



## **ПРОГРАММИРУЕМЫЕ ЛОГИЧЕСКИЕ КОНТРОЛЛЕРЫ**

### **Библиотека SmsOwenLib для CoDeSys v2.3**

### **Руководство по применению**

**Версия 02**

**Москва**

**2012**

## Содержание

Введение.....	3
Словарь условных сокращений и терминов .....	4
1. Установка дополнительных библиотек для проекта.....	4
2. Состав библиотеки SmsOwenLib. ....	8
<i>Блок «OpenPort» захвата/освобождения интерфейса .....</i>	<i>8</i>
<i>Блок «FB_SMS_CFG» конфигурирования модема .....</i>	<i>9</i>
<i>Блок «FB_SMS_SR» отправки/ получения СМС .....</i>	<i>11</i>
<i>Блок «FB_SMS_SL» вспомогательный для отправки сообщения группе абонентов .....</i>	<i>13</i>
Приложение А. Значение кода аварий ErrCode .....	Ошибка! Закладка не определена.
Лист изменений в версиях документа .....	15

## Введение

Компания ОВЕН предоставляет пользователю библиотеки дополнительных программных компонентов, облегчающие составление проекта работы программируемого логического контроллера (ПЛК) для решения наиболее распространенных практических задач. Приводимая в данном описании, библиотека предназначена для работы на контроллерах ОВЕН с внешним модемом: для отправки и получения СМС сообщений, фиксации входящего вызова и первоначальной конфигурации самого модема.

Библиотеки поставляются в виде файлов на компакт-диске, входящем в комплект поставки ОВЕН ПЛК (папка «Lib\Библиотеки ОВЕН»).

**Внимание!** У программных компонентов библиотек режим симуляции (Simulation Mode) не предусмотрен. Отладка программы проводится при подключенном контроллере, – программные компоненты при этом работают только в самом контроллере.

## Словарь условных сокращений и терминов

Далее в тексте для компактного описания используются следующие сокращения:

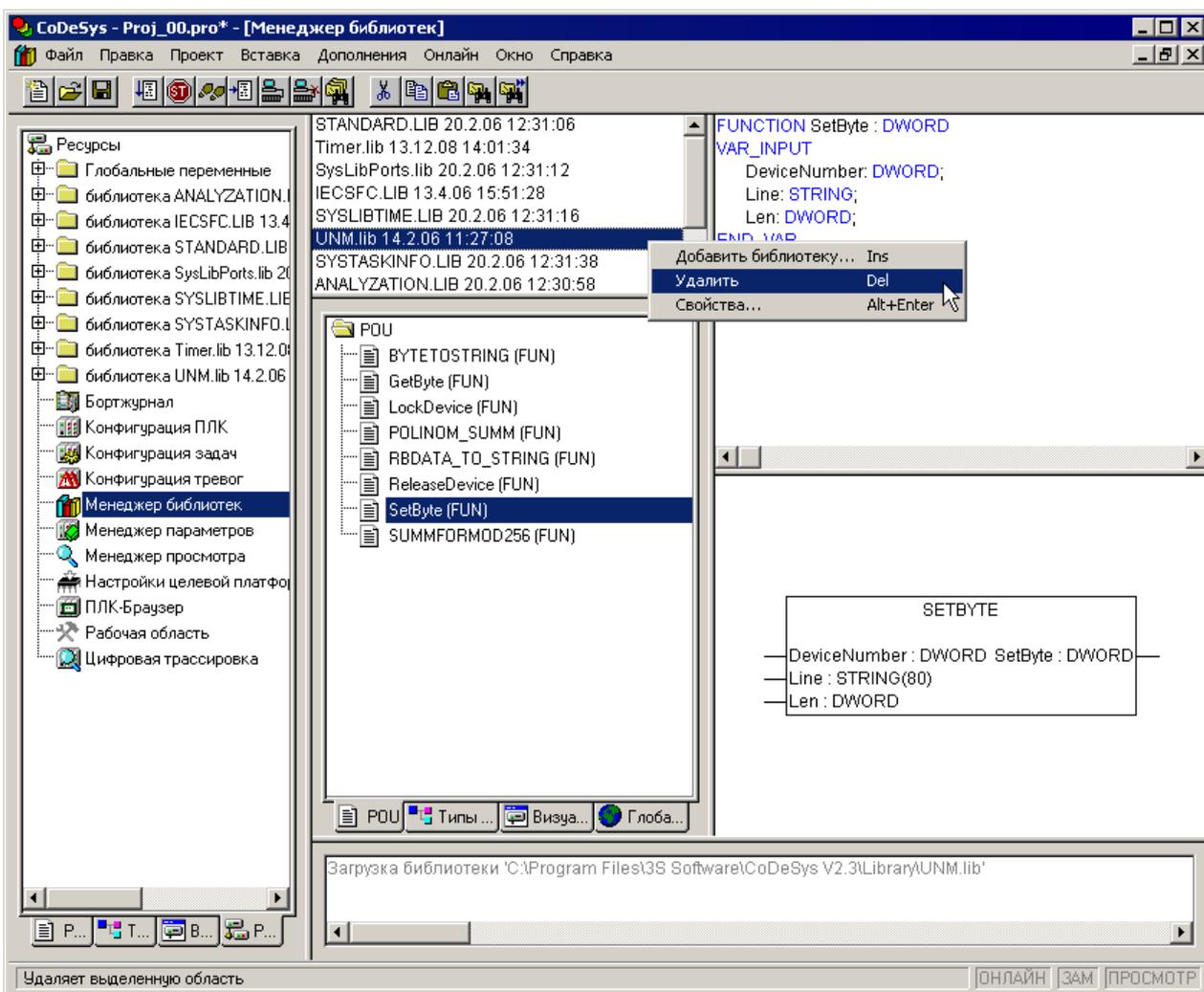
- CoDeSys** – Controllers Development System, программное обеспечение, специализированная среда программирования логических контроллеров. Торговая марка компании 3S-Smart Software Solutions GmbH.
- ST** – Structured Text, структурированный текст, язык программирования по МЭК 61131-3.
- ПЛК** – программируемый логический контроллер.
- РП** – руководство пользователя на программируемый логический контроллер (поставляется на компакт-диске вместе с контроллером).
- 0 и 1** – при описании переменных типа BOOL нулю соответствует значение «FALSE»; единице – значение «TRUE».

### 1. Установка дополнительных библиотек для проекта

В CoDeSys все файлы библиотек дополнительных программных компонентов имеют расширения \*.lib (Library) и находятся в папке Library. Она расположена по месту размещения основной программы на диске компьютера (по умолчанию – C:\Program Files\3S Software\CoDeSys V2.3\Library).

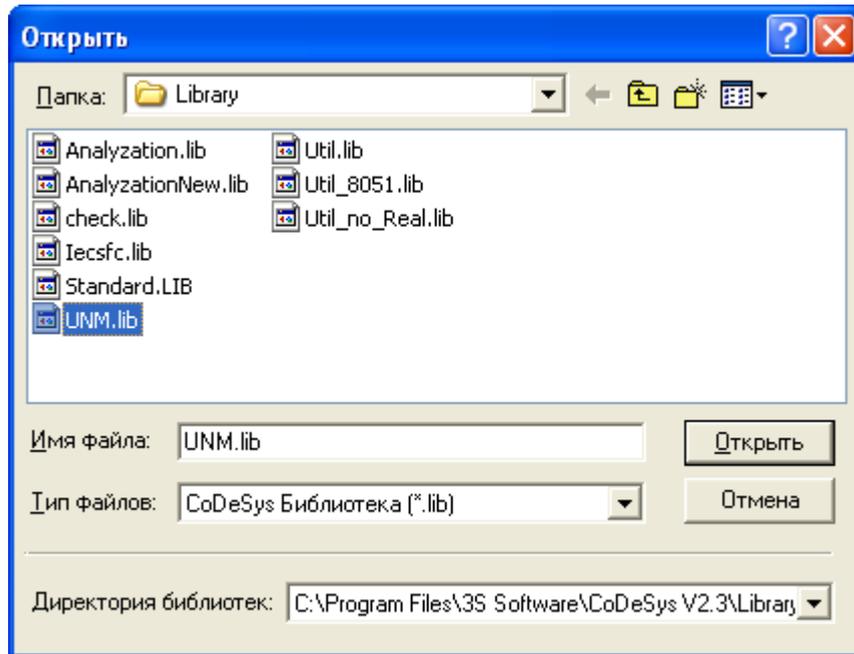
По умолчанию подключен (доступен) стандартный набор библиотек. Дополнительные библиотеки добавляются пользователем по мере необходимости в папку к уже имеющимся библиотекам. Для подключения новых библиотек к проекту соответствующие файлы переписываются пользователем в ту же папку, где находятся все используемые библиотеки.

