



Контроллер программируемый ЭЛСИ-ТМК. Поддержка протокола
передачи данных ГОСТ Р МЭК 61850

РУКОВОДСТВО ПО ПРИМЕНЕНИЮ

Часть 4

Име. № подл.	Подп. и дата	Взам. инв. №	Име. № дубл.	Подп. и дата

октябрь 2018

СОДЕРЖАНИЕ

СПИСОК ТЕРМИНОВ И СОКРАЩЕНИЙ.....	4
ИНФОРМАЦИЯ О ДОКУМЕНТЕ	5
1 ОПИСАНИЕ ПРОТОКОЛА IEC850	6
1.1 ОБЩИЙ ПРИНЦИП РАБОТЫ	6
2 КОНФИГУРИРОВАНИЕ IEC850	8
2.1 ДОБАВЛЕНИЕ УСТРОЙСТВ В ДЕРЕВО КОНФИГУРАЦИИ	8
2.2 ПАРАМЕТРЫ И СИГНАЛЫ УСТРОЙСТВ GOOSESUB И GOOSEPUB	10
3 ПРОГРАММИРОВАНИЕ IEC850.....	12
3.1 БИБЛИОТЕКА ПОДДЕРЖКИ ПРОТОКОЛА IEC850	12
3.2 СОЗДАНИЕ ПРОЕКТА ДЛЯ РАБОТЫ С IEC850	21
3.2.1 Создание проекта для GoosePub.....	21
3.2.1 Создание проекта для GooseSub.....	22
3.2.2 Пример программного кода для работы с IEC850.....	24
ПРИЛОЖЕНИЕ А (СПРАВОЧНОЕ) ПЕРЕЧЕНЬ МЕТОДОВ БИБЛИОТЕКИ ELSYIEC850LIBRARY .	27
ПРИЛОЖЕНИЕ Б (СПРАВОЧНОЕ) ПЕРЕЧЕНЬ ФУНКЦИЙ БИБЛИОТЕКИ ELSYIEC850LIBRARY .	29

СПИСОК ТЕРМИНОВ И СОКРАЩЕНИЙ

GOOSE	–	(англ. Generic Object Oriented Substation Event) Протокол передачи данных о событиях на подстанции (стандарт МЭК 61850-8-1);
ИЕС	–	International Electrotechnical Commission, См. также МЭК;
POU	–	Program Organization Unit. Компонент организации программ, программный компонент;
Кадр	–	Количество информации, состоящей из переменного числа байт передаваемой/получаемой протоколом за один раз;
Контроллер	–	Контроллер программируемый ЭЛСИ-ТМК;
КП	–	Контролируемый пункт;
МЭК	–	Международная электротехническая комиссия. См. также ИЕС;
ПЛК	–	Программируемый логический контроллер;
ПО	–	Программное обеспечение;
РЭ	–	Руководство по эксплуатации.

ИНФОРМАЦИЯ О ДОКУМЕНТЕ

Настоящее руководство по применению (далее - РП) содержит информацию по поддержке протокола передачи данных ГОСТ Р МЭК 61850-5-2011 (далее – МЭК850 (IEC850)). Данная информация необходима пользователю для правильной эксплуатации программируемого логического контроллера ЭЛСИ-ТМК (далее – контроллер).

РП является дополнением к основному документу «Контроллер программируемый ЭЛСИ-ТМК. Поддержка протокола передачи данных ГОСТ Р МЭК 61850 Руководство по применению».

Сведения, содержащиеся в документе, проверены на соответствие аппаратному и программному обеспечению на момент поставки контроллера. В связи с текущим совершенствованием продукции и документации, пользователю целесообразно следить за обновлениями через сайт производителя.

Авторские права на настоящий документ принадлежат компании АО «ЭлеСи». Копирование и распространение настоящего документа без письменного разрешения владельца авторских прав запрещено.

Контактная информация:

- почтовый адрес: **АО «ЭлеСи»**, 634021, г. Томск, ул. Алтайская, 161а;
- тел. (3822) 601-000, факс (3822) 601-001;
- официальный сайт компании: www.elesy.ru.

1 ОПИСАНИЕ ПРОТОКОЛА IEC850

1.1 ОБЩИЙ ПРИНЦИП РАБОТЫ

Программное обеспечение ЦП ПЛК ЭЛСИ-ТМК обеспечивает прием и передачу данных по протоколу GOOSE (IEC 61850 v1 и V2). Принимаемые и передаваемые данные доступны в задаче CoDeSys. Работа с программного обеспечения условно разделяется на три уровня (см. рисунок 1.1):

1. Уровень задачи пользователя;
2. Уровень конфигурирования;
3. Системный уровень.

На уровне задачи пользователя производится выделение прикладных данных получаемых GOOSE пакетов и формируются данные для передачи. Для упрощения обработки прикладных данных в состав TSP контроллера включена библиотека **ElsyIec850Library** в которой находятся необходимые функциональные блоки и функции.

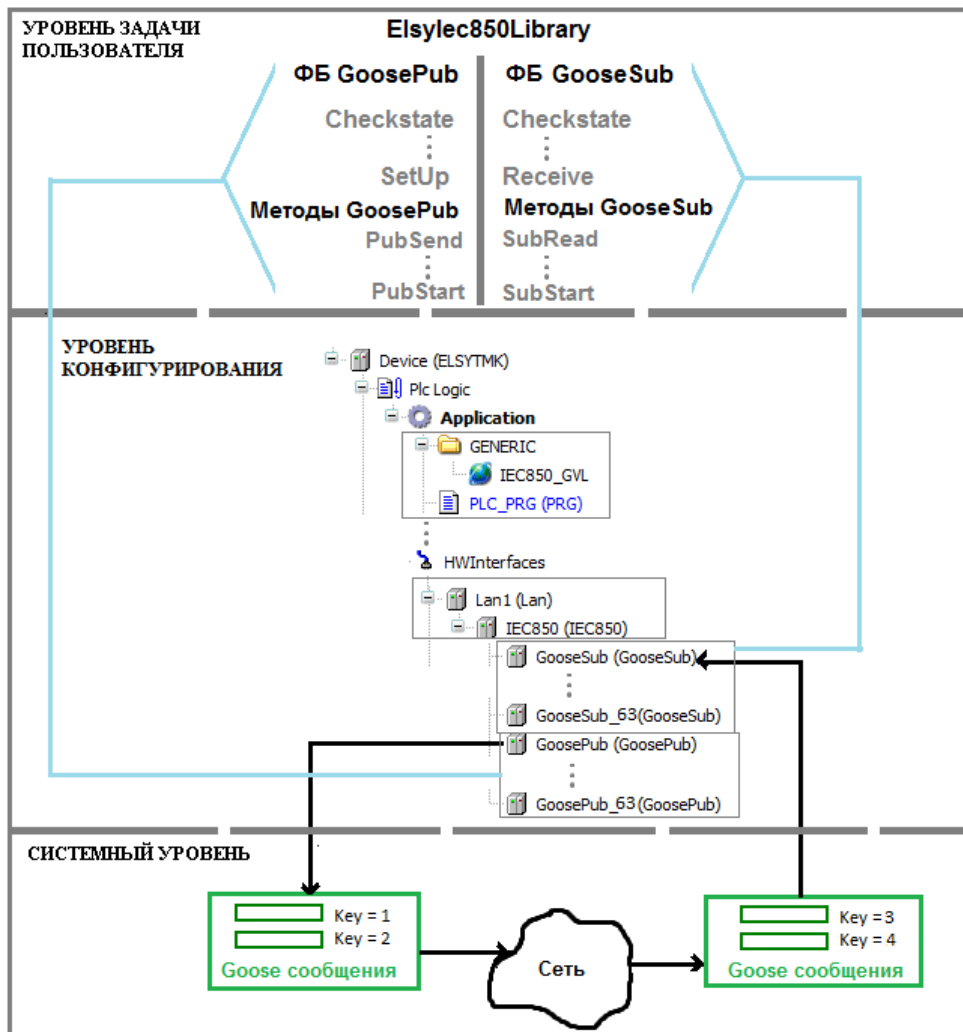


Рисунок 1.1 – Уровни обработки GOOSE сообщений

На уровне конфигурирования происходит выстраивание структуры системы.

