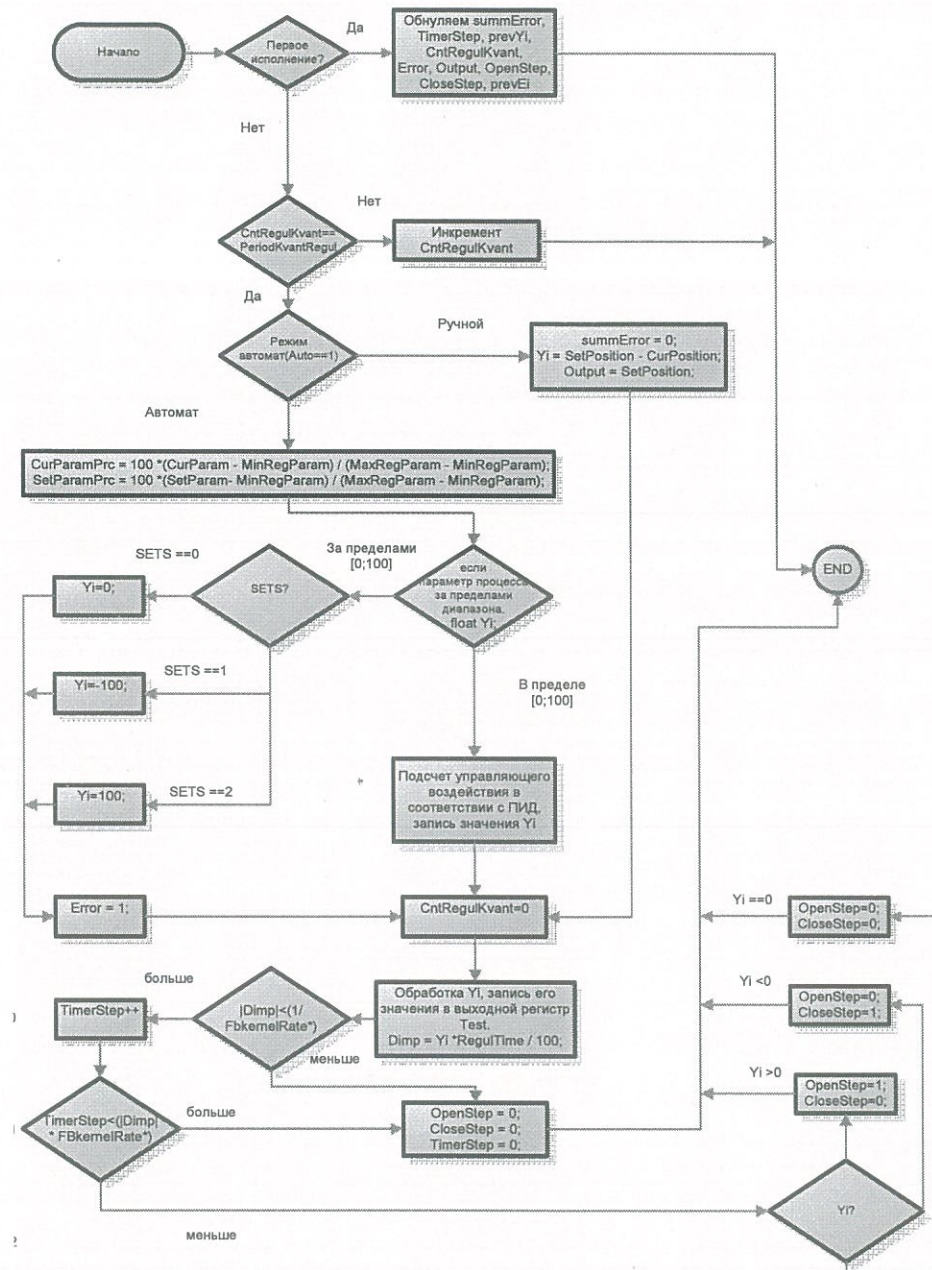


ФБ №82 «ПИД регулятор с задержкой выходного сигнала»

ФБ реализует аналоговое ПИД регулирование. ПИД регулятор формирует импульсы на открытие - закрытие, в зависимости от необходимого положения регулятора, его текущего положения, коэффициентов(kP,kL,kD) и режима работы.

Алгоритм работы ПИД регулятора:



- Enable
- ReverseU
- Rezet
- RequireValue
- CurrentValue
- kP
- kl
- kD
- Position
- GistTube
- DelavTime
- Error
- Output
- Test

Список входов/выходов

Входы	Выход
Команда выбора управления (Auto- bit)	Индикация ошибки входных параметров (Error - bit)
Команда Реверс (ReverseU - bit)	Необходимое положение регулятора в % (Output – uint8)
Регулируемый параметр (CurParam- float)	
Уставка регулирования (SetParam- float)	Импульс на открытие (OpenStep - bit)
Текущее положение регулятора (CurPosition- uint8)	Импульс на закрытие (CloseStep - bit)
Уставка положения регулятора (SetPosition- uint8)	Значение уровня воздействия (Test - float)
Коэффициент пропорциональности (kP- float)	Переменные
Коэффициент времени интегрирования (kI - float)	Накопленная ошибка интегратора (summError - float)
Коэффициент времени интегрирования (kD - float)	Предыдущее значение ошибки регулирования (prevEi – float)
Зона нечувствительности (enTube- float)	Текущее состояние воздействия на регулятор (prevYi - float)
Настройка (Sets- uint8)	Счетчик времени импульса управляющего воздействия (TimerStep – uint16)
Максимальное значение регулируемого параметра (MaxRegParam - float)	Номер кванта времени измерения (CntRegulKvant – uint8)
Минимальное значение регулируемого параметра (MinRegParam - float)	
Время перехода регулирующего органа в состояние от 0 до 100% в сек (RegulTime- uint16)	

Параметры Sets
0 - при значении входного параметра за пределами диапазона измерения - ничего не делать
1 - при значении входного параметра за пределами диапазона измерения - закрыть
2 - при значении входного параметра за пределами диапазона измерения - открыть

ФБ №86 «Резервуар»

ФБ реализует управление 4-мя насосами с выбором основных.

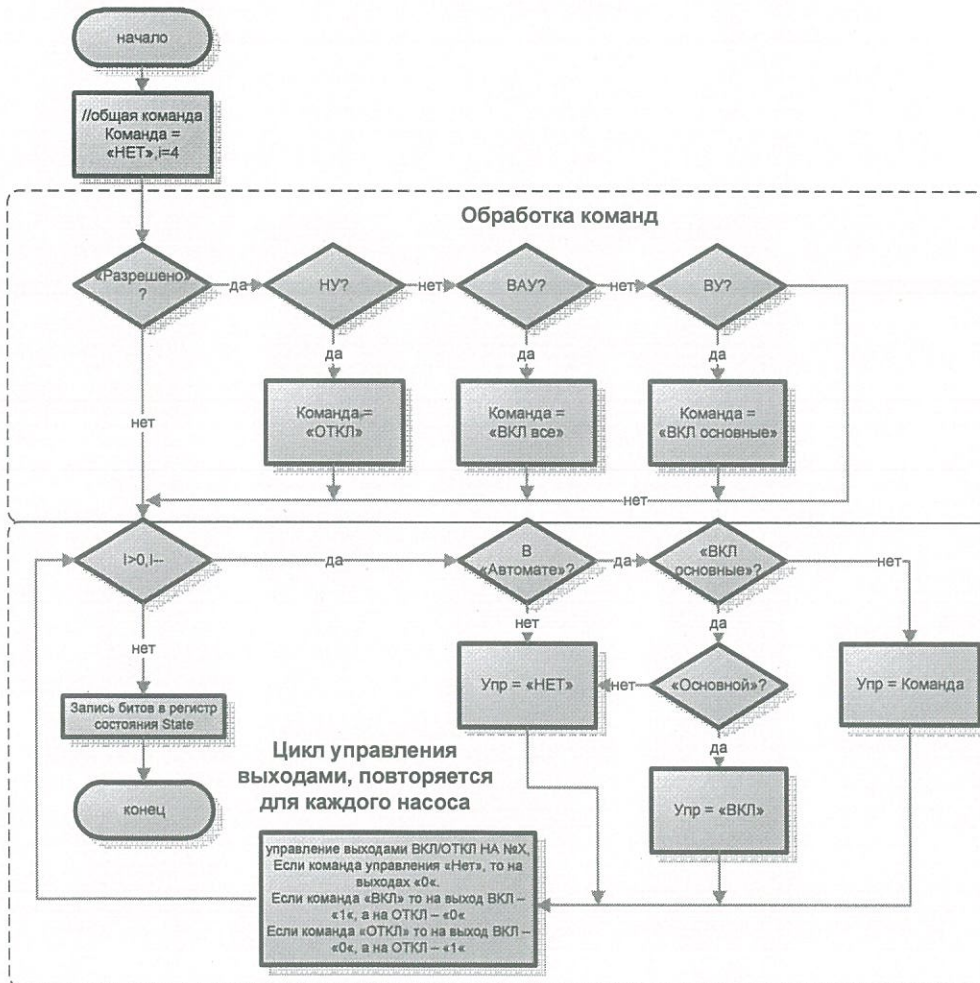
Список входов/выходов

DI	DO
Команда включение (Enable- bit)	ВКЛ НА №1(cmd0_ON)
Команда ВКЛ Все (hihi- bit)	ОТКЛ НА №1(cmd0_OFF)
Команда ВКЛ основные (hi- bit)	ВКЛ НА №2(cmd1_ON)
Команда ОТКЛ(lo- bit)	ОТКЛ НА №2(cmd1_OFF)
1-основной(main0- bit)	ВКЛ НА №3(cmd2_ON)
2-основной(main1- bit)	ОТКЛ НА №3(cmd2_OFF)
3-основной(main2- bit)	ВКЛ НА №4(cmd3_ON)
4-основной(main3- bit)	ОТКЛ НА №4(cmd3_OFF)
Признак управления насосом №1(auto0- bit)	AQ Состояние(State 16-и разрядный)
Признак управления насосом №2(auto1- bit)	
Признак управления насосом №3(auto2- bit)	
Признак управления насосом №4(auto3- bit)	

Флаги состояния ФБ

- «Распиновка» Состояния
- V0 - Разрешено
 - V1 - ВАУ
 - V2 - ВУ
 - V3 - НУ
 - V4 - команда: ВКЛ основные
 - V5 - команда: ВКЛ все
 - V6 - команда: ОТКЛ
 - V7-15 – не используются

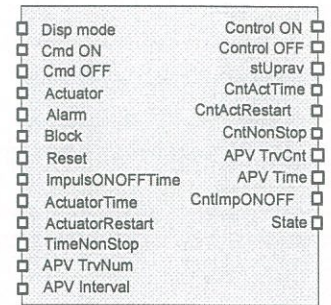
Алгоритм работы ФБ «Резервуар»:



4.12.1 ФБ №90 «Насос с АПВ» (автоматическое повторное включение)

ФБ реализует управление (включение/отключение) двигателем с контролем срабатывания защит и блокировок и автоматическим повторным включением после аварии.

- Вход ФБ «Автомат»(Dist) активизирует логику работы ФБ.
- На входы «Команда ВКЛ» и «Команда ОТКЛ» подаётся «1», после чего инициируется процедура включения/отключения насоса, «Команда ОТКЛ» имеет большую прерогативу чем «Команда ВКЛ».
- Вход «Защита» инициирует процедуру отключения насоса если уровень «1» держится на нём в течение времени задержки защит(CntNonStop) и «пускатель включен или идет процесс включения», при этом устанавливается флаг состояния flAlarm (авария) и управляющие воздействия через входы «Команда ВКЛ» и «Команда ОТКЛ» не действуют пока не пройдет импульс на вход Сброс (ResetAlarm).
- После выключения входного сигнала аварии возможен АПВ только если прошло время для разрешения АПВ (APV_Timer) и установлена команда на включение «Команда ВКЛ».
- Вход «Блокировки» инициирует процедуру отключения насоса немедленно независимо от текущего состояния пускателя. Устанавливаются флаги состояния «Блокировка», управляющие воздействия через входы «Команда ВКЛ» и «Команда ОТКЛ» не действуют, а также запрещается АПВ.
- Сброс флагов блокировки происходит с приходом импульса на вход Сброс (ResetAlarm)

**Процедура включения насоса.**

При получении импульса на вход «Команда ВКЛ», при условии что нет аварий, на выход «Управление ВКЛ» подаётся «1» на время «Время пуска»(CntImpONOFF), после чего на выходе «Управление ВКЛ» возвращается «0». Если в течение «Времени пуска» состояние пускателя не изменилось на «1», т.е. пускатель не включился, процедура повторяется до тех пор, пока есть попытки перезапуска (устанавливается входом «Попыток перезапуска»). Если за указанное количество перезапусков не удалось включить пускатель, на выход «Управление ОТКЛ» подаётся положительный импульс длительностью «Время пуска», устанавливается флаг состояния ActuatorFail_ON.

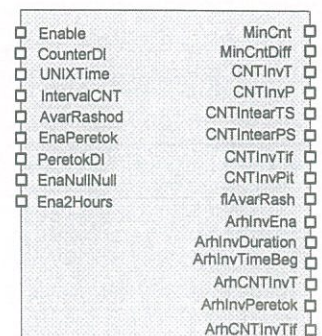
Процедура отключения насоса.

При получении импульса на вход «Команда ОТКЛ» либо при установленном флаге «Авария», на выход «Управление ОТКЛ» подаётся «1» на время «Время пуска», после чего на выходе «Управление ОТКЛ» возвращается «0». Если в течение «Времени пуска» состояние пускателя не изменилось на «0», т.е. пускатель не отключился, процедура повторяется до тех пор пока есть попытки останова (также устанавливается входом «Попыток перезапуска»), если не дождался необходимого значения пускателя устанавливается флаг ActuatorFail_OFF. Вход Сброс (ResetAlarm) так же сбрасывает флаги ActuatorFail_OFF, ActuatorFail_ON и флаг конца попыток АПВ(APV_TryCnt_Null). Флаг состояния «Авария» также может быть сброшен с помощью механизма автоматического повторного пуска (АПВ). Если установлен флаг «Авария», а флаг «Блокировка» снят и количество попыток перезапуска по АПВ (задаётся входом «АПВ – попыток при аварии») не = 0, начинается отсчёт интервала перезапуска АПВ, по истечении которого флаг «Авария» сбрасывается, а количество оставшихся перезапусков АПВ декрементируется.

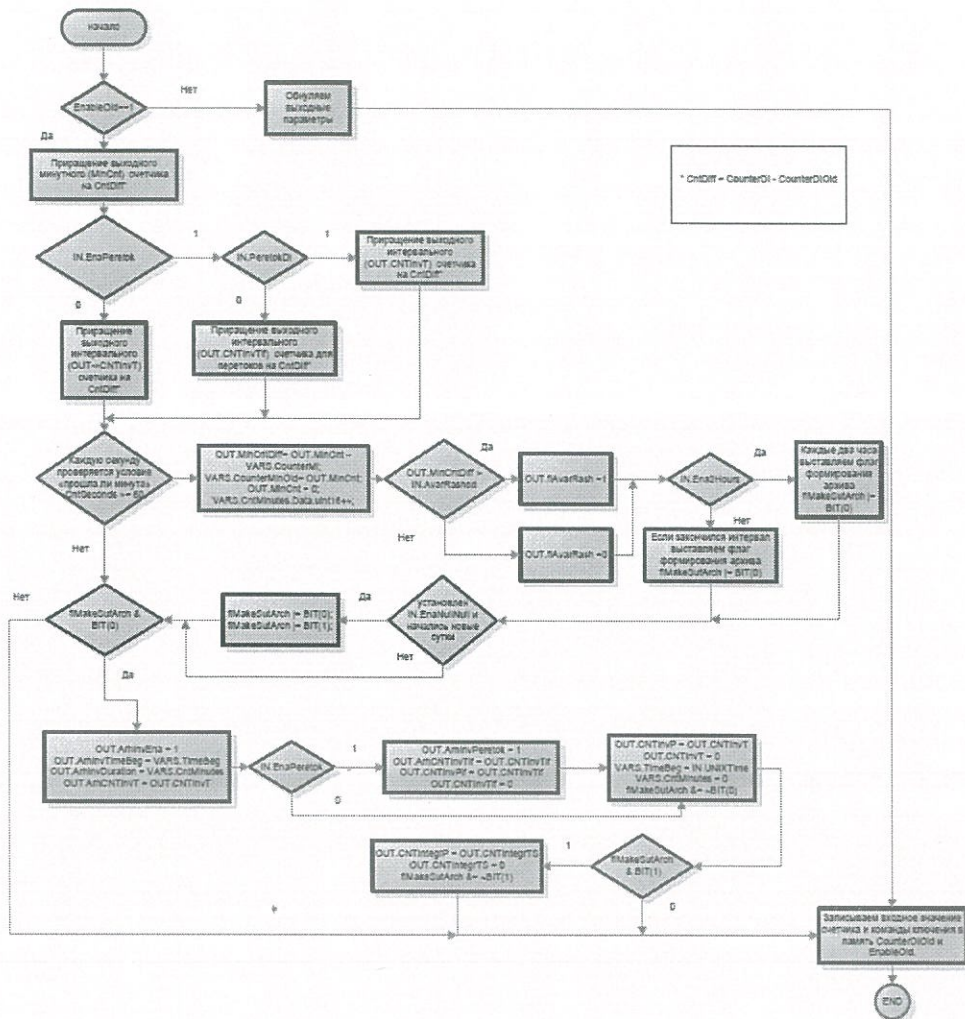
4.12.2 ФБ №91 «Счетчик БГ»

ФБ формирует показания приращения счетчика (вход CounterDI), за определенный интервал времени (в минутах) или за два часа, с возможностью выравнивания в 00:00 часов. На выходах можно наблюдать значение увеличения счетчика за минуту, за сутки, разницу изменения за последнюю и предыдущую минуту. Также формируется флаг аварийного расхода(flAvtRash), если за последнюю минуту расход увеличился, на значение, большее чем установлено в AvarRashod.

Для блоков архивирования, ФБ №91 выставляет бит ArchivEna и устанавливает необходимые значения, если закончился определенный интервал измерений (если установлен признак Ena2Hours то каждые 2 часа) или начались новые сутки.



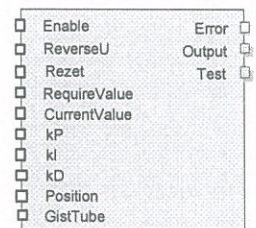
Блок-схема работы ФБ «Счетчик БГ»:



4.12.3 ФБ №99 «ПИД регулятор»

ФБ реализует аналоговое ПИД регулирование. ПИД регулятор формирует импульсы на открытие - закрытие, в зависимости от необходимого положения регулятора, его текущего положения, коэффициентов(kP,kL,kD) и режима работы.

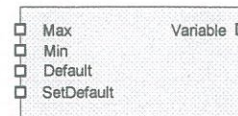
Принцип действия аналогичен ФБ №82, за исключением отсутствия времени задержки DelavTime.



4.13 ФБ для сервиса

4.13.1 ФБ №102 «Уставка»

При первом запуске ФБ переменной (выходу Variable) присваивается значение 0xFF. В дальнейшем значение переменной на выходе варьируется в пределах значений входов Max и Min. Если значение переменной изначально выходит за эти рамки, ФБ подгоняет значение под заданный диапазон. При необходимости обнулись значение переменной подают 1 на вход SetDefault.

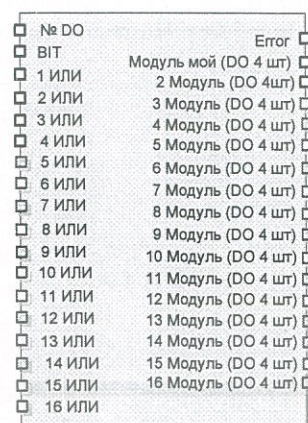


- Входы «Max», «Min» и «Default» и выход «Variable» по умолчанию типа Uint_16
- Вход «SetDefault» - типа BIT

4.13.2 ФБ №107 «Запись DO»

ФБ позволяет управлять состоянием выходов DO контроллера. Запись состояния DO записывается с маской записи. Старшие 4 бита в байте - маска на запись состояния, младшие - состояние DO.

ФБ при последовательном соединении накапливает состояние DO так, чтобы при записи значения DO на одном этапе работы логики не затерло состояние DO установленное на предыдущем шаге. Для этого выход Модуль мой (DO 4 шт.) надо соединить со входом 1 ИЛИ и так далее.



- Вход № DO. Задаёт номер DO для записи его состояния. К примеру: запись числа 7 будет соответствовать изменению состояния 3-го выхода DO 2 модуля (слева), так как у главного контроллера Мастера только 4 выхода DO. В зависимости от числа на входе ФБ будет определять в каком модуле (слева) изменить состояние DO.
- Вход BIT - строго bit. Задаёт состояние Вкл./ Выкл. DO
- Выход Error. Ошибка. Вход № DO содержит значение больше 64 или 0.
- Выход Модуль мой (DO 4 шт.) содержит значение на запись в состояние DO контроллера с маской состояния.
- Выход 2 Модуль (DO 4 шт.) – Аналогичен предыдущему выходу.

7. Процедура обновления файла конфигурации и прошивки контроллера

Обновление прошивки контроллера возможно только если он был первоначально прошит BOOT-прошивкой либо гибридной прошивкой, слитой из BOOT- и USER-версий.

Процедура обновления файлов конфигурации и прошивки контроллера производится в 2 этапа:

- Загрузка файла в буфер контроллера;
- Подача команды на обновление конфигурации/обновление прошивки.

Загрузка файлов в контроллер возможна с использованием протоколов RTM64 и MODBUS RTU (включая MODBUS RTUoverTCP) через окна в адресном пространстве (размер окна 64 байта).

Загрузка файлов может быть выполнена как в режиме USER (прерывать исполнение конфигурации при этом необязательно – фоновый режим загрузки), так и в режиме BOOT. В обоих режимах можно выполнить операцию по применению конфигурации, но обновление прошивки возможно только с переходом в режим BOOT.

Примечание: при загрузке файлов по протоколу RTM64 не прерывая исполнения конфигурации (в фоновом режиме), время загрузки увеличивается. Это связано с тем, что контроллер будет каждый RTM-пакет отдавать на проверку интерфейсным функциональным блокам и только после исполнения цепочки ФБ пакет будет обработан разборщиком RTM64-команд. Т. о. время загрузки в фоновом режиме зависит от частоты исполнения конфигурации логики, т.е. чем чаще исполняется цепочка ФБ, тем меньше будет задержка перед обработкой очередной команды и тем меньше будет время загрузки файла.

Примечание: процедура обновления прошивки может длиться до 10 секунд. Контроллер при этом не реагирует на входящие пакеты, но опрашивает I2C-модули 4 раза в секунду для предотвращения перезагрузки БП во время процедуры обновления. Обновление производится страницами по 256 байт в обратном порядке (начиная с последней страницы). Это сделано для того чтобы минимизировать вероятность порчи программы контроллера и невозможности его включения ни в одном из режимов (BOOT/USER). Если процедура обновления ПО будет прервана (кроме случая когда обновление прервано в самом конце процедуры когда повреждается 0-я страница памяти программ), контроллер при следующем включении будет в режиме BOOT и можно будет повторить процедуру обновления ПО. Если процедура обновления не была начата, можно вернуться к прошлой версии ПО переключившись в режим «USER».

При обновлении конфигурации/прошивки понадобится давать контроллеру команды. Для этого необходимо записывать определённые значения-пароли в специализированные системные регистры, перечень которых и допустимые значения приведены ниже:

Регистр «Команда перезагрузки контроллера (RESET)»:

- Значение «5300» для немедленной перезагрузки контроллера.

Регистр «Команда окончания загрузки файла конфигурации»:

- Значение «5400» для начала процедуры обновления конфигурации из буфера.
- Значение «5401» для копирования рабочей конфигурации в буфер.

Регистр «Команда окончания загрузки файла прошивки»:

- Значение «5500» для переключения контроллера из режима USER в режим BOOT. Если подпрограмма BOOT отсутствует, контроллер вернётся обратно в режим USER.
- Значение «5501» для переключения из режима BOOT в режим USER.
- Значение «5502» для обновления прошивки из буфера. Если контроллер находится в режиме USER, он будет автоматически переведён в режим BOOT. После успешного обновления контроллер будет переведён в режим USER и будет работать по обновлённой программе.

Регистр «Команда включения/выключения исполнения конфигурации»:

- Значение «5600» для отключения исполнения конфигурации. Настройка сохраняет своё действие только до перезагрузки контроллера.
- Значение «5601» для возобновления исполнения конфигурации.

Регистр «Команда расчёта CRC файлов в буферах контроллера»:

- Значение «5700» для расчёта файла конфигурации в буфере конфигурации. Результат помещается в регистр «CRC файла». Если файл отсутствует в буфере, значение в регистре «CRC файла» будет равно «0».
- Значение «5701» для расчёта файла прошивки в буфере прошивки. Результат помещается в регистр «CRC файла». Если файл отсутствует в буфере, значение в регистре «CRC файла» будет равно «0».

7.1 Порядок работы с файлом конфигурации

7.1.1 Чтение файла конфигурации из контроллера

- 1) Если контроллер находится в USER-режиме и исполняет конфигурацию, дать команду на остановку исполнения конфигурации, записан в системный регистр «Команда включения/выключения исполнения конфигурации» значение «5600». Для фонового режима чтения, а также при работе по MOD-BUS, можно пропустить этот шаг.
- 2) Если требуется прочитать текущий файл конфигурации, записать в регистр «Команда окончания загрузки файла конфигурации» значение «5401». Если требуется прочитать содержимое буфера файла конфигурации (например, для проверки корректности записи в буфер), пропустить этот шаг.
- 3) Установить нулевое смещение окна загрузки конфигурации записью «0» в регистры «Адрес внутри буфера файла конфигурации ст.» и «Адрес внутри буфера файла конфигурации мл.».
- 4) Прочитать из окна 2 байта размера файла. При этом контроллер автоматически увеличит смещение окна на 2 байта.
- 5) Прочитать очередной блок файла конфигурации. При этом контроллер автоматически увеличит смещение окна на размер прочитанного блока.
- 6) Выполнять п. 5 до окончания загрузки файла конфигурации. Если на запрос на чтение не пришёл ответ от контроллера, необходимо записать необходимое смещение окна, т.к. контроллер мог его уже увеличить.
- 7) По окончании чтения файла и после проверки CRC прочитанного файла, удалить из его начала 2 байта длины, а также последние 2 байта CRC.

7.1.2 Запись файла конфигурации в контроллер

- 1) Если контроллер находится в USER-режиме и исполняет конфигурацию, дать команду на остановку исполнения конфигурации, записан в системный регистр «Команда включения/выключения исполнения конфигурации» значение «5600». Для фонового режима записи, а также при работе по MOD-BUS, можно пропустить этот шаг.
- 2) Прочитать значение регистра «Максимальный размер файла конфигурации» и убедиться, что загружаемый файл помещается в буфер. В противном случае прекратить процедуру загрузки.
- 3) К файлу, который необходимо загрузить в контроллер, добавить в начало 2 байта длины файла конфигурации (перед этим увеличив его на 4) и 2 байта CRC (вычисленного файла вместе с байтами длины).
- 4) Разбить файл на блоки размером не более 64 байта.
- 5) Добавив в начало блока данных необходимое смещение и записать в контроллер одновременно смещение и данные.
- 6) Выполнять п. 5 до окончания загрузки файла конфигурации.
- 7) По окончании загрузки файла записать в регистр «Команда окончания загрузки файла конфигурации» значение «5400». После этого контроллер проверит CRC файла и если целостность файла не нарушена, обновит свой рабочий файл конфигурации из буфера.

7.2 Порядок работы с файлом прошивки

5.2.1 Чтение файла прошивки из буфера контроллера

- 1) Если контроллер находится в USER-режиме и исполняет конфигурацию, дать команду на остановку исполнения конфигурации, записан в системный регистр «Команда включения/выключения исполнения конфигурации» значение «5600». Для фонового режима чтения, а также при работе по MOD-BUS, можно пропустить этот шаг.
- 2) Установить нулевое смещение окна загрузки конфигурации записью «0» в регистры «Адрес внутри буфера файла прошивки ст.» и «Адрес внутри буфера файла прошивки мл.».
- 3) Прочитать из окна 4 байта размера файла. При этом контроллер автоматически увеличит смещение окна на 4 байта.
- 4) Прочитать очередной блок файла прошивки. При этом контроллер автоматически увеличит смещение окна на размер прочитанного блока.
- 5) Выполнять п. 4 до окончания загрузки файла прошивки. Если на запрос на чтение не пришёл ответ от контроллера, необходимо записать необходимое смещение окна, т.к. контроллер мог его уже увеличить.
- 6) По окончании чтения файла и после проверки CRC прочитанного файла, удалить из его начала 4 байта длины, а также последние 2 байта CRC.

5.2.2 Запись файла прошивки в контроллер (обновление ПО)

- 1) Если контроллер находится в USER-режиме и исполняет конфигурацию, дать команду на остановку исполнения конфигурации, записан в системный регистр «Команда включения/выключения исполнения конфигурации» значение «5600». Для фонового режима записи, а также при работе по MOD-BUS, можно пропустить этот шаг.
- 2) Прочитать значение регистра «Максимальный размер файла прошивки» и убедиться, что загружаемый файл помещается в буфер. В противном случае прекратить процедуру загрузки.
- 3) К файлу, который необходимо загрузить в контроллер, добавить в начало 4 байта длины файла конфигурации (перед этим увеличив его на 6) и 2 байта CRC (вычисленного файла вместе с байтами длины).
- 4) Разбить файл на блоки размером не более 64 байта.
- 5) Добавив в начало блока данных необходимое смещение и записать в контроллер одновременно смещение и данные.
- 6) Выполнять п. 5 до окончания загрузки файла конфигурации.
- 7) По окончании загрузки файла записать в регистр «Команда окончания загрузки файла прошивки» значение «5502». После этого контроллер проверит CRC файла и если целостность файла не нарушена, обновит своё ПО.

