



ПРОГРАММИРУЕМЫЕ ЛОГИЧЕСКИЕ КОНТРОЛЛЕРЫ

Библиотека UNM

Руководство по применению

Версия 01

Москва

2010

Содержание

Введение.....	3
Словарь условных сокращений.....	3
1. Установка дополнительных библиотек для проекта	4
2. Состав библиотеки UNM	7
Преобразование принятого байта (BYTETOSTRING).....	7
Получение данных из интерфейса (GETBYTE).....	8
Захват интерфейса (LOCKDEVICE)	9
Расчет контрольной суммы по алгоритму Dallas Semiconductors (POLINOM_SUMM).....	10
Преобразование принятого байта (RBDATA_TO_STRING).....	11
Освобождение захваченного интерфейса (RELEASEDEVICE)	12
Передача в интерфейс данных (SETBYTE)	13
Расчет суммы по модулю 256 (SUMMFORMOD256)	14
Инициализация модуля параметрами из COM_INI (INITCOM).....	15
Лист изменений в версиях документа	17

Введение

Компания ОВЕН предоставляет пользователю библиотеки дополнительных программных компонентов, облегчающие составление проекта работы программируемого логического контроллера (ПЛК) для решения наиболее распространенных практических задач. Эти библиотеки предназначены для работы на контроллерах ОВЕН ПЛК.

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

Библиотека UNM (файл UNM.lib) – обеспечивает прием/передачу информации через встроенные порты контроллера (RS-232/RS-485/Ethernet) для расширения возможностей стандартных модулей, поставляемых в составе системы CoDeSys (работа одновременно с модулями протоколов Modbus, DCON и ОВЕН на одном физическом интерфейсе). Назначение всех программных компонентов библиотеки указано в таблице 1.

Таблица 1

Имя функции	Назначение и область применения
BYTETOSTRING	Преобразование принятого байта
GETBYTE	Получение данных из интерфейса
LOCKDEVICE	Захват интерфейса
POLINOM_SUMM	Расчет контрольной суммы по алгоритму Dallas Semiconductors
RBDATA_TO_STRING	Преобразование принятого байта
RELEASEDEVICE	Освобождение захваченного интерфейса
SETBYTE	Передача в интерфейс данных
SUMFORMOD256	Расчет суммы по модулю 256
INITCOM	Инициализация модуля параметрами из COM_INI

<Выделенной цветом функции в составе библиотек нет. Может её и описывать не нужно?>

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

Далее в тексте для компактного описания используются следующие сокращения:

- CoDeSys** – Controllers Development System, специализированная среда программирования логических контроллеров. Торговая марка компании 3S-Software.
- 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) только стандартный набор библиотек. Дополнительные библиотеки добавляются пользователем по мере необходимости, в папку к уже имеющимся библиотекам. Для подключения новых библиотек к проекту сначала соответствующие файлы переписываются пользователем в ту же папку, где находятся все остальные библиотеки.

Чтобы увидеть какие библиотеки уже были раньше подключены к проекту и установить дополнительные библиотеки, используется «Менеджер библиотек (Library Manager)» – его можно открыть из главного меню CoDeSys командами «Окно (Window) ► Менеджер библиотек (Library Manager)» (или на вкладке организатора объектов «Ресурсы (Resources)» открывается папка «Менеджер библиотек (Library Manager)», см. рисунок 1.1. На рисунке в средней верхней части окна отображается список установленных библиотек.

