

# СИСТЕМА ЛИФТОВОГО ДИСПЕТЧЕРСКОГО КОНТРОЛЯ И СВЯЗИ «СЛДКС-1»

**ПРОГРАММА «LIFT4»  
РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ**

**ТОМ 3**

**ЕСАН.10001-01 34 01**

# Содержание

Подбор порога опроса .....	4
Окно подбора порога .....	4
Просмотр состояния лифта .....	7
Окно состояния лифта .....	8
Работа с аудиозаписями .....	11
Окно «Аудиозаписи» .....	11
Работа с журналом .....	13
Окно «Журнал» .....	13
Устранение неисправностей .....	15
Определение неисправности различных блоков .....	15
Быстрое определение неисправности блоков БКД .....	16
Просмотр журнал подключения БКД .....	16
Определение работоспособности блоков БКД .....	18
Поиск новых блоков .....	19
Проверка наличия в луче блока с определенным адресом .....	21
Дополнительные возможности БКД-М .....	22
Встроенный интерпретатор BASIC .....	24
Допустимые операторы .....	24
Составные операторы .....	24
Метки .....	24
Переменные .....	24
До версии 5.49 .....	24
Начиная с версии 5.49 .....	25
Преобразование типов .....	26
Системные переменные .....	26
Предопределенные константы .....	27
Числа .....	27
Функции .....	28
Допустимые арифметические операции .....	32
Выражения .....	32
Оператор IF THEN ELSE ENDIF .....	32
Оператор присвоения значения переменной .....	33
Оператор GOTO метка .....	33
Оператор GOSUB метка .....	33
Оператор RETURN .....	33
Оператор FOR, TO, STEP .....	33
Оператор NEXT .....	34
Оператор WHILE WEND .....	34
Оператор PRINT .....	34
Оператор END .....	35
Оператор REM .....	35
Оператор OPEN .....	36
Оператор CLOSE .....	36
Оператор INPUT .....	36
Оператор OUTPUT .....	37
Оператор RUN .....	37
Переменные объектов карты .....	38
Дополнительные поля БИУ (СОС-95 БИУ-Ф) .....	40
Дополнительные поля БАД .....	41
Дополнительные поля ККД .....	41
Дополнительные поля Индикатора .....	41
Дополнительные поля Аналогового Параметра .....	42
Дополнительные поля Управляемого переключателя (Тумблер) .....	42
Дополнительные поля БПДД .....	42
Дополнительные поля блока ИУ-1 .....	42
Дополнительные поля блока УИР-РЦ .....	43
Дополнительные поля блока БИУ-Л .....	43
Работа с неименованными объектами на карте .....	43
Ввод чисел в программу .....	44
Выполнение записи в общий журнал программы .....	45

Вызов окна программы .....	46
Редактирование программы .....	46
Вставка объектовых переменных в программу .....	46
Дополнительные возможности редактора программы .....	48
Проверка работы программы .....	48
Выполнение программы .....	48
Сохранение и загрузка программы .....	49
Постоянная работа программы .....	49
Дополнительные возможности отладочного окна .....	49
Дополнительные переменные области Bytes .....	49
Параметр лифта .....	51
БИУ-Л .....	51
БГС-П .....	52
Охранный шлейф .....	52
Датчик температуры .....	52
Блок аналоговых датчиков .....	52
БИУ СОС-95 .....	52
ККД СОС-95 .....	53
Индикатор .....	53
Аналоговый параметр .....	53
Программирование БПДД .....	53
Сброс .....	54
Посылка символа .....	54
Прием символа .....	54
Проверка приемного буфера .....	55
Установка таймаута .....	55
Программа БПДД .....	55
Программная работа с графиками .....	57
Программная работа с текстом .....	58

## Подбор порога опроса

Порогом называется специальное число, позволяющее настроить электронную систему приема ответов блока БДК-Л или блока БДК. Изменение порога позволяет подобрать оптимальный режим связи с блоком. Значение порога лежит в пределах от  $-128$  до  $+127$ .

Для подбора порога предназначено специальное окно программы, которое можно вызвать одним из двух способов:

1. Нажать кнопку «Порог» в окне свойств объекта на вкладке окна свойств «Параметры опроса».
2. Щелкнуть левой кнопкой мышки по объекту на карте, удерживая кнопки «Shift» и «Alt» на клавиатуре компьютера.

В обоих случаях откроется окно подбора порога.

## Окно подбора порога

Внешний вид окна подбора порога представлен на рисунке:

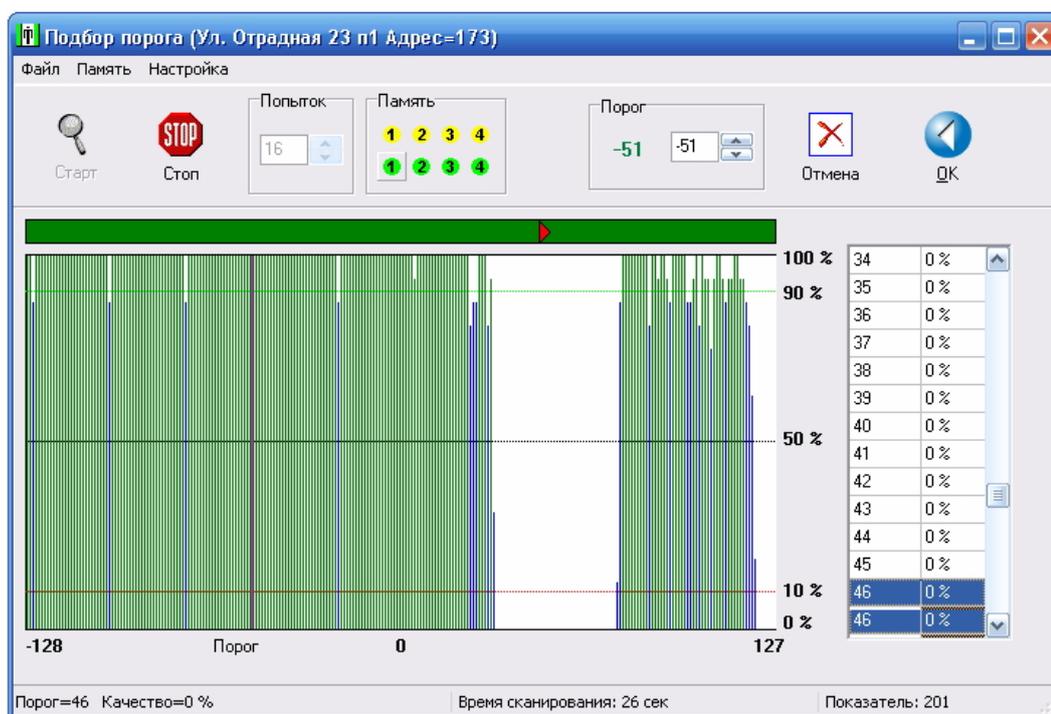


Рисунок – Окно «Подбор порога»

В верхней строке окна указано имя блока, для которого выполняется подбор порога.

### Графическая область подбора порога

В центральной левой части окна подбора порога расположена графическая область подбора порога. По горизонтальной оси располагаются значения порогов от  $-128$  до  $127$ . На вертикальной оси располагаются отсчеты качества связи с блоком в процентах (от 0 до 100%). Ноль процентов соответствует отсутствию ответов от блока, а 100 % обозначает, что на каждый запрос получен правильный ответ. Сканирование значений порогов выполняется после нажатия на кнопку «Старт».

### Таблица качества

Правее графической области подбора порога расположена таблица качества, где каждому значению порога соответствует определенное качество обмена.

### Кнопка «Старт»

Кнопка «Старт» позволяет начать процедуру подбора порога. После нажатия на кнопку очищается графическая область и таблица качества. Начинается проверка обмена на каждом значении порога. Устанавливается очередное значение порога и проверяется качество на этом пороге. Качество определяется

как соотношение в процентах сделанного успешного числа опросов к общему числу опросов на заданном пороге. Далее значение порога увеличивается на единицу и измерение качества повторяется. Полученные значения отображаются на графической области и в таблице качества. Значения порога изменяются от -128 до 127. Примерный вид окна подбора порога после завершения сканирования приведен на рисунке:

