ООО «ТРАНСМАШ-ТОМСК»

# ПО «Интегрированная среда разработки «МАЯК»

Руководство пользователя

# АННОТАЦИЯ

Настоящий программный документ представляет собой руководство пользователя программного обеспечения (ПО) «Интегрированная среда разработки «МАЯК», предназначенное для разработки и выполнения прикладных программ для программируемого логического контроллера (ПЛК) «МАЯК».

В разделе «Назначение программы» содержатся сведения о назначении программы и информация, достаточная для понимания функций программы и ее эксплуатации.

В разделе «Условия выполнения программы» указаны условия, необходимые для выполнения программы (минимальный состав аппаратурных и программных средств).

В разделе «Выполнение программы» описана последовательность действий оператора, обеспечивающих загрузку, запуск, выполнение и завершение программы, приведено описание функций, формата и возможных вариантов команд, с помощью которых оператор осуществляет загрузки и управляет выполнением программы, а также ответы программы на эти команды. Также раздел содержит ряд примеров использования разных языков при программировании программы.

В разделе «Сообщения оператору» приведены тексты сообщений, выдаваемых в ходе выполнения программы, описание их содержания и соответствующие действия оператора.

Настоящий документ разработан в соответствии с ГОСТ 19.505-79 «Единая система программной документации. Руководство оператора. Требования к содержанию и оформлению» и ГОСТ 19.106-78 «Единая система программной документации. Требования к программным документам, выполненным печатным способом».

# СОДЕРЖАНИЕ

Аннотация	2
1 Назначение программы	4
1.1 Назначение программы	4
1.2 Поддерживаемые функции	4
2 Условия выполнения программы	5
2.1 Программное обеспечение, необходимое для функционирования программ	ы5
2.2 Используемые технические средства	5
3 Выполнение программы	6
3.1 Установка и запуск	6
3.1.1 Установка	6
3.1.2 Запуск	9
3.1.3 Завершение работы с программой	9
3.2 Меню программы	9
3.2.1 Пункт меню «Файл»	9
3.2.2 Пункт меню «Редактировать»	10
3.2.3 Пункт меню «Вид»	12
3.2.4 Пункт меню «Помощь»	12
3.3 Панель инструментов	12
	14
3.4.1 Логика и алгоритм расоты программного олока	14 14
3.4.3 Добавление конфигурационных деременных проекта	14
3 4 3 1 Лобавление конфигурационных констант и переменных	18
3.4.3.2 Функциональное назначение кнопок, расположенных на пан	іели 10
переменных и констант	19
3.4.4 Написание алгоритма и логики выполнения программы	20
3.4.5 Пример создания функционального олока на языке FBD	20
3.4.0 Пример создания функционального блока на языке LD	22 24
3.4.8 Pecvnc	27
3.5 Сборка (компиляция и компоновка) и перелача полученного исполняем	лого
файла на целевое устройство	28
3.5.1 Сборка проекта в режиме симуляции	28
3.5.2 Генерация программы для последующей загрузки в программируе	мый
логических контроллер	28
4 Сообщения оператору	30
4.1 Информационное сообщение	30
4.2 Диалоговое сообщение	30
Приложение А (обязательное) Перечень и описание адре	есов
входов/выходов ПЛК	32

# 1 НАЗНАЧЕНИЕ ПРОГРАММЫ

## 1.1 Назначение программы

ПО «Интегрированная среда разработки «МАЯК» предназначено для разработки прикладных программ для ПЛК. ПО «Интегрированная среда разработки «МАЯК» осуществляет поддержку всех пяти языков программирования стандарта МЭК 61131-3:

- IL (Instruction List) – ассемблер-подобный язык;

- ST (Structured Text) – Pascal-подобный язык;

- LD (Ladder Diagram) – язык релейных схем;

- FBD (Function Block Diagram) – язык функциональных блоков;

– SFC (Sequential Function Chart) – язык диаграмм состояний.

# 1.2 Поддерживаемые функции

ПО «Интегрированная среда разработки «МАЯК» обеспечивает выполнение следующих функций:

– поддержка языков программирования стандарта ГОСТ Р МЭК 61131-3-2016;

– формирование файла алгоритма для его последующей загрузки в Среду исполнения «МАЯК»;

- обновление версии программы посредством сети Internet;

– возможность создания пользовательских библиотек, содержащих отдельные комплексные алгоритмические блоки для их последующего применения в проектах;

– поддержка промышленных сетевых протоколов обмена данными и управления: Modbus TCP, Modbus ASCII/RTU, TCP/IP, HTTP;

- возможность применения режима отладки, включая:
  - мониторинг/изменение/фиксацию значений переменных, точки останова;
  - контроль потока выполнения;
  - горячее обновление кода;
  - другие отладочные инструменты;
- синтаксический и пунктуационный механизм отслеживания.

# 2 УСЛОВИЯ ВЫПОЛНЕНИЯ ПРОГРАММЫ

# 2.1 Программное обеспечение, необходимое для функционирования программы

Программное обеспечение разработано для использования на компьютере с операционной системой Windows.

В текущей версии ПО «Интегрированная среда разработки «МАЯК» поддерживается только операционной системой Windows 10/11.

#### 2.2 Используемые технические средства

Минимальные системные требования, предъявляемые к конфигурации компьютера, следующие:

1) Процессор (CPU). Двухъядерный процессор с тактовой частотой от 1.0 ГГц и выше. Рекомендуется более мощный процессор для увеличения скорости работы.

2) Оперативная память (RAM). Минимум 2 ГБ оперативной памяти (RAM). Рекомендуется 4 ГБ и более для более комфортной работы, особенно при работе с крупными проектами.

3) Свободное дисковое пространство. Минимум 1 ГБ свободного дискового пространства для установки ПО «Интегрированная среда разработки «МАЯК» и необходимых компонентов.

4) Графический адаптер (GPU). Встроенный графический адаптер с поддержкой OpenGL 2.0 и выше. Рекомендуется иметь дискретный графический адаптер для более быстрой работы и 3D-графики.

5) Сетевое соединение. Для загрузки программы на физический ПЛК и взаимодействия с сетью убедитесь, что у вас есть подключение к сети Ethernet или другой необходимой сети.

6) Разрешение экрана. Рекомендуется экран с разрешением 1280×800 пикселей или выше для удобного просмотра и редактирования программ.

7) Дополнительное оборудование. В зависимости от конкретных задач и требований проекта, могут потребоваться дополнительные устройства, такие как программаторы ПЛК или симуляторы.

