

**Библиотека Codesys для реализации обмена  
данными по протоколу «Тензо-М»**

Данная библиотека CodeSys предназначена для работы с устройствами по протоколу Тензо-М, подробную информацию по протоколу можно найти по ссылке:

[Документация по протоколу](#)

Библиотека разработана в среде CodeSys v3.5 SP17 с использованием стандартных библиотек, поставляемых в комплекте с пакетом среды разработки. Так же требуемые для работы библиотеки размещены в папке “SystemLibs” архива с библиотекой.

Библиотека позволяет осуществлять циклический опрос одного или нескольких устройств, а так же однократные запросы к устройствам.

Циклический опрос выполняется с некоторыми ограничениями: возможно исполнение только команд, для которых при отправке запроса не нужен блок <Data>, исключение сделано только для запроса комплексной посылки с некоторыми допущениями.

Далее будет приведено более подробное описание состава библиотеки.

Небольшой пример реализации обмена данными с прибором ТВ-011 приведён в проекте “ExampleTensoM”, который можно найти в архиве с библиотекой.

Для возможности реализации протокола обмена на PLC, использующих иные среды исполнения, в архиве с библиотекой приведены исходные коды функциональных блоков, осуществляющих расчёт CRC, подготовку данных для отправки и обработку полученных данных, после небольшой доработки приведённый код может быть использован в других средах разработки.

#### Структуры данных библиотеки:

- UDT\_CalibrationPar (Struct)
- UDT\_ComplexData (Struct)
- UDT\_MainPar (Struct)
- UDT\_SpecialPar (Struct)
- UDT\_Weight (Struct)

#### UDT\_CalibrationPar

Структура данных калибровочных параметров по протоколу Тензо-М.

Name	Type	Comment
Zero_ADCValue	UDINT	Величина кода АЦП “нуля” весовой системы
Delta_ADCValue	UDINT	Величина изменения кода АЦП при нагрузке весов образцовым весом
Sample_Weight	UDINT	Образцовый вес, кг

#### UDT\_ComplexData (STRUCT)

Некоторые данные не включены в структуру, при необходимости можно расширить структуру.

Name	Type	Comment
Status_Byte	BYTE	Байт состояния весоизмерительной системы (побитовая расшифровка в соответствии с протоколом)
DO_State	DINT	Состояние дискретных выходов
DI_State	DINT	Состояние дискретных входов
Gross_Weight	UDT_Weight	Вес БРУТТО, кг

#### UDT\_MainPar (STRUCT)

Структура данных основных параметров по протоколу Тензо-М.

Name	Type	Comment
WeightLimit	REAL	Наибольший предел взвешивания, кг
DecPoint	INT	Количество знаков после запятой
Mode	BOOL	Текущий режим (0 - БРУТТО, 1 - НЕТТО)
IndDis	UDINT	Дискретность индикации веса
ADC_Freq	BYTE	Номер частоты обновления данных АЦП
TensoInfo	STRING	Информация по питанию и фильтрации тензодатчиков

## UDT\_SpecialPar (STRUCT)

Структура данных специальных параметров по протоколу Тензо-М.

Name	Type	Comment
DoseWeight	REAL	Вес разового отвеса, кг
StableTime	REAL	Время стабилизации показаний веса, сек
MaxDoseTime	REAL	Максимально разрешённое время разгрузки весового бункера, сек
EmtpyTankWeight	REAL	Вес пустого бункера, кг
CapacityLimit	REAL	Ограничение по производительности весов, т/ч
LimitDose	REAL	Задание ограниченной дозы, т

## UDT\_Weight (STRUCT)

Структура данных веса по протоколу Тензо-М.

Name	Type	Comment
Weight	REAL	Вес, кг
DecPointPos	INT	Количество знаков после запятой
Stable	BOOL	Флаг “успокоение”
Overload	BOOL	Флаг “перегруз”

Перечисления, используемые в библиотеке:

- CMD\_LIST (Enum)
- Comm\_STS (Enum)
- ERROR\_LIST (Enum)

## CMD\_LIST (ENUM)

Конкретный список команд зависит от модели прибора и версии прошивки. Данный список приведён для различных модификаций прибора ТВ-011.