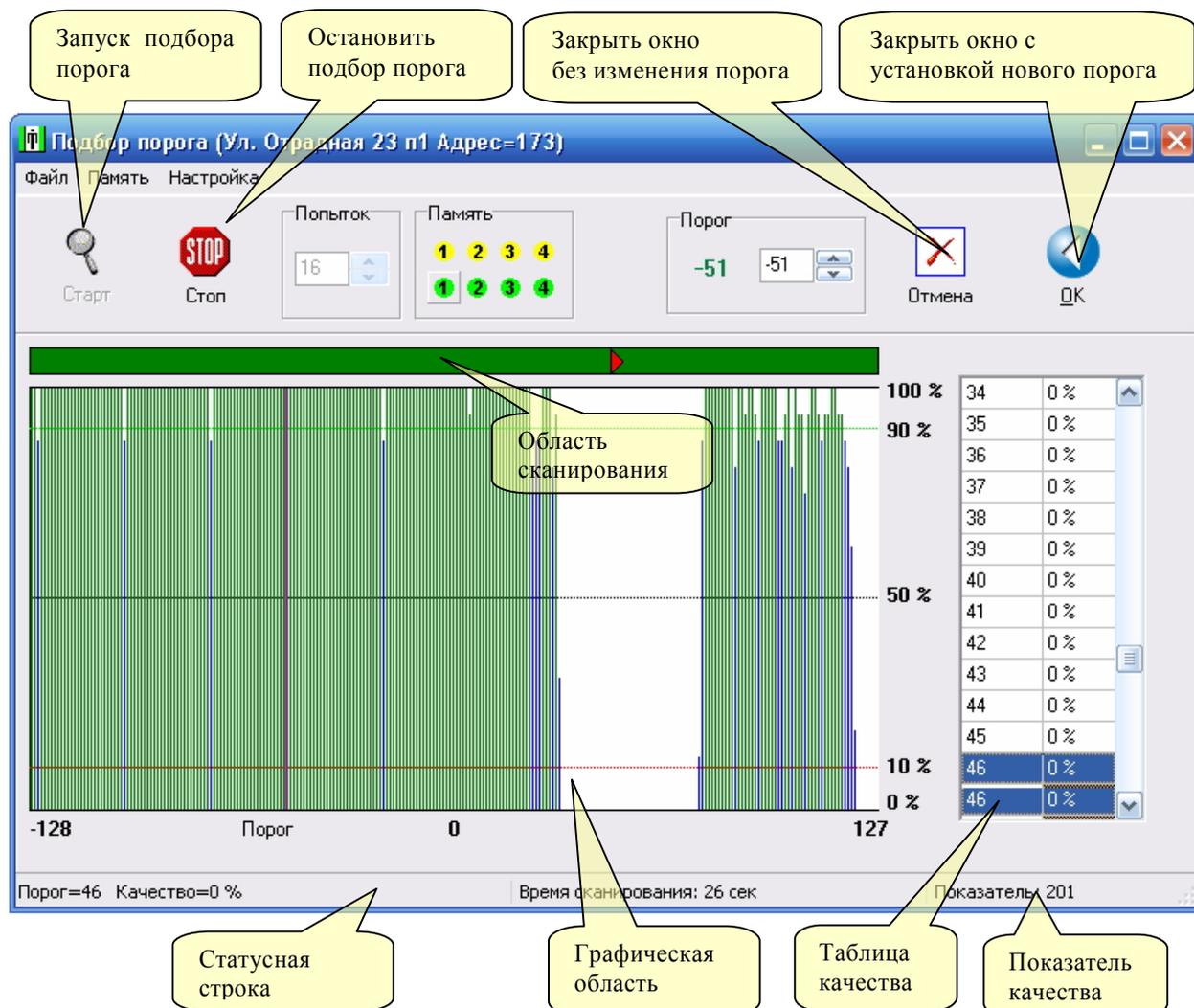


Рисунок – Элементы управления окна «Подбор порога»

На графической области зелеными линиями показаны значения порога, где качество связи более 90 %. Просмотреть значения порогов и соответствующего им качества можно в таблице качества или перемещая указатель мыши по графической области. При перемещении указателя мыши по графической области в статусной строке окна отображается значение порога и соответствующее ему качество обмена.

### **Кнопка «Стоп»**

Нажав на кнопку «Стоп» можно в любой момент времени остановить процедуру подбора порога.

### **Попыток**

В рамке «Попыток» находится поле ввода, в котором указано число попыток обмена на каждом пороге. Число 16 обозначает, что для определения качества обмена на текущем пороге будет сделано 16 опросов блока и качество будет подсчитано как соотношение числа успешных опросов к 16-ти (в процентах). Следует отметить, что в режиме «Ускоренный опрос» (см. ниже) число попыток может быть только 16.

### **Кнопки памяти**

Правее рамки «Начать от» расположены восемь кнопок памяти. Четыре кнопки желтого цвета предназначены для сохранения текущих данных графической области и таблицы качества в память. Всего можно сохранить до 4-х различных результатов подбора порога. Четыре кнопки зеленого цвета

предназначены для извлечения сохраненных ранее данных графической области и таблицы качества. Используя кнопки памяти можно просматривать изменения качества обмена по времени. Записанные значения памяти сохраняются на жестком диске.

### **Кнопка «Заккрыть»**

При помощи кнопки «Заккрыть» можно завершить работы по подбору порога без изменения текущего значения порога блока.

### **Кнопка «ОК»**

При помощи кнопки «ОК» можно завершить работы по подбору порога с установкой в качестве текущего порога блока значения указанного в рамке «Порог» правее кнопки «ОК».

### **Порог**

В рамке «Порог» расположено несколько элементов, связанных с выбором порога. В поле ввода задается значение, которое будет использоваться в качестве порога после нажатия на кнопку «ОК». Ниже поля ввода приведено числовое значение текущего порога блока. При помощи кнопки «Макс» можно заполнить поле ввода порога значением, при котором обмен равен 100%. Более удобным является другой способ выбора порога. Перемещайте указатель мыши по графической области. Как только достигнете желаемого значения порога, дважды нажмите левую кнопку мыши. Значение порога будет занесено в поле ввода порога.

### **Показатель качества**

В правой части статусной строки расположен обобщенный показатель качества, который характеризует качество линии связи с блоком. Большее значение соответствует лучшему качеству линии. Данный показатель является информационным.

### **Начать от**

В рамке «Начать от» расположено несколько элементов управления, которые позволяют задать число, с которого начинается процедура подбора порога. В поле ввода можно задать начальное минимальное значение порога. Правее поля расположены несколько кнопок со стрелками для редактирования данного значения. При помощи этих кнопок можно увеличивать и уменьшать минимальное значение порога на единицу или скачками по 16.

Ниже поля ввода расположены две кнопки зеленого цвета «min» и «Зап.». Кнопка «min» устанавливает минимально возможное значение порога в число –128. По кнопке «Зап.» (Запомнить) текущее значение поля ввода запоминается и восстанавливается при следующем запуске окна подбора порога.

### **Описание меню окна подбора порога**

В верхней части окна расположено меню окна. Ниже перечислены пункты меню и приведено их описание.

#### **Файл/Записать как...**

Выбрав этот пункт меню можно сохранить все данные графической области и таблицы качества на жестком диске под любым именем.

#### **Файл/Открыть...**

При выборе этого пункта меню можно заполнить графическую область и таблицу качества данными, хранимыми в файле с любым именем на жестком диске.

#### **Файл/Выход**

Выход из окна подбора порога без изменения текущего значения порога блока.

#### **Память/Записать N1**

Все данные графической области и таблицы качества будут сохранены на жестком диске под именем «N1». Действие этого пункта меню аналогично желтой кнопке «Записать 1», расположенной правее рамки «Начать от».

#### **Память/Записать N2**

Все данные графической области и таблицы качества будут сохранены на жестком диске под именем «N2». Действие этого пункта меню аналогично желтой кнопке «Записать 2», расположенной правее рамки «Начать от».

#### **Память/Записать N3**

Все данные графической области и таблицы качества будут сохранены на жестком диске под именем «N3». Действие этого пункта меню аналогично желтой кнопке «Записать 3», расположенной правее рамки «Начать от».

#### **Память/Записать N4**

Все данные графической области и таблицы качества будут сохранены на жестком диске под именем «N4». Действие этого пункта меню аналогично желтой кнопке «Записать 4», расположенной правее рамки «Начать от».

#### **Память/Прочитать N1**

Данные графической области и таблицы качества будут заполнены значениями, хранимыми в файле на жестком диске под именем «N1». Действие этого пункта меню аналогично зеленой кнопке «Прочитать 1», расположенной правее рамки «Начать от».

#### **Память/Прочитать N2**

Данные графической области и таблицы качества будут заполнены значениями, хранимыми в файле на жестком диске под именем «N2». Действие этого пункта меню аналогично зеленой кнопке «Прочитать 2», расположенной правее рамки «Начать от».

#### **Память/Прочитать N3**

Данные графической области и таблицы качества будут заполнены значениями, хранимыми в файле на жестком диске под именем «N3». Действие этого пункта меню аналогично зеленой кнопке «Прочитать 3», расположенной правее рамки «Начать от».

#### **Память/Прочитать N4**

Данные графической области и таблицы качества будут заполнены значениями, хранимыми в файле на жестком диске под именем «N4». Действие этого пункта меню аналогично зеленой кнопке «Прочитать 4», расположенной правее рамки «Начать от».

#### **Настройка/Обычный тест**

При выборе обычного теста можно установить произвольное число попыток для определения качества.

#### **Настройка/Ускоренный тест**

В случае выбора ускоренного теста нельзя установить произвольное число попыток для определения качества. Это число всегда равно 16-ти. Однако общее время проведения подбора порога будет в 2-3 раза меньше.

#### **Настройка/Повтор**

Данная настройка позволяет организовать циклический подбор порога – сканирование заданной области продолжается непрерывно.

#### **Настройка/Пауза**

Данная настройка позволяет ввести паузу при циклический подбор порога в режиме повтор.