Системный уровень обеспечивает поддержку приема и передачи MULTICAST сообщений в сети в соответствии с требованиями протокола ГОСТ Р МЭК 61850-5-2011 и

предварительную обработку принятых данных при необходимости. На данном уровне выполняется формирование необходимых для работы приемных буферов (устройство **GooseSub**, **GoosePub**) и производится фильтрация поступающих данных (параметры для фильтрации приведены в таблице 2.3), на следующий уровень передаются данные соответствующие заданным фильтрам. При необходимости пользователь может самостоятельно (в задаче пользователя) полностью обрабатывать принятые данные. Все параметры системного уровня задаются в устройствах **GooseSub**, **GoosePub** устройства IEC850 дерева устройств контроллера.

ВНИМАНИЕ! Поддержка протокола **Goose** в ПЛК ЭЛСИ-ТМК разрешается только при наличии лицензии на данную функцию.

Для получения данных протокола **GOOSE** в задаче пользователя необходимо выполнить следующие действия:

1. Убедиться и при необходимости добавить лицензию на поддержку протокола IEC850;
2. Создать конфигурацию контроллера, добавить в дерево устройство IEC850;
3. Добавить в устройство IEC850 необходимое количество устройств **GooseSub/GoosePub** и задать необходимые для фильтрации параметры;
4. Добавить в состав проекта библиотеку **ElsyIEC850Library** (данная библиотека добавляется автоматически);
5. Создать в декларативной части задачи пользователя (соответствующее заданным в дереве устройств **GooseSub/GoosePub**) необходимое количество экземпляров ФБ **ElsyIec850Library.GooseSub/ ElsyIec850Library.GoosePub**;
6. Создать в декларативной части задачи пользователя массивы для приема/передачи **GOOSE** сообщений;
7. Выполнить в задаче пользователя инициализацию экземпляра ФБ (передать указатели на массив для приема сообщения) и связывание его с необходимым устройством **GooseSub/GoosePub**. Связывание производится по ключу задаваемым параметром **Key**.
8. В процессе работы контролировать прием и передачу **GOOSE** сообщений и производить обработку данных.

2 КОНФИГУРИРОВАНИЕ IEC850

2.1 ДОБАВЛЕНИЕ УСТРОЙСТВ В ДЕРЕВО КОНФИГУРАЦИИ

Для работы с **IEC850** необходимо:

1. В дереве устройств найти узел **Lan** и нажать на него правой кнопкой мыши. В появившемся контекстном меню выбрать команду **Добавить устройство...**. В окне добавления устройств выбрать **Разное**→**IEC850** и нажать кнопку «Добавить устройство».
2. В дереве устройств найти **IEC850** и нажать правой кнопкой мыши. В появившемся контекстном меню выбрать команду **Добавить устройство...**. В окне добавления устройств выбрать **Оборудование компании ЭлеСи**→**ПЛК**→**IP**→**GooseSub** и нажать кнопку «Добавить устройство».
3. В дереве устройств найти **IEC850** и нажать правой кнопкой мыши. В появившемся контекстном меню выбрать команду **Добавить устройство...**. В окне добавления устройств выбрать **Оборудование компании ЭлеСи**→**ПЛК**→**IP**→**GoosePub** и нажать кнопку «Добавить устройство».

Итоговая конфигурация представлена на рисунке 2.1.

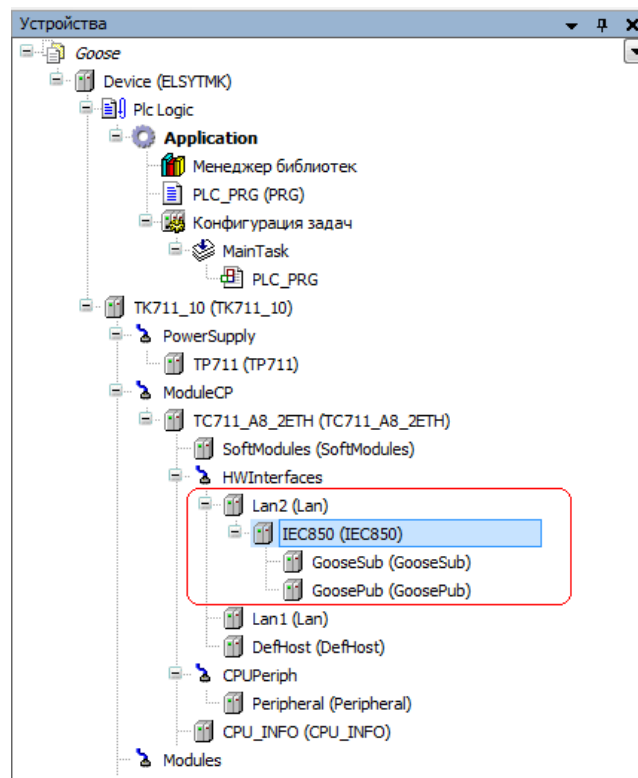


Рисунок 2.1 – Дерево устройств. Конфигурация **IEC850**

Примечания:

1. В дерево устройств можно добавить только одно устройство **IEC850**.
2. В дерево устройств можно добавить до 64 логических устройств **GooseSub**.
3. В дерево устройств можно добавить до 64 логических устройств **GoosePub**.
4. Работа с **IEC850** не предусмотрена для ЦП **TC712**.
5. При работе с **IEC850** рекомендуется использовать только **Lan1**.

Во вкладке **Редактор параметров** устройства **IEC850** представлены информационные параметры (таблица 2.1). Информационные параметры не доступны для редактирования пользователем.

Таблица 2.1 – Информационные параметры модуля «IEC850»

Имя параметра	Значение	Описание параметра
Информация модуля		
<i>License</i>	<i>no data</i>	Наличие лицензии (0 - отсутствует, 1 - присутствует)

Во вкладке **Редактор параметров** под информационными параметрами модуля расположена кнопка **«Generate»**, предназначенная для автоматической генерации глобальных переменных для **GooseSub** и **GoosePub** в соответствии с установленными конфигурационными параметрами. Для генерации массива, необходимо поставить галочку в графе **«Create array»** и нажать на кнопку **«Generate»** (рисунок 2.2). После нажатия кнопки **«Generate»** в дереве устройств появляется папка **GENERIC**, в которой содержатся глобальные переменные. При повторном нажатии на кнопку **«Generate»** папка **GENERIC** удаляется.

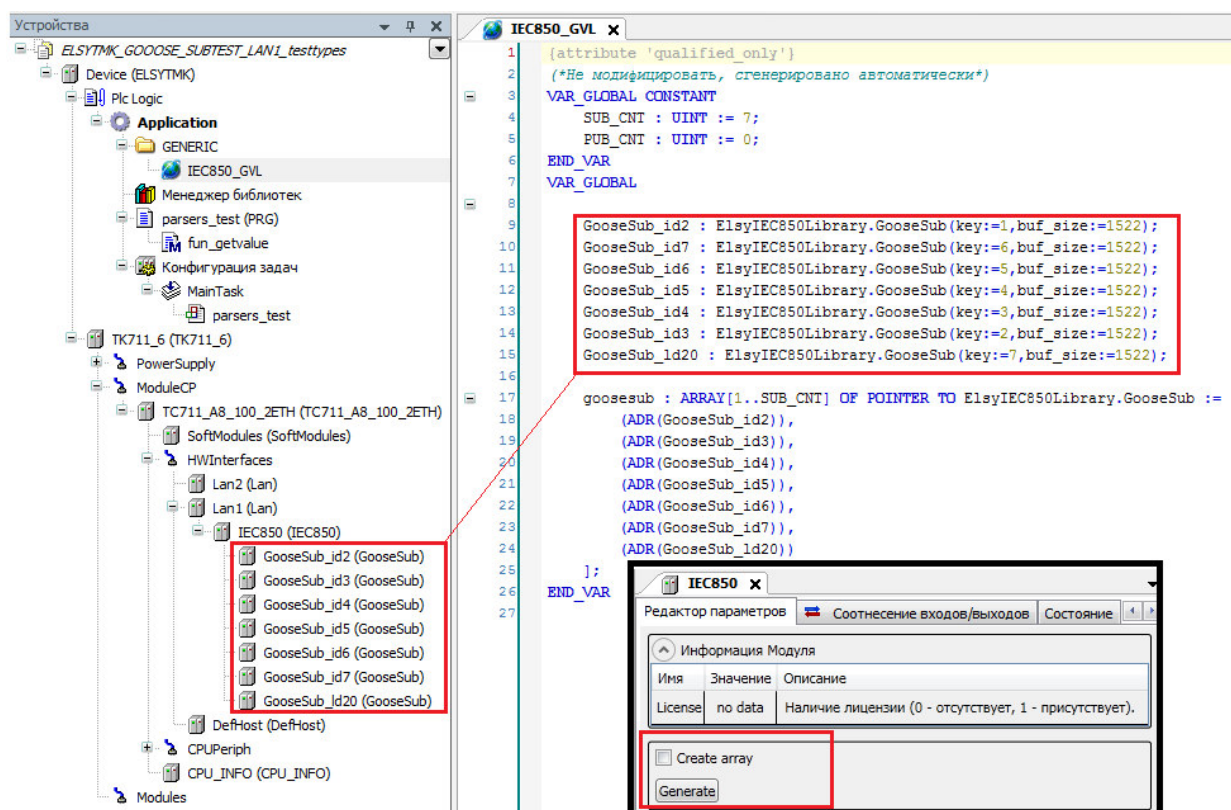


Рисунок 2.2 – Устройство **IEC850**. Область генерации глобальных переменных

Во вкладке **Соотнесение входов/выходов** для **IEC850** представлен перечень сигналов модуля (таблица 2.2).