Чтобы увидеть, какие библиотеки уже подключены к проекту, и подключить дополнительные библиотеки, используется «Менеджер библиотек (Library Manager)», – его можно открыть из главного меню CoDeSys командами «Окно (Window) ▶ Менеджер библиотек (Library Manager)» или выбором на вкладке организатора объектов «Ресурсы (Resources)» режима работы «Менеджер библиотек (Library Manager)», см. рисунок 1.1. В средней верхней части появившегося окна отображается список установленных библиотек.



**Рисунок 1.1 – Окно вкладки организатора объектов «Ресурсы (Resources)» с режимом работы «Менеджер библиотек (Library Manager)»**

Установка дополнительных библиотек выполняется из главного меню последовательным выбором команд: **Вставка (Insert) ▶ Добавить библиотеку (Additional Library) ▶** в открывшемся окне папки Library (рисунок 1.2) выделяется файл с именем нужной библиотеки (например, UNM.lib) и дается команда **Открыть**.



**Рисунок 1.2 – Окно выбора подключаемой к проекту дополнительной библиотеки**

Теперь в перечне библиотек, доступных в проекте, появится вновь установленная библиотека.

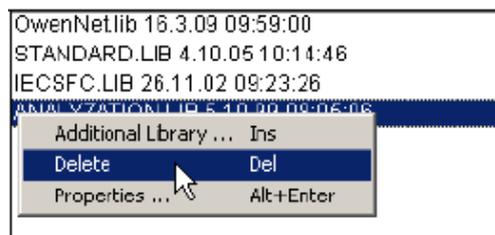
Для просмотра состава и свойств программных компонентов курсором выбирается нужная библиотека, – при этом появится папка с программными компонентами, в которой выделяется конкретный программный компонент (на рисунке 1.1 справа дана краткая справочная информация по его использованию).

#### **Примечания.**

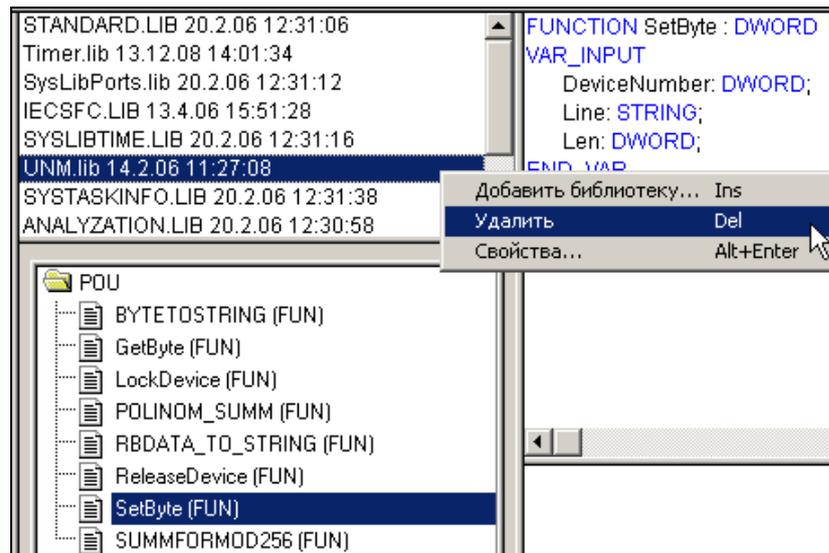
1. Рекомендуется размещать все библиотеки, которые планируется подключать, в папке для хранения библиотек, создаваемой CoDeSys автоматически.

2. Для каждого нового проекта добавление новых библиотек проводится индивидуально, при необходимости их применения.

Удаление выделенной библиотеки выполняется из контекстного меню командой **Удалить (Delete)** (или из главного меню командой **Правка (Edit) ▶ Удалить (Delete)**, рисунок 1.3 (или нажатием клавиши <Delete>).



a)



в)

**Рисунок 1.3 – Удаление дополнительной библиотеки:**  
а) для CoDeSys с английским интерфейсом; в) для CoDeSys с русским интерфейсом

## 2. Состав библиотеки SmsOwenLib.

Данная библиотека предназначена для работы на контроллерах ОВЕН с внешним модемом: для отправки и получения СМС сообщений, фиксации входящего вызова и первоначальной конфигурации самого модема.

### **Блок «OpenPort» захвата/освобождения интерфейса**

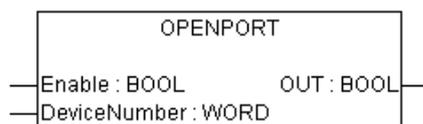


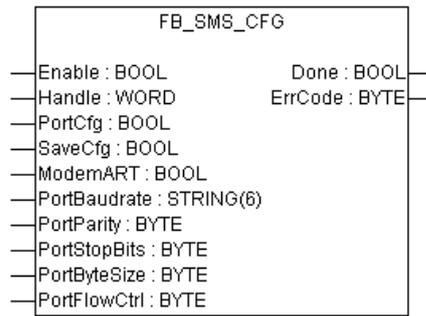
Рисунок 2.1 – Структурная схема

Таблица 2.1

<b>Имя программного компонента</b>	<b>OpenPort</b>		
<b>Тип программного компонента</b>	Функциональный блок <input checked="" type="checkbox"/>	Функция <input type="checkbox"/>	Программа <input type="checkbox"/>
<b>Особенности работы</b>	Для работы требуется установка в проекте библиотеки UNM		
<b>Применение на контроллерах</b>	ПЛК100, ПЛК150, ПЛК154, ПЛК110, ПЛК160.		
<b>Входная переменная:</b>	<b>Тип данных</b>	<b>Пояснения</b>	
<b>Enable</b>	BOOL	разрешение на работу	
<b>DeviceNumber</b>	WORD	номер модуля UNM – нумерация идет с 0 по порядку сверху вниз как расположены модули UNM в конфигурации, первый сверху модуль UNM имеет номер 0, следующий номер UNM – 1, и так далее	
<b>Выходная переменная:</b>	<b>Тип данных</b>	<b>Пояснения</b>	
<b>Out</b>	BOOL	состояние открытия порта, после окончания включен постоянно	

### **Описание**

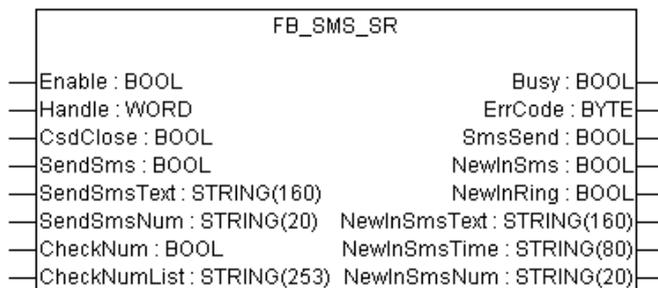
До начала передачи/приема в интерфейс данных из программы CoDeSys нужно «захватить» интерфейс и приостановить работу с этим интерфейсом модулей протоколов, подключенных в PLC Configuration. Для использования нужно создать в конфигурации контроллера один (или несколько) модулей Universal Network Module. Внутри каждого модуля нужно настроить параметры порта передачи данных (для последовательных портов это скорость, четность, биты данных, биты стопа, время задержки).