8. Приложения

8.1 Приложение А. Перечень системных настроек контроллера для CPU32

Адрес		Название настройки	Ед. изме- рения	Знач. по- умолчанию
RTM64	MODBUS			
1	2	3	4	5
0	0	RTM64-адрес	-	3
2	1	MODBUS-адрес (RTU-адрес)	-	3
4	2	MODBUS-порт (для MODBUS TCP)	-	502
6	3	IP-адрес (для Ethernet) [0]	-	192
8	4	IP-адрес (для Ethernet) [1]	-	168
10	5	IP-адрес (для Ethernet) [2]	-	0
12	6	IP-адрес (для Ethernet) [3]	-	3
14	7	Маска подсети (для Ethernet) [0]	-	255
16	8	Маска подсети (для Ethernet) [1]	-	255
18	9	Маска подсети (для Ethernet) [2]	-	255
20	10	Маска подсети (для Ethernet) [3]	-	0
22	11	IP-адрес основного шлюза (для Ethernet) [0]	-	192
24	12	IP-адрес основного шлюза (для Ethernet) [1]	-	168
26	13	IP-адрес основного шлюза (для Ethernet) [2]	-	0
28	14	IP-адрес основного шлюза (для Ethernet) [3]	-	1
30	15	MAC-адрес контроллера (локально управляемый) [0]	-	0x40
32	16	MAC-адрес контроллера (локально управляемый) [1]	-	0x45
34	17	MAC-адрес контроллера (локально управляемый) [2]	-	0xAA
36	18	MAC-адрес контроллера (локально управляемый) [3]	-	0x00
38	19	MAC-адрес контроллера (локально управляемый) [4]	-	0x00
40	20	MAC-адрес контроллера (локально управляемый) [5]	-	0x01
42	21	Системная дата и время: Год (от 1980 г) ст./ месяц мл.	-	0
44	22	Системная дата и время: день месяца ст./ час мл.	-	0
46	23	Системная дата и время: минуты ст./ секунды мл.	-	0
48	24	Команда перезагрузки контроллера (RESET)	-	-
50	25	Команда окончания загрузки файла конфигурации	-	-
52	26	Команда окончания загрузки файла прошивки	-	-
54	27	Команда включения/выключения исполнения конфи- гурации	-	-
56	28	Команда расчёта CRC файлов в буферах контроллера	-	-
58	29	CRC файла	-	58
60	30	Скорость порта COM0 CPU	100 бод/с	96
62	31	Скорость порта COM1 CPU	100 бод/с	96
64..123	32..61	Скорость порта COM2..COM31	100 бод/с	96
124..255	62..127	Адреса зарезервированы	-	-
256	128	Полное время исполнения конфигурации ФБ (с под- готовкой вх/вых)	0.1 мс	-
258	129	Чистое время исполнения конфигурации ФБ	0.1 мс	-
260	130	Время чтения входов	0.1 мс	-
262	131	Время сохранения выходов	0.1 мс	-
264	132	Пакетов в очереди	шт	-
266	133	Кол-во тайм-аутов	шт	-
268	134	Кол-во повреждённых/неопознанных пакетов	шт	-
270	135	Максимальный размер файла конфигурации	байт	-
272	136	Максимальный размер файла прошивки	Кбайт	-
274	137	Размер ОЗУ ФБ	байт	-
276	138	Размер сохраняемой области ОЗУ ФБ	байт	-
278	139	Кол-во сохраняемых архивов (R7)	шт	-
280	140	№ последнего архива (R7) ст. регистр	-	-
282	141	№ последнего архива (R7) мл. регистр	-	-

Адрес		Название настройки	Ед. измерения	Знач. по умолчанию
RTM64	MODBUS			
1	2	3	4	5
284	142	Общий размер конфигурации	байт	-
286	143	Размер конфигурации межмодульного обмена	байт	-
288	144	Размер конфигурации ФБ	байт	-
290	145	Частота исполнения конфигурации ФБ	Гц	-
292	146	Общее кол-во перезагрузок	шт	-
294	147	Причина последней перезагрузки	-	-
296	148	Версия прошивки CPU32	-	-
298	149	Флаги режимов BOOT/USER	-	-
300..819	150..409	Адреса зарезервированы	-	-
820	410	Номер запрашиваемого архива ст.	-	0
822	411	Номер запрашиваемого архива мл.	-	0
824..887	412..443	Данные (Архив R7) 0..31 слова	-	-
888	444	Адрес внутри буфера файла конфигурации ст.	-	0
890	445	Адрес внутри буфера файла конфигурации мл.	-	0
892..955	446..477	Блок данных файла конфигурации	-	-
956	478	Адрес внутри буфера прошивки ст.	-	0
958	479	Адрес внутри буфера прошивки мл.	-	0
960..1023	480..511	Блок данных файла прошивки	-	-

8.2 Приложение Б. Перечень i2c-адресов модулей

Адреса приведены в 8-битном формате (старшие 7 бит – i2c адрес, а младший бит = 0).

i2c-адрес		Наименование модуля
Dec	Hex	
1	2	3
8	0x08	Модуль БП
10	0x0A	Модуль модем GMSK/FFSK
12	0x0C	Модуль модем RM433
14	0x0E	Модуль модем GPRS
16	0x10	Модуль модем Bluetooth
18	0x12	Резерв
20	0x14	Резерв
22	0x16	Модуль ваттметрирования UI
24	0x18	Модуль DIAI № 1 (16 DI, 4 AI)
26	0x1A	Модуль DIAI № 2 (16 DI, 4 AI)
28	0x1C	Модуль DIAI № 3 (16 DI, 4 AI)
30	0x1E	Модуль DIAI № 4 (16 DI, 4 AI)
32	0x20	Модуль DO4 № 1 (4 DO)
34	0x22	Модуль DO4 № 2 (4 DO)
36	0x24	Модуль DO4 № 3 (4 DO)
38	0x26	Модуль DO4 № 4 (4 DO)
40	0x28	Модуль 4COM № 1 (4 порта RS232/RS485)
42	0x2A	Модуль 4COM № 2 (4 порта RS232/RS485)
44	0x2C	Модуль 8AI № 1 или 8AI2AO № 1 (8 AI, опция – 2 AO)
46	0x2E	Модуль 8AI № 2 или 8AI2AO № 2 (8 AI, опция – 2 AO)
48	0x30	Модуль DO24 № 1 (24 DO)
50	0x32	Модуль DO24 № 2 (24 DO)
52	0x34	Модуль DO24 № 3 (24 DO)
54	0x36	Модуль DO24 № 4 (24 DO)
56	0x38	Модуль модем BRS
58	0x3A	Модуль ДНМ
60	0x3C	Модуль UNIO № 1 (16 DI, 4 AI, 8 DO, 1 AO)
62	0x3E	Модуль UNIO № 2 (16 DI, 4 AI, 8 DO, 1 AO)
160	0xA0	Микросхема памяти EEPROM № 1
208	0xD0	Микросхема часов реального времени (RTC)

8.3 Приложение В. Виды связи протокола RTM64

Код вида связи		Наименование
Дес	Hex	
1	2	3
16	0x10	COM0 ретранслятора
32	0x20	Ближняя радиосвязь
48	0x30	Проводная линия
64	0x40	Дальняя радиосвязь
80	0x50	COM1 ретранслятора
96	0x60	GPRS
112	0x70	Ретрансляция MODBUS пакета, упакованного в RTM64var пакет. Не используется в МегаФБ32

8.4 Приложение Г. Перечень системных настроек контроллера для Мега12

Индекс	Имя	Тип	Размер массива	MOD-BUS адрес	Примечание
0	Адрес_RTМ	KodInt16	1	0	/"Адрес_RTМ" 0, 1 RTM64-адрес.(в старшем бите - "связной")
1	Адрес_Модбас	KodInt16	1	1	/"Адрес_Модбас" 2, 3 ModBus-адрес.(+ обработка есть, но не используется)
2	Rezerv	KodInt16	1	2	/"Rezerv" 4, 5 there was ModBus-TCP-адрес.
3	IP_Адрес	KodInt8	4	3	/"IP_Адрес" 6...9 IP-адрес.&IP ADDR DEFAULT&
4	Код ошибки1	KodInt8	1	5	/"Код ошибки1" 10
5	Код ошибки2	KodInt8	1	5	// "код ошибки2" 5 10
6	CRC_описания_СП	KodInt8	2	6	/"CRC_описания_СП" 12, 13 резерв 1.
7	Маска_IP	KodInt8	4	7	/"Маска_IP" 14...17 IP-маска.&IP MASK DEFAULT&
8	Резерв	KodInt8	4	9	/"Резерв" 18...21 резерв 2.
9	Шлюз	KodInt8	4	11	/"Шлюз" 22...25 IP-адрес шлюза.&IP GATE DEFAULT&
10	Резерв	KodInt8	4	13	/"Резерв" 26...29 резерв 3.
11	MAC1	KodInt8	6	15	/"MAC1" 30...35 MAC-адрес.
12	MAC2	KodInt8	6	18	/"MAC2" 36...41 резерв 4.
13	Год	KodInt8	1	21	/"Год" 42 год.
14	Месяц	KodInt8	1	21	/"Месяц" 43 месяц.
15	День	KodInt8	1	22	/"День" 44 день.
16	Час	KodInt8	1	22	/"Час" 45 час.
17	Минута	KodInt8	1	23	/"Минута" 46 минута.
18	Секунда	KodInt8	1	23	/"Секунда" 47 секунда.
19	Команда_сброса	KodInt16	1	24	/"Команда_сброса" 48-49 5300=Перезагрузка контроллера (RESET).
20	Команда_обн_конф	KodInt16	1	25	/"Команда_обн_конф" 50-51 5400=Обновить конфигурацию из буфера, 5401=Скопировать рабочую конфиг в буфер/надо?
21	Команда_конца_загрузки	KodInt16	1	26	/"Команда_конца_загрузки" 52-53 Команда оканчания загрузки файла прошивки

Индекс	Имя	Тип	Размер массива	MOD-BUS адрес	Примечание
22	Управление_конфигурацией	KodInt16	1	27	///"Управление_конфигурацией" 54-55 5600=отключить,5601=включить исполнение ФБ, 5602=сохранить данные на карту(принудительно)
23	Команда_расчета_CRC	KodInt16	1	28	///"Команда_расчета_CRC" 56-57 5700=посчитать црц буфера конфигурации,5701=буфера прошивки-не реализовано.
24	CRC_файла_прошивки	KodInt16	1	29	///"CRC_файла_прошивки" 58-59 CRC файла.
25	Настройка_COM1	KodInt16	1	30	///"Настройка_COM1" 60,61 скорость COM0. BitRateValue[] = {0-default,1-2400,2-4800,3-9600,4-14400,5-19200,6-28800,7-38400,8-56000,9-57600,10-76800,11-115200}
26	Настройка_COM2	KodInt16	1	31	///"Настройка_COM2" 62,63 скорость COM1.
27	Настройка_INT_BUS	KodInt16	1	32	///"Настройка_INT_BUS" 64,65 скорость COMIntBUS.
28	Настройка_COM_EXT	KodInt16	1	33	///"Настройка_COM_EXT" 66,67 скорость COMExt (GPRS).-записи нет
29	WhileTimer	KodInt32	1	34	// служебный регистр
30	статус_подключенного_устройства	KodInt32	1	36	// "статус_подключенного_устройства"BIT(0) - connected device
31	local_network_ipadr	KodInt8	4	38	//IP-адрес blowран подсети &LOCAL_IP_ADDR_DEFAULT&
32	local_network_ipmask	KodInt8	4	40	//IP-маска blowран подсети &LOCAL_IP_MASK_DEFAULT&-----
33	Адрес конфигурации	KodInt32	1	42	// "Адрес конфигурации"
34	Адрес прошивки	KodInt32	1	44	// "Адрес прошивки"
35	Адрес области	KodInt32	1	46	// "Адрес области"
36	CRC всего файла	KodInt16	1	48	///"CRC всего файла"
37	CRC сетевых переменных	KodInt16	1	49	///"CRC сетевых переменных"
38	Количество архивируемых СП	KodInt16	1	50	///"Количество_архивируемых_СП"
39	Размер_Области_Арх_ОднойСП	KodInt32	1	51	///"Размер_Области_Арх_ОднойСП"
40	ConfigNumber	KodInt16	1	53	//количество параллельно исполняемых конфигураций
41	CurrentConfig	KodInt16	1	54	//номер текущей конфигурации
42	Res6	KodInt16	7	55	//резерв 6. Скорости доп.COMов.
43	Время отключения	KodInt32	1	62	///"Время_отключения"124...127 Время отключения запакованное.
44	Дата отключения	KodInt32	1	64	///"Дата_отключения" 128...131 Дата отключения запакованная.
45	Резерв	KodInt16	2	66	///"Резерв" 132...135
46	Резерв	KodInt16	1	68	///"Резерв" 136-137 Размер свободного стека в байтах
47	Настройки Radio Near	KodInt16	1	69	///"Настройки_Radio_Near" 138-139 настройки ближнего радио
48	Частота_скоростных_каналов	KodInt32	2	70	///"Частота_скоростных_каналов" 140-143 и 144-147 Частота на

Индекс	Имя	Тип	Размер массива	MOD-BUS адрес	Примечание
					спецвыходах 15 и 16
49	Счетчики_цифровых_входов	KodInt32	18	74	/"Счетчики_цифровых_входов" 148-219 счетчики с 0 по 17 (18 шт)
50	Состояние выходов	KodInt16	1	110	/"Состояние выходов" 220...221
51	Состояние аналоговых входов	KodInt16	4	111	/"Состояние аналоговых входов" 222...229
52	Температура	KodInt16	1	115	/"Температура"
53	Предыдущее_состояние_DI	KodInt32	1	116	/"Предыдущее_состояние_DI"
54	Состояние-DI	KodInt32	1	118	/"Состояние-DI"
55	Фильтр DI	KodInt16	1	120	/"Фильтр_DI" период опроса входов DI - для ликвидации дребезга по счетным входам
56	Резерв	KodInt16	6	121	/"Резерв"
57	Флаги_исполнения_конф	KodInt16	1	127	/"Флаги_исполнения_конф"
58	Время_цикла_конф_(по100мкс)	KodInt16	1	128	/"Время_цикла_конф_(по100мкс)"
59	Резерв	KodInt16	1	129	/"Резерв"
60	Резерв	KodInt16	1	130	/"Резерв"
61	Резерв	KodInt16	1	131	/"Резерв"
62	Принятых пакетов	KodInt16	1	132	/"Принятых пакетов"
63	Пакетов_с_истекшим_таймаутом	KodInt16	1	133	/"Пакетов_с_истекшим_таймаутом"
64	BrakePaketsNum	KodInt16	1	134	
65	Макс_размер_конфиг(исполняемая часть)	KodInt16	1	135	/"Макс_размер_конфиг(исполняемая часть)"
66	Макс размер прошивки	KodInt16	1	136	/"Макс размер прошивки"
67	ОЗУ переменных	KodInt16	1	137	/"ОЗУ переменных"
68	ОЗУ_сохраняемое_во_флеш	KodInt16	1	138	/"ОЗУ_сохраняемое_во_флеш"
69	Архивов на странице	KodInt16	1	139	/"Архивов на странице"
70	Номер_последнего_архива	KodInt32	1	140	/"Номер_последнего_архива" всегда номер последнего архива
71	Размер конфигурации	KodInt16	1	142	/"Размер конфигурации"
72	Резерв	KodInt16	1	143	/"Резерв"
73	Кол-во СП	KodInt16	1	144	/"Кол-во СП"
74	Кол-во_СП_собственных	KodInt16	1	145	/"Кол-во_СП_собственных"
75	Перезагрузок	KodInt16	1	146	/"Перезагрузок"
76	Причина перезагрузок	KodInt16	1	147	/"Причина перезагрузок"
77	Версия ПО	KodInt16	1	148	/"Версия ПО"
78	BootFlags	KodInt16	1	149	/"BootFlags"
79	Флаги ошибок	KodInt32	1	150	/"Флаги_ошибок" при обновлении файла конфигурации
80	Флаги ошибок	KodInt32	1	152	/"Флаги_ошибок" при исполнении конфигурации
81	modbus shift	KodInt8	1	154	//смещение регистров модбас "0/1"
82	modbus function revers	KodInt8	1	154	//перенаправление 3 и 4 функции 0 - норм, 1 - инверсия
83	Элементов_очереди_приема	KodInt16	2	155	/"Элементов_очереди_приема" Количество занятых элементов очереди приема

Индекс	Имя	Тип	Размер массива	MOD-BUS адрес	Примечание
84	Настройка_дальнего_радио	KodInt16	1	157	//"Настройка_дальнего_радио"
85	RF mode	KodInt16	1	158	
86	FBCfgSize	KodInt16	1	159	
87	Кол-во тактов в сек	KodInt16	1	160	//"Кол-во тактов в сек"
88	RegTimeOut	KodInt16	9	161	//Секунд с прихода последней команды от мастера канала
89	Ethernetstatus	KodInt16	1	170	
90	GlobalTick	KodInt32	2	171	
91	KernelEventFlags	KodInt32	1	175	
92	ConfigTime	KodInt32	1	177	
93	GuID	KodInt32	3	179	
94	Slave1	KodInt8	12	185	// "слейв 1. Mac0+chan+adr"
95	Slave2	KodInt8	12	191	// "слейв 2. Mac0+chan+adr"
96	Slave3	KodInt8	12	197	// "слейв 3. Mac0+chan+adr"
97	Slave4	KodInt8	12	203	// "слейв 4. Mac0+chan+adr"
98	Slave5	KodInt8	12	209	// "слейв 5. Mac0+chan+adr"
99	Slave6	KodInt8	12	215	// "слейв 6. Mac0+chan+adr"
100	Slave7	KodInt8	12	221	// "слейв 7. Mac0+chan+adr"
101	Slave8	KodInt8	12	227	// "слейв 8. Mac0+chan+adr"
102	Slave9	KodInt8	12	233	// "слейв 9. Mac0+chan+adr"
103	Slave10	KodInt8	12	239	// "слейв 10. Mac0+chan+adr"
104	Slave11	KodInt8	12	245	// "слейв 11. Mac0+chan+adr"
105	Slave12	KodInt8	12	251	// "слейв 12. Mac0+chan+adr"
106	Slave13	KodInt8	12	257	// "слейв 13. Mac0+chan+adr"
107	Slave14	KodInt8	12	263	// "слейв 14. Mac0+chan+adr"
108	Slave15	KodInt8	12	269	// "слейв 15. Mac0+chan+adr"
109	Slave16	KodInt8	12	275	// "слейв 16. Mac0+chan+adr"
110	Slave17	KodInt8	12	281	// "слейв 17. Mac0+chan+adr"
111	Slave18	KodInt8	12	287	// "слейв 18. Mac0+chan+adr"
112	Slave19	KodInt8	12	293	// "слейв 19. Mac0+chan+adr"
113	Slave20	KodInt8	12	299	// "слейв 20. Mac0+chan+adr"
114	Slave21	KodInt8	12	305	// "слейв 21. Mac0+chan+adr"
115	Slave22	KodInt8	12	311	// "слейв 22. Mac0+chan+adr"
116	Slave23	KodInt8	12	317	// "слейв 23. Mac0+chan+adr"
117	Slave24	KodInt8	12	323	// "слейв 24. Mac0+chan+adr"
118	Slave25	KodInt8	12	329	// "слейв 25. Mac0+chan+adr"
119	Slave26	KodInt8	12	335	// "слейв 26. Mac0+chan+adr"
120	Slave27	KodInt8	12	341	// "слейв 27. Mac0+chan+adr"
121	Slave28	KodInt8	12	347	// "слейв 28. Mac0+chan+adr"
122	Slave29	KodInt8	12	353	// "слейв 29. Mac0+chan+adr"
123	Slave30	KodInt8	12	359	// "слейв 30. Mac0+chan+adr"
124	Slave31	KodInt8	12	365	// "слейв 31. Mac0+chan+adr"
125	Slave32	KodInt8	12	371	// "слейв 32. Mac0+chan+adr"
126	MapTable	KodInt16	12	377	// "Таблица соответствия номер в ФБ к номеру в таблице зар. слейвов."
127	SlaveCount	KodInt16	12	409	// "Количество слейвов в таблице зар. слейвов"
128	AdrArch	KodInt32	32	410	//410 820...821 адрес запрашиваемого архива, ст. мл.
129	DataArch	KodInt8	1	412	//412 824...887 данные архива, 64 байта.
130	AdrConfig	KodInt32	1	444	//444 888...891 адрес области конфигурации, ст. мл.
131	DataConfig	KodInt8	64	446	//446 892...953 данные конфигура-

Индекс	Имя	Тип	Размер массива	MOD-BUS адрес	Примечание
					ции, 64 байта.
132	AdrFlash	KodInt32	1	478	//478 956...959 адрес области ПО в SDcard, ст. мл.
133	DataFlash	KodInt8	64	480	//480 960...1023 данные ПО в SDcard, 64 байта.
134	Буф для записи	KodInt8	256	512	// "Буф для записи"
135	link err	KodInt16	1	640	//количество ошибок соединения
136	etharp err	KodInt16	1	641	//количество ошибок arp
137	etharp cachinit	KodInt16	1	642	// количество инициализации cach arp
138	ip frag err	KodInt16	1	643	//количество ошибок при фрагментации ip
139	ip err	KodInt16	1	644	// общее количество ошибок ip
140	icmp err	KodInt16	1	645	//количество ошибок ICMP
141	udp err	KodInt16	1	646	//количество ошибок UDP
142	tcp err	KodInt16	1	647	//количество ошибок TCP
143	mem max used	KodInt16	1	648	//максимальное количество задействованной памяти
144	mem err	KodInt16	1	649	//ошибок работы с памятью
145	tcp seg use	KodInt8	1	650	//количество использованных сегментов TCP
146	tcp seg err	KodInt8	1	650	//количество ошибок выделения сегментов TCP
147	tcp pcb use	KodInt8	1	651	//количество выделенных блоков памяти для TCP
148	tcp pcb err	KodInt8	1	651	//количество ошибок выделения блоков памяти для TCP
149	arp err	KodInt8	1	652	//количество ошибок работы стека ARP
150	pbuf pool err	KodInt8	1	652	//количество ошибок выделения памяти
151	pbuf ref err	KodInt8	1	653	//количество ошибок при ссылке на выделенную память
152	tcp listen err	KodInt8	1	653	//количество ошибок в входящих пакетах
153	route number	KodInt16	1	654	//количество маршрутов
154	NumRecvPacket	KodInt32	9	655	//количество принятых пакетов по каналам
155	reserv buff	KodInt8	190	673	//количество принятых пакетов по каналам

"Утверждаю"

Генеральный директор

ООО НПП "Авиатрон"

_____ Ю.В. Бикбулатов

"__" _____ 2024г.

ПО Конфигуратор ФБ32

Руководство по эксплуатации
ИНТ.01101.094 ИЗ

Издание второе (к релизу 1.14.08.19)

"Согласовано"

Директор департамента РВПАКУТП

С.В. Барбак

"__" _____ 2024 г.

Уфа 2024

Оглавление

Оглавление.....	2
Введение.....	3
Терминология.....	3
ПО конфигуратор ФБ32	Ошибка! Закладка не определена.
Требования конфигуратора МЕГА ФБ32.....	4
Требования к аппаратной части.....	4
Требования к ПО.....	4
Требования по эксплуатации	4
Установка.....	5
Конфигурирование функциональных блоков.....	5
Панель управления.....	7
Добавление схемы конфигурации	8
Редактирование параметров схемы конфигурации.....	8
Удаление схемы конфигурации	9
Чтение конфигурации из контроллера	9
Запись конфигурации из файла.....	10
Сравнение схем конфигурации.....	11
Настройки приложения.....	11
Подключение к БД.....	12
Настройка схемы конфигурации ФБ.....	14
Добавление функционального блока.....	15
Редактирование функционального блока	16
Удаление функционального блока	17
Связывание регистров.....	17
Удаление связи	17
Настройка справочника ФБ.....	18
Настройка списка шаблонов ФБ.....	18
Настройка регистров шаблона ФБ.....	19

Введение

Настоящее руководство составлено с целью ознакомления персонала, обслуживающего программную часть системы со структурой программного обеспечения системы, составляющими программами, их назначением, функциями, особенностями настройки и обслуживания.

Терминология

В настоящем руководстве используется следующая *терминология*:

- программируемый логический контроллер (ПЛК) - микропроцессорное устройство, предназначенное для выполнения алгоритмов управления; принцип работы ПЛК заключается в сборе и обработке данных по прикладной программе пользователя с выдачей управляющих сигналов на исполнительные устройства.

- шаблон функционального блока (далее *шаблон*) - обобщённый образ функционального блока, имеет строго определённый набор регистров и предназначен для выполнения определённой функциональности в контроллере, основываясь на значениях входных регистров и уставок, и выдающий результат своей работы в выходные регистры;

- функциональный блок (далее *ФБ*) - объект, созданный по шаблону функционального блока;

- тип регистра - переменная типа функционального блока;

- регистр - объект, создаваемый по типу регистра;

- входной регистр - вход функционального блока, значение которого может быть завязано на выходной регистр или иметь константное значение, предназначен для инициализации того или иного поведения функционального блока;

- внутренняя переменная или уставка - внутренняя переменная функционального блока, имеющая константное значение, предназначена для инициализации того или иного поведения функционального блока;

- выходной регистр - выход функционального блока, значение которого после выполнения функциональности функционального блока может быть передано на входной регистр другого функционального блока, также это значение может быть записано в память ПЛК;

- схема функциональных блоков - набор функциональных блоков устройства, связанных между собой посредством связей регистров;

- расшифровка кодов - словарь сопоставлений неких числовых и строковых значений для большего восприятия пользователем в момент формирования схемы функциональных блоков.

ПО конфигуратор ФБ32

ПО конфигуратор ФБ32 - программное обеспечение, предназначенное для конфигурирования схемы функциональных блоков контроллеров семейства Мега, производства ООО «НПП «Авиатрон».

Конфигурирование осуществляется заданием связей функциональных блоков либо установкой константных значений на физические входы контроллера или внутренние переменные (уставки) функциональных блоков.

Требования конфигуратора ФБ32

Требования к аппаратной части

Требования к аппаратной части ПО конфигуратора ФБ32:

	Минимум	Рекомендуется
Процессор (частота)	1 GHz	2.5 GHz или выше
Оперативная память	512 Мб	1 Gb и более
Жёсткий диск	50 Мб	100 Мб или более (определяется объёмом базы данных)
Монитор	800x600, 256 цветов	1024x768 или более, True Color
Манипуляторы	Клавиатура и мышь, совместимые с ОС Windows	
Принтер	Принтер, совместимый с ОС Windows	

Требования к ПО

Для работы ПО должна быть установлена одна из операционная систем семейства Microsoft Windows:

1. Microsoft Windows XP;
2. Microsoft Windows 7;
3. Microsoft Windows 8;
4. Microsoft Windows 8.1;
5. Microsoft Windows Server 2003;
6. Microsoft Windows Server 2008.
7. Microsoft Windows Server 2012

Для успешного функционирования системы требуется наличие установленного и сконфигурированного программного обеспечения:

- Windows Installer 3.1;
- .NET Framework 4 Release (поставляется вместе с установщиком конфигуратора МЕГА ФБ32).

Требования по эксплуатации

Ниже представлен перечень требований, соблюдение которых гарантирует стабильную работу системы:

- эксплуатация программного обеспечения ПО конфигуратор ФБ32 должна производиться заказчиком в строгом соответствии с документацией, переданной ему фирмой-разработчиком;
- эксплуатация программного обеспечения ПО конфигуратор ФБ32 должна производиться на исправном компьютере, соответствующем требованиям, предъявляемым программным обеспечением ПО конфигуратор ФБ32 к аппаратной и системной частям компьютера;
- эксплуатация программного обеспечения ПО конфигуратор ФБ32 должна производиться в условиях отсутствия на компьютере клиента компьютерных вирусов, программ-закладок, «тройных» программ и других вредоносных программ, работа которых может внести изменения в данные программного обеспечения ПО конфигуратор ФБ32 или способствовать распространению этих данных за пределы компьютера;
- при эксплуатации программного обеспечения ПО конфигуратор ФБ32 заказчик не может сдавать в наем или аренду, продавать, изменять, декомпилировать, дизассемблировать, изучать код программного обеспечения ПО конфигуратор ФБ32 другими способами, передавать программное обеспечение ПО конфигуратор ФБ32 или любые его составляющие третьей стороне, самостоятельно создавать новые версии программного обеспечения ПО конфигуратор ФБ32 .

- заказчик выделяет ответственного, который отвечает за обслуживание системы (резервное копирование и восстановление данных, настройку и обслуживание служб и программ, необходимых для функционирования системы ПО конфигуратор ФБ32, но не входящих в его поставку.)

- вся ответственность за возможные последствия, которые могут наступить вследствие эксплуатации программного обеспечения ПО конфигуратор ФБ32 в условиях, не соответствующих выше упомянутым, ложится на заказчика.

- при выполнении данных требований Конфигуратор ФБ32 распространяется бесплатно, не предполагает активации и использования ключа/кода активации, в комплекте с ПЛК производства ООО «НПП «Авиатрон».

Производительность системы

Требования к производительности применимы только при использовании рекомендуемых параметров аппаратной конфигурации запускающего компьютера и конфигурации ПО.

- Необходимая производительность редактора схемы конфигурации ФБ при наличии на схеме 2000 элементов (ФБ или групп ФБ):

- Время реакции на перемещение одного элемента схемы не более одной секунды.
- Время реакции на прокрутку схемы не более одной секунды.
- Время реакции на изменение масштаба на 10% не более одной секунды.

- Необходимая производительность модуля чтения/записи 20 000 объектов из БД не более пяти секунд.

- Общая продолжительность загрузки схемы/справочника ФБ не должна превышать пятнадцати секунд.

Установка

Для работы с новым конфигуратором установка дополнительных программ не требуется. Достаточно просто скопировать папку с приложением последней версии и запустить файл sfb32.exe.

Для работы с более ранними версиями ПО требуется перед началом основной установки ПО конфигуратор ФБ32 на машину установить:

- Windows Installer 3.1;
- .NET Framework 4 Release;

Конфигурирование функциональных блоков

Рабочее место **ПО конфигуратор ФБ32** служит для конфигурирования схемы функциональных блоков контроллера.

