnitPrice, SalesPrice, NettoWeight, AutoZeroMode, OutOfZero, NegWeight, BigWeight, NonStable, InvalidRange	21

* – В результате выполнения метода обновляются значения указанных свойств. Как уже говорилось, каждый метод так же заносит код ошибки в свойство **ResultCode**, ее описание в **ResultDescription**.

Методы работы с логическими устройствами

AddDevice

Добавить Устройство

Метод для создания нового логического устройства (номер и индекс присваиваются автоматически, а имя нового устройства устанавливается равным «Без названия»).

Выполняемые действия:

1. Если **DeviceCount** = 99, то возвращается -7 и больше ничего не делать.
2. Создается новое ЛУ (все свойства, имеющие тип свойства «L» принимают значения по умолчанию).
3. **DeviceCount** увеличивается на единицу.
4. Новому ЛУ присваивается минимальный из незанятых номеров.
5. Пересчитываются индексы:
 - ЛУ сортируются по возрастанию;
 - ЛУ с минимальным номером присваивается индекс 0, следующему - 1,..., ЛУ с максимальным номером – индекс (**DeviceCount** – 1).
6. Текущим делается созданное ЛУ.

См. также: свойства **CurrentDeviceNumber** , **CurrentDeviceIndex** , **DeviceCount** и **CurrentDeviceName**

DeleteDevice Удалить Устройство

Метод предназначен для удаления текущего логического устройства. Необходимо помнить, что нельзя удалить все устройства (должно оставаться хотя бы одно), поэтому при попытке удалить последнее логическое устройство метод возвращает -8. Будьте внимательны – отменить удаление невозможно.

При **AsyncMode** = TRUE при удалении ЛУ из очереди событий удаляются все сообщения, пришедшие от удаляемого ЛУ.

См. также: свойства **CurrentDeviceNumber** , **CurrentDeviceIndex** и **DeviceCount**

Методы работы с весами

ReadWeight

Получить Вес

Служит для чтения показаний весов (масса (**Weight**), цена за 1 кг (**UnitPrice**), стоимость (**SalePrice**) и масса тары (**TareWeight**)).

Для различных моделей весов драйвер передает или нет **UnitPrice**:

Описание драйвера (синхронный режим).

Model	В весы передается содержимое свойства UnitPrice
11	Нет
3 и 5	Да при UpdatePrice = TRUE

Если **UnitPrice** передается, то в весах устанавливается новая цена за 1 кг и весы вычисляют новую стоимость, иначе при вычислении стоимости используется текущая цена, установленная в весах.

При **Model** = 5 или 11 обновляются флаги **NettoWeight**, **AutoZeroMode**, **OutOfZero**, **NegWeight**, **BigWeight**, **NonStable**, **InvalidRange**.

См. также: свойства **Model**, **UpdatePrice**, **AsyncMode** и методы **SetTareWeight**, **Tare**, **ZeroScale**

ZeroScale**Ноль**

Служит для установки нуля на весах.

Метод допустим только при **Model** = 5 или 11.

При **Model** = 5 обновляются флаги **NettoWeight**, **AutoZeroMode**, **OutOfZero**, **NegWeight**, **BigWeight**, **NonStable**, **InvalidRange**.

См. также: свойство **Model** и методы **ReadWeight**, **Tare**, **SetTareWeight**

SetTareWeight**Установить Тару**

Служит для задания значения «Масса тары» в весах. Устанавливаемая масса задается в свойстве **TareWeight**.

Метод допустим только при **Model** = 5 или 11.

При **Model** = 5 обновляются флаги **NettoWeight**, **AutoZeroMode**, **OutOfZero**, **NegWeight**, **BigWeight**, **NonStable**, **InvalidRange**.

См. также: свойство **Model** и методы **ReadWeight**, **Tare**, **ZeroScale**

Tare**Тара**

Служит для установления значения «Масса тары» в весах равной массе груза установленного в данный момент на чаше весов.

Метод допустим при любом значении **Model**.

При **Model** = 5 обновляются флаги **NettoWeight**, **AutoZeroMode**, **OutOfZero**, **NegWeight**, **BigWeight**, **NonStable**, **InvalidRange**.

См. также: свойство **Model** и методы **ReadWeight**, **SetTareWeight**, **ZeroScale**

Reset**Сброс**

Метод перезапускает весы (весы перейдут в состояние, аналогичное состоянию после включения питания).

Метод допустим только при **Model** = 5 или 11.

Весы «Штрих ВП» после выполнения метода выключаются.

См. также: свойство **Model**

Методы программирования весов

SetValue

УстановитьПараметр

Метод используется для программирования системных настроек весов – значение содержится в свойстве **Value**, номер настройки – в свойстве **ValuePurpose**.

Метод может вызываться только при **Model** = 5 или 11.

См. также: свойства **Value**, **ValuePurpose** и метод **GetValue**

GetValue

ПолучитьПараметр

Метод используется для получения значений системных настроек весов. Номер настройки содержится в свойстве **ValuePurpose**, в случае успешного завершения значение настройки сохраняется в свойстве **Value**.

Метод допустим только при **Model** = 5.

См. также: свойства **Value**, **ValuePurpose** и метод **SetValue**

Прочие методы

GetDeviceMetrics

ПолучитьПараметрыУстройства

Универсальная команда запроса типа устройства. Заполняет свойства драйвера:

Свойство	Описание
UProtocolVersion	Версия формата универсальной команды
UType	Тип оборудования
UModel	Модель оборудования
UMode	Поддерживаемые оборудованием режимы
UMajorVersion	Версия оборудования
UMinorVerion	Подверсия оборудования
UCodePage	Используемая оборудованием кодовая страница
UDescription	Описание оборудования

Фирма «Штрих-М» вводит для всех устройств, разрабатываемых ею, единую систему идентификации устройств. В рамках данного проекта принято соглашение о наличии данной команды у всех новых продуктов «Штрих-М».

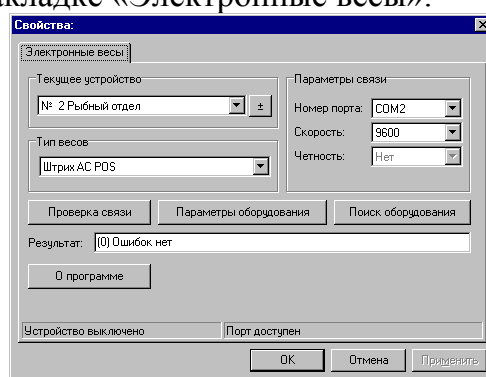
Метод допустим только при **Model** = 5 или 11.

