Научно-инженерный центр
ИНКОМСИСТЕМ

ИНСТРУКЦИЯ

По работе с системой CoDeSys в ИВК «АБАК+»

Версия 1.9

2020

Содержание

1 Основные обозначения

В документе приняты следующие основные обозначения:

- ИВК АБАК+ (ИВК);
- Функциональный блок (ФБ).

2 История версий target-файлов ИВК

ABAK_V1.3:

1) В ФБ MBMaster добавлен режим работы с дискретными регистрами (coils)
2) В ФБ MBMaster добавлен флаг управления опросом PollEnabled и режим разового запроса-ответа OnePollRegim.
3) Реализована внутренняя централизованная процедура выделения ресурсов в AbakUtils.(исправлен редкий дефект с работой ФБ AbakUtils, проявляющийся на этапе написания кода и вызывающий зависание пользовательской программы).

ABAK_V1.4:

4) В ФБ MBMaster добавлен флаг SuccessRead (см. описание ФБ MBMaster).
5) В ФБ MBMaster изменен алгоритм чтения ответа от устройства. Теперь не всегда жестко выжидается время RespTime ожидания ответа от устройства, а постоянно ожидается ответ и если он получен, то сразу возвращается результат. Данное изменение алгоритма чтения позволит ускорить цикл обмена с устройствами.
6) Добавлен новый ФБ AbakIO для удобной работы с каналами ввода-вывода Абака.
7) Добавлен новый ФБ SamplerA для работы с Пробоотборниками Стандарт А.
8) Добавлен новый ФБ ChromFB для чтения комп. состава с Хроматографа.

ABAK_V1.51:

1) Исправлена работа метода CopyRegs ФБ MDB (не работал с версии 1.3).

ABAK_V1.6:

1) Добавлен новый ФБ GigromFB для чтения значений точки росы и давления с гигрометров "КОНГ-Прима-10", "AMETEK 241", “Hygrovision – BL” по последовательному порту.
2) Добавлен новый ФБ PollingBC для опроса корзины расширения по последовательному порту.
3) Добавлена поддержка трех виртуальных Modbus пространств в функциональном блоке MDB.
4) В группу MBAbakGrp добавлен метод CopyRegsExt для копирования группы регистров из виртуального или основного Modbus пространства в виртуальное или основное Modbus пространство.

ABAK_V1.61:

1) В ФБ MBMaster добавлен режим работы с OMNI 3000/6000 по протколу Modbus RTU OMNI Compatible (см. параметр DeviceType ФБ MBMaster).
2) В ФБ SamplerA добавлены параметры virtual_type, CMD, MB_VAL, Time_ET (см. описание ФБ SamplerA).

ABAK_V1.62:

1) В группу SamplerGrp добавлен новый ФБ SamplerJisk210 для работы с пробоотборниками Jiskoot 210 серии.
2) В группу SamplerGrp добавлен новый ФБ SamplerJisk710 для работы с пробоотборниками Jiskoot 710 серии.

ABAK_V1.63:

1) Исправлена ошибка FB MBMaster (появившаяся с версии 1.61), касающаяся команды записи в группу регистров.
2) Доработан FB PollingBC (оптимизирована процедура опроса модулей корзины расширения)

ABAK_V1.64:

1) В ФБ AbakIO добавлены расширенные методы для одновременной работы с пространства Абака и корзин расширения. Новые методы: GetAI_EX, GetDI_EX, GetFI_EX, GetPI_EX, SetAO_EX, SetDO_EX.
2) Добавлен ФБ RelationsVirtReal для выполнения функций взаимодействия тегов реального и регистров виртуального адресных пространств ИВК АБАК+.
3) Исправлена ошибка FB PollingBC касающаяся работы дискретных сигналов модуля FI.
4) Исправлена ошибка FB TDB связанная с недоступностью некоторых тегов.

ABAK_V1.65:

1) В раздел BusCoupler добавлен ФБ SuperBC позволяющий одновременную работу с тремя корзинами расширения ПЛК АБАК.
2) В ФБ PollingBC добавлены новые параметры (BCNum – номер корзины расширения, StandAlone – признак работы корзины в одиночном режиме, стал доступен блок MBMaster - mb), которые позволяют вариативную настройку подключения корзин.
3) В разделе AbakIO откорректированы расширенные методы для одновременной работы с пространства Абака и дополнительными корзинами расширения.
4) Исправлена ошибка FB PollingBC касающаяся работы компактных значений дискретных сигналов модуля DI.

ABAK_V1.651:

1) В ФБ PollingBC, SuperBC изменен принцип работы с параметром StandAlone.

ABAK_V1.66:

1) Исправлена ошибка ФБ PollingBC связанная с погрешностью выставления выходных значений на АО каналах

ABAK_V1.67:

1) Исправлена ошибка ФБ SamplerA связанная с вычислением оставшегося времени пробоотбора по времени.

ABAK_V1.68:

1) Исправлена ошибка метода SetString ФБ MDB связанная с отбрасыванием последнего символа строки нечетной длины.
2) Добавлена возможность задания Modbus адреса в реальном времени без перезагрузки для корзины расширения в ФБ PollingBC.
3) Исправлен метод GetDI_EX ФБ AbakIO выдававший некорректные значения 9 и 10 каналов модуля DIO для 2 и 3 корзины.

ABAK_V1.69:

1) Устранена уязвимость потери данных типов каналов модулей в корзине расширения.

ABAK_V1.7:

1) Добавлен сброс алгоритма виртуального пробоотборника при завершении циклов.

ABAK_V1.8:

1) Добавлена возможность корректировки приоритета задач Codesys.

ABAK_V1.9:

1) Добавлен ФБ SamplerJisc710_ext для подключения сигнала исполнения отбора на частотный модуль и обработкой состояния резервирования ИВК.
2) В ФБ ФБ SamplerA добавлена обработка состояния резервирования ИВК.

3 Установка и настройка среды программирования CoDeSys

Среда программирования CoDeSysV3 предназначена для разработки прикладных программ на языках стандарта МЭК 61131-3.

Для установки среды программирования CoDeSysV3 требуется запустить файл Setup_CoDeSysV3xxx.exe из каталога установки. Далее выбрать Стандартную установку и следовать всем предложенным указаниям.

После завершения установки требуется установить target-файлы с определениями для устройства «Абак» в среду CoDeSysV3. Для этого необходимо запустить файл ABAK_Vx.x.package, после чего автоматически запустится Менеджер пакетов CoDeSysV3. После завершения работы Менеджера пакетов в Репозитарий устройств CoDeSysV3 будет установлено описание устройства «АБАК», а в Репозитарий библиотек - библиотека AbakUtils.library.

4 Пример создания проекта простой программы

1) Для создания проекта простой программы в CoDeSysV3 требуется запустить CoDeSysV3 и выбрать пункт меню – «Новый проект».

2) Будет выведен диалог выбора устройства, для которого создается данный проект. Выбираем устройство «Абак».

3) Далее требуется выбрать язык программирования основной программы PLC_PRG. Выбираем язык, например, ST.

4) Пишем нужный код программы.

5) Компилируем код без ошибок.

6) Настраиваем подключение. Для этого соединяем ИВК и ПК через Ethernet. ИВК и ПК должны быть в одной подсети. После подключения выполняем поиск ИВК (двойной клик на устройство «Абак», вкладка «Установки соединения», кнопка «Сканировать сеть»). После успешного поиска нажимаем кнопку «Установить активный путь».

Важно!!! Если несколько Абаков в сети как выбрать нужный?

В конце названия устройства добавлен серийный номер Абака т.е. устройство с названием «Абак0324» - это Абак с серийным номером 324.

