



ПРОГРАММИРУЕМЫЕ ЛОГИЧЕСКИЕ КОНТРОЛЛЕРЫ

Библиотеки WORK_MODE и Ind_Mode Руководство по применению

Версия 01

Москва

2010

Содержание

Введение.....	3
Словарь условных сокращений и терминов	4
1. Установка дополнительных библиотек для проекта	5
2. Состав библиотеки WORK_MODE	8
Функция очистки экрана (ClearScreen)	8
Получение номера текущего рабочего экрана (GetWorkScreen).....	9
Получение числа рабочих экранов (GetWorkScreenCount)	10
Установка номера текущего экрана (SetWorkScreen).....	11
Установка числа рабочих экранов (SetWorkScreenCount).....	12
Вывод числа типа DINT в указанной позиции (ShowDint).....	13
Вывод числа типа Real в указанной позиции (ShowReal).....	14
Вывод строки в указанной позиции (ShowString)	15
3. Состав библиотеки Ind_Mode	17
Очистка содержимого ЖКИ в монопольном режиме (Clear_Display).....	18
Получение текущего режима работы ЖКИ (Get_Display_Mode)	19
Чтение строки с ЖКИ в рабочем режиме индикации (ReadDisplay)	20
Переход в монопольный режим работы ЖКИ (Switch_To_Single_Mode).....	21
Возврат из рабочего монопольного режима в предшествующий режим (Switch_To_PREV_Mode).....	22
Переход из монопольного режима в режим редактирования параметра (Switch_To_EDITOR).....	23
Вывод строки на ЖКИ в монопольном режиме индикации (Write_Display)	27
Приложение А. Строка формата для вывода численных значений на ЖКИ из функций ShowReal и ShowDint.....	29
Приложение Б. Отображаемые на ЖКИ символы и их коды	31
Лист изменений в версиях документа	33

Введение

Компания ОВЕН предоставляет пользователю библиотеки дополнительных программных компонентов, облегчающие составление проекта работы программируемого логического контроллера (ПЛК) для решения наиболее распространенных практических задач. Приводимые в данном описании библиотеки предназначены для работы на контроллерах ОВЕН с встроенными жидкокристаллическими индикаторами (ЖКИ) для управления отображением информации на экране ЖКИ.

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

Библиотека WORK_MODE (файл WORK_MODE.lib) используется для работы ПЛК с отображением информации на экране ЖКИ.

Библиотека Ind_Mode (файл Ind_Mode.lib) используется для отображения символьной информации на экране ЖКИ и для переключения между режимами индикации.

Назначение всех компонентов библиотек указано в таблицах 1 и 2.

Таблица 1 – Библиотека WORK_MODE

Имя компонента	Назначение и область применения
ClearScreen	Очистка экрана, номер которого передан аргументом на вход функции
GetWorkScreen	Получение номера текущего рабочего экрана
GetWorkScreenCount	Получение числа (количества) рабочих экранов
SetWorkScreen	Установка номера текущего экрана
SetWorkScreenCount	Установка числа (количества) рабочих экранов
ShowDint	Вывод на заданный экран в заданную позицию числа типа DINT
ShowReal	Вывод на заданный экран в заданную позицию числа типа Real
ShowString	Вывод на заданный экран в заданную позицию строки текста

Таблица 2 – Библиотека Ind_Mode

Имя компонента	Назначение и область применения
Clear_Display	Очистка содержимого ЖКИ в монопольном режиме
Get_Display_Mode	Получение текущего режима работы ЖКИ
Switch_To_Single_Mode	Переход в рабочий монопольный режим работы ЖКИ
Read_Display	Чтение строки с ЖКИ в монопольном режиме индикации
Switch_To_PREV_Mode	Возврат из рабочего монопольного режима в предшествующий режим индикации.
Switch_To_EDITOR	Возврат из рабочего монопольного режима в редактор параметра
Write_Display	Вывод строки на ЖКИ в монопольном режиме индикации

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.
ЖКИ	–	жидкокристаллический индикатор (на лицевой панели ПЛК).
ПЛК	–	программируемый логический контроллер.
РП	–	руководство пользователя на программируемый логический контроллер (поставляется на компакт-диске вместе с контроллером).
РЭ	–	руководство по эксплуатации на программируемый логический контроллер (поставляется вместе с контроллером).
0 и 1	–	при описании переменных типа BOOL нулю соответствует значение «FALSE»; единице – значение «TRUE».

В тексте используются следующие технические термины:

Экран – пронумерованная совокупность данных, программно заданных пользователем для отображения на ЖКИ. Число доступных для вызова на ЖКИ экранов (рабочих экранов) задается функцией «setworkscreencount», либо (если указанная функция не вызывалась) задается в параметре «Quantity of work screens» модуля «WorkInd» в разделе «Конфигурация ПЛК (PLC Configuration)», см. РП. Максимально в режиме индикации может быть доступно до 16 экранов (нумеруемых от 0 до 15), каждый из которых может быть выведен на ЖКИ.

Монопольный режим индикации (single_mode) – режим индикации, при котором на ЖКИ контроллера отображается только один выбранный экран и заблокированы системные функции клавиатуры.

Рабочий режим индикации (work_mode) – режим, при котором пользователь может управлять выбором отображаемого на ЖКИ экрана и работой ПЛК при помощи кнопок на лицевой панели.

Режим меню (menu_mode) – режим отображения меню текущей конфигурации ПЛК (заданного в рабочей программе ПЛК), предназначен для выбора элемента меню на ЖКИ.

Режим редактирования (editor_mode) – режим редактирования элемента меню на ЖКИ.

Режим отладочного меню (debug_mode) – служебный режим работы, предназначенный для выполнения сервисных функций.

Текущий экран – экран, который отображается на ЖКИ в текущий момент времени.

Дескриптор – структура, однозначно описывающая выбранный параметр дерева «Конфигурация ПЛК (PLC_configuration)»: значение, минимум, максимум, адрес modbus и т. д. (см. РП).

Тип данных – формат представления переменных согласно РП.

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.