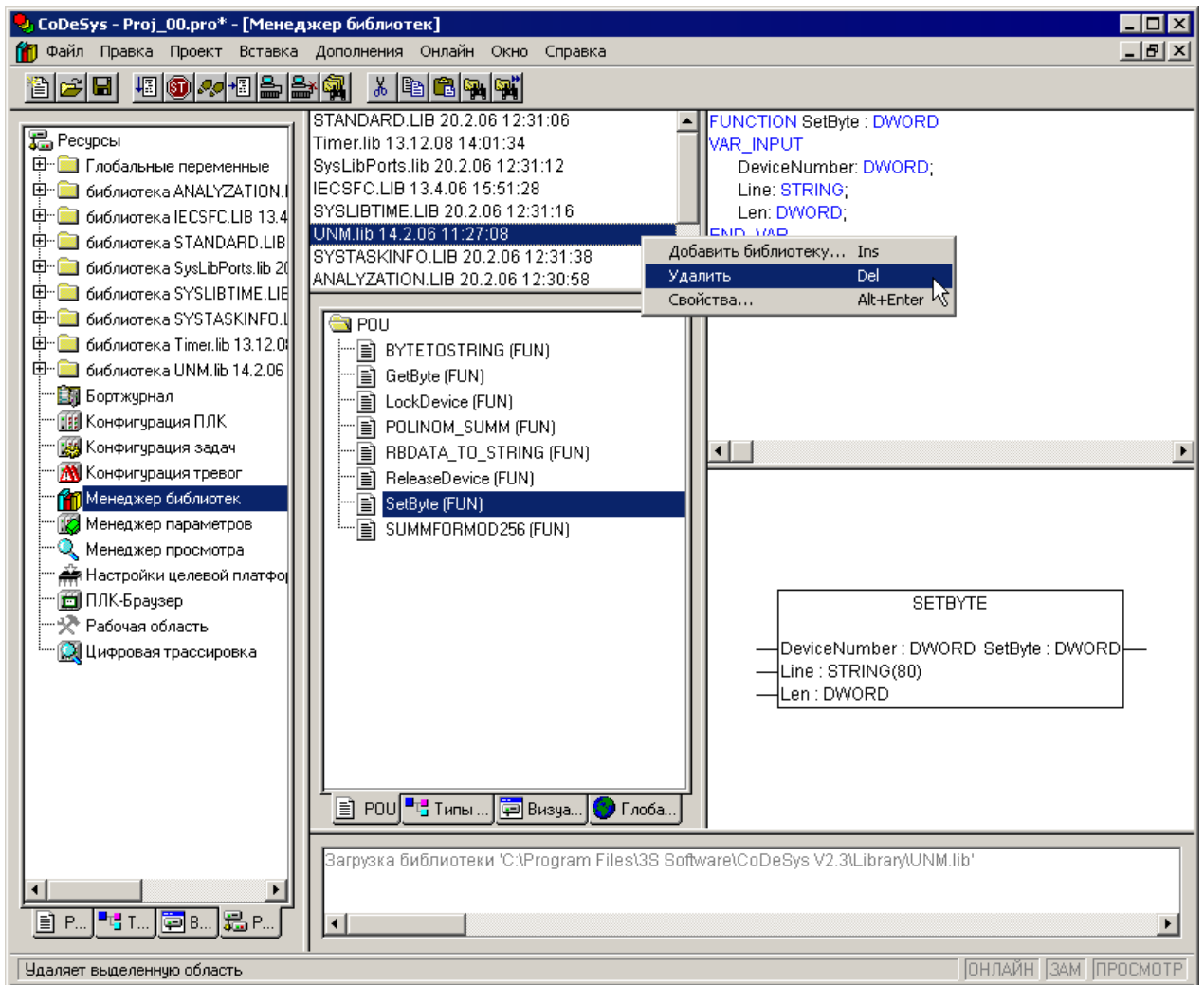


Рисунок 1.1 – Окно вкладки организатора объектов «Ресурсы (Resources)» с режимом работы «Менеджер библиотек (Library Manager)»

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**.