Name	Initial	Comment
Not_selected	16#0	Команда не выбрана
CV_Counter_Get	16#12	Получить от устройства счётчик (только для конвейерных весов)
CV_Capacity_Get	16#15	Получить от устройства производительность (только для конвейерных весов)
SerialNo_Get	16#A1	Получить от устройства серийный номер
WeightPoint_Get	16#B1	Получить от устройства настройки весовой точки
SpecialPar_Get	16#B3	Получить специальные параметры
SpecialPar_Set	16#B4	Установить специальные параметры
Status_Get	16#BF	Получить состояние весоизмерительной системы
Zero_Set	16#C0	Обнулить текущие показания веса
MainPar_Get	16#C1	Получить настройку основных параметров преобразователя
NetWeight_Get	16#C2	Получить вес НЕТТО
GrossWeight_Get	16#C3	Получить вес БРУТТО
DI_Sts_Get	16#C4	Получить состояние дискретных входов
DO_Sts_Get	16#C5	Получить состояние дискретных выходов
Indicator_Set	16#C6	Вывести сообщение на основной и дополнительный индикаторы
Counter_Get	16#C8	Получить данные счётчиков
LastButton_Get	16#C9	Получить код последней нажатой клавиши
ComplexData_Get	16#CA	Получить комплексные данные от устройства
CalibrationData_Get	16#CB	Получить калибровочные параметры
ADC_Get	16#CC	Получить данные АЦП
WeightIndication_Set	16#CD	Перевести преобразователь в режим индикации веса
DO_Set	16#D0	Установить состояние дискретных выходов

Name	Initial	Comment
WeightPoint_Set	16#D1	Установить настройки весовой точки
MainIndicatorMsg_Set	16#D2	Вывести сообщение на основной индикатор
AddIndicatorMsg_Set	16#D3	Вывести сообщение на дополнительный индикатор
Filter_Set	16#DA	Установить значение фильтра
ComponentDosing_Parameter_Get	16#DC	Получить от устройства параметры дозирования компонента
ComponentDosing_Parameter_Set	16#DD	Установить параметры дозирования компонента
Recipe_Set	16#DE	Выбрать рецепт дозирования
WeightControl_Set	16#DF	Управление процессом перевешивания
MenuPar_Get	16#E0	Получить параметры меню SEL_3, SEL_5, SEL_6, SEL_8, SEL_9
MenuPar_Set	16#E1	Установить параметры меню SEL_3, SEL_5, SEL_6, SEL_8, SEL_9
RecipeParameter_Get	16#E2	Получить от устройства параметры дозирования из рецепта
RecipeParameter_Set	16#E3	Установить параметры дозирования в рецепт
Capacity_BCD_Set	16#E4	Установить производительность (в BCD формате)
Capacity_BCD_Get	16#E5	Получить от устройства заданную производительность (в BCD формате)
Capacity_Float_Set	16#E6	Установить производительность (в формате Float)
Capacity_Float_Get	16#E7	Получить от устройства заданную производительность (в формате Float)
DeviceInfo_Get	16#F0	Получить информацию о приборе (команда, неподдерживаемая протоколом, в ответ прибор должен прислать наименование и версию прошивки)

## Comm\_STS (ENUM)

Статус обмена данными.

Name	Initial	Comment
PORT_CLOSED	0	Порт закрыт
IDLE	1	Ожидание
WRITE	2	Запись данных
READ	3	Чтение данных
ERROR	4	Ошибка

## ERROR\_LIST (ENUM)

Список ошибок на выходе менеджера обмена данными.