Описание драйвера (синхронный режим).**ShowProperties**

Метод выводит на экран визуальную страницу свойств, через которую можно:

- добавлять, изменять и удалять ЛУ;
- настраивать параметры связи с оборудованием;
- проверить связь с оборудованием;
- произвести поиск COM-порта, к которому подключено оборудование;
- настроить параметры оборудования.

Данный метод не может вызываться, если драйвер используется как внешняя компонента для программ системы «1С:ПРЕДПРИЯТИЕ». В этом случае страницу свойств можно вывести, выбрав пункт меню «Настройка / Параметры...», на закладке «Электронные весы».



По нажатию на странице свойств кнопки «Проверка связи»:

1. У весов запрашивается масса (если для текущего ЛУ **DeviceEnabled** = FALSE, то драйвер временно занимает порт).
2. Если операцию выполнить не удалось, то в поле «Результат:» выдается описание ошибки и более ничего не делается.
3. Если масса успешно получена, то в поле «Результат:» выдается «Масса: <Weight>».

Результат нажатия на странице свойств кнопки «Параметры оборудования» подробно описан на странице 15.

«Поиск оборудования» открывает стандартное окно поиска оборудования (см. выше).

DeleteEvent**УдалитьСообщение**

Метод удаляет из буфера событие с номером, задаваемым в свойстве **EventNumber**. Метод доступен только в асинхронном режиме (**AsyncMode** = TRUE).

См. также: свойства **EventNumber** и **AsyncMode**

Свойства драйвера

Англоязычное название	Русскоязычное название	Тип данных	Дост. ¹	По умолчанию	Тип св-а ²	Стр.
ApplicationHandle		Целое	RW	??? ⁴	G-	40
AsyncMode	АсинхронныйРежим	Логич.	RW	FALSE	L-	34
AutoDisable	АвтоВыключение	Логич.	RW	FALSE	G-	35
AutoZeroMode	РежимАвтонуля	Логич.	R	FALSE	L-	33
BaudRate	СкоростьОбмена	Целое	RW	7	L+	30
BigWeight	БольшойВес	Логич.	R	FALSE	L-	33
CurrentDeviceIndex	ИндексТекущегоУстройства	Целое	RW	0	G-	28
CurrentDeviceName	НаименованиеТекущегоУстройства	Строка	RW	'Без названия'	G-	28
CurrentDeviceNumber	НомерТекущегоУстройства	Целое	RW	1	G-	28
DataCount	ЧислоДанныхВОчереди	Целое	R	0	G-	35
DataEventEnabled	ПосылкаДанных	Логич.	RW	False	G-	34
DeviceCount	КоличествоУстройств	Целое	R	1	G+	27
DeviceDescription	ОписаниеУстройства	Строка	R	'Электронные весы'	L+	36
DeviceEnabled	УстройствоВключено	Логич.	RW	FALSE	L-	30
EventNumber	НомерСообщения	Целое	RW	0	G-	35
InvalidRange	НеверныйДиапазонВеса	Логич.	R	FALSE	L-	34
LockDevices	БлокироватьУстройства	Логич.	RW	FALSE	G-	29
Model	Модель	Целое	RW	5	L+	31
NegWeight	ОтрицательныйВес	Логич.	R	FALSE	L-	33
NettoWeight	ВесНетто	Логич.	R	FALSE	L-	32
NonStable	ВесНеСтабилен	Логич.	R	FALSE	L-	34
OutOfZero	НольВнеДиапазонаНуля	Логич.	R	FALSE	L-	33
Parity	Четность	Целое	RW	0	L+	30
PollingInterval	ПериодОпроса	Целое	RW	3	L-	36
PortNumber	НомерПорта	Целое	RW	1	L+	29
ResultCode	Результат	Целое	R	0	G-	26
ResultDescription	ОписаниеРезультат	Строка	R	'Ошибок нет'	G-	27

Описание драйвера (синхронный режим).

	a					
SalesPrice	ОбщаяЦена	Денеж.	R	0,00	L-	32

Продолжение таблицы.

Англоязычное название	Русскоязычное название	Тип данных	Дост. ¹	По умолчанию	Тип св-а ²	Стр.
TareWeight	ВесТары	Дроб.	RW	0,000	L-	31
UCodePage	УКодоваяСтраница	Целое	R	0	L-	37
UDescription	УОписание	Строка	R	пустая строка	L-	38
UMajorVersion	УСтаршаяВерсия	Целое	R	0	L-	38
UMinorVersion	УМладшаяВерсия	Целое	R	FALSE	L-	38
UMode	УРежим	Целое	R	0	L-	39
UModel	УМодель	Целое	R	0	L-	39
UnitPrice	ЦенаЗаЕдиницу	Денеж.	RW	0,00	L-	32
UpdatePrice	ОбновлятьЦену	Логич.	RW	FALSE	L-	32
UProtocolVersion	УВерсияПротокол а	Целое	R	0	L-	39
UType	УТип	Целое	R	0	L-	40
Value	Параметр	Целое	RW	2	L-	37
ValuePurpose	НазначениеПараметра	Дроб.	RW	0	L-	37
Version	Версия	Строка	R	??? ⁴	G+ ³	27
Weight	Вес	Дроб.	R	0,000	L-	31
WeightStep	ШагВеса	Дроб.	RW	0,000	L-	36

1 – Возможно два типа доступа к свойству: RW – свойство доступно для записи и чтения, R – свойство доступно только для чтения.

2 – Тип свойства кодируется двумя признаками: АВ

А: «G» – единое для всех ЛУ свойство

«L» – свойство ЛУ, то есть для каждого ЛУ существует отдельный экземпляр данного свойства

В: «+» – свойство сохраняется при выгрузке драйвера и восстанавливается при следующей загрузке драйвера.

«-» – свойство не сохраняется при выгрузке драйвера, а при следующей загрузке драйвера принимает значение по умолчанию.

3 – Это свойство – параметр двоичного файла. Изменяется файл – изменяется и свойство.

4 – Данное свойство не имеет специального значения по умолчанию.

Внимание! При попытках записи нового значения в свойство, имеющее доступ RW, драйвер: для свойств типа «Целое» проверяется вхождение в диапазон допустимых значений; для свойств типа «Дробное» и «Денежное» проверяется вхождение в диапазон допустимых значений и отклонение от точности (см. выше описание типов); для свойств типа «Строка» проверяется допустимость длины строки и отсутствие недопустимых символов. Если новое значение является недопустимым, то драйвер не записывает его в свойство, а сохраняет прежнее и выставляет -6 в **ResultCode**.