# з выполнение программы

#### 3.1 Установка и запуск

## 3.1.1 Установка

Установка программы выполняется следующим образом:

1) Запустить установочный файл Mayak.exe.

2) В открывшемся диалоговом окне «Лицензионное Соглашение», пример которого приведен на рисунке 1, установить маркер в поле «Я принимаю условия соглашения» и нажать на кнопку «Далее».

пжить. иять условия эт почаемым межд ю данного	0000
пжить. иять условия эт кочаемым межд ю данного	( ого
іять условия эт кочаемым межд ю данного	000
ючаемым межд ю данного	w
0)), виде.	1
анмы в любых	
но особо.	
а эмпонентам ПО ачи внаем или	
ли иное не	~
	имы в любых но особо. мпонентам ПО, ачи внаем или ли иное не

Рисунок 1 – Окно «Лицензионное соглашение»

3) В открывшемся диалоговом окне, пример которого приведен на рисунке 2, указать каталог, в который будет выполнена установка программы, и нажать на кнопку «Далее».

		-	4
ыбор папки установки			5
В какую папку вы хотите установить МАЯК?			6
Програнна установит МАЯК в следующую папку.			
Нажните «Далее», чтобы продолжить. Если вы хотите выбрат	ь другую папку, н	ажните «	Обзор».
C:\Users\TEST\MAYAK +		06	бзор
Требуется как минимум 1,52 Гб свободного дискового простра	ютва,		

Рисунок 2 – Выбор папки установки

4) На следующем шаге установки необходимо задать название папки в меню «Пуск», в которой будет создан ярлык программы. Пример диалогового окна выбора папки в меню «Пуск» приведен на рисунке 3.

🖌 Установка — МАЯК, версия 1.03	-		×
Выберите папку в меню «Пуск»			FI
т де программа установки должна создать ярлыки:			(10)
в — Программа создаст ярлыки в следующей папке меню «Пуск».			
Нажните «Далее», чтобы продолжить. Если вы хотите выбрать другую папку, н	вжните	«Обзор»	•
МАЯК		Обзор	
Пне создавать папку в меню «Пуск»			
Назад Дал	ee	От	мена

Рисунок 3 – Указание названия папки в меню «Пуск»

5) Нажать на кнопку «Далее», в результате чего откроется диалоговое окно, информирующее о готовности начать установку программы, как это показано на рисунке 4.

			T
е готово к установке			۶
програнна установки потова начать установку милк на ваш контвютер.			6
Нажните «Установить», чтобы продолжить, или «Назад», если вы хотите прос изменить опции установки.	мотреть или	•	
Папка установки: C:\Users\TEST\MAYAK			^
Папка в меню «Пуск»: МАЯК			
Дополнительные задачи: Дополнительные значки: Создать значок на Рабочем столе			
			~

Рисунок 4 – Диалоговое окно, информирующее о готовности начать установку программы

6) Нажатием на кнопку «Установить» инициировать процесс установки программы.

7) Появление диалогового окна, пример которого приведен на рисунке 5, свидетельствует об успешном завершении установки программы.



Рисунок 5 – Окно, информирующее об успешном завершении установки программы

8) Нажатием на кнопку «Завершить» закрыть диалоговое окно.

# 3.1.2 Запуск

Запуск программы осуществляется щелчком левой кнопки мыши по ярлыку **м**ачак, расположенному в одноименной папке меню «Пуск».

При первом запуске программы в автоматическом режиме будет выполнена настройка виртуальной среды Python, а также установка дополнительных зависимостей, необходимых для работы программы.

В результате, будет загружено рабочее окно программы, приведенное на рисунке 6.



Рисунок 6 – Рабочее окно программы

## 3.1.3 Завершение работы с программой

Завершение работы с программой выполняется одним из двух способов:

1) Щелчком по пункту меню программы «Файл» раскрыть список элементов, в котором выбрать «Выход».

2) Нажать на кнопку ⊠ «Закрыть», расположенную в правом верхнем углу окна программы.

## 3.2 Меню программы

Весь доступный пользователю инструментарий сгруппирован и распределен по следующим пунктам Главного меню программы:

- «Файл»;
- «Редактировать»;
- «Вид»;
- «Помощь».

#### 3.2.1 Пункт меню «Файл»

Пункт Главного меню «Файл» предназначен для работы с проектом. Нажатие на пункт меню «Файл» вызывает открытие списка элементов, как это показано на рисунке 7.

🔺 MA	ЯК - project				
<u>Ф</u> айл	<u>Редактировать</u>	Вид	Помощь		
н	овый		CTRL+N		
0	ткрыть		CTRL+O		
H	едавние проекты		3	×	
C	охранить		CTRL+S		
C	охранить как		CTRL+SHIFT+S		
3	акрыть вкладку		CTRL+W		
3	акрыть проект		CTRL+SHIFT+W		
н	астройки страниць	ы	CTRL+ALT+P		
П	росмотр		CTRL+SHIFT+P		
П	ечать		CTRL+P		
В	ыход		CTRL+Q		

Рисунок 7 – Элементы пункта меню «Файл»

Назначение элементов пункта меню «Файл»:

- «Новый» – создание нового проекта;

- «Открыть» – открытие существующего проекта;

 – «Недавние проекты» – используется для быстрого доступа к списку десяти последних, недавно редактированных проектов и быстрого открытия одного из них;

- «Сохранить» – сохранение текущего проекта;

– «Сохранить как» – сохранение текущего проекта в папку отличную от той, из которой он был загружен;

- «Закрыть вкладку» – закрытие активной вкладки открытого проекта;

- «Закрыть проект» – закрыть текущий, открытый проект;

– «Настройки страницы» – настройка параметров страницы для печати на принтере активной программы, представленной в виде диаграммы;

– «Просмотр» – предварительный просмотр результата перед печатью на принтере активной программы;

– «Печать» – печать на принтере активной программы;

- «Выход» – закрытие текущего проекта и выход из программы.

#### 3.2.2 Пункт меню «Редактировать»

Пункт Главного меню «Редактировать» предназначен для работы с редакторами языков стандарта МЭК 61131-3.

Нажатие на пункт меню «Редактировать» вызывает открытие списка элементов, как это показано на рисунке 8.