Name	Initial	Comment
NO_ERRORS	0	Ошибки отсутствуют
TIME_OUT	5001	Ошибка тайм-аута
ABORT	5002	Запрос отменён
HANDLE_INVALID	5003	Неверный идентификатор
ERROR_UNKNOWN	5004	Неизвестная ошибка
WRONG_PARAMETER	5005	Неверный параметр
WRITE_INCOMPLETE	5006	Запись не завершена
PORT_ALREADY_OPENED	5050	Порт уже открыт
PORT_ALREADY_CLOSED	5051	Порт уже закрыт
PORT_NOT_OPENED	5052	Порт ещё не открыт
NO_START_CHAR	5061	Нет разделителя в начале посылки
WRONG_RESPONSE_ADR	5062	Неверный адрес устройства в ответе
WRONG_RESPONSE_CMD	5063	Неверная команда в ответе
NO_END_CHAR	5064	Нет разделителей в конце посылки
WRONG_CRC	5065	Неверный CRC в ответе от устройства
NO_DATA	5071	Нет данных

Name	Initial	Comment
PARAMETER_ERROR	5072	Ошибка параметров
WRONG_ZERO_WEIGHT	5073	Ошибка диапазона обнуления веса
PARAMETER_LOCKED	5074	Изменение параметров заблокировано
INPUT_BUFFER_ERROR	5075	Ошибка превышения длины входного буфера
CRC_NOT_VALID	5076	Неверный CRC в запросе к устройству
PARAMETER_SAVE_ERROR	5077	Ошибка сохранения параметров
UNKNOWN_ERROR_FROM_DEVICE	5078	Неизвестная ошибка от устройства

#### Функции, реализованные в библиотеке:

- CRC\_Calc (Function)
- FROM\_BCD\_ARRAY (Function)
- RawData\_TO\_CalibrationPar (Function)
- RawData\_TO\_ComplexData (Function)
- RawData\_TO\_MainPar (Function)
- RawData\_TO\_SpecPar (Function)
- RawData\_TO\_Weight (Function)
- TO\_BCD\_ARRAY (Function)

### CRC\_Calc (FUN)

Функция для вычисления CRC по протоколу Тензо-М.

- При отправке данных: после блока данных <Data> в буфере должен быть нулевой байт, он используется функцией для вычисления CRC.
- При приёме данных: при проверке в функцию передаются блоки <Data> и <CRC>, в этом случае вычисленный функцией CRC должен быть равным 0.
- Служебные байты 16#FE не должны участвовать в расчёте CRC.

На вход функции необходимо передать указатель на стартовый адрес и размер данных, для которых необходимо вычислить CRC. На выход передаётся вычисленный CRC.

Scope	Name	Type	Comment
Return	CRC_Calc	BYTE	
Input	InData	POINTER TO BYTE	Указатель на буфер данных
	DataLen	INT	Размер данных для расчёта CRC

### FROM\_BCD\_ARRAY (FUN)

Функция для преобразования байтового массива в формате BCD к целому числу.

На вход функции необходимо передать указатель на стартовый адрес и длину данных, которые необходимо привести к целому числу. На выход передаётся преобразованное целое число в формате DINT.

Внутри функции осуществляется проверка на формат BCD. Если один из байтов не соответствует формату, то на выходе функции будет число 0.

Scope	Name	Type	Comment
Return	FROM_BCD_ARRAY	DINT	
Input	BCD_Array	POINTER TO BYTE	Указатель стартового байта
	Len	INT	Кол-во данных для расшифровки

### RawData\_TO\_CalibrationPar (FUN)

Функция преобразования сырых данных для калибровочных параметров.

На вход функции необходимо передать указатель на стартовый байт данных для преобразования к структуре данных UDT\_CalibrationPar. На выходе функция выдаёт структуру калибровочных параметров UDT\_CalibrationPar.

Scope	Name	Type
Return	RawData_TO_CalibrationPar	UDT_CalibrationPar
Input	InData	POINTER TO BYTE

### RawData\_TO\_ComplexData (FUN)

Функция преобразования сырых данных для комплексной посылки. Используются не все данные комплексной посылки. Запрашиваемые данные: состояние системы, DI, DO, вес брутто (данные в запросе должны быть равны 16#4D: включены биты 0, 2, 3 и 6).