Кроме этого, в квадратных скобках после названия каждому Абаку автоматом присваивается адрес узла, который равен 16ричному коду двух последних чисел в IP адресе контроллера, например, если IP адрес = 192.168.3.66 то адрес узла=[0342] (отображается в квадратных скобках).

7) Подключаемся к Абаку. Для соединения нажимаем кнопку «Логин» на панели инструментов.

8) После подключения выводится вопрос о загрузке написанной программы в контроллер. Отвечаем да.

9) Для запуска программы после загрузки нажимаем кнопку «Старт» на панели инструментов.

Примечание. Если ПК содержит несколько сетевых адаптеров, то возможна проблема с подключением к ИВК. Для подключения можно попробовать один из двух способов:

1) отредактировать файл c:\Program Files (x86)\3S CoDeSys\GatewayPLC\Gateway.cfg. Удалить раздел [CmpRouter] и вставить вместо него следующий код:

[CmpRouter]
NumRouters=1
MaxRouters=1
0.MainNet=ether1
[CmpBlkDrvUdp]
itf.0.ipaddress=192.168.12.123
itf.0.networkmask=255.255.255.0
itf.0.name=ether1

в строке itf.0.ipaddress указать IP адрес ПК, а в строке itf.0.networkmask указать маску подсети, которая **должна быть одинакова в настройках ИВК АБАК и в сетевых настройках ПК!**

После сохранения изменений остановить и повторно запустить Gateway через SysTray.

2) в настройках Gateway указать вместо localhost IP адрес ИВК.

Первый способ более правильный т.к. обеспечивает более стабильное подключение.

4.1 Дополнительные действия при создании проекта сложной и объемной программы.

Если создаваемая программа имеет значительный объем строк кода, использует много самописных функциональных блоков или работает с большими массивами данных, то необходимы дополнительные действия. Необходимо после пункта 7 параграфа 4 произвести дополнительную загрузку, нажав кнопку «Создать загрузочное приложение» на вкладке «Онлайн».

5 Взаимодействие программы на CoDeSys с адресным пространством Modbus ИВК

5.1 Использование ФБ MDB для доступа к адресному пространству ИВК через адреса регистров.

Для взаимодействия с адресным пространством ИВК используется функциональный блок MDB. Функциональный блок MDB содержит методы для чтения и записи параметров из адресного пространства Modbus ИВК.

1) Методы чтения из адресного пространства:
REAL GetReal(reg:WORD);
WORD GetWord(reg:WORD);
DWORD GetDWord(reg:WORD);
STRING GetString(reg:WORD);
LREAL GetLReal(reg:WORD); (добавлен в ABAK_V1.2)

2) Методы записи в адресное пространство:
BYTE SetReal(reg:WORD, val:REAL);
BYTE SetWord(reg:WORD, val:WORD);
BYTE SetDWord(reg:WORD, val:DWORD);
BYTE SetString(reg:WORD, val:STRING);
BYTE SetLReal(reg:WORD, val:LREAL); (добавлен в ABAK_V1.2)

где reg – адрес регистра в адресном пространстве +1 (так же как задан в Конфигураторе),
val – записываемое значение.

3) Метод копирования группы регистров (добавлен в ABAK_V1.2)
BYTE CopyRegs(SrcRegAdr:WORD, SrcIsVirtual:BOOL, DstRegAdr:WORD, DstIsVirtual:BOOL, RegCnt:WORD), где

SrcRegAdr – начальный адрес регистра источника,
SrcIsVirtual – признак виртуального или реального пространства регистра источника,
DstRegAdr – начальный адрес регистра назначения,
SrcIsVirtual – признак виртуального или реального пространства регистра назначения,
RegCnt – количество копируемых регистров.

Для использования функционального блока MDB в нескольких местах проекта требуется объявить его экземпляр в разделе глобальных переменных и использовать его.
В примере объявлен экземпляр gMDB:

```
VAR_GLOBAL
  gMDB:MDB;
END_VAR
```

А его использование:

```
gMDB.SetReal(4805,rval);
```

для записи влажности среды №1.

5.2 Виртуальное Modbus адресное пространство ИВК.

ИВК поддерживает три полноценных адресных пространств с адресацией до 65535 регистров, любые адреса которых можно использовать по своему усмотрению в CoDeSys. Для взаимодействия с виртуальным пространством используется тот же самый функциональный блок MDB, только перед вызовом требуется задать значение его входного параметра IsVirtualModbus := TRUE, а в параметр VirtualModbusNum записать номер виртуального пространства (1-3).

Пример:
gMDB.IsVirtualModbus := TRUE;
gMDB.VirtualModbusNum := 2;
gMDB.SetReal(1000,rval);
для записи регистра 2-го виртуального пространства с адресом 1000.

Метод копирования группы регистров (добавлен в ABAK_V1.6)

BYTE CopyRegsExt(SrcRegAdr:WORD, DstRegAdr:WORD, RegCnt:WORD, SrcVirtualModbusNum:WORD, DstVirtualModbusNum:WORD), где

SrcRegAdr:WORD – начальный адрес регистра источника;
DstRegAdr:WORD – начальный адрес регистра назначения;
RegCnt:WORD – количество копируемых регистров;
SrcVirtualModbusNum:WORD - номер модбас пространства источника (0- основное пространство ИВК Абак, 1..3 – виртуальные пространства)
DstVirtualModbusNum:WORD – номер модбас пространства назначения (0- основное пространство ИВК Абак, 1..3 – виртуальные пространства)

5.3 Использование ФБ TDB для доступа через названия тегов(предпочтительно).

Функциональный блок TDB выполняет те же функции доступа к адресному пространству ИВК АБАК+, что и ФБ MDB, только вместо адресов регистров использует названия тегов. Обращение к параметрам по названию их тегов предпочтительнее, потому что исключает задание неправильного адреса в modbus пространстве и в случае обновления программного обеспечения в будущем (если сместятся адреса modbus) автоматически привяжет теги к новым modbus адресам.

1) Методы чтения из адресного пространства:
REAL GetReal(intag:STRING, inidx:INT);
WORD GetWord(intag:STRING, inidx:INT);
DWORD GetDWord(intag:STRING, inidx:INT);
STRING GetString(intag:STRING, inidx:INT);
LREAL GetLReal(intag:STRING, inidx:INT); (добавлен в ABAK_V1.2)

2) Методы записи в адресное пространство:
BYTE SetReal(intag:STRING, inidx:INT, val:REAL);
BYTE SetWord(intag:STRING, inidx:INT, val:WORD);
BYTE SetDWord(intag:STRING, inidx:INT, val:DWORD);
BYTE SetString(intag:STRING, inidx:INT, val:STRING);
BYTE SetLReal(intag:STRING, inidx:INT, val:LReal); (добавлен в ABAK_V1.2)

где intag – строковое название тега (так же как назван в Конфигураторе),
inidx – индекс тега (если не используется, то должен быть = -1)
val – записываемое значение.

Пример записи 100% метана во вторую среду:
- вариант с индексом:
gTDB.SetReal('SRC_CH4_COMP_', 2, 100.0);
- вариант без индекса:
gTDB.SetReal('SRC_CH4_COMP_2', -1, 100.0);

Пример чтения контрольной суммы ПО ИВК:
dwtmp:=gTDB.GetDWord('PRG_CRC32',-1);

5.4 Взаимодействие реального и виртуального Modbus адресных пространств ИВК.

Функциональный блок RelationsVirtReal выполняет функции взаимодействия тегов реального и регистров виртуального адресных пространств ИВК АБАК+. При помощи этого блока упрощается виртуализация реальных значений параметров ИВК АБАК+, вызванная необходимостью изменения компоновки регистров адресного пространства, типов данных или принципов представления типов данных.