**Блок «FB\_SMS\_CFG» конфигурирования модема****Рисунок 2.2 – Структурная схема****Таблица 2.2**

<b>Имя программного компонента</b>	FB_SMS_CFG		
<b>Тип программного компонента</b>	Функциональный блок <input checked="" type="checkbox"/>	Функция <input type="checkbox"/>	Программа <input type="checkbox"/>
<b>Особенности работы</b>	Для работы требуется установка в проекте библиотеки UNM		
<b>Применение на контроллерах</b>	ПЛК100, ПЛК150, ПЛК154, ПЛК110, ПЛК160.		
<b>Входная переменная:</b>	<b>Тип данных</b>	<b>Пояснения</b>	
Enable	BOOL	разрешение на работу	
Handle	WORD	номер модуля UNM, порт открывается с помощью ФБ «OpenPort»	
PortCfg	BOOL	разрешение на настройку порта модема, по умолчанию False (выкл.)	
SaveCfg	BOOL	сохранение параметров в модеме после конфигурирования, по умолчанию False (не сохранять)	
ModemART	BOOL	включение/отключение автоподъема трубки, по умолчанию False (выкл.),	
PortBaudrate	STRING(6)	параметр настройки порта - скорость, по умолчанию '0'	
PortParity	BYTE	параметр настройки порта - четность, 0- нет 1-нечет 2-чет, по умолчанию 0-нет	
PortStopBits	BYTE	параметр настройки порта – число стоп-бит, 1-1бит, 3-2бита, по умолчанию 1-1бит	
PortByteSize	BYTE	параметр настройки порта - Биты данных, по умолчанию 8-8 бит	
PortFlowCtrl	BYTE	параметр настройки порта - управление потоком, 0 – отключено, 1-XON/XOFF, 2-аппаратный, по умолчанию 0-отключено	
<b>Выходная переменная:</b>	<b>Тип данных</b>	<b>Пояснения</b>	
Done	BOOL	флаг окончания настройки, после окончания включен постоянно	
ErrCode	BYTE	код ошибки	

### **Описание работы**

Необходим для конфигурирования модема. Должен вызываться до блока FB\_SMS\_SR. Если модем уже был настроен ранее (вручную или с помощью данного ФБ с **SaveCfg:=TRUE**) и в ходе работы настройки модема меняться не будут, то разрешается не использовать данный ФБ в программе.

**Блок «FB\_SMS\_SR» отправки/ получения СМС****Рисунок 2.3 – Структурная схема****Таблица 2.3**

<b>Имя программного компонента</b>	<b>FB_SMS_SR</b>		
<b>Тип программного компонента</b>	Функциональный блок <input checked="" type="checkbox"/>	Функция <input type="checkbox"/>	Программа <input type="checkbox"/>
<b>Особенности работы</b>	Для работы требуется установка в проекте библиотеки UNM		
<b>Применение на контроллерах</b>	ПЛК100, ПЛК150, ПЛК154, ПЛК110, ПЛК160.		
<b>Входная переменная:</b>	<b>Тип данных</b>	<b>Пояснения</b>	
<b>Enable</b>	BOOL	разрешение на работу	
<b>Handle</b>	WORD	номер модуля UNM, порт открывается с помощью ФБ «OpenPort»	
<b>CsdClose</b>	BOOL	запрет входящих вызовов: определяет поведение при входящем звонке, если True(вкл.), то всегда вешает трубку, если False(выкл.), то выставляет флаг NewInRing, по умолчанию True (вкл.)	
<b>SendSms</b>	BOOL	команда на отправку SMS сообщения, срабатывает по фронту	
<b>SendSmsText</b>	STRING	текст отправляемого сообщения (на русском макс 70 символов, на латинице макс 140 символов)	
<b>SendSmsNum</b>	STRING	номер абонента, кому отправить СМС, пишется целиком с кодом страны без скобок и «+», например «79012345678»	
<b>CheckNum</b>	BOOL	включить/отключить проверку номера отправителя СМС (для входящих сообщений), по умолчанию False (выкл)	
<b>CheckNumList</b>	STRING	список разрешенных номеров, формат: «xxxxxxxxxxx;xxxxxxxxxxx;» номер пишется целиком без скобок и «+», через точку с запятой, если функция выключена то не используется, по умолчанию ‘ ’	
<b>Выходная переменная:</b>	<b>Тип данных</b>	<b>Пояснения</b>	
<b>Busy</b>	BOOL	флаг указывающий, что блок выполняет в данный момент какие либо операции	

<b>ErrCode</b>	BYTE	код ошибки
<b>SmsSend</b>	BOOL	флаг успешной отправки СМС сообщения, включается на 1н цикл
<b>NewInSms</b>	BOOL	флаг наличия нового входящего СМС сообщения, включается на 1н цикл
<b>NewInRing</b>	BOOL	флаг наличия входящего вызова
<b>NewInSmsText</b>	STRING	текст последнего принятого СМС сообщения
<b>NewInSmsTime</b>	STRING	время отправки последнего входящего СМС сообщения
<b>NewInSmsNum</b>	STRING	номер отправителя последнего входящего СМС сообщения, 10 цифр

### **Описание работы**

Необходим для отправки/получения СМС сообщений и индикации входящих вызовов

## Блок «FB\_SMS\_SL» вспомогательный для отправки сообщения группе абонентов

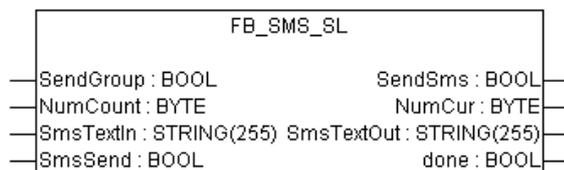


Рисунок 2.4 – Структурная схема

Таблица 2.4

Имя программного компонента	FB_SMS_SL		
Тип программного компонента	Функциональный блок <input type="checkbox"/>	Функция <input checked="" type="checkbox"/>	Программа <input type="checkbox"/>
Особенности работы	Для работы не требуется установка в проекте дополнительных библиотек		
Применение на контроллерах	ПЛК63, ПЛК73, ПЛК410		
<b>Входная переменная:</b>	<b>Тип данных</b>	<b>Пояснения</b>	
SendGroup	BOOL	команда на отправку группе	
NumCount	BYTE	количество адресатов	
SmsTextIn	STRING	текст отправляемого сообщения	
SmsSend	BOOL	контроль за отправкой сообщений	
<b>Выходная переменная:</b>	<b>Тип данных</b>	<b>Пояснения</b>	
SendSms	BOOL	команда на отправку сообщения	
NumCur	BYTE	текущий номер адресата	
SmsTextOut	STRING	текст отправляемого сообщения	
Done	BOOL	флаг завершения работы, включается на 1н цикл	

### Описание работы

По фронту сигнала на входе «SendGroup» БФ запоминает текст отправляемого сообщения, который подается на вход «SmsTextIn», и выдает его на выход «SmsTextOut» (после этого значение на входе «SmsTextIn» может меняться – на работу блока это никак не повлияет). Вход «SmsSend» и выход «SendSms» необходимо соединить с одноименными входами и выходами ФБ «FB\_SMS\_SR» - с их помощью будет осуществляться управление отправкой сообщений. Выход «NumCur» определяет порядковый номер текущего адресата, нумерация начинается с нуля. Когда сообщение будет разослано, на выход «Done» будет подан единичный импульс.

## Приложение А. Коды ошибок ErrCode

Текстовое обозначение	Код	Описание
NORMA	0	Ошибок нет
UNDEFINED	99	Неопознанная ошибка
SIMPL_REQ_UNO	1	несмог установить команду АТЕ0
SIMPL_REQ_DUO	2	несмог установить параметр ICF (определение рамок стартового и стопового бита)
SIMPL_REQ_TRE	3	несмог установить параметр IFC (управление потоком)
SIMPL_REQ_QUA	4	несмог установить параметр IPR (скорость передачи данных)
SIMPL_REQ_VIS	5	неудалось сохранить параметры в энергонезависимую память
SIMPL_REQ_SEI	6	неудалось выполнить команду АТ
SIMPL_REQ_SET	7	неудалось проверить регистрацию в сети устройства
SIMPL_REQ_OTT	8	неудалось проверить статус активности устройства
SIMPL_REQ_NEU	9	неудалось проверить баланс
SIMPL_REQ_DEC	10	неудалось добиться ответа при отправке смс
SIMPL_REQ_SSE	26	неудалось отправить смс, ошибка кодировки
ME_NOREG	11	регистрация показывает что устройство не в домашней сети
ME_BUSY	12	статус активности показывает что устройство занято
TIMEOUT	13	таймаут
SMS_DEL	21	возникла ошибка при удалении всех смс
SMS_READ	22	возникла ошибка при чтении смс
SMS_CLEAN	23	по данному индексу смс отсутствует (ячейка пустая)
CUSD_NOT_OK	24	неудалось проверить баланс
SMS_SINGL_DEL	25	возникла ошибка при удалении всех смс
CALL_NOT	31	несмог проверить состояние вызовов телефона

## Приложение Б. Примеры программ.

### Пример №0. Открытие порта и настройка модема.

Вначале необходимо добавить в конфигурацию ПЛК модуль Universal Network Module, указать к какому интерфейсу будет подключен модем. Затем произвести настройку порта (указать скорость, число стоп-битов и т.д.), рис.1. Режим можно указывать любой, как ASCII так и RTU.