Запустив программу **ПО конфигуратор ФБ32** перед вами появится следующее предупреждение:

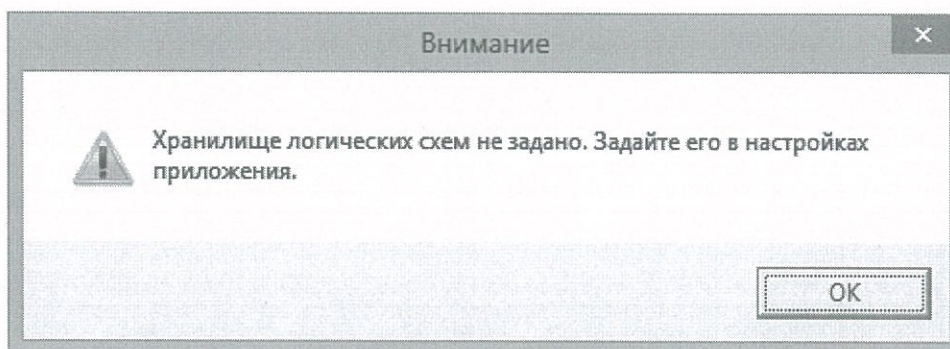


Рисунок 1 – Предупреждение о задании репозитория

Это значит, что в настройках необходимо будет указать репозиторий для хранения схем и шаблонов на компьютере, на котором установлен configurator. После указания в настройках папок для хранения шаблонов и схем появится главное окно программы:

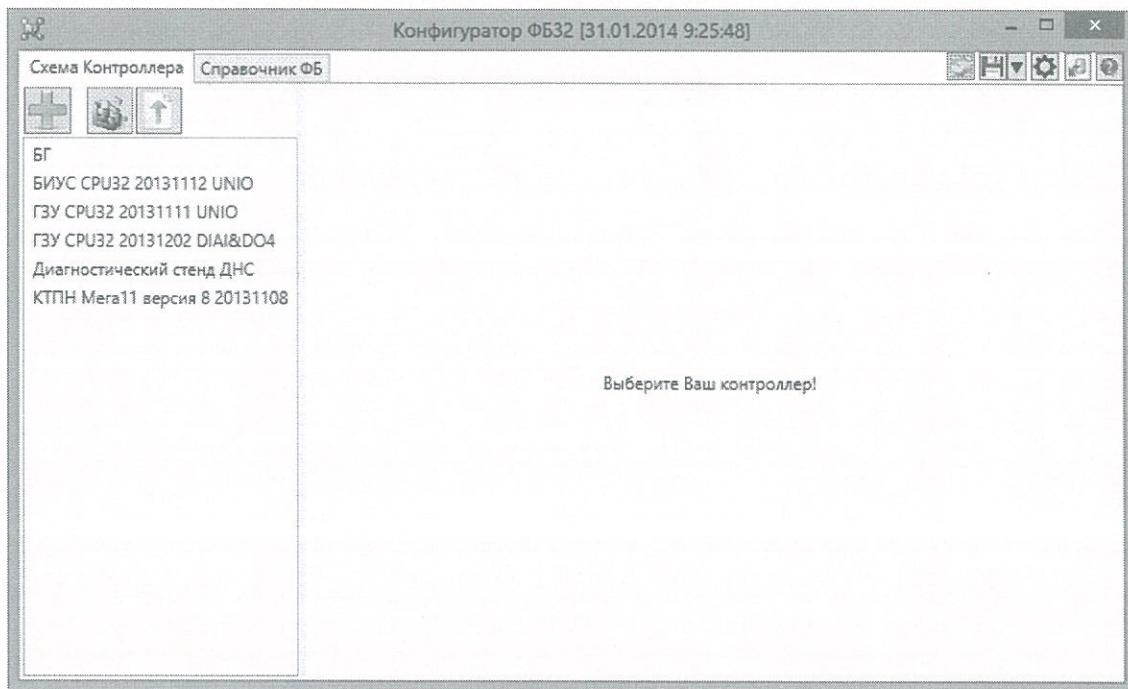


Рисунок 2 – Главное окно программы

Окно содержит две вкладки: вкладка со схемами конфигураций и вкладка с шаблонами Функциональных Блоков.

На вкладке «Схема Контроллера» в левой части расположен список схем конфигураций контроллеров, зарегистрированных в программе. Справа находится панель, в которой отображается схема конфигурации, выделенная в списке.

На вкладке «Справочник ФБ» расположены шаблоны для Функциональных Блоков и функционал для их редактирования.

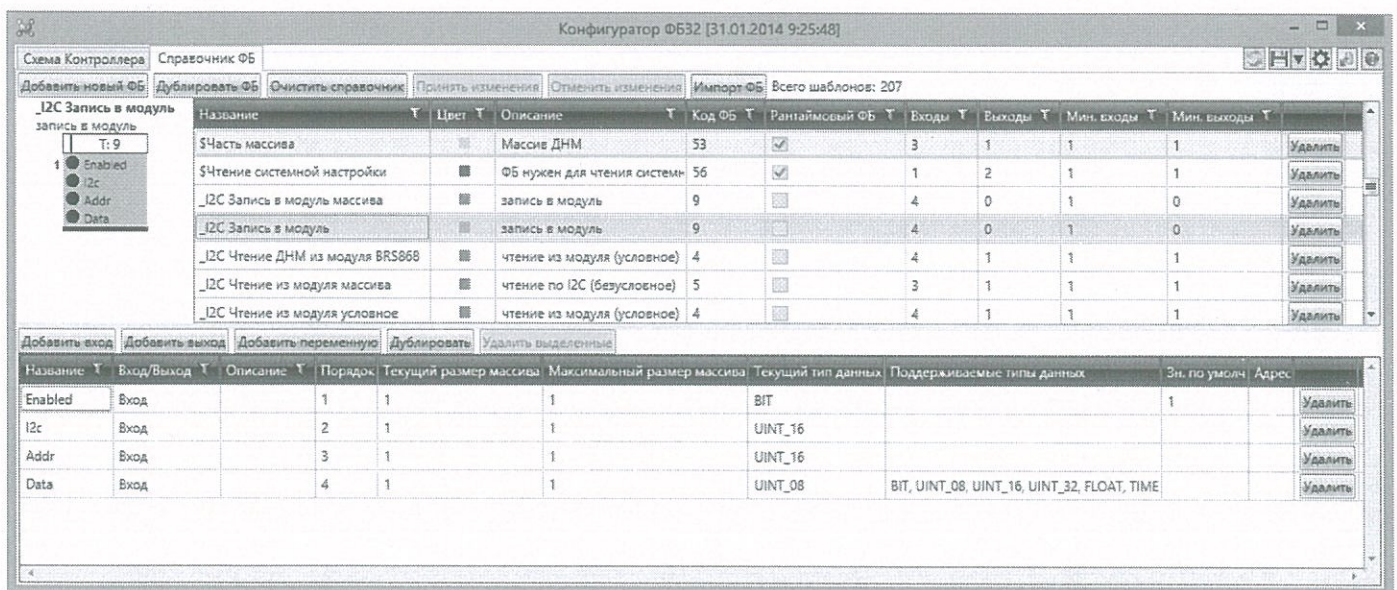


Рисунок 3 – Справочник шаблонов ФБ

Панель управления

Панель управления построена таким образом, что в ней отображаются доступные действия над списком и элементами схем конфигурации.

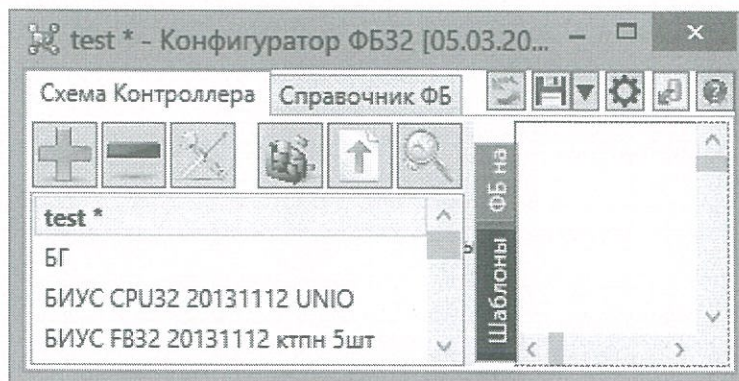













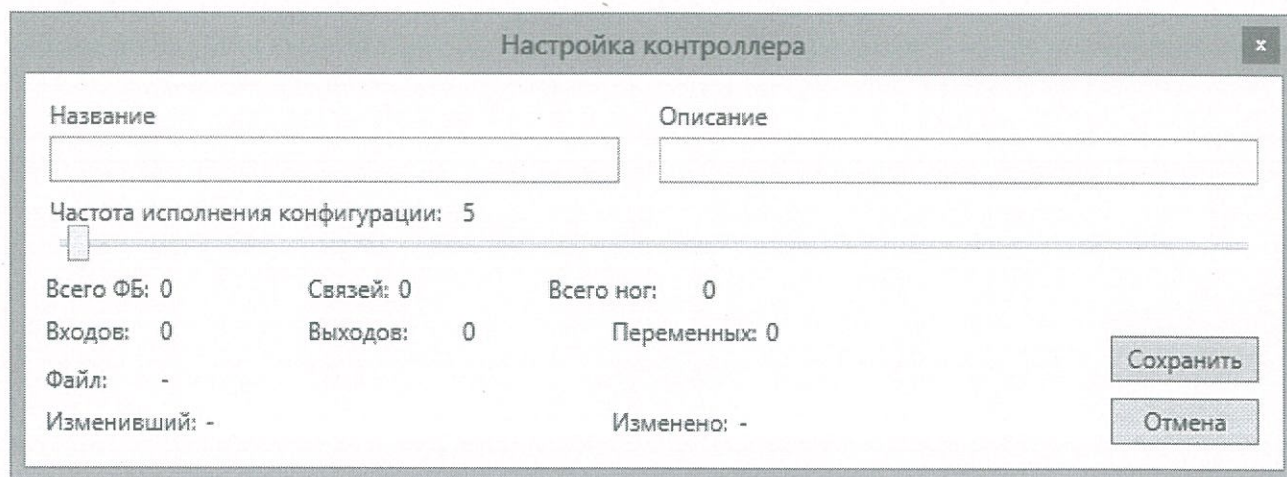
Рисунок 4 – Панель управления

В панели управления могут быть доступны следующие кнопки и действия:

-  **Добавить:** позволяет добавить новую схему конфигурации;
-  **Настроить:** позволяет изменять параметры выбранной схемы конфигурации;
-  **Удалить:** позволяет удалить выделенную схему конфигурации. Удаление схемы всегда сопровождается необходимостью подтверждения;
-  **Читать конфигурацию из контроллера**
-  **Записать конфигурацию из файла**
-  **Сравнить конфигурацию**
-  **Обновить:** перезагружает конфигуратор; предварительно интересуясь о сохранении текущих изменений.
 -  **Сохранить:** позволяет сохранить текущие изменения, либо сохранить изменения в текущей схеме или во всех схемах
 -  **Настройки:** позволяет настраивать параметры работы приложения;
 -  **Подключение к БД:** позволяет подключаться к БД для импорта шаблонов и схем;
 -  **About:** Показывает информацию об используемой версии конфигуратора

Добавление схемы конфигурации

Для **добавления** новой схемы конфигурации необходимо, находясь на элементе схемы в списке, нажать на кнопку «Добавить» на панели инструментов. После этого откроется диалоговое окно, в котором необходимо будет ввести имя схемы и указать частоту исполнения конфигурации контроллером. Здесь также можно задать описание для схемы (рис. 5).



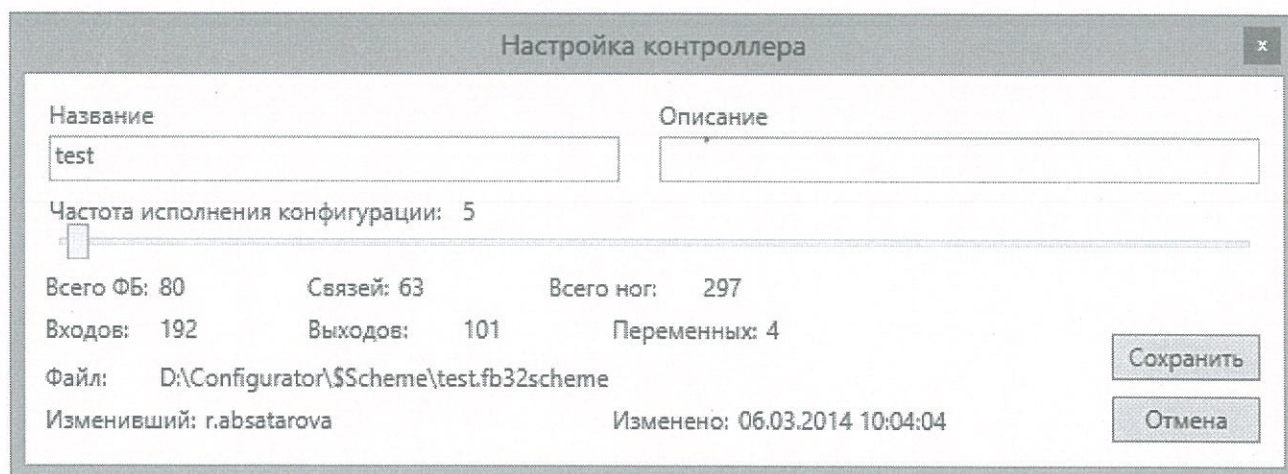
The screenshot shows a dialog box titled "Настройка контроллера" (Controller Settings). It contains the following fields and controls:

- Two text input fields: "Название" (Name) and "Описание" (Description).
- A slider control for "Частота исполнения конфигурации: 5" (Configuration execution frequency: 5).
- Statistical data: "Всего ФБ: 0", "Связей: 0", "Всего ног: 0", "Входов: 0", "Выходов: 0", "Переменных: 0".
- File information: "Файл: -" and "Изменивший: -".
- Buttons: "Сохранить" (Save) and "Отмена" (Cancel).

Рисунок 5 – Добавление новой схемы

Редактирование параметров схемы конфигурации

Для **редактирования** параметров необходимо, находясь на элементе схемы в списке, нажать на кнопку «Настроить» на панели инструментов. После этого откроется диалоговое окно, в котором будет предложена возможность изменить имя устройства, описание и частоту исполнения конфигурации контроллером (рис. 6). Остальные параметры здесь недоступны для редактирования.



The screenshot shows the same "Настройка контроллера" dialog box, but with the following updated information:

- "Название" (Name) field contains the text "test".
- "Частота исполнения конфигурации: 5" (Configuration execution frequency: 5) is shown with the slider.
- Statistical data: "Всего ФБ: 80", "Связей: 63", "Всего ног: 297", "Входов: 192", "Выходов: 101", "Переменных: 4".
- File information: "Файл: D:\Configurator\SScheme\test.fb32scheme" and "Изменивший: r.absatarova".
- Timestamp: "Изменено: 06.03.2014 10:04:04".
- Buttons: "Сохранить" (Save) and "Отмена" (Cancel).

Рисунок 6 – Редактирование схемы

Удаление схемы конфигурации

Для **удаления** схемы конфигурации необходимо, находясь на элементе схемы в списке, нажать на кнопку «Удалить» на панели инструментов.

Чтение конфигурации из контроллера

После нажатия на кнопку «Читать конфигурацию из контроллера» отобразится следующее окно для настройки контроллера:

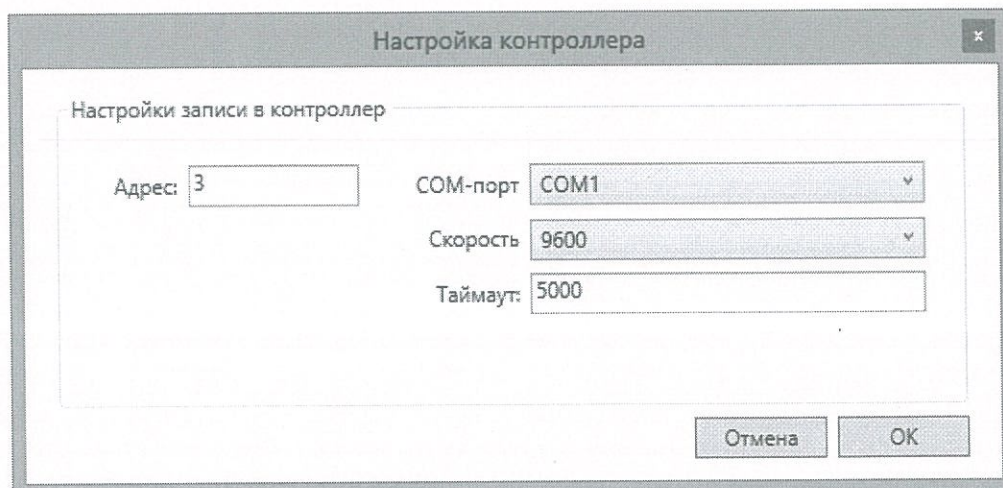


Рисунок 7 – Настройка записи в контроллер

Перед компиляцией конфигуратор обычно запрашивает о копировании карты адресов:

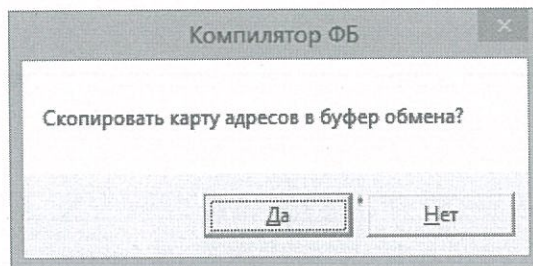


Рисунок 8 – Запрос при компиляции

Если выбрать «Да», в буфер обмена операционной системы будет скопирована карта адресов скомпилированной конфигурации, которую можно будет посмотреть с помощью любого текстового редактора вставив в него содержимое буфера обмена (например, путем Ctrl+V) (рис. 9):

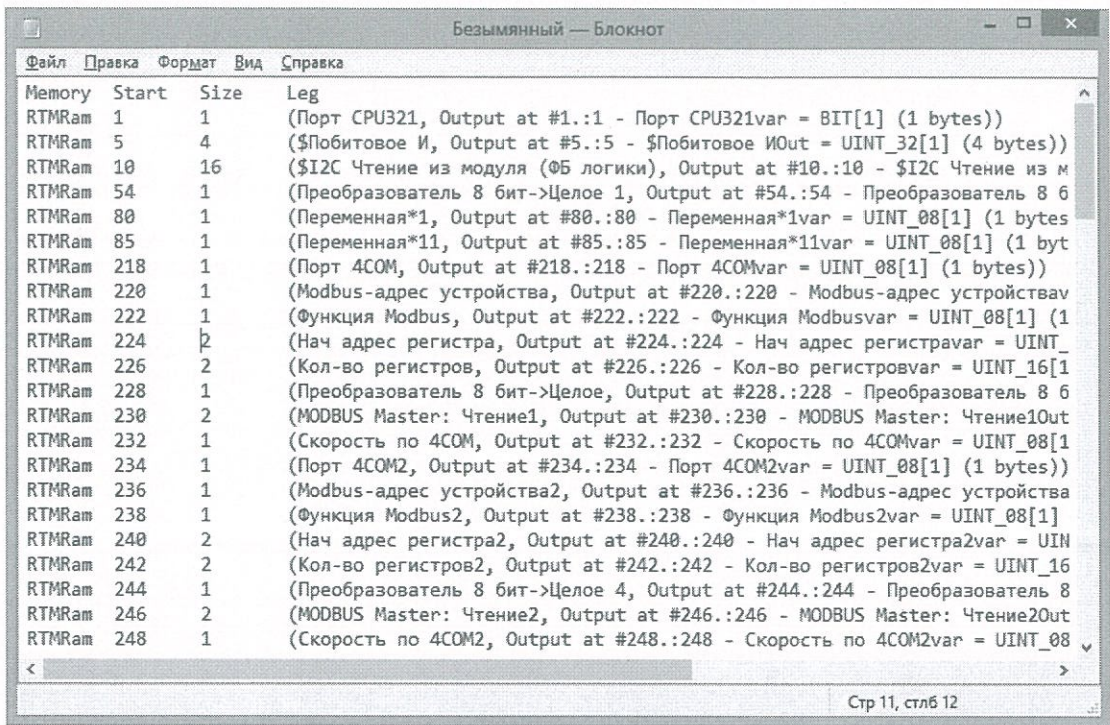


Рисунок 9 – Карта адресов в блокноте

Результат скомпилированной конфигурации, который можно будет записать в контроллер, либо сохранить в бинарный файл для последующей записи, отобразится в окне конфигуратора:

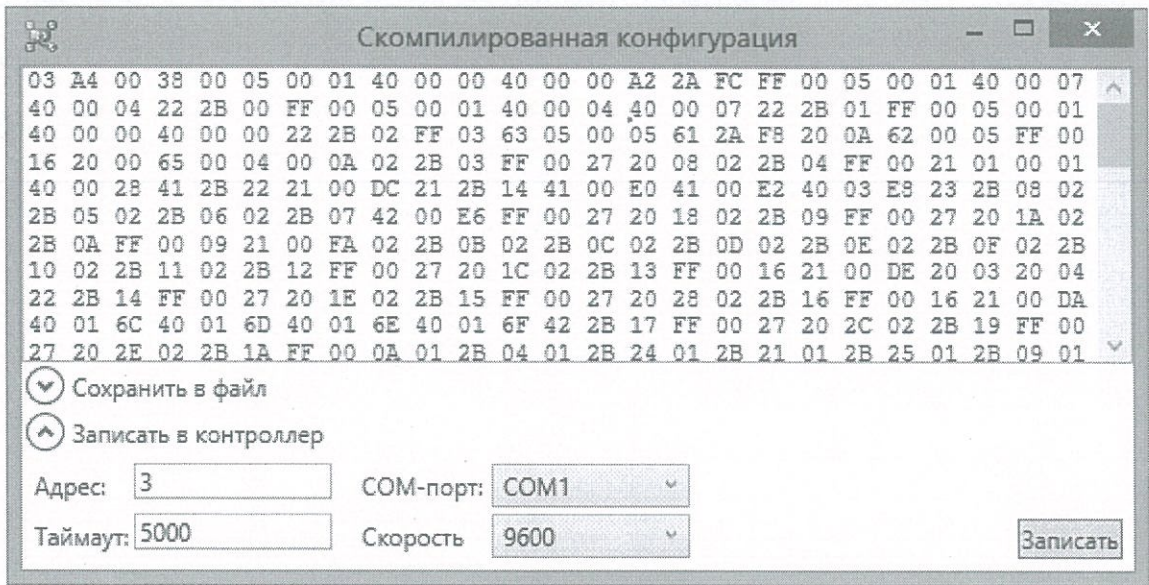


Рисунок 10 – Скомпилированная конфигурация

Запись конфигурации из файла

Выбрав кнопку «Записать конфигурацию из файла», указать файл из которого считывается конфигурация, после чего появится окно, как на рис. 10, и нажать «Записать».

Сравнение схем конфигурации

Для сравнения схем конфигурации необходимо выбрать кнопку «Сравнить конфигурацию», затем из списка представленных схем выбрать ту, которую необходимо сравнить с текущей. После этого Конфигуратор выдаст окно с информацией об имеющихся различиях:

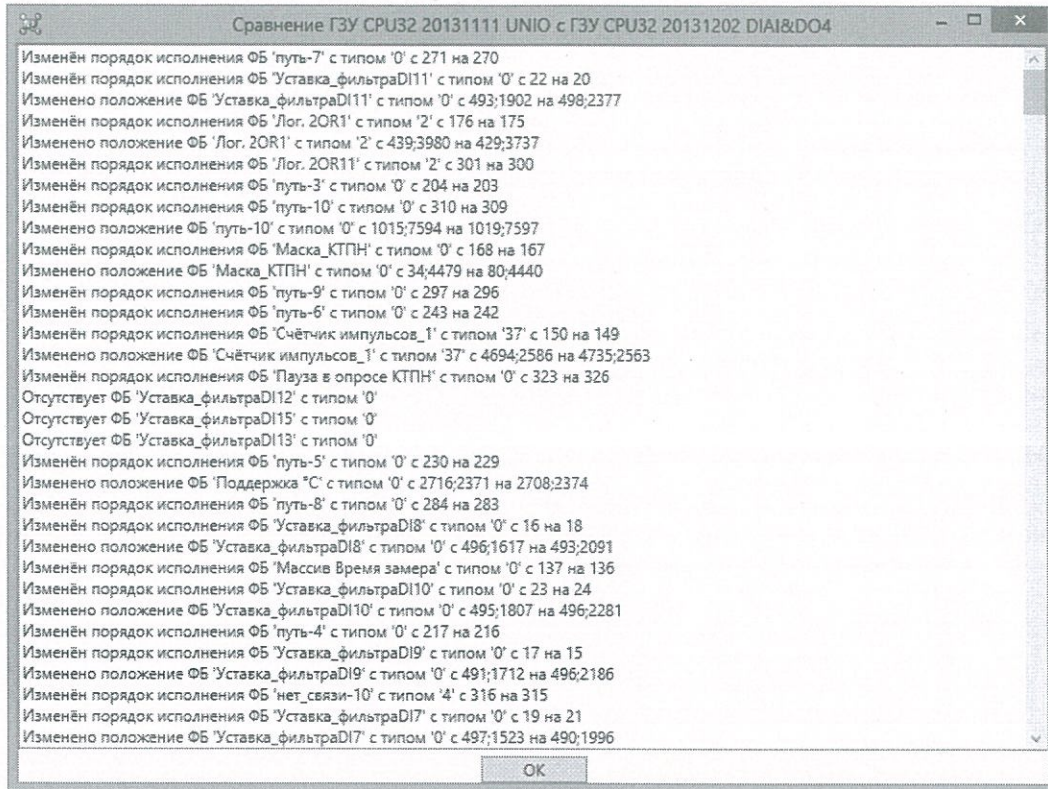


Рисунок 11 – Сравнение схем

Настройки приложения

В настройках приложения задаются хранилища для шаблонов и схем, все проделанные изменения будут заноситься в указанные папки, а не в базу данных.

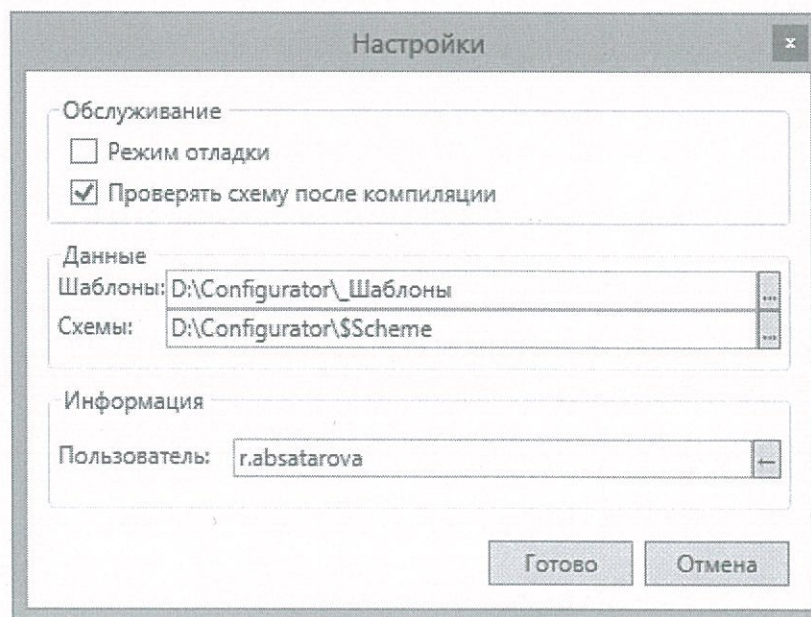


Рисунок 12 – Настройки приложения

Флажок «Режим отладки» по умолчанию не выставляется.

Флажок «Проверять схему после компиляции» позволит Конфигуратору после компиляции сравнивать полученную конфигурацию со схемой и, если количество регистров (ног) ФБ в позиции не будет совпадать, то Конфигуратор выдаст соответствующую ошибку:

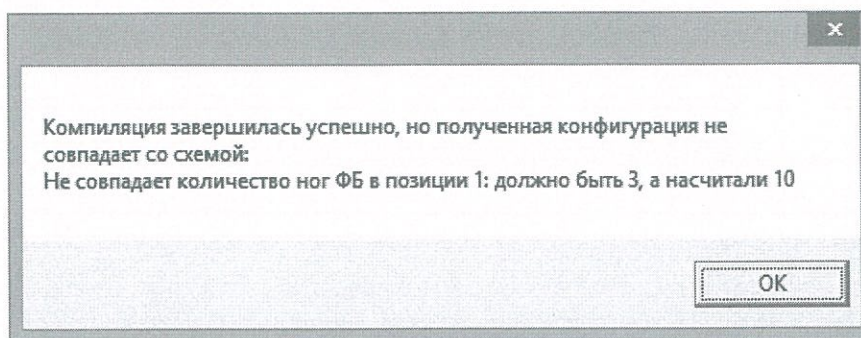


Рисунок 13 – Ошибка при неправильной компиляции

Следующая ошибка возникнет при попытке скомпилировать схему конфигурации ФБ с константой TIME:

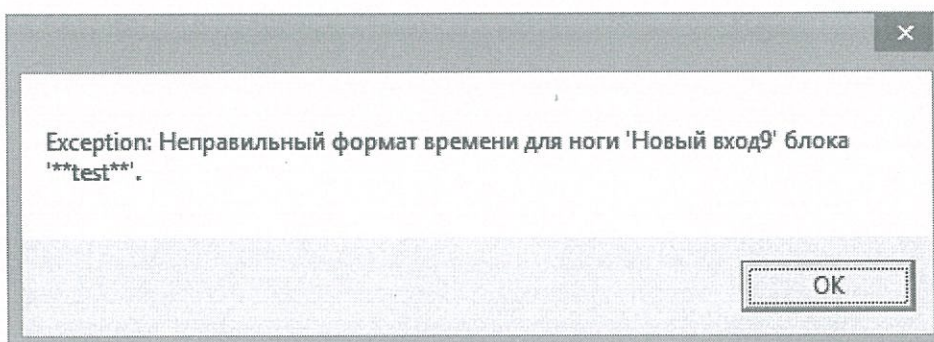


Рисунок 14 – Ошибка компиляции

Подключение к БД

Для *подключения к БД* необходимо нажать на кнопку «Подключение к БД» (рис. 15)

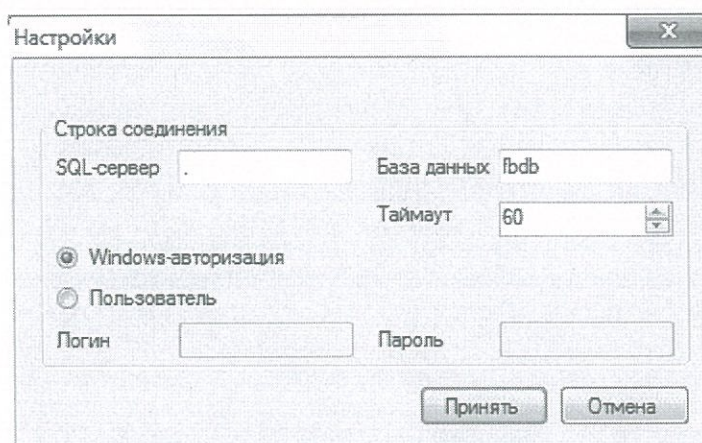


Рисунок 15 – Подключение к БД

В появившемся диалоговом окне (рис. 16) указать шаблоны, которые нужно импортировать из базы. Шаблоны, которые необходимы для функционирования импортируемой схемы подгрузятся автоматически.