Параметры настройки функционального блока RelationsVirtReal:

Название параметра	**Тип**	**Значение по умолчанию**	**Описание**
v_num	WORD	1	Номер используемого виртуального пространства ИВК АБАК+
r_param	STRING	-	Имя тега из реального пространства ИВК АБАК+
r_idx	INT	-	Индекс тега из реального пространства ИВК АБАК+
v_param	WORD	-	Номер регистра виртуального адресного пространства ИВК АБАК+
r_type	WORD	0	Тип параметра из реального пространства (0 - int(word), 1 - float(real), 2-ulong(dword), 3 - double(lreal))
v_type	WORD	0	Тип параметра из виртуального пространства (0 - int(word), 1 - float(real), 2-ulong(dword), 3 - double(lreal))
rw	WORD	0	Выбор режима чтения/записи (0 - read only, 1 - read and write)
SwapBytes	BOOL	FALSE	Если TRUE, то будет производиться перестановка байтов в регистрах данных
SwapRegs	BOOL	FALSE	Если TRUE, то будет производиться перестановка регистров в массиве данных.

Возможны следующие варианты совмещения типов данных:

Тип данных реального пространства	**Тип данных виртуального пространства**	**Режим чтения/записи**
int(word)	*int(word)*	*read and write, read only*
float(real)	*float(real)*	*read and write, read only*
int(word)	*float(real)*	*read and write, read only*
float(real)	*int(word)*	*read and write, read only*
ulong(dword)	*ulong(dword)*	*read only*
ulong(dword)	*float(real)*	*read only*
double(lreal)	*double(lreal)*	*read only*
double(lreal)	*float(real)*	*read only*

При изменении типов данных необходимо учитывать последствия возможного искажения данных.

Для каждого параметра адресных пространств ИВК АБАК+ необходимо использовать свой экземпляр ФБ RelationsVirtReal.

Пример использования данных ФБ:

```
VAR
    Var1        :RelationsVirtReal;
    Var2        :RelationsVirtReal;
END_VAR

{PLC_PRG}

Var1(r_param:=P_VAL_LINE_',r_idx:=1,v_param:=1,r_type:=1,v_type:=1,rw:=0,v_num:=1,SwapRegs:=1,SwapBytes:=0);

Var2(r_param:=REGIM_PROVE', r_idx:=-1, v_param:=3, r_type:=0, v_type:=0, rw:=1,v_num:=1,SwapRegs:=0,SwapBytes:=0);
```

6 Реализация функций Modbus Master

Функции Modbus Master реализуются при помощи группы MBMasterGrp.
Основные элементы группы MBMasterGrp:

Название	**Тип**	**Описание**
MBMaster	Функциональный блок	Главный блок, выполняющий обмен с устройством по протоколу Modbus
MBMastersRun	Функциональный блок	Блок, который управляет первоначальной настройкой и последовательностью вызовов блоков MBMaster.

Параметры настройки функционального блока MBMaster:

Название параметра	**Тип**	**Значение по умолчанию**	**Описание**
Port	UINT	1	Номер последовательного порта от 1 до 4, через который будет осуществляться обмен с устройством
ipPort	UINT	502	IP порт устройства
ipAddress	STRING	‘127.0.0.1’	IP адрес устройства
Address	BYTE	1	Modbus адрес устройства
IsWrite	BOOL	FALSE	Если FALSE – то будет выполнятся чтение из устройства, если TRUE –запись
StartReg	WORD	0	Начальный запрашиваемый регистр modbus устройства(**начинается с нуля!**)
RegsCnt	WORD	0	Количество запрашиваемых регистров modbus устройства
pBuffer	POINTER	-	POINTER TO ARRAY[0..99] OF WORD Указатель на массив, куда будут складываться результаты чтения или откуда будут браться данные для записи в устройство
InterReqTime	TIME	500ms	Интервал времени между запросами
RespTime	TIME	100ms	Интервал времени ожидания ответа от устройства
SwapBytes	BOOL	FALSE	Режим переворота байтов данных в modbus пакете
SwapRegs	BOOL	FALSE	Режим переворота регистров данных в modbus пакете

Protocol	BYTE	0	Тип протокола: 0-ModbusRTU, 1-ModbusTCP, 2-ModbusRTUoverTCP (RTU пакет передается через канал TCP)
ReadCmd	BYTE	3	Команда чтения по modbus: 3-Read Holding Regs, 4-Read Input Regs, 1-Discrete Coils*, 2-Discrete Inputs*
DeviceType	BYTE	0	Тип устройства: 0 – стандартное modbus устройство, 1 – ИВА(влагомер с нестандартным протоколом modbus) 2 – OMNI (используется для обмена данными с поточным компьютером OMNI 3000/6000, в случае когда Com-порт OMNI настроен в режиме Modbus RTU (OMNI Compatible), и только для чтения/записи параметров с типом FLOAT – параметр RegsCnt при этом означает количество параметров типа FLOAT(4 байта) на чтение или запись)
ReplyReadCnt	BYTE	5	Число попыток чтения до обнуления данных в буфере
*IsUseCoils**	BOOL	FALSE	Режим работы с дискретными регистрами (**При чтении требуется обязательно выбрать команду чтения ReadCmd=1 или 2**)
*PollEnabled**	BOOL	TRUE	Флаг управления опросом. Если TRUE – выполняется опрос.
*OnePollRegim**	BOOL	FALSE	Режим одной транзакции запрос-ответ. Если включен данный режим, то после установки PollEnabled в TRUE выполнится только одна **успешная** транзакция запрос-ответ, после чего PollEnabled будет установлен в FALSE. Данный режим используется для одноразовой операции записи или чтения.

* добавлен в ABAK_V1.3

Выходные параметры функционального блока MBMaster:

Название параметра	**Тип**	**Значение по умолчанию**	**Описание**
ResCode	BYTE	0	Статус опроса: 0 – отправлен запрос, 1 – нет ответа или ошибка, 2 – ответ получен успешно
PollComplete	BOOL	TRUE	Флаг завершения транзакции запрос-ответ
*LinkError**	BOOL	FALSE	Флаг ошибки связи с устройством. Выполняется ReplyReadCnt попыток связи с устройством. Если так и нет связи, то выставляется данный флаг. При восстановлении связи – флаг сбрасывается автоматически.

* добавлен в ABAK_V1.2

Входные-выходные параметры функционального блока MBMaster:

Название параметра	**Тип**	**Значение по умолчанию**	**Описание**

*SuccessRead**	BOOL	FALSE	Флаг удачного чтения:
			Устанавливается блоком в TRUE после каждого удачного чтения данных.
			Может быть сброшен в FALSE из внешнего кода для однократной обработки результатов.
			VAR
			mb:MBMaster;
			END_VAR;
			{code}PLC_PRG
			IF mb.SuccessRead THEN
			//обработка результатов чтения
			mb.SuccessRead:=FALSE;
			END_IF;

* добавлен в ABAK_V1.4

В зависимости от типа протокола (параметр Protocol) требуется настраивать разные параметры.

Если тип протокола Modbus RTU, то нужно настраивать параметр *Port* и не нужно настраивать параметры *ipPort* и *ipAddress*.
Если тип протокола Modbus TCP или ModbusRTUoverTCP, то нужно настраивать параметр *ipPort* и *ipAddress* и не нужно настраивать параметр *Port*.

