

## Описание работы программы.

Используемое оборудование :

Промышленный терминал ТВ-015\ТЦ-015\ТЦ-017 с Микропроцессорным блоком управления и обработки данных. Степень защищенности – IP-65

Клавиатура, обеспечивающая ввод цифровых кодов. Индикация веса, тары ЖКИ и интерфейс CENTRONIX для вывода символьной информации.

Адаптер RS232-RS485 (как минимум 1 шт.) для работы с интерфейсом RS или преобразователь RS-Ithernet для работы по лвс.

ПК, обеспечивающий нормальную работу в соответствующей операционной среде и имеющий как минимум один рабочий и не занятый СОМ-порт.

Линии коммуникации. Стандартный интерфейсный кабель для связи ПК -> адаптер и витая пара 5-й категории для связи адаптер -> терминалы

При необходимости каждое рабочее место может оборудоваться контроллером считывателя штрих кодов\принтера.

### ТРЕБОВАНИЯ К ОПЕРАЦИОННЫМ СРЕДАМ.

Функционирование ПРОГРАММЫ обеспечивается в операционных средах:

- Windows 2000 Professional
- Windows XP Professional

ПРОГРАММА выполняет следующие функции:

- Взаимодействует с системой 1С:Предприятие или любой другой внешней системой, поддерживающей соответствующую технологию обмена данными между приложениями
- Обеспечивает синхронную работу всех рабочих мест в целом

Коррекция, сохранение и проводка созданных документов осуществляется средствами внешней системы.

### Система 1С:Предприятие или любая другая система должна экспортировать 1 метод :

- **EventsHandler** : функция, осуществляющая обработку событий.  
Входные параметры : **Adr** целое число(адрес устройства)  
**ScNum** целое число(номер весовой платформы)  
**Event** целое число(номер возникшего события)  
**Cod** строка(введенный код)  
**Weight** вещественное число(вес)  
**Date** строка(дата)  
**Time** строка(время)

Возвращаемое значение : Строка. При выводе на индикатор, длина строки не должна превышать 20 символов. Строке должен предшествовать обязательный префикс **ADR=n**, где **n** – адрес устройства вывода . Префикс должен заканчиваться байтом **0D**.

При выводе на устройство печати, в строке после префикса с адресом должен быть указан префикс **PRN(M\L)=n** ,где **M** – вывод на матричный принтер, **L** – вывод на этикеточный принтер, **n** – количество копий. Префикс должен заканчиваться байтом **0D**. При выводе на принтер **N** – номер принтера. Префикс должен заканчиваться байтом **0D**.

Пример написания строки возврата на языке Delphi Pascal :

'ADR=2'+#0d+'Марка изделия'

'ADR=3'+#0d+'PRNM=2'+#0d+'1'+#0d+'Марка изделия'

В первом случае строка «Марка изделия» будет выведена на индикатор терминала с адресом 2, во втором случае – дважды на матричный принтер номер 1,подключенный к устройству с адресом 3.

### Предопределенные события :

**Event=\$30** – отказ от запрошенного внешней системой действия.

**Event=\$31** – подтверждение запрошенного внешней системой действия.

**Event=\$70** – считан штрих код. Параметр **Cod** содержит значение считанного кода.

**Event=\$80** – в устройстве отсутствует модуль принтера.

**Event=\$81** – принтер не готов.

### ПРОГРАММА экспортирует 3 метода :

- **GetStatus (AdrTerm, ChannelN)** : Возвращает текущий статус весовой системы.

Метод имеет 2 входных параметра – адрес терминала и № канала  
Входные параметры представляют собой целые числа : **AdrTerm** от 0 до 255 (для 232 интерфейса), **ChannelN** - 1 или 2. **ChannelN** никак не связан с № СОМ-порта, иными словами 1 канал не обязательно может быть на СОМ1. Возвращаемое значение представляет собой целое число, отражающее значение текущего статуса. Возможные значения :

- 5000 - терминал не обнаружен
- 5001 - с терминалом нет связи
- 5002 - весы успокоились
- 5003 - весы не успокоились.

- **GetWeight(AdrTerm, ChannelN)** : Возвращает текущий вес.

Метод имеет 2 входных параметра – адрес терминала и № канала  
Входные параметры представляют собой целые числа : **AdrTerm** от 0 до 255 (для 232 интерфейса), **ChannelN** - 1 или 2. **ChannelN** никак не связан с № СОМ-порта, иными словами 1 канал не обязательно может быть на СОМ1. Возвращаемое значение представляет собой вещественное число, отражающее значение текущего веса. Возвращаемые значения -5000 и -5001 означают «терминал не обнаружен» и «с терминалом нет связи» соответственно.