Таблица 2.2 – Сигналы устройства «IEC850»

Имя канала	Тип	Адрес	Начальное значение	Описание параметра
<i>SigLicense</i>	BYTE	%IB209	0	Наличие лицензии (0 - отсутствует, 1 - присутствует)

Примечание: в таблицах 2.1 и 2.2 указаны переменные, информирующие пользователя о наличии/отсутствии лицензии на возможность применения в контроллере ЭЛСИ-ТМК **IEC850**. Признаком отсутствия лицензии является нулевое значение

параметров в указанных таблицах. Получение лицензии производится в соответствии с документом «Программа установки лицензий “setlic”. Инструкция по применению».

2.2 ПАРАМЕТРЫ И СИГНАЛЫ УСТРОЙСТВ GOOSESUB И GOOSEPUB

Слот **GooseSub** предназначен для приема Goose сообщений. Во вкладке **Редактор параметров** устройства **GooseSub** представлены информационные и конфигурационные параметры (таблица 2.3). Информационные параметры не доступны для редактирования пользователем.

Таблица 2.3 – Конфигурационные параметры устройства «GooseSub»

Имя параметра	Тип	Значение по умолчанию	Описание параметра
Информация модуля			
Key	UDINT	1	Ключ соединения
SlotReg	INT	no data	Результат регистрации коммуникационного слота в системе: 1 – устройство не обработано драйвером.
Конфигурационные параметры			
Dest_MAC	STRING	01:0C:CD:01:0:0	MAC адреса мультикаст группы (6 байт).
Src_MAC	STRING	00:00:00:00:00:0 0	MAC адрес источника (если поле заданно 00.00.00.00.00.00, то оно не участвует в формировании фильтра сообщение и приём ведётся с любым значением этого поля) (6 байт)
VLAN_ID	UINT	0	Номер VLAN (0 – работа без VLAN) (2 байт) в случае значение 0, кадр меньше на 2 байта
EtherType	UINT	16x88b8	Тип Ethernet кадра
APP_ID	UINT	16x0002	Идентификатор приложения (значение в диапазоне 0..0xFFFF)
Buf_size	UINT	1522	Максимальный размер буфера на прием.
GoCRef	STRING		Идентификатор блока управления (ASCII строка длиной от 0 до 128 символов). Если длина строки 0, то этот параметр не участвует в формировании фильтра сообщения.
datSet	STRING		Идентификатор набора данных (ASCII строка длиной от 0 до 128 символов). Если длина строки 0, то этот параметр не участвует в формировании фильтра сообщения.
goID	STRING		Идентификатор сообщения (строка от 0 до 128 символов). Если длина строки 0, то этот параметр не участвует в формировании фильтра сообщения.
confRev	UDINT	1	Версия GOOSE сообщения, если содержимое пакета меняется, то по этому полю осуществляется контроль.

ВНИМАНИЕ! Существует возможность не задавать необходимые параметры фильтрации на системном уровне, а выполнять обработку полученного GOOSE сообщения полностью в задаче пользователя. В этом случае все данные будут получены в один буфер в конкурирующем доступе, обработка будет производиться со скоростью работы задачи пользователя. В данном случае возможна потеря данных.

Примечания:

1. При задании фильтра типа STRING (**GoCRef**, **datSet**, **goID**) необходимо указывать "\$\$", который соответствует ASCII символу \$. Это связано с особенностью текстовых строк стандарта IEC 61131-3. В текстовых строках CoDeSys символ "\$" является "специальным" и знак после него рассматривается как код, для ввода непечатаемых символов. Эта информация есть в описании CoDeSys и является общим правилом.
2. Существует возможность частичного задания фильтра (учитываются только заданные начальные символы). Например, для в параметре **GoCRef** вместо "IED1LD7/LLN0\$GO\$gcb07" можно задать "IED1LD7" и получать все данные

IED1 от логического устройства LD7. Пользоваться с осторожностью! При задании параметра *GoCBRef* "IED1" будут получены данные всех логических устройств в один буфер в конкурирующем доступе.

Слот *GoosePub* предназначен для передачи Goose сообщений. Во вкладке *Редактор параметров* устройства *GoosePub* представлены информационные и конфигурационные параметры модуля (таблица 2.4). Информационные параметры не доступны для редактирования пользователем.

Таблица 2.4 – Конфигурационные параметры устройства «GoosePub»

Имя параметра	Тип	Значение по умолчанию	Описание параметра
Информация модуля			
<i>Key</i>	UDINT	1	Ключ соединения
<i>SlotReg</i>	INT	no data	Результат регистрации коммуникационного слота в системе: 0 - устройство не обработано драйвером; 1- слот добавлен в систему успешно; -1 – ошибка добавления слота в систему.
Конфигурационные параметры			
<i>Dest_MAC</i>	STRING	01:0C:CD:01:00:00	MAC адрес мультикаст группы (6 байт)
<i>Src_MAC</i>	STRING	00:00:00:00:00:00	MAC адрес источника (если поле задано 00:00:00:00:00:00, то оно не участвует в формировании фильтра сообщения и прием ведется с любым значением этого поля) (6 байт)
<i>VLAN_ID</i>	UINT	0	Номер VLAN (0 – работа без VLAN)
<i>EtherType</i>	UINT	16x88b8	Тип Ethernet кадра
<i>APP_ID</i>	UINT	16x0002	Идентификатор приложения (значение в диапазоне 0..0xFFFF)
<i>Buf_size</i>	UINT	1522	Максимальный размер буфера на прием.
<i>GoCBRef</i>	STRING		Идентификатор блока управления (ASCII строка длиной от 0 до 128 символов). Если длина строки 0, то этот параметр не участвует в формировании фильтра сообщения.
<i>datSet</i>	STRING		Идентификатор набора данных (ASCII строка длиной от 0 до 128 символов). Если длина строки 0, то этот параметр не участвует в формировании фильтра сообщения.
<i>goID</i>	STRING		Идентификатор сообщения (строка от 0 до 128 символов). Если длина строки 0, то этот параметр не участвует в формировании фильтра сообщения.
<i>confRev</i>	UDINT	1	Версия GOOSE сообщения, если содержимое пакета меняется, то по этому полю осуществляется контроль.
<i>MinTime</i>	UDINT	10	Время первого повтора передачи GOOSE сообщения, мс.
<i>Multiplier</i>	UDINT	2	Множитель времени последующих повторов передачи GOOSE сообщения, до достижения максимального значения, определенного параметром final.
<i>MaxTime</i>	UDINT	200	Максимальное время между передачей двух GOOSE сообщений, мс.

3 ПРОГРАММИРОВАНИЕ IEC850

3.1 БИБЛИОТЕКА ПОДДЕРЖКИ ПРОТОКОЛА IEC850

Для работы с **IEC850** в **CoDeSys** существует библиотека **ElsyIec850Library**. Данная библиотека добавляется автоматически. В случае удаления библиотеки пользователем, добавить библиотеку можно следующим образом:

1. В дереве устройств проекта найти **Менеджер библиотек** (либо в верхней строке меню **CodeSys** найти меню **Инструменты**→**Менеджер библиотек**). В появившейся вкладке менеджера библиотек нажать на кнопку «Добавить библиотеку» (рисунок 3.1).