Параметр pBuffer содержит указатель на массив значений или адрес одного значения для записи или чтения. Например, можно объявить переменную *R1:REAL* и указать блоку MBMaster, чтобы он возвращал туда результаты чтения параметра из устройства: *MBMaster1.pBuffer:=ADR(R1);*
Естественно, количество регистров для чтения при этом должно быть равно 2.
MBMaster1. RegsCnt:=2;

Для протокола Modbus RTU кроме номера последовательного порта (параметр *Port*), требуется настраивать и другие параметры порта, такие как скорость, четность, число стопбит. Данные настройки последовательного порта выполняются через Конфигуратор ИВК АБАК в группе «Настройки контроллера», подгруппа «Настройки последовательного порта №», при этом режим работы Modbus Slave данного порта должен быть отключен т.к. ИВК не может одновременно выполнять функции Modbus Master и Slave по одному и тому же последовательному порту.

Если требуется опрашивать несколько устройств или запрашивать разные группы параметров из одного устройства, то удобно воспользоваться функциональным блоком MBMastersRun. Данный блок содержит указатели на массив из функциональных блоков MBMaster и выполняет последовательный или параллельный вызов данных блоков в процессе работы программы.

Функциональный блок MBMastersRun содержит 1 основной метод:
AddMaster(pMBMaster:POINTER TO MBMaster, bparallelpoll:BOOL)
Который добавляет переданный через указатель экземпляр функц.блока MBMaster в цикл опроса.
Флаг *bparallelpoll* указывает очередность вызова функц.блока MBMaster внутри цикла опроса. Если задано значение *bparallelpoll=TRUE*, то данный функц.блок MBMaster вызывается каждый цикл, иначе, вызов данного функц.блока MBMaster выполняется в порядке очередности т.е. последовательно (актуально, например, в случае опроса через один последовательный порт).

Пример использования данных ФБ:

```
VAR
    mbruns:MBMastersRun();
    mb1,mb2:MBMaster;
    firstloop:BOOL:=TRUE;
    wrcoil:WORD:=0;
```

```
    rdregs:ARRAY[0..99] OF WORD;
END_VAR

{PLC_PRG}
IF firstloop THEN
    firstloop:=FALSE;

    mb1.Address:=1;
    mb1.port:=2;
    mb1.IsUseCoils:=TRUE;
    mb1.IsWrite:=TRUE;
    mb1.StartReg:=3;
    mb1.RegsCnt:=1;
    mb1.pBuffer:=ADR(wrcoil);
    mbruns.AddMaster(ADR(mb1),FALSE);

    mb2.Address:=1;
    mb2.port:=2;
    mb2.IsWrite:=FALSE;
    mb2.ReadCmd:=1;
    mb2.StartReg:=0;
    mb2.RegsCnt:=100;
    mb2.SwapBytes:=TRUE;
    mb2.pBuffer:=ADR(rdregs);
    mbruns.AddMaster(ADR(mb2),FALSE);
END_IF;

mbruns();
```

7 Реализация функций управления при помощи ПИД-регуляторов и использование дополнительных функций

Среда CoDeSysV3 имеет в своем составе предустановленную библиотеку Util.lib, которую можно подключить к проекту при помощи диалога «Добавить библиотеку» Менеджера библиотек проекта.

Данная библиотека имеет встроенные функциональные блоки для работы с регуляторами PD, PID и PID_Fixcycle, которые подробно рассмотрены в документации на данную библиотеку.

Для реализации разнообразных прикладных задач можно использовать множество функций, уже реализованных в бесплатной библиотеке для CoDeSysV3 под названием OSCAT. Последнюю версию и всю необходимую документацию можно скачать с сайта www.OSCAT.de.

8 Энергонезависимые переменные в CoDeSysV3

Среда программирования CoDeSysV3 поддерживает 3 вида переменных, в зависимости от поведения при перезагрузке, изменении и т.д.:
VAR, RETAIN, PERSISTENT.

Поведение переменных при работе с проектом и работы программы представлено в следующей таблице:

команда Online	**VAR**	**RETAIN**	**PERSISTENT**
Сброс <приложение>	-	x	x
Сброс холодный <приложение>	-	-	x
Сброс заводской <приложение>	-	-	-
Загрузка <приложение>	-	-	x
Онлайн-изменение <приложение>	x	x	x
Перезапуск ПЛК	-	x	x

x = значение сохраняется
- = переменная переинициализируется

9 Чтение итоговых накопителей по линиям, изменяемых раз в час через ФБ TOTS

Функциональный блок TOTS предназначен для чтения итоговых накопителей со всех измерительных линий ИВК.

ФБ TOTS содержит только один метод, который возвращает значение накопителя в зависимости от заданного номера индекса:

LREAL GetLReal(idx:WORD).

где *idx* – номер индекса накопителя (0..9 – 1 линия, 10..19 – 2 линия и т.д.)

10 Функциональный блок AbakIO. Работа с каналами ввода-вывода ИВК

Данный ФБ содержит функции для удобной работы с каналами ввода-вывода ИВК и корзин расширения.

Реализованные методы:

1	Метод чтения значения аналогового входа: *REAL GetAI(module:WORD, channel:WORD);* возвращается значение в мА или В.
2	Метод чтения значения дискретного входа: *BOOL GetDI(module:WORD, channel:WORD);* возвращается значение FALSE для разомкнутого контакта и TRUE – для замкнутого.
3	Метод чтения значения частотного входа: *REAL GetFI(module:WORD, channel:WORD);* возвращается значение в Гц.
4	Метод чтения значения импульсного входа: *DWORD GetPI(module:WORD, channel:WORD);* возвращается количество импульсов.
5	Метод записи значения аналогового выхода: *BOOL SetAO(module:WORD, channel:WORD, val:REAL);* в переменную val записывается значение в мА или В.
6	Метод записи значения дискретного выхода: *BOOL SetDO(module:WORD, channel:WORD, val:BOOL);* в переменную val записывается значение FALSE для того, чтобы разомкнуть контакт и TRUE – чтобы замкнуть контакт.
7	Метод чтения значения аналогового входа: *REAL GetAI_EX(buscoup:WORD, module:WORD, channel:WORD);* возвращается значение в мА или В.
8	Метод чтения значения дискретного входа: *BOOL GetDI_EX (buscoup:WORD, module:WORD, channel:WORD);* возвращается значение FALSE для разомкнутого контакта и TRUE – для замкнутого.
9	Метод чтения значения частотного входа: *REAL GetFI_EX (buscoup:WORD, module:WORD, channel:WORD);* возвращается значение в Гц.
10	Метод чтения значения импульсного входа: *DWORD GetPI_EX (buscoup:WORD, module:WORD, channel:WORD);* возвращается количество импульсов.

11	Метод записи значения аналогового выхода: *BOOL SetAO_EX (buscoup:WORD, module:WORD, channel:WORD, val:REAL);* в переменную val записывается значение в мА или В.
12	Метод записи значения дискретного выхода: *BOOL SetDO_EX (buscoup:WORD, module:WORD, channel:WORD, val:BOOL);* в переменную val записывается значение FALSE для того, чтобы разомкнуть контакт и TRUE – чтобы замкнуть контакт.

Значение аргументов методов:

buscoup – номер корзины расширения Абака (1..3)*

module – номер модуля ввода-вывода Абака (1..6)

channel – номер канала ввода-вывода Абака (1..8) (1..10 для методов *GetDI, SetDO*)

**- можно использовать расширенные методы(_EX) и с модулями ввода-вывода, расположенными в самом Абаке, для этого необходимо задавать значение параметра buscoup равное 0.*

Для диагностики успешного выполнения любого метода используется параметр *error* функционального блока, который принимает значение TRUE в случае ошибки.

11 Функциональный блок SamplerA. Работа с пробоотборниками СтандартА.