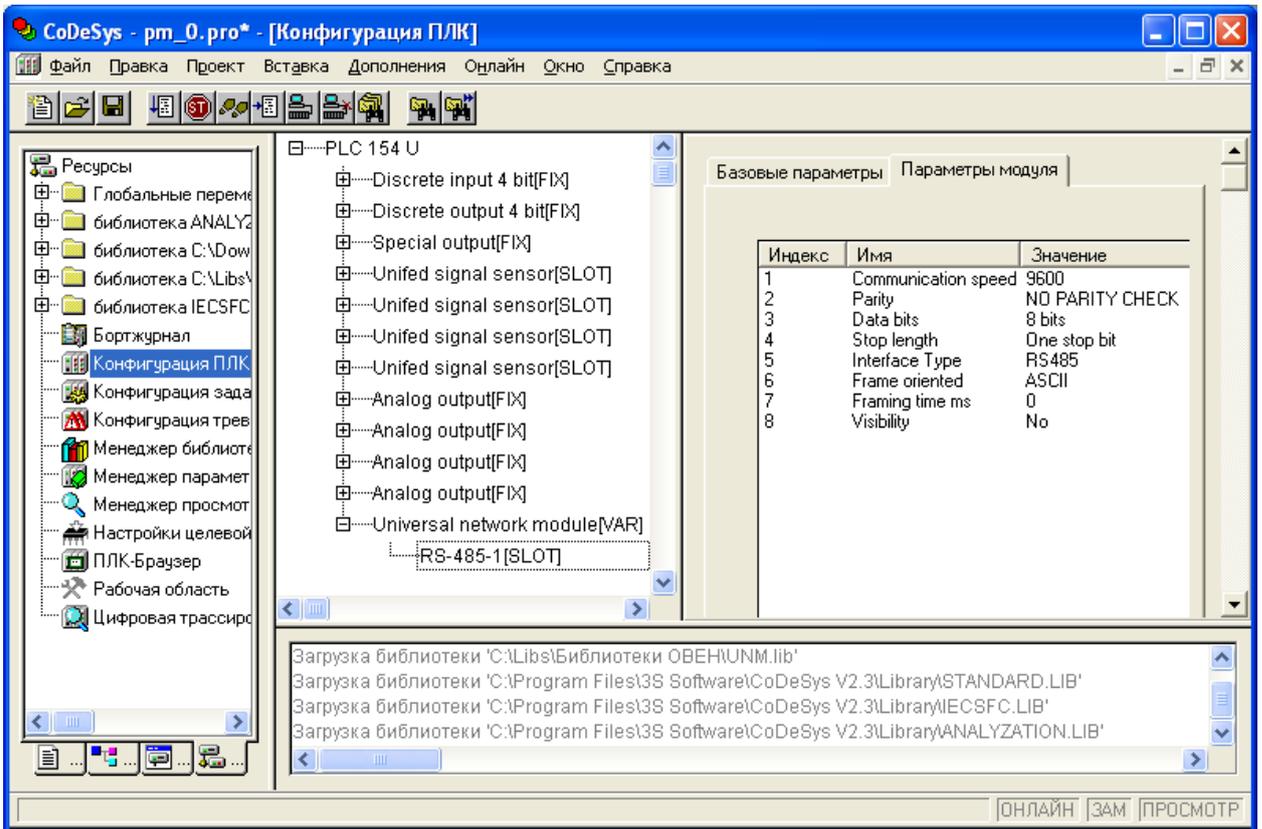


Рис.1. Настройка модуля Universal Network Module.

Программа:

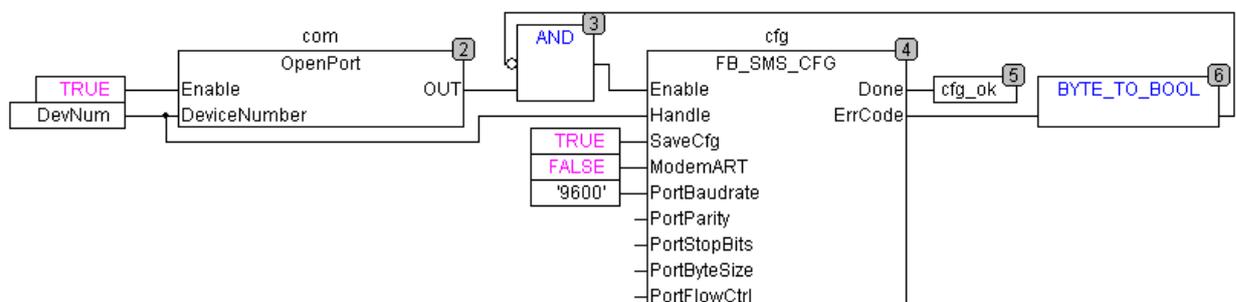


Рис.2. Алгоритм работы программы.

Список переменных:

PROGRAM PLC\_PRG

VAR

```

Com      : OpenPort;
cfg      : FB_SMS_CFG;

cfg_ok   : BOOL;

```

```
    DevNum    :WORD    :=0;  
END_VAR
```

В данном примере порт будет открыт всегда, при необходимости освободить его достаточно подать на вход «**Enable**» ФБ «**com**» значение *False*.

ФБ «**cfg**» по окончанию конфигурации модема установит на выходе «**Done**» значение *True* в не зависимости от того, были ли ошибки или нет, поэтому необходимо проверять значение на выходе «**ErrCode**». В данном примере в случае возникновения ошибки блок будет перезапущен и конфигурация модема будет произведена повторно.

Так же данной блок устанавливает кроме доступных пользователю для задания и другие параметры, необходимые для корректной работы модема и библиотеки (форматы ответа модема на команды, режимы отправки и получения СМС сообщений и т.д.).

Если модем уже настроен и его параметры не будут меняться, то можно не использовать данный ФБ в программе или вызывать его только один раз при подключении нового модема.

### Пример №1\_1. Отправка СМС сообщений.

Добавим к предыдущему примеру переменную (*var1*), которая будет постоянно расти и по достижению ею определенного уровня (*var1\_Alarm*) необходимо отправить сообщение. Текстом сообщения будет служить значение этой переменной, номер получателя заранее жестко задан.

Программа:

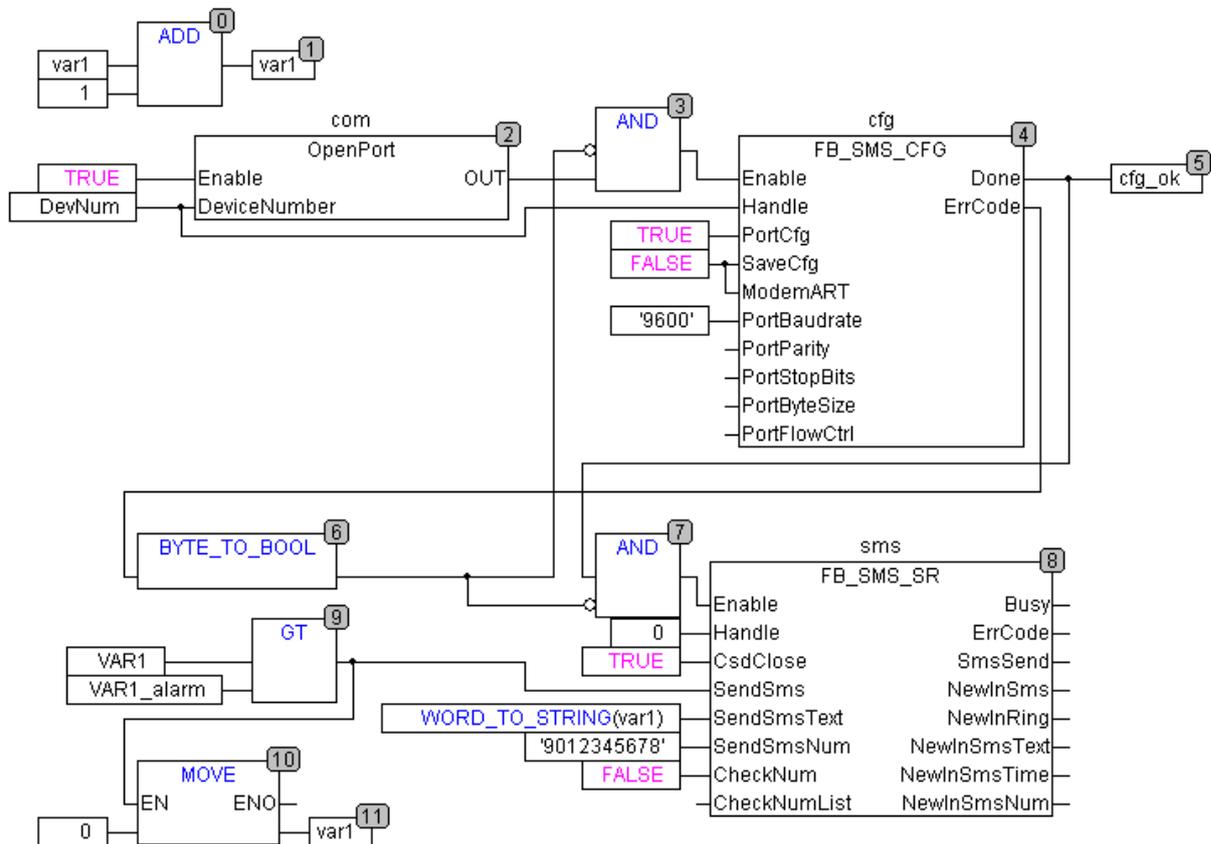


Рис.3. Алгоритм работы программы.

Список переменных  
PROGRAM PLC\_PRG

VAR

```

Com      : OpenPort;
cfg      :FB_SMS_CFG;
sms      :FB_SMS_SR;

cfg_ok   :BOOL;
DevNum   :WORD    :=0;

```

```

var1     :WORD; (*переменная, которая будет расти*)
var1_Alarm :WORD    :=5000; (*ее аварийный уровень*)

```

END\_VAR

После открытия порта и конфигурирования модема начинает работать блок отправки и получения СМС. Так как режим CSD передачи данных в данном примере не используется на вход «**CsdClose**» подается сигнал *False* (все входящие вызовы будут сброшены). Входящие СМС сообщения так же не используются, поэтому на входа «**CheckNumList**» и «**CheckNum**» можно ничего не подавать.

На вход «**SendSmsNum**» подается десятизначный номер получателя в формате строки (*String*). На вход «**SendSmsText**» подается преобразованной к типу *String* значение переменной «*var1*». По приходу на вход «**SendSms**» значения *True*, значения на двух предыдущих входах будут зафиксированы внутри ФБ. Как только сообщение будет отправлено на выход «**SmsSend**» будет подан единичный импульс (значение *True* будет установлено на один цикл).

## Пример №1\_2. Отправка СМС сообщений. Отправка нескольким получателям.

Пример аналогичен предыдущему №1\_1 с той разницей, что сообщение с одним и тем же текстом надо отправить нескольким получателям сразу. Для этого можно использовать ФБ «FB\_SMS\_SL», входящий в состав данной библиотеки.

Программа:

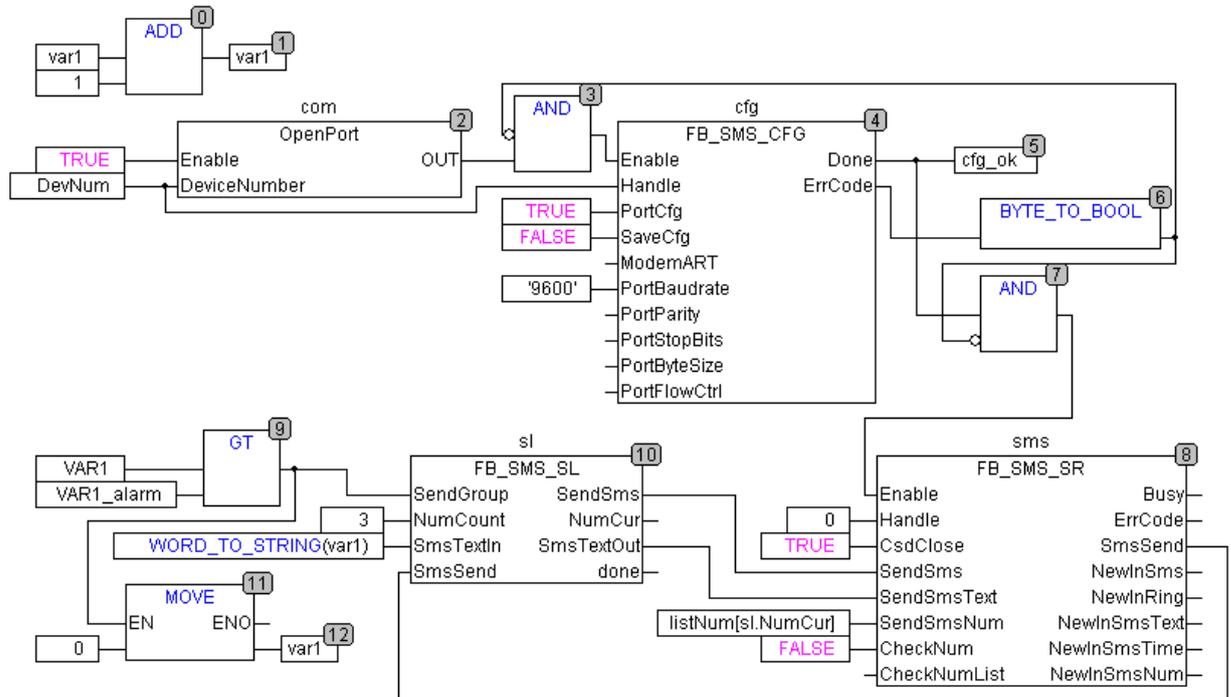


Рис.4. Алгоритм работы программы.

Список переменных  
PROGRAM PLC\_PRG  
VAR

Com :OpenPort;  
cfg :FB\_SMS\_CFG;  
sms :FB\_SMS\_SR;

cfg\_ok :BOOL;  
DevNum :WORD :=0;

var1 :WORD;  
var1\_Alarm :WORD :=5000;

sl :FB\_SMS\_SL; (\*дополнительный блок для отправки группе получателей\*)  
listNum :ARRAY [0..2] OF STRING(10):=3('79012345678'); (\*массив номеров получателей, обязательно с начинаться должен с нуля\*)  
END\_VAR

## Пример №2. Получение СМС сообщений.

В данном примере будет показано, как организовать удаленное управление при помощи СМС сообщений.

Программа:

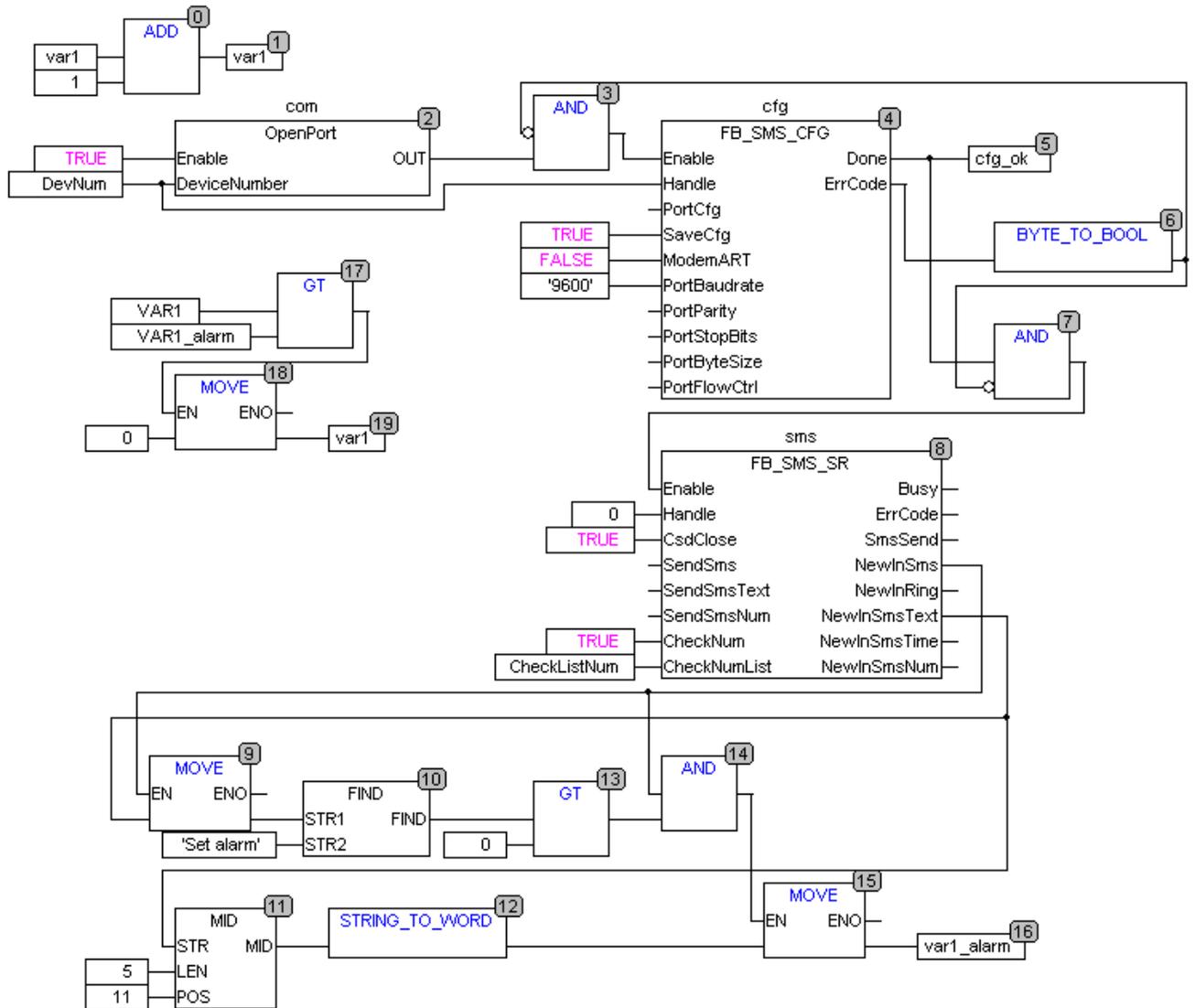


Рис.5. Алгоритм работы программы.

Список переменных:

PROGRAM PLC\_PRG

VAR

Com :OpenPort;

cfg :FB\_SMS\_CFG;

sms :FB\_SMS\_SR;

DevNum :WORD;

cfg\_ok: BOOL;

var1 :WORD;

var1\_Alarm :WORD :=5000;

CheckListNum: STRING :='79012345678;7987654321';

END\_VAR

Как и в предыдущих примерах необходимо добавить в конфигурацию ПЛК модуль Universal Network Module, произвести его настройку. Также необходимо в программе открыть порт и произвести настройку модема. (Можно взять за основу пример №0) Зависимость поведения ФБ от значений на входах «**CheckNum**» и «**CheckNumList**» представлена в следующей таблице:

Таблице 1.

№	Вход « <b>CheckNum</b> »	Вход « <b>CheckNumList</b> »	Результат
1	Flase	любое	Принимает все входящие СМС
2	True	Пустая строка (‘’)	Не принимает входящие СМС
3	True	Номер или список номеров ('79012345678;7987654321')	Принимает СМС только с указанных номеров

Блоки 9-16 нужны для расшифровки текста сообщения: блоки 9,10,13 выделяют командное слово, блоки 11,12 выделяют значение переменной из текста.

Как только сообщение будет принято на выход «**NewInRing**» будет подан единичный импульс (одновременно с этим будут обновлены значения на выходах «**NewInSmsText**», «**NewInSmsTime**» и «**NewInSmsNum**»), что совместно с блоками 14,15 позволяет осуществить однократную запись нового значения в переменную «*var1\_Alarm*». Это может быть полезно, если значение этой переменной задается несколькими способами, например, еще и с панели оператора.

Выход «**NewInSmsNum**» позволяет организовать разделение абонентов по уровням доступа.

### Пример №3\_1. Отправка и получение СМС сообщений. Задание уставки при помощи СМС.

В данном примере будет показано, как одновременно отправлять и получать СМС сообщения.

Программа:

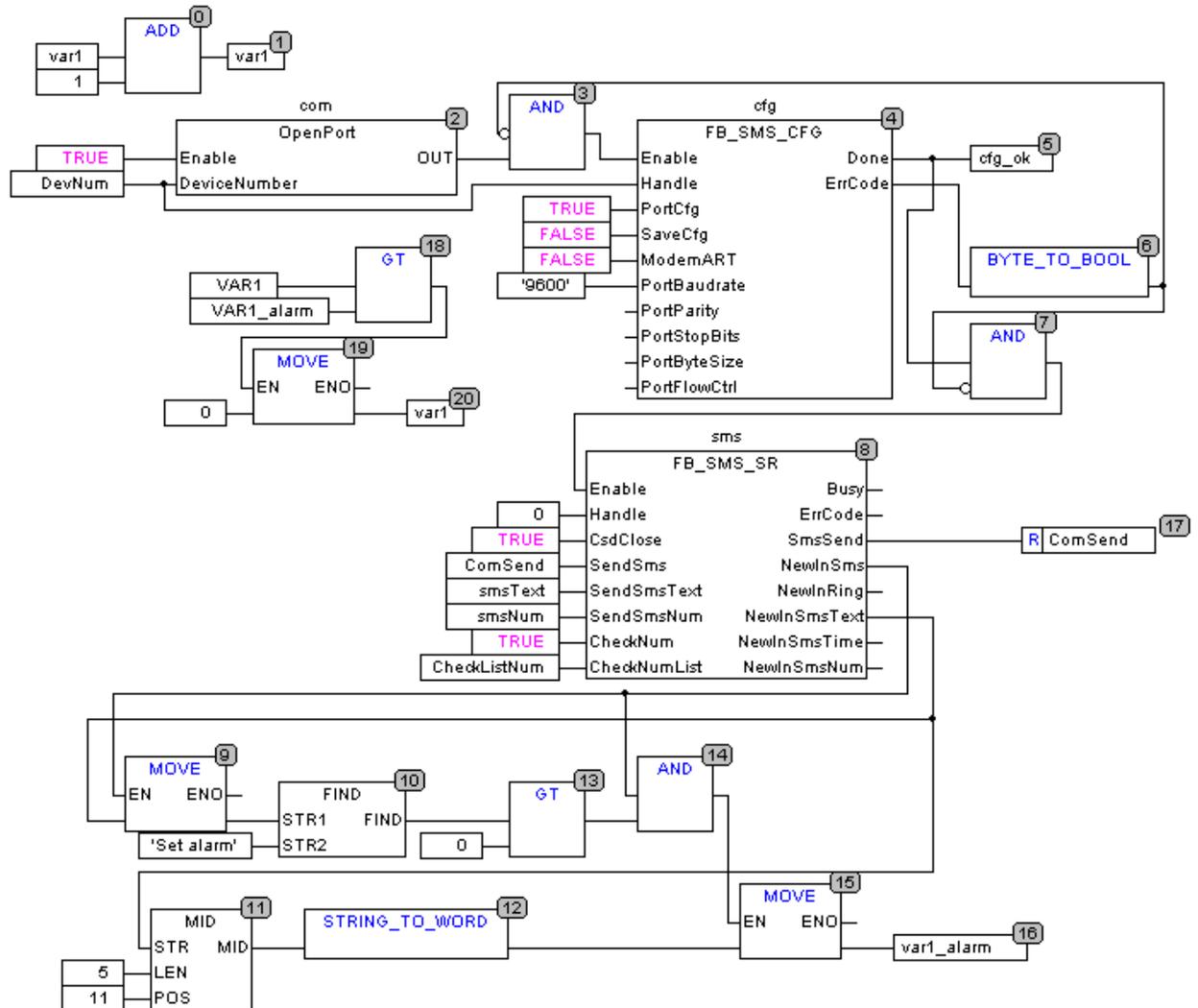


Рис.6. Алгоритм работы программы.