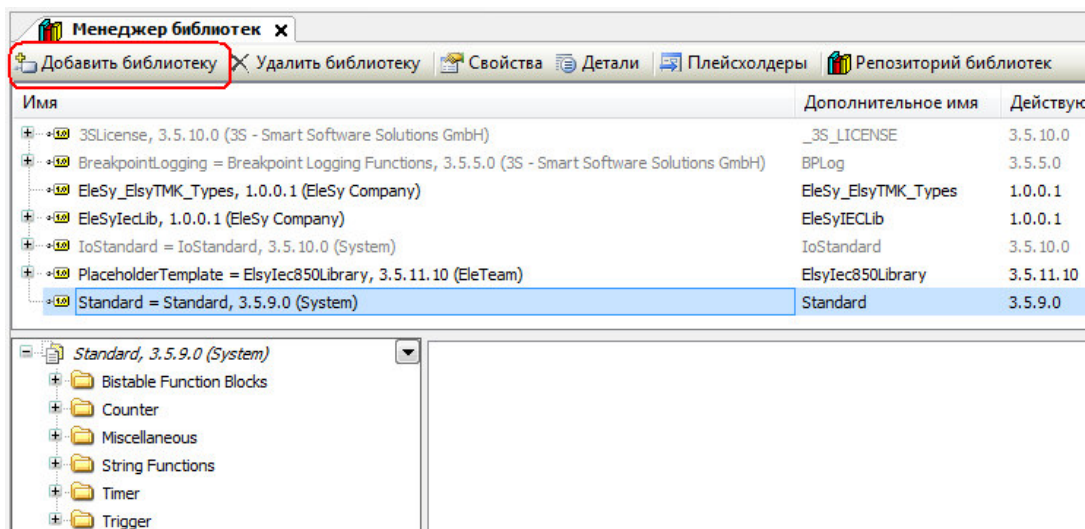


Рисунок 3.1 – Вкладка **Менеджер библиотек**. Добавление библиотеки

2. В появившемся окне **Библиотека** в верхней строке ввода ввести название библиотеки – **ElsyIec850Library**.
3. Выбрать необходимую библиотеку в столбце **Совпадение** окна **Библиотека** и нажать кнопку «ОК» (рисунок 3.2).

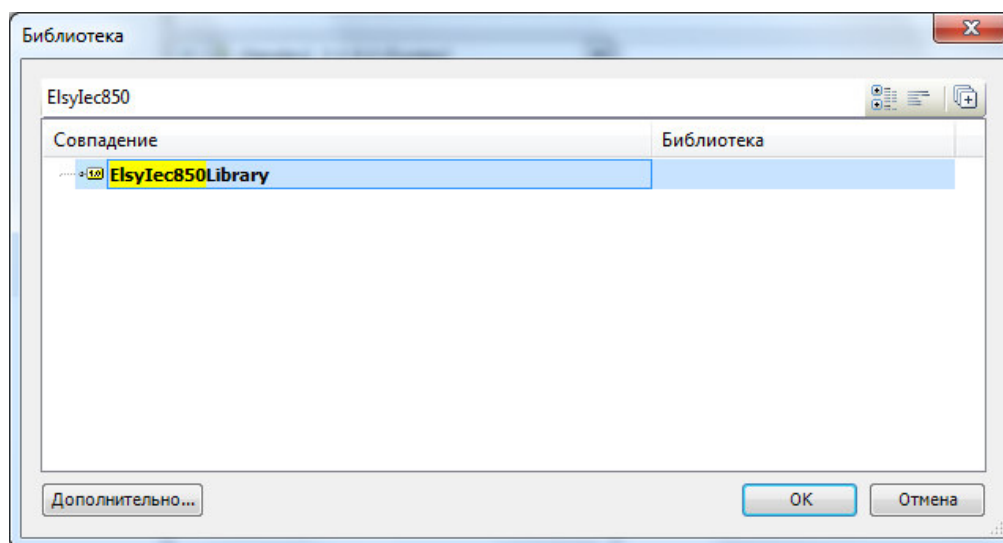


Рисунок 3.2 – Вкладка **Менеджер библиотек**. Добавление библиотеки

Библиотека **ElsyIec850Library** содержит функциональные блоки **GooseSub** и **GoosePub** для работы с **IEC850**.

Назначение ФБ **GooseSub** – связь с заданным в дереве устройством **GooseSub** и обеспечение доступа к данным из программы пользователя через данные и методы ФБ. Для включения в проект CoDeSys экземпляра функционального блока необходимо в разделе переменных указать переменную с типом **ElsyIec850Library.GooseSub**. Графическая структура ФБ **GooseSub** изображена на рисунке 3.3.

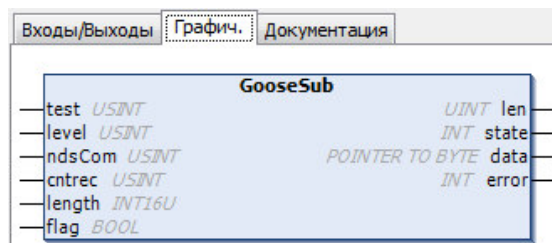


Рисунок 3.3 – Структура ФБ **GooseSub**

Описание переменных ФБ представлено в таблице 3.1.

Таблица 3.1 – Перечень переменных ФБ **GooseSub**

Имя	Тип	Описание
<i>test</i>	USINT	Признак работы в тестовом режиме.
<i>level</i>	USINT	Уровень вложения структур.
<i>ndsCom</i>	USINT	Признак неисправности оборудования.
<i>cntrec</i>	USINT	Количество записей.
<i>length</i>	INT16U	Размер datasets.
<i>flag</i>	BOOL	Флаг приема.
<i>len</i>	UINT	Принятый размер данных.
<i>state</i>	INT	Состояние ФБ, обновляется по вызову любого метода. 0 bit – инициализация ФБ: 1 – ФБ инициализирован; 0 – ошибка инициализации. 1 bit – связь ФБ с сетевой компонентой: 1 – наличие связи с сетевой компонентой; 0 – отсутствие связи с сетевой компонентой. 2 bit – нахождение ФБ в состоянии «старт/стоп»: 1 – состояние «старт»; 0 – состояние «стоп». 3 bit - наличие в буфере новых несчитанных данных: 1 – есть новые несчитанные данные; 0 – нет новых несчитанных данных. 4 bit – размер принятого кадра: 1 – нет; 0 – размер принятого кадра больше пользовательского буфера (необходимо увеличить размер буфера)
<i>data</i>	POINTER TO BYTE	Буфер под данные на прием
<i>error</i>	INT	Ошибка завершения вызова блока: 0 – нет ошибки; -1 – ошибка; Коды ошибок: 1 – данные ФБ не инициализированы; 2 – ошибка регистрации слота; 3 – ошибка запуска ФБ; 4 – ошибка остановки ФБ.

ФБ **GooseSub** содержит следующие методы:

- *CheckState* – проверка состояния ФБ.
- *Init* – обеспечивает инициализацию и связывание экземпляра ФБ с функцией открытого в системе сетевого сокета.

- *Receive* – прием пользовательских данных.
- *Stop* – блокирует прием данных (задаче данные не доступны);
- *Start* – возобновляет прием данных в задачу.

Примечание: Методы *receive*, *start* и *stop* можно вызывать только после метода *init*. Метод *checkstate* можно вызвать на любом этапе работы программы.

Более подробное описание методов представлено в приложении А.

Назначение ФБ **GoosePub** – передача Goose сообщений. Для включения в проект CoDeSys экземпляра функционального блока необходимо в разделе переменных указать переменную с типом **ElsyIec850Library.GoosePub**. Графическая структура ФБ **GoosePub** изображена на рисунке 3.4.

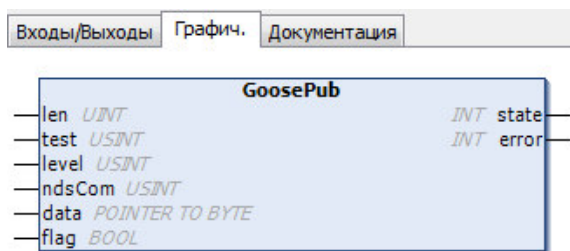


Рисунок 3.4 – Структура ФБ **GoosePub**

Описание переменных ФБ **GoosePub** представлено в таблице 3.2.