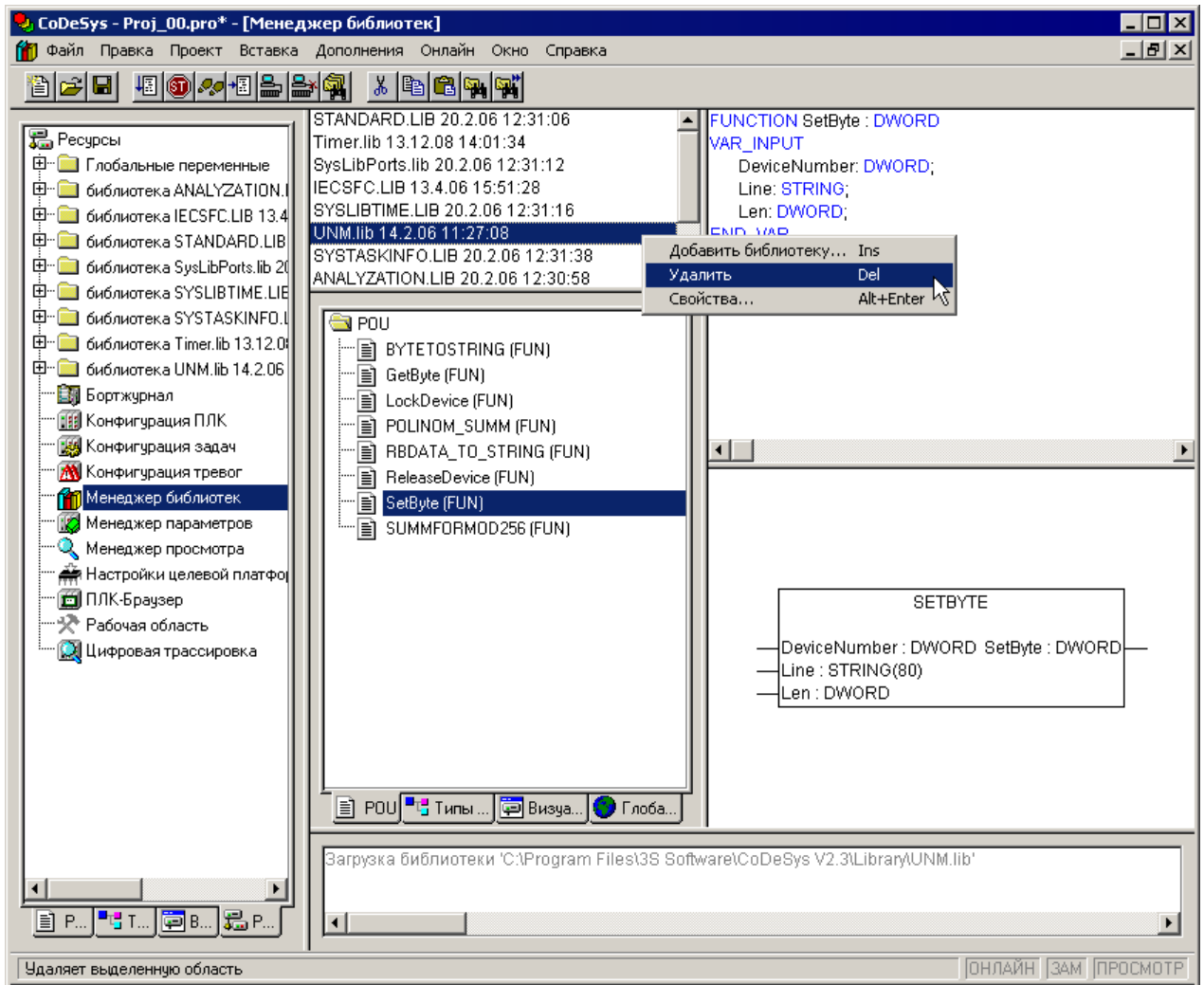


Рисунок 1.1 – Окно вкладки организатора объектов «Ресурсы (Resources)» с режимом работы «Менеджер библиотек (Library Manager)»

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, WORK_MODE.lib) и дается команда **Открыть**.

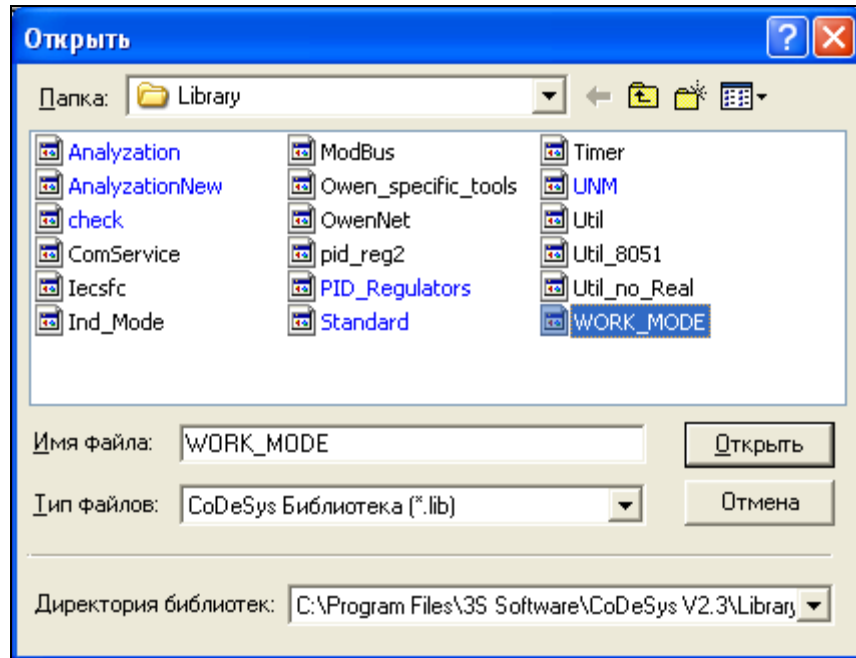


Рисунок 1.2 – Окно выбора подключаемой к проекту дополнительной библиотеки

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

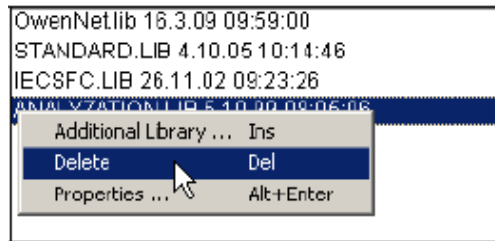
Для просмотра состава и свойств программных компонентов курсором выбирается нужная библиотека, – при этом появится папка с программными компонентами, в которой выделяется конкретный программный компонент (на рисунке 1.1 справа дана краткая справочная информация по его использованию).

Примечания.

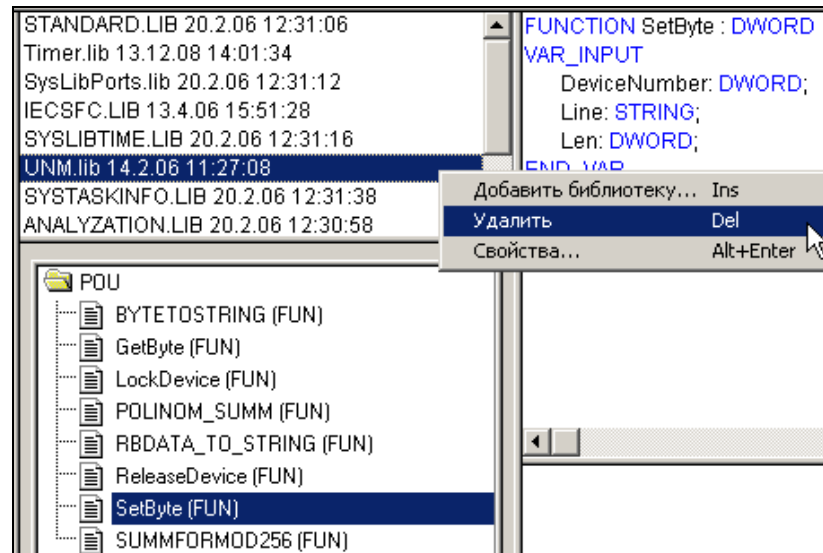
1. Рекомендуется размещать все библиотеки, которые планируется подключать, в папке для хранения библиотек, создаваемой CoDeSys автоматически.