Список переменных:

PROGRAM PLC\_PRG

VAR

```

Com          :OpenPort;
cfg          :FB_SMS_CFG;
sms          :FB_SMS_SR;
DevNum       :WORD;
cfg_ok       :BOOL;

var1         :WORD;
var1_Alarm   :WORD      :=5000;
CheckListNum: STRING:= '79012345678;79876543210';
ComSend      :BOOL;
smsText      :STRING    := 'Set alarm 8000';
smsNum       :STRING    := '79012345678';

```

END\_VAR

Как и в предыдущих примерах необходимо добавить в конфигурацию ПЛК модуль Universal Network Module, произвести его настройку. Также необходимо в программе открыть порт и произвести настройку модема. (Можно взять за основу пример №0) Основные моменты по отправке и получению и отправке были рассмотрены в примерах №1\_1 и №2.

Необходимо понимать, что модем одновременно может выполнять только одно действие: или читать входящие сообщения или отправлять, при этом на выполнение любого действия необходимо время. Поэтому работа с модемом осуществляется в асинхронном режиме. В среднем необходимо не более 20 сек. на отправку одного СМС сообщения. Как следствие необходимо использовать флаг, который бы показывал готовность блока к отправке нового сообщения. В данном примере в качестве такого флага можно использовать переменную «*ComSend*», которая так же является командой на отправку, Новое сообщение можно отправлять, только когда она равна *False*, если же она равна *True*, то при попытке отправить еще одно сообщение оно будет потеряно.

### Пример №3\_2. Отправка и получение СМС сообщений. Запрос баланса.

В данном примере будет показано, как осуществлять запрос баланса.

Программа:

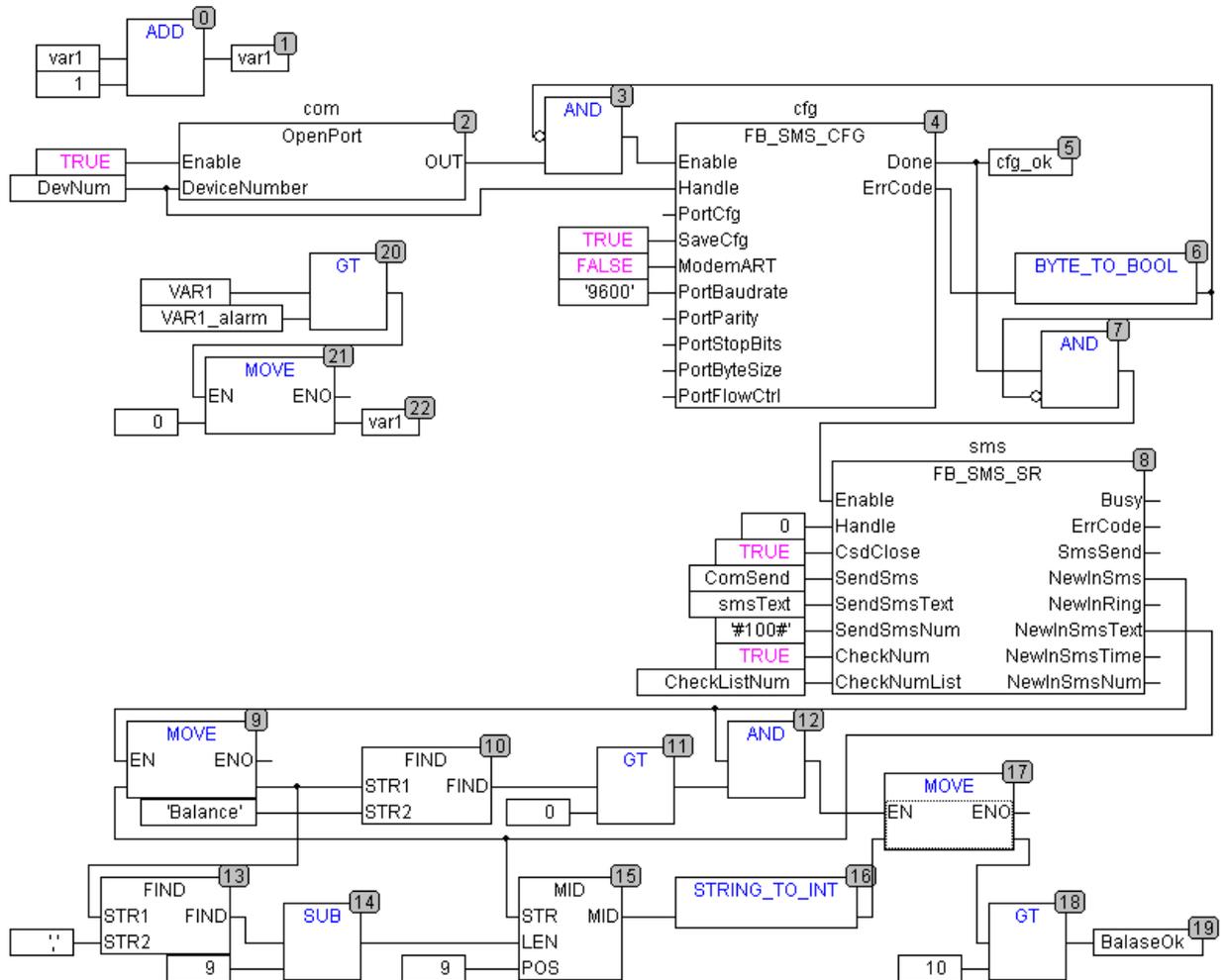


Рис.7. Алгоритм работы программы.

Список переменных:

PROGRAM PLC\_PRG

VAR

Com	:OpenPort;
cfg	:FB_SMS_CFG;
sms	:FB_SMS_SR;
DevNum	:WORD;
cfg_ok	:BOOL;
var1	:WORD;
var1_Alarm	:WORD :=5000;
CheckListNum	:STRING :=";
ComSend	:BOOL;
smsText	:STRING :=";
BalaseOk	:BOOL;

END\_VAR

Данный пример отличается от предыдущего №3\_1 только настраиваемыми параметрами. В качестве номера используется короткий номер запроса баланса, у каждого оператора он свой, здесь используется номер МТС. У некоторых операторов несколько номеров: для запроса на русском языке и запроса на латинице – рекомендуется использовать номер запроса на латинице. Так как запрос баланса по своей сути является USSD запросом, то внутри блока имеет несколько другую обработку, поэтому в данном случае блок игнорирует значения на входах «**CheckNum**» и «**CheckNumList**» (считает что «**CheckNum**» имеет значение *False*). Поле текста отправляемого сообщения нужно оставлять пустым.

#### Пример №4. Режим передачи данных и работа с СМС сообщениями. Удаленный опрос (ПЛК – ModBus master) через канал CSD и отправка СМС сообщений.

В данном примере будет показано, как опрашивать удаленный объект при помощи канала CSD и одновременно при необходимости отправлять СМС сообщения.