ФБ управления пробоотборником СтандартА.

Параметры настройки функционального блока SamplerA:

Название параметра	**Тип**	**Значение по умолчанию**	**Описание**
probenum	WORD	1	Номер пробоотборника (1..3). Выбирается один из пробоотборников, настройки и выходные параметры которого заведены в стандартной конфигурации ИВК.
virtual_type	BOOL	FALSE	Включение режима виртуального пробоотборника (TRUE), DI/DO каналы не задействуются данным функциональным блоком
DImodule	BYTE	1	Номер модуля и канала дискретного ввода, принимающего сигнал от поршня (завершения отбора одной пробы).
DIcnl	BYTE	1	
DOmodule	BYTE	1	Номер модуля и канала дискретного вывода, выдающего управляющий сигнал на запуск двигателя.
DOcnl	BYTE	1	

Выходные параметры функционального блока SamplerA:

Название параметра	**Тип**	**Значение по умолчанию**	**Описание**
CMD	WORD	0	Команда на пробоотбор. Дублирует параметр «REGIM_PROBE_» ИВК Абак
MB_VAL	REAL	0	Текущий массовый расход по включенным источникам. Дублирует параметр «M_VAL_PROBE_» ИВК Абак
Time_ET	Time	0	Время до окончания пробоотбора. В случае использования уставки по массе, Time_ET – прогнозируемое время до окончания пробоотбора при текущем расходе

Все остальные настройки выполняются через Конфигуратор ИВК АБАК или AbakTool путем настройки параметров соответствующего пробоотборника, выбранного при помощи параметра *probenum*. Просмотр процесса отбора выполняется там же.

Выполнение ФБ осуществляется простым вызовом экземпляра блока в основном цикле программы.

Алгоритм отбора.

Отбор осуществляется только на рабочем или одиночном ИВК.

ФБ SamplerA поддерживает режим автоматического или ручного отбора.

Автоматический отбор может выполняться по массе или по времени.

Для настройки автоматического отбора нужно выбрать режим работы пробоотборника = автомат и обязательно указать количество циклов отбора.

Для настройки автоматического отбора по времени нужно задать уставку по времени в часах, если при этом выставлен параметр «Отбирать по времени только если есть расход», то отбор будет выполнен только при наличии расхода.

Для настройки автоматического отбора по массе нужно задать уставку по массе в тоннах и источник массового расхода (Это может быть любая измерительная линия или станция. В случае выбора нескольких источников их показания суммируются).

После завершения заданного количества циклов отбора в автоматическом режиме, режим работы изменяется на ручной.

После смены баллона требуется снова перевести режим работы в автомат, при этом сбросится счетчик циклов отбора проб и начнется новый отбор.

Количество отобранных проб отображается в параметре «Цикл отбора пробы».

Если режим работы ручной, то можно отбирать пробы при помощи команды на отбор пробы «старт».

Алгоритм отбора одной пробы следующий:

- выдается DO сигнал на запуск двигателя
- ожидается DI сигнал от поршня
- если пришел сигнал поршня, то отбор одной пробы успешный, если за максимальное время отбора одной пробы (настраивается через Конфигуратор) не пришел сигнал, то выдается сообщение в параметре Диагностика - «ошибка поршня».

Пример использования ФБ:

```
VAR
    Sampler:SamplerA;
END_VAR

{PLC_PRG}
Sampler.probenum:=1;
Sampler.DOmodule:=6;
```

Sampler.DOcnl:=1;
Sampler.DImodule:=6;
Sampler.DIcnl:=2;
Sampler();

Остальные настройки и просмотр процесса отбора выполняются через Конфигуратор ИВК АБАК или AbakTool.

12 Функциональный блок ChromFB. Чтение состава с Хроматографа.

ФБ чтения состава с хроматографа.

Параметры настройки функционального блока ChromFB:

Название параметра	**Тип**	**Значение по умолчанию**	**Описание**
mb	MBMaster		Блок модбас мастера опроса
numsostav	BYTE	1	Номер параметров среды (1..12) куда будет сложен считываемый состав
comparr	ARRAY[1..21] OF REAL		Компонентный состав, считанный с хроматографа
compnotupdate	ARRAY[1..21] OF BOOL	1	Массив признаков необновляемых компонентов состава (например условно-постоянных компонентов, значения которых не должны меняться при записи состава в параметры среды). По-умолчанию, для всех компонентов стоит FALSE т.е. будет выполняться запись в параметры среды.

Выполнение ФБ осуществляется простым вызовом экземпляра блока в основном цикле программы.

Пример простого использования:

```
VAR
    firstloop:BOOL:=TRUE;
    mbruns:MBMastersRun();
    chrom:ChromFB;
END_VAR

{PLC_PRG}

IF firstloop THEN
    firstloop:=FALSE;
    chrom.mb.Address:=1;
    chrom.mb.port:=3;
    chrom.mb.StartReg:=0;
    chrom.mb.RegsCnt:=22;
    mbruns.AddMaster(ADR(chrom.mb),FALSE);
```

```
    chrom.numsostav:=1; //номер параметров среды
END_IF;

mbruns();
chrom();
```

Если порядок компонентов, читаемых с хроматографа не совпадает с порядком компонентов в параметрах среды абака, то пример будет выглядеть так:

```
mbruns();
(*перестановка компонентов*)
IF chrom.mb.SuccessRead THEN
    chrom.comparr[4]:=chrom.comparr[2];
    chrom.comparr[1]:=chrom.comparr[3];
    //и т.д.
END_IF;
chrom();
```

Наличие связи с хроматографом диагностируется с помощью параметра «Состояние хроматографа» - тег OUT_ERROR_HMT_COMP в пространстве Абака («Параметры среды» - «Дополнительные выходные параметры»).

13 Функциональный блок GigromFB. Чтение данных с гигрометров.

Функциональный блок GigromFB поддерживает три модели гигрометров: "КОНГ-Прима-10", "AMETEK 241", “Hygrovision – BL”.

Параметры настройки функционального блока GigromFB:

Название параметра	**Тип**	**Значение по умолчанию**	**Описание**
mb	MBMaster		Блок модбас мастера опроса
Num	BYTE	1	Номер гигрометра в адресном пространстве Абак (1-3)
gigrom_type	BYTE	1	Модель гигрометра (1-"КОНГ-Прима-10", 2-"AMETEK 241", 3 - “Hygrovision – BL”)
add_pressure	BOOL	FALSE	Разрешение чтения значений давления с гигрометра (0-нет, 1-да)
Units	BYTE	1	Данный параметр применяется только для гигрометра "AMETEK 241" (0 - читаем значение точки росы в градусах, 1 – читаем содержание воды в ppm)

Для правильной работы ФБ необходимо в программе “Абак Конфигуратор” или “Абак Tool” установить источник сигнала для параметров, считываемых с гигрометра в состояние - «внешний ресурс». Наличие связи с гигрометром диагностируется с помощью параметра «Код ошибки» - тег OUT_ERROR_HYGRO в пространстве Абака («Гигрометр» - «Дополнительные выходные параметры»).

Выполнение ФБ осуществляется простым вызовом экземпляра блока в основном цикле программы.

Пример простого использования:

```
VAR
    firstloop:BOOL:=TRUE;
    gigrom: GigromFB;
END_VAR

{PLC_PRG}

IF firstloop THEN
```

```
        firstloo p := FALSE;
        gigrom.mb.Address := 1;
        gigrom.mb.port := 3;
        gigrom. gigrom _type :=2; // выбираем "AMETEK 241"
        gigrom. add_pressure :=TRUE; // используем значение давления с гигрометра
        gigrom. units :=1; // используем параметр содержание воды в ppm

END_IF;

gigrom ();
```

14 Функциональный блок PollingBC. Опрос корзины расширения Абак.