Таблица 3.2 – Перечень переменных ФБ **GoosePub**.

Имя	Тип	Описание
<i>len</i>	UINT	Размер буфера на передачу
<i>test</i>	USINT	Признак работы в тестовом режиме.
<i>level</i>	USINT	Уровень вложения структур
<i>ndsCom</i>	USINT	Признак неисправности оборудования.
<i>data</i>	POINTER TO BYTE	Буфер под данные на передачу.
<i>flag</i>	BOOL	Флаг передачи.
<i>state</i>	INT	Состояние ФБ, обновляется по вызову любого метода. 0 bit – бит инициализации ФБ: 1 – ФБ инициализирован; 0 – ошибка инициализации. 1 bit – связь ФБ с сетевой компонентой: 1 – наличие связи с сетевой компонентой; 0 – отсутствие связи с сетевой компонентой. 2 bit – нахождение ФБ в состоянии «старт/стоп»: 1 – состояние «старт»; 0 – состояние «стоп».
<i>error</i>	INT	Ошибка ФБ, обновляется по вызову любого метода. Коды завершения вызова ФБ: 0 – успешное завершение; -1 – ошибка. Коды ошибок: 1 – данные ФБ не инициализированы; 2 – ошибка регистрации слота; 3 – ошибка запуска ФБ; 4 – ошибка остановки ФБ; 5 – превышен размер передаваемого буфера; 6 – ошибка передачи.

ФБ **GoosePub** содержит следующие методы:

- *Checkstate* – проверка состояния ФБ.
- *Init* – обеспечивают инициализацию и связывание экземпляра ФБ с функцией открытого в системе сетевого сокета.
- *SetUp* - ставит на передачу заданный в устройстве буфер (Goose-сообщение).
- *Stop* – остановка передачи на системном уровне последнего пакета данных;
- *Start* – запуск передачи данных (после старта возобновляется передача старого пакета с временным разгоном (Multiplier, MinTime, MaxTime)).

П р и м е ч а н и е : Методы *setup*, *start* и *stop* можно вызывать только после метода *init*. Метод *checkstate* можно вызвать на любом этапе работы программы.

Более подробное описание методов представлено в приложении А.

В библиотеке **ElsyIec850Library** реализован набор функций для упрощения работы с данными GOOSE сообщений. Все функции разделены на 5 группы:

1. Служебные;
2. Функции группы *ByIdx*;
3. Функции группы *ByOffset*;
4. Функции группы *Publisher*;
5. Функции группы *Subscribe*.

Служебные функции предназначены для расчета определения смещения блока данных внутри GOOSE сообщения (в соответствии с протоколом IEC850 блок данных может иметь различное смещение). Группа функций *ByIdx* предназначена для получения данных по номеру индекса. Группа функций *ByOffset* предназначена для получения данных по смещению внутри Goose-сообщения. Перечень функций и их описание представлены в приложении Б.

Функции группы *Publisher* предназначены для работы при отправке Goose-сообщений. Функции группы *Subscribe* предназначены для работы при получении Goose-сообщений. Перечень функций и их описание:

Функции группы *Subscribe*:

- *Goose_GetArray* – функция получения записи Array из dataset по порядковому номеру. Возвращаемые значения:
 - 1 – ошибка;
 - 0 – ошибок нет.
- *Goose_GetBinTime* – функция получения Bin Time из dataset по порядковому номеру. Возвращаемые значения:
 - 1 – ошибка;
 - 0 – ошибок нет.
- *Goose_GetBool* – функция получения записи Bool из dataset по порядковому номеру. Возвращаемые значения:
 - 1 – ошибка;
 - 0 – ошибок нет.

- `Goose_GetByte` – функция получения записи `Byte` из `dataset` по смещению.
Возвращаемые значения:
 - 1 – ошибка;
 - 0 – ошибок нет.
- `Goose_GetDataRec` – функция получения записи из `dataset/структуры`.
Возвращаемые значения:
 - 1 – ошибка;
 - 0 – ошибок нет.
- `Goose_GetDInt` – функция получения записи `DInt` из `dataset` по смещению.
Возвращаемые значения:
 - 1 – ошибка;
 - 0 – ошибок нет.
- `Goose_DWord` – функция получения записи `DWord` из `dataset` по смещению.
Возвращаемые значения:
 - 1 – ошибка;
 - 0 – ошибок нет.
- `Goose_GetInt` – функция получения записи `Int` из `dataset` по смещению.
Возвращаемые значения:
 - 1 – ошибка;
 - 0 – ошибок нет.
- `Goose_GetLInt` – функция получения записи `LInt` из `dataset` по смещению.
Возвращаемые значения:
 - 1 – ошибка;
 - 0 – ошибок нет.
- `Goose_GetLReal` – функция получения записи `LReal` из `dataset` по смещению.
Возвращаемые значения:
 - 1 – ошибка;
 - 0 – ошибок нет.
- `Goose_GetLWord` – функция получения записи `LWord` из `dataset` по смещению.
Возвращаемые значения:
 - 1 – ошибка;
 - 0 – ошибок нет.
- `Goose_GetMString` – функция получения записи `GetMString` из `dataset` по смещению.
Возвращаемые значения:
 - 1 – ошибка;
 - 0 – ошибок нет.
- `Goose_GetOString` – функция получения записи `OString` из `dataset` по смещению.
Возвращаемые значения:

-1 – ошибка;

0 – ошибок нет.

- `Goose_GetReal` – функция получения записи `Real` из `dataset` по смещению. Возвращаемые значения:

-1 – ошибка;

0 – ошибок нет.

- `Goose_GetSInt` – функция получения записи `SInt` из `dataset` по смещению. Возвращаемые значения:

-1 – ошибка;

0 – ошибок нет.

- `Goose_GetString` – функция получения записи `String` из `dataset` по смещению. Возвращаемые значения:

-1 – ошибка;

0 – ошибок нет.

- `Goose_GetUDInt` – функция получения записи `Udint` из `dataset` по смещению. Возвращаемые значения:

-1 – ошибка;

0 – ошибок нет.

- `Goose_GetUInt` – функция получения записи `UInt` из `dataset` по смещению. Возвращаемые значения:

-1 – ошибка;

0 – ошибок нет.

- `Goose_GetULInt` – функция получения записи `ULInt` из `dataset` по смещению. Возвращаемые значения:

-1 – ошибка;

0 – ошибок нет.

- `Goose_GetUSInt` – функция получения записи `USInt` из `dataset` по смещению. Возвращаемые значения:

-1 – ошибка;

0 – ошибок нет.

- `Goose_GetUTCTime` – функция получения записи `UTS Time` из `dataset` по порядковому номеру. Возвращаемые значения:

-1 – ошибка;

0 – ошибок нет.

- `Goose_GetValue` – функция получения любой записи из `dataset` по смещению. Возвращаемые значения:

-1 – ошибка;

0 – ошибок нет.

- `Goose_GetWord` – функция получения записи `Word` из `dataset` по смещению. Возвращаемые значения:

-1 – ошибка;

0 – ошибок нет.

- `Goose_SubRead` – функция получения всех записей из `dataset`. Возвращаемые значения:

-1 – ошибка;

0 – ошибок нет.

- `Goose_SubStart` – функция получения смещения записи до `dataset`. Возвращаемые значения:

-1 – ошибка;

0 – ошибок нет.

Функции группы *Publisher*:

- `Goose_PubSend` – функция для формирования `Datasets` из массива данных. Возвращаемые значения:

-1 – ошибка;

0 – ошибок нет.

- `Goose_PubStart` – функция для сброса элементов `DataSets`. Возвращаемые значения:

-1 – ошибка;

0 – ошибок нет.

- `Goose_PutArray` – функция для формирования записи `Goose` сообщения типа `Array`. Возвращаемые значения:

-1 – ошибка;

0 – ошибок нет.

- `Goose_PutBegin` – функция для формирования записи `Goose` сообщения типа `ExStruct`. Возвращаемые значения:

-1 – ошибка;

0 – ошибок нет.

- `Goose_PutBinTime` – функция для формирования записи `Goose` сообщения типа `Bin Time`. Возвращаемые значения:

-1 – ошибка;

0 – ошибок нет.

- `Goose_PutBool` – функция для формирования записи `Goose` сообщения типа `Bool`. Возвращаемые значения:

-1 – ошибка;

0 – ошибок нет.