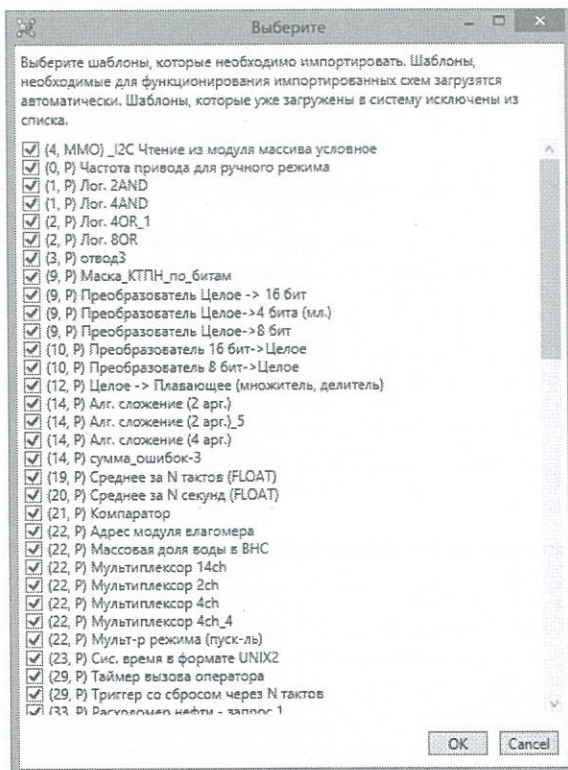


Рисунок 16 – Шаблоны для импорта из БД

Далее следует выбрать схемы, которые нужно импортировать:

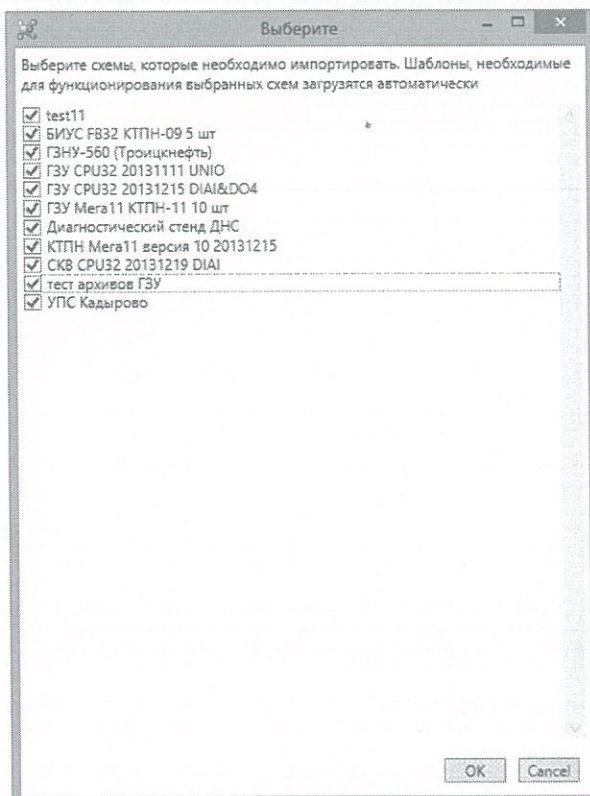


Рисунок 17 – Импорт схем из БД

При подключении к БД со схемами из предыдущих версий конфигуратора к существующим шаблонам добавятся все шаблоны, которые используются в импортируемых схемах. Импортированные шаблоны и схемы будут располагаться в папках, указанных в настройках приложения соответственно. При дальнейшей работе с конфигуратором все внесенные по шаблонам или схемам изменения записываются в данные папки, из БД теперь можно только забрать схемы и шаблоны.

Настройка схемы конфигурации ФБ

Для **изменения схемы функциональных блоков** устройства необходимо, выделить элемент схемы в списке, после чего справа появится панель, представляющая возможности редактирования схемы функциональных блоков (рис. 18).

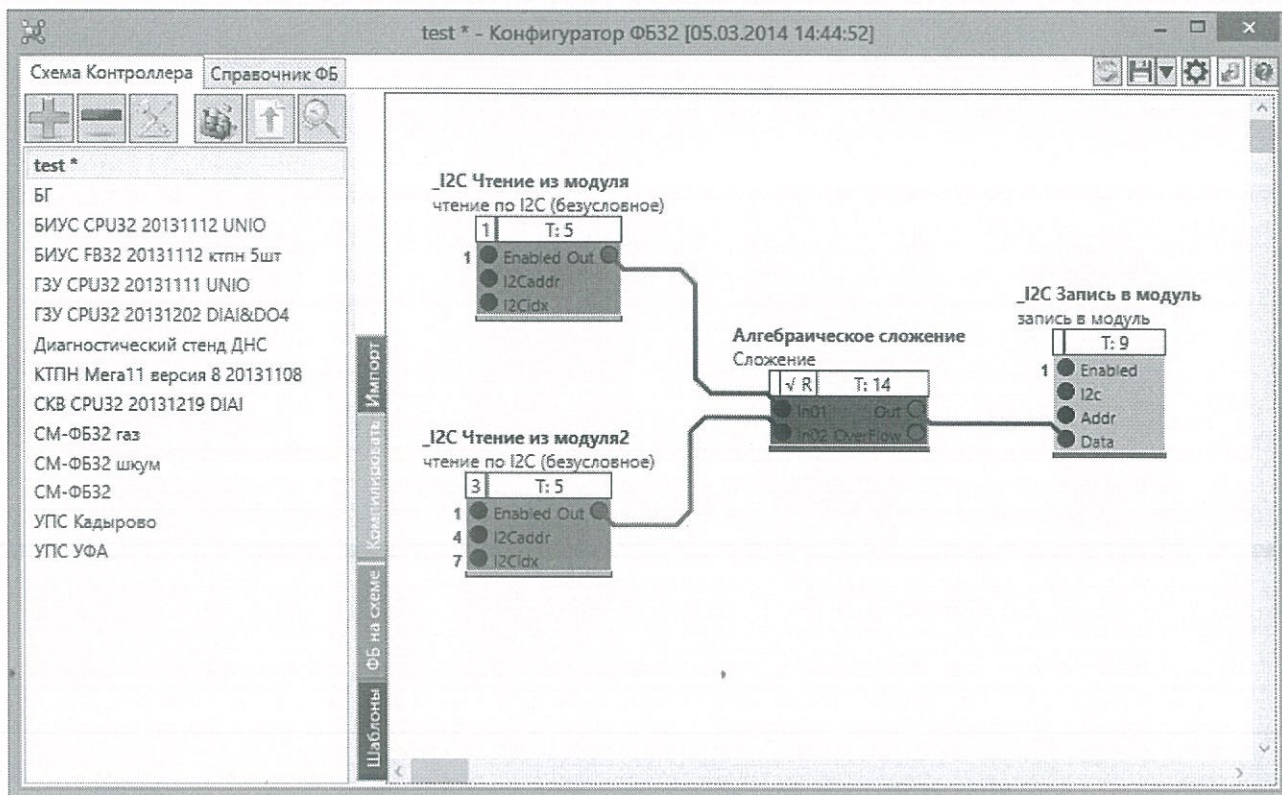


Рисунок 18 – Пример схемы с функциональными блоками

Функциональные блоки на схеме состоят из следующих элементов:

- Сверху название данного ФБ. Название можно изменить для более полного обозначения сути данного ФБ;
 - Под названием расположено описание данного типа ФБ. Описание задаётся в шаблоне. Все ФБ, созданные из данного шаблона имеют одинаковые описания;
 - Ниже расположено тело ФБ. В верхней его части представлены три параметра:
 - 1) порядок исполнения ФБ на схеме (расположен слева, представлен числом)
 - 2) признак принадлежности ФБ (расположен справа) к межмодульному обмену данными (поле пусто) или к логике работы самого контроллера (поле содержит надпись Runtime);
 - 3) Код ФБ из справочника шаблонов ФБ
 - В левой части тела ФБ расположены *входные* параметры (регистры);
 - В правой части тела ФБ расположены *выходные* параметры (регистры).
- Связи между регистрами отображаются красно-синими линиями (синий – вход, красный – выход).

Для входных регистров может быть задано константное значение, которое будет отображено слева от входного регистра.

Для выходных регистров может быть задан адрес расположения регистра в памяти. Если адрес регистра не задан, то он назначается автоматически в момент создания исполняемого файла конфигурации и может отличаться от файла к файлу.

Редактирование типов функциональных блоков в данном контексте невозможно.

Здесь доступны следующие действия:

- изменение масштаба отображения - увеличивать и уменьшать схему для просмотра можно зажав кнопку Ctrl на клавиатуре, прокручивая колёсико мыши. Если мышка без колёсика – масштаб, к сожалению, поменять не получится.

- перемещать ФБ по схеме с помощью стрелок на клавиатуре, удерживая Ctrl

- создать исполняемый файл конфигурации - для создания такого файла нужно нажать кнопку «Компилировать» в левой части панели редактирования и указать имя исполняемого файла конфигурации;

- импорт схемы ФБ - для импорта нужно нажать кнопку «Импорт» в левой части панели редактирования и указать файл импорта схемы;

- отображение списка ФБ на схеме (цвет, порядок, название) - для вызова списка ФБ на схеме нужно нажать кнопку «ФБ на схеме» в левой части панели редактирования;

- отображение доступных шаблонов ФБ - для отображения всех возможных шаблонов нужно нажать кнопку «Шаблоны» в левой части панели редактирования.

Добавление функционального блока

Для **добавления** функционального блока в схему необходимо нажать на объект в панели шаблонов ФБ (рис. 19) с зажатой клавишей Ctrl (выделить можно как элемент в списке названий шаблонов ФБ, так и визуальное представление нужного шаблона). После этого панель шаблонов скроется и выделенный элемент можно положить в нужное место схемы.

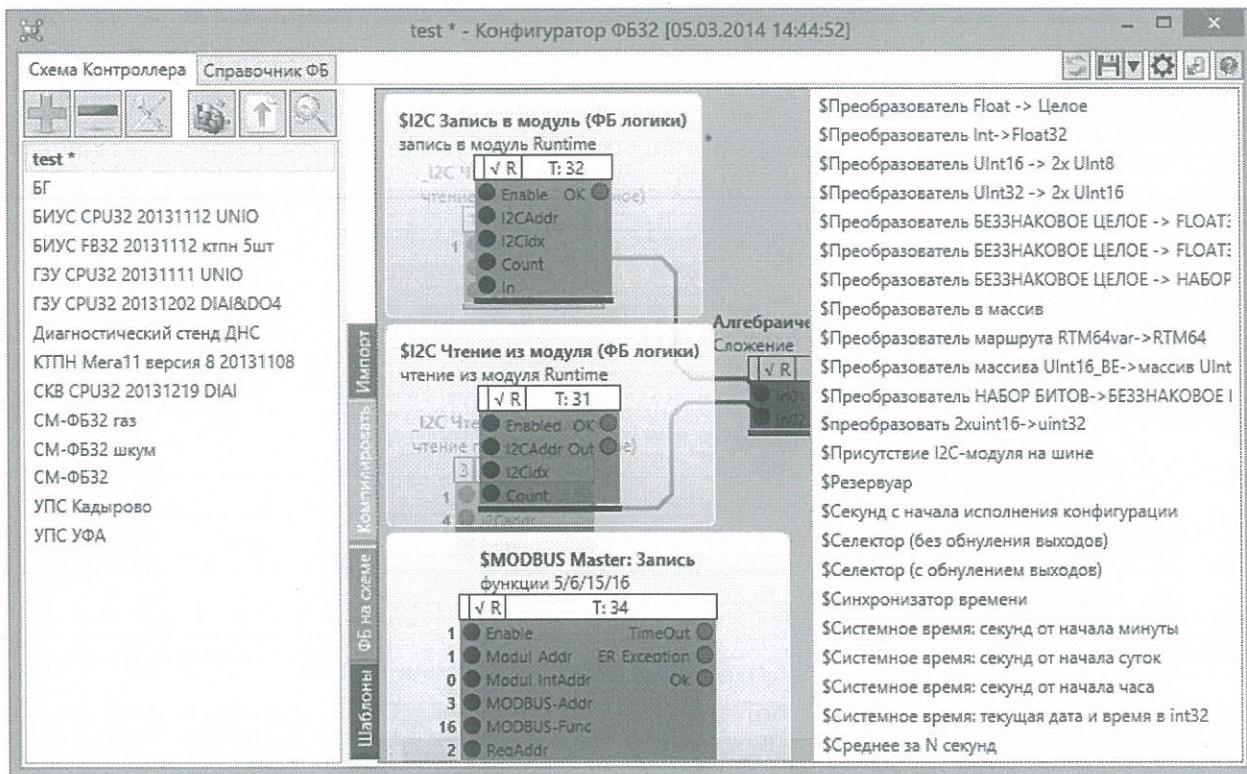


Рисунок 19 – Добавление ФБ из шаблонов на схему

Функция копирования ФБ на схеме путем привычного Ctrl+C, Ctrl+V избавит от необходимости каждый раз добавлять новый ФБ по шаблону.

Редактирование функционального блока

Для **редактирования** функционального блока необходимо, дважды кликнуть на объекте функционального блока на правой панели или выделив объект функционального блока, нажать на кнопку «Редактировать» в появившемся контекстном меню в правой нижней части панели редактирования схемы (рис. 20).

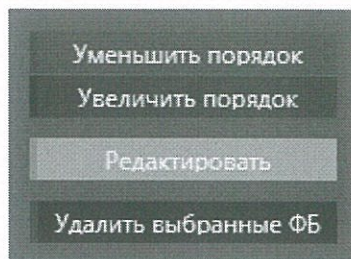


Рисунок 20 – Редактирование ФБ на схеме

После чего в левой части редактирования схемы появится панель редактирования ФБ (рис 21).

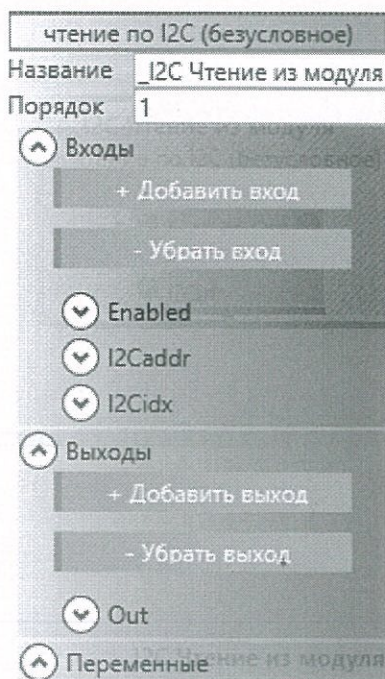


Рисунок 21 – Панель редактирования ФБ

Один функциональный блок должен содержать уникальные имена для входов и выходов, иначе при загрузке схемы с ФБ, которые содержат одинаковые имена входных выходных регистров Конфигуратор ФБ32 выдаст следующее сообщение об ошибке:

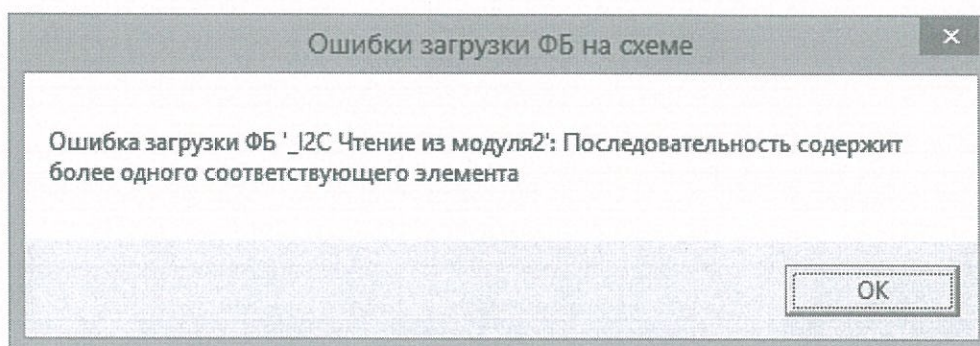


Рисунок 22 – Ошибка загрузки схемы

Такой функциональный блок на схеме отображаться не будет.

Задание константы входному регистру

Для задания константного значения входному регистру необходимо вызвать панель редактирования функционального блока. Далее нужно найти в списке входов нужный, развернуть его и задать ему параметр «Value».

Выбор типа данных регистра ФБ

Для задания типа данных регистра необходимо вызвать панель редактирования функционального блока. Далее нужно найти в списке входов или выходов нужный, развернуть его и задать ему параметр «Type». Данное действие возможно только в том случае, если для соответствующего регистра настроены несколько поддерживаемых типов.

Задание адреса в памяти выходному регистру

Для задания адреса в памяти выходному регистру необходимо вызвать панель редактирования функционального блока. Далее нужно найти в списке выходов нужный, развернуть его и задать ему параметр «Address».

Удаление функционального блока

Для удаления функционального блока необходимо, выделить объект функционального блока и нажать кнопку «Удалить выбранные ФБ» в появившемся контекстном меню в правой нижней части панели редактирования схемы (рис. 16). При этом будут удалены все связи функционального блока.

Возможно также множественное удаление функциональных блоков. Для этого предварительно необходимо выбрать несколько объектов.

Связывание регистров

Связывание регистров функциональных блоков осуществляется проведением линии от одного из выходов ФБ ко одному из входов другого (или того же самого) ФБ. Также линию можно проводить в обратном направлении.

Для создания линии связи регистров необходимо выполнить следующую последовательность действий:

- 1) подвести курсор к входному или выходному регистру до изменения курсора на рисунок руки;
- 2) нажать и держать левую кнопку мыши;
- 3) подвести курсор в выходному или входному регистру;
- 4) отпустить кнопку мыши.

При наличии одной связи у входного регистра вторую связь установить невозможно.

При наличии одной связи у выходного регистра возможно установить вторую связь.

Если проводить линию от входного регистра к выходному, при наличии связи у данного входа с выходом, эта связь отсоединяется от входа и её можно привязать в другому входу любого ФБ. Если бросить линию на пустом месте, то она пропадёт. После проделанных действий новая связь автоматически прорисовывается линией.

Удаление связи

Удаление связи регистров осуществляется следующим способом:

- 1) подвести курсор к соответствующему входному регистру;
- 2) нажать левую кнопку мыши;
- 3) переместить указатель мыши на пустое место схемы;
- 4) отпустить кнопку мыши.

Настройка справочника ФБ

На вкладке «**Справочник ФБ**» главного окна программы представлен перечень шаблонов функциональных блоков, зарегистрированных в системе и инструменты их настройки (рис. 23). Для отображения макета шаблона ФБ необходимо выделить соответствующую строку в списке шаблонов функциональных блоков.

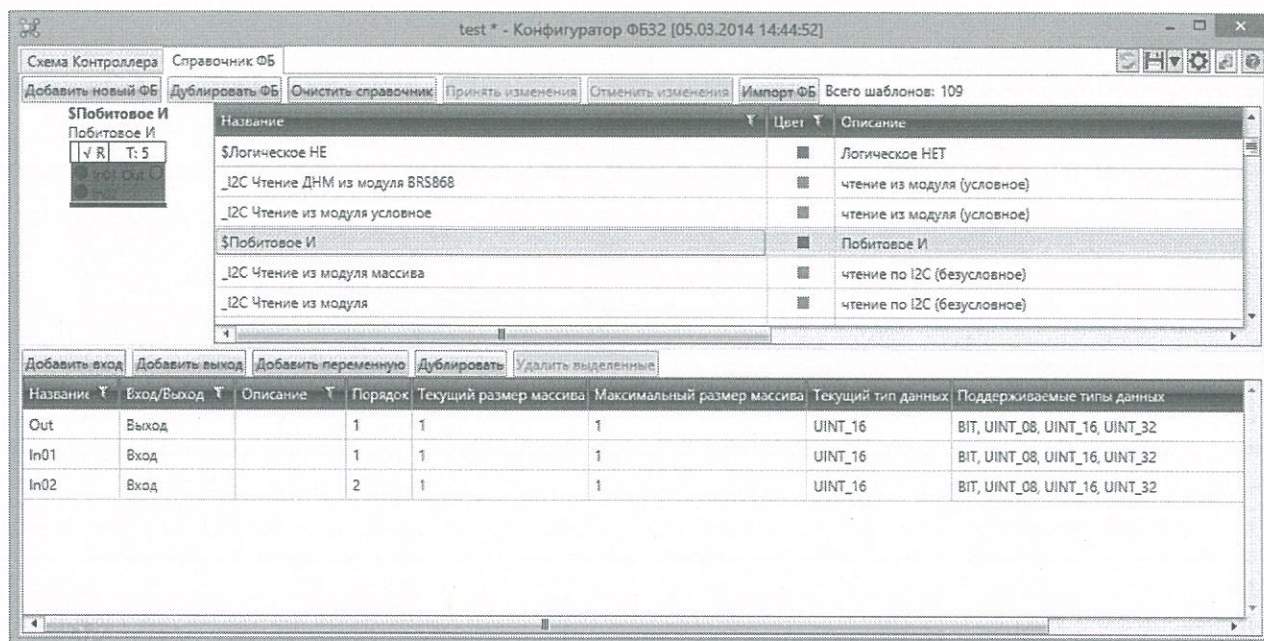


Рисунок 23 – Шаблон ФБ

У шаблона ФБ есть три типа регистров: входы, выходы и переменные.

Все параметры ФБ настраиваются в верхней части окна, регистры шаблона функционального блока в нижней части. Слева расположен макет выделенного шаблона ФБ.

Настройка списка шаблонов ФБ

Для **добавления** нового шаблона ФБ необходимо нажать на кнопку «Добавить новый ФБ». После этого в список ФБ добавится новая строка шаблона, в которой нужно будет ввести необходимые параметры:

- Поле «Название» - название шаблона,
- Поле «Цвет» - цвет тела шаблона (будет отображаться на схеме конфигурации),
- Поле «Описание» - описание шаблона (будет отображаться на схеме конфигурации),
- Поле «Код ФБ» - код ФБ - одна из частей идентификатора шаблона,
- Поле «Рантаймовые ФБ» - признак принадлежности ФБ к логике работы самого контроллера (флажок выставлен) или к межмодульному обмену данными (флажок снят),
- Поле «Входы» - количество входов,
- Поле «Выходы» - количество выходов,
- Поле «Мин. входы» - минимальное количество входов,
- Поле «Мин. выходы» - минимальное количество выходов.

Также добавить шаблон ФБ можно дублированием одного из имеющихся шаблонов. Для этого нужно выделить соответствующий имеющийся шаблон ФБ и нажать на кнопку «Дублировать ФБ». После этого в список ФБ добавится новая строка шаблона, параметры которой, а также все регистры заполнены аналогично исходному шаблону ФБ. Регистры при редактировании шаблона тоже можно дублировать.

Добавлять новые шаблоны следует очень осторожно, четко осознавая, зачем необходимы внесенные изменения и, какую функцию будет выполнять добавленный шаблон ФБ. Желательно, использовать шаблоны, которые уже применялись в схемах конфигурации предыдущих версий.

Если изменить шаблон уже после того, как по нему будут созданы ФБ на схеме, то соответствующие ФБ на схеме обновятся по данному шаблону. Это касается следующих изменений:

- цвет ФБ;
- названия регистров;
- поддерживаемые типы данных. Если из поддерживаемых типов данных убрать тот тип, который уже используется в ФБ на схеме, то в ФБ на схеме он заменится на первый по порядку из списка поддерживаемых типов;
- порядок. При изменении порядка регистров в шаблоне ФБ на схеме связи тоже перепривяжутся, т.е. как было привязано, например, к определенной ноге, так к этой ноге прицеплено и останется.

Если добавлен новый вход или выход в шаблоне, то на схеме он без необходимости не появится, надо будет с помощью панели редактирования ФБ (рис. 20) добавлять вход или выход.

Если добавлена или удалена новая переменная, то данное изменение в схеме конфигурации появится только после перезагрузки конфигуратора.

Если удален выход/вход, то на схеме останется ранее привязанная к данной ноге связь. Она будет привязана либо к следующей по порядку ноге, либо просто останется на схеме непривязанной, а при перезагрузке схемы будет выдавать ошибку:

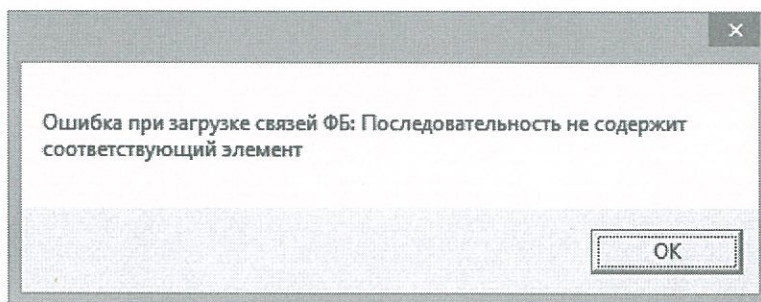


Рисунок 24 – Ошибка при загрузке схемы с некорректными связями

Нужно будет удалить такую связь со схемы вручную.

Для **удаления** шаблона функционального блока необходимо выделить нужный шаблон в списке шаблонов и нажать на кнопку «Удалить» в конце данной строки. При этом будут удалены все регистры этого шаблона и все ФБ, созданные по данному шаблону со всех схем конфигурации.

Настройка регистров шаблона ФБ

Для **добавления** нового регистра к шаблону ФБ необходимо выделить нужный шаблон ФБ в списке и нажать на кнопку «Добавить вход», «Добавить выход» или «Добавить переменную». После этого в список регистров шаблона ФБ добавится новая строка регистра, в которой нужно будет ввести необходимые параметры:

- Поле «Название» - название регистра,
- Поле «Вход/Выход» - тип регистра (вход/выход/переменная) не редактируется,
- Поле «Описание» - описание регистра,
- Поле «Порядок» - порядок регистра в шаблоне, входы, выходы и переменные упорядочиваются по отдельности,
 - Поле «Текущий размер массива» - размер массива регистра по умолчанию,
 - Поле «Максимальный размер массива» - максимально возможный размер массива регистра,
 - Поле «Текущий тип данных» - тип данных регистра по умолчанию,
 - Поле «Поддерживаемые типы данных» - список возможных типов данных регистра, если список не задан, значит используется только тип данных по умолчанию,
 - Поле «Зн. по умолчанию» - константное значение регистра по умолчанию (доступно только для входов),

- Поле «Адрес» - адрес переменной в памяти, в которую сохраняется текущее значение регистра (доступно только для выходов).

Также добавить регистр можно дублированием. Для этого нужно выделить нужный регистр и нажать на кнопку "Дублировать".

Для **удаления** регистра шаблона ФБ необходимо выделить в списке регистров шаблона нужную строку и нажать на кнопку «Удалить» в конце строки. При удалении регистра шаблона ФБ удаляются все регистры функциональных блоков, созданных на основе этого шаблона ФБ.

Следует учитывать, что шаблон ФБ не должен содержать одинаковые имена входов и выходов. При загрузке схемы с ФБ, шаблон которого содержит одинаковые имена входов или выходов Конфигуратор ФБ32 выдаст ошибку с рис. 22 и схема загрузится без данного ФБ и его связей.

ИНТ.01101.094 ИЗ

СОСТАВИЛИ

Наименование организации, предприятия	Должность исполнителя	Фамилия И.О.	Подпись	Дата
ООО НПП «Авиатрон»	Вед. инженер-программист	Афлятунов М.Р.		

СОГЛАСОВАНО

Наименование организации, предприятия	Должность исполнителя	Фамилия И.О.	Подпись	Дата
ООО НПП «Авиатрон»				
ООО НПП «Авиатрон»				
ООО НПП «Авиатрон»				

УТВЕРЖДАЮ

Наименование организации, предприятия	Должность исполнителя	Фамилия И.О.	Подпись	Дата
ООО НПП «Авиатрон»				

"Утверждаю"

Генеральный директор

ООО «НПП «Авиатрон»

Ю. В. Бикбулатов

2024г.



Контроллеры серии Мега ФБ32.

Руководство по конфигурированию ПЛК Мега
с помощью ПО конфигуратора ФБ
ИНТ.01101.094-01 ИЗ

Издание второе (к релизу 1.19.02.19)

"Согласовано"

Директор департамента

Ю. В. Бикбулатов

"24" *с.м.в.* 2024г.