Рисунок 8 – Элементы пункта меню «Редактировать»

Назначение элементов пункта меню «Редактировать»:

- «Отмена» – отмена последнего действия в редакторе;

– «Повторить» – повтор отмененного последнего действия в редакторе;

– «Вырезать» – переместить в буфер обмена выделенный(е) элемент(ы) в редакторе;

– «Копировать» – копировать в буфер обмена выделенный(е) элемент(ы) в редакторе;

– «Вставить» – вставить из буфера обмена находящиеся в нем элемент(ы) в указанное место в редакторе;

- «Поиск» – поиск в текущем функциональном блоке;

– «Поиск следующего» – подсветка следующего вхождения строки поиска;

- «Поиск в проекте» – вызов диалога поиска данных в проекте;

– «Добавить элемент» – добавление одного из следующих элемента в текущий проект:

• «Типы данных» – новый тип данных;

• «Функции» – новая функция;

• «Функциональный блок» – новый функциональный блок;

• «Программы» – новая программа;

• «Ресурсы» – новый ресурс;

• «С расширение» – новый модуль для написания программы на языке С.

– «Выделить все» – выделение всех элементов в активной вкладке редактора;

– «Удалить» – удаление программного модуля, выбранного в дереве проекта.

# 3.2.3 Пункт меню «Вид»

Пункт Главного меню «Вид» предназначен для работы с редакторами языков стандарта IEC-61131.

Нажатие на пункт меню «Вид» вызывает открытие списка элементов, как это показано на рисунке 9.

🛦 MAЯK - project			
<u>Ф</u> айл <u>Р</u> едактировать	Вид	Помощь	
i 🗅 🗀 🙉 👼		Обновить	CTRL+R
		Очистить ошибки	CTRL+K
Проект		Приближение	>
💷 🔊 Unnamed		Сменить представление	F12
🛃 ST progran		Полный экран	Shift-F12
		Сбросить представление	

Рисунок 9 – Элементы пункта меню «Вид»

Назначение элементов пункта меню «Вид»:

- «Обновить» – обновление данных и снятие выделения в редакторе;

- «Очистить ошибки» – очистка указателей ошибок в редакторе;

- «Приближение» –выбор масштаба в процентах;

– «Сменить представление» – скрытие всех панели, за исключением рабочего поля;

– «Полный экран» – разворачивание окна программы на весь экран;

– «Сбросить представление» – восстановление расположения панелей до исходного состояния.

## 3.2.4 Пункт меню «Помощь»

Для получения сведений о программе служит элемент «О программе», расположенный в пункте Главного меню «Помощь», как это показано на рисунке 10.



Рисунок 10 – Элемент «О программе» пункта Главного меню «Помощь»

## 3.3 Панель инструментов

Панель инструментов представляет собой набор инструментов для быстрого доступа к наиболее часто используемым средствам среды разработки. Она состоит из нескольких панелей, содержащих элементы в виде кнопок:

- Панель инструментов Главного меню;

– Панель инструментов сборки проекта и установки связи с ПЛК;

– Панель графического представления программы.

Внешний вид панели инструментов Главного меню приведен на рисунке 11.



Рисунок 11 – Панель инструментов Главного меню

Описание функционального назначения элементов панелей инструментов Главного меню приведено в таблице 1.

Таблица 1 – Описание функционального назначения элементов панелей инструментов

Внешний вид кнопки	Название кнопки	Выполняемая функция						
Панель инструментов Главного меню программы								
4	Новый проект	Создать новый проект						
1 m	Открыть проект	Открыть существующий проект						
R.	Сохранить проект	Сохранить текущий проект						
	Сохранить проект как	Сохранить текущий проект в указанную папку						
	Печать	Печать на принтере текущей программы						
<u>ج</u>	Отменить	Отмена последнего действия в среде разработки						
<i>i</i>	Повторить	Повтор ранее отмененного действия в среде разработки						
×	Вырезать	Переместить в буфер обмена выделенные элементы в редакторе						
	Копировать	Копировать в буфер обмена выделенные элементы в редакторе						
	Вставить	Вставить из буфера обмена находящиеся там элементы в редактор						
a	Поиск в проекте	Вызов диалога поиска данных в проект						
20	Развернуть/свернуть	Развернуть окно на полный экран/свернуть						
Пан	- окно ель инструментов сборк	и проекта и установки связи с ПЛК						
x	Запустить симуляцию	Запустить программу в режиме симуляции ПЛК						
4	Сгенерировать ST файл	Создание программы для среды исполнения						
Инстру	менты меню панели гра	афического представления программы						
₹2	Выберите объект	Выбор объекта в области редактирования графических элементов программы						
௹	Переместить отображение	Перемещение в области редактирования графических элементов программы						
	Создать новый комментарий	Создание комментария в области редактирования графических элементов программы						
VAR-	Создать новую переменную	Добавление новой переменной в область редактирования графических элементов программы						
<b>FB</b>	Создать новый блок	Добавление нового блока в область редактирования графических элементов						

Внешний вид кнопки	Название кнопки	Выполняемая функция
		программы
23	Создать новое подключение	Добавление нового подключения в область редактирования графических элементов программы

#### 3.4 Создание нового проекта

Описание создания нового проекта проиллюстрировано на примере создания программы.

В качестве примера будет разработана новая программа на языке FBD. Разрабатываемая программа будет являться счетчиком, увеличивающим значение выхода на единицу до тех пор, пока на входе «RES» не будет установлено значение *True*. Инкрементация значения будет происходить в каждом цикле основной программы. Регулировать интервал цикла можно будет изменяя длительность задачи для экземпляра основной программы в панели ресурсов.

Также будет добавлены функциональные блоки счетчиков, написанных на языках FBD, LD и ST.

#### 3.4.1 Логика и алгоритм работы программного блока

Логика и алгоритм работы данного программного модуля следующие:

– определена переменная RES типа BOOL, отвечающая за включение/выключение каждого из трех счетчиков;

– определены три переменные OUT1...OUT3 типа INT, в которых хранится значение каждого из трех счетчиков;

– добавлены три функциональных блока, представляющих собой инкрементирующий счетчик на языках FBD, LD и ST;

– при запуске программы начальное значение переменной RES устанавливается по умолчанию *True*;

– значения счетчиков начнут увеличиваться, начиная с 0, когда переменная RES примет значение *False*;

– для обнуления счетчиков переменную RES необходимо форсировать значением *True*;

– переменным OUT1..OUT3 будет присвоено начальное значение конфигурационной константы ZERO, таким образом значения счетчиков обнулятся.

#### 3.4.2 Создание нового проекта

Создание нового проекта выполняется следующим образом:

1) Щелчком по кнопке 🤷 «Новый проект» инициировать процедуру создания нового проекта.