Свойства общего назначения

ResultCode

Результат

Тип: Integer / Целое (свойство доступно только для чтения)

Свойство содержит код ошибки, возникшей в результате выполнения последней операции (вызова функции или записи свойства драйвера). Если ошибки не произошло, то значение данного свойства устанавливается в 0 (Ошибок нет). Свойство общее для всех логических устройств и не восстанавливается при следующей загрузке драйвера в память. Возможны следующие значения:

Код ошибки	Описание ошибки
0	Ошибок нет
-1	Нет связи
-3	Порт недоступен
-4	Ключ защиты не найден
-6	Недопустимое значение
-7	Невозможно добавить устройство
-8	Невозможно удалить устройство
-9	Логическое устройство не найдено
-10	Неверная последовательность команд
-11	Устройство не включено
-12	Не поддерживается в данной версии оборудования
-14*	Порт занят приложением: (\\<имя ПК>\<описание приложения-клиента>\<описание драйвера>) <i>Например:</i> «Порт занят приложением: (\\Server\1C:Предприятие\Драйвер весов CAS серии LP»
-2600	Неверный тип протокола
-2601	Неверная скорость
-2602	Неверное значение режима "Sleep"
-2603	Неверное значение режима "Тара"
-2604	Неверное значение массы тары
-2605	Неверная команда
-2606	Переполнение поля "Стоимость"
17	Ошибка в значении тары
120	Неизвестная команда
121	Неверная длина данных команды
122	Неверный пароль
123	Команда не реализуется в данном режиме
124	Неверное значение параметра
150	Ошибка при попытке установки нуля
151	Ошибка при установке тары

Описание драйвера (синхронный режим).

152	Вес не фиксирован
166	Сбой энергонезависимой памяти
167	Команда не реализуется интерфейсом
170	Исчерпан лимит обращения с неверным паролем
180	Режим градуировки заблокирован градуировочным переключателем
181	Клавиатура заблокирована
182	Нельзя поменять тип текущего канала
183	Нельзя выключить текущий канал
184	С данным каналом вообще нельзя ничего сделать
185	Неверный номер канала

* – если драйвер смог определить, что порт занят одним из драйверов семейства «Штрих-М: Драйверы торгового оборудования», то возвращается данная ошибка (<имя ПК> – имя ПК, на котором запущено клиентское приложение, а <описание драйвера> – описание драйвера, занявшего порт).

См. также: свойство **ResultDescription**

ResultDescription**ОписаниеРезультата**

Тип: WideString / Строка (свойство доступно только для чтения)

Свойство содержит строку с описанием на русском языке кода ошибки последней выполнявшейся операции (см. столбец «Описание ошибки» в разделе **ResultCode**). Свойство общее для всех логических устройств и не восстанавливается при следующей загрузке драйвера в память.

См. также: свойство **ResultCode**

Version**Версия**

Тип: WideString / Строка (свойство доступно только для чтения)

Свойство содержит версию данного драйвера. Свойство общее для всех логических устройств.

См. также: свойство **UMajorVerion**, **UMinorVerion**, **UCodePage** и метод **GetDeviceMetrics**

Свойства для работы с ЛУ**DeviceCount****КоличествоУстройств**

Тип: Integer / Целое (свойство доступно только для чтения)

Значение свойства равно количеству существующих на данный момент логических устройств. Свойство общее для всех логических устройств.

См. также: свойства **CurrentDeviceIndex** и **CurrentDeviceNumber**

CurrentDeviceIndex

ИндексТекущегоУстройства

Тип: Integer / Целое

Свойство **CurrentDeviceIndex** содержит индекс текущего логического устройства. Изменение данного свойства приводит к установке текущим логического устройства с указанным индексом.

Если логического устройства с таким индексом не существует, то значение свойства **CurrentDeviceIndex** сохраняет значение, содержащееся до операции присвоения, а в **ResultCode** заносится -9.

Для определения числа логических устройств, существующих на данный момент, используйте свойство **DeviceCount**. Индекс первого логического устройства равен 0, а последнего **DeviceCount** – 1.

См. также: свойства **CurrentDeviceName**, **CurrentDeviceNumber** и **DeviceCount**

CurrentDeviceNumber

НомерТекущегоУстройства

Тип: Integer / Целое

Каждому логическому устройству при создании присваивается уникальный номер из диапазона от 1 до 99 включительно. Присваиваемый номер выбирается минимально возможным из незанятых.

Свойство **CurrentDeviceNumber** содержит номер текущего логического устройства. Изменение данного свойства приводит к установке текущим логического устройства с указанным номером.

Если логического устройства с таким номером не существует, то значение свойства **CurrentDeviceNumber** сохраняет значение, содержащееся до операции присвоения, а в **ResultCode** заносится -9.

См. также: свойства **CurrentDeviceName**, **CurrentDeviceIndex** и **DeviceCount**.

CurrentDeviceName

НаименованиеТекущегоУстройства

Тип: WideString / Строка

Данное свойство содержит название (описание) логического устройства до 20 символов (при попытке записи более длинной строки в свойство **ResultCode** будет записано -6, а свойство сохранит прежнее значение). Название логических устройств используются только для удобного различения их (устройств) между собой. Обычно содержит описание той единицы оборудования, для работы с которыми это устройство используется. Например: «Рыбный отдел» – весы, установленные в рыбном отделе.

Присвоение нового корректного значения свойству приводит к переименованию логического устройства.

Описание драйвера (синхронный режим).

См. также: свойства **CurrentDeviceNumber** и **CurrentDeviceIndex**.

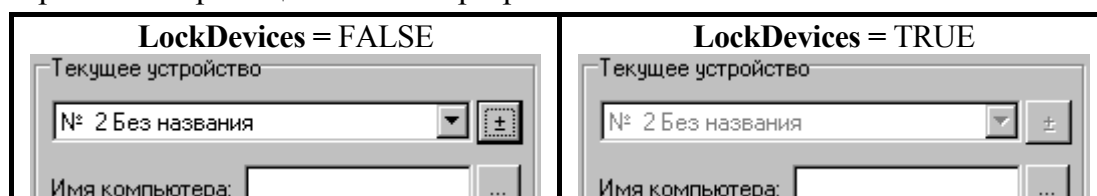
LockDevices**Блокировать Устройства**

Тип: WordBool / Логическое

Драйвер имеет визуальную страницу свойств, с помощью которой, в частности, можно создавать новые, удалять существующие, изменять текущие логические устройства. Иногда требуется запретить подобные действия, оставив возможность наглядного изменения свойств только текущего логического устройства. Этим и управляет данное свойство. Если **LockDevices=TRUE**, то работа с логическими устройствами с помощью визуальной страницы свойств заблокирована, иначе (FALSE) – разрешена.