На вход функции необходимо передать указатель на стартовый байт данных для преобразования к структуре данных UDT\_ComplexData. На выходе функция выдаёт структуру комплексной посылки UDT\_ComplexData.

Scope	Name	Type
Return	RawData_TO_ComplexData	UDT_ComplexData
Input	InData	POINTER TO BYTE

### RawData\_TO\_MainPar (FUN)

Функция преобразования сырых данных для основных параметров.

На вход функции необходимо передать указатель на стартовый байт данных для преобразования к структуре данных UDT\_MainPar. На выходе функция выдаёт структуру основных параметров UDT\_MainPar.

Scope	Name	Type
Return	RawData_TO_MainPar	UDT_MainPar
Input	InData	POINTER TO BYTE

### RawData\_TO\_SpecPar (FUN)

Функция преобразования сырых данных для специальных параметров.

На вход функции необходимо передать указатель на стартовый байт данных для преобразования к структуре данных UDT\_SpecialPar и кол-во знаков после запятой. На выходе функция выдаёт структуру специальных параметров UDT\_SpecialPar.

Scope	Name	Type	Comment
Return	RawData_TO_SpecPar	UDT_SpecialPar	
Input	InData	POINTER TO BYTE	Указатель на стартовый байт для преобразования
	DecPoint	INT	Кол-во знаков после запятой

### RawData\_TO\_Weight (FUN)

Функция преобразования сырых данных к весу.

На вход функции необходимо передать указатель на стартовый байт данных для преобразования к структуре данных UDT\_Weight. На выходе функция выдаёт структуру веса UDT\_Weight.

Scope	Name	Type
Return	RawData_TO_Weight	UDT_Weight
Input	InData	POINTER TO BYTE

### TO\_BCD\_ARRAY (FUN)

Функция для преобразования целого числа к байтовому массиву в формате BCD.

На вход функции необходимо передать целое число, которое необходимо преобразовать и указатель на стартовый байт массива, в который необходимо сложить данные. При преобразовании младшие байты идут первыми.

Scope	Name	Type	Comment
Return	TO_BCD_ARRAY	BOOL	
Input	InValue	UDINT	Входное значение
	BCD_Array	POINTER TO BYTE	Указатель стартового байта

#### Функциональные блоки библиотеки:

- ComPortManager (FunctionBlock)
- CommManager (FunctionBlock)
- CommManager\_OneCmd (FunctionBlock)
- RcvDataParser (FunctionBlock)
- SndDataWrapper (FunctionBlock)

### CommManager (FB)

FB управляет обменом данных. В данном FB реализовано два вида обмена данных:

1. циклический опрос,
2. однократный (ациклический) опрос.

Циклический опрос осуществляется последовательно для списка устройств и списка команд из входных массивов: для выбранного устройства перебираются все заданные команды, далее переходит к следующему устройству. Входные массивы команд и устройств обрабатываются до первого нулевого значения, поэтому адреса устройств и команд в массиве необходимо задавать последовательно. Однократный опрос приостанавливает исполнение циклического. После исполнения однократного опроса циклический опрос будет продолжен. В циклическом опросе должны задаваться только команды без данных, в которых блок <Data> в устройство не отправляется. Исключение составляет только команда 16#CA “Запрос комплексной посылки”, для неё блок <Data> формируется внутри FB, при этом запрашиваются только следующие данные:

- состояние системы,
- состояние DI,
- состояние DO,
- вес брутто.