2) В открывшемся диалоговом окне, пример которого приведен на рисунке 12, указать папку, в котором будут храниться файлы создаваемого

🛦 Выберите пустой ка	аталог д	ля проекта			×
← → • ↑ 📙 >	TEST	> OpenPLC_Editor >	∨ ₫ Поис	к в: OpenPLC_Editor	P
Упорядочить 🔻	Новая г	папка			- ?
📌 Быстрый доступ		Имя	Дата изме	енения Тиг	n
Рабочий стол	*	editor matiec	11.07.2024 11.07.2024	13:17 Па 12:54 Па	пка с файл пка с файл
Загрузки 🚊 Документы	Я Я	ningw	11.07.2024	12:57 Па 12:58 Па	пка с файла пка с файла
📧 Изображения	*	Project	12.07.2024	10:25 Па	пка с файл
<ul> <li>Видео</li> <li>Музыка</li> <li>OneDrive</li> <li>Этот компьютер</li> </ul>		python	11.07.2024	13:16 Na	пка с файла
🚅 Сеть	<				1
г	Тапка:	Project			
			Выбор	о папки От	мена

проекта. В приводимом примере – это папка «Project».

Рисунок 12 – Диалоговое окно выбора папки для проекта

3) Нажать на кнопку «Выбор папки», в результате чего диалоговое окно закроется и появится диалоговое окно «Создать новый POU», пример которого приведен на рисунке 13.

Примечание – Здесь под POU (Program Organization Unit) следует понимать компоненты, образующие код прикладного программного обеспечения ПЛК, имеющие собственное наименование, интерфейс и описание на одном из языков программирования.

Создать новы	ый POU X
Имя POU:	program0
Тип POU:	программа 🗸 🗸
Язык:	~
OK	Cancel

Рисунок 13 – Диалоговое окно «Создать новый POU»

4) В поле «Тип POU» задать тип программного модуля, а именно: «программа». При этом, имя программному модулю будет присвоено автоматически, которое в дальнейшем может быть изменено. В последующем, в проект можно добавить дополнительные программные модули, функции и функциональные блоки.

5) Щелчком по полю «Язык» раскрыть список доступных языков программирования, в котором выбрать один из языков стандарта IEC 61131-3 (IL, ST, LD, FBD, SFC), на котором будут реализованы алгоритмы и логика работы данного добавляемого программного модуля. Выпадающий список доступных для выбора языков программирования приведен на рисунке 14.



Рисунок 14 – Список доступных для выбора языков

6) Щелчком по кнопке «ОК» завершить создание программного модуля: в проект будет добавлен основной программный модуль с выбранными параметрами, ресурс проекта будет конфигурирован по умолчанию: добавлена одна задача циклического выполнения с интервалом 20 мс, и один экземпляр основной программы.

7) Щелчком левой кнопки мыши под деревом проекта вызвать контекстное меню, в списке элементов которого выбрать «Функциональный блок», а затем в поле «Язык» диалогового окна «Создать новый POU» выбрать «FBD» и нажать кнопку «OK».

В качестве альтернативы для вызова контекстного меню может использоваться щелчок левой кнопки мыши по кнопке , см. рисунок 15.

8) Аналогичным образом добавить функциональные блоки счетчиков, написанных на языках LD и ST. В результате, панель проекта примет вид, пример которого приведен на рисунке 15.

MARK - project-	1 1	1	1	, ,	1 2					
galar       Peaceraporans       But Chances         Poer       Image: Chances       Image: Chances         Image: Chances       Image: Chances       Image: Chances </td <td>🛦 MAЯK - ~project~</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td>- 0</td> <td>×</td>	🛦 MAЯK - ~project~								- 0	×
Image: Section of the section of th	<u>Ф</u> айл <u>Р</u> едактировать <u>В</u>	ид <u>П</u> омощь								
Pipers       Indicative Granue         Image: Comparison of the comparis	🗋 🚨 🚨 🗐	3 🦛 🌧	🔏 🖷 💼	a 🔣 🗡 🖣	, 🗿 🐐 📗	2				
Onacasuse       Outroamed       Outroamed       Outroamed         Image: Comparison of productions/survey for access of productions/survey for acce	Проект			1 🛃 🔊 program0	ST functionBlo	ck0 × gp	unctionBlock1	functionBlock2 🕨 🗸	Библиотеки Отладчик	
Image: Construction of the second	Unnamed			Описание:			Фильтр класса:	Bce	Q. Поиск	
	Constant Program0     Res0	ные блоки	4	# Имя < 1	Класс	Тип	Исходное значение	• Квалификатор	G. Стандартные функциональные блоки     Дополнительные функциональные блоки     Дополнительные функциональные блоки     Дополнительные функциональные блоки     Gommunication     PIAM Modules     Media     Modules     Jaguar     Gopasoeanne типоe     Marewaruweckne функции     Marewaruka     Speana     Gauroseie операции     Suroseie операции     Buroseie операции     Suroseie операции	^
Q+ Поиск         В	Ŧ		~ ~~						th Consequence	* ^
Q+ Поиск         Консоль         Лог ПЛК				¢				>		~
				Поиск Консоль	lor D/K					
	Or Dourr		0	TRACK RONCOND 1						
	S. HOREK		w.							

Рисунок 15 – Окно проекта

#### 3.4.3 Добавление конфигурационных переменных проекта

Панель списка переменных и констант, пример которой приведен на рисунке 16, представляет собой таблицу, соответствующую выбранному программному модулю, ресурсу или, в целом, проекту.

Конфигурационные переменные позволяют программным модулям типа «Программа» и «Функциональный блок» использовать общие переменные, которые будут определены в глобальной области видимости проекта.



Рисунок 16 – Панель списка переменных и констант

### 3.4.3.1 Добавление конфигурационных констант и переменных

Добавление конфигурационных констант и переменных выполняется следующим образом:

1) Двойным щелчком по корневому элементу дерева проекта открыть Панель списка переменных и констант.

2) Щелчком по кнопке 📌 «Добавить переменную» добавить конфигурационную константу «ZERO» типа INT и начальным значением 0.

3) Аналогичным образом добавить переменную RES типа BOOL, отвечающую за включение/выключение каждого из трех счетчиков, а также три переменные OUT1, OUT2 и OUT3 типа INT. На рисунке ниже показан результат объявления конфигурационных переменных.