Уфа 2024

Оглавление

1.	Общие сведения.....	3
1.1.	Общие алгоритмы работы.....	3
1.2.	Режимы индикации МегаФБ32	3
1.3.	Структура ПО МегаФБ32	4
2.	Адресное пространство МегаФБ (для доступа снаружи)	5
2.1	. Системные настройки МегаФБ32	5
2.2	. Окно для чтения архивов R7	5
2.3	. Окно для записи файла конфигурации	6
2.4	. Окно для загрузки файла прошивки	6
2.5	. ОЗУ ФБ.....	6
3.	Описание формата файла конфигурации	7
3.1	Структура файла конфигурации	7
3.2	Способ записи значений (входов/выходов/внутренних переменных).....	7
3.3	Описание функциональных блоков МегаФБ32 конфигурации.....	9
3.3.1	Условное чтение из модуля i2c	9
3.3.2	Безусловное чтение из модуля i2c	9
3.3.3	Безусловная запись в модуль i2c.....	9
4	Описание функциональных блоков МегаФБ32 конфигурации логики работы.....	10
4.1	Логические ФБ	10
	ФБ №1 «Логическое И»	10
	ФБ №2 «Логическое включающее ИЛИ»	10
	ФБ №3 «Логическое исключающее ИЛИ»	11
	ФБ №4 «Логическое НЕ».....	11
4.2	Побитовые логические ФБ.....	11
4.3	ФБ преобразования Типов.....	11
	ФБ №63 «Преобразователь UInt16 -> 2x UInt8».....	11
	ФБ № 73 «Преобразователь UInt32/Float -> 2x UInt16».....	11
	ФБ №75 «Преобразователь 2x UInt16 -> UInt32/Float».....	11
	ФБ №74 «Преобразователь Int->Float32».....	12
	ФБ №76 «Масштаб величины»	12
	ФБ №9 «Беззнаковое целое -> Набор битов»	12
	ФБ №10 «Набор битов -> Беззнаковое целое».....	12
	ФБ №11 «Беззнаковое целое -> Float32».....	12
	ФБ №12 «Беззнаковое целое -> Float32 (с множителем и делителем)»	12
	ФБ №13 «Float32 -> целое (отброс дробной части)».....	13
4.4	Алгебраические операции	13
	ФБ №14 «Алгебраическое сложение»	13
	ФБ №15 «Алгебраическая разность».....	13
	ФБ №16 «Алгебраическое умножение»	13
	ФБ №17 «Алгебраическое деление»	13
	ФБ №18 «Алгебраическое возведение в степень»	14
4.5	Вспомогательные ФБ	14
	ФБ №56 «Чтение системной настройки»	14
	ФБ №57 «Запись системной настройки».....	15
	ФБ №19 «Среднее за N тактов»	16
	ФБ №20 «Среднее за N секунд».....	16
	ФБ №21 «Компаратор»	17
	ФБ №22 «Мультиплексор (до 32 входов)».....	17
	ФБ №40 «Селектор (до 32 выходов)»	17
	ФБ №71 «Селектор без обнуления выходов (до 32 выходов)»	17
	ФБ №41 «RS-триггер с приоритетным 0»	17
	ФБ №42 «Триггер с автосбросом»	17
	ФБ №69 «Детектор изменений»	18
	ФБ №22 «Мультиплексор массив»	18
	ФБ №43 «Одновибратор по фронту».....	18
	ФБ №29 «Триггер со сбросом через N тактов»	18
	ФБ №67 «Фильтр дискретный».....	18
	ФБ №106 «Мультиплексор DI».....	18
4.6	ФБ работы с массивами	19
	ФБ №55 «Массив uint8/16/32 -> 1 uint/float»	19

ФБ №53 «Часть массива»	19
ФБ №54 «Компоновщик данных в блоки»	19
ФБ №60 «Преобразователь в массив»	20
ФБ №65 «Преобразователь массива uint16->массив uint8»	20
4.7 ФБ работы со временем и счётчики	20
ФБ №23 «Системное время в UNIX-формате»	20
ФБ №44 «Синхронизатор времени»	20
ФБ №45 «Конвертер времени из UNIX-формата» и ФБ №46 «Конвертер времени в UNIX-формат»	20
ФБ №58 «Первая секунда исполнения конфигурации»	20
ФБ №59 «Секунд с начала исполнения конфигурации»	21
ФБ №37 «Счётчик»	21
ФБ №28 «Длительность импульса (в тактах)»	21
ФБ № 89 «Счетчик входных импульсов»	21
ФБ №81 «Весовой импульс с счетчика»	22
ФБ №61 «Первое исполнение ФБ логики»	22
ФБ №24 «Системное время: секунд от начала минуты»	22
ФБ №26 «Системное время: секунд от начала суток»	22
ФБ №25 «Системное время: Секунд от начала часа»	22
ФБ №78 «Счетчик с ограничением по максимальному приращению»	23
ФБ №27 «Таймер: секунд от начала периода»	23
4.8 ФБ-архиваторы событий	23
ФБ №77 «Архиватор R7 Замер ГЗУ-массомер (тип 16)»	23
ФБ №47 «Архиватор R7 Замер ГЗУ (тип 17, подтип 1)»	24
ФБ №48 «Архиватор R7 Замер БГ (тип 17, подтип 2)»	24
ФБ №49 «Архиватор R7 Изменение состояния объекта (тип 17, подтип 4)»	24
ФБ №50 «Архиватор R7 Изменение DI (тип 17, подтип 5)»	25
ФБ №51 «Архиватор R7 Изменение AI (тип 17, подтип 6)»	25
ФБ №52 «Архиватор R7 Протокол работы объекта (тип 17, подтип 7)»	25
ФБ №30 «Архиватор R7 универсальный (тип 18)»	26
ФБ №88 «Архиватор R7 для ДНМ»	26
ФБ №83 «Архиватор R7 Состояние DI КП КТПН»	27
ФБ №93 «Двухчасовка (счетчик-архиватор)»	27
ФБ №104 «Архивирование сетевых пакетов (СП)»	28
4.9 ФБ работы с I2C-модулями	28
ФБ №39 «Присутствие I2C-модуля на шине»	28
ФБ №5 «Чтение из I2C-модуля безусловное»	28
ФБ №4 «Чтение из I2C-модуля условное»	28
ФБ №9 «Запись в I2C-модуль»	29
4.10 ФБ MODBUS-Master	30
ФБ №33 «MODBUS RTU Master: Чтение - функция 1/2/3/4»	30
ФБ №34 «MODBUS RTU Master: Запись - функция 5/6/15/16»	30
ФБ №110 MODBUS-Модем	31
ФБ №109 MODBUS TCP-Master: Запись	32
ФБ №108 MODBUS TCP-Master: Чтение	33
4.11 ФБ RTM-Master	34
ФБ №35 «RTM64var Master»	34
ФБ №36 «RTM64 Master»	34
ФБ №105 «Запись сетевых переменных»	35
ФБ №103 «Чтение сетевых переменных»	37
4.12 ФБ, специфические для объектов «ГЗУ», «ГЗУ-Массомер»	37
ФБ №38 «ГЗУ (алгоритм Башнефть)» v.1	37
ФБ №66 «ГЗУ (алгоритм Башнефть)» v.2	39
ФБ №72 «Счётчик-Массомер»	39
ФБ №64 «Вычислитель <i>Кнар</i> »	40
ФБ № 68 «Автомат управления приводом по КНАП» (автоматическое повторное включение) ..	40
ФБ №80 «Задвижка»	41
ФБ №82 «ПИД регулятор с задержкой выходного сигнала»	43
ФБ №86 «Резервуар»	45
4.12.1 ФБ №90 «Насос с АПВ» (автоматическое повторное включение)	46
4.12.2 ФБ №91 «Счетчик БГ»	46
4.12.3 ФБ №99 «ПИД регулятор»	47

4.13	ФБ для сервиса.....	48
4.13.1	ФБ №102 «Уставка»	48
4.13.2	ФБ №107 «Запись DO».....	48
5.	Процедура обновления файла конфигурации и прошивки контроллера	49
5.1	Порядок работы с файлом конфигурации	50
5.1.1	Чтение файла конфигурации из контроллера.....	50
5.1.2	Запись файла конфигурации в контроллер.....	50
5.2	Порядок работы с файлом прошивки	51
5.2.1	Чтение файла прошивки из буфера контроллера.....	51
5.2.2	Запись файла прошивки в контроллер (обновление ПО).....	51
6.	Приложения	52
6.1	Приложение А. Перечень системных настроек контроллера для CPU32	52
6.2	Приложение Б. Перечень i2c-адресов модулей	53
6.3	Приложение В. Виды связи протокола RTM64	54
6.4	Приложение Г. Перечень системных настроек контроллера для Mega12	54

1. Общие сведения

1.1. Общие алгоритмы работы

При включении контроллер инициализирует свою периферию, выдаёт приветствие в СОМ-порты и свой RTM64-адрес, производит поиск последней архивной записи R7 в пуле архивов.

Далее контроллер проверяет рабочий файл конфигурации (CRC файла, корректность размеров частей файла). После успешной проверки контроллер читает записанную в постоянной памяти сохраняемую часть ОЗУ, добавляет в цикл межмодульного обмена операции, заданные функциональными блоками межмодульного обмена и настраивает таймер для исполнения логики ядра.

Исполнение конфигурации может быть приостановлено соответствующей командой (при этом также приостанавливается межмодульный обмен, заданный конфигурацией). Возобновить исполнение конфигурации можно командой сверху. Исполнение конфигурации также возобновится при перезагрузке контроллера.

1.2. Режимы индикации МегаФБ32

Контроллер с ПО МегаФБ32 имеет, как минимум, 4 служебных индикатора: «Доступ», «Пакет», «Ответ», «Ошибка».

Индикатор «Доступ» в режиме обновления ПО мигает с частотой 1 Гц. В рабочем режиме этот индикатор горит с половинной яркостью и загорается в полную яркость в те моменты, когда контроллер работает с постоянной памятью (поиск последнего архива R7 в пуле архивов при включении, запись сохраняемой части ОЗУ, сохранение кэша архивов, работа с буферами в постоянной памяти по командам сверху и т.п.). Цвет индикатора может отличаться в различных аппаратных версиях контроллера, обычно красный либо зелёный.

Индикаторы «Пакет», «Ответ», «Ошибка» загораются на 25 мс при соответствующих событиях.

Индикатор жёлтого цвета «Пакет» мигает при получении контроллером адресованного ему корректного запроса либо при получении ответа от устройств, к которым ранее контроллер давал запрос. В первом случае мигание инициирует парсер соответствующего протокола, во втором случае – ФБ, ранее выдавший запрос.

Индикатор зелёного цвета «Ответ» мигает когда контроллер выдаёт ответ на адресованный ему запрос, либо когда контроллер выдаёт запрос вторичному устройству с помощью коммуникационных ФБ. В первом случае мигание инициирует парсер соответствующего протокола в конце своей работы, во втором случае – коммуникационная ФБ.

Индикатор красного цвета «Ошибка» мигает в следующих случаях:

- Коммуникационный ФБ в ответ на свой запрос получил пакет от устройства, содержащий признак ошибки при обработке запроса устройством.
- Входящий пакет имеет недопустимый размер, принятый пакет повреждён или принадлежит неподдерживаемому протоколу. Это может являться следствием, например, неверно установленных настроек канала связи.
- Обнаружены ошибки в файле конфигурации при его исполнении. В этом случае частота мигания индикатора соответствует частоте исполнения конфигурации логики. Причину возникновения ошибки можно считать из системного регистра «Ошибки при исполнении конфигурации».
- Не хватает памяти для размещения очередного принимаемого/отправляемого пакета, очередь пакетов переполнена, кэш архивов переполнен.

Также, непрерывное свечение индикатора «Ошибка» до перезагрузки контроллера сторожевым таймером может быть вызвано неисправностью накопителя постоянной памяти. В этом случае дальнейшая работа контроллера невозможна.

1.3. Структура ПО МегаФБ32

Структура ПО представлена на рисунке 1.

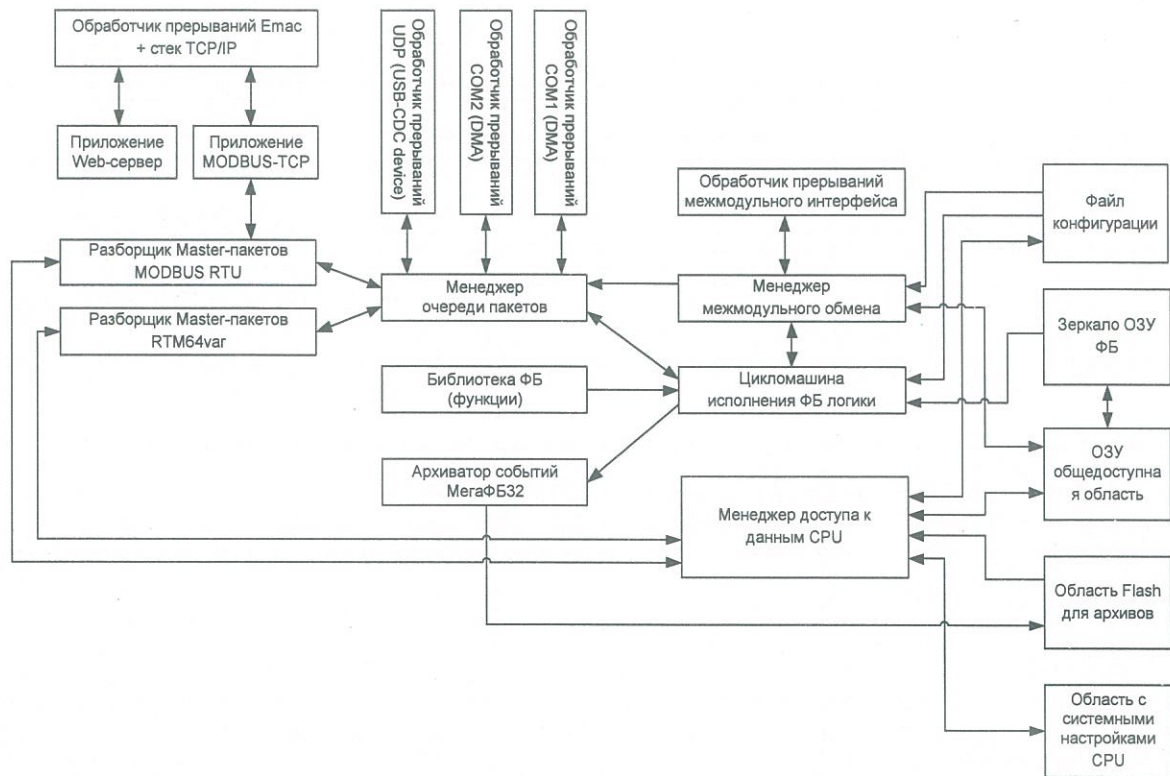


Рисунок 1. Структура ПО контроллера МегаФБ32

2. Адресное пространство МегаФБ (для доступа снаружи)

Контроллер МегаФБ предоставляет доступ следующим областям адресного пространства:

- Область системных настроек.
- Окно для чтения архивов из контроллера.
- Окно для записи файла конфигурации в контроллер.
- Окно для записи файла прошивки в контроллер.
- Область ОЗУ ФБ.

Важно! Порядок байт во всех областях адресного пространства «от младшего к старшему» (Little Endian). RTM64 использует такой же порядок байт, а MODBUS – Big Endian протокол, т.е. контроллер при чтении 03-04 функциями будет менять местами байты в регистрах. Соответственно при записи в контроллер по MODBUS нужно также разворачивать байты в регистрах, так чтобы сначала шёл старший байт, а затем младший.

0h		0		
	Системные настройки		820 байт	
333h		819		
334h	Окно для чтения архивов R7	820	68 байт	1024 байта
377h		887		
378h	Окно для записи конфигурации	888	68 байт	
3BBh		955		
3BCb	Окно для записи прошивки	956	68 байт	
3FFh		1023		
400h		1024		
	ОЗУ ФБ		32768 байт	
83FFh		33791		

2.1. Системные настройки МегаФБ32

Основная настройка контроллера производится с помощью файла конфигурации. Тем не менее, некоторые настройки контроллера (RTM-, MODBUS-, IP-адреса, скорости COM-портов и т.п.) не могут быть произведены только с помощью файла конфигурации, они были вынесены в отдельное адресное пространство CPU32 (адреса с 0 по 819 в RTM64 или HOLDING-регистры с 1 по 409 в MODBUS).

Также в адресном пространстве системных настроек расположены некоторые статистические переменные, а также константы, которые могут меняться от прошивки к прошивке. Эти данные доступны только для чтения.

В памяти контроллера эти настройки поделены на 16-битные регистры. Полный перечень системных настроек приведён в приложении А.

2.2. Окно для чтения архивов R7

Окно для чтения состоит из 34 регистров, первые 2 из которых – номер считываемой архивной записи. Следующие 32 регистра (64 байта) – сама архивная запись R7.

Механизм чтения архивов выглядит следующим образом.

Сначала необходимо записать в первые 2 регистра номер архивной записи, которую нужно считать. Далее необходимо прочитать саму запись из следующих 32 регистров.

Примечание: номер считываемой записи (первые 2 регистра окна) сохраняют своё значение до их перезаписи или до перезагрузки контроллера.

2.3. Окно для записи файла конфигурации

Окно для записи файла конфигурации состоит из 34 регистров, первые 2 из которых – **байтовый** адрес относительно начала файла. Следующие 32 регистра (64 байта) – 64-байтный блок файла.

Блок файла длиной 64 байта записывается за 1 операцию записи вместе с относительным адресом для записи.

Примечания:

- При записи файла конфигурации через это окно, физически данные записываются в буфер во flash памяти контроллера. В процессе записи контроллер продолжает исполнять прошлый вариант файла конфигурации (если он был).
- Если запись была прервана или контроллер перезагружался, допускается продолжить запись с того блока на который было последнее подтверждение об окончании записи.
- При чтении из этого окна, считывается блок текущего используемого файла конфигурации.
- После окончания записи необходимо дать контроллеру команду обновления конфигурации (см. перечень системных настроек в приложении А).

2.4. Окно для загрузки файла прошивки

Окно для записи файла прошивки состоит из 34 регистров, первые 2 из которых – **байтовый** адрес относительно начала файла. Следующие 32 регистра (64 байта) – 64-байтный блок файла.

Блок файла длиной 64 байта записывается за 1 операцию записи вместе с относительным адресом для записи.

Примечания:

- При записи файла прошивки, физически данные записываются в буфер во flash памяти контроллера. В процессе записи контроллер продолжает работать и исполнять конфигурацию.
- Если запись была прервана или контроллер перезагружался, допускается продолжить запись с того блока на который было последнее подтверждение об окончании записи.
- При чтении из этого окна, считывается блок текущего используемого файла прошивки.
- После окончания записи необходимо дать контроллеру команду обновления прошивки (см. перечень системных настроек в приложении А). Это приведёт к переводу контроллера в режим BOOT с последующей прошивкой из подготовленного буфера во flash-памяти.

2.5. ОЗУ ФБ

С адреса 1024 располагается ОЗУ ФБ, т.е. память, используемая ядром ФБ для связи функциональных блоков.

3. Описание формата файла конфигурации

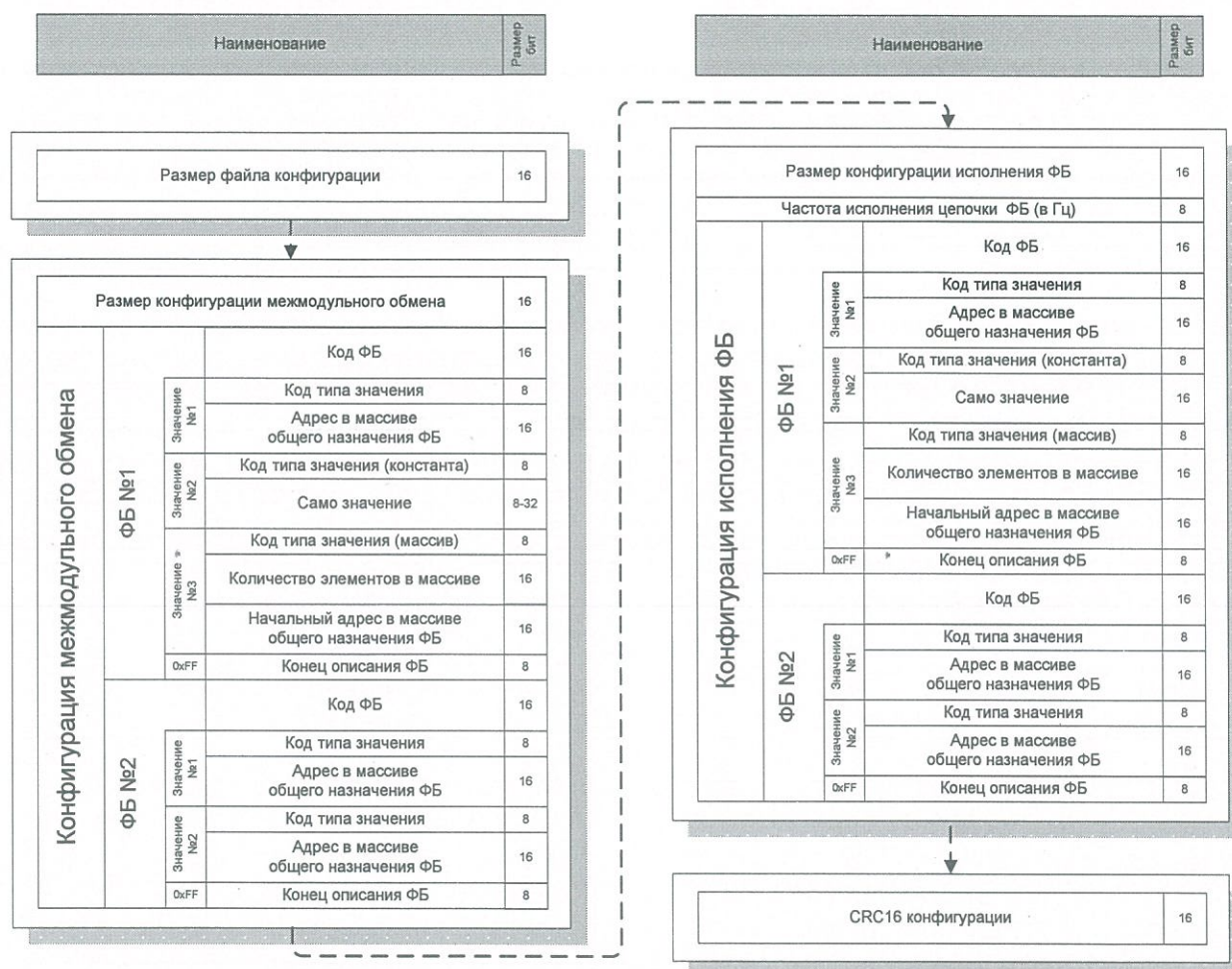
3.1 Структура файла конфигурации

Файл конфигурации ФБ – бинарный файл, структура которого «знаком» контроллеру МегаФБ32. Файл конфигурации ФБ разбит на 2 части: часть описания междомодульного обмена и часть описания логики работы контроллера. Добавлены поля, в которых указывается размер первой и второй частей для быстрого перехода ко второй части конфигурации и для определения конца первой части и конца конфигурации.

Части конфигурации имеют одинаковый формат описания функциональных блоков, в начале очередной части указывается её размер, а для части описания исполнения ФБ также указывается частота исполнения цепочки.

Порядок байт при записи значений размерностью более 8 бит – от старшего к младшему (Big Endian).

Структура файла конфигурации:



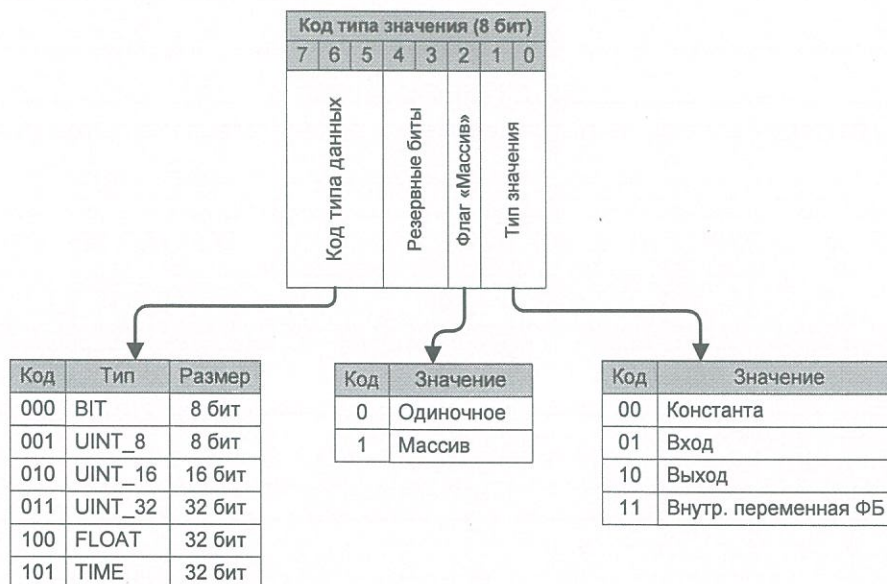
При описании очередного функционального блока в файле конфигурации, первым указывается 16-битный код (номер) ФБ, затем описываются все входы, выходы и переменные ФБ. В конце описания ФБ ставится байт 0xFF.

3.2 Способ записи значений (входов/выходов/внутренних переменных)

Каждый вход/выход/переменная ФБ описывается с помощью кода типа значения, а также служебных полей (начальный адрес в массиве общего назначения, кол-во элементов массива и т.п.), набор и порядок следования которых зависит от типа значения.

Код типа значения описывает тип значения, тип данных, флаг «массив» и т.п. Для констант вместо адреса указывается само константное значение. Для массивов после кода значения указывается кол-во элементов в массиве (от 0 до 65535), а затем начальный адрес его хранения в массиве общего назначения ФБ. Массивы могут быть строго одномерные.

Код типа значения – 1 байт, состоящий из следующих бит:



Примеры записи регистров (входов/выходов/переменных) ФБ:

Пример 1. Описание входа – переменная типа uint16, которая будет прочитана контроллером из ОЗУ ФБ, начиная с адреса 64 (2 байта – 64..65):

0x41	0x0040
Код типа значения	Начальный адрес = 64

Пример 2. Описание входа – константа типа uint32, равная 1'234'567:

0x60	0x0012D687
Код типа значения	Значение = 1234567

Пример 3. Описание выхода – массив из 16 элементов типа uint16, который будет записан в ОЗУ ФБ, начиная с адреса 64:

0x47	0x0010	0x0040
Код типа значения	Кол-во элементов = 16	Начальный адрес = 64

Пример 4. Описание входа – константный массив в файле конфигурации, состоящий из 8 элементов типа uint8:

0x22	0x0008	0x32	0x33	0x34	0x35	0x36	0x37	0x38	0x39
Код типа значения	Кол-во элементов = 8	Эл.массива = 51	Эл.массива = 52	Эл.массива = 53	Эл.массива = 54	Эл.массива = 55	Эл.массива = 56	Эл.массива = 57	Эл.массива = 58

3.3 Описание функциональных блоков МегаФБ32 конфигурации Контроллер ведёт непрерывный интенсивный обмен с модулями и микросхемами, находящимися на шине i2c. Обмен ведётся циклически. Время полного цикла обмена со всеми модулями зависит от кол-ва подключенных модулей и конфигурации межмодульного обмена.

Т. к. i2c-модули перезагружаются при отсутствии обмена с ними, контроллер ежесекундно производит процедуру поиска устройств на шине, попыткой чтения 1 байта из всех 7-битных адресов i2c.

Цикл межмодульного обмена состоит из следующих этапов:

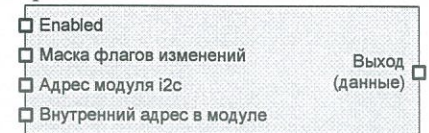
- внеочередные запросы к модулям (если есть);
- чтение даты и времени из микросхемы часов реального времени;
- чтение флагов изменений коммуникационных модулей (модемов, модулей 4COM и т. п.), присутствующих на шине. Если модуль держит в своих буферах пакет, CPU выставит внеочередной запрос на чтение этого пакета и передаст его менеджеру пакетов;
- обмен в соответствии с конфигурацией межмодульного обмена.

Конфигурация межмодульного обмена задаётся с помощью функциональных блоков. Формат задания входов и выходов тот же, что и у ФБ конфигурации логики, но есть некоторые особенности, которые будут рассмотрены в данном разделе

3.3.1 Условное чтение из модуля i2c

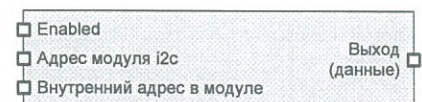
Требования к типам входов/выходов и их количеству:

- Вход «Enabled» типа bit. Может быть константой либо переменной в ОЗУ ФБ.
- Вход «Маска флагов изменений» типа uint32 (строго) содержит 24-битную флагов изменений (младшие 24 бита). Каждый модуль предоставляет доступ к своим внутренним переменным, которые поделены на небольшие группы. При изменении значений в каждой группе, модуль выставляет соответствующий флаг изменений, по которому контроллер узнаёт, какие данные обновились в модуле и ещё не были считаны контроллером. Если модуль установил флаг, который присутствует в маске флагов изменений ФБ условного чтения, то контроллер поставит в очередь менеджера межмодульного обмена соответствующую команду на чтение из модуля.
- Входы «Адрес модуля i2c» и «Внутренний адрес в модуле» строго константы типа uint16.
- Выход – 1 массив, 1 или несколько значений одинакового (любого) типа данных.
- Кол-во байт для чтения из модуля контроллер определяет по размерности выходного массива и типу данных его элементов.



3.3.2 Безусловное чтение из модуля i2c

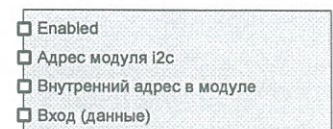
Требования к типам входов/выходов ФБ и их количеству совпадают с ФБ условного чтения, за исключением того, что вход «Маска флагов изменений» отсутствует.



3.3.3 Безусловная запись в модуль i2c

Требования к типам входов/выходов и их количеству совпадают с ФБ безусловного чтения, за исключением того, что выход считываемых данных здесь заменён на вход записываемых данных.

- Вход «Вход» – 1 массив, 1 или несколько значений одинакового (любого) типа данных.
- Кол-во байт для записи в модуль контроллер определяет по размерности массива «Вход» и типу данных его элементов.



4 Описание функциональных блоков МегаФБ32 конфигурации логики работы

4.1 Логические ФБ

Общие требования к типам входов и выходов и их количеству:

- Все логические ФБ работают только с дискретными значениями типа **bit**;
- Выходом логических ФБ является одно дискретное значение;
- ФБ «Логическое НЕ» может иметь только один логический вход. Если на вход будет подано несколько входов, ФБ воспримет только первый, а остальные проигнорирует;
- ФБ "Логическое И", "Логическое ИЛИ" и "Логическое исключаящее ИЛИ" могут иметь разное кол-во входов в пределах от 2 до 32. Если кол-во входов выходит из этих пределов, на выход подаётся «0».

ФБ №1 «Логическое И»

ФБ реализует функцию конъюнкции (или логического умножения) булевой алгебры (это функция двух, трёх или более переменных). Таблица истинности для 3 входов (А, В и С):

А	В	С	Выход
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

Т. е. на выходе будет «1» если все входы ФБ = «1». В противном случае выход = «0».

ФБ №2 «Логическое включающее ИЛИ»

ФБ реализует функцию «дизъюнкции» булевой алгебры (это функция двух, трёх или более переменных). Таблица истинности для 3 входов (А, В и С):

А	В	С	Выход
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

Т. е. на выходе будет «0» если все входы ФБ = «0». В противном случае выход = «1».