- **SendMessage(AdrTerm, ChannelN,Msg)** Посылает сообщение на индикатор терминала.

Метод имеет 3 входных параметра – адрес терминала, № канала и посылаемое сообщение. Первые два параметра представляют собой целые числа : **AdrTerm** от 0 до 255 (для 232 интерфейса), **ChannelN** - 1 или 2. **ChannelN** никак не связан с № СОМ-порта, иными словами 1 канал не обязательно может быть на СОМ1. Последний параметр – строка, длиной не более 20 символов. Возвращаемое значение представляет собой целое число. Возвращаемые значения -5000 и -5001 означают «терминал не обнаружен» и «с терминалом нет связи» соответственно, -5002 – успешное завершение отсылки.

Имя сервера для программного доступа к методам **TB015.TB**

Для использования внешней системой методов программы, необходимо при старте системы создать требуемый объект.

### Пример использования в 1С-Предприятии 7.7

В “Глобальный модуль” поместить код создания объекта. Для этого в секции объявления глобальных переменных объявить соответствующую переменную.

*Перем Устройство Экспорт;*

А в процедуре **ПриНачалеРаботыСистемы** создать объект

*Устройство = СоздатьОбъект(“TB015.TB ”);*

Далее в нужных местах программы осуществлять вызовы необходимых методов.