#### **Настройка/Автозапуск**

Данная настройка указывает стартует ли подбор порога при открывании окна

### **Задание области сканирования**

Выше графической области порога расположена область сканирования. В этой области задается начальная и конечная точка области сканирования. Для изменения левой точки щелкните левой кнопкой мышки по нужной точке в области сканирования. Для изменения правой точки щелкните правой кнопкой мышки по нужной точке в области сканирования. Для задания полного диапазона сканирования следует щелкнуть по области сканирования средней кнопкой мышки или нажать на колесо мышки.

### **Действия инженера по подбору порога при пуско-наладке блока**

При выполнении пуско-наладочных работах следует выполнить следующие действия:

1. Откройте окно подбора порога для нужного блока
2. Нажмите кнопку «Старт»
3. Дождитесь завершения прохода по всем значениям порога
4. Наведите указатель мыши на нужное значение порога в графической области и дважды нажмите левую кнопку мыши. Значение порога будет выбрано в поле ввода «Порог» (Правее кнопки «ОК»)
5. Нажмите кнопку «ОК». Новое значение порога будет записано в качестве текущего.

## **Просмотр состояния лифта**

Вызов окна просмотра состояния объекта «Блок диспетчерского контроля лифтовой» можно выполнить следующим образом:

1. Наведите указатель мыши на объект «Блок диспетчерского контроля лифтовой» и нажмите правую кнопку мыши.
2. Около места нажатия появится контекстное меню объекта:

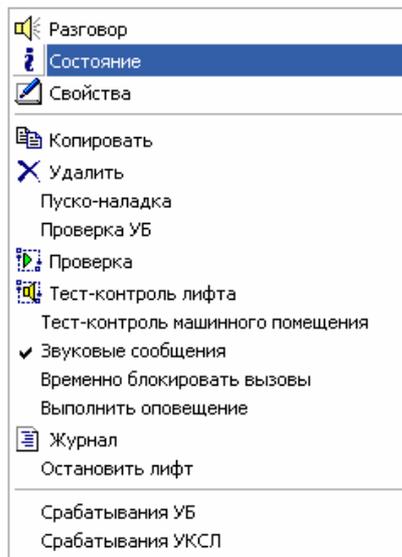


Рисунок – Контекстное меню блока БДК-Л

3. Наведите указатель мыши на строку «Состояние» в появившемся меню и нажмите левую кнопку мыши.
4. После проведенных действий в центре карты появится окно «Состояние лифта».
  - Ускоренный вызов окна просмотра состояния можно выполнить следующим образом: нажмите и удерживайте нажатой клавишу «Ctrl», расположенную в левом нижнем углу клавиатуры и щелкните правой кнопкой мыши по объекту, состояние которого нужно просмотреть. Появится окно изменения свойств «Блока диспетчерского контроля лифтового».

## Окно состояния лифта

Окно состояния лифта появляется в центре карты и является немодальным, т.е. одновременно с окном состояния лифта можно вызвать любое другое окно. Внешний вид окна состояния лифта приведен на рисунке:



Рисунок – Окно «Состояние лифта»

### Сигналы лифта

В рамке «Сигналы лифта» приведено состояние отдельных сигналов лифтового шкафа, подсоединенных к блоку БДК-Л. Сигналы, отображаемые в рамке, могут иметь различные названия в зависимости от типа лифта. Правее каждой надписи расположен индикатор наличия напряжения. Зеленый цвет обозначает наличие напряжения, а серый цвет индикатора обозначает отсутствие напряжения.

### Аварии лифта

В рамке «Аварии лифта» приведено состояние сигналов аварий лифта. Сигналы, отображаемые в рамке, могут иметь различные названия в зависимости от типа лифта. Правее каждой надписи расположен индикатор наличия аварии. Красный цвет обозначает наличие аварии по соответствующему сигналу, а серый цвет индикатора обозначает отсутствие аварии. Желтый цвет обозначает что аварийная ситуация по сигналу возникла, но в свойствах лифта данная авария отключена. Формирование аварий по сигналам лифта выполняется на уровне лифтового контроллера блока БДК-Л.

### Параметры

Ниже рамки «Аварии лифта» расположена область параметров лифта:

<b>Температура</b>	мгновенное значение температуры, считанное с температурного датчика без дополнительной программной обработки
<b>Напряжение питания</b>	реальное значение напряжения +110 вольт, используемого для питания шкафа управления лифтом
<b>Напряжение Ui (код)</b>	код из канала аналого-цифрового преобразователя, используемого для работы «устройства безопасности». Данный параметр предназначен для специалистов по пуско-наладке системы
<b>Частота оптического датчика</b>	частота в герцах сигнала оптического датчика, используемого для работы «устройства контроля скорости лифта». Данный параметр предназначен для специалистов по пуско-наладке системы
<b>Цепь безопасность (код)</b>	Данный параметр предназначен для специалистов по пуско-наладке системы.
<b>Счетчик фазы А</b>	индикатор наличия фазы «А» питания лифта (~380 вольт). Если фаза присутствует, то отображается текст «Норма». При пропадании фазы

	отображается текст «Ошибка». Отображаемые после надписи цифры пропорциональны величине фазного напряжения
<b>Счетчик фазы В</b>	индикатор наличия фазы «В» питания лифта (~380 вольт). Если фаза присутствует, то отображается текст «Норма». При пропадании фазы отображается текст «Ошибка». Отображаемые после надписи цифры пропорциональны величине фазного напряжения
<b>Счетчик фазы С</b>	индикатор наличия фазы «С» питания лифта (~380 вольт). Если фаза присутствует, то отображается текст «Норма». При пропадании фазы отображается текст «Ошибка». Отображаемые после надписи цифры пропорциональны величине фазного напряжения
<b>Порядок фаз</b>	индикатор правильности порядка фаз «АВС» питания лифта (~380 вольт). Если порядок правильный, то отображается текст «Норма». При нарушении порядка фаз или пропадании любой фазы отображается текст «Ошибка». Отображаемые после надписи цифра предназначена для специалистов по пуско-наладке системы

Таблица – Область параметров лифта

### **УБ включено**

Индикатор «УБ включено» отображает состояние тумблера блокировки устройства безопасности на лицевой панели блока БДК-Л («Ревизия»). Зеленый цвет индикатора отображается в том случае, когда тумблер находится в нормальном положении. Серый цвет индикатора обозначает, что блок находится в режиме «Ревизия» - устройство безопасности отключено.

### **Работоспособность охр.шлейфов**

Блок БДК-Л выполняет контроль электронной схемы, измеряющей состояние охранных шлейфов. При выходе электронной схемы из строя индикатор «Работоспособность охр.шлейфов» отображается серым цветом. Рабочее состояние схемы отображается зеленым цветом индикатора. В случае появления неработоспособности следует заменить плату «Контроллер связи» блока диспетчерского контроля лифтового.

### **Обмен с лифтовым контроллером**

В процессе работы происходит постоянный обмен между двумя контроллерами, входящими в состав блока БДК-Л. Данный индикатор отражает работоспособность канала обмена информацией между «Контроллером связи» и «Лифтовым контроллером». Зеленый цвет индикатора указывает на нормальный процесс обмена. Серый цвет индикатора говорит об отключении питания лифта или о неисправности платы «лифтового контроллера». В этом случае следует заменить плату «Лифтового контроллера».

### **Вызовы:**

Правее надписи «Вызовы:» расположены четыре индикатора, которые отображают текущее состояние кнопок вызова, подсоединенных к блоку БДК-Л. Зеленый цвет индикатора соответствует разомкнутой кнопке вызова. Красный цвет индикатора соответствует замыканию кнопки вызова.

При наведении мышкой на индикатор возникает дополнительная подсказка о расположении и назначении кнопки вызова.

### **Охраны:**

Правее надписи «Охраны:» расположены пять индикаторов, которые отображают текущее состояние пяти охранных шлейфов, подсоединенных к блоку БДК-Л. Правый индикатор соответствует шлейфу номер один. Левый индикатор соответствует шлейфу номер пять. Зеленый цвет индикатора соответствует замкнутому состоянию шлейфа. Красный цвет индикатора соответствует разомкнутому состоянию шлейфа.

При наведении мышкой на индикатор возникает дополнительная подсказка о расположении шлейфа.

### **Реле:**

Правее надписи «Реле:» расположены шесть индикаторов, которые отображают текущее состояние шести реле управления, входящих в состав блока БДК-Л. Зеленый цвет индикатора соответствует включенному реле. Серый цвет индикатора соответствует выключенному стоянию реле. В нормальном состоянии (нет разговора с блоком) все индикаторы должны быть серыми.

При наведении мышкой на индикатор реле возникает дополнительная подсказка о назначении реле БДК-Л.

### **Кнопка «Вызов»**

Кнопка предназначена для начала переговоров с машинным помещением или кабиной лифта.

## **Кнопка «Заккрыть»**

Кнопка предназначена для закрывания окна состояния лифта.

## **Работа с аудиозаписями**

Для вызова окна аудиозаписей выберите пункт меню «Окно/Аудиозаписи...»

Появится окно «Аудиозаписи» в котором можно прослушать сделанные аудиозаписи и удалить ненужные.