Параметры настройки функционального блока PollingBC:

Название параметра	**Тип**	**Значение по умолчанию**	**Описание**
port	UINT	1	Номер последовательного порта ***корзины*** от 1 до 4, через который будет осуществляться обмен с корзиной расширения.
BCNum	WORD	0	Номер корзины расширения от 1 до 3.
StandAlone	BOOL	TRUE	Если TRUE, то корзина опрашивается и работает в одиночном режиме – без блока SuperBC.
mb	MBMaster	-	Блок MBMaster для конфигурирования подключения АБАКа к корзине.
mb.port	UINT	1	Номер последовательного порта ***ИВК АБАК*** от 1 до 4, через который будет осуществляться обмен с корзиной расширения.

Для правильной работы ФБ необходимо в программе “Абак Конфигуратор” или “Абак Tool” настроить параметры последовательного порта, через который будет осуществляться обмен с корзиной расширения:

- Режим Modbus Slave: выкл
- Физический интерфейс (только для COM1) - должен соответствовать используемому (RS-232 или RS-485).
- Скорость (бит/с) – должна соответствовать скорости порта корзины расширения, к которому подключен ИВК Абак.

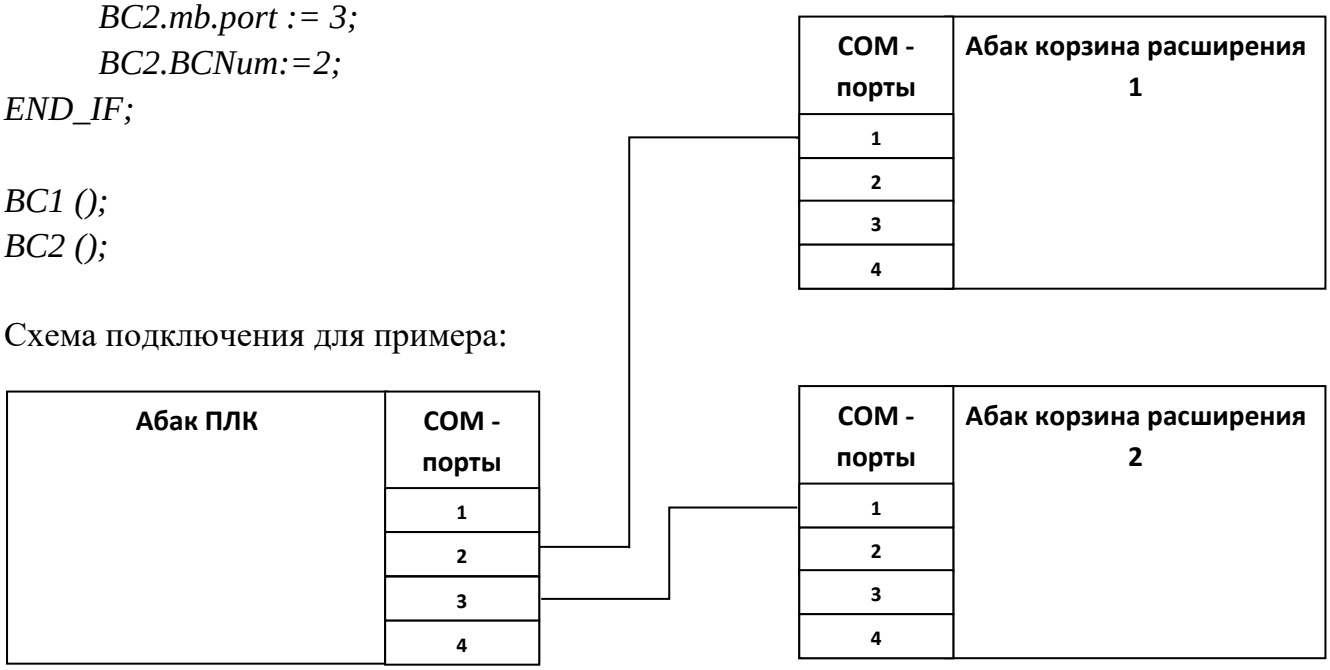
По умолчанию для портов корзины расширения назначаются адреса Modbus ID: 1, 2, 3, 4 (соотв. портам 1,2,3,4). Скорость по умолчанию 9600 бит/с для всех портов.

Выполнение ФБ осуществляется простым вызовом экземпляра блока в основном цикле программы.

Пример использования:

```
VAR
        firstloop:BOOL:=TRUE;
        BC1, BC2: PollingBC;
END_VAR

{PLC_PRG}

IF firstloop THEN
        BC1.port := 1;
        BC1.mb.port := 2;
        BC1.BCNum:=1;
        BC2.port := 1;
        BC2.mb.port := 3;
        BC2.BCNum:=2;
END_IF;

BC1 ();
BC2 ();
```

Схема подключения для примера:



При этом в Modbus пространстве ИВК Абак актуализируются индицирующие и управляющие параметры корзины расширения, и ее модулей ввода-вывода. При первом подключении корзины необходимо произвести их настройку.

После того, как связь с корзиной расширения установлена, появляется возможность ее конфигурирования с помощью программы “Абак Конфигуратор” или “Абак Tool”, а также из приложения CodeSys.

15 Функциональный блок SuperBC. Работа с несколькими корзинами расширений.

Используется для работы с несколькими корзинами расширения. Позволяет задействовать различные варианты подключения корзин к портам. Параметры настройки функционального блока SuperBC:

Название параметра	**Тип**	**Значение по умолчанию**	**Описание**
BC	ARRAY[1..3] OF PollingBC	-	Экземпляр ФБ PollingBC, характеризующий корзину расширения.

Для правильной работы ФБ необходимо настроить все используемые корзины расширения, согласно настройки функционального блок PollingBC.