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 WORK_MODE

Данная библиотека предназначена для работы в экранном рабочем режиме (режиме ЖКИ по умолчанию). Ее рекомендуется использовать для отображения цифровых значений и символьных строк на ЖКИ при выполнении пользовательской программы.

Примечание – Во избежание мигания индикации запрещается вывод на один и тот же экран информации чаще, чем с интервалом в 200 мс.

Функция очистки экрана (ClearScreen)

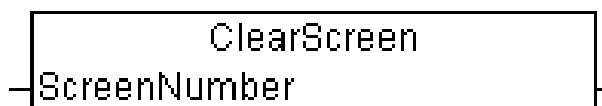


Рисунок 2.1 – Структурная схема

Таблица 2.1

Имя программного компонента	ClearScreen		
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		
Входная переменная:	Тип данных	Пояснения	
ScreenNumber	DINT	Номер очищаемого экрана (значение от 0 до [число рабочих экранов -1]) – задается пользователем	
Выходная переменная:	Тип данных	Пояснения	
ClearScreen	DINT	Возвращаемое значение – номер очищенного экрана или -1, в случае ошибки	

Описание работы

Функция очищает экран, номер которого передан аргументом на вход. Первый экран всегда имеет номер 0.

Например, для очистки экрана с номером 0 в программу необходимо вставить строку вызова функции:

```
ClearScreen(0);
```


Получение номера текущего рабочего экрана (GetWorkScreen)

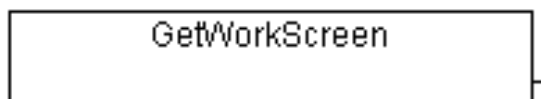


Рисунок 2.2 – Структурная схема

Таблица 2.2

Имя программного компонента	GetWorkScreen		
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		
Выходная переменная:	Тип данных	Пояснения	
GetWorkScreen	DINT	Возвращаемое значение – номер текущего экрана или –1, в случае ошибки	

Описание работы

Функция возвращает номер текущего рабочего экрана.

Например, для получения номера текущего рабочего экрана в программу необходимо вставить строку вызова функции:

```
qty_screen:=GetWorkScreen();
```

Получение числа рабочих экранов (GetWorkScreenCount)

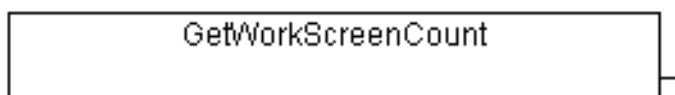


Рисунок 2.3 – Структурная схема

Таблица 2.3

Имя программного компонента	GetWorkScreenCount		
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		
Выходная переменная:	Тип данных	Пояснения	
GetWorkScreenCount	DINT	Возвращаемое значение – число рабочих экранов	

Описание работы

Функция возвращает число рабочих экранов.

Например, для получения числа рабочих экранов в программу необходимо вставить строку вызова функции:

```
qty_screen:=GetWorkScreenCount();
```

Установка номера текущего экрана (SetWorkScreen)

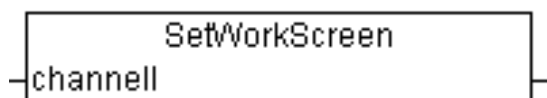


Рисунок 2.4 – Структурная схема

Таблица 2.4

Имя программного компонента	SetWorkScreen		
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		
Входная переменная:	Тип данных	Пояснения	
channell	DINT	Номер экрана (значение от 0 до [число рабочих – 1]) – задается пользователем	
Выходная переменная:	Тип данных	Пояснения	
SetWorkScreen	DINT	Возвращаемое значение – строка с шестнадцатеричным кодом символа (номер установленного экрана или –1, в случае ошибки)	

Описание работы

Функция устанавливает текущим экран, номер которого передан аргументом на вход.

Первый экран всегда имеет номер 0. Например, для установки второго экрана (имеющего номер 1) в программу необходимо вставить строку вызова функции:

```
SetWorkScreen(1);
```

Установка числа рабочих экранов (SetWorkScreenCount)

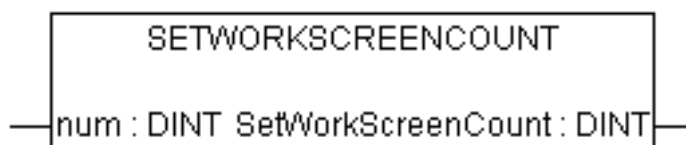


Рисунок 2.3 – Структурная схема

Таблица 2.3

Имя программного компонента	SetWorkScreenCount		
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		
Входная переменная:	Тип данных	Пояснения	
num	DINT	Устанавливаемое число рабочих экранов	
Выходная переменная:	Тип данных	Пояснения	
SetWorkScreenCount	DINT	Возвращаемое значение – текущее число рабочих экранов	

Описание работы

Функция устанавливает число рабочих экранов. Задаваемое число экранов должно находиться в границах, заданных в дереве PLC Configuration во ветви WorkInd во вкладке module_parameters в параметре Quantity of work screens.

Например, для установки числа рабочих экранов = 4, в программу необходимо вставить строку вызова функции:

```
qty_screen:=SetWorkScreenCount(4);
```

Вывод числа типа DINT в указанной позиции (ShowDint)

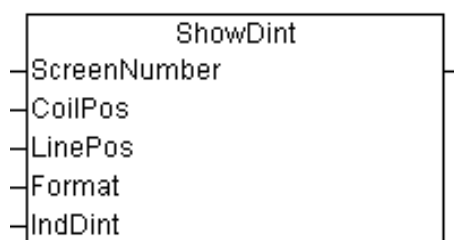


Рисунок 2.5 – Структурная схема

Таблица 2.5

Имя программного компонента	ShowDint		
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		
Входные переменные:	Тип данных	Пояснения	
ScreenNumber	DINT	Номер экрана (значение от 0 до [число рабочих экранов -1])	
CoilPos	DINT	Номер столбца (значение от 0 до 15) – номер символа, считая с левого крайнего, начиная с которого будет осуществлен вывод информации (задается пользователем)	
LinePos	DINT	Номер строки (значение от 0 до 3*) в которой будет осуществлен вывод информации (задается пользователем)	
Format	String[16]	Строка формата (аналогичная строке формата функции PrintF языка Си, см. Приложение А) – задается пользователем	
IndDint	DINT	Выводимое число на ЖКИ	
Выходная переменная:	Тип данных	Пояснения	
ShowDint	DINT	Возвращаемое значение – 1 при успешном выводе или -1, в случае ошибки	
* Для контроллера с меньшим количеством строк на дисплее максимальное значение должно соответствовать РЭ.			

Описание работы

Функция выводит на заданный экран в заданную позицию данные типа DINT (целое число).

Первый экран всегда имеет номер 0. Например, для отображения на экране с номером 1 ЖКИ числового значения, поступающего с канала аналогового входа (имеющего имя **rom**), необходимо вставить в программу строку вызова функции:

```
ShowDint(0,0,0,'%d',rom);
```

Выводимые на экран ЖКИ символы приведены в приложении Б.