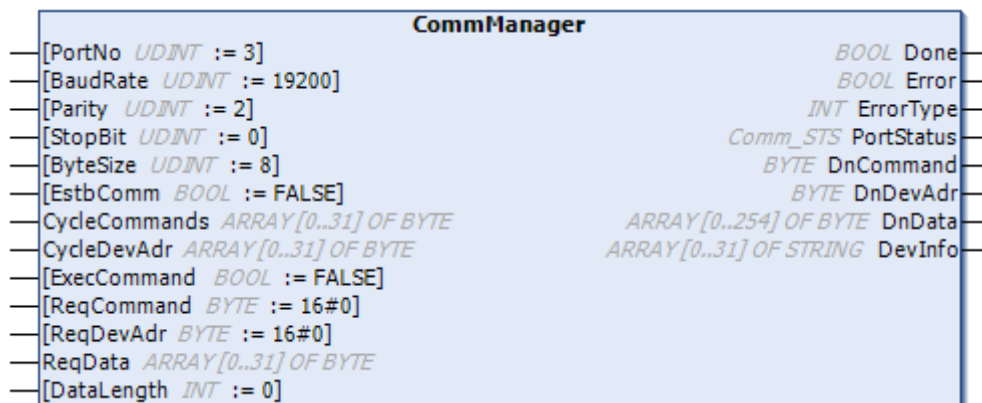
Однократный опрос выполняется в соответствии с входными данными FB, при этом ограничений по возможным командам нет.

Обмен данными осуществляется при наличии логической истины “TRUE” на входе EstbComm, при этом открывается COM порт. При снятии сигнала на входе EstbComm производится закрытие COM порта. При успешном выполнении опроса на выходе формируется бит Done, выдаётся команда, для которой выполнен опрос, адрес устройства и массив полученных данных (только блок <Data> из ответа устройства без служебных символов). Если в циклический опрос включена команда 16#F0 (либо другая команда, неподдерживаемая протоколом Тензо-М), то на выходе FB формируется массив DevInfo, содержащий описание устройств (индекс в массиве соответствует индексу устройства во входном массиве сетевых адресов): тип устройства и версия прошивки. В случае возникновения ошибки на выходе формируется бит Error и тип ошибки ErrorType.

InOut:

Scope	Name	Type	Initial	Comment
Input	PortNo	UDINT	3	Номер COM порта
	BaudRate	UDINT	19200	Скорость
	Parity	UDINT	2	Чётность: 0 - even, 1 - odd, 2 - none
	StopBit	UDINT	0	Стоповые биты: 0 - 1 stopbit, 1 - 1.5 stopbits, 2 - stopbits
	ByteSize	UDINT	8	Биты данных
	EstbComm	BOOL	FALSE	Запрос начала обмена данными
	CycleCommands	ARRAY [0..31] OF BYTE		Циклические команды (только команды без данных)
	CycleDevAdr	ARRAY [0..31] OF BYTE		Адреса устройств для циклического запроса данных
	ExecCommand	BOOL	FALSE	Выполнить ациклическую команду
	ReqCommand	BYTE	16#0	Ациклическая команда
	ReqDevAdr	BYTE	16#0	Сетевой адрес для ациклического запроса команды
	ReqData	ARRAY [0..127] OF BYTE		Массив данных для ациклического запроса
Output	DataLength	INT	0	Длина данных для отправки
	Done	BOOL	FALSE	Флаг получения новых данных (сбрасывается при начале запроса новых данных)
	Error	BOOL	FALSE	Флаг ошибки (сбрасывается при начале запроса новых данных)
	ErrorType	INT	0	Тип ошибки (сбрасывается при начале запроса новых данных)

Scope	Name	Type	Initial	Comment
				данных)
	PortStatus	Comm_STS	0	Статус работы COM порта
	DnCommand	BYTE	16#0	Команда, для которой получены данные
	DnDevAdr	BYTE	16#0	Адрес, для которого получены данные
	DnData	ARRAY [0..254] OF BYTE		Полученные данные
	DevInfo	ARRAY [0..31] OF STRING		Массив с информацией об устройствах



Графическое представление функционального блока CommManager

### CommManager\_OneCmd (FB)

FB управляет обменом данными. В данном FB реализована возможность однократного выполнения опроса устройства. Однократный опрос выполняется в соответствии с входными данными FB, при этом ограничений по возможным командам нет.