- `Goose_PutByte` – функция для формирования записи `Goose` сообщения типа `Byte`. Возвращаемые значения:

-1 – ошибка;

0 – ошибок нет.

- `Goose_PutDInt` – функция для формирования записи Goose сообщения типа `DInt`. Возвращаемые значения:

-1 – ошибка;

0 – ошибок нет.

- `Goose_PutDWord` – функция для формирования записи Goose сообщения типа `DWord`. Возвращаемые значения:

-1 – ошибка;

0 – ошибок нет.

- `Goose_PutEnd` – функция для формирования завершения записи Goose сообщения типа `ExStruct`. Возвращаемые значения:

-1 – ошибка;

0 – ошибок нет.

- `Goose_PutInt` – функция для формирования записи Goose сообщения типа `Int`. Возвращаемые значения:

-1 – ошибка;

0 – ошибок нет.

- `Goose_PutLInt` – функция для формирования записи Goose сообщения типа `LInt`. Возвращаемые значения:

-1 – ошибка;

0 – ошибок нет.

- `Goose_PutLReal` – функция для формирования записи Goose сообщения типа `LReal`. Возвращаемые значения:

-1 – ошибка;

0 – ошибок нет.

- `Goose_PutLWord` – функция для формирования записи Goose сообщения типа `LWord`. Возвращаемые значения:

-1 – ошибка;

0 – ошибок нет.

- `Goose_PutMString` – функция для формирования записи Goose сообщения типа `MMS String`. Возвращаемые значения:

-1 – ошибка;

0 – ошибок нет.

- `Goose_PutOString` – функция для формирования записи Goose сообщения типа `Octet String`. Возвращаемые значения:

-1 – ошибка;

0 – ошибок нет.

- `Goose_PutReal` – функция для формирования записи Goose сообщения типа `Real`.
Возвращаемые значения:
 - 1 – ошибка;
 - 0 – ошибок нет.
- `Goose_PutSInt` – функция для формирования записи Goose сообщения типа `SInt`.
Возвращаемые значения:
 - 1 – ошибка;
 - 0 – ошибок нет.
- `Goose_PutString` – функция для формирования записи Goose сообщения типа `String`.
Возвращаемые значения:
 - 1 – ошибка;
 - 0 – ошибок нет.
- `Goose_PutUDInt` – функция для формирования записи Goose сообщения типа `UDInt`.
Возвращаемые значения:
 - 1 – ошибка;
 - 0 – ошибок нет.
- `Goose_PutUInt` – функция для формирования записи Goose сообщения типа `UInt`.
Возвращаемые значения:
 - 1 – ошибка;
 - 0 – ошибок нет.
- `Goose_PutULInt` – функция для формирования записи Goose сообщения типа `ULInt`.
Возвращаемые значения:
 - 1 – ошибка;
 - 0 – ошибок нет.
- `Goose_PutUSInt` – функция для формирования записи Goose сообщения типа `USInt`.
Возвращаемые значения:
 - 1 – ошибка;
 - 0 – ошибок нет.
- `Goose_PutUTCTime` – функция для формирования записи Goose сообщения типа `UTC Time`.
Возвращаемые значения:
 - 1 – ошибка;
 - 0 – ошибок нет.
- `Goose_PutValue` – функция для формирования записи Goose сообщения.
Возвращаемые значения:
 - 1 – ошибка;
 - 0 – ошибок нет.
- `Goose_PutWord` – функция для формирования записи Goose сообщения типа `Word`.
Возвращаемые значения:

- 1 – ошибка;
- 0 – ошибок нет.

3.2 СОЗДАНИЕ ПРОЕКТА ДЛЯ РАБОТЫ С IEC850

Ниже представлены создания проектов для работы с **IEC850**.

3.2.1 Создание проекта для *GoosePub*

Для создания проекта необходимо:

1. Открыть среду разработки *CoDeSys* и создать новый *стандартный проект*, указав его имя и расположение на компьютере. В окне стандартного проекта выбрать устройство **ELSYTMK** и язык программирования **ST**.
2. Сконфигурировать проект. Для этого:
 - a. Правой кнопкой мыши нажать на устройство **Device** и выбрать в контекстном меню команду *Добавить устройство....* В появившемся окне выбрать коммутационную панель **TK711_6**.
 - b. Выбрать блок питания **TP711**, нажав на устройство **PowerSupply** и выбрав команду контекстного меню *Добавить устройство....*
 - c. Нажать на устройство **ModuleCP** правой кнопкой мыши и выбрать в контекстном меню команду *Добавить устройство....* В появившемся окне выбрать устройство **TC711_A8_100_2ETH**.
 - d. Нажать на устройство **Lan1** правой кнопкой мыши и выбрать в контекстном меню команду *Добавить устройство....* В появившемся окне выбрать устройство **IEC850**.
 - e. Нажать на устройство **IEC850** правой кнопкой мыши и выбрать в контекстном меню команду *Добавить устройство....* В появившемся окне выбрать устройство **GoosePub**.
 - f. Перейти во вкладку *Редактор параметров* для **GoosePub**, нажав на него дважды левой кнопкой мыши, и выбрать необходимые конфигурационные параметры устройства.
 - g. Сгенерировать глобальные переменные. Для этого перейти во вкладку *Редактор параметров* устройства **IEC850**, нажав на него дважды левой кнопкой мыши. Поставить галочку в графе *Create array* и нажать кнопку **«Generate»**. Сгенерированные глобальные переменные помещаются в дерево устройств в папку **GENERIC**.
3. Создать программный код:
 - a. Перейти во вкладку создания программы **PLC_PRG**, дважды нажав на нее левой кнопкой мыши.
 - b. В верхней области вкладки **PLC_PRG** ввести переменные:

```
PROGRAM PLC_PRG
VAR
    state : INT := 0;
    result : INT;
    recs : ARRAY[1..2] OF ElsylEC850Library.DataRec;
END_VAR
```

с. В нижней области вкладки **PLC_PRG** ввести код программы:

```

CASE state OF
  0: (* инициализируем *)
    result := IEC850_GVL.gpub_array[1]^init();
    IF result = 0 THEN
      state := 1;
    END_IF

  1: (* подготовка пакета *)
    result:= ElsylEC850Library.Goose_PubStart(ADR(IEC850_GVL.gpub_array[1]));
    IF result = 0 THEN
      (* подготовить записи recs*)
      result := ElsylEC850Library.Goose_PubSend(ADR(IEC850_GVL.gpub_array[1]),ADR(recs),2);
      IF result = 0 THEN
        state := 2;
      END_IF
    END_IF

  2: (* отправка пакета *)
    result := IEC850_GVL.gpub_array[1]^setup();
    IF result = 0 THEN
      state := 1;
    END_IF
END_CASE

```

Программный код начинается с инициализации, затем происходит подготовка пакета данных, подготовка записи recs и отправка пакета.

Подготовить записи recs можно тремя способами:

- С помощью функции PubSend, которая формирует DataSets из массива данных.
- С помощью функции PutValue.
- С помощью функций стандартных переменных (PutReal, PutInt и др.).

Примечание: при формировании datasets возможно использование структур, глубина вложенности структур в структурах не должна превышать 5.

- d. Сгенерировать код, выбрав в верхней панели меню **Компиляция→Генерировать код**. Устранить ошибки при их наличии.
4. Установить связь с контроллером. Для этого необходимо дважды нажать левой кнопкой мыши на устройство **Device**, перети во вкладку **Установка соединения** и нажать кнопку «Сканировать сеть...». В появившемся окне выбрать необходимое устройство.

Более подробное описание этапов создания проекта представлено в документе «Контроллер программируемый ЭЛСИ-ТМК. Руководство по применению. Часть 1».