Вывод числа типа Real в указанной позиции (ShowReal)

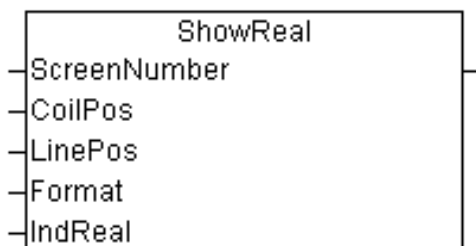


Рисунок 2.6 – Структурная схема

Таблица 2.6

Имя программного компонента	ShowReal		
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		
Входные переменные:	Тип данных	Пояснения	
ScreenNumber	DINT	Номер экрана (значение от 0 до [число рабочих экранов -1])	
CoilPos	DINT	Номер столбца (значение от 0 до 15) – номер символа, считая с левого крайнего, начиная с которого будет осуществлен вывод информации (задается пользователем)	
LinePos	DINT	Номер строки (значение от 0 до 3*) в которой будет осуществлен вывод информации (задается пользователем)	
Format	String[16]	Строка формата (аналогичная строке формата функции PrintF языка Си, см. Приложение А) – задается пользователем	
IndReal	Real	Выводимое число на ЖКИ	
Выходная переменная:	Тип данных	Пояснения	
ShowReal	DINT	Возвращаемое значение – 1 при успешном выводе или -1, в случае ошибки	
* Для контроллера с меньшим количеством строк на дисплее максимальное значение должно соответствовать РЭ.			

Описание работы

Функция выводит на заданный экран ЖКИ в заданную позицию число типа Real (числа с плавающей точкой).

Первый экран всегда имеет номер 0. Например, для отображения на экране с номером 1 значения температуры, поступающего с канала аналогового входа (имеющего имя **term**), необходимо вставить в программу строку вызова функции:

```
ShowReal(0,0,0,'%2.1f',term);
```

Вывод строки в указанной позиции (ShowString)

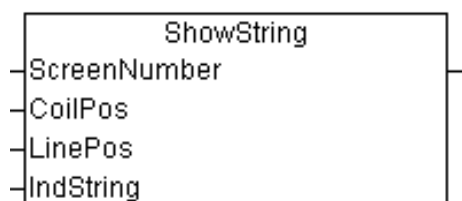


Рисунок 2.7 – Структурная схема

Таблица 2.7

Имя программного компонента	ShowString		
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		
Входные переменные:	Тип данных	Пояснения	
ScreenNumber	DINT	Номер экрана (значение от 0 до [число рабочих экранов -1])	
CoilPos	DINT	Номер столбца (значение от 0 до 15) – номер символа, считая с левого крайнего, начиная с которого будет осуществлен вывод информации (задается пользователем)	
LinePos	DINT	Номер строки (значение от 0 до 3*) в которой будет осуществлен вывод информации (задается пользователем)	
IndString	String[16]	Выводимая на ЖКИ строка (задается пользователем)	
Выходная переменная:	Тип данных	Пояснения	
ShowString	DINT	Возвращаемое значение – 1 при успешном выводе или -1, в случае ошибки	
* Для контроллера с меньшим количеством строк на дисплее максимальное значение должно соответствовать РЭ.			

Описание работы

Функция выводит на заданный экран ЖКИ в заданную позицию строку.

Первый экран всегда имеет номер 0. Например, для отображения строки на экране с номером 1 в позиции 3 на строке 2 текста '**Hello world**', необходимо вставить в программу строку вызова функции:

```
ShowString (1, 2, 1, 'Hello world');
```


3. Состав библиотеки Ind_Mode

Библиотека предназначена для переключения между режимами индикации ЖКИ и для вывода строковых констант на ЖКИ в монопольном рабочем режиме. Библиотеку следует использовать при создании нестандартных способов отображения сообщений на ЖКИ, когда функций экранного рабочего режима не хватает.

Библиотека использует следующие структуры данных для описания режимов работы ЖКИ:

```

TYPE DISP_MODES:
(DEBUG_MODE,      – отладочный режим индикации (служебный режим)
 WORK_MODE,      – рабочий экранный режим индикации
 MENU_MODE,      – конфигурационный режим индикации (переходы по меню)
 SINGLE_MODE,    – рабочий монопольный режим индикации
 EDITOR_MODE,    – редактирование параметра
 MAX_MODES      – служебный параметр, определяющий количество
                  режимов, не изменять!
);
END_TYPE

```

Дополнительные структуры пользовательских типов данных (PARAMETER_DESCRIPTOR и PARAM_NODE), доступные через «Ассистент ввода» (рисунок 3.1), пользователем не редактируются.

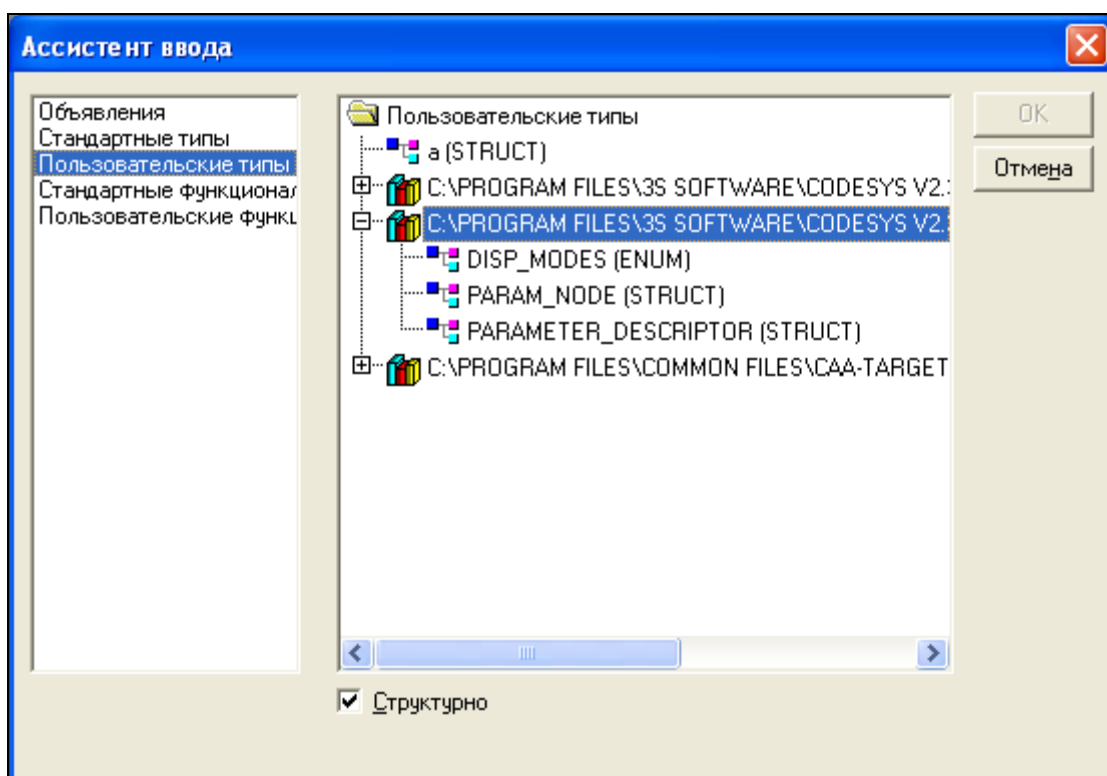


Рисунок 3.1 – Окно установки пользовательских типов данных

Примечание – Во избежание мигания индикации запрещается вывод на один и тот же экран информации чаще, чем с интервалом в 200 мс.