## **Окно «Аудиозаписи»**

Внешний вид окна «Аудиозаписи» представлен на рисунке:

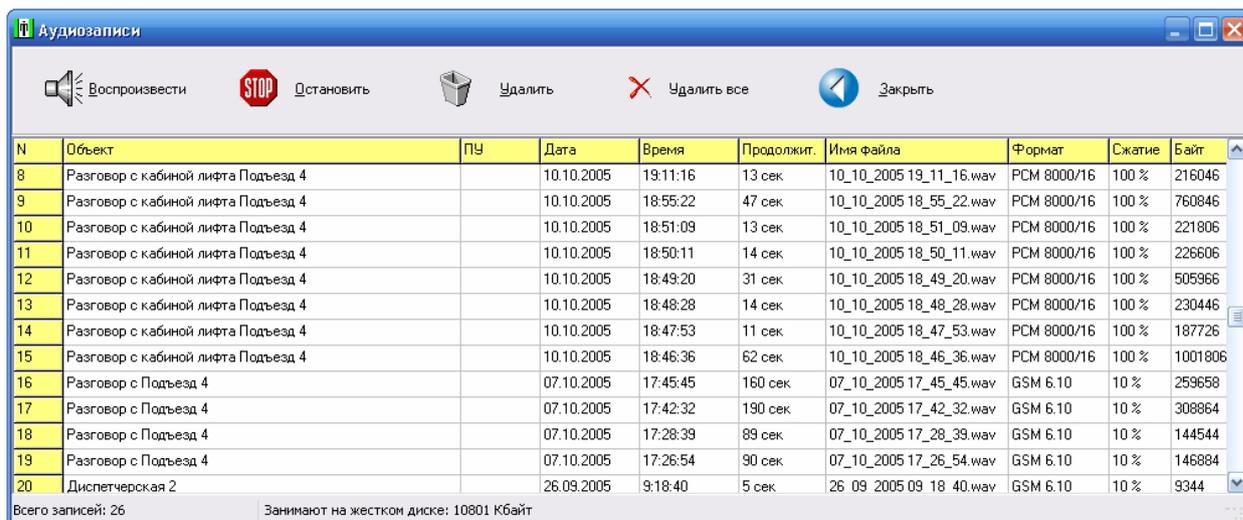


Рисунок – Окно «Аудиозаписи»

## **Таблица разговоров**

Большую часть экрана занимает таблица, которая содержит список всех переговоров, записанных на жесткий диск. Каждая строка таблицы содержит информацию об одном записанном переговоре. Строки в таблице упорядочены в порядке убывания времени. Таблица содержит следующие поля:

Поле	Описание
<b>N</b>	Номер по порядку. Младшие номера соответствуют последним разговорам
<b>Объект</b>	Содержит название объекта (адрес)
<b>ПУ</b>	Дополнительная информация о переговорном устройстве
<b>Дата</b>	Дата записи разговора
<b>Время</b>	Время записи разговора
<b>Продолжительность</b>	Продолжительность разговора в секундах
<b>Имя файла</b>	Имя звукового файла, содержащего данные разговора
<b>Формат</b>	Использованный аудиокодек для сжатия звуковых данных
<b>Сжатие</b>	Степень сжатия относительно исходного звукового файла без сжатия. 100 % соответствует отсутствию сжатия. 20% соответствует сжатию звуковых данных в 5 раз
<b>Байт</b>	размер файла в байтах

Для выбора строки (разговора) в качестве текущей просто щелкните левой кнопкой мышки на нужной строке.

## **Кнопка «Воспроизвести»**

При нажатии на кнопку «Воспроизвести» текущий разговор будет воспроизводиться через звуковые колонки.

### **Кнопка «Остановить»**

При нажатии на кнопку «Остановить» текущий прослушиваемый разговор будет остановлен.

### **Кнопка «Удалить»**

После нажатия на кнопку «Удалить» текущий разговор (текущая строка таблицы разговоров) будет удалена и связанный с ней звуковой файл будет удален.

### **Кнопка «Удалить все»**

При помощи кнопки «Удалить все» можно удалить все разговоры, записанные на жестком диске. При большом количестве записываемых переговоров процедуру удаления накопленных записей следует производить ежемесячно.

### **Кнопка «Заккрыть»**

Кнопка предназначена для закрывания окна аудиозаписей.

### **Статусная строка**

В нижней части окна приведена статистическая информация по аудиозаписям:

Общее количество хранимых аудиозаписей – «Всего записей: 5».

Объем занимаемого места на жестком диске – «Занимают на жестком диске: 2627 Кбайт».

### **Контекстное меню окна «Аудиозаписи»**

Для вызова контекстного меню щелкните левой кнопкой мышки по таблице разговоров.

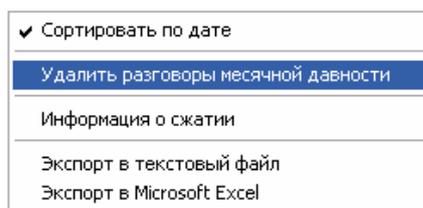


Рисунок – Контекстное меню окна «Аудиозаписи»

Появится меню со следующими настройками и функциями:

<b>Пункт меню</b>	<b>Пояснение</b>
<b>Сортировать по дате</b>	Задает, нужно ли сортировать аудиозаписи в таблице по дате разговоров. Установка данной опции требует выполнения процедуры сортировки, что требует времени – при открытии окна увеличивается время сообщения «Подготовка просмотра записей»
<b>Удалить разговоры месячной давности</b>	Удаляются все разговоры, записанные более месяца назад
<b>Информация о сжатии</b>	Данный пункт действует по отношению к текущей выбранной записи – показывается дополнительная информация о времени, которое потребовалось для сжатия звуковых данных
<b>Экспорт в текстовый файл</b>	Выполняется запись таблицы разговоров в файл с расширением «ТХТ»
<b>Экспорт в Microsoft Excel</b>	Выполняется запись таблицы разговоров в файл с расширением «CSV» и последующее открытие файла в программе Microsoft Excel

## Работа с журналом

Для вызова окна аудиозаписей выберите пункт меню «Окно/Общий журнал...»

Появится окно «Журнал» в котором можно просмотреть записи о работе системы, сделанные программой.

## Окно «Журнал»

Внешний вид окна «Журнал» представлен на рисунке:

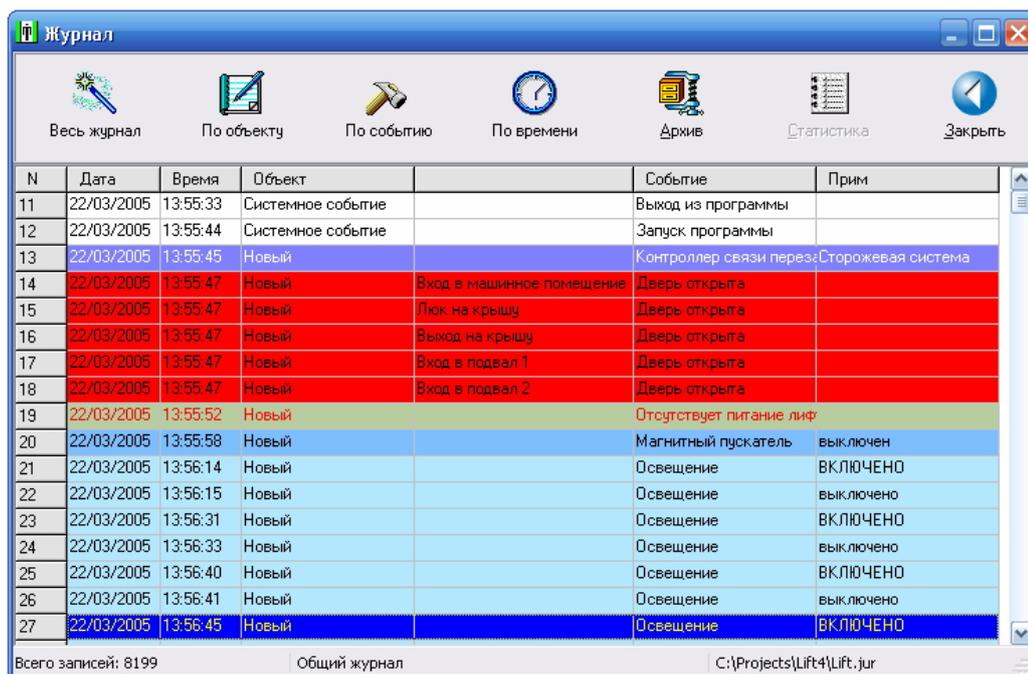


Рисунок – Окно «Журнал»

### Записи журнала

Большую часть экрана занимает таблица записей журнала, которая содержит все записи журнала. Каждая строка таблицы содержит информацию об одной записи. Строки в таблице упорядочены в порядке возрастания времени. Таблица содержит следующие поля:

Поле	Описание
<b>N</b>	Номер по порядку. Младшим номерам соответствуют более ранние записи в журнале
<b>Дата</b>	Дата выполнения записи в журнал
<b>Время</b>	Время выполнения записи в журнал
<b>Объект</b>	Содержит название объекта (адрес)
<b>Событие</b>	Текст события
<b>Прим</b>	Примечание, дополнительная информация по событию. Зависит от типа события

Для выбора строки в качестве текущей щелкните левой кнопкой мышки на нужной строке. Текущая строка выделяется ярко-синим цветом. Для просмотра журнала используйте кнопки «Вниз» и «Вверх» на клавиатуре компьютера

### Кнопка «Весь журнал»

Кнопка позволяет показать все содержимое журнала. Кнопка используется после проведения выборки по объекту, времени или по событию.