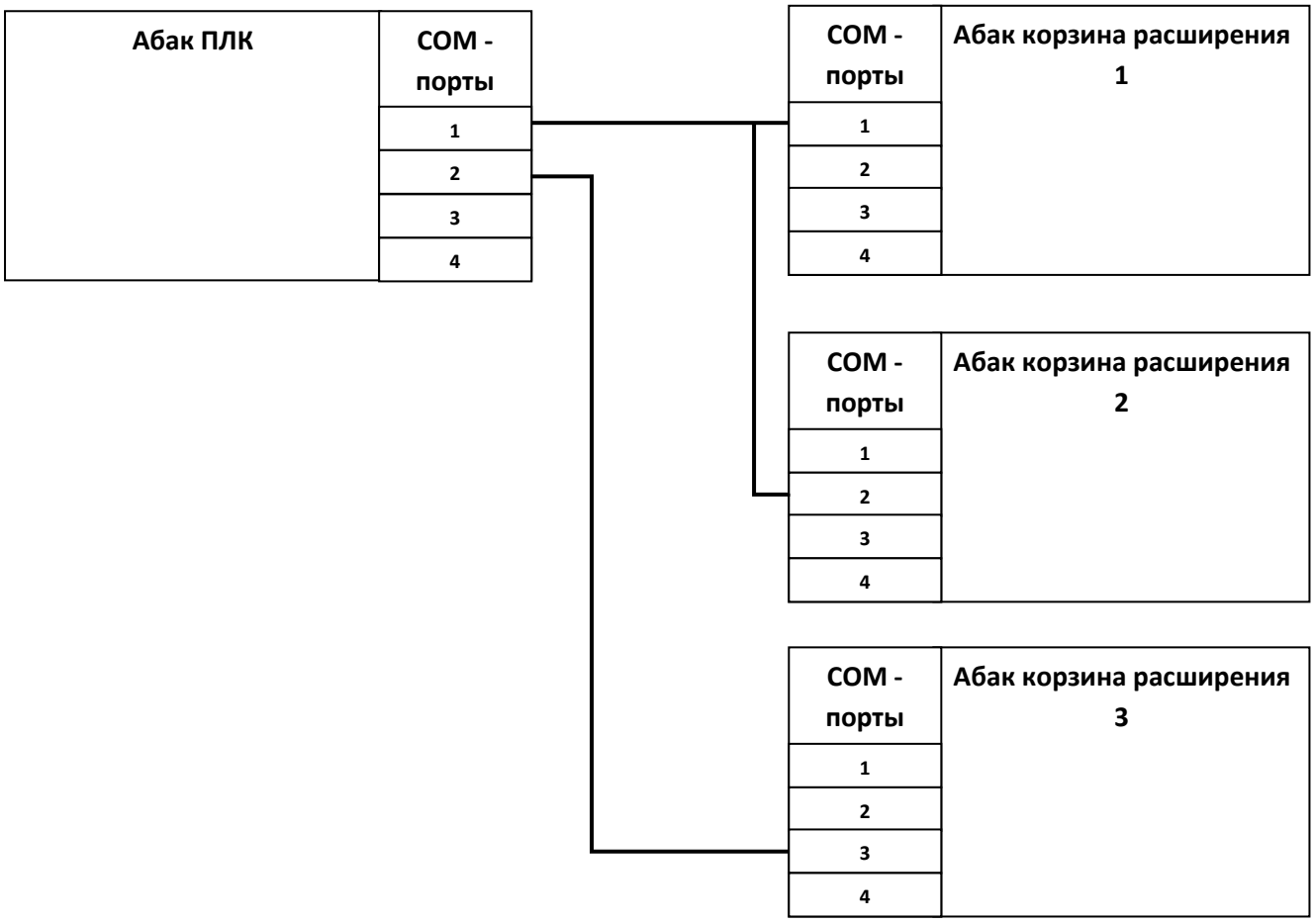
Выполнение ФБ осуществляется простым вызовом экземпляра блока в основном цикле программы.

Пример использования:

```
VAR
	firstloop:BOOL:=TRUE;
	SBC: SuperBC;
END_VAR

{PLC_PRG}

IF firstloop THEN
	SBC.BC[1].port := 1;
	SBC.BC[1].mb.port := 1;
	SBC.BC[1].BCNum:=1;
	SBC.BC[2].port:=2;
	SBC.BC[2].mb.port := 1;
	SBC.BC[2].BCNum:=2;
	SBC.BC[3].port:=3;
	SBC.BC[3].mb.port := 2;
	SBC.BC[3].BCNum:=3;

END_IF;

	SBC ();
```

В данном примере 1-я и 2-я корзина подключены на первый порт АБАКа, а 3-я корзина – на второй порт.

Схема подключения для примера:



16 Функциональный блок SamplerJisk210. Работа с пробоотборниками Jiskoot 210 серии.

ФБ управления пробоотборником Jiskoot 210.

Параметры настройки функционального блока SamplerJisk210:

Название параметра	**Тип**	**Значение по умолчанию**	**Описание**
prnum	INT	1	Номер пробоотборника (1..3). Выбирается один из пробоотборников, настройки и выходные параметры которого заведены в стандартной конфигурации ИВК.
Module4	WORD	1	Номер модуля и канала дискретного ввода, принимающего сигнал от дискретника заполнения пробоприемника на 4%.
Channel4	WORD	1	
Module80	WORD	1	Номер модуля и канала дискретного ввода, принимающего сигнал от дискретника заполнения пробоприемника на 80%.
Channel80	WORD	1	
ModuleDO	WORD	1	Номер модуля и канала дискретного вывода, выдающего управляющий сигнал на запуск поршня.
ChanneDO	WORD	1	
MinTime	TIME	TIME#1S	Минимальное время физической возможности исполнения отбора.
ImpulseTime	TIME	TIME#500MS	Время длительности импульса управляющего сигнала на исполнение отбора (зависит от модели).

Выходные параметры функционального блока SamplerJisk210:

Название параметра	**Тип**	**Значение по умолчанию**	**Описание**
BetweenTime	REAL	0	Время между циклами исполнения отбора.
ToTheEnd	Time	0	Время до окончания пробоотбора. В случае использования уставки по массе, ToTheEnd– прогнозируемое время до окончания пробоотбора при текущем расходе.

Все остальные настройки выполняются через Конфигуратор ИВК АБАК или AbakTool путем настройки параметров соответствующего пробоотборника, выбранного при помощи параметра *prnum*. Просмотр процесса отбора выполняется там же.

Выполнение ФБ осуществляется простым вызовом экземпляра блока в основном цикле программы.

Алгоритм отбора.

ФБ SamplerJisk210 поддерживает режим автоматического.

Автоматический отбор может выполняться по массе или по времени.

Для настройки автоматического отбора по времени нужно задать уставку по времени в часах, если при этом выставлен параметр «Отбирать по времени только если есть расход», то отбор будет выполнен только при наличии расхода.

Для настройки автоматического отбора по массе нужно задать уставку по массе в тоннах и источник массового расхода (Это может быть любая измерительная линия или станция. В случае выбора нескольких источников их показания суммируются).

Для запуска автоматического отбора необходимо задать параметры пробоприемника, дозы, количества циклов отбора и уставки, затем подать команду «старт», после этого счетчик циклов отбора пробы обнулится и начнет отсчитывать прошедшие пробы.

Количество отобранных проб отображается в параметре «Цикл отбора пробы».

При проведении отбора возможны следующие варианты диагностических состояний:

- норма,
- ошибка данных – отсутствуют данные по расходу, либо он нулевой.

Пример использования ФБ:

```
VAR
        Sampler: samplerJisk210;
END_VAR

{PLC_PRG}
Sampler.prnum:=1;
Sampler.ChannelDO:=1;
Sampler.ModuleDO:=6;
Sampler.Channel4:=2;
Sampler.Module4:=6;
Sampler.Channel80:=3;
Sampler.Module80:=6;
Sampler.MinTime:=TIME#2S;
Sampler();
```

Остальные настройки и просмотр процесса отбора выполняются через Конфигуратор ИВК АБАК или AbakTool

17 Функциональный блок SamplerJisk710. Работа с пробоотборниками Jiskoot 710 серии.

ФБ управления пробоотборником Jiskoot 710.

Параметры настройки функционального блока SamplerJisk710:

Название параметра	**Тип**	**Значение по умолчанию**	**Описание**
prnum	INT	1	Номер пробоотборника (1..3). Выбирается один из пробоотборников, настройки и выходные параметры которого заведены в стандартной конфигурации ИВК.
MinTime	TIME	TIME#1S	Минимальное время физической возможности исполнения отбора.
MinLevel	INT	4	Минимальный уровень для начала отбора
MaxLevel	INT	80	Максимальный уровень для окончания отбора
ModuleDI	WORD	1	Номер модуля и канала дискретного ввода, принимающего сигнал от поршня (завершения отбора одной пробы).
ChannelDI	WORD	1	
ModuleDO	WORD	1	Номер модуля и канала дискретного вывода, выдающего управляющий сигнал на запуск поршня.
ChannelDO	WORD	1	
ModuleLev	WORD	1	Номер модуля и канала аналогового канала, принимающий уровень заполнения пробоприемника.
ChannelLev	WORD	1	
AI_ON	BOOL	TRUE	Включен аналоговый вход для уровня заполнения
Level_IN	REAL	0	Вход для альтернативного источника уровня заполнения, используемый при значении AI_ON:=FALSE

Выходные параметры функционального блока SamplerJisk710:

Название параметра	**Тип**	**Значение по умолчанию**	**Описание**
BetweenTime	REAL	0	Время между циклами исполнения отбора.