Очистка содержимого ЖКИ в монопольном режиме (Clear_Display)

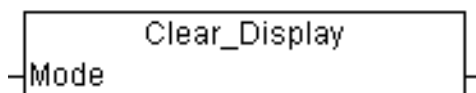


Рисунок 3.2 – Структурная схема

Таблица 3.1

Имя программного компонента	Clear_Display		
Тип программного компонента	Функциональный блок <input type="checkbox"/>	Функция <input checked="" type="checkbox"/>	Программа <input type="checkbox"/>
Особенности работы	Для работы требуется подключение к проекту библиотеки arm7_specific.lib		
Применение на контроллерах	ПЛК63, ПЛК73, ПЛК410		
Входная переменная:	Тип данных	Пояснения	
Mode	DISP_MODES	Режим работы ЖКИ (задается пользователем). Значение должно быть всегда Single_MODE	
Выходная переменная:	Тип данных	Пояснения	
Clear_Display	BOOL	Возвращаемое значение 1 в случае успешного выполнения операции	

Описание работы

Функция очищает ЖКИ монопольного режима работы в независимости от того, каков текущий режим ЖКИ.

Пример вызова данной функции на языке ST будет выглядеть так:

```
Switch_To_Single_Mode();      (*переход в монопольный режим*)
Clear_Display(SINGLE_MODE);   (*очищаем экран*)
```

Получение текущего режима работы ЖКИ (Get_Display_Mode)

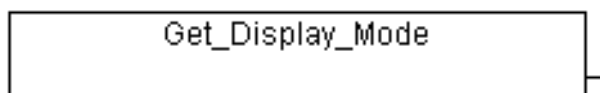


Рисунок 3.3 – Структурная схема

Таблица 3.2

Имя программного компонента	Get_Display_Mode		
Тип программного компонента	Функциональный блок <input type="checkbox"/>	Функция <input checked="" type="checkbox"/>	Программа <input type="checkbox"/>
Особенности работы	Для работы требуется подключение к проекту библиотеки arm7_specific.lib		
Применение на контроллерах	ПЛК63, ПЛК73, ПЛК410		
Выходная переменная:	Тип данных	Пояснения	
Get_Display_Mode	DISP_MODES	Возвращаемое значение – текущий режим работы ЖКИ	

Описание работы

Функция возвращает текущий режим работы ЖКИ.

Например, пусть требуется замкнуть выход №1 при входе в режим конфигурирования, а в рабочем экранном режиме разомкнуть. Тогда фрагмент программы на языке ST будет выглядеть так:

```
IF (Get_Display_Mode())=MENU_MODE) THEN out1:=1; END_IF
IF (Get_Display_Mode())=WORK_MODE) THEN out1:=0; END_IF
```

Чтение строки с ЖКИ в рабочем режиме индикации (ReadDisplay)

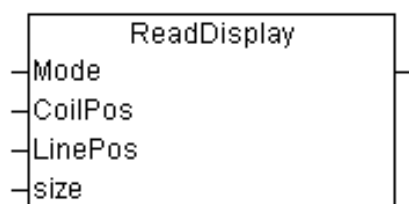


Рисунок 3.4 – Структурная схема

Таблица 3.3

Имя программного компонента	ReadDisplay		
Тип программного компонента	Функциональный блок <input type="checkbox"/>	Функция <input checked="" type="checkbox"/>	Программа <input type="checkbox"/>
Особенности работы	Для работы требуется подключение к проекту библиотеки arm7_specific.lib		
Применение на контроллерах	ПЛК63, ПЛК73, ПЛК410		
Входные переменные:	Тип данных	Пояснения	
Mode	DISP_MODES	Режим работы ЖКИ (задается пользователем). Может принимать значения Work_mode либо Single_mode	
CoilPos	BYTE	Номер столбца (значение 0 или 15) – количество символов от начала строки (задается пользователем)	
LinePos	BYTE	Номер строки (значение от 0 до 3*) в которой будет осуществлен вывод информации (задается пользователем)	
size	DWORD	Количество считываемых с ЖКИ символов (задается пользователем). Значение должно быть от 0 до 16.	
Выходная переменная:	Тип данных	Пояснения	
ReadDisplay	String[16]	Считанная строка	

Описание работы

Функция возвращает символьную строку, считывая ее с ЖКИ.

Например, пусть требуется выводить надпись на индикатор и считывать ее в монопольном режиме. Тогда фрагменты программы на языке ST будут выглядеть:

```
Switch_To_Single_Mode(); (*переходим в монопольный режим*)
Write_Display(SINGLE_MODE, 0, 0, 'abcdef'); (*выводим надпись в монопольном режиме*)
a:=ReadDisplay(SINGLE_MODE, 0, 0, 3); (*считываем 3 символа в монопольном режиме*)
```

В результате выполнения программного кода переменная **a** примет значение **'abc'**. Выводим надпись на индикатор и считываем ее в рабочем режиме индикации:

```
ShowString(0, 0, 0, 'abcdef'); (*выводим надпись в экранном режиме*)
a:=ReadDisplay(WORK_MODE, 0, 0, 3); (*считываем 3 символа в экранном режиме*)
(*В результате выполнения программного кода переменная a примет значение 'abc'*)
```

Переход в монопольный режим работы ЖКИ (Switch_To_Single_Mode)

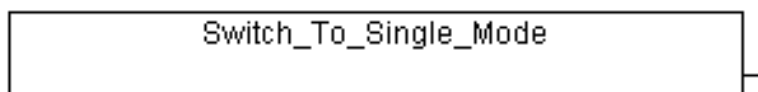


Рисунок 3.5 – Структурная схема

Таблица 3.4

Имя программного компонента	Switch_To_Single_Mode		
Тип программного компонента	Функциональный блок <input type="checkbox"/>	Функция <input checked="" type="checkbox"/>	Программа <input type="checkbox"/>
Особенности работы	Для работы требуется подключение к проекту библиотеки arm7_specific.lib		
Применение на контроллерах	ПЛК63, ПЛК73, ПЛК410		
Выходная переменная:	Тип данных	Пояснения	
Switch_To_Single_Mode	BOOL	Возвращаемое значение 1 в случае успешного выполнения операции перехода	

Описание работы

Функция переводит ЖКИ в рабочий монопольный режим.

Переход в монопольный режим не возможен из режима отладочного меню (DEBUG_MODE) или из режима редактирования параметра (EDITOR_MODE).

Пример вызова данной функции приведен при описании использования функции **Switch_To_PREV_Mode**, см. ниже.