### **Кнопка «По объекту»**

Кнопка «По объекту» позволяет показать записи, относящиеся только к одному объекту. Для выборки событий по объекту следует щелкнуть мышкой по объекту в таблице записей (установка текущего объекта) и нажать кнопку «По объекту». В таблице записей журнала будут отображены только записи, относящиеся к текущему объекту.

Если щелкнуть по кнопке «По объекту» правой кнопкой мышки, то откроется окно со списком всех объектов. Выберите нужный объект и нажмите кнопку «Выбрать». В таблице записей журнала будут отображены только записи, относящиеся к выбранному объекту.

### **Кнопка «По событию»**

Кнопка «По событию» позволяет показать записи с одинаковыми записями в поле «Событие». Для выборки событий по объекту следует щелкнуть мышкой по объекту в таблице записей (установка текущего объекта) с нужным событием и нажать кнопку «По событию». В таблице записей журнала будут отображены только записи с указанным событием.

Если щелкнуть по кнопке «По событию» правой кнопкой мышки, то откроется окно со списком всех событий. Выберите нужное событие и щелкните по нему левой кнопкой мыши. В таблице записей журнала будут отображены только записи с выбранным событием.

### **Кнопка «По времени»**

Кнопка «По времени» позволяет просмотреть события, произошедшие в определенный день. Нажмите кнопку «По времени». Появится окно «Выбор дня». Выберите нужный месяц и день и нажмите кнопку «ОК». В таблице записей журнала будут отображены только записи, сделанные в указанный день.

### **Кнопка «Архив»**

Кнопка «Архив» позволяет просмотреть события, хранящиеся в отдельном файле журнала. Нажмите кнопку «Архив». Появится окно «Выбор архива». Выберите нужный архив щелчком мыши и нажмите кнопку «Показать». В таблице записей журнала будут отображены только записи из выбранного архива.

### **Кнопка «Заккрыть»**

Кнопка предназначена для закрывания окна журнала.

## Устранение неисправностей

### Определение неисправности различных блоков

Неисправность отдельного оборудования в информационно-питающей линии определяется визуально, по наличию красного диагонального креста, перечеркивающего объект на карте. Например:



Рисунок – Отображение неисправности блока БДК-Л на карте

Следует разделить данные неисправности на две части:

- I. Неисправны сразу все блоки определенного луча. Наиболее вероятными причинами могут быть:
  - неисправен блок БКД, связанный с данным лучом
  - неисправен последовательный порт компьютера, связанный с данным лучом
  - неисправен соединительный кабель компьютер – БКД
  - отсутствует питание (~220 В) блока БКД
  - замыкание или обрыв в информационно-питающей линии
- II. Неисправен один или несколько блоков в линии (но не все сразу). Наиболее вероятными причинами могут быть:
  - неисправен один из блоков и информационно-питающей линии
  - замыкание или обрыв в информационно-питающей линии

Далее рассмотрена последовательность действий для этих групп неисправностей.

- I. Неисправны сразу все блоки определенного луча. Выполните следующие действия:
  1. Определите неисправность БКД по разделу: «Быстрое определение неисправности блоков БКД»
  2. Определите неисправность БКД по разделу: «Просмотр журнал подключения БКД»
  3. Определите неисправность БКД по разделу: «Определение работоспособности блоков БКД»
  4. Замените блок БКД или плату луча в случае определения неисправности
  5. Замените соединительный кабель компьютер – БКД заведомо исправным
  6. Проверьте наличия питания блока БКД по сети 220 вольт.
  7. В блоке БКД-Л-4 на аудио плате должны гореть все три светодиода питания. При отсутствии свечения светодиодов проверьте исправность блока питания БКД-Л-4 методом замены блока питания на исправный блок. В блоке БКД-М должны загораться желтые светодиоды выполнения обмена. При отсутствии свечения замените блок на исправный.
  8. Если неисправность не найдена, то следует проверить информационно питающую линию на замыкание или обрыв. Для этого померьте напряжение вольтметром постоянного тока напряжение в линии на блока грозозащиты и далее на всех участках информационно-питающей линии. Во всех точках должно быть напряжение 24-28 вольт. Отсутствие напряжения в какой-либо точке говорит об обрыве в линии. Отсутствие напряжения на выходе блока БКД (при его исправности) говорит о наличии замыкания в линии. Отсоедините блок БКД от ближайшего блока грозозащиты. Проверьте омметром наличие замыкания в линии. Для определения точки замыкания следует рассоединять линию в отдельных точках и проверять сопротивление в обоих участках линии.
- II. Неисправен один или несколько блоков в линии (но не все сразу). Выполните следующие действия:
  1. Выполните проверку порога неисправного блока, так, как описано в разделе «Подбор порога

опроса»

2. Выполните поиск новых блоков, так, как описано в разделе «Поиск новых блоков». Возможны случаи «слетания» адреса блока в информационно-питающей линии на ремонтный адрес 255. Если найден несуществующий блок с адресом 255 того же типа, что и неработающий блок, то следует задать утерянному блоку адрес 255 и далее восстановить старый утерянный адрес.

3. Замените неисправный блок заведомо исправным. Проконтролируйте отсутствие в требуемом луче блоков с адресом, совпадающим с адресом нового блока (раздел «Проверка наличия в луче блока с определенным адресом»).

4. Если после установки нового заведомо исправного блока работоспособность не восстановлена, то проверьте отсутствие замыкания или обрыва в линии.

## Быстрое определение неисправности блоков БКД

Для быстрого определения неисправности блоков БКД в нижней правой части основного окна программы отображается набор светодиодов. Каждый светодиод отражает текущее состояние одного блока БКД.



Рисунок – Светодиоды состояния блоков БКД

Каждый светодиод отображает состояние одного блока БКД изменением цвета. В нормальной ситуации светодиод мигает зеленым и ярко-зеленым цветом с периодом опроса данного БКД.

Красный цвет светодиода обозначает неисправность блока БКД. Для самого левого светодиода появление красного цвета так же обозначает ошибку в БЕЙСИК – программе. Поэтому если красным цветом загорелся левый светодиод, то следует вызвать окно работы с программой (Ctrl+P) и снять переключатель «Разрешить работу программы». Если светодиод по-прежнему светится красным цветом, то это обозначает неисправность блока БКД. Для уточнения характера неисправности следует просмотреть журнал подключения блоков БКД, а так же определить работоспособность блоков БКД.

## Просмотр журнал подключения БКД

Для просмотра журнал подключения блоков БКД следует:

1. В основном меню программы выбрать «Настройка/Параметры программы...»:

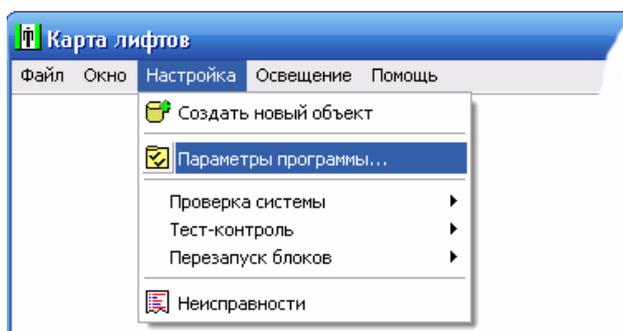


Рисунок – Вызов окна «Параметры программы»

2. В открывшемся окне перейдите на вкладку «БКД»:

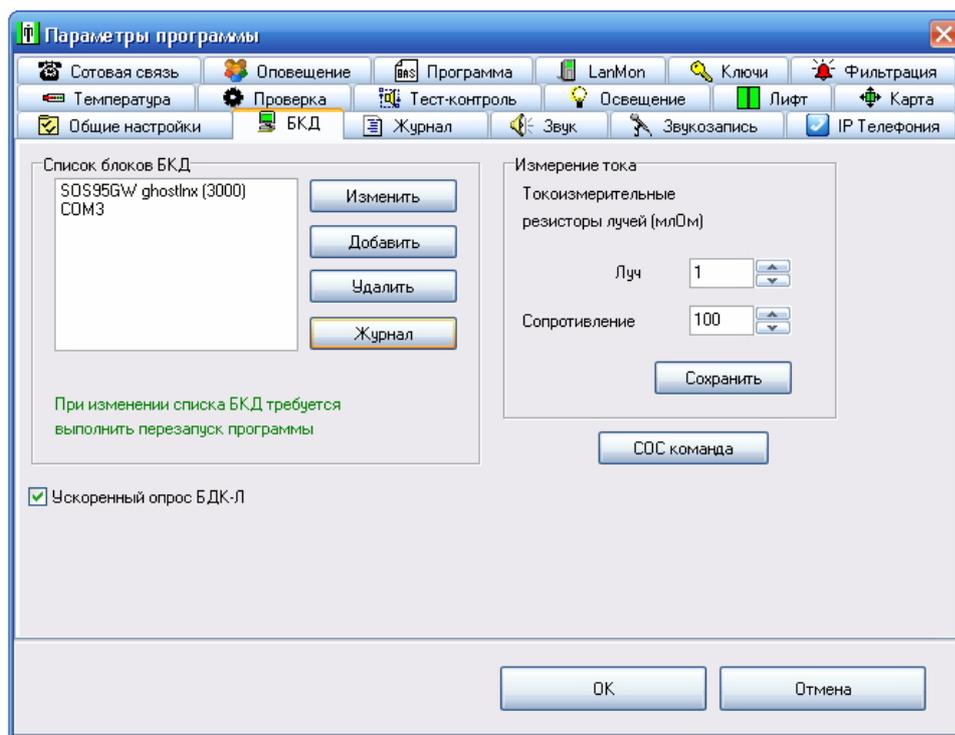


Рисунок – Вкладка «БКД» окна «Параметры программы»