Отметим, что методы **AddDevice**, **DeleteDevice** и свойства **CurrentDeviceName**, **CurrentDeviceNumber** и **CurrentDeviceIndex** доступны вне зависимости от содержимого свойства **LockDevices**.

Фрагмент страницы свойств при различных значениях **LockDevices**:



См. также: свойства **CurrentDeviceName**, **CurrentDeviceNumber**, **CurrentDeviceIndex** и методы **AddDevice**, **DeleteDevice** и **ShowProperties**

Свойства ЛУ, параметры связи с весами**PortNumber****НомерПорта**

Тип: Integer / Целое

Содержит номер порта ПК к которому подключены весы, на работу с которыми настроено данное логическое устройство.

Значение	1	...	32
Имя порта	COM1	...	COM32

Если **DeviceEnabled = TRUE**, то присвоение свойству **PortNumber** нового значения приводит к установке **DeviceEnabled = FALSE**, и последующей попытке восстановить **DeviceEnabled = TRUE** с новыми параметрами. После установки свойства необходимо проверить значение свойства **DeviceEnabled**, для определения, удалось ли проинициализировать новый порт, так как если порт занять не удалось **DeviceEnabled** принимает FALSE, а **ResultCode = 0**.

См. также: свойства **BaudRate**, **Parity** и **DeviceEnabled**

BaudRate

СкоростьОбмена

Тип: Integer / Целое

Содержит номер скорости обмена данными с весами (порт, через который производится связь с весами указывается свойством **PortNumber**):

Значение	3	4	5	7	10	12	14	18
Скорость, бод	1200	2400	4800	9600	19200	38400	57600	115200

Допустимые значения скорости обмена для различных типов весов:

Model = 3	3	4	5	7	-	-	-	-
	1200	2400	4800	9600	-	-	-	-
Model = 5	3	4	5	7	-	-	-	-
	1200	2400	4800	9600	-	-	-	-
Model = 11	-	4	5	7	10	12	14	18
	-	2400	4800	9600	19200	38400	57600	115200

См. также: свойства **PortNumber**, **Parity** и **DeviceEnabled**.

Parity

Четность

Тип: Integer / Целое

Задаёт метод проверки четности при обмене с весами. Допустимы значения и их интерпретация:

Parity	Model		
	3	5	11
0	+	+	+
2	+	-	-

«+» – допустимое значение, «-» – недопустимое значение.

При попытке записи недопустимых значений в **ResultCode** помещается -6 и значение свойства не изменяется (остается прежнее значение).

См. также: свойства **PortNumber** и **DeviceEnabled**.

DeviceEnabled

УстройствоВключено

Тип: WordBool / Логическое

Свойство указывает занял драйвер в монопольном режиме порт (см. **PortNumber**) или нет. Если клиент выставляет **DeviceEnabled** = TRUE, то после установки этого свойства необходимо проверить, удалось ли открыть порт в монопольном режиме, в противном случае **DeviceEnabled** остается FALSE (см. также **ResultCode**).

Описание драйвера (синхронный режим).

См. также: свойства **PortNumber**, **BaudRate** и **Parity**

Model**Модель**

Тип: Integer / Целое

Тип весов, на работу с которыми настроено данное ЛУ. Допустимы значения и их интерпретация:

Значение	Модели	Протокол
3	Штрих МІ Штрих МІІІ Штрих М5 Штрих М5Ф Штрих М5ФС	Штрих 5 Штрих 6
5	Штрих МІІ	POS1.0
11	BM100 Штрих М5Ф POS Штрих ВІ	POS2.0

Внимание! При **Model** = 3 установите на весах протокол обмена Штрих 6. При работе по протоколу Штрих 5 возможна потеря старшего разряда цены, что является особенностью данного протокола!

Параметры, получаемые от весов**Weight****Вес**

Тип: Double / Дробное (свойство доступно только для чтения)

Свойство содержит массу товара, находящегося на чаше весов.

Свойство заполняется методом **ReadWeight** и является параметром асинхронного сообщения.

См. также: свойства **TareWeight**, **NegWeight**, **AsyncMode** и метод **ReadWeight**

TareWeight**ВесТары**

Тип: Double / Дробное

Свойство содержит массу тары, установленную в весах (заполняется методом **ReadWeight**) и задает массу тары для весов (используется методом **SetTareWeight**).

Свойство так же является параметром асинхронного сообщения.

См. также: свойства **Weight**, **AsyncMode** и методы **ReadWeight** и **SetTareWeight**

UpdatePrice

ОбновлятьЦену

Тип: WordBool / Логическое

Если свойство содержит TRUE, то при вызове метода **ReadWeight** драйвер сначала задает в весах цену товара за 1 кг (из свойства **UnitPrice**), а уж потом запрашивает данные с весов. В этом случае весы самостоятельно подсчитывают стоимость товара, используя только что полученную цену. Допустимо только при **Model** = 3 или 5. Если **UpdatePrice** = FALSE, то весы при вычислении стоимости используют цену, введенную с клавиатуры, и **ReadWeight** заносит ее (цену за 1 кг) в свойство **UnitPrice**.

См. также: свойства **UnitPrice**, **Weight**, **SalesPrice** и метод **ReadWeight**

UnitPrice

ЦенаЗаЕдиницу

Тип: Currency / Денежное

Свойство содержит цену за 1 кг, установленную в весах (заполняется методом **ReadWeight**) и задает цену за 1 кг для весов (используется методом **ReadWeight** при [(**UpdatePrice** = TRUE) И ((**Model** = 3) ИЛИ (**Model** = 5))].

Свойство так же является параметром асинхронного сообщения.

См. также: свойства **Weight**, **SalesPrice**, **AsyncMode** и метод **ReadWeight**

SalesPrice

ОбщаяЦена

Тип: Currency / Денежное (свойство доступно только для чтения)

Свойство содержит стоимость (Стоимость = Цена за 1 кг * Масса), вычисленную весами (заполняется методом **ReadWeight**).

Свойство так же является параметром асинхронного сообщения.

См. также: свойства **Weight**, **UnitPrice**, **AsyncMode** и метод **ReadWeight**

NettoWeight

ВесНетто

Тип: WordBool / Логическое (свойство доступно только для чтения)

Свойство содержит TRUE если в весах задана ненулевая масса тары (заполняется методом **ReadWeight**).

Свойство так же является параметром асинхронного сообщения.

Свойство имеет смысл только при **Model** = 5 или 11.

См. также: свойства **Weight**, **AsyncMode** и метод **ReadWeight**

AutoZeroMode

РежимАвтонуля

Тип: WordBool / Логическое (свойство доступно только для чтения)

Свойство содержит TRUE если на чаше весов груза нет (заполняется методом **ReadWeight**).