*Результат = Устройство.GetWeight(1,1);*

*Результат = Устройство.SendMessage(1,1, “Нажмите клавишу 2”);*

### Пример конфигурирования 1С-Предприятия.

*//Глобальные описания переменных*

*// В качестве индекса массивов используется общее количество*

*//рабочих мест(весов)*

*Перем СписокДокументы[4] Экспорт;*

*Перем СписокТоваровМассив[4] Экспорт;*

*Перем СписокКолвоТоваров[4] Экспорт;*

*Перем СписокДатаПокупки[4] Экспорт;*

*Перем СписокВремяПокупки[4] Экспорт;*

*Перем СписокВесовщики[4] Экспорт;*

*//Описание функции экспорта*

*Функция EventHandler(Терминал, Платформа, Событие, Код, Количество, ДатаПокупки, ВремяПокупки) Экспорт*

```

var ИскомыйСправочник, Док, Контрагенты, Номенклатура, КодКлиента, Сч, КодТовара, Весовщики;
Если Событие = 1 Тогда
  Если (Код = "") or (Код = "0") Тогда Возврат "ADR="+Терминал+Chr(13)+"Код не введен";
  КонецЕсли;
  ИскомыйСправочник = СоздатьОбъект("Справочник.Весовщики");
  Если ИскомыйСправочник.НайтиПокоду(Код) = 1 Тогда
    СписокВесовщики[Терминал] = Код;
    Возврат "ADR="+Терминал+Chr(13)+ИскомыйСправочник.ПолноеНаименование();
  Иначе
    Возврат "ADR="+Терминал+Chr(13)+"Ошибка авторизации";
  КонецЕсли;
КонецЕсли;

Если Событие = 112 Тогда
  Возврат "ADR="+Терминал+Chr(13)+"PRNL=1"+Chr(13)+"N"+Chr(13)+Chr(10)+
    "A450,20,0,4,1,1,N,""Сервилат"+Строка(Количество)+"""+Chr(13)+Chr(10)+
    "A450,50,0,4,1,1,N,""Штрих код "+Код+""""+Chr(13)+Chr(10)+
    "A450,80,0,4,1,1,N,""Штрих код "+Код+""""+Chr(13)+Chr(10)+
    "A450,110,0,4,1,1,N,""Штрих код "+Код+""""+Chr(13)+Chr(10)+
    "A450,140,0,4,1,1,N,""Штрих код "+Код+""""+Chr(13)+Chr(10)+
    "A450,170,0,4,1,1,N,""Штрих код "+Код+""""+Chr(13)+Chr(10)+
    "A450,200,0,4,1,1,N,""Штрих код "+Код+""""+Chr(13)+Chr(10)+
    "A450,230,0,4,1,1,N,""Штрих код "+Код+""""+Chr(13)+Chr(10)+
    "A450,260,0,4,1,1,N,""Штрих код "+Код+""""+Chr(13)+Chr(10)+
    "P1"+Chr(13)+Chr(10);
  КонецЕсли;

Если Событие = 2 Тогда
  Если (Код = "") or (Код = "0") Тогда Возврат "ADR="+Терминал+Chr(13)+"Код не введен";
  КонецЕсли;
  ИскомыйСправочник = СоздатьОбъект("Справочник.ДокументыНаОтгрузку");
  Если ИскомыйСправочник.НайтиПокоду(Код) = 1 Тогда
    СписокДокументы[Терминал] = Код;
    Возврат "ADR="+Терминал+Chr(13)+ИскомыйСправочник.ПолноеНаименование();
  Иначе
    Возврат "ADR="+Терминал+Chr(13)+"Ошибочный код";
  КонецЕсли;
КонецЕсли;

Если Событие = 3 Тогда
  Если (Код = "") or (Код = "0") Тогда Возврат "ADR="+Терминал+Chr(13)+"Код не введен";
  КонецЕсли;
  ИскомыйСправочник = СоздатьОбъект("Справочник.Номенклатура");
  Если ИскомыйСправочник.НайтиПокоду(Код) = 1 Тогда
    СписокТоваровМассив[Терминал].ДобавитьЗначение(Код);
    Возврат "ADR="+Терминал+Chr(13)+ИскомыйСправочник.ПолноеНаименование();
  Иначе
    Возврат "ADR="+Терминал+Chr(13)+"Ошибочный код";
  КонецЕсли;
КонецЕсли;

Если Событие = 4 Тогда
  Если (Код = "") or (Код = "0") Тогда Возврат "ADR="+Терминал+Chr(13)+"Кол-во не введено";
  КонецЕсли;
  Возврат "ADR="+Терминал+Chr(13)+"Кол-во "+Код+" Верно ?";
КонецЕсли;

Если Событие = 6 Тогда
  СписокТоваровМассив[Терминал].ДобавитьЗначение(КодТовара);
  СписокТоваров[Терминал].ПолучитьЗначение(СписокТоваров[Терминал].РазмерСписка());
  СписокКолвоТоваров[Терминал].ДобавитьЗначение(Количество);
  СписокДатаПокупки[Терминал].ДобавитьЗначение(ДатаПокупки);
  СписокВремяПокупки[Терминал].ДобавитьЗначение(ВремяПокупки);
  Возврат "ADR="+Терминал+Chr(13)+"Вес сохранен";
КонецЕсли;

Если Событие = 7 Тогда
  Док = СоздатьОбъект("Документ.РасходнаяНакладная");
  Контрагенты = СоздатьОбъект("Справочник.Контрагенты");
  Номенклатура = СоздатьОбъект("Справочник.Номенклатура");

```

```
Весовщики = СоздатьОбъект("Справочник.Весовщики");
Док.Новый();
Если Весовщики.НайтиПоКоду(СписокВесовщики[Терминал])=1 Тогда
  Док.Весовщик=Весовщики.ТекущийЭлемент();
Иначе
  Сообщить("Весовщик не найден, код " + СписокВесовщики[Терминал], "!");
  Возврат "ADR="+Терминал+Chr(13)+"Весовщик не найден, код ";
КонецЕсли;
```

```
Для Сч = 1 По СписокТоваровМассив[Терминал].РазмерСписка() Цикл
  Док.НоваяСтрока();
```

```
  КодТовара = СписокТоваровМассив[Терминал].ПолучитьЗначение(Сч);
  Если Номенклатура.НайтиПоКоду(КодТовара) = 1 Тогда
    Док.Товар = Номенклатура.ТекущийЭлемент();
  Иначе
    Сообщить("Товар не найден, код "+КодТовара, "!");
    Возврат "ADR="+Терминал+Chr(13)+"Товар не найден, код ";
  КонецЕсли;
  Док.Количество = СписокКолвоТоваров[Терминал].ПолучитьЗначение(Сч);
КонецЦикла;
Док.Записать();
Возврат "ADR="+Терминал+Chr(13)+"Документ сохранен";
КонецЕсли;
```

```
Если Событие = 8 Тогда
  СписокДокументы[Терминал] = "";
  СписокВесовщики[Терминал] = "";
  СписокКлиентовМассив[Терминал] = "";
  СписокТоваровМассив[Терминал].УдалитьВсе();
  СписокКолвоТоваров[Терминал].УдалитьВсе();
  СписокДатаПокупки[Терминал].УдалитьВсе();
  СписокВремяПокупки[Терминал].УдалитьВсе();
  Возврат "ADR="+Терминал+Chr(13)+"Работа завершена";
КонецЕсли;
```

```
Если (Событие = 10) or (Событие = 11) Тогда
  Если (Код = "") or (Код = "0") Тогда Возврат "ADR="+Терминал+Chr(13)+"Код не введен";
  Иначе Возврат "ADR="+Терминал+Chr(13)+"Событие "+Событие+" Код "+Код;
  КонецЕсли;
КонецЕсли;
```

```
КонецФункции
//Действия, выполняемые в предопределенной процедуре 1С-Предприятия
Процедура ПриНачалеРаботыСистемы() Экспорт
  Перем i;
  Для i=1 По 4 Цикл
    СписокДокументы[i]=СоздатьОбъект("СписокЗначений");
    СписокВесовщики[i]=СоздатьОбъект("СписокЗначений");
    СписокТоваровМассив[i]=СоздатьОбъект("СписокЗначений");
    СписокКолвоТоваров[i]=СоздатьОбъект("СписокЗначений");
    СписокДатаПокупки[i]=СоздатьОбъект("СписокЗначений");
    СписокВремяПокупки[i]=СоздатьОбъект("СписокЗначений");
  КонецЦикла;
```

```
КонецПроцедуры
```