3. Нажмите кнопку «Журнал»:

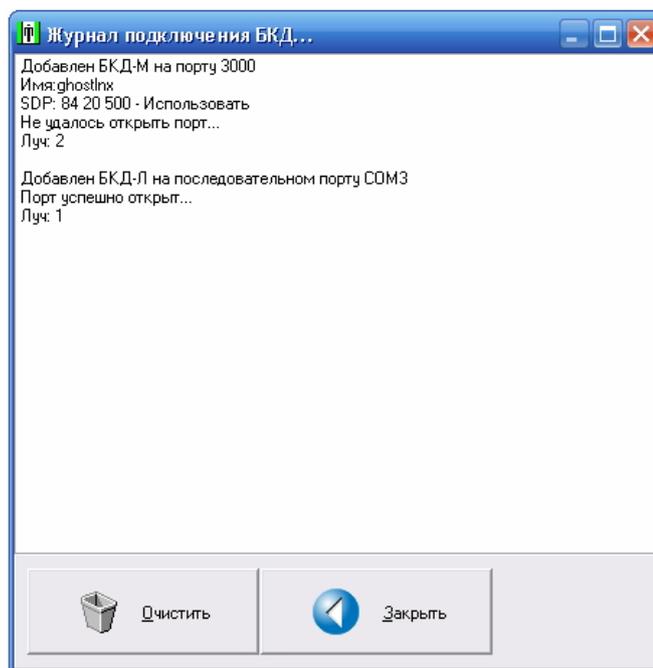


Рисунок – Журнал подключения блоков БКД

Журнал является текстовым документом. Внимательно просмотрите текст журнала для определения возникших проблем. Порядок записей в журнал о создании лучей блоков БКД соответствует порядку расположения светодиодов состояния БКД (см. раздел выше) слева направо.

В приведенном на рисунке журнале видна надпись «Не удалось открыть порт», относящаяся к блоку БКД-М, связанному с лучом номер 2. Т.к. блок БКД-М добавлен на порту 3000, то можно сделать вывод о том, что шлюз «Sos95gw» работоспособен, а вот сам блок БКД-М либо отсутствует, либо неисправен.

Исправные блоки БКД рекомендуется так же проверять способом, описанным в следующем разделе.

## Определение работоспособности блоков БКД

Определение работоспособности блоков БКД следует выполнять следующим образом:

1. В основном меню программы выберите «Окно/Состояние БКД»

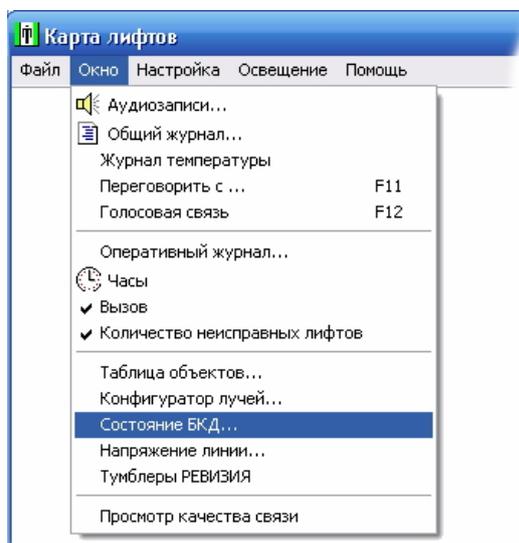


Рисунок – Открытие окна «Состояние БКД»

2. Откроется окно «Состояние БКД»:



Рисунок – Окно «Состояние БКД»

В окне «Состояние БКД» отображается текущее состояние всех лучей БКД, зарегистрированных в программе Lift4. У блоков БКД-Л-4 может быть до 4-х лучей. Блоки БКД-М связаны только с одним лучом.

На рабочем поле окна отображается от одной до четырех рамок, причем каждая рамка связана с одним лучом. В заголовки рамки указан номер луча. На рисунке выше показаны два луча «Луч 1» и «Луч 2». В рамке луча отображаются дополнительные элементы управления, связанные с лучом. Прежде всего, это светодиод зеленого или красного цвета. Красный цвет говорит о неисправности соответствующего луча и блока БКД, с которым данный луч связан. Замените плату луча блока БКД-Л-4 или замените БКД-М. Зеленый цвет светодиода говорит об исправности БКД. Однако для полноты картины следует проверить связь с блоком БКД по последовательному интерфейсу.

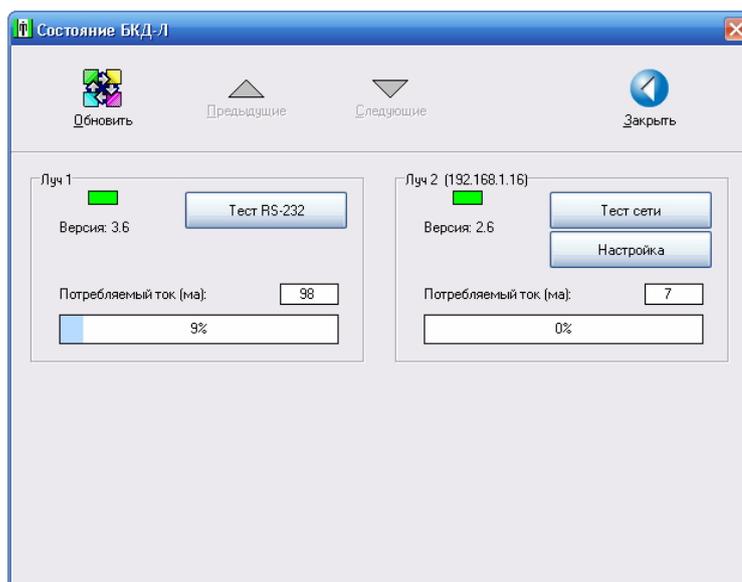


Рисунок – Окно «Состояние БКД» - оба луча исправны

- Для проверки обмена по последовательному интерфейсу нажмите кнопку «Тест RS-232» у исправного луча БКД-Л-4. Появится окно тестирования последовательного канала:



Рисунок – Тестирование последовательного канала связи с лучом

В данном тесте программа непрерывно посылает изменяющиеся байты по последовательному интерфейсу и проверяет их получение обратно. Дождитесь отправки 256 байтов и убедитесь в отсутствии ошибок. В этом случае луч считается исправным на уровне взаимодействия компьютер-плата луча БКД-Л-4. При появлении ошибок следует заменить плату луча или весь блок БКД-Л-4.

- Блоки БКД-М подсоединяются по более сложному интерфейсу – сначала выполняется передача по локальной сети UDP пакетов и далее по последовательному интерфейсу данные попадают в БКД-М. Поэтому кнопка тестирования связи с блоком БКД-М называется «Тест сети». После нажатия кнопки «Тест сети» выполняется тест связи с блоком БКД-М. При появлении ошибок во время выполнения теста замените БКД-М.

Таким способом определяется заведомо неисправные блоки БКД. Однако возможна ситуация, когда несмотря на исправность работы с компьютером по последовательному интерфейсу, блок БКД неисправен. В этом случае наиболее вероятно, что неисправна схема формирования сигнала в информационно-питающей линии СОС-95. Признаком такой неисправности является то, что блок БКД «не видит» ни одного блока в информационно-питающей линии. Для определения этой ситуации выполните «Поиск новых блоков» так, как описано в следующем разделе, но сняв переключатель «Только новые». Если ни один блок в линии не будет найден - замените блок БКД при возникновении такой неисправности.