Свойство так же является параметром асинхронного сообщения.

Свойство имеет смысл только при **Model** = 5 или 11.

См. также: свойства **Weight**, **AsyncMode** и метод **ReadWeight**

OutOfZero

НольВнеДиапазонаНуля

Тип: WordBool / Логическое (свойство доступно только для чтения)

Свойство содержит TRUE если весы либо сломаны, либо перешли в нерабочее состояние – выключите и включите весы (заполняется методом **ReadWeight**).

Свойство так же является параметром асинхронного сообщения.

Свойство имеет смысл только при **Model** = 5 или 11.

См. также: свойства **Weight**, **AsyncMode** и метод **ReadWeight**

NegWeight

ОтрицательныйВес

Тип: WordBool / Логическое (свойство доступно только для чтения)

Свойство содержит TRUE если на весах показания индикатора «вес» содержит отрицательное значение (заполняется методом **ReadWeight**).

Свойство так же является параметром асинхронного сообщения.

Свойство имеет смысл только при **Model** = 5 или 11.

См. также: свойства **Weight**, **AsyncMode** и метод **ReadWeight**

BigWeight

БольшойВес

Тип: WordBool / Логическое (свойство доступно только для чтения)

Свойство содержит TRUE если на чаше весов находится груз с массой, превышающей «Наибольший Предел Взвешивания» данных весов (заполняется методом **ReadWeight**).

Свойство так же является параметром асинхронного сообщения.

Свойство имеет смысл только при **Model** = 5 или 11.

См. также: свойства **Weight**, **AsyncMode** и метод **ReadWeight**

NonStable

ВесНеСтабилен

Тип: WordBool / Логическое (свойство доступно только для чтения)

Свойство содержит TRUE если весы не могут определить взвешиваемую массу, так как она нестабильна (заполняется методом **ReadWeight**).

Свойство так же является параметром асинхронного сообщения.

Свойство имеет смысл только при **Model** = 5 или 11.

См. также: свойства **Weight**, **AsyncMode** и метод **ReadWeight**

InvalidRange

НеверныйДиапазонВеса

Тип: WordBool / Логическое (свойство доступно только для чтения)

Свойство содержит TRUE если то на чаше весов находится груз с массой, превышающей «Наибольший Предел Взвешивания» или меньший (по модулю), чем «Наименьшего Предела Взвешивания» весов или если масса не стабильна (см. также **NonStable**) (заполняется методом **ReadWeight**).

Свойство так же является параметром асинхронного сообщения.

Свойство имеет смысл только при **Model** = 5 или 11.

См. также: свойства **Weight**, **AsyncMode** и метод **ReadWeight**

Свойства асинхронного режима

AsyncMode

АсинхронныйРежим

Тип: WordBool / Логическое

Если свойство содержит TRUE, то данное логическое устройство находится в асинхронном режиме, то есть драйвер самостоятельно опрашивает весы с интервалом времени, заданным в **PollingInterval**, и если изменение показаний весов превысило значение свойства **WeightStep** (по абсолютному значению), то драйвер вызывает асинхронное событие, оповещающее клиента об изменениях в показаниях весов.

Свойство имеет смысл только при **Model** = 5 или 11.

Внимание! При установке свойства **DeviceEnabled** = FALSE свойство **AsyncMode** также устанавливается равным FALSE, а при обратном присвоении **DeviceEnabled** = TRUE **AsyncMode** не восстанавливается.

См. также: свойства **EventNumber**, **Model** и метод **DeleteEvent**

DataEventEnabled

ПосылкаДанных

Тип: WordBool / Логическое

Если свойство содержит TRUE, то включается генератор событий

Описание драйвера (синхронный режим).

(механизм передачи сообщений из драйвера в обработчик событий), в противном случае (**DataEventEnabled** = FALSE), генератор событий останавливается, и события, получаемые от включенных (**DeviceEnabled** = TRUE) ЛУ, запоминаются в буфере драйвера (размер буфера неограничен).

Имеет смысл только при **AsyncMode** = TRUE, но при изменении **AsyncMode** свойство сохраняет свое содержимое.

См. также: свойства **DeviceEnabled** , **AutoDisable** и **DataCount**

AutoDisable**АвтоВыключение**

Тип: WordBool / Логическое

Если свойство содержит TRUE, то перед передачей сообщения на обработку (в пользовательский обработчик событий) свойство **DataEventEnabled** сбрасывается в FALSE самим драйвером. Таким образом, на время обработки сообщения генератор событий останавливается. Чтобы возобновить (из пользовательского приложения) работу механизма передачи сообщений необходимо установить **DataEventEnabled** в TRUE, это можно сделать в конце обработчика внешнего события.

Имеет смысл только при **AsyncMode** = TRUE, но при изменении **AsyncMode** свойство сохраняет свое содержимое.

См. также: свойства **DataEventEnabled**

DataCount**ЧислоДанныхВОчереди**

Тип: Integer / Целое (свойство доступно только для чтения)

Свойство содержит количество сообщений, находящихся в буфере драйвера.

Имеет смысл только при **AsyncMode** = TRUE, но при изменении **AsyncMode** свойство сохраняет свое содержимое.

Примечание: При **AsyncMode** = FALSE свойство **DataCount** всегда содержит 0. То есть при выходе из асинхронного режима (присвоение **AsyncMode** = FALSE) буфер сообщений очищается.

См. также: свойства **EventNumber**, **AsyncMode** и метод **DeleteEvent**

EventNumber**НомерСообщения**

Тип: Integer / Целое

Свойство содержит номер текущего сообщения, параметрами которого заполнены свойства драйвера.

При вызове обработчика асинхронного события драйвер заносит в **EventNumber** номер первого (с минимальным номером) необработанного сообщения, расположенного в буфере.

При записи в **EventNumber** нового значения драйвер проверяет,

Описание драйвера (синхронный режим).

находится ли сообщение с заданным номером в буфере? Если в буфере находится сообщение с заданным номером, то делает это сообщение текущим (при этом в свойства драйвера заносятся параметры нового текущего сообщения, в том числе делается текущим то ЛУ, от которого пришло данное сообщение). Если сообщения с таким номером в буфере нет, то текущим останется прежнее сообщение, а в **ResultCode** запишется -6.

Имеет смысл только при **AsyncMode** = TRUE, но при изменении **AsyncMode** свойство сохраняет свое содержимое.

См. также: свойства **EventNumber**, **AsyncMode** и метод **DeleteEvent**

PollingInterval**ПериодОпроса**

Тип: Integer / Целое

Свойство задает интервал времени между опросами весов в асинхронном режиме:

Значение	0	1	2	3	4
Интервал, мс	50	100	200	500	1000

Имеет смысл только при **AsyncMode** = TRUE, но при изменении **AsyncMode** свойство сохраняет свое содержимое.