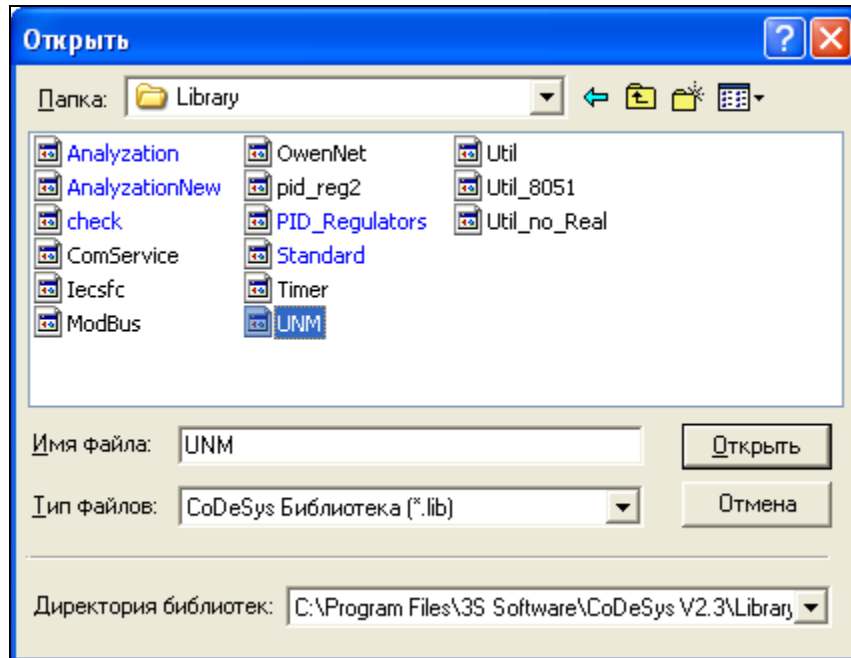


Рисунок 1.2 – Окно выбора подключаемой к проекту дополнительной библиотеки

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

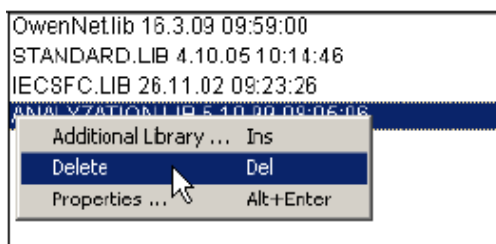
Для просмотра состава и свойств программных компонентов курсором выбирается нужная библиотека, – при этом появится папка с программными компонентами, в которой выделяется конкретный программный компонент (на рисунке 1.1 справа дана краткая справочная информация по его использованию).

Примечания.

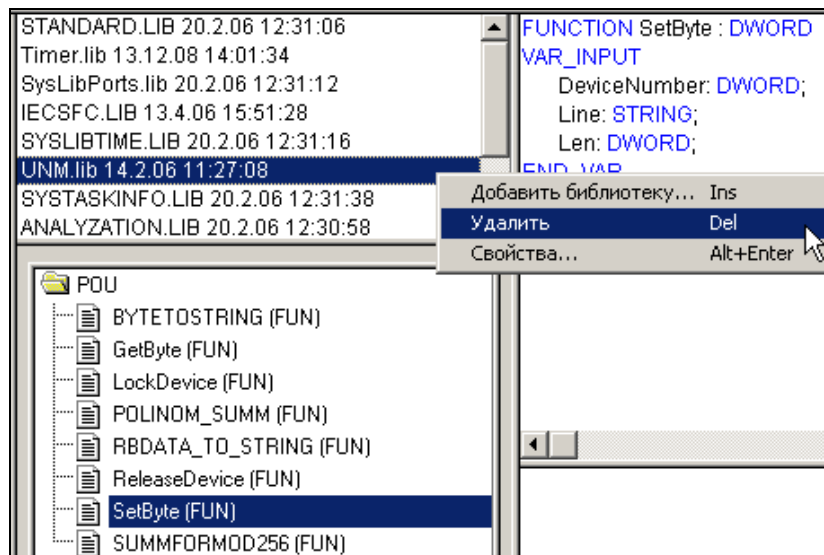
1. Рекомендуется размещать все библиотеки, которые планируется подключать, в папке для хранения библиотек, создаваемой CoDeSys автоматически.

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>).



а)



в)

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

2. Состав библиотеки UNM

Библиотека Universal Network Module (далее UNM) предназначена для приема/передачи последовательности байтов через встроенные порты контроллера (RS-232/RS-485/Ethernet).

Для использования библиотеки нужно создать в конфигурации контроллера один (или несколько) модулей Universal Network Module. Внутри каждого модуля нужно настроить параметры функционирования порта передачи данных. Для последовательных портов это скорость, четность, биты данных, биты стопа, время задержки. Для Ethernet это порт TCP/IP протокола.

Особенностью данной библиотеки является возможность работать одновременно с модулями протоколов Modbus, DCON и OВЕН на одном физическом интерфейсе. Это позволяет создать модуль опроса устройства стандартными командами или выдавать в интерфейс и получать из интерфейса в нужное время произвольную последовательность байтов. Например, в случае если на порт RS-232 установлен модем, до начала работы стандартного модуля опроса программа пользователя позволяет установить связь с удаленным устройством используя «АТ» последовательности. После чего начинает работать модуль опроса устройства через стандартный протокол.