1 0	1 71		1			
A MAЯK ~project~						– 🗆 ×
<u>Ф</u> айл <u>Р</u> едактировать <u>В</u> ид <u>П</u> омощь						
🕞 🖴 🖉 🦉 🔒   🍝 🗢   🔏 🛍	a 🚼 🛪 🦊 🧿	*				
Проект	Program0 ST funct	ionBlock0 🖉 🗗 fu	nctionBlock1	₽ functionBlock2 💒 ~	Проект~ Х 🗸	Библиотеки Отладчик
= <mark>е<sup>®</sup> Unnamed</mark> ⊞ ∰ Функциональные блоки	МЭК-над Файлан провита					Q. Поиск 
	Конфигурационные перем	нные Свойства пр	оекта Конфигур	ация		<ul> <li>Дополнительные функциональна</li> </ul>
Res0	Фильтр класса: Все	~		4	= 🕈 🕹	Arduino     Communication
	Имя Класс	Тип	Адрес	Исходное значение	Квалификатор	PIAM Modules     PAQTT
	ZERO Глобальны	ă INT		0	Константа	Sequent Microsystems Modules
	RES Глобальны	BOOL				— Преобразование типов
	ОUT1 Глобальны	ă INT				Э-Математические функции
	ОUT2 Глобальны	ă INT				— Математика
	ОUT3 Глобальны	i INT				е-Время
						— Сдвиговые операции
						Ф. Выбол
T Config0						< >>
🖳 Res0 ()						
💽 ZERO (INT) 🗸	<				>	×
💌 RES (BOOL) 💞	Поиск Консоль Лог ПЛК				ANTIAROL	una Windows
					Чтобы акт	ивиповать Windows, перейдите в
C HONCK					раздел "Па	араметры".

Рисунок 17 – Результат объявления конфигурационных переменных

4) Для того чтобы к данным конфигурационным переменным можно было обращаться из программных модулей типа «Программа» или «Функциональный блок» необходимо для них в их панелях переменных и констант создать переменные с таким же именем, как и ранее объявленные глобальные, и установить их класс «Внешний» (External).

Каждая переменная имеет следующие параметры:

– «Имя» – представляет собой уникальный идентификатор переменной в пределах ее области видимости и действия;

– «Класс» – «Глобальный», «Вход», «Выход», «Вход/Выход», «Локальный», «Внешний», «Временный»;

– «Тип» – определяет тип переменной и может принадлежать базовому типу (в соответствии со стандартом IEC 61131-3), пользовательскому типу (псевдониму и поддиапазону существующего типа, перечислению, массиву, структуре) или типу функционального блока (стандартному или пользовательскому);

– «Адрес» – идентификатор, необходимый для связывания данной переменной с переменной плагина модуля (перечень и описание адресов входов/выходов ПЛК приведен в приложении А);

– «Исходное значение» – инициализация переменной некоторым начальным значением;

– «Квалификатор» – задание константности, реманентности (сохранение ее значения в энергонезависимой памяти) и нереманентности переменной;

– «Описание» – комментарий к назначению данной переменной или константы.

Первый символ имени переменной или константы должен быть буквой, или символом подчеркивания, за которым могут следовать цифры, буквы латинского алфавита и символы подчеркивания. Набор возможных вариантов классов переменных зависит от типа элемента проекта, редактирования которого осуществляется.

Панель переменных и констант предоставляет возможность фильтровать отображаемые переменные по их конкретным классам («Вход», «Выход», «Вход/Выход», «Внешний», «Локальный», «Временный») или сгруппированным классам («Интерфейс» и «Переменные»). Данная операция выполняется с помощью функции «Фильтр класса», перечень элементов которого приведен на рисунке 18.



Рисунок 18 – Перечень элементов фильтра класса

# 3.4.3.2 Функциональное назначение кнопок, расположенных на панели переменных и констант

Функциональное назначение кнопок, расположенных на панели переменных и констант приведено в таблице 2.

Внешний вид кнопки	Название кнопки	Функция кнопки
-	Добавить переменную	Добавить новую переменную в панель переменных и констант со значениями по умолчанию
	Удалить переменную	Удалить выделенную переменную или константу
1	Переместить переменную вверх	Перемещение переменной в таблице переменных и констант вверх на одну позицию
÷	Переместить переменную вниз	Перемещение переменной в таблице переменных и констант вниз на одну позицию

Таблица 2 – Функциональное назначение кнопок на панели переменных и констант

#### 3.4.4 Написание алгоритма и логики выполнения программы

Для написания алгоритма и логики выполнения данной программы будут использованы функциональные блоки счетчиков.

Для удобства редактирования FBD-диаграмм в редакторе реализована функция Drag&Drop, используя которую необходимые функциональные блоки и переменные могут быть добавлены путем перетаскивания в поле редактирования непосредственно из таблицы переменных. Для этого необходимо левой клавишей мыши зажать номер соответствующей переменной в столбце «#» панели переменных и констант, перенести указатель в область редактирования FBD диаграммы, а затем отпустить кнопку мыши.

#### 3.4.5 Пример создания функционального блока на языке FBD

Пример создания функционального блока с именем «FBD», в котором инструментами языка FBD реализован счетчик, принимающий на вход переменную RES типа BOOL, и возвращающий значение счетчика OUT, приведен на рисунке 19.

<u>Файл Редактировать Вид П</u> омощь									
💫 🖴 💆 🛎 🙁 🛠 📣 🕺 🖷 🛍 🔍 💦 🧪 🖌 🗳 🖍 🕐 📾 📟 🕆 🖙									
記録 program0 記録 FBD × 記号 FB_LD									
Project	-	Description Clare Filter All							
B Vinnamed	Descr	iption:	Class Filte				-		
	#	Name	Class	Туре	Initial Value	Option			
FB_LD	1	RES	Input	BOOL			Сброс счетчика		
🔁 🌮 program0	2	COUNT	Local	INT			Значение счетчика		
nesource1	3	ZERO	External	INT		Constant	Начальное значение после сброса		
	4	OUT	Output	INT			Возвращаемое значение ФБ		
Config.resource1.instance0.FBD0 Correct Count (INT) Correct Count		RES ADD IN1 OUT IN2 ZERO	SEL OUT NO N1	TUD					

Рисунок 19 – Описание пользовательского функционального блока на языке FBD

Для написания алгоритма и логики выполнения данной программы будут добавлены две функции: «ADD» и «SEL».

Функция «ADD» находится во вкладке «Математика» в Библиотеке функций и функциональных блоков. Функция означает сложение входных значений (в нашем примере их 2) на входах «IN1» и «IN2», возвращает результат вычисления на выход «OUT».