См. также: свойства **AsyncMode** и **WeightStep**

WeightStep**ШагВеса**

Тип: Double / Дробное

Свойство задает «чувствительность» драйвера к изменению показаний весов в асинхронном режиме. При изменении показаний весов (массы) более чем на **WeightStep** драйвер генерирует сообщение. Более подробно см. ниже.

Имеет смысл только при **AsyncMode** = TRUE, но при изменении **AsyncMode** свойство сохраняет свое содержимое.

См. также: свойства **AsyncMode** и **PollingInterval**

Прочие свойства**DeviceDescription****ОписаниеУстройства**

Тип: WideString / Строка (свойство доступно только для чтения)

Для текущего логического устройства свойство содержит описание оборудования, режима его работы и прочую информацию.

Для данного драйвера свойство всегда содержит «Электронные весы».

См. также: свойство **Model**

Описание драйвера (синхронный режим).**ValuePurpose****НазначениеПараметра****Тип:** Integer / ЦелоеНомер системной настройки для методов **GetValue** и **SetValue**.

Список возможных значений определяется таблицей:

Значение	Описание	Возможные значения для Value	Model
0	Тип протокола	0 – Штрих 1 – Электроника 2 – Штрих AC POS	5
1	Скорость связи*	3 – 1200 бод; 4 – 2400 бод; 5 – 4800 бод; 7 – 9600 бод.	5 и 11
2	Режим "Sleep"	FALSE (0) – выключен; TRUE (не ноль) – включен.	5
3	Режим "Тара"	FALSE (0) – единичная установка тары; TRUE (не ноль) – множественная установка тары.	5
4	Пароль**	0..9999	11

* – после удачной записи параметра в весы драйвер автоматически изменяет значение свойства **BaudRate**, что бы связь не прерывалась.

** – используется при выполнении методов **ReadWeight**, **Reset**, **SetTareWeight**, **Tare**, **ZeroScale** и **SetValue** (установка скорости обмена).

См. также: свойство **Value** и методы **SetValue** и **GetValue****Value****Параметр****Тип:** Double / Дробное

Численное значение для программирования системных настроек весов методом **SetValue** или заполняется методом **GetValue** при чтении системных настроек весов.

Используется для логических и целочисленных значений, драйвер сам определяет тип целевого свойства и производит, если необходимо, преобразование.

См. также: свойство **ValuePurpose** и методы **SetValue** и **GetValue****UCodePage****УКодоваяСтраница****Тип:** Integer / Целое (свойство доступно только для чтения)

Параметр обновляется методом **GetDeviceMetrics**. В свойстве содержится «кодовая таблица» (код языка), для которого адаптированы все ресурсы устройства. Данный параметр обычно выводится как третья

составляющая версии устройства.

Например: 1.6.Rus (версия 1.6 для России), 1.6.Blr (версия 1.6 для Белоруссии).

На сегодня «Штрих-М» ввело следующие коды языковых таблиц:

Код	Язык
0	Русская
1	Армянская
2	Молдавская
3	Белорусская
4	Украинская
5	Казахская
6	Английская
7..255	Зарезервировано

Примечание. Код языковой таблицы не имеет никакого отношения с CodePage ОС MS-DOS или Microsoft Windows.

См. Также: свойства **UmajorVersion**, **UminorVersion** и метод **GetDeviceMetrics**

Udescription

Уописание

Тип: WideString / Строка (свойство доступно только для чтения)

Параметр обновляется методом **GetDeviceMetrics**. В свойстве содержится строка, описывающая оборудование. Длина строки может быть произвольной и зависит только от длины названия оборудования, заложенного его разработчиками.

Например: «Штрих АС POS», «ЭЛБЕС-МИКРО-Ф», «Memo Plus III»

См. также: свойства **Utype**, **Umodel** и метод **GetDeviceMetrics**

UmajorVersion

УстаршаяВерсия

Тип: Integer / Целое (свойство доступно только для чтения)

Параметр обновляется методом **GetDeviceMetrics**. В свойстве содержится версия оборудования. Данный параметр обычно выводится как первая составляющая версии устройства.

Например: 1.6.Rus (версия 1), 2.0.Rus (версия 2).

См. также: свойства **UminorVersion**, **UcodePage** и метод **GetDeviceMetrics**

UminorVersion

УмладшаяВерсия

Тип: Integer / Целое (свойство доступно только для чтения)

Параметр обновляется методом **GetDeviceMetrics**. В свойстве содержится «подверсия» оборудования. Данный параметр обычно выводится как вторая составляющая версии устройства.

Описание драйвера (синхронный режим).

Например: 1.6.Rus (версия 1, подверсия 6), 1.7.Rus (версия 1, подверсия 7).

См. также: свойства **UmajorVersion**, **UcodePage** и метод **GetDeviceMetrics**

Umode**Урежим**

Тип: Integer / Целое (свойство доступно только для чтения)

Параметр обновляется методом **GetDeviceMetrics**. В свойстве содержится двухбайтовое беззнаковое целое число, представляющее собой битовую кодировку поддерживаемых оборудованием режимов.

Для Штрих АС POS всегда 0.

См. также: свойства **Utype**, **Umodel** и метод **GetDeviceMetrics**

Umodel**Умодель**

Тип: Integer / Целое (свойство доступно только для чтения)

Параметр обновляется методом **GetDeviceMetrics**. В свойство заносится код модели оборудования. Для каждого типа оборудования введено отдельное множество кодов моделей.

На сегодня «Штрих-М» ввело следующие коды моделей:

Тип	Модель	Описание модели
2 (Весы)	1	Штрих АС POS
3 (весовой модуль)	0	BM100

См. также: свойства **UType**, **UModel** и метод **GetDeviceMetrics**

UProtocolVersion**УВерсияПротокола**

Тип: Integer / Целое (свойство доступно только для чтения)

Параметр обновляется методом **GetDeviceMetrics**. В свойство заносится версия поддерживаемой универсальной команды определения типа оборудования (**GetDeviceMetrics**).

Данный драйвер корректно может обрабатывать только версию 1 (на сегодня это последняя версия этой команды). Если значение свойства отлично от 1, то содержимое свойств **UType**, **UModel**, **Umode**, **UmajorVersion**, **UminorVersion**, **UcodePage**, **Udescription** не было обновлено из-за того, что драйвер «не умеет» работать с данной версией команды.