ToTheEnd	TIME	0	Время до окончания пробоотбора. В случае использования уставки по массе, ToTheEnd– прогнозируемое время до окончания пробоотбора при текущем расходе.
SampleTime	TIME	0	Время за которое проходит отбор (между сработками DO и DI)

Все остальные настройки выполняются через Конфигуратор ИВК АБАК или AbakTool путем настройки параметров соответствующего пробоотборника, выбранного при помощи параметра *prnum*. Просмотр процесса отбора выполняется там же.

Выполнение ФБ осуществляется простым вызовом экземпляра блока в основном цикле программы. Выбор источника уровня заполнения происходит с помощью параметра *AI_ON*. По умолчанию значение этого входа равно «TRUE» и в этом случае используется физический аналоговый канал устройства, для использования альтернативного варианта источника уровня заполнения необходимо задать параметру *AI_ON* значение «FALSE» и подать данные на вход *Level_IN*.

Алгоритм отбора.

ФБ SamplerJisk710 поддерживает режим автоматического.

Автоматический отбор может выполняться по массе или по времени.

Для настройки автоматического отбора по времени нужно задать уставку по времени в часах, если при этом выставлен параметр «Отбирать по времени только если есть расход», то отбор будет выполнен только при наличии расхода.

Для настройки автоматического отбора по массе нужно задать уставку по массе в тоннах и источник массового расхода (Это может быть любая измерительная линия или станция. В случае выбора нескольких источников их показания суммируются).

Для запуска автоматического отбора необходимо задать параметры пробоприемника, дозы, количества циклов отбора и уставки, установить максимальное время отбора одной пробы в секундах, затем подать команду «старт», после этого счетчик циклов отбора пробы обнулится и начнет отсчитывать прошедшие пробы.

Количество отобранных проб отображается в параметре «Цикл отбора пробы».

При проведении отбора возможны следующие варианты диагностических состояний:

- норма,
- ошибка данных – отсутствуют данные по расходу, либо он нулевой,
- превышено ожидание – случай когда время между сработкой DO и ответной DI превысило максимальное время отбора одной пробы.

Пример использования ФБ:

```
VAR
        Sampler: samplerJisk710;
        MDBdata:MDB;
END_VAR

{PLC_PRG}
Sampler.prnum:=1;
Sampler.MinTime:=TIME#2S;
Sampler.ChannelDO:=1;
Sampler.ModuleDO:=6;
Sampler.ChannelDI:=2;
Sampler.ModuleDI:=6;
Sampler.ChannelLev:=3;
Sampler.ModuleLev:=5;
Sampler.MinLevel:=4;
Sampler.MaxLevel:=80;

Sampler();
```

Остальные настройки и просмотр процесса отбора выполняются через Конфигуратор ИВК АБАК или AbakTool

18Функциональный блок SamplerJisk710_ext. Работа с пробоотборниками Jiskoot 710 серии и подключение обратной связи на частотный модуль.

ФБ управления пробоотборником Jiskoot 710.

Параметры настройки функционального блока SamplerJisk710:

Название параметра	**Тип**	**Значение по умолчанию**	**Описание**
prnum	INT	1	Номер пробоотборника (1..3). Выбирается один из пробоотборников, настройки и выходные параметры которого заведены в стандартной конфигурации ИВК.
MinTime	TIME	TIME#1S	Минимальное время физической возможности исполнения отбора.
MinLevel	INT	4	Минимальный уровень для начала отбора
MaxLevel	INT	80	Максимальный уровень для окончания отбора
ModuleDI	WORD	1	Номер модуля и канала дискретного ввода, принимающего сигнал от поршня (завершения отбора одной пробы).
ChannelDI	WORD	1	
ModuleDO	WORD	1	Номер модуля и канала дискретного вывода, выдающего управляющий сигнал на запуск поршня.
ChannelDO	WORD	1	
ModuleLev	WORD	1	Номер модуля и канала аналогового канала, принимающий уровень заполнения пробоприемника.
ChannelLev	WORD	1	
AI_ON	BOOL	TRUE	Включен аналоговый вход для уровня заполнения
Level_IN	REAL	0	Вход для альтернативного источника уровня заполнения, используемый при значении AI_ON:=FALSE
Do_On_Timer	TIME	270ms	время удержания сигнала DO

Выходные параметры функционального блока SamplerJisk710:

Название параметра	**Тип**	**Значение по умолчанию**	**Описание**
BetweenTime	REAL	0	Время между циклами исполнения отбора.

ToTheEnd	TIME	0	Время до окончания пробоотбора. В случае использования уставки по массе, ToTheEnd– прогнозируемое время до окончания пробоотбора при текущем расходе.
SampleTime	TIME	0	Время за которое проходит отбор (между сработками DO и DI)

Все остальные настройки выполняются через Конфигуратор ИВК АБАК или AbakTool путем настройки параметров соответствующего пробоотборника, выбранного при помощи параметра *prnum*. Просмотр процесса отбора выполняется там же.

Выполнение ФБ осуществляется простым вызовом экземпляра блока в основном цикле программы. Выбор источника уровня заполнения происходит с помощью параметра *AI_ON*. По умолчанию значение этого входа равно «TRUE» и в этом случае используется физический аналоговый канал устройства, для использования альтернативного варианта источника уровня заполнения необходимо задать параметру *AI_ON* значение «FALSE» и подать данные на вход *Level_IN*. Также необходимо задать время удержания сигнала DO для исполнения отбора, по умолчанию время равно 270 мс.

Алгоритм отбора.

ФБ SamplerJisk710 поддерживает режим автоматического отбора.

Отбор выполняется только на рабочем или одиночном ИВК.

Автоматический отбор может выполняться по массе или по времени.

Для настройки автоматического отбора по времени нужно задать уставку по времени в часах, если при этом выставлен параметр «Отбирать по времени только если есть расход», то отбор будет выполнен только при наличии расхода.

Для настройки автоматического отбора по массе нужно задать уставку по массе в тоннах и источник массового расхода (Это может быть любая измерительная линия или станция. В случае выбора нескольких источников их показания суммируются).

Для запуска автоматического отбора необходимо задать параметры пробоприемника, дозы, количества циклов отбора и уставки, установить максимальное время отбора одной пробы в секундах, затем подать команду «старт», после этого счетчик циклов отбора пробы обнулится и начнет отсчитывать прошедшие пробы.

Количество отобранных проб отображается в параметре «Цикл отбора пробы».

При проведении отбора возможны следующие варианты диагностических состояний:

- норма,
- ошибка данных – отсутствуют данные по расходу, либо он нулевой,
- превышено ожидание – случай когда время между сработкой DO и ответной DI превысило максимальное время отбора одной пробы.

Пример использования ФБ:

```
VAR
        Sampler: samplerJisk710;
        MDBdata:MDB;
END_VAR

{PLC_PRG}
Sampler.prnum:=1;
Sampler.MinTime:=TIME#2S;
Sampler.ChannelDO:=1;
Sampler.ModuleDO:=6;
Sampler.ChannelDI:=2;
Sampler.ModuleDI:=6;
Sampler.ChannelLev:=3;
Sampler.ModuleLev:=5;
Sampler.MinLevel:=4;
Sampler.MaxLevel:=80;

Sampler();
```

Остальные настройки и просмотр процесса отбора выполняются через Конфигуратор ИВК АБАК или AbakTool