Возврат из рабочего монопольного режима в предшествующий режим (Switch_To_PREV_Mode)

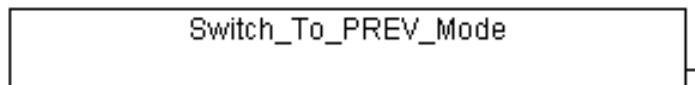


Рисунок 3.6 – Структурная схема

Таблица 3.5

Имя программного компонента	Switch_To_PREV_Mode		
Тип программного компонента	Функциональный блок <input type="checkbox"/>	Функция <input checked="" type="checkbox"/>	Программа <input type="checkbox"/>
Особенности работы	Для работы требуется подключение к проекту библиотеки arm7_specific.lib		
Применение на контроллерах	ПЛК63, ПЛК73, ПЛК410		
Выходная переменная:	Тип данных	Пояснения	
Switch_To_PREV_Mode	BOOL	Возвращаемое значение 1 в случае успешного выполнения операции перехода	

Описание работы

Функция переводит ПЛК в режим индикации, предшествовавший текущему режиму индикации.

Пример вызова данной функции для поочередного (последовательного) перевода ЖКИ контроллера в экранный режим и монопольный режим и, соответственно, указанием в каком режиме находится индикация, на языке ST будет выглядеть так:

```

PROGRAM PLC_PRG
VAR
PROGRAM PLC_PRG
VAR
    ton1: TON;           (*таймер задержки*)
    starting:BOOL:=FALSE;
    mode:BOOL:=FALSE;   (*текущий режим FALSE – экранный режим TRUE – монопольный режим*)
END_VAR

IF NOT starting THEN   (*то что выполняется один раз при запуске программы*)
    SetWorkScreenCount(1);
    ClearScreen(0);
    starting:=TRUE;
END_IF;
ShowString(0, 0, 0, 'Экранный режим');           (*выводим надпись "Экранный режим"*)
ton1(in:=NOT ton1.q,pt:=t#3000ms);              (*организуем задержку*)
IF ton1.Q THEN
    IF mode=FALSE THEN
        Switch_To_Single_Mode();                 (*переходим в монопольный режим*)
        Clear_Display(SINGLE_MODE);              (*очищаем экран*)
    
```

```

Write_Display(SINGLE_MODE, 0, 0, 'Монопольн. режим');    (*выводим надпись
                                                         "Монопольный режим"*)
mode:=TRUE;
ELSE
    Switch_To_PREV_Mode();    (*переходим обратно в экранный режим*)
    mode:=FALSE;
END_IF
END_IF
    
```

Переход из монопольного режима в режим редактирования параметра (Switch_To_EDITOR)

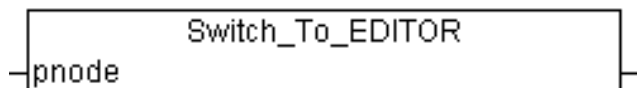


Рисунок 3.7 – Структурная схема

Таблица 3.6

Имя программного компонента	Switch_To_EDITOR		
Тип программного компонента	Функциональный блок <input type="checkbox"/>	Функция <input checked="" type="checkbox"/>	Программа <input type="checkbox"/>
Особенности работы	Для работы требуется подключение к проекту библиотеки arm7_specific.lib		
Применение на контроллерах	ПЛК63, ПЛК73, ПЛК410		
Входная переменная:	Тип данных	Пояснения	
pnode		Режим работы ЖКИ (задается пользователем)	
Выходная переменная:	Тип данных	Пояснения	
Switch_To_EDITOR	PARAM_NJDE*	Описывает параметры для редактирования	
* Структура данных определена в библиотеке, см. вкладку «Data types».			



Описание работы

Функция переводит ЖКИ в режим EDITOR_MODE (режим редактирование параметра конфигурации).

Особенности работы:

1. Перевод в **editor_mode** возможен только из режима **single_mode**;
2. Для перевода из режима **work_mode** необходимо:
 - а) запомнить номер отображаемого экрана;
 - б) перейти из режима **single_mode** в **work_mode**;
 - в) подождать 1 сек;
 - г) переключиться в режим **editor_mode**;
 - д) после завершения редактирования прибор автоматически перейдет в режим **single_mode**;

- е) как только после редактирования текущий режим установится **single_mode**, необходимо подождать 1 сек и подать команду на переключение в предыдущий режим;
- ж) после возврата в режим **work_mode** прибор производит индикацию экрана 0, соответственно, необходимо подать команду на смену отображаемого экрана на запомненный в пункте (а).

Программа работы ПЛК63 для регулировки температуры в печи по 2-х позиционному закону с заданной уставкой и гистерезисом будет иметь вид, представленный ниже. При этом уставку можно редактировать как через меню прибора, так и посредством быстрого доступа к уставке. Быстрый доступ к уставке вызывается комбинацией клавиш  +  на лицевой панели контроллера.

```

PROGRAM PLC_PRG
VAR
    ton1,ton2: TON;                (*таймер задержки*)
    starting:BOOL:=FALSE;
    temp:REAL;
    df:DECODE_FLOAT;
    param_ptr:POINTER TO parameter_descriptor;
    error:BYTE:=0;                (*код ошибки*)
    pnode_: DWORD;
    flag: BBOOL:= FALSE;
    flag1: BOOL:= FALSE;
    flag2: BOOL:= FALSE;
    dm:disp_modes;
    ft: F_TRIG;
END_VAR
VAR CONSTANT
    key:BYTE:=136;                (*код сочетания альт+стрелка вверх*)
END_VAR

IF NOT starting THEN            (*то, что выполняется один раз при запуске программы*)
    SetWorkScreenCount(1);
    ClearScreen(0);
    ShowString(0, 0, 0, 'Темп.:');    (*выводим надпись *)
    ShowString(0, 0, 1, 'Уставка:');  (*выводим надпись *)
    param_ptr:=READ_BY_HASH(16#37B7,0); (*чтение дескриптора параметра*)
    pnode_:=param_ptr^.pnode;        (*ссылка на элемент меню в pnode_*)
    ver:='1.0';                      (*внесем информацию о версии ПО*)
    starting:=TRUE;                  (*установим флаг чтобы больше не входить в этот цикл*)
END_IF;
df(value:=ain1, def_value:=0,_ERR=>error,out_value=>temp); (*декодируем измеренное значение*)

IF error=0 THEN (*если нет ошибки*)
    IF ain1>sp1+hys1 THEN (*выключение выхода, если температура выше уставки + гистерезис*)
        out1:=0.0;
    END_IF
    IF ain1<sp1-hys1 THEN (*включение выхода, если температура ниже уставки – гистерезис*)
        out1:=1.0;
    END_IF