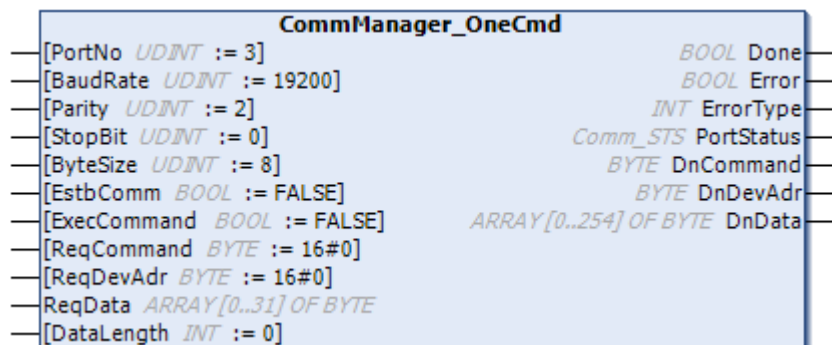
Обмен данными осуществляется при наличии логической истины "TRUE" на входе EstbComm, при этом открывается COM порт. При снятии сигнала на входе EstbComm производится закрытие COM порта. При успешном выполнении опроса на выходе формируется бит Done, выдаётся команда, для которой выполнен опрос, адрес устройства и массив полученных данных (только блок <Data> из ответа устройства без служебных символов). В случае возникновения ошибки на выходе формируется бит Error и тип ошибки ErrorType.

InOut:

Scope	Name	Type	Initial	Comment
Input	PortNo	UDINT	3	Номер COM порта
	BaudRate	UDINT	19200	Скорость
	Parity	UDINT	2	Чётность: 0 - even, 1 - odd, 2 - none
	StopBit	UDINT	0	Стоповые биты: 0 - 1 stopbit, 1 - 1.5 stopbits, 2 - stopbits
	ByteSize	UDINT	8	Биты данных
	EstbComm	BOOL	FALSE	Запрос начала обмена данными
	ExecCommand	BOOL	FALSE	Выполнить команду
	ReqCommand	BYTE	16#0	Команда
	ReqDevAdr	BYTE	16#0	Сетевой адрес для запроса команды
	ReqData	ARRAY [0..127] OF BYTE		Массив данных для запроса
	DataLength	INT	0	Длина данных для отправки
Output	Done	BOOL	FALSE	Флаг получения новых данных (сбрасывается при начале запроса новых данных)
	Error	BOOL	FALSE	Флаг ошибки (сбрасывается при начале запроса новых данных)
	ErrorType	INT	0	Тип ошибки (сбрасывается при начале запроса новых данных)
	PortStatus	Comm_STS	0	Статус работы COM порта



Scope	Name	Type	Initial	Comment
	DnCommand	BYTE	16#0	Команда, для которой получены данные
	DnDevAdr	BYTE	16#0	Адрес, для которого получены данные
	DnData	ARRAY [0..254] OF BYTE		Полученные данные



Графическое представление функционального блока CommManager\_OneCmd

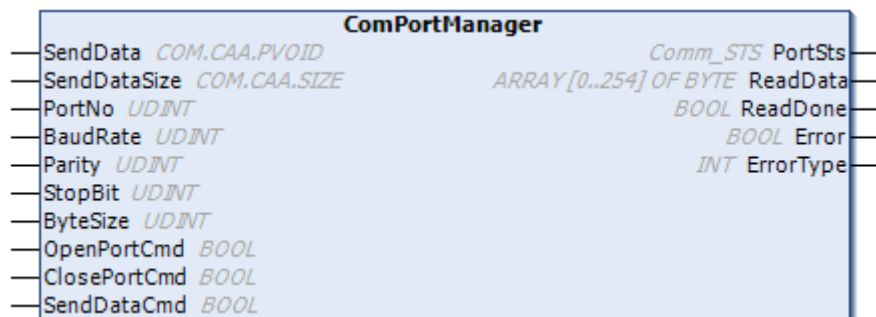
## ComPortManager (FB)

FB для работы с COM портом. Менеджер COM порта, осуществляет следующий функционал:

1. открытие порта по команде OpenPortCmd,
2. закрытие порта по команде ClosePortCmd,
3. отправку и чтение данных по команде SendDataCmd,
4. формирование ошибок, связанных с работой с COM портом.