Функция «SEL» означает «Выбор одного из двух значений» и находится во вкладке «Выбор». Она содержит три входных переменных «G», «IN0», «IN1» и одну выходную «OUT». Если «G» равно 0 (или FALSE), то выходной переменной «OUT» присваивается значение «IN0». Если «G» равно 1 (или TRUE), то выходной переменной «OUT» присваивается значение «IN1».

Добавление данных функций также можно осуществить путем переноса соответствующей функции с помощью мыши (Drag&Drop) из панели Библиотеки функций и функциональных блоков в область редактирования FBD диаграммы функционального блока.

Соединение блоков осуществляется нажатием левой кнопки мыши на коннекторе блока и последующим соединением его с коннектором присоединяемого блока.

Функциональный блок становится доступным в панели библиотеки функций и функциональных блоков «User-defined POUs» и может в дальнейшем использоваться в программных модулях типа «Программа» и «Функциональный блок».

П р и м е ч а н и е – Наименование функционального блока не должно совпадать с существующими в библиотеке.

## 3.4.6 Пример создания функционального блока на языке LD

Пример создания функционального блока с именем «FBD», в котором инструментами языка LD будет реализован счетчик, принимающий на вход переменную RES типа BOOL и возвращающий значение счетчика OUT.

Создание функционального блока выполняется следующим образом:

1) Добавить в панель переменных и констант возвращаемое значение «OUT» типа INT и класса «Выход», локальную переменную «COUNT» типа INT, внешнюю конфигурационную переменную «ZERO» типа INT и входную переменную «RES» типа BOOL.

2) Добавить шину питания, к которой присоединить контакт, связанный с переменной «RES», как это показано на рисунке 20.

Edit Contact Values	×
Modifier: Normal Negated Rising Edge Falling Edge	Preview:
Variable:	1 1
ОК	Cancel

Рисунок 20 – Диалоговое окно добавления контакта

Полученная конструкция будет подавать сигнал на сброс счетчика при переходе значения переменной «RES» в *True*, как это показано на рисунке 21.

Рисунок 21 – Контакт, ассоциированный с переменной RES

3) Добавить числовой литерал со значением «1», для чего щелчком правой кнопки мыши вызвать появление контекстного меню, в списке элементов которого выбрать «Добавить», а затем – «Переменная». В результате, будет открыто диалоговое окно создания переменной, пример которого приведен на рисунке 22.

4) В поле «Выражение» ввести «1». Таким способом задается шаг инкрементации счетчика.

5) Нажать на кнопку «ОК».

Свойства переменных	×
Класс: Вход ✓ Порядок исполнения: 0 💌	Выражение: 1 COUNT OUT1 OUT3 RES ZERO
Просмотр:	1
1	K Cancel

Рисунок 22 – Диалоговое окно добавления переменной

6) Для написания алгоритма и логики выполнения данной программы добавить две функции: «ADD» и «SEL» из панели Библиотеки функций и функциональных блоков в область редактирования LD диаграммы функционального блока.

На рисунке 23 приведен пример реализации функционального блока на языке LD.

<u>Ф</u> айл <u>Р</u> едактировать <u>В</u> ид <u>П</u> омощь										
🕞 🚔 🖉 🚇 ( 🏎 / »   🔏 👘   🗟 👯 🗡 🖕 🕓 📾 🔤 📅 📴										
Project	to program0 (2) ≥ FBD (2) pr fB_LD ×									
P Unnamed	Desc	iption:	Class F	ilter: All 🗸						
FBD     FBD	#	Name	Class	Туре	Initial Value	Option				
FB_LD	1	RES	Input	BOOL						
🛃 🍉 program0	2	COUNT	Local	INT						
🕎 resource1	3	ZERO	External	INT		Constant				
	4	OUT	Output	INT						
Image: configuresource1.instance0.FB_LD0     Image: configuresource1.instance0.FB_LD0     Image: configuresource1.instance0.FB_LD0       Image: configuresource1.instance0.FB_LD0     Image: configuresource1.inst		RES ADD I IN1 OUT IN1 OUT IN1 ZERO	UT COUNT							
ERO (INT)										

Рисунок 23 – Пример реализации функционального блока на языке LD

Функциональный блок становится доступным в панели библиотеки функций и функциональных блоков «User-defined POUs» и может использоваться в программных модулях типа «Программа» и «Функциональный блок».

П р и м е ч а н и е – Наименование функционального блока не должно совпадать с существующими в библиотеке.

На рисунке 24 показан пример использования созданного функционального блока «FB\_LD» в основном программном модуле, написанном на языке FBD.

<u>Файл</u> <u>Редактировать</u> <u>Вид</u> <u>П</u> омощь										
📴 🖳 🖳 🖳 🐟 🧀 🕺 🖶 🛍 🔤 💱 🥻 🤞 🕒 📾 📼 🕧 🖾										
	ter program0 × 12 pr F8_LD									
Project	Desci	ription:	CI	ss Filter: All 🗸						
Unnamed     Function Blocks	L .	Name	Class	Tures	Location	Initial Value	Ontion			
PBD FBD	L.	Name	Ciass	Туре	Location	initial value	Option			
FB_LD		RES	External	BOOL				Сброс счетчиков		
z Programu	2	0011	External	IN I				Значения счетчка FBD		
	L÷.	0012	External	INT				Значения счетчка LD		
	-	6013	External	ERD ERD				эначения счетчка эт		
	6	FB LDO	Local	FBLD						
Config.resource1.instance0 ✓ ↔     Config.resource1.instance0 ✓ ↔     RES (BOOL) ↔     OUT1 (INT) ↔     OUT2 (INT) ↔     OUT2 (INT) ↔     OUT3 (INT) ↔     FED0 (FBD)		RES FBD RES FB_LDO FB_LDO FB_LDO RES FB_LDO OUT								

Рисунок 24 – Пример использования созданного функционального блока «FB\_LD» в основном программном модуле, написанном на языке FBD

# 3.4.7 Пример создания функционального блока на языке ST

Пример создания функционального блока с именем «FB\_ST», в котором инструментами языка ST будет реализован счетчик, принимающий на вход переменную RES типа BOOL, и возвращающий значение счетчика OUT.

На рисунке 25 приведен пример реализации функционального блока на языке ST.



Рисунок 25 – Описание пользовательского функционального блока на языке ST

Функциональный блок становится доступным в панели библиотеки функций и функциональных блоков «User-defined POUs» и может

использоваться в программных модулях типа «Программа» и «Функциональный блок».