Преобразование принятого байта (BYTETOSTRING)

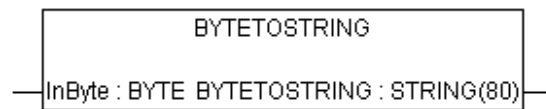


Рисунок 2.1 – Структурная схема

Таблица 2.1

Имя программного компонента	BYTETOSTRING		
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, ПЛК100, ПЛК110, ПЛК150, ПЛК154		
Входная переменная:	Тип данных	Пояснения	
InByte	BYTE	Значение для перекодирования	
Выходная переменная:	Тип данных	Пояснения	
BYTETOSTRING	STRING	Возвращаемое значение – строка с шестнадцатеричным кодом символа	

Особенности работы

Функция обеспечивает преобразование принятого байта в строку. Выходная строка содержит значение в шестнадцатеричном коде. Например, при InByte=16 – возврат «0F».

Получение данных из интерфейса (GETBYTE)

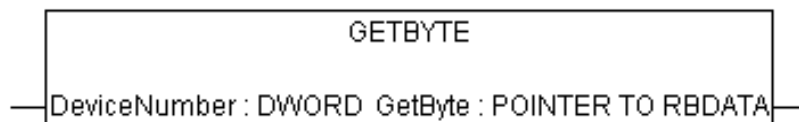


Рисунок 2.2 – Структурная схема

Таблица 2.2

Имя программного компонента	GETBYTE		
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, ПЛК100, ПЛК110, ПЛК150, ПЛК154		
Входная переменная:	Тип данных	Пояснения	
DeviceNumber	DWORD	Номер модуля UNM (задается пользователем)	
Выходная переменная:	Тип данных	Пояснения	
GetByte	POINTER TO RBDATA	Возвращаемое значение является указателем на структуру RBDATA	

Особенности работы

Функция обеспечивает последовательное получение данных из интерфейса (строки байта).

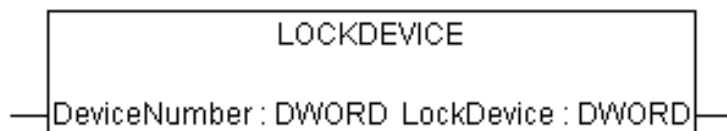
Описание структуры RBDATA:

STRUCT			
data: BYTE;			– принятый байт
flags: BYTE;			– флаги
Флаги			
unsigned char	reserved:	1;	– не используется
unsigned char	noktime:	1;	– время между байтами >1.5 символов
unsigned char	new_frame:	1;	– для внутреннего использования
unsigned char	end_frame:	1;	– время между байтами >3.5 символов
unsigned char	temp_el_full:	1;	– для внутреннего использования
unsigned char	overrun:	1;	– ошибка переполнения буфера
unsigned char	frame_error:	1;	– ошибка стоп бита
unsigned char	parity_error:	1;	– ошибка четности

Примечания.

1. Если неверно указан номер интерфейса, – функция на выходе принимает значение 0 (NULL).

2. Если нет принятых данных, – функция на выходе принимает значение 0 (NULL).

Захват интерфейса (LOCKDEVICE)**Рисунок 2.3 – Структурная схема****Таблица 2.3**

Имя программного компонента	LOCKDEVICE		
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, ПЛК100, ПЛК110, ПЛК150, ПЛК154		
Входная переменная:	Тип данных	Пояснения	
DeviceNumber	DWORD	Номер модуля UNM (задается пользователем). 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	
Выходная переменная:	Тип данных	Пояснения	
LockDevice	DWORD	Возвращаемое значение: 1 – интерфейс захвачен; (-1) – неверный номер интерфейса	

Особенности работы

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.

Расчет контрольной суммы по алгоритму Dallas Semiconductors (POLINOM_SUMM)

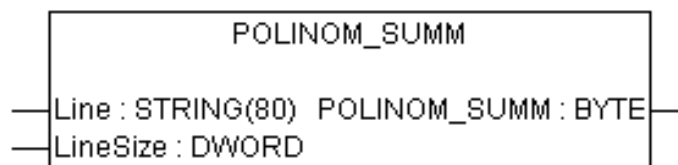


Рисунок 2.4 – Структурная схема

Таблица 2.4

Имя программного компонента	POLINOM_SUMM		
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, ПЛК100, ПЛК110, ПЛК150, ПЛК154		
Входные переменные:	Тип данных	Пояснения	
Line	STRING	Массив для расчета контрольной суммы	
LineSize	DWORD	Длина массива	
Выходная переменная:	Тип данных	Пояснения	
POLINOM_SUMM	BYTE	Рассчитанная контрольная сумма	

Особенности работы

Функция расчета восьми бит контрольной суммы по алгоритму Dallas Semiconductors (используется также в DCON).