ФБ №3 «Логическое исключающее ИЛИ»

ФБ реализует функцию «сложения по модулю 2». Для 3 и более операндов выполняется последовательное сложение по модулю 2.

Таблица истинности для 3 входов (А, В и С):

А	В	С	Выход
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

ФБ №4 «Логическое НЕ»

ФБ реализует функцию логического отрицания или логической инверсии. ФБ может иметь только 1 дискретный вход и 1 дискретный выход. Таблица истинности:

Вход	Выход
0	1
1	0

4.2 Побитовые логические ФБ

Побитовые операции повторяют логические операции (см. ФБ 1...4), применяя их к каждому биту входов.

Общие требования к типам входов и выходов и их количеству:

- Все побитовые логические ФБ работают только с целыми беззнаковыми типами данных (uint8/16/32). При этом входы и выход должны быть одинакового типа;
- ФБ "Побитовое И", "Побитовое ИЛИ" и "Побитовое исключающее ИЛИ" должны иметь 2 входа, ФБ «Побитовое НЕ» - 1 вход;
- Выходом побитовых ФБ является одно целое беззнаковое значение.

4.3 ФБ преобразования Типов

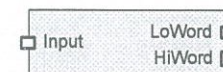
ФБ №63 «Преобразователь UInt16 -> 2x UInt8»

ФБ нужен когда необходимо разделить одно 16 битное целое значение на 2 байта: младший и старший, например, при извлечении данных из 16бит MODBUS-регистров.



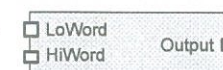
ФБ № 73 «Преобразователь UInt32/Float -> 2x UInt16»

ФБ нужен когда необходимо представить одно 32-битное значение в виде пары 16-битных, например, для разбиения на MODBUS-регистры.



ФБ №75 «Преобразователь 2x UInt16 -> UInt32/Float»

ФБ нужен когда необходимо преобразовать пару 16-битных значений в одно 32-битное, например, чтобы «собрать» из двух MODBUS-регистров значение типа uint32 или float32.



ФБ №74 «Преобразователь Int->Float32»

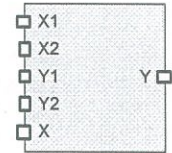
ФБ – простой преобразователь типов. Нужен когда необходимо согласовать вход и выход функциональных блоков, преобразовав целое значение в float32.



ФБ №76 «Масштаб величины»

ФБ «Масштаб величины» производит масштабирование по двум ключевым точкам входной величины X в выходную величину Y.

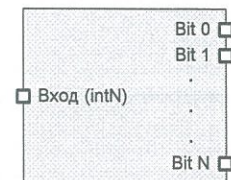
Все входы и выходы ФБ могут быть любого типа данных. ФБ перед вычислением Y приводит все входные величины к типу float32.



ФБ №9 «Беззнаковое целое -> Набор битов»

Требования к типам входов и выходов и их количеству:

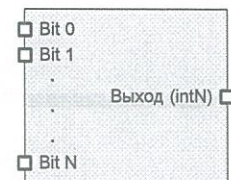
- Вход один, его тип – целый беззнаковый (uint8/16/32).
- Тип выходов – bit, их кол-во соответствует битности входа (8/16/32) и определяется ФБ по типу данных входа.
- Выход может быть задан как отдельными битами в разных частях ОЗУ, так и массивом бит если они в ОЗУ расположены подряд.



ФБ №10 «Набор битов -> Беззнаковое целое»

Требования к типам входов и выходов и их количеству:

- Тип входов – bit, их кол-во соответствует битности выхода (8/16/32) и определяется ФБ по типу данных выхода.
- Вход может быть задан как отдельными битами в разных частях ОЗУ, так и массивом бит если они в ОЗУ расположены подряд.
- Выход один, его тип – целый беззнаковый (uint8/16/32).



ФБ №11 «Беззнаковое целое -> Float32»

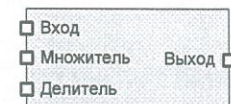
Требования к типам входов и выходов и их количеству:

- Вход один, тип значения – целый беззнаковый (uint8/16/32) либо bit.
- Выход один, тип значения – float32.



ФБ №12 «Беззнаковое целое -> Float32 (с множителем и делителем)»

ФБ приводит значение входа к типу float32, а затем умножает его на множитель и делит на делитель. Если множитель или делитель = 1, соответствующая операция пропускается.



Требования к типам входов и выходов и их количеству:

- Входа три: входное значение (Value) типа bit/uint8/uint16/uint32/float32, множитель (Multiplier) типа bit/uint8/uint16/uint32/float32 и делитель (Divider) типа bit/uint8/uint16/uint32/float32.
- Делитель не может быть = 0. Если он = 0, ФБ принимает его = 1.
- Выход один, тип значения – float32.

ФБ №13 «Float32 -> целое (отброс дробной части)»

Требования к типам входов и выходов и их количеству:

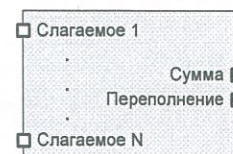


- Входа один, типа float32.
- Выход один, тип значения – uint8/16/32. Допускается также тип bit – на выход будет выдана «1» если вход не равен «0». Если тип выхода float32, выходному значению будет присвоено входное.

4.4 Алгебраические операции

ФБ №14 «Алгебраическое сложение»

Требования к типам входов и выходов и их количеству:



- Входы – слагаемые, от 2 до 32 шт.
- Выход «Сумма» – результат сложения.
- Выход «Переполнение» - битовый выход, «1» на котором означает что целочисленный результат сложения не помещается в переменную типа выхода «Сумма».
- Входы и выходы могут быть разного типа (целочисленные либо float). Если результат не умещается в тип данных выхода, он будет обрезан в соответствии с правилами преобразования типов.
- Если результат сложения вещественный, а тип выхода целый, то в выход запишется целая часть результата работы ФБ.

ФБ №15 «Алгебраическая разность»

Особенность ФБ заключается в том, что результат вычитания выдаётся на выход всегда по модулю. А признак отрицательного результата выдаётся на дискретный выход «Негатив».



Требования к типам входов и выходов и их количеству:

- Входа 2: первый вход – уменьшаемое, второй вход - вычитаемое.
- Выход «Разность» – результат вычитания по модулю.
- Входы и выходы одного типа (целочисленные беззнаковые либо float). При этом если тип значений – целое и результат вычитания отрицательный, на битовый выход «Негатив» будет подана «1».

ФБ №16 «Алгебраическое умножение»

Требования к типам входов и выходов и их количеству:

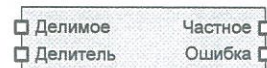


- Входы: 2 аргумента операции умножения
- Выход «Выход» результат умножения.
- Выход «Переполнение» - битовый выход, «1» на котором означает что целочисленный результат умножения не помещается в переменную типа выхода «Сумма».
- Входы и выходы одного типа (целочисленные беззнаковые либо float). При этом если результат не умещается в тип данных выхода, он будет обрезан в соответствии с правилами преобразования типов.

ФБ №17 «Алгебраическое деление»

Требования к типам входов и выходов и их количеству:

- Входа 2: делимое и делитель.
- Входы «Делимое», «Делитель» и выход «Частное» могут быть следующих типов: bit, uint8/16/32 или float32.
- Выход «Частное» – результат деления.
- Выход «Ошибка» - битовый выход, «1» на котором означает что делитель = «0».
- Если выход целочисленного типа, ему будет присвоена целая часть от результата деления.



ФБ №18 «Алгебраическое возведение в степень»

Требования к типам входов и выходов и их количеству:

- Входа 2: основание и степень, в которую её надо возвести.
- Выход «Выход» – результат возведения в степень.
- Выход «Переполнение» - битовый выход, «1» на котором означает что целочисленный результат возведения в степень не помещается в переменную типа выхода «Выход».
- Вход «искомая величина» и выход должны быть одного типа (целочисленные беззнаковые либо float).
- Вход «степень» строго целочисленная беззнаковая.



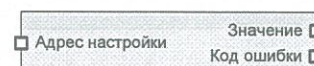
Примечание: если степень = «0», выход будет = «1».

4.5 Вспомогательные ФБ

ФБ №56 «Чтение системной настройки»

ФБ нужен когда какую-либо из системных настроек или системных переменных статистики необходимо использовать (учитывать) в логике конфигурации функциональных блоков.

Требования к типам входов и выходов и их количеству:



5. Вход «Адрес настройки» типа uint16 (допускается uint8 и uint32) задаёт № системной настройки для чтения. Т.к. системные настройки представляют собой 2-байтовые значения, на вход ФБ нужно подавать номер такой 2-байтовой переменной – номера настроек совпадают с адресами MODBUS-регистров (см. Приложения

- Приложение А. Перечень системных настроек контроллера). Например, для чтения скорости порта COM0 модуля CPU32 нужно подать на вход «Адрес настройки» значение «30».
- Выход «Значение» - значение прочитанной настройки («0» если настройку прочитать не удалось). Тип значения – uint16, допускается также uint32.
- Выход «Код ошибки» - код ошибки операции чтения системного регистра. Тип uint8.

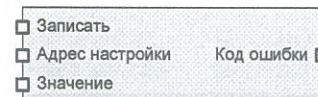
Перечень возможных кодов:

Код	Значение
0	NO ERRORS – нет ошибок при чтении
2	ILLEGAL DATA ADDRESS – неверный (несуществующий) адрес настройки

ФБ №57 «Запись системной настройки»

ФБ нужен когда какую-либо из системных настроек необходимо задать (изменить) в логике конфигурации функциональных блоков.

Требования к типам входов и выходов и их количеству:



- Вход «Записать» - битовый вход, который даёт ФБ команду на запись настройки. Для записи достаточно одного положительного импульса на 1 такт исполнения цепочки ФБ.

Примечание: при записи сохраняемых системных настроек контроллера с помощью данного ФБ, контроллер будет производить их запись во flash-память модуля, что увеличит время исполнения цепочки ФБ. К тому же, ресурс flash-памяти ограничен 100000 циклов перезаписи. Поэтому не рекомендуется оставлять «1» на входе ФБ «Записать» более чем на 1 такт исполнения цепочки ФБ, а подача на вход «Записать» константы «1» может привести к повреждению страницы flash-памяти модуля из-за частых операций перезаписи.

6. Вход «Адрес настройки» типа uint16 (допускается uint8 и uint32) задаёт № системной настройки для чтения. Т.к. системные настройки представляют собой 2-байтовые значения, на вход ФБ нужно подавать номер такой 2-байтовой переменной – номера настроек совпадают с адресами MODBUS-регистров (см. Приложения

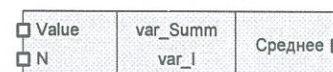
- Приложение А. Перечень системных настроек контроллера). Например, для записи скорости порта COM0 модуля CPU32 нужно подать на вход «Адрес настройки» значение «30».
- Выход «Значение» - значение настройки для записи.
- Выход «Код ошибки» - код ошибки операции чтения системного регистра. Тип uint8.

Перечень возможных кодов:

Код	Значение
0	NO ERRORS – нет ошибок при чтении
2	ILLEGAL DATA ADDRESS – неверный (несуществующий) адрес настройки
3	ILLEGAL DATA VALUE – недопустимое значение для записи

ФБ №19 «Среднее за N тактов»

Требования к типам входов и выходов и их количеству:



- Входа 2: «Value» – входная величина, которую необходимо усреднить, «N» - размер выборки для усреднения.
- Выход один – результат усреднения.
- Кроме того есть 2 внутренние переменные ФБ: «var_Summ» - накапливаемая сумма и «var_I» - кол-во просуммированных входных значений.
- Значения «N» и «var_I» - строго целочисленные.
- Значения «Value», «var_Summ» и «Среднее» должны быть одного типа (либо целочисленные uint8/16/32, либо float32). При выборе типов данных нужно избегать переполнения переменных.

ФБ №20 «Среднее за N секунд»

Требования к типам входов и выходов и их количеству:



- Входа 2: «Value» – входная величина, которую необходимо усреднить, «N» - кол-во секунд для усреднения.
- Выход один – результат усреднения.
- Кроме того есть 3 внутренние переменные ФБ: «var_Summ» - накапливаемая сумма и «var_I» - кол-во прошедших секунд и «var_StartTime» – time32-переменная начала времени начала периода усреднения.
- Значения «N» и «var_I» - строго целочисленные.
- Значение «var_StartTime» - time32 (работать будет и с uint32).
- Значения «Value», «var_Summ» и «Среднее» должны быть одного типа (либо целочисленные uint8/16/32, либо float32). При выборе типов данных нужно избегать переполнения переменных.

ФБ №21 «Компаратор»

Требования к типам входов и выходов и их количеству:

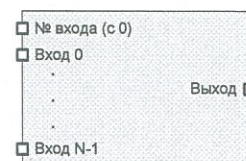
- Входа 3: «Value» - входная величина, «Верхний предел» и «Нижний предел».
- 2 дискретных выхода (тип строго bit).
- Все входы должны быть одинакового типа (либо целочисленные uint8/16/32, либо float32).



ФБ №22 «Мультиплексор (до 32 входов)»

Требования к типам входов и выходов и их количеству:

- Кол-во входов зависит от кол-ва необходимых входных каналов.
- Первый вход всегда – «№ входа» - строго целочисленный, нумерация с «0». Далее следуют входы (максимум 32 шт.). Тип входов и выхода должен быть одинаковым. Допускаются bit, uint8/16/32 или float32.



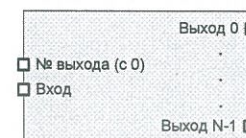
Примечание: если вход «№ входа» > «31», то на выход будет выдан «0».

ФБ №40 «Селектор (до 32 выходов)»

ФБ «Селектор» выдаёт на выход с номером «№ выхода» значение, поданное на вход «Вход». На остальные выходы ФБ выдаёт «0».

Требования к типам входов и выходов и их количеству:

- Кол-во выходов зависит от кол-ва необходимых выходных каналов.
- Первый вход – «№ входа» - строго целочисленный, нумерация с 0. Выходов может быть от 1 до 32. На все выходы кроме выбранного подаётся «0». Тип входов и выхода должен быть одинаковым. Допускаются bit, uint8/16/32 или float32.



ФБ №71 «Селектор без обнуления выходов (до 32 выходов)»

ФБ «Селектор без обнуления выходов» аналогичен ФБ «Селектор», за исключением того, что все выходы кроме выбранного сохраняют свои старые значения. ФБ триггеры

ФБ №41 «RS-триггер с приоритетным 0»

Требования к типам входов и выходов и их количеству:

- 2 битовых входа: «R» (reset) и «S» (set).
- 2 битовых выхода: «T» (прямой выход) и «неT» (инверсный выход).



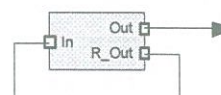
Примечание: если одновременно «R» = «1» и «S» = «1» то триггер сбрасывается.

ФБ №42 «Триггер с автосбросом»

Триггер с автосбросом применяется для подачи команд функциональным блокам, т.е. когда нужно подать на вход ФБ лог. «1» на один такт исполнения цепочки ФБ, после чего сбросить его значение.

Требования к типам входов и выходов и их количеству:

- 1 битовый вход, лог. «1» на котором инициирует выдачу положительного импульса на выход «Out».
- 2 битовых выхода: «Out» для подачи на вход ФБ положительного импульса на 1 такт исполнения и «R_Out» - выход для сброса зна-

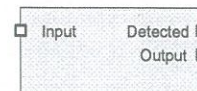


чения, поданного на вход «In». Для сброса входного значения, выход «R_Out» необходимо соединить с входом «In».

ФБ №69 «Детектор изменений»

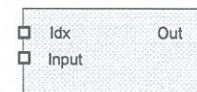
Выдает импульс на выход Detected и записывает его в Output, если входное значение изменилось.

Типы переменных на входе и выходе должны совпадать.



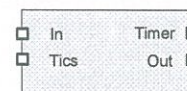
ФБ №22 «Мультиплексор массив»

Выдает на выход значение соответствующее элементу массива Input с индексом Idx. Индексация начинается с нуля. Типы переменных на входе и выходе должны совпадать.



ФБ №43 «Одновибратор по фронту»

Входы		Выходы	
In		Timer	Таймер
Tics	Тактов импульса	Out	Выход (импульс)



При каждом положительном фронте на входе In выставляет «1» в Out на количество тактов равное Tics. Если во время удержания «1» на выходе, прошел еще один фронт на входе то время начинает отсчитываться с этого момента.

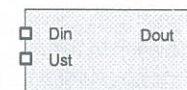
ФБ №29 «Триггер со сбросом через N тактов»

При «1» на входе In FB выставляет «1» в Out и сбрасывает по истечению тактов SetTime после спада на входе Input. Если момент времени при котором на входе «0» меньше чем SetTime, то на выходе «1» не сбросится.



ФБ №67 «Фильтр дискретный»

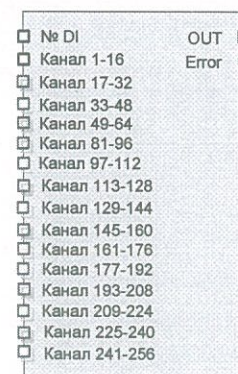
Выход изменяет свое состояние только в том случае если входное значение продержалось больше Ust тактов, тогда Dout == Din.



ФБ №106 «Мультиплексор DI»

ФБ позволяет выделить состояние конкретного входа DI из множества каналов DI. Значения каналов DI могут быть получены путем чтения сетевых переменных у других контроллеров. Таким образом, в логике главного контроллера будет казаться, что у него больше 16 его собственных входов DI.

- Вход №DI. Номер DI.



- Вход Канал 1-16. Значение 16-ти входов DI контроллера Слейва.
- Выход OUT. Состояние DI.
- Выход Error. Ошибка. Номер DI больше 256 или равен 0.

4.6 ФБ работы с массивами

ФБ №55 «Массив uint8/16/32 -> 1 uint/float»

ФБ «Массив uint8/16 -> 1 bit/uint/float» нужен в тех случаях, когда имеется массив байт (полученный, к примеру, от интерфейсных ФБ), который фактически представляет собой единичное значение типа uint8/16/32 или float32. Поскольку в исходном виде (массивом) это значение нельзя использовать в ФБ, принимающих единичные значения указанных типов, необходимо предварительно преобразовать массив в значение соответствующего типа.

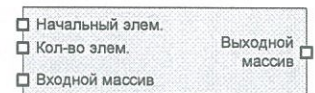


ФБ имеет битовый вход «BigEndian», с помощью которого можно изменять порядок байт, используемый для представления многобайтных значений (Big Endian – от старшего к младшему или Little Endian – от младшего к старшему).

На вход ФБ подаётся массив байт или 2-4 байтных слов, из которых необходимо получить одиночное значение определённого типа. ФБ по типам выходных и выходного значений осуществляет преобразование.

ФБ №53 «Часть массива»

ФБ «Часть массива» нужен для выделения во входном массиве его фрагмента в отдельный массив меньшего размера. Т.е. ФБ производит простое копирование элементов входного массива в выходной.



Вход «Начальный элемент» задаёт индекс (с 0) элемента входного массива, начиная с которого ФБ будет копировать элементы.

Вход «Количество элементов» указывает ФБ, сколько элементов нужно скопировать.

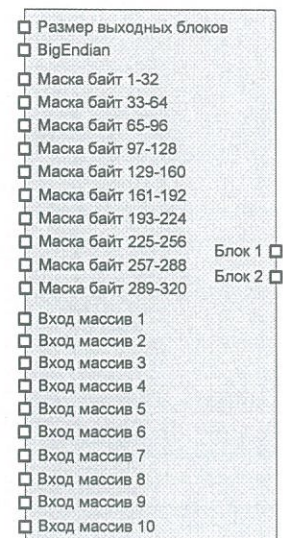
Входной и выходной массивы могут быть любого типа, но типы их должны быть одинаковыми.

ФБ №54 «Компоновщик данных в блоки»

ФБ «Компоновщик данных в блоки» служит для объединения нескольких разнотипных данных (возможно, нескольких массивов) в блоки данных (т.е. в массивы с элементами типа uint8). При этом ФБ позволяет задать маску включаемых входных байт в выходной блок. Это делается с помощью 10 входов «Маска байт», имеющих тип uint32 и задающих флаги включения в сумме для 320 входов. Если масками флагов включения задаётся невключение определённых входов, то эти элементы в выходные блоки не попадают, а все следующие данные в выходных блоках будут смещены вверх.

Если на вход ФБ подаются типы данных, состоящие из нескольких байт, то порядок байт в этих многобайтных типах данных можно выбрать. Для порядка от старшего байта к младшему (Big Endian) необходимо на вход «BigEndian» подать «1», для порядка от младшего к старшему (Little Endian) – «0».

Вход «Размер входных блоков» указывает ФБ, какого размера должны быть сформированы выходные блоки, т.е. если очередной вход не помещается в блок, то ФБ оставляет оставшиеся в блоке байты нулевыми и переходит к формированию следующего блока данных.



ФБ №60 «Преобразователь в массив»

ФБ нужен для объединения нескольких однотипных входных значений в выходной массив.

Максимальное кол-во элементов массива = 256.



ФБ №65 «Преобразователь массива uint16->массив uint8»

ФБ делит каждый элемент входного массива на 2 байта: младший и старший и собирает из этих байт выходной массив. Порядок следования байт в каждой паре можно изменить с помощью входа «BigEndian» («0» - сначала младший, затем старший байт, «1» - наоборот).

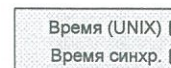


Например, при считывании по протоколу MODBUS (ФБ «MODBUS Master чтение») блока данных функциями 03h или 04h, зачастую MODBUS-устройство упаковывает блок данных в регистры подряд, образуя порядок байт в регистрах LittleEndian, хотя протокол MODBUS использует BigEndian. В этом случае полученный массив 16-битных регистров необходимо разбить на байты, поменяв местами байты в каждой паре. Лог. «1» на входе «BigEndian» заставляет ФБ при сборке выходного массива менять местами байты элементов входного массива. Полученный выходной блок (массив байт) будет иметь правильный порядок байт.

4.7 ФБ работы со временем и счётчики

ФБ №23 «Системное время в UNIX-формате»

ФБ выдаёт на выход «Время (UNIX)» системное время контроллера, которое он берёт из микросхемы часов реального времени. Формат UNIX – uint32 число равное количеству секунд, прошедшему с 1 января 1970 года.



Также ФБ имеет битовый выход «Время синхронизировано», «0» на котором означает что с момента включения контроллера, его системное время ещё ни разу не было синхронизировано командами от ПО верхнего уровня либо ФБ «Синхронизатор времени».

ФБ №44 «Синхронизатор времени»

ФБ нужен для синхронизации системного времени контроллера через адресное пространство ОЗУ ФБ. Для синхронизации необходимо подать на вход «Время (UNIX)» текущее время, предварительно время в неUNIX формате может быть сконвертировано ФБ-конвертером времени. Далее следует на битовый вход «Синхронизировать» подать положительный импульс. ФБ инициирует запись в микросхему часов реального времени и установит системный флаг «время синхронизировано».



ФБ №45 «Конвертер времени из UNIX-формата» и ФБ №46 «Конвертер времени в UNIX-формат»

ФБ необходимы для конвертирования времени между 2 представлениями (UNIX и отдельными значениями текущего дня, месяца, года (год отсчитывается от 1980 г.), часа, минуты и секунды).



ФБ №58 «Первая секунда исполнения конфигурации»

ФБ имеет 1 битовый выход, который = «1» в первую секунду исполнения конфигурации, далее = «0».

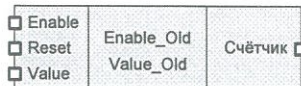


ФБ №59 «Секунд с начала исполнения конфигурации»

ФБ имеет 1 time32/uint32 выход, на который выдаёт кол-во секунд, прошедших с момента начала исполнения конфигурации.



ФБ №37 «Счётчик»



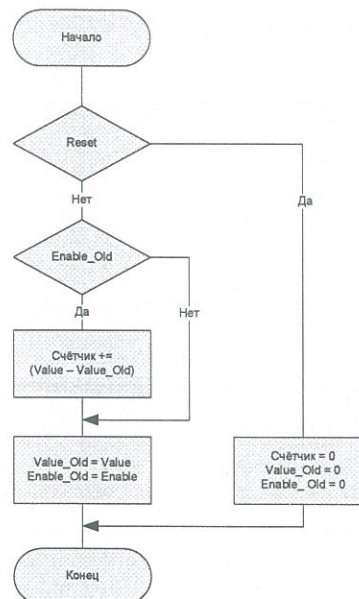
На вход «Value» ФБ «Счётчик» может быть подано любое увеличивающееся значение вещественного или целочисленного типа. Например, текущее время в формате UNIX, различные внешние или внутренние программные счётчики.

Счёт ведётся пока вход «Enable» = «1». Когда «Enable» = «0», счёт приостанавливается. Вход «Reset» сбрасывает значение счётчика.

Переменные «Enable_Old» и «Value_Old» используются для запоминания прошлых значений входов «Enable» и «Value».

Требования к типам входов и выходов и их количеству:

- Входы «Enable» и «Reset», а также переменная «Enable_Old» - дискретные.
- Вход «Value», переменная «Value_Old» и выход «Счётчик» могут быть вещественными (float32) или целочисленными (uint8/16/32), но обязательно должны быть одного типа.



ФБ №28 «Длительность импульса (в тактах)»

Требования к типам входов и выходов и их количеству:



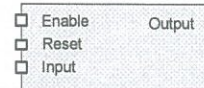
- Вход один, его тип – строго bit.
- Выход один – целочисленный. На выход выдаётся длительность последнего входного импульса.
- Кроме того есть 1 внутренняя переменная ФБ: «var_Dlit» - накапливаемая длительность.
- Значение выхода и «var_Dlit» - строго целочисленные (uint8/16/32). При выборе типов данных нужно избегать переполнения переменных.

ФБ № 89 «Счетчик входных импульсов»

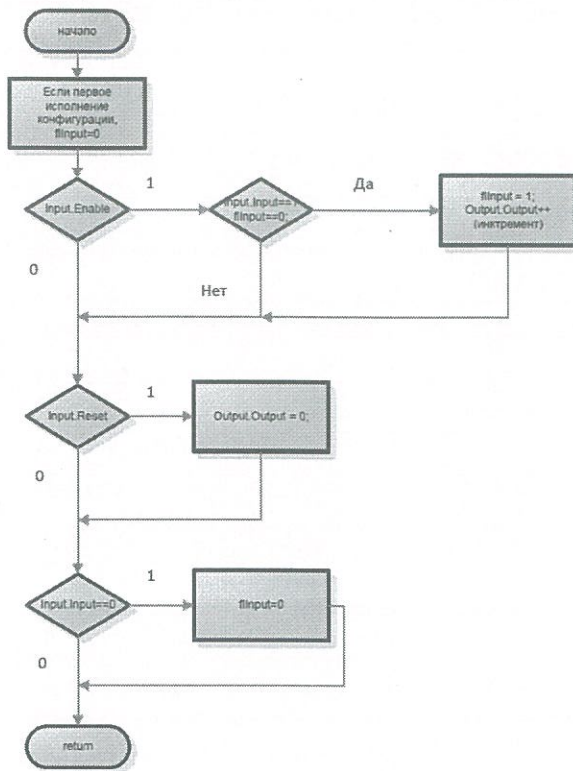
ФБ инкрементирует счетчик по каждому возрастающему фронту сигнала на входе Input и выводит на регистр выхода Output.

Список входов/выходов

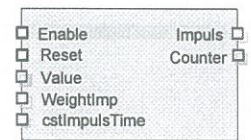
DI	AO
Разрешение тактирования (Enable- bit)	Счетчик входных тактов (Output – uint8,uint16,uint32)
Обнуление счетчика (Reset – Bit)	
Вход для тактового сигнала (Input – Bit)	



Блок-схема ФБ «Счетчик входных импульсов»:



ФБ №81 «Весовой импульс с счетчика»



ФБ №61 «Первое исполнение ФБ логики»

Выход Out – признак первого такта исполнения ФБ.



ФБ №24 «Системное время: секунд от начала минуты»

ФБ подает на выход число секунд от начала минуты.



ФБ №26 «Системное время: секунд от начала суток»

ФБ подает на выход число секунд от начала суток.



ФБ №25 «Системное время: Секунд от начала часа»

ФБ подает на выход число секунд от начала часа.



ФБ №78 «Счетчик с ограничением по максимальному приращению»

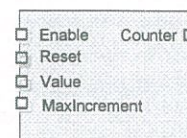
На вход «Value» может быть подано любое увеличивающееся значение вещественного или целочисленного типа. Например, текущее время в формате UNIX, различные внешние или внутренние программные счётчики.

Счёт ведётся пока вход «Enable» = «1», а значение выхода «Counter» меньше либо равно значению входа «Maxincrement». Когда «Enable» = «0» либо значение выхода «Counter» достигло значения входа «Maxincrement», счёт приостанавливается. Вход «Reset» сбрасывает значение счётчика.

Переменные «Enable_Old» и «Value_Old» используются для запоминания прошлых значений входов «Enable» и «Value».

Требования к типам входов и выходов и их количеству:

- Входы «Enable» и «Reset», а также переменная «Enable_Old» - дискретные.
- Вход «Value», переменная «Value_Old» и выход «Counter» могут быть вещественными (float32) или целочисленными (uint8/16/32), но обязательно должны быть одного типа.



ФБ №27 «Таймер: секунд от начала периода»

Если на вход Reset подается 1, на выход PeriodTime выводится 0. Как только на вход Reset подается 0, на выход PeriodTime выводится количество секунд до того как на вход Reset поступит 1.



4.8 ФБ-архиваторы событий

МегаФБ32 для архивации событий использует записи R7 длиной 64 байта типов 16, 17 и 18.

Тип 16 – архив замера ГЗУ-массомер.

Тип 17 был реализован для совместимости с существующими разборщиками архивов R5 в ПО верхнего уровня и представляет собой записи R5 стандартных подтипов (ГЗУ, БГ, протокол работы объекта и т.п.) упакованные внутри записей R7.