П р и м е ч а н и е – Наименование функционального блока не должно совпадать с существующими в библиотеке.

Пример использования функционального блока на языке ST в основном программном модуле приведен на рисунке 26.



Рисунок 26 – Пример использования функционального блока на языке ST в основном программном модуле

## **3.4.8** Pecypc

Согласно стандарту IEC 61131-3, каждый проект должен иметь как минимум один ресурс, с определенным в нем как минимум одним экземпляром.

Экземпляр представляет собой элемент, связанный с программным модулем типа «Программа» и одной определенной задачей. По умолчанию, инструментальная среда разработки создает для нового проекта один ресурс.

Глобальные переменные ресурса объявляются аналогично глобальным переменным проекта на панели переменных и констант выбранного ресурса с использованием кнопки «Добавить переменную» либо «Добавить переменные».

Пример объявления в проекте глобальной переменной приведен на рисунке 27.

<b>B</b> S	program0 🖳	Config0.Res0 ×	🛃 STs0.instar	nce0				$\sim$
Фильтр класса: Все 🗸							- 🔶 🗕 🔶	1
#	Имя	Класс	Тип	Адрес	Исходное значение	Квалификатор	Описание	
1	LocalVar0	Глобальный	INT		10			
2	GlobalFlag	Глобальный	BOOL		True			

Рисунок 27 – Пример объявления в проекте глобальной переменной

Использование данных глобальных переменных на уровне ресурса также аналогично использованию конфигурационных переменных проекта в программных модулях.

Для использования в программном модуле глобальной переменной ресурса необходимо добавить в модуль переменную класса «Внешняя» (External) с таким же именем, как у глобальной переменной, объявленной для ресурса.

Для создания экземпляра необходимо наличие как минимум одного программного модуля типа «Программа» в проекте и как минимум одной задачи, определенной в панели редактирования ресурса.

После добавления задачи с помощью кнопки «Добавить» (данная кнопка аналогична кнопке «Добавить» на панели переменных и констант), необходимо задать ее уникальное имя (поле «Имя») и выбрать тип выполнения задачи (поле «Запуск»):

– «Циклический» — выполнение программного модуля типа «Программа» через заданный интервал времени, указанный в поле «Интервал»;

– «Прерывание» — выполнение программного модуля типа «Программа» один раз при наступлении значения *TRUE* глобальной переменной типа BOOL, определенной на уровне проекта, либо на уровне ресурса, указанной в поле «Single».

Выбор типа выполнения задачи из списка возможных значений приведен на рисунке 28.

Задачи:					-	ſ	↓
Имя	Запуск	Источник	Интервал	Приоритет			
task0	Циклический 🗸		T#20ms	0			
task1	Прерывание Циклический		T#20ms	0			

Рисунок 28 – Выбор типа выполнения задачи

В случае выбора типа выполнения «Циклический», в поле «Интервал» необходимо задать интервал, с которым будет выполняться данная задача. Двойной щелчок левой кнопкой мыши по полю «Интервал» приводит к появлению кнопки «...», нажатие на которую вызывает появление диалогового окна «Редактировать длительность», в котором можно указать продолжительность задачи, используя для этого микросекунды, миллисекунды, секунды, минуты, часы и дни.

Пример диалогового окна «Редактировать длительность» приведен на рисунке 29.

Редактировать длитель	ность					×
Дни: 0	Часы: 0	Минуты: 0	Секунды: 0	Миллисекунды: 20 ОК	Микросекунды: 0.000 Cancel	

Рисунок 29 – Диалоговое окно редактирования длительности задачи

Завершение ввода времени кнопкой «ОК» приводит к закрытию диалога и добавлению заданного интервала времени в поле «Интервал» добавляемой задачи.

В случае выбора типа выполнения «Прерывание» в поле «Источник» необходимо указать переменную типа BOOL, определенную глобально либо на уровне проекта, либо на уровне ресурса. На рисунке 30 выбрана переменная «GlobalFlag», определенная в данном ресурсе.

Задачи:					-	ſ	<b>ا</b>
Имя	Запуск	Источник	Интервал	Приоритет			
task0	Прерывание	GlobalFlag		0			
task1	Циклический		T#20ms	0			

Рисунок 30 – Выбор переменной типа BOOL как источника прерывания для начала выполнения задачи

Задача будет выполнена один раз, как только значение переменной, заданной в этом поле, будет *TRUE*.

Поле «Приоритет» позволяет задать приоритет выполнения задачи. По умолчанию, все задачи имеют приоритет 0.

Следует учесть, что в ресурсе должна быть определена, как минимум, одна задача с типом выполнения «Циклический». В противном случае будет ошибка компиляции в отладочной консоли.

После того как задачи определены, их можно использовать в экземплярах.

Создание экземпляра выполняется аналогичным образом с помощью кнопки «Добавить». Необходимо выбрать уникальное имя экземпляра и далее указать программный модуль типа «Программа» в поле «Тип» и одну из задач в поле «Задача».

В каждом проекте в ресурсе должен быть определен, как минимум, один экземпляр. В противном случае, в отладочной консоли будет выдана ошибка компиляции.

# 3.5 Сборка (компиляция и компоновка) и передача полученного исполняемого файла на целевое устройство

Сборка проекта осуществляется с помощью соответствующих кнопок, находящихся на панели инструментов. Для успешного завершения данной операции каждый проект должен иметь как минимум один ресурс, в котором должна быть определена, как минимум, одна задача циклического типа и один экземпляр. Соответственно, проект обязан содержать, как минимум, один программный модуль типа «Программа», причем тело, т.е. алгоритм и логика его выполнения, не может быть пустым (в противном случае будет ошибка компиляции).

#### 3.5.1 Сборка проекта в режиме симуляции

Для сборки проекта в режиме симуляции необходимо нажать кнопку 🗡 «Запустить симуляцию».

Результаты сборки выводятся в консоль, расположенную в нижней части окна программы. При наличии ошибки она выделяется красным шрифтом. В примере, приведенном на рисунке 31, после сборки в консоль выведено сообщение о том, что сборка прошла с ошибками (текст ошибки на желтом фоне).



Рисунок 31 – Результаты сборки в режиме симуляции выведены в консоль

# 3.5.2 Генерация программы для последующей загрузки в программируемый логических контроллер

Формирование файла программы выполняется следующим образом:

1) Нажать на кнопку 🦊 «Сгенерировать ST файл».

2) По итогу завершения процесса создания файла программы откроется диалоговое окно, в котором необходимо задать имя файла и указать место его сохранения. Результат загрузки выводится в консоль, как это показано на рисунке 32.