```



```

ELSE (*если есть ошибка измерения*)
    out1:=0.0;
END_IF
ton1(in:=NOT ton1.q,pt:=t#200ms); (*индикацию обновляем раз в 200 мс*)
IF ton1.Q THEN
    ShowString(0,6,0,' '); (*сотрем что было написано*)
    ShowString(0,8,1,' '); (*сотрем что было написано*)
    IF error=0 THEN (*если измерение корректное*)
        ShowReal(0, 6, 0, '%2.1f',ain1); (*выводим значение температуры*)
    ELSE
        ShowString(0,6,0,'ошибка'); (*выводим надпись "ошибка" при
                                         некорректном измерении*)
    END_IF
    ShowReal(0,8,1,'%2.1f',sp1); (*И уставку*)
END_IF
ft(clk:=kbrd=key); (*если пользователь нажал заданное сочетание клавиш*)
IF ft.Q THEN (*отработаем по отпусканию клавиш*)
    flag:=TRUE; (*установим флаг первой стадии редактирования*)
    flag1:=TRUE; (*и флаг необходимости переключения в single_mode*)
END_IF
IF flag THEN (*первая стадия редактирования*)
    IF flag1 THEN (*если необходимо переключение в single_mode*)
        Switch_To_Single_Mode(); (*осуществим переключение в single_mode*)
        flag1:=FALSE; (*сбросим флаг необходимости перехода в single_mode*)
    END_IF
    ton2(in:=TRUE,pt:=t#1000ms); (*с момента запроса перехода в режим
                                     редактирования подождем 1000 мс*)
    IF ton2.Q THEN (*1000 мс прошло?*)
        Switch_To_EDITOR(pnode_); (*переключаемся в режим редактора*)
        flag:=FALSE; (*сбрасываем флаг первой стадии редактирования*)
        flag2:=TRUE; (*устанавливаем флаг второй стадии редактирования*)
        ton2(in:=FALSE,pt:=t#1000ms); (*сбрасываем таймер ton2*)
    END_IF
END_IF
IF flag2 THEN (*последняя стадия – отслеживаем редактирование*)
    dm:=Get_Display_Mode(); (*в dm занесем текущий режим*)
    IF dm<>editor_mode THEN (*пользователь вышел из редактирования? тогда в dm
                                     будет single_mode*)
        ton2(in:=TRUE,pt:=t#1000ms); (*подождем после выхода 1000 мс*)
        IF ton2.Q THEN (*1000 мс прошло?*)
            Switch_To_PREV_Mode(); (*переключаемся в предыдущий режим,
                                         то есть в work_mode*)
            flag2:=FALSE; (*сбросим флаг второй стадии редактирования*)
            ton2(in:=FALSE,pt:=t#1000ms); (*сбросим таймер ton2*)
        END_IF
    END_IF
END_IF
END_IF

```

Пример находится на компакт-диске в папке «samples», файл **ModbusPLC63 чтение INT и REAL (ST).pro**.

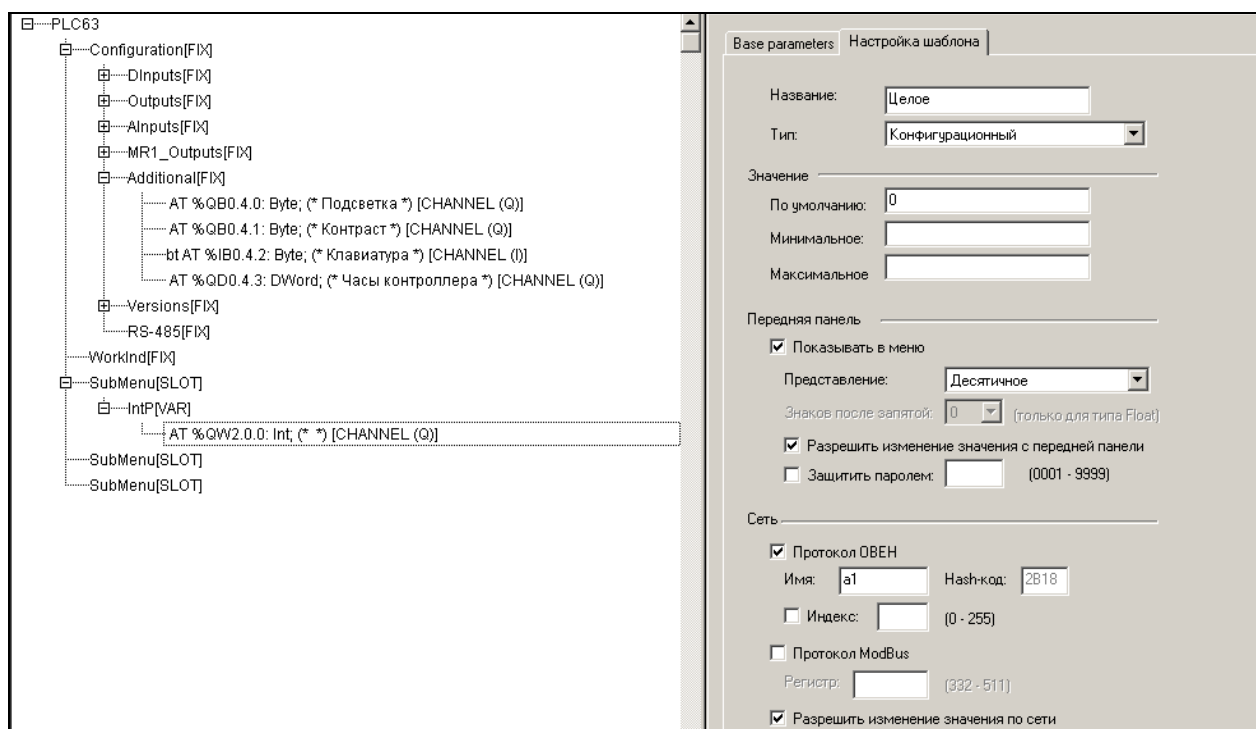


Рисунок 3.8 – Окно режима «Конфигурация ПЛК (PLC Configuration)»

Вывод строки на ЖКИ в монопольном режиме индикации (Write_Display)

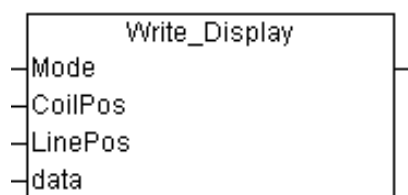


Рисунок 3.9 – Структурная схема

Таблица 3.7

Имя программного компонента	Write_Display		
Тип программного компонента	Функциональный блок <input type="checkbox"/>	Функция <input checked="" type="checkbox"/>	Программа <input type="checkbox"/>
Особенности работы	Для работы требуется подключение к проекту библиотеки arm7_specific.lib		
Применение на контроллерах	ПЛК63, ПЛК73, ПЛК410		
Входные переменные:	Тип данных	Пояснения	
Mode	DISP_MODES	Режим работы ЖКИ (задается пользователем). Должен принимать только значение Single_mode	
CoilPos	DINT	Номер столбца (значение от 0 до 15) – количество символов от начала строки (задается пользователем)	
LinePos	DINT	Номер строки (значение от 0 до 3*) (задается пользователем)	
data	String[16]	Выводимая на ЖКИ строка (задается пользователем)	
Выходная переменная:	Тип данных	Пояснения	
Write_Display	BOOL	Возвращаемое значение 1 в случае успешного выполнения операции	
* Для контроллера с меньшим количеством строк на дисплее максимальное значение должно соответствовать РЭ.			

Описание работы

Функция выводит на ЖКИ строку в монопольном режиме работы ЖКИ.