InOut:

Scope	Name	Type	Initial	Comment
Input	SendData	COM.CAA.PVOID		Данные для отправки
	SendDataSize	COM.CAA.SIZE		Размер данных для отправки
	PortNo	UDINT		Номер COM порта
	BaudRate	UDINT		Скорость
	Parity	UDINT		Чётность: 0 - even, 1 - odd, 2 - none
	StopBit	UDINT		Стоповые биты: 0 - 1 stopbit, 1 - 1.5 stopbits, 2 - stopbits
Inout	ByteSize	UDINT		Биты данных
	OpenPortCmd	BOOL		Команда открытия COM порта
	ClosePortCmd	BOOL		Команда закрытия COM порта
Output	SendDataCmd	BOOL		Команда для отправки данных через COM порт, данные из COM порта читаются автоматически после отправки данных
	PortSts	Comm_STS		Статус работы
	ReadData	ARRAY [0..254] OF BYTE		Прочитанные данные
	ReadDone	BOOL	FALSE	Флаг завершения чтения данных
	Error	BOOL	FALSE	Флаг ошибки
	ErrorType	INT	0	Код ошибки



Графическое представление функционального блока ComPortManager

## RcvDataParser (FB)

FB для проверки полученных данных и удаления служебных битов. На выходе выдаёт чистый блок <Data> без служебных битов. В случае ошибки при проверке данных формирует флаг и тип ошибки.

Критерии проверки полученных данных:

1. Наличие сепараторов в начале и конце блока данных.
2. Проверка адреса устройства.
3. Совпадение команд в запросе и ответе.
4. Проверка CRC.

InOut:

Scope	Name	Type	Initial	Comment
Input	Enable	BOOL		Запуск на исполнение
	InData	ARRAY [0..254] OF BYTE		Массив данных для обработки
	Command	BYTE		Отправленная в устройство команда
	DeviceAdr	INT		Адрес устройства, от которого предполагалось получить данные
Output	Error	BOOL	FALSE	Флаг ошибки
	ErrorType	INT	0	тип ошибки
	OutData	ARRAY [0..254] OF BYTE		массив выходных данных
	Done	BOOL	FALSE	флаг завершения проверки без ошибок



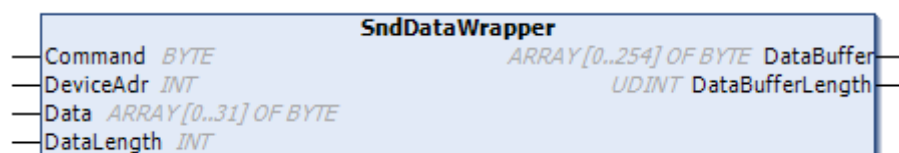
Графическое представление функционального блока RcvDataParser

## SndDataWrapper (FB)

FB подготавливает данные для отправки в прибор в зависимости от команды. Данные на входе: команда, данные команды, длина данных команды, адрес прибора. Данные на выходе: сформированный массив данных, длина буфера для отправки.

InOut:

Scope	Name	Type	Comment
Input	Command	BYTE	код команды
	DeviceAdr	INT	адрес устройства
	Data	ARRAY [0..127] OF BYTE	данные команды, если применимо
	DataLength	INT	длина данных команды
Output	DataBuffer	ARRAY [0..254] OF BYTE	выходной буфер данных для отправки
	DataBufferLength	UDINT	длина буфера данных для отправки



Графическое представление функционального блока SndDataWrapper