См. также: метод **GetDeviceMetrics**

UType**УТип**

Тип: Integer / Целое (свойство доступно только для чтения)

Параметр обновляется методом **GetDeviceMetrics**. В свойство заносится код типа оборудования.

См. также: свойства **UModel**, **UDescription** и метод **GetDeviceMetrics**

ApplicationHandle

Тип: Integer

После загрузки драйвера (библиотеки) в это свойство можно записать дескриптор главного окна приложения-клиента. Это предотвратит появление отдельных кнопок в панели задач при отображении визуальной страницы свойств и других окон драйвера.

Внимание! При использовании драйвера в качестве внешней компоненты данное свойство не поддерживается, так как драйвер при загрузке инициализирует свойство корректным значением.

Предупреждение! При записи значения в данное свойство следует проявлять особую аккуратность, так как запись некорректного значения может привести к нарушениям работы системы (вплоть до полного повисания).

АСИНХРОННЫЙ РЕЖИМ

Введение

При построении АРМК бывает необходимо выводить текущее значение массы товара, расположенного на чаше весов, на монитор кассира и дисплей покупателя (далее оба устройства будут называться «дисплей»). Для этого следует периодически опрашивать весы и обновлять изображение на дисплее. Для сред программирования интерпретаторного типа (например 1С:ПРЕДПРИЯТИЕ, VBA) это вызывает определенные сложности. Поэтому драйвер берет на себя эту обязанность.

В асинхронном режиме драйвер периодически запрашивает показания весов. Если данные получены, то драйвер вычисляет изменение массы Δm_i :

$$\Delta m_i = |m_{i-1} - m_i|,$$

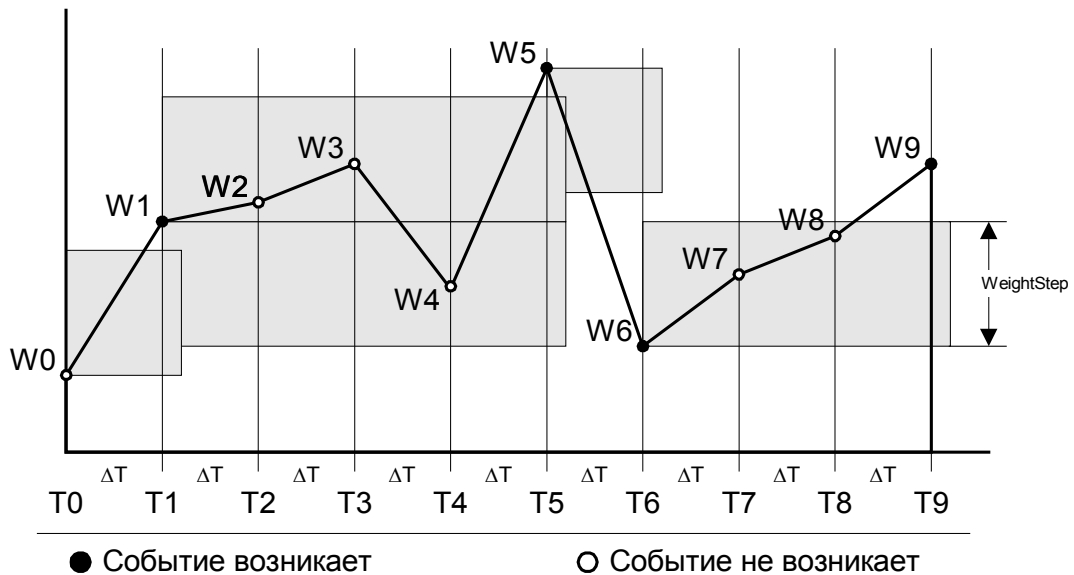
где:

Δm_i – изменение показаний весов;

m_{i-1} – показания весов, полученные при предыдущем* опросе;

m_i – показания весов, полученные при данном опросе;

* – более наглядно термин «предыдущий» опрос демонстрирует рисунок и Пояснение3:



Пояснение1: в момент времени T_0 был включен асинхронный режим.

Пояснение2: серые области – предел изменения (**WeightStep**). На интервалах $T_0 \dots T_1$, $T_5 \dots T_6$ и $T_6 \dots T_9$ указаны только половины диапазонов (что бы не загромождать рисунок), а для диапазона $T_1 \dots T_5$ предел изменения указан полностью.

Пояснение3: W_0 является предыдущим значением для W_1 , W_1 (но не W_4) для W_5 , W_5 для W_6 , W_6 (но не W_8) для W_9 .

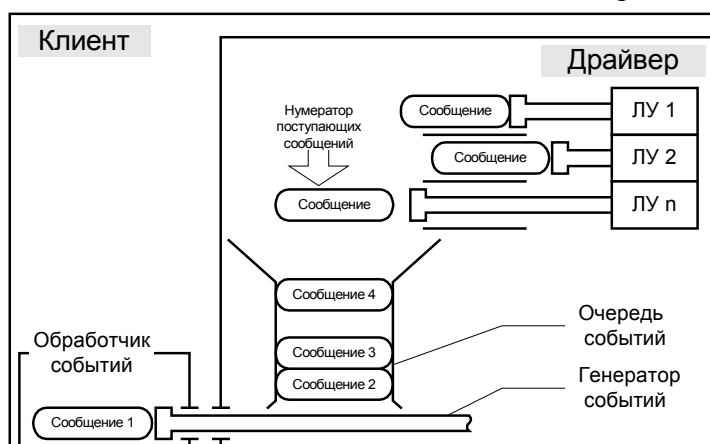
Драйвер позволяет настроить ΔT (свойство **PolingInterval**) и предел изменения массы – свойство **WeightStep**.

Если значение Δm_i превысило значение свойства **WeightStep**, то драйвер помещает новое сообщение в очередь.