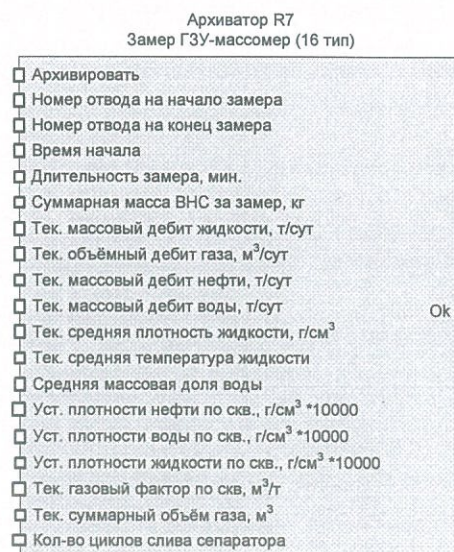
Тип 18 – свободный, позволяет формировать задать тело архива в справочнике конфигурирования ФБ32.

ФБ №77 «Архиватор R7 Замер ГЗУ-массомер (тип 16)»

ФБ создаёт новую архивную запись в каждом цикле исполнения ФБ если битовый вход «Архивировать» = «1». При использовании ФБ в конфигурациях необходимо одновременно с положительным импульсом на входе «Архивировать» подать соответствующие данные на остальные входы ФБ для заполнения записи. После добавления архивной записи в очередь архиватор ФБ выставляет «1» на своём выходе «Ok» на 1 цикл исполнения ФБ. Этот импульс можно использовать, например, для сброса счётчиков и начала нового замера.

Требования к типам входов и выходов и их количеству:

- Вход «Архивировать» - строго bit. Иницирует создание архивной записи.
- Входы «Номер отвода на начало замера» и «Номер отвода на конец замера» типа uint8.
- Вход «Время начала замера» - время начала замера в формате UNIX.
- Вход «Длительность мин.» - длительность замера



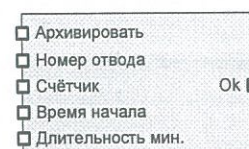
в минутах.

- Входы «Суммарная масса ВНС», «Текущий массовый дебит жидкости», «Текущий объёмный дебит газа», «Текущий массовый дебит нефти», «Текущий массовый дебит воды», «Текущая средняя плотность жидкости», «Текущая средняя температура жидкости», «Средняя массовая доля воды», «Текущий газовый фактор по скважине», «Текущий суммарный объём газа» типа Float32.
- Входы «Уставка плотности нефти по скважине», «Уставка плотности воды по скважине», «Уставка плотности жидкости по скважине», «Кол-во циклов слива сепаратора» типа Uint16.
- После добавления архивной записи в очередь архиватора ФБ выставляет «1» на своём выходе «Ок» на 1 цикл исполнения ФБ.

ФБ №47 «Архиватор R7 Замер ГЗУ (тип 17, подтип 1)»

ФБ создаёт новую архивную запись в каждом цикле исполнения ФБ если битовый вход «Архивировать» = «1». При использовании ФБ в конфигурациях необходимо одновременно с положительным импульсом на входе «Архивировать» подать соответствующие данные на остальные входы ФБ для заполнения записи.

После добавления архивной записи в очередь архиватор ФБ выставляет «1» на своём выходе «Ок» на 1 цикл исполнения ФБ. Этот импульс можно использовать, например, для сброса счётчиков и начала нового замера.

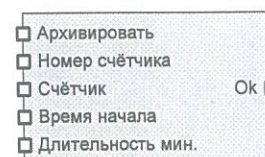


Требования к типам входов и выходов и их количеству:

- Вход «Архивировать» - строго bit. Иницирует создание архивной записи.
- Вход «Номер отвода» тип uint8.
- Вход «Счётчик» типа uint32.
- Вход «Время начала» - время начала замера в формате UNIX.
- Вход «Длительность мин.» - длительность замера в минутах.
- Выход «Ок» типа bit.

ФБ №48 «Архиватор R7 Замер БГ (тип 17, подтип 2)»

ФБ создаёт новую архивную запись в каждом цикле исполнения ФБ если битовый вход «Архивировать» = «1». При использовании ФБ в конфигурациях необходимо одновременно с положительным импульсом на входе «Архивировать» подать соответствующие данные на остальные входы ФБ для заполнения записи. После добавления архивной записи в очередь архиватор ФБ выставляет «1» на своём выходе «Ок» на 1 цикл исполнения ФБ. Этот импульс можно использовать, например, для сброса счётчиков и начала нового замера.



Требования к типам входов и выходов и их количеству:

- Вход «Архивировать» - строго bit. Иницирует создание архивной записи.
- Вход «Номер счётчика» тип uint8.
- Вход «Счётчик» типа uint32.
- Вход «Время начала» - время начала замера в формате UNIX.
- Вход «Длительность мин.» - длительность замера в минутах.
- Выход «Ок» типа bit.

ФБ №49 «Архиватор R7 Изменение состояния объекта (тип 17, подтип 4)»

ФБ создаёт новую архивную запись при 2 условиях: ФБ активна (т.е. вход «Enable» = «1») и на входе «Вход» было изменение относительно его значения в прошлом такте исполнения.



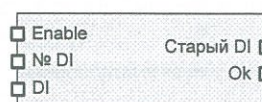
На вход «Вход» необходимо завести до 16 бит состояния, изменение которых нужно архивировать. Это могут быть 2 байта флагов работы объекта Sost и Sost1 (по аналогии с Мега09).

Требования к типам входов и выходов и их количеству:

- Вход «Архивировать» - строго bit, активизирует в ФБ отслеживание изменений входа «Вход».
- Выход «Старый вход» нужен для запоминания прошлого состояния входа «Вход».
- Выход «Ok» типа bit. На этом выходе формируется положительный импульс после добавления новой записи в очередь.

ФБ №50 «Архиватор R7 Изменение DI (тип 17, подтип 5)»

ФБ создаёт новую архивную запись при 2 условиях: ФБ активна (т.е. вход «Enable» = «1») и на битовом входе «DI» было изменение относительно его значения в прошлом такте исполнения.

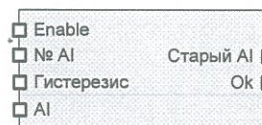


Требования к типам входов и выходов и их количеству:

- Вход «Архивировать» - строго bit, активизирует в ФБ отслеживание изменений входа «DI».
- Вход «№ DI» тип uint8 помещается в тело архивной записи.
- Вход «DI» типа bit.
- Выход «Старый DI» нужен для запоминания прошлого состояния входа «DI»..
- Выход «Ok» типа bit.

ФБ №51 «Архиватор R7 Изменение AI (тип 17, подтип 6)»

ФБ создаёт новую архивную запись при 2 условиях: ФБ активна (т.е. вход «Enable» = «1») и на uint16-входе «AI» было изменение относительно его значения в прошлом такте исполнения более чем на величину, поданную на вход «Гистерезис».

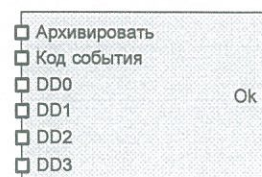


Требования к типам входов и выходов и их количеству:

- Вход «Архивировать» - строго bit, активизирует в ФБ отслеживание изменений входа «DI».
- Вход «№ AI» тип uint8 помещается в тело архивной записи.
- Вход «AI» типа uint16.
- Вход «Гистерезис» типа uint16, задаёт максимальное отклонение входа «AI» за 1 такт, не приводящее к формированию новой архивной записи.
- Выход «Старый AI» нужен для запоминания прошлого состояния входа «DI»..
- Выход «Ok» типа bit.

ФБ №52 «Архиватор R7 Протокол работы объекта (тип 17, подтип 7)»

ФБ создаёт новую архивную запись в каждом цикле исполнения ФБ если битовый вход «Архивировать» = «1». При использовании ФБ в конфигурациях необходимо одновременно с положительным импульсом на входе «Архивировать» подать соответствующие данные на остальные входы ФБ для заполнения записи. После добавления архивной записи в очередь архиватор ФБ выставляет «1» на своём выходе «Ok» на 1 цикл исполнения ФБ. Этот импульс можно использовать, например, для сброса счётчиков и начала нового замера.



Требования к типам входов и выходов и их количеству:

- Вход «Архивировать» - строго bit. Иницирует создание архивной записи.
- Вход «Код события» тип uint8.
- Входы «DD0», «DD1», «DD2», «DD3» типа uint8.
- Выход «Ок» типа bit.

ФБ №30 «Архиватор R7 универсальный (тип 18)»

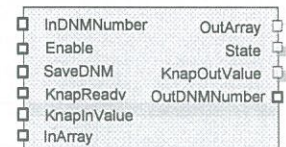
Требования к типам входов и выходов и их количеству:

- Вход «Архивировать» - строго bit. Даёт команду ФБ на архивацию данных, поданных на вход «Данные».
- Вход «Данные[N]» - данные, любого типа, одиночное значение или массив. Может быть несколько разнотипных массивов. Общий размер входа «Данные» не может превышать 55 байт. Это связано с размером тела архива R7 18 типа. Общее кол-во элементов данных не должно превышать 250 шт.
- ФБ группирует данные типа bit в байты для экономии места в теле архива.
- После добавления архивной записи в очередь архиватора ФБ выставляет «1» на своём выходе «Ок» на 1 цикл исполнения ФБ.



ФБ №88 «Архиватор R7 для ДНМ»

ФБ реализует преобразование и архивирование блока ДНМ.



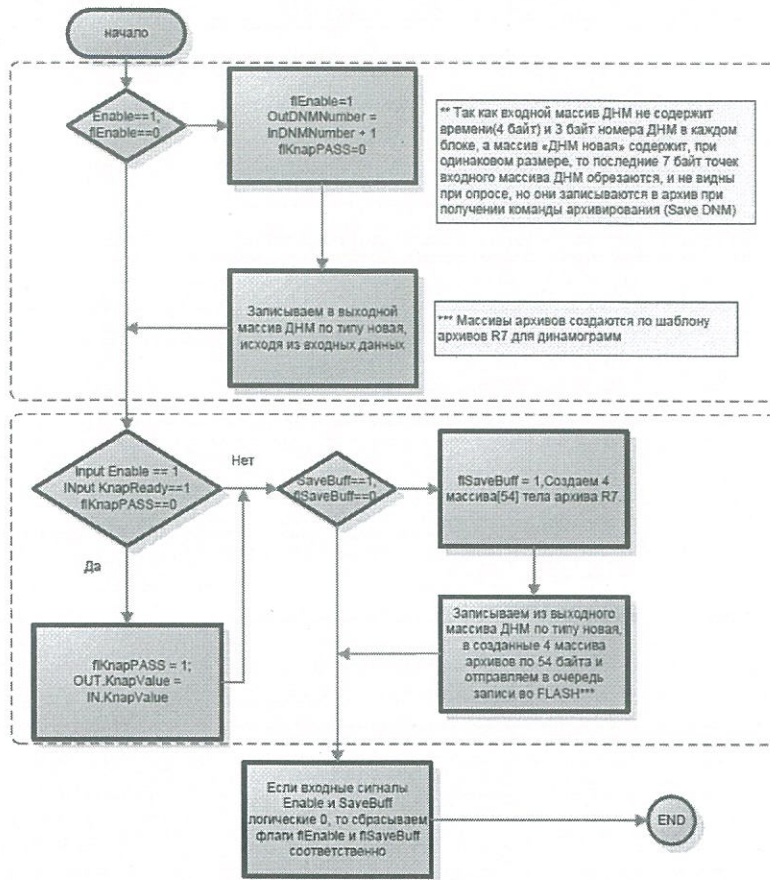
Список входов/выходов

DI	AQ
Команда включения и формирования массива (Enable- bit)	Выходной массив данных ДНМ (OutArray[192] – uint8)
Команда архивирования (SaveDNM – Bit)	Регистр состояния State – (uint8)
Команда записи Кнап (KnapReady – Bit)	Кнап массива ДНМ (KnapValue – uint8)*
Номер предыдущей ДНМ (InDNMNumber– uint8)*	Текущий номер ДНМ (OutDNMNumber – uint8)*
Значение Кнап из модуля подсчета Кнап (KnapValue– uint8)	
Входной массив данных ДНМ (InArray[192]– uint8)	

flEnable (0) - флаг прошедшей процедуры записи ДНМ в выходной массив
 flSaveDNM (1) - флаг прошедшей процедуры архивирования
 flKnapPASS (3) - флаг успешного расчета и записи КНАП

* В случае архивирования одной ДНМ выход OutDNMNumber соединяем со входом InDNMNumber своего ФБ. Если необходимо архивировать несколько последних ДНМ, выход OutDNMNumber подводим ко входу InDNMNumber следующего ФБ, закидывая их по кругу (см. стр. примеры).

Блок-схема работы ФБ «Архиватор R7 для динамограмм»:



ФБ №83 «Архиватор R7 Состояние DI КП КТПН»

- CmdArch Ok
- NChan
- AuthorArch
- SostDI
- PrevSostDI
- DurationSostDI
- EventTime

Название	Вход/выход	Описание	Тип данных	Значение по умолчанию
Ok	Выход		BIT	
CmdArch	Вход	Команда на архивацию	BIT	
NChan	Вход	Номер канала	UINT_08	1
AuthorArch	Вход	Инициатор создания архива	UINT_08	1
SostDI	Вход	Текущее состояние DI	UINT_16	
PrevSostDI	Вход	Предыдущее состояние DI	UINT_16	0
DurationSostDI	Вход	Продолжительность предыдущего состояния DI	UINT_32	0
EventTime	Вход	Время возникновения события	UINT_32	

ФБ №93 «Двухчасовка (счетчик-архиватор)»

- CntNum Ok
- Period CounterOut
- AuthorArch OldCounterIn
- CounterIn StartTime

Название	Вход/Выход	Описание	Тип данных
Ok	Выход	На 1 такт после архивирования	BIT
CounterOut	Выход	Счетчик за текущий интервал	UINT_16
OldCounterIn	Выход	Значение входа в прошлый такт	UINT_32
StartTime	Выход	Время начала замера (UnixTime)	
CntNum	Вход	Номер счетчика в архив	UINT_08
Period	Вход	Периодичность архивирования	UINT_16
CounterIn	Вход	Увеличивающееся значение	UINT_32

ФБ №104 «Архивирование сетевых пакетов (СП)»



Название	Вход/Выход	Описание	Тип данных
Ok	Выход	Прошла команда на архивирование	BIT
Enable	Вход	Разрешение архивирования	BIT
Адрес (GUID)	Вход	Адрес который указали в конфигураторе	UINT_16

4.9 ФБ работы с I2C-модулями

ФБ №39 «Присутствие I2C-модуля на шине»

ФБ выдаёт «1» на дискретный выход «Подключен» если модуль с i2c-адресом, поданным на вход «i2c-адрес модуля» присутствует на шине и с ним идёт периодический обмен.



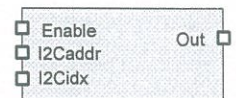
Адрес, подаваемый на вход «i2c-адрес модуля» должен быть 8-битным чётным.

Перечень адресов (с младшим битом 0) существующих модулей приведён в приложении Б.

ФБ №5 «Чтение из I2C-модуля безусловное»

Требования к типам входов и выходов и их количеству:

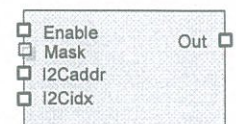
- Вход Enable - строго bit. Разрешает ФБ инициировать чтение из модуля.
- Входы «I2Caddr» и «I2Cidx» - целочисленные
- «I2Cidx» не более 256 (байт)
- Выход «Out» – выходные данные, любого типа, одиночное значение или массив. Может быть несколько разнотипных массивов. Общее кол-во элементов не должно превышать 250 шт.



ФБ №4 «Чтение из I2C-модуля условное»

Требования к типам входов и выходов и их количеству:

- Вход Enable - строго bit. Разрешает ФБ инициировать чтение из модуля.

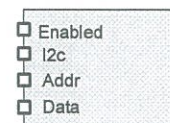


- Вход Mask – маска для чтения
- Входы «I2Caddr» и «I2Cidx» - целочисленные
- «I2Cidx» не более 256 (байт).
- Выход «Out» - выходные данные, любого типа, одиночное значение или массив. Может быть несколько разнотипных массивов. Общее кол-во элементов не должно превышать 250 шт.

ФБ №9 «Запись в I2C-модуль»

Требования к типам входов и выходов и их количеству:

- Вход Enable - строго bit. Разрешает ФБ инициировать запись в модуль.
- Входы «I2Caddr» и «I2Cidx» - целочисленные.
- «Кол-во байт» не более 256 (байт).



4.10 ФБ MODBUS-Master

Два ФБ реализуют протокол Modbus RTU (master, т.е. контроллер инициирует обмен с другими Modbus-устройствами - слейвами). ФБ работают с одним из каналов связи модуля CPU32 или одного из коммуникационных модулей (модемы, модули расширения портов).

При размещении в конфигурации нескольких ФБ работающих с одним каналом связи, все эти ФБ получают управление по очереди, т.е. за одно исполнение конфигурации только 1 из этих ФБ сможет инициировать обмен. При следующем исполнении конфигурации, эта ФБ обработает ответ (при условии, что он уже получен и тайм-аут ответа не истёк) и в этом же цикле следующая ФБ, работающая с этим каналом, сможет получить доступ к каналу связи.

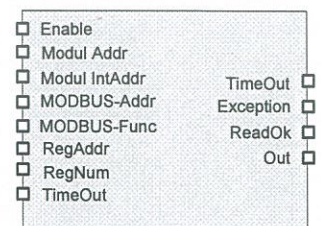
Для идентификации канала связи используется i2c-адрес модуля и адрес буфера канала связи внутри i2c модуля. Поскольку порты модуля CPU не принадлежат i2c-модулям, им присвоены виртуальные адреса:

- для порта COM1 необходимо указать «Адрес i2c-модуля»=«0» и «внутренний адрес»=«0»;
- для порта COM2 необходимо указать «Адрес i2c-модуля»=«1» и «внутренний адрес»=«0».

ФБ №33 «MODBUS RTU Master: Чтение - функция 1/2/3/4»

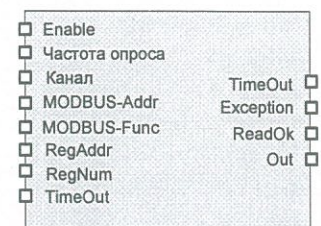
Требования к типам входов и выходов и их количеству:

- Вход Enable - строго bit. Разрешает ФБ отправлять пакеты.
- Входы «адрес i2c-модуля», «внутренний адрес» и «кол-во байт» - целочисленные.
- «MODBUS-адрес» – RTU адрес слейва, от 1 до 247.
- «MODBUS-функция» - для данного ФБ допустимы значения «1» и «2» для чтения битовых данных, «3» и «4» для чтения 16-битных регистров. При чтении функциями 1 и 2 выход «Данные» должен быть строго битового типа. При чтении функциями 3 и 4, выход «Данные» может быть любого типа. Можно использовать несколько элементов разного типа, а также несколько массивов разного типа. Важно при этом чтобы вход «кол-во регистров» соответствовал набору типов и кол-ву элементов выхода «Данные».
- Общее кол-во элементов выхода «Данные» не должно превышать 253.
- Максимальный TimeOut ответа = 6350 мс.
- Внутренняя переменная «№ пакета» обязательна. Нужна для идентификации пакета в очереди пакетов и корректного отработывания истечения тайм-аута ответа и переключения нескольких интерфейсных ФБ на одном канале связи, её тип uint8.



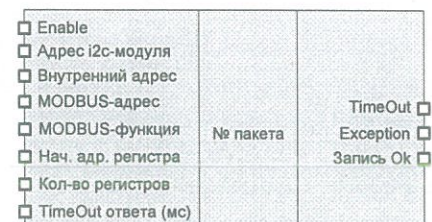
«MODBUS RTU Master: Чтение Мега12 - функция 1/2/3/4»

- Канал - 0-COMExt,1-COM0,2-БР,3-ЛТМ,4-ДР,5-COM1,6-GPRS,7-ComIntBus,8-Eth,9-SMS (по умолчанию 5)
- Частота опроса - В тактах исполнения кфг



ФБ №34 «MODBUS RTU Master: Запись - функция 5/6/15/16»

Требования к типам входов и выходов и их количеству:



- Вход Enable - строго bit. Разрешает ФБ отправлять пакеты.
- Входы «адрес i2c-модуля», «внутренний адрес» и «кол-во байт» - целочисленные.
- «MODBUS-адрес» – RTU адрес слейва, от 1 до 247.
- «MODBUS-функция» - для данного ФБ допустимы значения «5» и «15» для записи битовых данных, «6» и «16» для записи 16-битных регистров. При записи функциями 5 и 15 вход «Данные» должен быть строго битового типа. При записи функциями 6 и 16, вход «Данные» может быть любого типа. Можно использовать несколько элементов разного типа, а также несколько массивов разного типа. Важно при этом, чтобы вход «кол-во регистров» соответствовал набору типов и кол-ву элементов входа «Данные».
- Общее кол-во элементов входа «Данные» не должно превышать 248.
- Максимальный TimeOut ответа = 6350 мс.
- Внутренняя переменная «№ пакета» обязательна. Она нужна для идентификации пакета в очереди пакетов и корректного отрабатывания истечения тайм-аута ответа и переключения нескольких интерфейсных ФБ на одном канале связи, её тип uint8.

«MODBUS RTU Master: Запись - функция 5/6/15/16»

- Канал - 0-COMExt,1-COM0,2-БР,3-ЛТМ,4-ДР,5-COM1,6-GPRS,7-ComIntBus,8-Eth,9-SMS (по умолчанию 5)
- Частота опроса - В тактах исполнения кфг

<input type="checkbox"/> Enable	№ пакета	TimeOut Exception Запись Ok
<input type="checkbox"/> Частота опроса		
<input type="checkbox"/> Внутренний адрес		
<input type="checkbox"/> MODBUS адрес		
<input type="checkbox"/> MODBUS функция		
<input type="checkbox"/> Нач адр регистра		
<input type="checkbox"/> Кол во регистров		
<input type="checkbox"/> TimeOut ответа (мс)		
<input type="checkbox"/> Данные [N]		

ФБ №110 MODBUS-Модем

ФБ №110 настраивает в PacketManager маршрутизацию Modbus пакетов. Функциональный блок записывает новый RTM маршрут для указанного Modbus адреса, в дальнейшем приходящий пакет разбирается и если он не совпадает с собственным адресом Modbus, но совпадает с адресом указанным в ФБ на входе ModbusAddress формируется пакет RTM_MW куда вкладывается Modbus пакет, при достижении пакета последнего адресата (последнее ненулевое значение NodeAddress) если Modbus адрес вложенный в пакет не совпал со своим то контроллер распаковывает Modbus и отправляет в канал указанный в NodeChannel.

- Маршрут в пакете формируется согласно входам «NodeAddress», «NodeChannel», все ненулевые узлы становятся ретрансляторами.
- Вход ModbusAddress может быть массивом до 50 элементов для того чтобы можно было опросить несколько устройств подключенных на одну шину RS485(channel 5).
- Вход ChanelFrom откуда приходят пакеты для ретрансляции, если указан канал 8 (Ethernet) но пакет пришел по каналу 1 (RS232), пакет не будет ретранслироваться даже если есть совпадение адресов.
- Вход ChanelTo указывает куда дальше ретранслируется пакет.

Если адрес первого ретранслятора(NodeAddress1) равен "0", но Modbus адрес совпал с одним из указанных во входе ModbusAddress создается маршрут в котором адрес первого ретранслятора равен Modbus адресу в пакете.

Если адрес первого узла «1», то формируется широковещательный маршрут, в котором адрес первого узла заменяется адресом в Modbus пакете. Данный маршрут будет задействован только в случае если в других не было совпадения Modbus адреса.

Для ретрансляции Modbus пакета без промежуточного RTM устройства, в NodeAddress1 необходимо записать 4095, тогда Modbus пакет будет ретранслироваться сразу в другой Modbus на другом канале.

<input type="checkbox"/> Enable	TableNoteNum	Status
<input type="checkbox"/> Chanel From		
<input type="checkbox"/> Chanel To		
<input type="checkbox"/> Node address 1		
<input type="checkbox"/> Node Chanel 1		
<input type="checkbox"/> Node address 2		
<input type="checkbox"/> Node Chanel 2		
<input type="checkbox"/> Node address 3		
<input type="checkbox"/> Node Chanel 3		
<input type="checkbox"/> Node address 4		
<input type="checkbox"/> Node Chanel 4		
<input type="checkbox"/> Node address 5		
<input type="checkbox"/> Node Chanel 5		
<input type="checkbox"/> ModbusAddress		

На входе ModbusAddress можно указать широковещательный адрес 251, тогда все Modbus пакеты будут ретранслироваться по указанному маршруту.

Номера каналов:

- 0 – ближнее радио(внешний модуль дальность до 5000 км);
- 1 - COM1 RS232;
- 2 – chNearRadio ближнее радио(до 2500 км);
- 3 - chWire was not make!!!!;
- 4 - chFarRadio дальнее радио(motorola or kenwood);
- 5 - COM2 RS485;
- 6 - chGSMGPRS was not make!!!!;
- 7 – chCOMIntBUS внутренняя шина(по умолчанию скорость 2 мбит/с);
- 8 - chEthernet Ethernet(IP).

Список входов/выходов

Входы	Выход
Enable (bit, 1 – использовать маршрут)	Status
ChanelFrom	TableNoteNumm(номер в таблице маршрутов)
Chanelto	
NodeAddress1	
NodeChanel1	
NodeAddress2	
NodeChanel2	Переменные
NodeAddress3	
NodeChanel3	
NodeAddress4	
NodeChanel4	
NodeAddress5	
NodeChanel15	
ModbusAddress	

Параметры Status
0 – маршрут был добавлен

ФБ №109 MODBUS TCP-Master: Запись

1	Enable	Status
192	ip4	Error count
168	ip3	Ok count
1	ip2	ConnectionTimer
232	ip1	IntervalTimer
502	port	PassNumm
1500	ConnectionTime	PassCnt
0	IntervalTime	Ok
3	Mdb address	
6	Mdb function	
3	Reg address	
2	Reg numm	
	Input	

ФБ осуществляет запись по mdbtcp протоколу в указанный IP4 – IP1 адрес с периодичностью «IntervalTime»(мс.), тайм аут на соединение указывается в переменной «ConnectionTime». Если устройство не отвечает то ФБ начинает записывать реже до 25 «Intervaltime».

Входа:

- “Enable” – включение/выключение ФБ;
- “Mdb_address” – адрес Slave устройства;
- “Mdb_function” - функция записи, поддерживает«5,6,15,16»;

”Reg_address” – регистр с которого начинаем запись;

“Reg_numm” – количество регистров на запись;

“Input” – данные на запись(возможные типы bit,uint16,uint32).

Выхода:

“Status” - статус выполнения;

WORKING 0 фб в работе (заняло канал TCP)

CONNECTION_STATE 1 было установлено соединение

PASSED 2 фб освободило канал TCP

“Err_count” – счетчик неудачных пакетов;

“ok_count” – счетчик удачных пакетов;

“ConectionTimer” – таймер ожидания ответа;

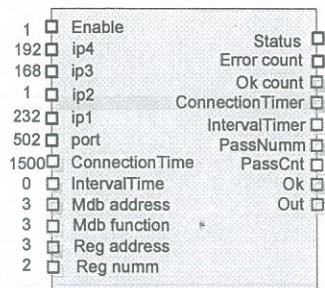
“IntervalTimer” – таймер замера между передачами;

“PassNumm” – сколько раз пропускаем очередь, данное значение нарастает после неудачных пакетов;

“PasCnt” – счетчик очередей которых осталось пропустить;

“Ok” – выдает импульс в случае удачной записи.

ФБ №108 MODBUS TCP-Master: Чтение



ФБ осуществляет чтение по mdbtcp протоколу с указанного IP4 – IP1 адреса с периодичностью «IntervalTime»(мс.), тайм аут на соединение указывается в переменной «ConnectionTime». Если устройство не отвечает то ФБ начинает опрашивать реже до 25 «Intervaltime».

Входа:

“Enable” – включение/выключение ФБ;

“Mdb_address” – адрес Slave устройства;

“Mdb_function” - функция чтения, поддерживает типы «1/2/3/4»;

”Reg_address” – регистр с которого начинаем читать;

“Reg_numm” – количество регистров для чтения;

Выхода:

“Status” - статус выполнения;

WORKING 0 фб в работе (заняло канал TCP)

CONNECTION_STATE 1 было установлено соединение

PASSED 2 фб освободило канал TCP

“Err_count” – счетчик неудачных пакетов;

“ok_count” – счетчик удачных пакетов;

- “ConectionTimer” – таймер ожидания ответа;
- “IntervalTimer” – таймер замера между передачами;
- “PassNumm” – сколько раз пропускаем очередь, данное значение нарастает после неудачных пакетов;
- “PasCnt” – счетчик очередей которых осталось пропустить;
- “Ok” – выдает импульс в случае удачной записи.
- “Output” – данные прочитанные из устройства(возможные типы bit,uint8,uint16,uint32).

4.11 ФБ RTM-Master

При задании маршрута следования RTM64-пакета через цепочку ретрансляторов, необходимо подать на входы RTM64-Master ФБ адреса всех ретрансляторов и виды связи, в которые они должны ретранслировать пакет данного ФБ (см. Приложение В. Виды связи протокола RTM64).

ФБ №35 «RTM64var Master»

Требования к типам входов и выходов и их количеству:

- Вход Enable - строго bit. Разрешает ФБ отправлять пакеты.
- Вход «CRC16» - дискретный. «1» указывает, что контрольную сумму в исходящем пакете следует рассчитывать по алгоритму CRC16. В противном случае, считается как простая сумма байт пакета. Тип контрольной суммы ответного пакета ФБ определит автоматически.
- Входы «адрес i2c-модуля», «внутренний адрес» и «кол-во байт» - целочисленные.
- Входы «RTM-адрес N» - адреса ретрансляторов и получателя пакета. Допустимы адреса в диапазоне от 1 до 4095, кроме адреса 126. Тип входа – uint16.
- Входы «Вид связи N» - значение (0x10/0x20/0x30/0x40/0x50/0x60 – см. Приложение В. Виды связи протокола RTM64), обозначающее вид связи, который должен быть использован устройством при ретрансляции пакета. У получателя пакета вход «Вид связи» должен быть равен «0». Тип входа – uint8.
- Максимальный TimeOut ответа = 6350 мс.
- Внутренняя переменная «№ пакета» обязательна. Она нужна для идентификации пакета в очереди пакетов и корректного отрабатывания истечения тайм-аута ответа и переключения нескольких интерфейсных ФБ на одном канале связи, её тип uint8.