Пример вызова функции: Пусть требуется выводить сообщения о возникновении аварийной ситуации на ЖКИ в любом случае, даже когда ЖКИ находится в режиме Конфигурирование. Тогда фрагмент программы на языке ST будет выглядеть так:

```
IF ALARM THEN
  IF (GET_DISPLAY_MODE()<>SINGLE_MODE) THEN
    flag:=SWITCH_TO_SINGLE_MODE(); (*переход в рабочий монопольный
                                   режим*)
  END_IF;
  IF flag THEN
    Clear_Display(SINGLE_MODE); (*очистка ЖКИ в рабочем монопольном
                                режиме*)
    Write_Display(SINGLE_MODE,5,0, 'Авария!'); (*вывод на ЖКИ строки
                                                сообщения*)
    Write_Display(SINGLE_MODE, 0,1, 'Для возврата нажми Выход');
  ELSE
    flag:=SWITCH_TO_SINGLE_MODE(); (*повторная попытка перехода в рабочий
                                    монопольный режим*)
  END_IF;
  END_IF;
  IF Btn=64 THEN (*если нажата кнопка "выход" то*)
    Switch_To_PREV_Mode(); (*возврат из рабочего монопольного режима*)
  END_IF;
```

(*Переход между режимами осуществляется не мгновенно (может потребовать до 0,5 с) и возможен не из всех режимов. Для гарантированного перехода может потребоваться многократный вызов функции SWITCH_TO_SINGLE_MODE*)

Приложение А. Строка формата для вывода численных значений на ЖКИ из функций ShowReal и ShowDint

Строка формата для задания значений функций имеет вид:

%[ширина][точность] [флаги] тип

Длина строки не должна превышать 10 символов.

Обязательными составными частями строки являются символы начала управляющей последовательности (%) и тип (например, '%3d'); остальных полей может не быть.

Назначение полей в строке формата функции приведено ниже.

Тип

Тип указывает на конкретное представление выводимой величины (например, числа могут выводиться в десятичном или шестнадцатеричном виде). Записывается в виде одного символа. В отличие от остальных полей является обязательным.

Значения типов для функции ShowDint:

- d, i** – десятичное знаковое число. По умолчанию записывается с правым выравниванием, знак пишется только для отрицательных чисел;
- o** – восьмеричное беззнаковое число;
- u** – десятичное беззнаковое число;
- x и X** – шестнадцатеричное число (**x** – использует строчные буквы (abcdef); **X** – прописные (ABCDEF)).

Значения типов для функции ShowReal:

- f и F** – числа с плавающей запятой; по умолчанию выводятся с точностью 6; если число по модулю меньше единицы, то перед десятичной точкой пишется 0; аргумент по умолчанию имеет размер 4 байта;
- e и E** – числа с плавающей запятой в экспоненциальной форме записи (вида 1.1e+44) (**e** – выводит символ «e» в нижнем регистре; **E** – в верхнем (3.14E+0));
- g и G** – число с плавающей запятой. Форма представления зависит от значения величины (**f** или **e**);
- a и A** – число с плавающей запятой в шестнадцатеричном виде.

Флаги

Флаги указывают в каком виде должна отображаться информация на ЖКИ. Перечень «флагов», применяемых в строке формата, приведен в таблице А.1.

Таблица А.1

Знак	Название знака	Выполняемое действие	При отсутствии этого знака	Примечание
-	дефис	выводимое значение выравнивается по левому краю в пределах минимальной ширины поля	выводимое значение выравнивается по правому краю	
+	плюс	всегда указывается знак (плюс или минус) для выводимого десятичного числового значения	указывается знак только для отрицательных чисел	
	пробел	помещать перед результатом пробел, если первый символ значения не знак	вывод может начинаться с цифры	Символ + имеет больший приоритет, чем пробел. Используется только для десятичных числовых значений
0	ноль	дополнять поле символом 0 до ширины, указанной в поле ширина управляющей последовательности	дополнять поле пробелами	Для типов d, i, o, u, x, X , если точность указана, этот флаг игнорируется. Для остальных типов поведение не определено

Точность

Точность задается в виде точки с последующим десятичным числом. Если число (присутствует только точка), то предполагается, что число равно нулю. Точка для указания точности используется даже в том случае, если при выводе чисел с плавающей запятой выводится запятая.

Точность указывает:

- 1) на минимальное количество символов, которое должно появиться при обработке типов **d, i, o, u, x, X**;
- 2) на минимальное количество символов, которое должно появиться после десятичной запятой (точки) при обработке типов **a, A, e, E, f, F**;
- 3) максимальное количество значащих символов для типов **g** и **G**.

Ширина

Ширина (десятичное число) указывает минимальную ширину поля (включая знак для чисел). Если представление величины больше, чем ширина поля, то запись выходит за пределы поля (например, **%2i** для величины 100 даст значение поля в три символа). Если представление величины меньше указанного числа, то оно будет дополнено по умолчанию пробелами справа, поведение может меняться предшествующими флагами.

Приложение Б. Отображаемые на ЖКИ символы и их коды

При необходимости функции «ShowDInt», «ShowReal» и «ShowString» библиотеки Work_Mode позволяют вывести на заданную позицию (или позиции) ЖКИ требуемые специализированные символы.

Доступные для отображения символы и их коды приведены в таблице Б.1.

В верхней строке таблицы приведены значения старшего бита кода символа, в первом левом столбце приведены значения младшего бита кода. В клетках таблицы приведены символы, отображаемые на ЖКИ при заданном значении кода. В рамке приведено такое изображение символа, каково оно на ЖКИ; справа от него приведено буквенно-цифровое обозначение символа (если таковое существует).

Например, строка 1, столбец С. В ячейке на их пересечении отображена буква Б. Это значит, что символу с кодом С1 соответствует на дисплее символ «Б».

Таблица Б.1

		Старший бит кода символа													
		2	3	4	5	6	7	8	9	A	B	C	D	E	F
Младший бит кода символа	0		0	@	P		p					А	Р	а	р
	1	!	1	А	Q	а	q					Б	С	б	с
	2	"	2	В	R	b	r					В	Т	в	т
	3	#	3	С	S	с	s					Г	У	г	у
	4	\$	4	D	Т	d	t					Д	Ф	д	ф
	5	%	5	E	U	e	u					Е	Х	е	х
	6	&	6	F	V	f	v					Ж	Ц	ж	ц
	7		7	G	W	g	w					З	Ч	з	ч
	8	(8	H	X	h	x			Ё	ё	И	Ш	и	ш
	9)	9	I	Y	i	y					Й	Щ	й	щ

Продолжение таблицы Б.1

		Старший бит кода символа													
		2	3	4	5	6	7	8	9	A	B	C	D	E	F
Младший бит кода символа	A	*	:	J	Z	j	z			K		Ъ	Ь	к	ъ
	B	+	;	K	[k			O	a		Л	Ы	л	ы
	C	,	<	L	I				¼	+		M	Ъ	м	ь
	D	-	=	M]	m	E	N	O	§		H	Э	h	э
	E	.	>	N	^	n	e	I	½	M		O	Ю	o	ю
	F	/	?	O	_	o	e	W	¾			П	Я	п	я

Лист изменений в версиях документа

Номер версии	Дата выпуска	Содержание изменений
01	10.06.2010	Новый документ