**Внимание ! Имя метода предопределено и не может быть произвольным. Метод, экспортируемый внешней системой, полностью обеспечивается Заказчиком, экспортируемые ПРОГРАММОЙ методы полностью обеспечиваются Исполнителем.**

**Программный комплекс должен обеспечивать следующий порядок работы системы :**

ВЕСОВЩИК включает ТЕРМИНАЛ, ТЕРМИНАЛ выполняет обычную самопроверку готовности к работе, предусмотренную техническим паспортом изделия. Если самотестирование ТЕРМИНАЛА завершается успешно, то ТЕРМИНАЛ переходит в режим обычного взвешивания.

Дальнейшее управление ТЕРМИНАЛОМ выполняет ПРОГРАММА, запущенная на ПЭВМ ОПЕРАТОРОМ.

ПРОГРАММА должна обеспечивать поддержку не более 9-и стандартных событий (с 1 по 9), генерируемых по нажатию соответствующих цифровых клавиш ТЕРМИНАЛА. События, за исключением события 1, полностью определяются ЗАКАЗЧИКОМ в соответствии с собственными нуждами и требуемым алгоритмом работы. Информация по самостоятельному описанию событий приведена ниже.

#### Установленные ИСПОЛНИТЕЛЕМ события для приведенного в данном документе примера конфигурирования 1С-Предприятия и описания файла конфигурации ПРОГРАММЫ.

1. Авторизация весовщика. Данное событие зарезервировано и не может быть переназначено.
2. Ввод кода документа «Заявка на отгрузку».
3. Ввод кода продукта.
4. Ввод количества печатаемых этикеток.
6. Завершение работы с продуктом (подтверждение веса текущего выбранного продукта).
7. Завершение работы с текущим документом.
8. Завершение работы весовщика.
9. Расширение кодов событий.
10. Ввод кода покупателя.
11. Проверочное событие.

Любое событие подтверждается ВЕСОВЩИКОМ нажатием клавиши «Ввод» на клавиатуре терминала. При вводе любого кода ВЕСОВЩИК может отказаться от последней введенной цифры (удалить последнюю цифру кода), нажатием клавиши «>0<», или полностью осуществить сброс введенного кода нажатием клавиши «С», а также полностью отказаться от сгенерированного события (например, событие сгенерировано ВЕСОВЩИКОМ ошибочно) нажатием клавиши «Б/Н». Порядок следования событий, а также их допустимость в конкретном контексте полностью определяются и обеспечиваются ЗАКАЗЧИКОМ. После нажатия клавиши «Ввод» на клавиатуре терминала, ПРОГРАММА осуществляет вызов внешнего метода EventsHandler. Актуальность того или иного параметра вызова полностью зависит от возникшего события и отслеживается ЗАКАЗЧИКОМ. Например, при возникновении события 8 в приведенном примере конфигурирования не актуальны параметры Cod и Weight. Возвращенное данной функцией значение ПРОГРАММА выводит в нижнюю строку индикатора ТЕРМИНАЛА или на принтер и переходит в режим ожидания возникновения следующего события.