<input type="checkbox"/> Enable <input type="checkbox"/> CRC16 <input type="checkbox"/> Адрес i2c-модуля <input type="checkbox"/> Внутренний адрес <input type="checkbox"/> RTM-адрес 1 <input type="checkbox"/> Вид связи 1 <input type="checkbox"/> RTM-адрес 2 <input type="checkbox"/> Вид связи 2 <input type="checkbox"/> RTM-адрес 3 <input type="checkbox"/> Вид связи 3 <input type="checkbox"/> RTM-адрес 4 <input type="checkbox"/> Вид связи 4 <input type="checkbox"/> RTM-команда <input type="checkbox"/> Данные туда <input type="checkbox"/> TimeOut ответа (мс) <input type="checkbox"/> Данные туда [N]	№ пакета	<input type="checkbox"/> TimeOut <input type="checkbox"/> Exception <input type="checkbox"/> Чтение Ok <input type="checkbox"/> Данные оттуда <input type="checkbox"/> Данные оттуда [N]
-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

ФБ №36 «RTM64 Master»

Требования к типам входов и выходов и их количеству:

- Вход Enable - строго bit. Разрешает ФБ отправлять пакеты.
- Входы «адрес i2c-модуля», «внутренний адрес» и «кол-во байт» - целочисленные.
- Входы «RTM-адрес N» и «RTM-адрес «кому»» - адреса ретрансляторов и получателя пакета. До-

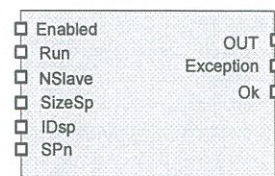
<input type="checkbox"/> Enable <input type="checkbox"/> Адрес i2c-модуля <input type="checkbox"/> Внутренний адрес <input type="checkbox"/> RTM-адрес 1 <input type="checkbox"/> Вид связи 1 <input type="checkbox"/> RTM-адрес 2 <input type="checkbox"/> Вид связи 2 <input type="checkbox"/> RTM-адрес 3 <input type="checkbox"/> Вид связи 3 <input type="checkbox"/> RTM-адрес «кому» <input type="checkbox"/> Вид связи «кому» <input type="checkbox"/> RTM-команда TX <input type="checkbox"/> RTM-команда RX <input type="checkbox"/> Данные туда <input type="checkbox"/> TimeOut ответа (мс) <input type="checkbox"/> Данные туда [N]	№ пакета	<input type="checkbox"/> TimeOut <input type="checkbox"/> Exception <input type="checkbox"/> Чтение Ok <input type="checkbox"/> Данные оттуда <input type="checkbox"/> Данные оттуда [N]
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

пустимы адреса в диапазоне от 1 до 4095, кроме адреса 126. Тип входа – int16.

- Входы «Вид связи N» и «Вид связи «кому»» - значение (0x10/0x20/0x30/0x40/0x50/0x60 – см. Приложение В. Виды связи протокола RTM64), обозначающее вид связи, который должен быть использован устройством при ретрансляции пакета. У получателя пакета вход «Вид связи» равен «0». Тип входа – uint8.
- Максимальный TimeOut ответа = 6350 мс.
- Внутренняя переменная «№ пакета» обязательна. Она нужна для идентификации пакета в очереди пакетов и корректного отрабатывания истечения тайм-аута ответа и переключения нескольких интерфейсных ФБ на одном канале связи, её тип uint8.

ФБ №105 «Запись сетевых переменных»

ФБ реализует межмодульный обмен данными между контроллерами по протоколу RTM-MW, а так же при наличии этой ФБ в конфигурации контроллера с настроенными некоторыми параметрами делает этот контроллер Мастером канала и запускает процесс регистрации подчиненных устройств слейвов.



- Вход Enabled - строго bit. Разрешает работу ФБ. Если Enabled = 0 то ФБ только передает свое значение входа Run в выход Run, для того чтобы при последовательном соединении нескольких ФБ процесс обмена не прерывался если у одной из ФБ вход Enabled = 0.
- Вход Run - строго bit. Вход Триггерный. При изменении состояния запускает процесс обмена данными – отправка запроса контроллеру. После отправки запроса ФБ ожидает ответа от контроллера некоторое установленное время (таймаут) для каждого канала таймаут установлен разный. Если был получен ответ или ответ за отведенное время не пришел логика ФБ считается выполненной и выход Run = 1, следовательно, на вход Run должен приходиться сигнал с выхода Run предыдущей ФБ (при последовательном соединении нескольких ФБ) или с выхода Run текущей ФБ. Для начала запуска обмена данными на вход Run один раз нужно подать 1, дальше эта 1 на входе будет присутствовать с выхода Run текущей ФБ или предыдущей ФБ№103. Соединение нескольких ФБ последовательно будет более правильным, особенно если эти ФБ производят обмен по одному каналу, так как в текущий момент времени происходит обмен данными только с одним из контроллеров на данном канале.
- Вход «Канал» задает номер канала для межмодульного обмена.
- Вход «Количество СП» задает количество запрашиваемых сетевых переменных у контроллера.
- Вход «ID СПx» задает ID сетевой переменной запрашиваемой у контроллера, где x чисто запрашиваемых сетевых переменных.
- Вход Выбор Адрес-1/Слот-0 - строго bit. Задает способ работы ФБ, если установлено «1» то считается что на входе Адрес/Слот записан адрес контроллера формате RTM64. Если «0» то номер слота (подробней описано ниже), контроллер работает в режиме Мастера канала и запускает процесс регистрации подчиненных устройств слейвов.
- Вход «Адрес/Слот» задает адрес контроллера в формате RTM64 или номер слота в зависимости от входа Выбор Адрес-1/Слот-0.
- Выход «Run» - строго bit. Равен «1» когда логика ФБ выполнена и был получен ответ или ответ за отведенное время не пришел. Должен быть соединен с входом Run.
- Выход «Exception» - строго bit. Информировать об ошибке в процессе обмена данными, неверные данные или ответ за время таймаута не пришел. Выход Триггерный.
- Выход «Ok» - строго bit. Информировать об успешном межмодульном обмене данными, когда равен «1».
- Выход Данные. Массив данных контроллера полученный в ходе успешного межмодульного обмена. Может быть в формате Uint8 или Uint16. Количество элементов массива задается самостоятельно путем сложения размера данных каждой запрашиваемой сетевой переменной контроллера.

В этом режиме контроллер является мастером (главным) в канале который указан в ФБ. Главный контроллер занимается регистрацией контроллеров слейвов подключенных по данному каналу, задает слейвам адрес в формате RTM64 и хранит список зарегистрированных слейвов. Каждому зарегистрированному слейву соответствует своя сетевая переменная Slave1... Slave32 в адресном пространстве мастера, в ней храниться мак адрес слейва, црц сетевых переменных, канал и адрес слейва который назначил ему мастер. Процесс регистрации слейвов происходит в следующем порядке:

1. Мастер 1 раз в минуту отправляет специальный пакет (содержащий текущее время мастера) в канал указанный в ФБ.
2. Слейвы которые еще не зарегистрированы у Мастера подключенные к данному каналу принимают пакет, светодиод «Ответ'' у слейва начинает часто мигать. Теперь текущее время слейва такое же как и у мастера.
3. Один раз в минуту в определенный промежуток времени (для каждого типа канала свой) Мастер приостанавливает межмодульный обмен (Время тишины).
4. Во время тишины не зарегистрированные слейвы отправляют запрос регистрации мастеру канала.
5. Мастер на запрос о регистрации отправляет слейву его новый адрес в формате RTM64 и записывает информацию о нем в сетевую переменную Slave1... Slave32. Если пришел запрос о регистрации уже зарегистрированного ранее слейва то ему присвоится тот же адрес с которым он и был ранее зарегистрирован.
6. Светодиод «Ответ'' у слейва перестает мигать.

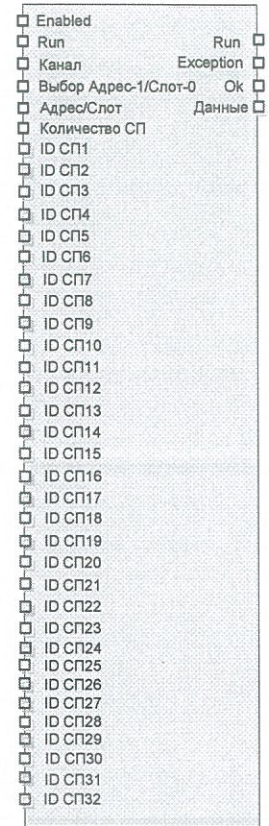
Один слейв может быть зарегистрирован у мастера несколько раз, в зависимости от каналов к которым он подключен к мастеру. Слейв приняв специальный пакет ретранслирует его в другие каналы, тем самым если к слейву были подключены другие контроллеры то они тоже регистрируются у мастера и мастер присваивает им адрес.

Слот – это специальный номер который можно сопоставить с 1 из 32 слейвов. Это соответствие храниться в сетевой переменной MapTable (массив 32-х элементов типа Uint16) у мастера. Первые 2 слота зарезервированы. Допустим мы сопоставим к числу 3 первого слейва (Slave1), то тогда надо записать в сетевой переменной MapTable в 3-м элементе массива число 1, теперь используя в ФБ слот №3 мастер будет осуществлять межмодульный обмен с первым зарегистрированным слейвом.

ФБ №103 «Чтение сетевых переменных»

ФБ реализует возможность записи значений в сетевые переменные контроллеров слейвов (Подробнее в описании ФБ103).

- Вход «Enabled» - строго bit. Разрешает работу ФБ. Если Enabled = 0 то ФБ только передает свое значение входа Run в выход OUT, для того чтобы при последовательном соединении нескольких ФБ процесс обмена не прерывался если у одной из ФБ вход Enabled = 0.
- Вход «Run» - строго bit. Вход Триггерный. При изменении состояния запускает процесс записи значения сетевой переменной контроллера Слейва. После отправки запроса на запись ФБ ожидает ответа от контроллера об успешной записи некоторое установленное время (таймаут) для каждого канала таймаут установлен разный. Если был получен не верный ответ или ответ за отведенное время не пришел логика ФБ считается выполненной и выход OUT = 1.
- Вход «№ Слота». Вход задает число которому сопоставлен слейв (подробней в описании ФБ103).
- Вход «Количество СП». Задает количество элементов(1 элемент = 1 тип данных Uint8,16,32) на запись в зависимости от типа входа - Данные.
- Вход «ID СП». Задает id сетевой переменной, в которую происходит запись данных.
- Вход «Данные». Задается массив данных, который надо записать в сетевую переменную слейва. Может быть в формате Uint8, Uint16, Uint32.
- Выход «OUT» - строго bit. Равен «1» когда логика ФБ выполнена и был получен ответ о успешной записи значения или ответ за отведенное время не пришел.
- Выход «Ехсертион» - строго bit. Информировывает об ошибке в процессе записи данных.
- Выход «Ок» - строго bit. Информировывает об успешном процессе записи данных.

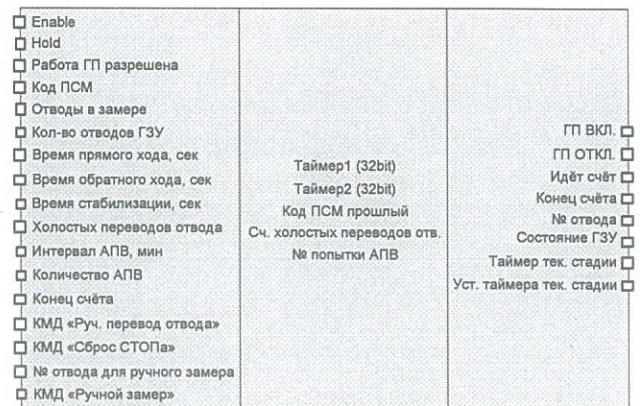


4.12 ФБ, специфические для объектов «ГЗУ», «ГЗУ-Массомер»

ФБ №38 «ГЗУ (алгоритм Башнефть)» v.1

Требования к типам входов/переменных/выходов и их количеству:

- Вход Enable - строго bit. Разрешает работу ФБ.
Примечание: подача «0» на вход «Enable» сбрасывает состояние ГЗУ.
- Вход «Hold» при подаче на него «1» отключает логику работы ФБ. При этом счётчики, таймеры и состояние ФБ «ГЗУ» сохраняют своё прежнее состояние (до подачи «1» на вход «Hold»).
- Вход «Работа ГП разрешена» за-



прещает включение гидропривода. При этом если при работе ФБ ГЗУ не потребуется перевод отвода, будет продолжена работа по алгоритму. Если необходимо включение ГП, ФБ ГЗУ останется в стадии прямого хода, но ГП не включит пока на вход «Работа ГП разрешена» не будет подана «1».

- Вход «Код ПСМ» - целочисленный вход. ФБ использует только 4 младших бита этого значения.
- «Кол-во отводов» - уставка, необходимая для корректного обнаружения неисправности ПСМ (неверный порядок переключения).
- Уставки «Время полного хода», «Время обратного хода», «Время стабилизации» строго целочисленные, задают длительности соответствующих этапов работы ГЗУ в секундах. При счёте этих интервалов используется внутренняя переменная «Таймер1».
- Уставка «Холостых переводов отвода» указывает, сколько раз ГЗУ может переводить отвод не начав замер («холостой перевод»). Для подсчёта кол-ва холостых переводов используется внутренняя переменная «Сч. Холостых переводов отв.».
- Входы «Интервал АПВ» задаёт период времени (в минутах), по истечении которого флаг «Стоп» ГЗУ будет снят и ГЗУ будет перезапущена в режиме циклического замера.
- «Количество АПВ» - сколько раз ГЗУ будет перезапускаться с помощью АПВ до установки флага состояния ГЗУ «Полный стоп». Попытки АПВ считаются с помощью внутренней переменной «№ попытки АПВ».
- Вход «Конец счёта» нужен для определения момента окончания счёта по отводу и начало нового либо начало операции перевода отвода.
- Дискретные входы «КМД руч. перевод отвода», «КМД сброс СТОП» эквивалентны соответствующим командам Ротора.
- Дискретный вход «КМД ручной замер» переводит ГЗУ в режим ручного замера по отводу, номер которого подан на вход «№ отвода для ручного замера».
- Внутренняя переменная «Код ПСМ прошлый» используется для определения неправильного порядка переключения – неисправностей ПСМ и ГП.
- На дискретных выходах «ГП ВКЛ.» и «ГП ОТКЛ.» появляется «1» в начале и в конце прямого хода ГП соответственно. Длительность импульса = периоду исполнения цепочки ФБ.
- Выход «Идёт счёт» = «1» во время замера. Эквивалентен флагу «Zamer» в состоянии ГЗУ.
- На дискретном выходе «Конец счёта» появляется «1» в конце замера (когда вход «Конец счёта» = «1»), а также когда ГЗУ досрочно заканчивает замер, например, при ручном переводе ПСМ во время замера. Длительность импульса = периоду исполнения цепочки ФБ.
- Выход «№ отвода» указывает на текущий отвод в замере.
- Выходы «Таймер текущей стадии» и «Уставка таймера текущей стадии» вытаскивают наружу значение таймера и уставку таймера текущей стадии (прямой, обратный ход, стабилизация перед замером). Указанные значения выводятся в секундах.
- Выход «Состояние ГЗУ» - значение типа uint16, содержащее до 16 флагов состояния:

Номер бита	Значение флага
0	Enabled. «1» если вход «Enable» = «1». Иначе «0», логика ФБ запрещена и состояние ГЗУ сброшено.
1	Holded. «1» если вход «Hold» = «1». Иначе «0», логика ФБ приостановлена. Промышленное состояние ФБ не изменено.
2	GP En. «1» если работа ГП разрешена входом «Работа ГП разрешена».
3	NeedPerevod. «1» если в данный момент по логике работы ФБ должна попытаться переключить отвод.
4	Forward. «1» во время прямого хода ГП.
5	Reverse. «1» во время обратного хода ГП.
6	Stab. «1» во время периода стабилизации.
7	Zamer. «1» во время замера по отводу.
8	Stop. «1» когда состояние ГЗУ – СТОП.

9	FullStop. «1» когда состояние ГЗУ – полный СТОП (СТОП и кончились попытки АПВ). Примечание: Одновременно с флагом FullStop всегда устанавливается флаг Stop.
10	PSMbraked. «1» если причина СТОПа – неисправность ПСМ.
11	GPbraked. «1» если причина СТОПа – неисправность ГП.
12	TryCnt0. «1» если причина СТОПа – превышено максимальное кол-во «холо-стых» переключений.
13	Kluch. Признак ручного переключения ПСМ оператором с помощью ключа. Примечание: бит служит как внутренний флаг ФБ и устанавливается лишь на короткий период до включения обратного хода ГП.
14	Manual. «1» - режим ГЗУ: ручной замер, «0» - режим ГЗУ: циклический замер.
15	ManualCnt. «1» – признак того, что ручной режим включен лишь на 1 замер, по-сле чего будет произведено переключение ФБ ГЗУ в циклический режим.

ФБ №66 «ГЗУ (алгоритм Башнефть)» v.2

ФБ версии v2 отличается от v1 способом пере-ключения в режим ручного замера. В v2 отсутствует вход «КМД «Ручной замер», вместо него добавлен вы-ход «Отвод ручного замера». Для переключения ГЗУ в режим ручного замера достаточно на вход «Отвод руч-ного замера» подать номер отвода для ручного замера. Для переключения ГЗУ в циклический режим можно подать на этот вход «0».

<input type="checkbox"/> Enable <input type="checkbox"/> Hold <input type="checkbox"/> Работа ГП разрешена <input type="checkbox"/> Код ПСМ <input type="checkbox"/> Отводы в замере <input type="checkbox"/> Кол-во отводов ГЗУ <input type="checkbox"/> Время прямого хода, сек <input type="checkbox"/> Время обратного хода, сек <input type="checkbox"/> Время стабилизации, сек <input type="checkbox"/> Холостых переводов отвода <input type="checkbox"/> Интервал АПВ, мин <input type="checkbox"/> Количество АПВ <input type="checkbox"/> Конец счёта <input type="checkbox"/> КМД «Руч. перевод отвода» <input type="checkbox"/> КМД «Сброс СТОПа» <input type="checkbox"/> Отвод ручного замера	Таймер1 (32bit) Таймер2 (32bit) Код ПСМ прошлый Сч. холодных переводов отв. № попытки АПВ	<input type="checkbox"/> ГП Вкл. <input type="checkbox"/> ГП Откл. <input type="checkbox"/> Идёт счёт <input type="checkbox"/> Конец счёта <input type="checkbox"/> № отвода <input type="checkbox"/> Состояние ГЗУ <input type="checkbox"/> Отвод ручного замера <input type="checkbox"/> Таймер тек. стадии <input type="checkbox"/> Уст. таймера тек. стадии
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

ФБ №72 «Счётчик-Массомер»

Требования к типам входов и выходов и их количеству:

- Вход «Enable» - строго bit. Разрешает счёт в ФБ. Появление «0» на этом входе сбрасывает накопленные счётчики ФБ.
- Вход «Hold» - строго bit. Лог. «1» на этом входе приостанавли-вает счёт не обнуляя накопленные счётчики ФБ.
- Вход «Reset» - строго bit. Лог. «1» на этом входе обнуляет накоп-ленные счётчики ФБ.
- Вход «Секунд с начала замера» целого типа – Uint32.
- Входы «Есть массомер жидкости», «Есть мас-сомер газа», «Есть влагомер жидкости» - битовые.

<input type="checkbox"/> Enable <input type="checkbox"/> Hold <input type="checkbox"/> Reset <input type="checkbox"/> Секунд с начала замера <input type="checkbox"/> Есть массомер жидкости <input type="checkbox"/> Счётчик суммарной массы жидкости, кг <input type="checkbox"/> Плотность жидкости, г/см ³ <input type="checkbox"/> Массовый расход жидкости, кг/мин <input type="checkbox"/> Температура жидкости <input type="checkbox"/> Есть массомер газа <input type="checkbox"/> Счётчик суммарной массы газа, кг <input type="checkbox"/> Плотность газа, г/см ³ <input type="checkbox"/> Массовый расход газа, кг/мин <input type="checkbox"/> Температура газа <input type="checkbox"/> Есть влагомер жидкости <input type="checkbox"/> Текущая влажность жидкости, % <input type="checkbox"/> Давление жидкости в коллекторе, Мпа *1000 <input type="checkbox"/> Уст. плотности жидкости по скв., г/см ³ *10000 <input type="checkbox"/> Уст. плотности нефти по скв., г/см ³ *10000 <input type="checkbox"/> Уст. плотности воды по скв., г/см ³ *10000 <input type="checkbox"/> Уст. массовой доли воды <input type="checkbox"/> Уст. газ. фактора по скв., м ³ /т <input type="checkbox"/> Уст. поправочный коэф. по жидкости (Кр), *100 <input type="checkbox"/> Уст. поправочный коэф. по газу (Кг), *100	Счётчик-Массомер Enable_Old Value_Liquid_Old Value_Gas_Old Value_Dryness_Old Value_Density_Liquid_Old Wm_Water Wc_Water Err_Mass_Liquid Err_Mass_Gas Err_Dryness_Liquid Err_Summ_Liquid Err_Summ_Gas Summ_Mass_Water Summ_Temperature_Liquid Summ_Temperature_Gas Summ_CNT Summ_CNT_Gas	<input type="checkbox"/> Тек. суммарная масса жидкости, кг <input type="checkbox"/> Тек. суммарный объём жидкости, м ³ <input type="checkbox"/> Тек. массовый дебит жидкости, т/сут <input type="checkbox"/> Тек. объёмный дебит жидкости, м ³ /сут <input type="checkbox"/> Тек. средняя плотность жидкости, г/см ³ <input type="checkbox"/> Тек. средняя температура жидкости <input type="checkbox"/> Тек. суммарная масса газа, кг <input type="checkbox"/> Тек. суммарный объём газа, м ³ <input type="checkbox"/> Тек. массовый дебит газа, т/сут <input type="checkbox"/> Тек. объёмный дебит газа, м ³ /сут <input type="checkbox"/> Тек. плотность газа, г/см ³ <input type="checkbox"/> Тек. средняя температура газа <input type="checkbox"/> Средняя массовая доля воды <input type="checkbox"/> Средняя объёмная доля воды <input type="checkbox"/> Тек. массовый дебит воды, т/сут <input type="checkbox"/> Тек. объёмный дебит воды, м ³ /сут <input type="checkbox"/> Тек. массовый дебит нефти, т/сут <input type="checkbox"/> Тек. объёмный дебит нефти, м ³ /сут <input type="checkbox"/> Тек. газовый фактор по скв., м ³ /т <input type="checkbox"/> Режим работы
----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

- Входы «Счётчик суммарной массы жидкости», «Плотность жидкости», «Массовый расход жидкости», «Температура жидкости» имеют тип Float32 и заполняются данными, полученными с массомера жидкости.
- Входы «Счётчик суммарной массы газа», «Плотность газа», «Массовый расход газа», «Температура газа» имеют тип Float32 и заполняются данными, полученными с массомера газа.
- Вход «Текущая влажность жидкости» имеет тип Float32 и заполняется данными, полученными с влагомера жидкости.
- Вход «Давление жидкости в коллекторе» имеет тип Uint16.
- Уставки «плотности жидкости по скважине», «плотности нефти по скважине», «плотности воды по скважине», «газового фактора по скважине» имеют тип Uint16.
- Поправочные коэффициенты по жидкости и газу имеют тип Uint8.
- Вход «Уставка массовой доли воды» имеет тип Float32.
- Все выходы кроме выхода «Режим работы» имеют тип Float32.
- Выход «Режим работы» имеет тип Uint8 и содержит набор флагов работы ФБ (пока не используется).

ФБ №64 «Вычислитель Knap»

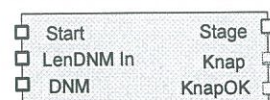
ФБ Вычисляет Knap, по входному массиву динамограммы.

Примечания:

ФБ выполняется не за один цикл конфигурации а за 23 цикла (в связи с длительностью обработки данных). Для определения стадии выполнения, используется выходной параметр Stage[0;22].

- ФБ вычисляет Knap по входному массиву динамограммы, после прохождения полной процедуры вычисления, выставляет признак KnapOK и само значение Knap.
- Команда начала работы Start, разрешает подсчет Knap, проверка проходит только на первой стадии работы, поэтому данный вход может быть импульсным и держать его до подтверждения расчета Knap необязательно.
- Признак рассчитанного Knap, устанавливается только на последней стадии расчета, и сбросится на следующем цикле.
- ФБ который считывает значение Knap, должно дождаться установки флага KnapOK и только после этого прочитать сам коэффициент.

ВНИМАНИЕ!!! В конфигурации можно использовать только один ФБ расчета КНАП в связи с использованием переменных статичных для функции обработки.



ФБ № 68 «Автомат управления приводом по КНАП» (автоматическое повторное включение)

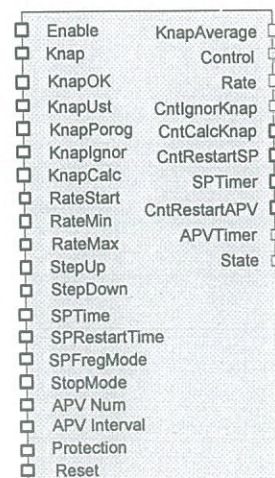
ФБ реализует алгоритм управления частотой привода по коэффициенту наполнения (КНАП), включения (отключения) пускателя, контроля срабатывания срыва подачи и обработки аварии (с возможностью повторного включения).

Основные задачи ФБ:

- регулировка выходной частоты (Out->Rate) и управление состоянием пускателя(Out->Control);
- определение СП (срыв подачи) и отсечка таймера простоя;
- реагирование на Защиты (In->Protection);

Описание работы по режимам:

- В режиме «ВКЛ всегда» (значение Mode 4-5) Control установится в «1», частота Rate будет равна частоте «Начальная частота при-

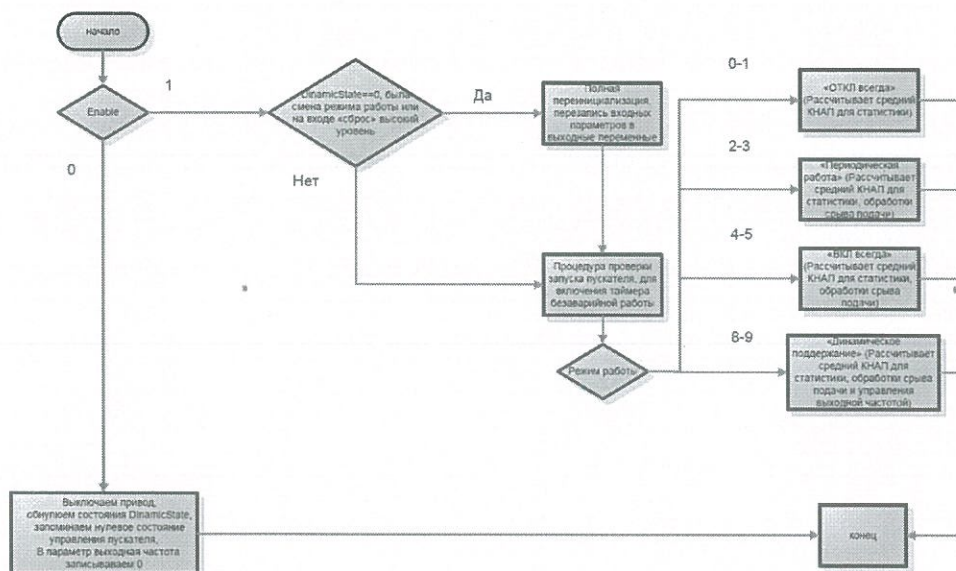


вода Start Rate» в режиме 4 или равна 500 в режиме 5.

- В случае если произошел срыв подачи (средний КНАП меньше порогового $K_{napPorog}$) произойдет останов на время простоя, если на вход Защиты подана «1» произойдет останов на время АПВ ($APV_Interval$).
- Параметры $SPFreqMode$ и $StopMode$ не действительны для данного режима, в случае останова по срыву подачи частота будет 0 и пускатель выключится.
- Режим «Периодическая работа» (значение $Mode$ 2-3), обусловлен параметрами $PeriodicModeTime$ (время включенного состояния) и $SPTime$ (время простоя). После перехода на этот режим произойдет включения СКВ на время $PeriodicModeTime$, таймер $PeriodicModeTimer$ показывает сколько тактов осталось до простоя на время накопления $SPTime$, таймер $SPTimer$ показывает сколько тактов осталось до включения СКВ.
- Параметры $SPFreqMode$ и $StopMode$, не действительны для данного режима, в случае останова по срыву подачи частота будет 0 и пускатель выключится.
- Режим «Динамическое поддержание», поддерживает выходную частоту в зависимости от среднего значения КНАП на входе K_{nap} , параметры K_{napUst} и $K_{napGist}$ образует интервал $[K_{napUst} - K_{napGist}; K_{napUst} + K_{napGist}]$ стабильной работы, в случае если КНАП превысит $K_{napUst} + K_{napGist}$ частота $Rate$ увеличится на $StepUp$ (шаг увеличения частоты), если будет меньше $K_{napUst} - K_{napGist}$ уменьшится на $StepDown$ (шаг уменьшения частоты), при этом частота $Rate$ не может выйти за интервал $[RateMin; RateMax]$.

ВНИМАНИЕ!!! Во всех режимах останов по срыву подачи или защите отключен на время пуска ($Pusk_Time$).

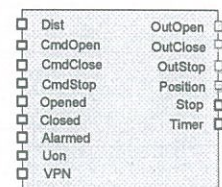
Краткий алгоритм работы ФБ



ФБ №80 «Задвижка»

ФБ управляет открытием/закрытием задвижки. На вход подаются:

- $Dist$ - Способ управления (1 – Дистанционный, 0 – Местный)
- $CmdOpen$, $CmdClose$, $CmdStop$ - Команда открыть, закрыть или остановиться в текущем положении
- $Opened$, $Closed$ - Сигнал, открыта или закрыта задвижка в данный момент
- $Alarmed$ - Авария муфты
- Uon - Наличие напряжения
- VPN - Время полного хода



Выходы:

- OutOpen, OutClosed, OutStop – выход управления открыть/закрыть/стоп
- Position – предполагаемая позиция
- Stop – состояние
- Timer – обратный таймер времени хода задвижки

Блок-схема работы ФБ «Задвижка»