Преобразование принятого байта (RBDATA_TO_STRING)

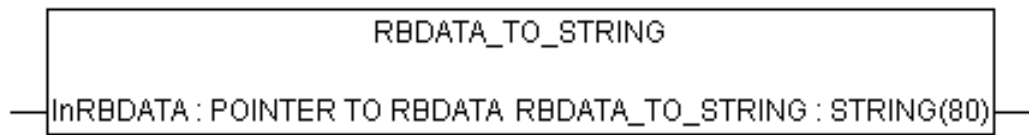


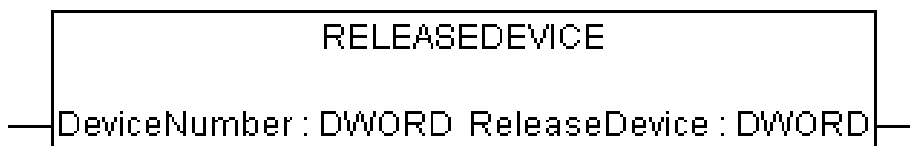
Рисунок 2.5 – Структурная схема

Таблица 2.5

Имя программного компонента	RBDATA_TO_STRING		
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, ПЛК100, ПЛК110, ПЛК150, ПЛК154		
Входная переменная:	Тип данных	Пояснения	
InRBDATA	POINTER TO RBDATA	Указатель структуры RBDATA (см. функцию GetByte)	
Выходная переменная:	Тип данных	Пояснения	
RBDATA_TO_STRING	STRING(80)	Возвращаемое значение будет строка, содержащая один символ с кодом ASCII, который соответствует значению поля data указателя на структуру RBDATA	

Особенности работы

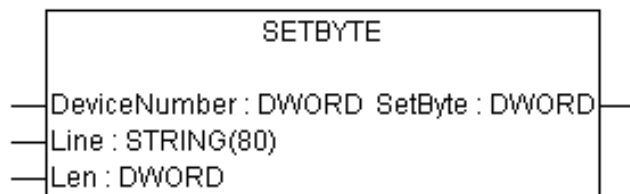
Функция обеспечивает преобразование принятого байта (из структуры RBDATA) в строку.

Освобождение захваченного интерфейса (RELEASEDEVICE)**Рисунок 2.6 – Структурная схема****Таблица 2.6**

Имя программного компонента	RELEASEDEVICE		
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, ПЛК100, ПЛК110, ПЛК150, ПЛК154		
Входная переменная:	Тип данных	Пояснения	
DeviceNumber	DWORD	Номер модуля UNM (задается пользователем)	
Выходная переменная:	Тип данных	Пояснения	
ReleaseDevice	DWORD	Возвращаемое значение: 1 – интерфейс освобожден; (-1) – неверный номер интерфейса	

Особенности работы

Функция обеспечивает освобождение захваченного ранее интерфейса. Все прочие модули PLC Configuration на данном физическом интерфейсе возобновляют свою работу.

Передача в интерфейс данных (SETBYTE)**Рисунок 2.7 – Структурная схема****Таблица 2.7**

Имя программного компонента	SETBYTE		
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, ПЛК100, ПЛК110, ПЛК150, ПЛК154		
Входные переменные:	Тип данных	Пояснения	
DeviceNumber	DWORD	Номер модуля UNM (задается пользователем)	
Line	STRING	Строка, содержащая массив байтов для последовательной передачи (до 256 байт)	
Len	DWORD	Длина массива данных	
Выходная переменная:	Тип данных	Пояснения	
SetByte	DWORD	Возвращаемое значение: (-1) – неверный номер интерфейса; N – количество переданных байт	

Особенности работы

Функция обеспечивает последовательную передачу в интерфейс строки байта данных.