Во всех режимах ввода кода, при попытке ВЕСОВЩИКА ввести более чем шестизначный код, ПРОГРАММА игнорирует вводимые ВЕСОВЩИКОМ цифры с седьмой (по порядку) и более.

#### **ПОРЯДОК ЗАПУСКА ПРОГРАММЫ.**

Запуск ПРОГРАММЫ выполняется ОПЕРАТОРОМ ПЭВМ. Непосредственно после запуска, ПРОГРАММА выполняет поиск файла конфигурации settings.ini. Поиск файла конфигурации выполняется ПРОГРАММОЙ в том же каталоге из которого ПРОГРАММА запускается. В случае обнаружения файла конфигурации, ПРОГРАММА выполняет считывание и проверку корректности значений параметров, указанных в файле. В случае не корректности какого-либо из параметров, равно как и его отсутствия в файле конфигурации, ПРОГРАММА использует умалчиваемое значение. В случае отсутствия файла конфигурации, ПРОГРАММА создает таковой с параметрами, определенными по умолчанию. В рабочем окне ПРОГРАММЫ должна обеспечиваться возможность просмотра всей истории работы ПРОГРАММЫ, с момента ее запуска ОПЕРАТОРОМ ПЭВМ.

#### **Пример файла конфигурации ПРОГРАММЫ.**

Файл конфигурации settings.ini является обычным текстовым файлом, редактировать который возможно любым текстовым редактором. Например, NotePad.

Файл конфигурации содержит три вида информации:

##### 1. Комментарии

Комментарием считается ЛЮБАЯ строка, не являющаяся секцией или строкой конфигурационного параметра.

Строка комментария игнорируется ПРОГРАММОЙ (считается не значащей).

Строки с комментариями могут следовать в любом месте в файле конфигурации.

##### 2. Секции.

Секцией считается строка, обрамленная в квадратные скобки.

В начале строки, до символа "[" (открывающая квадратная скобка), равно как и в конце строки после символа "]" (закрывающая квадратная скобка), могут присутствовать пробелы, но другие символы не допускаются.

Ведущие и концевые пробелы в строке секции игнорируются ПРОГРАММОЙ.

Секции могут следовать в любом порядке в файле конфигурации.

##### 3. Строки с конфигурационными параметрами.

Строка с конфигурационными параметрами считаются строки, которые содержат ключевые слова и не являются секциями или комментариями.

Каждый конфигурационный параметр представляет строку вида

Имя\_параметра=Значение\_Параметра

Пробелы между **Имя параметра** , знаком равенства и **Значение\_Параметра** недопустимы. **Имя параметра** является зарезервированным словом. В начале строки, до **Имя параметра**, равно как и в конце строки, после **Значение\_Параметра** могут присутствовать пробелы, но другие символы не допускаются. Строки с конфигурационными параметрами могут следовать в любом порядке в пределах своей секции, т.е. нет жесткого порядка следования строк с ключевыми словами. Ошибочно заданные или отсутствующие параметры ПРОГРАММА заменяет на значения по умолчанию.

Начало примера файла конфигурации

#### [Общие]

##### **Имя сервера OLE=V1CEnterprise.Application**

;Данный параметр определяет используемую версию 1С:Предприятия  
; или имя любого другого сервера, экспортирующего метод EventsHandler  
;Для 1С:Предприятия версии 7.7 возможны следующие значения :  
;V1CEnterprise.Application - версия независимый ключ  
;V77.Application - версия зависимый ключ  
;V77S.Application - версия зависимый ключ, SQL-версия  
;V77L.Application - версия зависимый ключ, локальная версия  
;V77M.Application - версия зависимый ключ, сетевая версия  
; Для 1С:Предприятия версии 8.0  
; V8.Application

##### **Строка инициализации=/DE:\Baza1C\DB\ /NАдминистратор /P**

; Данный параметр актуален только при использовании 1С:Предприятия  
; Ключи : /D – путь к базе данных, /N – имя пользователя, /P – пароль  
; Для 1С:Предприятия версии 8.0  
;Строка инициализации представляет собой набор параметров, каждый из которых представляет собой фрагмент вида  
;*Имя параметра=Значение*, где *Имя параметра* — имя параметра, а *Значение* — его значение.  
;Фрагменты отделяются друг от друга символами ';. Если значение содержит пробельные символы, то оно  
;должно быть заключено в двойные кавычки (").Для файлового варианта определен параметр:  
;*File* — каталог информационной базы (файловый режим);  
;Для клиент-серверного варианта определены параметры:  
;*Srvr* — имя сервера 1С:Предприятия;  
;*Ref* — имя информационной базы на сервере;  
;Для всех вариантов определены параметры:  
;*Usr* — имя пользователя;  
;*Pwd* — пароль.  
; После разделителя \*(звездочка) указывается дополнительный символ, который показывает/скрывает пользовательский  
;интерфейс 1С:Предприятия : 0 – скрыть, 1 – показать. По умолчанию 0.  
; Пример : **Строка инициализации=File=E:\Baza1C\DB ; Usr=Администратор ; \*0**