При установке **DataEventEnabled** = TRUE включается генератор событий. Генератор событий – специальный механизм, постоянно опрашивающий очередь сообщений драйвера. Если в очереди генератор обнаруживает событие, еще не посылавшееся клиенту, то:

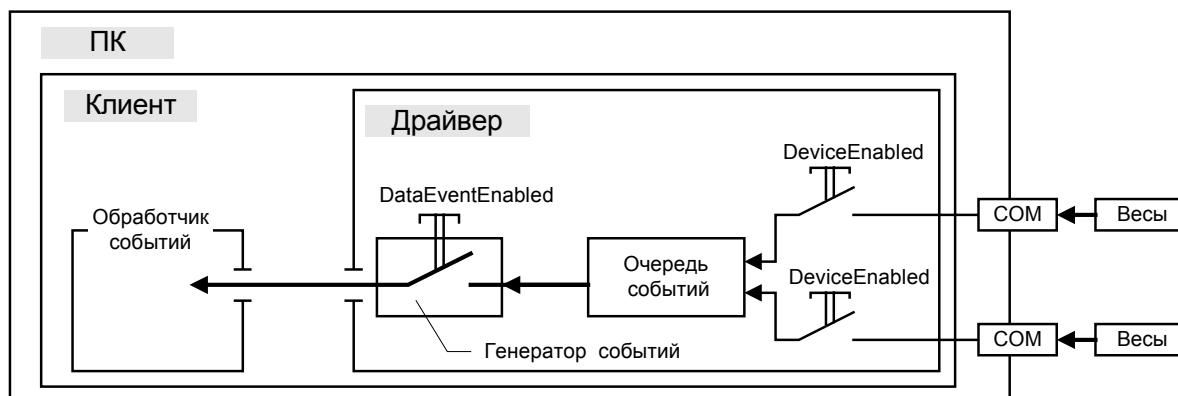
1. Заполняет свойства драйвера параметрами текущего (см. описание свойства **EventNumber**) сообщения.
2. Событие помечается как «посылавшееся клиенту».
3. Если **AutoDisable** = TRUE, то **DataEventEnabled** = FALSE.
4. Вызывается обработчик асинхронного события клиента.

Примечание: если клиентский обработчик событий выполняется, то генератор ожидает его окончания, то есть многопоточкового вызова производиться не будет.



Примечание: из рисунка очевидно, что сообщения от разных ЛУ помещаются в очередь по мере поступления.

Обобщенно, схема поступления данных выглядит следующим образом:



Примечание: если сообщение попало в очередь, то значение свойства **DeviceEnabled** для данного сообщения уже не играет роли.

Очевидно, что автоматического уничтожения событий из очереди не производится – это задача клиента. То есть в обработчике клиента следует удалять события из очереди драйвера явно, вызывая метод драйвера **DeleteEvent**. Если этого не делать, то сообщения будут накапливаться в очереди, что приведет к напрасному расходу оперативной памяти ПК и некоторому замедлению работы системы в целом.

Прием событий от драйвера

Система программ 1С:ПРЕДПРИЯТИЕ v.7.x

В системе 1С:ПРЕДПРИЯТИЕ (далее просто «система») существует предопределенная процедура «ОбработкаВнешнегоСобытия», имеющая синтаксис:

Синтаксис:

ОбработкаВнешнегоСобытия (Источник, Событие, Данные)

Англоязычный синоним:

ExternEventProcessing

Параметры:

Источник	строка, в случае данного драйвера, содержащая константу «Штрих АС POS».
Событие	строка, в случае данного драйвера, содержащая константу «Номер сообщения».
Данные	строка, в случае данного драйвера, содержащая константу – номер события.

Возвращаемое значение:

отсутствует

Описание:

Данная процедура вызывается системой при приходе очередного события. При получении события системой будет вызвана процедура **ОбработкаВнешнегоСобытия**, определенная в модуле активной на данный момент времени формы. Если в модуле активной формы данная процедура не описана, то будет вызвана одноименная процедура из глобального модуля. Если в глобальном модуле процедура не описана, то выдается сообщение об этом. Процедура **ОбработкаВнешнегоСобытия** в глобальном модуле не вызывается, если она вызывалась в модуле формы.

Пример: простейший обработчик внешнего события

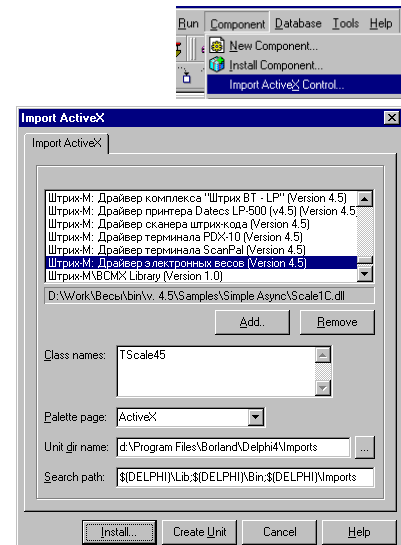
```
//текст модуля
Процедура ОбработкаВнешнегоСобытия (Источник, Событие, Данные)
    Сообщить ("Источник: " + Источник);
    Сообщить ("Событие : " + Событие);
    Сообщить ("Данные : " + Данные);
    Сообщить ();
    Scl.DeleteEvent ();
    // Scl – имя объекта весов
    // DeleteEvent – см. стр. 23
КонецПроцедуры
```

Borland Delphi 3 и Delphi 4

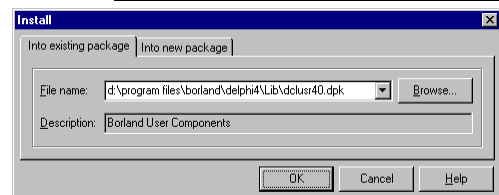
В Delphi3 и Delphi4 компонент «драйвер электронных весов» можно поместить на палитру компонентов.

Выберите пункт «Import ActiveX Control...»
меню «Component»

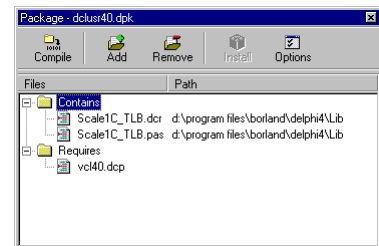
- Выберите элемент «Штрих-М: Драйвер электронных весов (Version 4.5)»;
- в поле «Palette Page» укажите страницу палитры, на которую желаете добавить компонент;
- в поле «Unit dir name» задайте директорию, в которой будет сохранен импортируемый модуль;
- нажмите кнопку «Install...».



Укажите имя существующего (закладка «Into existing package») или нового (закладка «Into new package») пакетного файла.



- Нажмите «Yes» для подтверждения перекомпиляции пакетного файла.
- Нажмите «OK» в окне сообщения об удачной перекомпиляции.
- Закройте окно пакетного файла.
- Нажмите «Yes» для подтверждения сохранения пакетного файла.



После этого на указанной закладке будет расположен компонент «Scale45». Экземпляры данного компонента имеют обработчик внешних событий – **OnDataEvent**. Для получения асинхронных событий назначьте обработчик события **OnDataEvent** для компонента весов.

Пример: простейший обработчик внешнего события

```
//текст модуля формы, на которую добавлен компонент Scale45
//на форме присутствует компонент TMemo с именем Memo1
procedure TForm1.Scale451DataEvent(Sender: TObject);
begin
    Memo1.Lines.Add (IntToStr (Scale451.EventNumber) +
                      ': ' +
                      FloatToStr (Scale451.Weight));
    Scale451.DeleteEvent;
end;
```