## Поиск новых блоков

Для выполнения поиска новых блоков следует выполнить следующие действия:

1. В основном меню программы выберите «Окно/Таблица объектов»

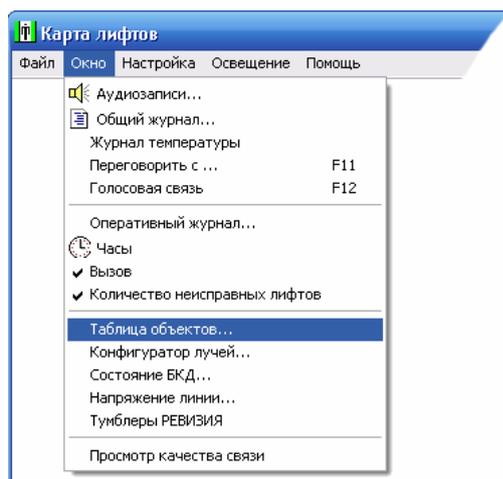


Рисунок – Открытие окна «Таблица объектов»

2. Появится окно «Таблица объектов»

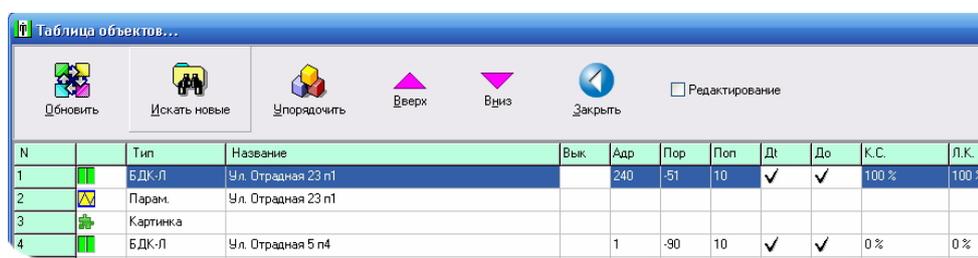


Рисунок – Окно «Таблица объектов»

3. Нажмите кнопку «Искать новые»

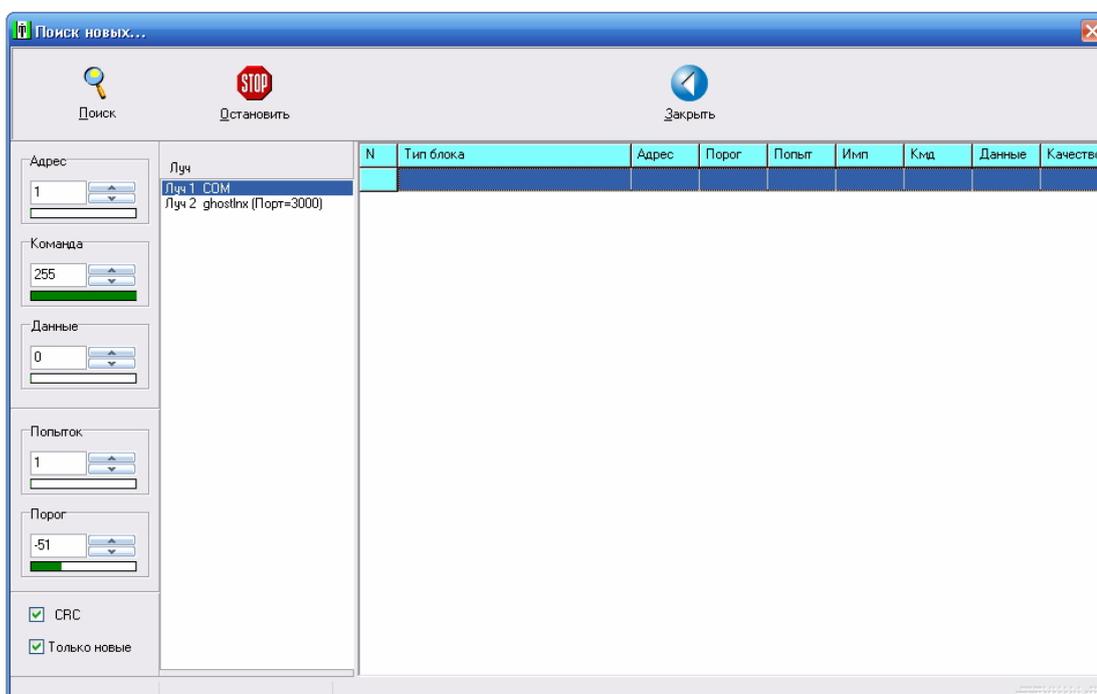


Рисунок – Окно «Поиск новых»

Выберите нужный луч в списке лучей и нажмите кнопку «Поиск». Будет выполнен поиск в выбранном луче и в списке найденных блоков (справа от списка лучей) появятся новые блоки найденные при выполнении поиска:

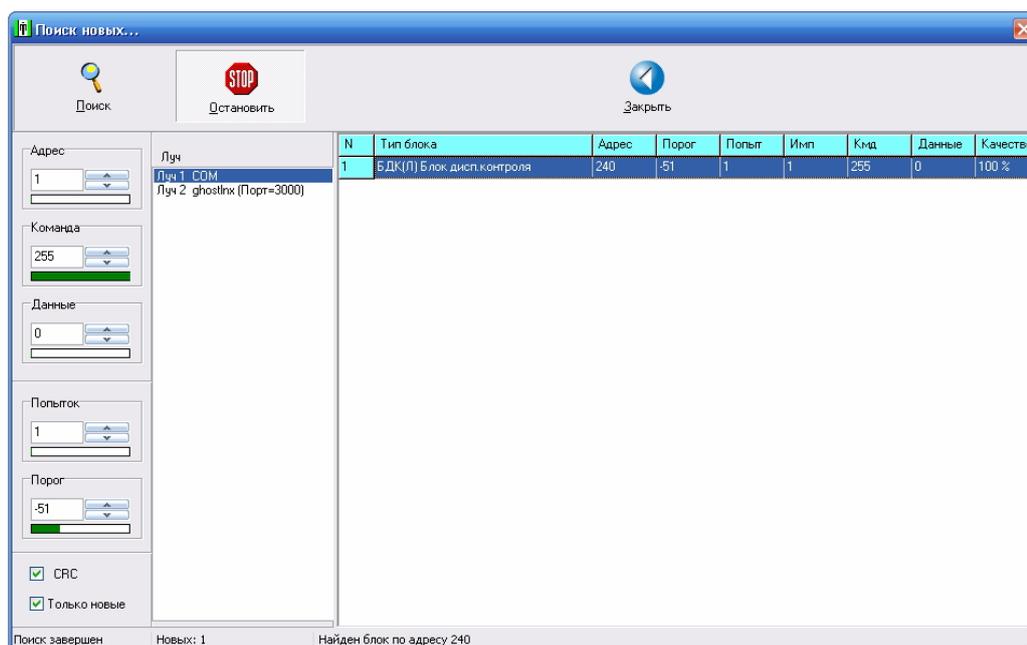


Рисунок – Результат поиска – найден один блок

Возможны случаи забывания («слетания») адреса блока в информационно-питающей линии на ремонтный адрес 255. Если найден несуществующий блок с адресом 255 того же типа, что и неработающий блок, то следует задать утерянному блоку адрес 255 и далее восстановить старый утерянный адрес.

Для некоторых блоков из системы СОС-95 может потребоваться снять переключатель «CRC», т.к. эти блоки не используют протокол SOS-CRC.

Если убрать переключатель «Только новые», то в таблице найденных блоков будут показаны все найденные блоки линии.

## Проверка наличия в луче блока с определенным адресом

1. Вызовите окно «Таблица объектов» так, как описано в разделе «Поиск новых блоков»
2. Щелкните правой кнопкой мышки по таблице объектов и в появившемся контекстном меню выберите «Просмотр адресов»:

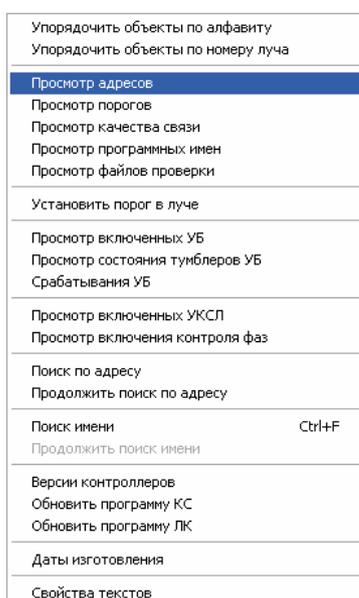


Рисунок – Контекстное меню окна «Таблица объектов»

3. Появится окно просмотра всех блоков в информационно-питающей линии с указанием имени блока, адреса и номера луча. Данные в таблице упорядочены по возрастанию адреса.

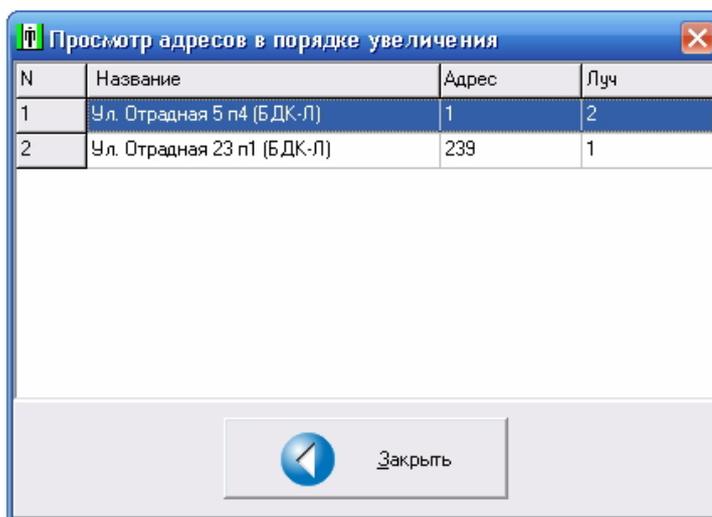


Рисунок – Просмотр адресов в порядке увеличения

4. Просмотрите список и убедитесь, что новый блок имеет адрес, не совпадающий с уже установленными блоками в заданном луче.