##### **Мониторинг(1-да,0-нет)=0**

;Данный параметр используется для получения информации обмена с СОМ-портом  
;в случае возникновения проблем связи с терминалом. Используется только при необходимости

#### [Локальная сеть]

##### **Использование лвс=1**

;1 – связь по лвс, 0 – связь по RS

##### **Порт=9002**

; Для RS-интерфейса данный параметр не актуален

#### [Настройка оборудования]

##### **Количество каналов=1**

;Количество используемых СОМ-портов. Возможные значения : 1 или 2.

##### **Порт первого канала=1**

;Физический № СОМ-порта, связываемый с каналом.

##### **Скорость обмена первого канала=9600**

##### **Кол-во устройств первого канала=4**

;Данный параметр должен содержать общее число устройств,  
;подключенных к физическому каналу связи.

##### **Тип индикатора терминалов первого канала=0**

;Данный параметр определяет тип установленного  
;в терминалах индикатора

;0 – ЖКИ

;1 – люминесцентный(для ТВ-018 тип индикатора - 0)

##### **Адреса терминалов первого канала=1,2**

;Адреса терминалов перечисляются через запятую.

##### **Считыватели первого канала= 8388609/3,8388610/4**

;Строка описания адреса контроллеров считывателя\принтера  
;должна иметь следующий формат : **SN/Adr**, где **SN** – серийный номер  
;контроллера(назначается заводом изготовителем и указывается непосредственно на изделии),  
; **Adr** – адрес, присваиваемый контроллеру на сеанс работы.  
;При использовании более одного контроллера, строки описания адреса должны следовать

;через запятую. Если контроллеры не используются, данный параметр не должен содержать  
;никаких значений.  
;Важно !!!! Адреса терминалов и присваиваемые адреса контроллеров, находящихся на одном  
;канале НЕ ДОЛЖНЫ совпадать.

**Порт второго канала=2**

**Скорость обмена второго канала=9600**

**Кол-во устройств второго канала=2**

**Тип индикатора терминалов второго канала=1**

**Адреса терминалов второго канала=5,6**

**Считыватели второго канала=**

**Таймерные тики(18 тиков - 1 сек)=1**

;Временной интервал, по истечению которого в терминал будет выдана команда

;в случае отсутствия ответа от последнего. Для RS-интерфейса

;обычно 1-2, для локальной сети подбирается эмпирически.

**Задержка(мск)=100**

;Определяет задержку, которую необходимо сделать перед

;вызовом метода EventHandler. Подбирается эмпирически.

**Количество команд на которые устройство не ответило=5**

;При превышении значения данного параметра программа будет считать, что связь

;с устройством потеряна.

**Количество обращений к принтеру=40**

**[События]**

**Количество событий=12**

; Формат строки описания событий :

; T,строка

; где T - тип события : 0 - информационное событие

; 1 - событие ввода некоторого числа

; 2 - расширение кодов событий

; строка - выводимое сообщение на индикатор терминала

; для T = 1 и 2 должно быть не более 14 символов,

; для T = 0 должно быть не более 20 символов

**Событие0=1,""**

;Событие 0 не генерируется терминалом и используется

;в качестве шлюза для расширенных событий.

**Событие1=1,Код оператора \_\_\_\_\_**

**Событие2=1,Код документа**

**Событие3=1,Код продукта**

**Событие4=1,Кол.этикеток**

**Событие5=**

**Событие6=0,Сохранить вес ?**

**Событие7=0,Сохранить документ ?**

**Событие8=0,Завершить работу ?**

**Событие9=2,Код события**

**Событие10=1,Покупатель**

**Событие11=0,Проверимся ?**

Конец примера файла конфигурации ПРОГРАММЫ

Исполнитель :

Начальник ОИТ Батырева О.Г.

Разработчик Моисеев С.Л.

Заказчик :