3.2.1 Создание проекта для GooseSub

Для создания проекта необходимо:

1. Открыть среду разработки *CoDeSys* и создать новый *стандартный проект*, указав его имя и расположение на компьютере. В окне стандартного проекта выбрать устройство **ELSYTMK** и язык программирования **ST**.
2. Сконфигурировать проект. Для этого:

- a. Правой кнопкой мыши нажать на устройство **Device** и выбрать в контекстном меню команду **Добавить устройство...**. В появившемся окне выбрать коммутационную панель **TK711_6**.
- b. Выбрать блок питания **TP711**, нажав на устройство **PowerSupply** и выбрав команду контекстного меню **Добавить устройство...**
- c. Нажать на устройство **ModuleCP** правой кнопкой мыши и выбрать в контекстном меню команду **Добавить устройство...**. В появившемся окне выбрать устройство **TC711_A8_100_2ETH**.
- d. Нажать на устройство **Lan1** правой кнопкой мыши и выбрать в контекстном меню команду **Добавить устройство...**. В появившемся окне выбрать устройство **IEC850**.
- e. Нажать на устройство **IEC850** правой кнопкой мыши и выбрать в контекстном меню команду **Добавить устройство...**. В появившемся окне выбрать устройство **GooseSub**.
- f. Перейти во вкладку **Редактор параметров** для **GooseSub**, нажав на него дважды левой кнопкой мыши, и выбрать необходимые конфигурационные параметры устройства.
- g. Сгенерировать глобальные переменные. Для этого перейти во вкладку **Редактор параметров** устройства **IEC850**, нажав на него дважды левой кнопкой мыши. Поставить галочку в графе **Create array** и нажать кнопку **«Generate»**. Сгенерированные глобальные переменные помещаются в дерево устройств в папку **GENERIC**.

5. Создать программный код:

- a. Перейти во вкладку создания программы **PLC_PRG**, дважды нажав на нее левой кнопкой мыши.

- b. В верхней области вкладки **PLC_PRG** ввести переменные:

```
PROGRAM PLC_PRG
VAR
    state : INT := 0;
    result : INT;
    recs : ARRAY[1..2] OF Elsylec850Library.DataRec;
    gsub : POINTER TO Elsylec850Library.GooseSub;
END_VAR
```

- c. В нижней области вкладки **PLC_PRG** ввести код программы:

```
gsub := ADR(IEC850_GVL.gsub_array[1]);

CASE state OF
    0: (* инициализируем *)
        result := gsub^.init();
        IF result = -1 THEN RETURN; END_IF;
        state := 1;

    1: (* получение пакета *)
        (*прием данных*)
        result := gsub^.receive();
        IF result = -1 THEN RETURN; END_IF;

    (*установка datasets в начало *)
    result := Elsylec850Library.Goose_SubStart(gsub);
    IF result = -1 THEN RETURN; END_IF;
```

```

(*извлекаем данные из datasets*)
result := Elsylec850Library.Goose_GetValue(gsub,1,ADR(recs[1]));
END_CASE

```

Программный код начинается с инициализации, затем происходит получение пакета данных, которое включает в себя прием данных и установку datasets в начало.

Прием данных можно произвести тремя способами:

- С помощью функции SubRead, которая получает все записи из datasets.
- С помощью функции GetValue.
- С помощью функций стандартных переменных (GetReal, GetInt и др.).

- d. Сгенерировать код, выбрав в верхней панели меню **Компиляция→Генерировать код**. Устранить ошибки при их наличии.
6. Установить связь с контроллером. Для этого необходимо дважды нажать левой кнопкой мыши на устройство **Device**, перейти во вкладку **Установка соединения** и нажать кнопку «Сканировать сеть...». В появившемся окне выбрать необходимое устройство.

Более подробное описание этапов создания проекта представлено в документе «Контроллер программируемый ЭЛСИ-ТМК. Руководство по применению. Часть 1».

3.2.2 Пример программного кода для работы с IEC850

Ниже представлен пример программного кода выделения данных GOOSE сообщения с помощью набора функций групп *ByIdx* и *ByOffset* библиотеки **ElsyIec850Library**. В данном примере осуществляется прием данных по смещению данных в Goose сообщении.

```

PROGRAM parsers_test
VAR CONSTANT
  RX_LEN : UDINT := 1522; // размер буфера
END_VAR

VAR_OUTPUT
  // переменные для хранения данных
  // элементы структуры
  s_byVar : BYTE;
  s_uiVar : UINT;
  s_udiVar : UDINT;
  s_bVar : BOOL;
  s_iVar : INT;
  s_rVar : REAL;
  // базовые элементы
  fVar : REAL;
  udiVar : UDINT;

  CntErr_InitGoose : UDINT; // счетчики ошибок
  Errors : INT;             // значение ошибки
END_VAR

VAR
  goosesub : Elsylec850Library.GooseSub; // Добавление в программу ФБ GooseSub
  gs_result : INT;                       // результат, возвращаемый с методов ФБ
  RX_BUF : ARRAY [0..RX_LEN] OF BYTE;   // буфер для хранения кадра
  mainstate : UINT;                     // автомат состояний
END_VAR

```

```
CASE mainstate OF
0: // установка параметров для принимаемых данных
   goosesub.Key_GOOSE_slot := 1;      // связывание с ключом слота
   goosesub.Rx_data := ADR(RX_BUF);    // указатель на буфер
   goosesub.Rx_len := RX_LEN;          // значение длины буфера
   mainstate := 1;

1: // инициализация блока
   gs_result := goosesub.init();
   IF gs_result <> -1 THEN               // не было ошибки при инициализации
      mainstate := 2;
   ELSE
      CntErr_InitGoose := CntErr_InitGoose + 1;
      Errors := goosesub.error;
      mainstate := 1001;                // состояние ошибки - не обрабатывается
   END_IF

2: // прием данных (цикл)
   gs_result := goosesub.receive();      //результат метода приема данных
   IF gs_result > 0 THEN
      PARSE();                        // извлечение данных
   END_IF;
END_CASE
```

Программный код для метода **PARSE()** представлен ниже:

```
METHOD PARSE
VAR_INPUT
END_VAR
VAR
   NDSE : INT;                                // число записей в dataset

   mDATASET_POINTER : POINTER TO UDINT;       // указатель на тэг dataset
   DATASET_OFFSET : INT;                      // смещение до dataset

   mSTRUCT_POINTER : POINTER TO UDINT;        // указатель на тэг пользовательской структуры
   STRUCT_OFFSET : INT;                       // смещение до структуры во фрейме

   // смещения на базовые элементы
   fVar_OFFSET : INT;
   udiVar_OFFSET : INT;

   // результат выполнения
   RESULT : INT;
   pRESULT : POINTER TO INT := ADR(RESULT);
END_VAR

(* получить число записей в data set *)
NDSE := Elsylec850Library.Goose_GetNumDataSetEntries( ADR(RX_BUF) );

(* получить смещение data set *)
DATASET_OFFSET := Elsylec850Library.Goose_GetDatasetOffset( ADR(RX_BUF) );

(* проверить на ошибки смещения *)
IF DATASET_OFFSET <> -1 AND NDSE <> -1 THEN

   (* получить базовые элементы ***** *)
```

```

(* установить указатель на тэг dataset *)
mDATASET_POINTER := ADR( RX_BUF[DATASET_OFFSET] );

(* получить элементы по смещению - не безопасно хранить offset и использовать для новых кадров *)
(* определить смещение *)
fVar_OFFSET := Elsylec850Library.Goose_GetEntryOffsetByIdx( mDATASET_POINTER, 2);
udiVar_OFFSET := Elsylec850Library.Goose_GetEntryOffsetByIdx( mDATASET_POINTER, 3);
(* получить элементы *)
fVar := Elsylec850Library.Goose_GetFloatByOffset(mDATASET_POINTER, fVar_OFFSET, pRESULT);
udiVar := Elsylec850Library.Goose_GetInt32UByOffset(mDATASET_POINTER, udiVar_OFFSET, pRESULT);
(* ИЛИ получить элементы по индексу - вычисление offset при каждом вызове *)
// сброс
fVar := 0;
udiVar := 0;
// получение элементов
fVar := Elsylec850Library.Goose_GetFloatByIdx(mDATASET_POINTER, 2, pRESULT);
udiVar := Elsylec850Library.Goose_GetInt32UByIdx(mDATASET_POINTER, 3, pRESULT);

(* получить элементы структуры ***** *)
(* получить смещение структуры *)
STRUCT_OFFSET := Elsylec850Library.Goose_GetEntryOffsetByIdx( mDATASET_POINTER, 1);
IF STRUCT_OFFSET <> -1 THEN
    (* установка указателя на тэг структуры *)
    mSTRUCT_POINTER := mDATASET_POINTER + INT_TO_DWORD(STRUCT_OFFSET); // или
    (* извлечение элементов из структуры по индексу *)
    s_byVar := Elsylec850Library.Goose_GetByteByIdx( mSTRUCT_POINTER, 1, pRESULT);
    s_uiVar := Elsylec850Library.Goose_GetInt16UByIdx( mSTRUCT_POINTER, 2, pRESULT);
    s_udiVar := Elsylec850Library.Goose_GetInt32UByIdx( mSTRUCT_POINTER, 3, pRESULT);
    s_bVar := Elsylec850Library.Goose_GetBoolByIdx( mSTRUCT_POINTER, 4, pRESULT);
    s_iVar := Elsylec850Library.Goose_GetInt16ByIdx( mSTRUCT_POINTER, 5, pRESULT);
    s_rVar := Elsylec850Library.Goose_GetFloatByIdx( mSTRUCT_POINTER, 6, pRESULT);
END_IF
END_IF