Программа:

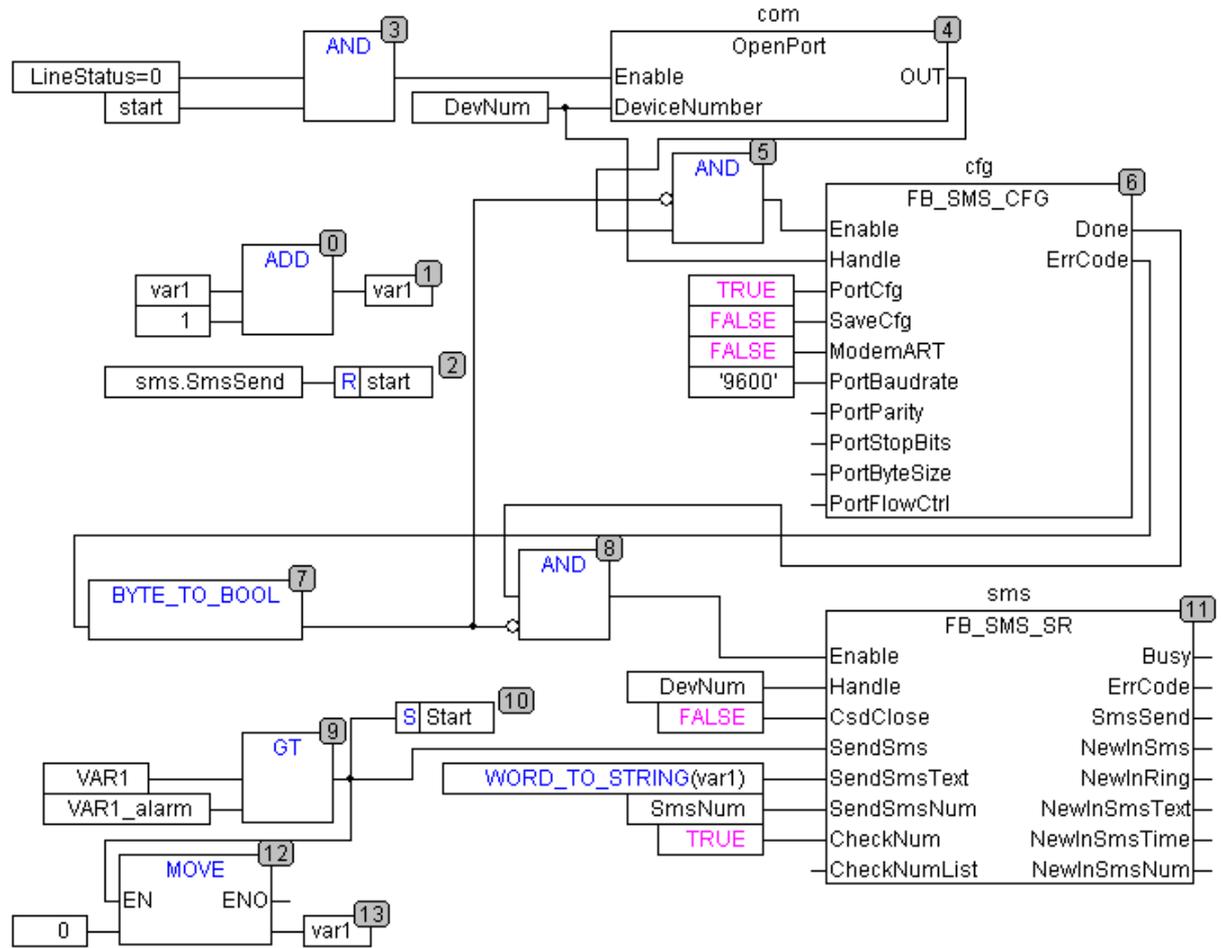


Рис.8. Алгоритм работы программы.

Список переменных:

```
PROGRAM PLC_PRG
```

```
VAR
```

```

Com      :OpenPort;
cfg      :FB_SMS_CFG;
sms      :FB_SMS_SR;

DevNum   :WORD;

Start    : BOOL      :=FALSE;
cfg_ok   :BOOL;

var1     AT%QW12.1.1.0  :WORD;
var1_Alarm AT%IW12.1.0.0 :WORD      :=8000;

LineStatus AT%QB12.0.0   : BYTE;
SmsNum     :STRING      := '79012345678';

```

```
END_VAR
```

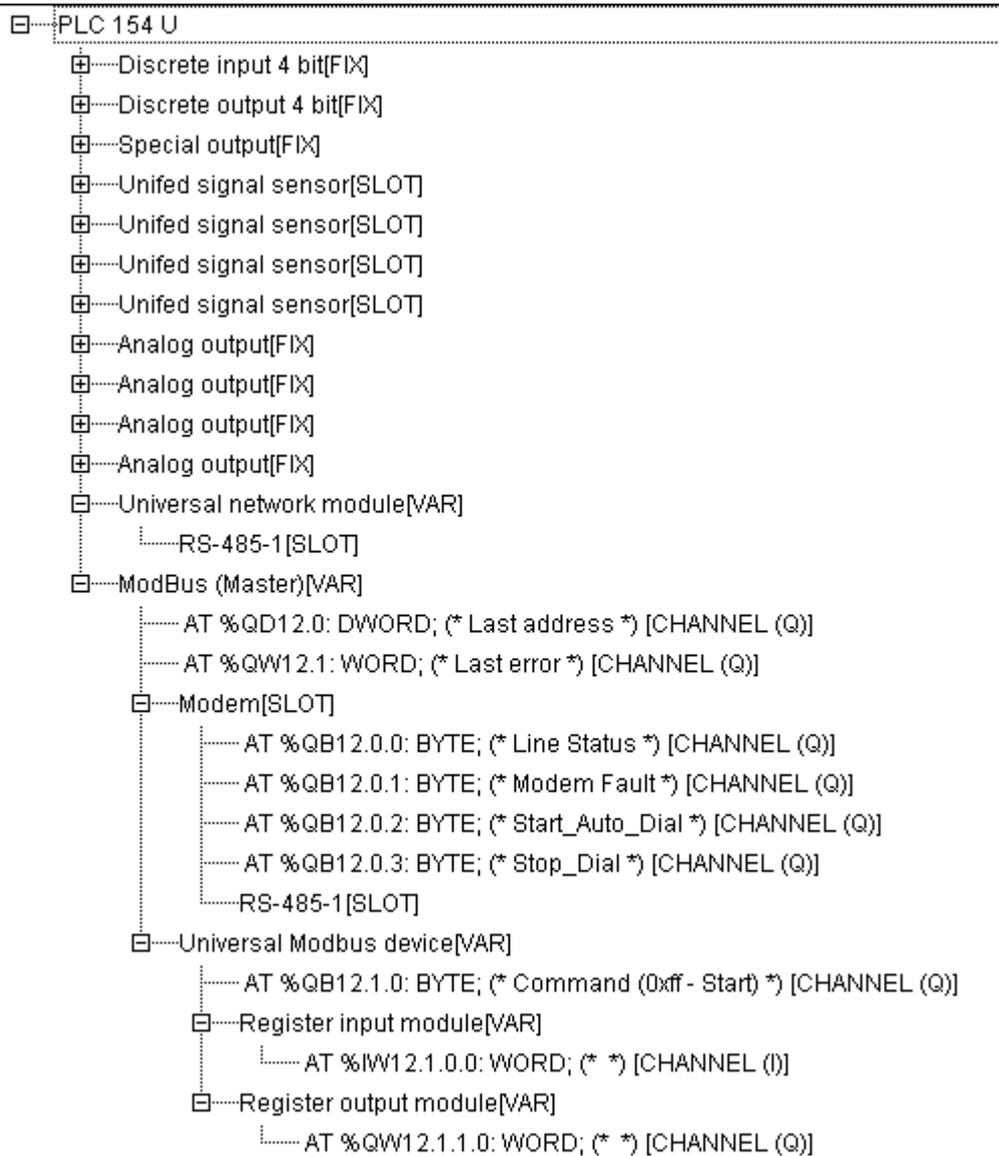


Рис.9. Конфигурация ПЛК

В данном примере ПЛК будет инициировать связь с удаленным объектом (ПК-оператора), куда будет передавать текущее значение «*var1*»- контролируемый параметр, и откуда будет считывать значение переменной «*var1\_Alarm*»- аварийный уровень. При этом если текущее значение переменной превысит аварийный уровень (далее авария) отправить СМС сообщение и обнулить ее.

Настройка связи с удаленным объектом настраивается в конфигурации ПЛК, как это показано на рис.9.

Алгоритм работы следующий: В фоновом режиме происходит периодический обмен между ПЛК и ПК. В случае аварии выставляется флаг о необходимости отправить СМС «*Start*», блоку «*FB\_SMS\_SR*» подается команда на отправку, номер абонента и текст сообщения. Далее если в текущий момент модем не занят (переменная «*LineStatus*» равна нулю), то захватываем порт, при необходимости производим настройку модема, после чего происходит непосредственно отправка сообщения. После отправки флаг снимается (при помощи выхода «*SmsSend*» блока «*FB\_SMS\_SR*»)и происходит освобождение порта.

Лист изменений в версиях документа

<b>Номер версии</b>	<b>Дата выпуска</b>	<b>Содержание изменений</b>
01	19.07.2012	Новый документ
02	09.08.2012	Изменение оформления, Изменение описания входов блоков, коррекция примеров.