Расчет суммы по модулю 256 (SUMMFORMOD256)

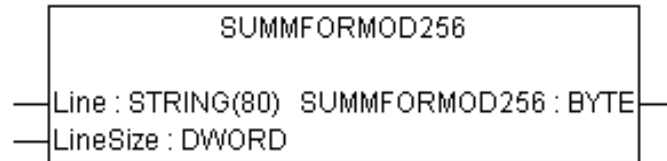


Рисунок 2.8 – Структурная схема

Таблица 2.8

Имя программного компонента	SUMMFORMOD256		
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, ПЛК100, ПЛК110, ПЛК150, ПЛК154		
Входные переменные:	Тип данных	Пояснения	
Line	STRING	Массив для расчета суммы по модулю 256	
LineSize	DWORD	Длина массива	
Выходная переменная:	Тип данных	Пояснения	
SUMMFORMOD256	BYTE	Возвращаемое значение рассчитанной суммы по модулю 256	

Особенности работы

Функция обеспечивает расчет суммы по модулю 256 (суммирование байтов без учета переполнения).

Инициализация модуля параметрами из COM_INI (INITCOM)

Таблица 2.9

Имя программного компонента	INITCOM		
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, ПЛК100, ПЛК110, ПЛК150, ПЛК154		
Входные переменные:	Тип данных	Пояснения	
DeviceNumber	DWORD	Номер модуля UNM (задается пользователем)	
INI	POINTER TO COM_INI	<p>Описание структуры COM_INI STRUCT</p> <p>unsigned char speed; //Скорость бит/с 115200=0, 57600=1, 38400=2, 28800=3, 19200=4, 14400=5, 9600=6, 4800=7, 2400=8, 1200=9.</p> <p>unsigned char parity; //Четность EVEN=0, ODD=1, SPACE=2, MARK=3, NO_PARITY=4, MULTY_DROP = 6.</p> <p>unsigned char bit_lenght; //Биты данных BITS_5=0, BITS_6=1, BITS_7=2, BITS_8=3.</p> <p>unsigned char stop_lenght; //Бит стоп STOP_1=0, STOP_1_5=1, STOP_2=2.</p> <p>Примечание. Остальные поля структуры не используются т. к. изменение параметров возможно только для последовательных портов (не Ethernet)</p>	
Выходная переменная:	Тип данных	Пояснения	
INITCOM	DWORD*	Возвращаемое значение: 1 – все нормально; (-1) – неверный номер интерфейса	

* Реализована, начиная с версии прошивки 1.31.0 (для ПЛК).

Особенности работы

Инициализации модуля параметрами, заданными в структуре COM_INI. Функция позволяет инициализировать последовательный порт новыми параметрами, например, изменить скорость передачи данных.

3. Пример использования

Пример на языке ST реализует работу с модемом: посылку команды инициализации, набор номера и установку связи. Для работы примера в PLC Configuration должен быть подключен модуль UNM и настроен интерфейс обмена с модемом.

```
VAR
    Error:BYTE;
    P:POINTER TO RBDATA;
    itsOK:STRING:= 'OK$N$R';
    itsCONNECT:= 'CONNECT$N$R';
    iter:INT;
END_IF

If(LockDevice(0)!=1) then
    Error:=1;
    Return;
End_if

If SetByte(0,'ATZ$N$R',5) !=1 then
    Error:=1;
    Return;
End_IF

(* Здесь задержка которая зависит от модема времени инициализации модема.
После этого мы должны получить 'OK' *)
For iter:=1 to len(itsOK) do
    P:=GetByte(0);
    If p=0 then
        Error:=1;
        return
    else
        if RBDATA_TO_STRING(p)<> MID (itsOK,1,iter); then
            error:=1;
            return;
        end_if
    End_if
END_FOR
(*Набор заданного номера*)
If SetByte(0,'ATDT89161234567$N$R',5) !=1 then
    Error:=1;
    Return;
End_IF

For iter:=1 to len(itsConnect) do
    P:=GetByte(0);
    If p=0 then
        Error:=1;
        return
    else
        if RBDATA_TO_STRING(p)<> MID (itsCONNECT,1,iter); then
            error:=1;
            return;
        end_if
    End_if
END_FOR
(*набор заданного номера – закончен – соединение прошло – остановим модуль UNM *)
ReleaseDevice(0);
```