```

ПРИЛОЖЕНИЕ А
(СПРАВОЧНОЕ)
ПЕРЕЧЕНЬ МЕТОДОВ БИБЛИОТЕКИ ELSYIEC850LIBRARY

Методы ФБ **GooseSub**:

- *CheckState* – проверка состояния ФБ. Метод возвращает следующие значения:
 - 0 – успешное завершение;
 - 1 – неудачное завершение.
- *Init* – обеспечивает инициализацию и связывание экземпляра ФБ с функцией открытого в системе сетевого сокета. Метод возвращает следующие коды завершения:
 - 0 – успешное завершение;
 - 1 – неудачное завершение (код ошибки должен быть установлен на выходе error).
- *Receive* – прием пользовательских данных. Метод возвращает следующие значения:
 - 1 – ошибка;
 - 0 – успешное завершение, нет данных;
 - >0 – успешное завершение, размер полученных данных.
- *Start* – возобновляет прием данных в задачу. Метод возвращает следующие значения:
 - 0 – успешное завершение;
 - 1 – неудачное завершение (код ошибки должен быть установлен на выходе error).
- *Stop* – блокирует прием данных (задаче данные не доступны). Метод возвращает следующие значения:
 - 0 – успешное завершение;
 - 1 – неудачное завершение (код ошибки должен быть установлен на выходе error).

Методы ФБ **GoosePub**:

- *CheckState* – проверка состояния ФБ. Метод возвращает следующие значения:
 - 0 – успешное завершение;
 - 1 – неудачное завершение.
 - *Init* – обеспечивает инициализацию и связывание экземпляра ФБ с функцией открытого в системе сетевого сокета. Метод возвращает следующие коды завершения:
 - 0 – успешное завершение;
 - 1 – неудачное завершение (код ошибки должен быть установлен на выходе error).
 - *Setup* – ставит на передачу заданный в устройстве буфер. Метод принимает следующие значения:
 - False* – сформирован пакет в режиме «работа»;
 - True* – сформирован пакет в режиме «тест».
- Возвращает следующие коды ошибок:

0 – успешное завершение, нет данных;

-1 – успешное завершение, размер полученных данных.

- *Stop* – остановка передачи на системном уровне последнего пакета данных. Метод возвращает следующие значения:

0 – успешное завершение;

-1 – неудачное завершение (код ошибки должен быть установлен на выходе error).

- *Start* – запуск передачи данных (после старта возобновляется передача старого пакета с временным разгоном (Multiplier, MinTime, MaxTime)). Метод возвращает следующие значения:

0 – успешное завершение;

-1 – неудачное завершение (код ошибки должен быть установлен на выходе error).

ПРИЛОЖЕНИЕ Б
(СПАРВОЧНОЕ)
ПЕРЕЧЕНЬ ФЕНКЦИЙ БИБЛИОТЕКИ ELSYIEC850LIBRARY

Перечень функций библиотеки **ElsyIec850Library**:

Служебные функции:

- *Goose_GetDatasetOffset* – функция получения смещения на тэг dataset в кадре goose. Возвращает следующие значения:
 - 1 – ошибка вычисления смещения;
 - 0..n – смещение на тэг dataset.
- *Goose_GetNumDataSetEntries* – функция получения числа записей в dataset кадра goose. Возвращает следующие значения:
 - 1 – ошибка;
 - 0..n – число записей в dataset.

Функции группы ByIdx:

- *Goose_GetBoolByIdx* – функция получения bool записи из dataset/структуры по порядковому номеру. Результат выполнения помещается по указателю RESULT:
 - 1 – ошибка;
 - 0 – нет ошибок, функция возвращает запрашиваемое значение.
- *Goose_GetByteByIdx* – функция получения int8u записи из dataset/структуры по порядковому номеру. Результат выполнения помещается по указателю RESULT:
 - 1 – ошибка;
 - 0 – нет ошибок, функция возвращает запрашиваемое значение.
- *Goose_GetEntryOffsetByIdx* – функция получения смещения записи из dataset/структуры по порядковому номеру. Возвращает следующие значения:
 - 1 – ошибка;
 - 0..n – смещение в dataset на тэг записи.
- *Goose_GetFloatByIdx* – функция получения float32 записи из dataset/структуры по порядковому номеру. Результат выполнения помещается по указателю RESULT:
 - 1 – ошибка;
 - 0 – нет ошибок, функция возвращает запрашиваемое значение.
- *Goose_GetInt16ByIdx* – функция получения int16 записи из dataset/структуры по порядковому номеру. Результат выполнения помещается по указателю RESULT:
 - 1 – ошибка;
 - 0 – нет ошибок, функция возвращает запрашиваемое значение.
- *Goose_GetInt16UByIdx* – функция получения int16u записи из dataset/структуры по порядковому номеру. Результат выполнения помещается по указателю RESULT:
 - 1 – ошибка;
 - 0 – нет ошибок, функция возвращает запрашиваемое значение.

- *Goose_GetInt32UByIdx* – функция получения int32u записи из dataset/структуры по порядковому номеру. Результат выполнения помещается по указателю RESULT:
-1 – ошибка;
0 – нет ошибок, функция возвращает запрашиваемое значение.

Функции группы ByOffset:

- *Goose_GetBoolOffset* – функция получения bool записи из dataset/структуры по смещению. Результат выполнения помещается по указателю RESULT:
-1 – ошибка;
0 – нет ошибок, функция возвращает запрашиваемое значение.
- *Goose_GetByteOffset* – функция получения int8u записи из dataset/структуры по смещению. Результат выполнения помещается по указателю RESULT:
-1 – ошибка;
0 – нет ошибок, функция возвращает запрашиваемое значение.
- *Goose_GetfloatOffset* – функция получения float32 записи из dataset/структуры по смещению. Результат выполнения помещается по указателю RESULT:
-1 – ошибка;
0 – нет ошибок, функция возвращает запрашиваемое значение.
- *Goose_GetInt16Offset* – функция получения int16 записи из dataset/структуры по смещению. Результат выполнения помещается по указателю RESULT:
-1 – ошибка;
0 – нет ошибок, функция возвращает запрашиваемое значение.
- *Goose_GetInt16UOffset* – функция получения int16u записи из dataset/структуры по смещению. Результат выполнения помещается по указателю RESULT:
-1 – ошибка;
0 – нет ошибок, функция возвращает запрашиваемое значение.
- *Goose_GetInt32UOffset* – функция получения int32u записи из dataset/структуры по смещению. Результат выполнения помещается по указателю RESULT:
-1 – ошибка;
0 – нет ошибок, функция возвращает запрашиваемое значение.

П р и м е ч а н и е : более подробное описание функций представлено во вспомогательной документации среды программирования *CoDeSys*.

Список литературы

- 1) «Контроллер программируемый ЭЛСИ-ТМК. Руководство по применению».
- 2) «Разработка ПО для поддержки протокола GOOSE в ПЛК ЭЛСИ-ТМК. Частное техническое задание».

Контактная информация

По всем вопросам, связанным с эксплуатацией контроллера, обращаться в сервисный центр АО «ЭлеСи»:

тел.: +7 (3822) 49-94-94

E-mail: service@elesy.ru

Сервисный центр располагается в г. Томске (часовой пояс +4 МСК).

При обращении просим сообщать следующие данные:

- полное наименование изделия (указано на изделии или в паспорте);
- проект *CoDeSys*, в котором возникает проблема;
- версия установленного на компьютере пакета *EleSy PLC ELSYTMK TSP (Target Support Package)*;
- подробное описание проблемы (попытайтесь наиболее полно пояснить суть проблемы и обстоятельства или условия, которые привели к ней). приемнр