Поиск	Консоль	Jor IIJK	
	Collection	ng data types	^
	Collecti	ng POUs	
	Generate	POU program0	-
	Generate	Config(s)	
Коза	поляция М	ЭК-программы в С-код	
Экст	торт лока	льных переменных	
C-mo	од успеши	о стемерирован.	
плк			
	[CC] plc	main.c -> plc_main.o	
	[CC] plc	_debugger.c -> plc_debugger.o	
плк	:		
	[CC] Con	fig0.c -> Config0.o	
	[CC] Res	0.c -> Res0.o	
Лини	consta:		
	[CC] plc	_main.o plc_debugger.o Config0.o Res0.o -> project.dll	
Сбор	риа прошл	а успешно.	
Oper	nPLC prog	ram generated successfully	
			¥

Рисунок 32 – Результат загрузки проекта в ПЛК

ST-файл, который после компиляции будет загружен в ПЛК, находится в папке проекта с названием *plc.st*. На рисунке 33 показан путь в папке проекта файла и название файла.

Файл Главная По	делиться Вид						~ 0
🕂 🚽 🕆 🔂 > TEST > MAYAK > project					~ Ö	Поиск в: project	Q
	Имя	Дата изменения	Тип	Размер			
🖈 выстрый доступ	build	30:07.2024 16:39	Папка с файлами				
Рабочий стол 🚿	project_files	30.07.2024 12:51	Папка с файлами				
👆 Загрузки 💉	C beremiz	30.07.2024 12:51	Microsoft Edge H	1 KE			
📋 Документы 📝	] plc.st	30.07.2024 12:40	Файл "ST"	1 КБ			
📰 Изображения 💉	C plc	30.07.2024 12:51	Microsoft Edge H	2 KB			

Рисунок 33 – Расположение ST-файла проекта

# 4 СООБЩЕНИЯ ОПЕРАТОРУ

В процессе взаимодействия с программой возможны ситуации, при которых пользователь получает информационные сообщения от ПО «Интегрированная среда разработки «МАЯК».

Сообщения, получаемые пользователем, бывают двух типов:

- информационные;

- диалоговые.

## 4.1 Информационное сообщение

Информационное сообщение – это сообщение, информирующее пользователя о возникновении той или иной проблемы.

На рисунке 34 приведен пример сообщения, полученного от ПО «Интегрированная среда разработки «МАЯК», причиной появления которого являются некорректные действия при создании нового программного блока, а именно: не выбран язык программирования.



Рисунок 34 – Сообщение об ошибке

Для корректного завершения процедуры создания нового программного блока пользователь должен:

1) Щелчком по кнопке «ОК» закрыть окно сообщения об ошибке.

3) Заполнить поле «Язык», выбрав нужный вариант из списка доступных языков программирования.

## 4.2 Диалоговое сообщение

Диалоговое сообщение – это сообщение в виде диалогового окна, появление которого явилось реакцией ПО «Интегрированная среда разработки «МАЯК» на те или иные действия пользователя, которые могут повлиять на работоспособность проекта, а значит необходимо получение от пользователя подтверждения выполняемых действий либо отказ от них.

Примером появления такого рода сообщения, приведенного на рисунке 35, является ситуация, при которой пользователь инициировал процесс удаления программного блока.

Здесь пользователь должен выбрать один из двух предлагаемых вариантов: щелчком по кнопке «Да» подтвердить удаление программного блока либо отказаться от его удаления, нажав на кнопку «Нет».

Удалить POU		
"program0" используется одним ил	и более РОՍ. Прод	олжить?
	Да	<u>Н</u> ет

Рисунок 35 – Диалоговое окно подтверждения удаления программного блока

# Приложение А

# (обязательное)

# Перечень и описание адресов входов/выходов ПЛК

В таблице А.1 приведен перечень адресов входов/выходов ПЛК и их описание.

Таблица А.1 – Перечень и описание адресов входов/выходов ПЛК

Адрес	Описание				
%IX0.0	Дискретный вход 1 (1 – Норма, 0 – Ошибка)				
%IX0.1	Дискретный вход 2 (1 – Норма, 0 – Ошибка)				
%IX0.2	Дискретный вход 3 (1 – Норма, 0 – Ошибка)				
%IX0.3	Дискретный вход 4 (1 – Норма, 0 – Ошибка)				
%IX0.4	Дискретный вход 5 (1 – Норма, 0 – Ошибка)				
%IX0.5	Дискретный вход 6 (1 – Норма, 0 – Ошибка)				
%IX0.6	Дискретный вход 7 (1 – Норма, 0 – Ошибка)				
%IX0.7	Дискретный вход 8 (1 – Норма, 0 – Ошибка)				
%IX1.0	Дискретный вход 1. При 0 (0 – обрыв, 1 – K3)				
%IX1.1	Дискретный вход 2. При 0 (0 – обрыв, 1 – K3)				
%IX1.2	Дискретный вход 3. При 0 (0 – обрыв, 1 – КЗ)				
%IX1.3	Дискретный вход 4. При 0 (0 – обрыв, 1 – K3)				
%IX1.4	Дискретный вход 5. При 0 (0 – обрыв, 1 – K3)				
%IX1.5	Дискретный вход 6. При 0 (0 – обрыв, 1 – K3)				
%IX1.6	Дискретный вход 7. При 0 (0 – обрыв, 1 – K3)				
%IX1.7	Дискретный вход 8. При 0 (0 – обрыв, 1 – K3)				
%QX0.0	Дискретный выход 1				
%QX0.1	Дискретный выход 2				
%QX0.2	Дискретный выход 3				
%QX0.3	Дискретный выход 4				
%QX0.4	Дискретный выход 5				
%IW0	Аналоговый вход 1				
%IW1	Аналоговый вход 2				
%IW2	Аналоговый вход 3				
%IW3	Аналоговый в ход 4				
%IW4	Аналоговый вход 5				
%IW5	Аналоговый вход 6				
%IW6	Аналоговый вход 7				
%IW7	Аналоговый вход 8				
%IW8	Частотный вход (до 23кГц)				
%MW0-					
%MW1023	лдроса в намяти для хранения данных				

# Лист регистрации изменений

	Номера листов (страниц)			Всего				
Изм.	изме-	заме-		аннули-	листов	No	Поля	Пото
	нен-	нен-	новых	рован-	(страниц)	докум.	подп.	дата
	ных	ных		НЫХ	в докум.			