## Дополнительные возможности БКД-М

Для перехода к окну дополнительных возможностей БКД-М нажмите кнопку «Настройка» (см. рисунок – Окно «Состояние БКД» - оба луча исправны). Появится окно, приведенное на следующем рисунке:

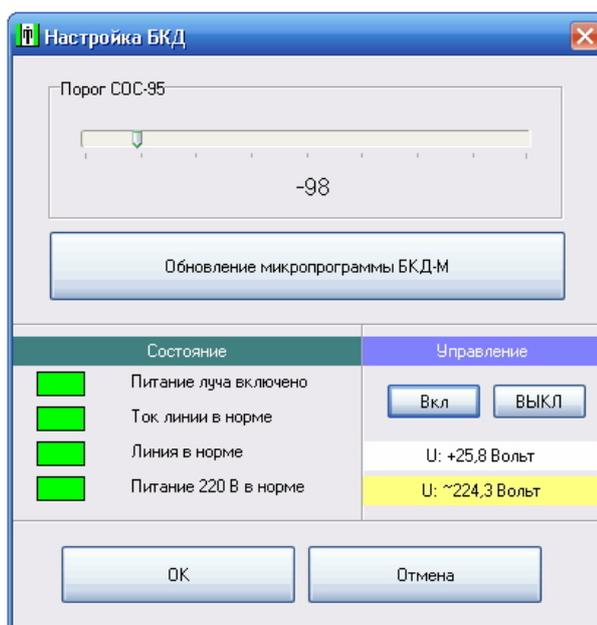


Рисунок – Окно «Настройка БКД»

Для обновления программы БКД-М нажмите кнопку «Обновление микропрограммы БКД-М». Действия по перепрошивке описаны в разделе «Перепрошивка БКД-М»

Текущее состояние БКД-М показывается четырьмя «светодиодами» и надписями правее в разделе «Состояние».

**Питание луча включено/Питание луча ВЫКЛЮЧЕНО** – показывает, что выходное напряжение в информационно-питающей линии (ИПЛ) присутствует или отсутствует. В разделе «Управление» показаны две кнопки «Вкл» и «ВЫКЛ», предназначенные для подачи и снятия питания в ИПЛ соответственно.

**Ток линии в норме/Перегрузка в линии (ток > 1.5 А )** – показывает превышение максимально допустимого тока в ИПЛ.

**Линия в норме/Короткое замыкание в линии** - помимо превышения тока БКД-М так же распознает ситуацию короткого замыкания в линии.

**Питание 220 В в норме/Отсутствует питание 220 В** – показывает наличие/отсутствие питания 220 вольт для питания блока БКД-М. БКД-М может получать питание из ИПЛ. В этом случае при отсутствии питания 220 вольт и выдается такое сообщение.

В правой части окна показано текущее напряжение на выходе блока БКД-М: **U:+25.8 Вольт** (Надпись на белом фоне) и текущее питающее напряжение сети 220 вольт: **U: ~224.3 Вольт** (Надпись на желтом фоне) .

Кнопка «ВЫКЛ» позволяет снять питание с луча БКД-М без физического отключения блока от сети 220 вольт. Кнопка «Вкл» позволяет подать питание на луча БКД-М. Используйте кнопки для временного отключения и включения напряжения в информационно-питающей линии.

# Встроенный интерпретатор BASIC

## Допустимые операторы

Переменная = выражение  
**IF ,THEN, ELSE,ENDIF**  
**GOTO** метка  
**GOSUB** метка  
**RETURN**  
**FOR, TO, STEP**  
**NEXT**  
**PRINT**  
**END**  
**REM**  
**OPEN**  
**CLOSE**  
**INPUT**  
**OUTPUT**  
**RUN**  
**DIM AS**  
**WHILE, WEND**

## Составные операторы

Операторы допускается располагать в одной строке, разделяя их символом точка с запятой:

Пример:

```
A=5; B=6;
```

## Метки

Метки используются в операторах **GOTO** и **GOSUB**. Метка состоит из двух частей:

1. Собственно метка – Начинается с символа латиницы и заканчивается двоеточием. **После двоеточия необходимо поставить точку с запятой.**
2. Ссылка на метку – то же самое, но двоеточие и символ точка с запятой отсутствуют

Пример:

```
IF Biu2.UPR2<>0 THEN GOTO TryToRestart
...
ENDIF
TryToRestart:;
Var=2;
GOSUB MyProg
...
MyProg:; Var=Var+1; RETURN
...
END
```

## Переменные

### До версии 5.49

Переменной считается любой текст, начинающийся с буквы латинского алфавита.

В качестве имени переменной используется только первая буква текста. Переменная нечувствительна к регистру текста. Т.е. переменные Auto, a2000, A, Aqwerty ссылаются на одну и ту же переменную 'a'.

## Начиная с версии 5.49

Переменной считается любой текст, начинающийся с буквы латинского алфавита и длиной не более 15 символов. Переменная нечувствительна к регистру текста. Т.е. переменные Auto, auto,AUTO, AuTo ссылаются на одну и ту же переменную.

Различаются три типа переменных:

Тип	Описание
Целочисленная	Целое знаковое 32-разрядное число
Плавающая	8 байтовое число с плавающей точкой
Строковая	Набор символов длиной до 256 байт

Переменная может быть описана явно следующим способом с использованием оператора **DIM**:

```
DIM name1[=value1],name2[=value2]...name16[=value16] AS INTEGER
```

```
DIM name1[=value1],name2[=value2]...name16[=value16] AS STRING
```

```
DIM name1[=value1],name2[=value2]...name16[=value16] AS FLOAT
```

В строке определения имен допускается указывать от одного до 16-ти имен переменных, разделенных запятыми. Всем переменным будет присвоено нулевое значение, если не указан знак равенства и значение при старте

Пример:

```
rem Определение целочисленной переменной MyInt с присвоением нулевого знач.
```

```
DIM MyInt AS INTEGER
```

```
rem Определение плавающей переменной MyFloat
```

```
DIM MyFloat AS FLOAT
```

```
rem Определение строковой переменной My
```

```
DIM My AS STRING
```

```
rem Определение трех переменных с присвоением значений.
```

```
DIM Var1=2000,Var2=100,Var3=65536 AS INTEGER
```

```
rem Определение строковых переменных с инициализацией
```

```
DIM str="Неизвестно",dtr="Выполнено" AS STRING
```

```
rem Определение пяти нулевых целочисленных переменных
```

```
DIM K1,K2,Cnt,Result,Summa AS INTEGER
```

Альтернативным вариантом является неявное определение переменных по заданному присвоению. Если переменной присваивается целочисленное значение, то переменная определяется как целочисленная:

```
rem Определение целочисленной переменной Var в момент присвоения
```

```
Var=2005
```

Если происходит присвоение плавающего значения, то переменная автоматически определяется как плавающая:

```
rem Определение плавающей переменной FVar в момент присвоения
```

```
FVar=3.14156
```

Если происходит присвоение строкового значения, то переменная автоматически определяется как строковая:

```
rem Определение строковой переменной SVar в момент присвоения
```

```
SVar="Это строковая переменная"
```

Аналогичным образом происходит определение типа при присвоении значения уже определенных переменных:

```
DIM s AS STRING
```

```
s="ASDF123"
```

```
rem Определение строковой переменной SVar в момент присвоения
```

```
SVar=s
```

```

DIM k as INTEGER
DIM f as FLOAT
rem Определение плавающей переменной FVar в момент присвоения
rem Результат сложения плавающей и целой переменных – плавающее число
FVar=f+k

```

После определения переменной явным или неявным образом тип переменной изменить невозможно.

## Преобразование типов

При присвоении переменных одного типа другому происходит преобразование типов. Например, если требуется присвоить целочисленной переменной значение плавающей переменной, то плавающее значение будет преобразовано в целочисленное методом отбрасывания дробной части:

```

DIM MyInt AS INTEGER
DIM MyFloat AS FLOAT
MyFloat=3.14156
MyInt= MyFloat
rem MyInt присвоено значение 3

```

При присвоении плавающей переменной целочисленного значения происходит преобразование целого числа в плавающее.

При присвоении значения строковой переменной целочисленного или плавающего значения происходит «строковая печать» значения в этой переменной. Формат печати задается системной переменной @Format. При присвоении плавающей переменной строкового значения происходит преобразование плавающего числа из текстовой строки. Аналогично при присвоении целочисленной переменной строкового значения происходит преобразование целого числа из текстовой строки.

При выполнении арифметических операций с разнотипными переменными происходит автоматическое преобразование типов. Результатом арифметической операции +-\* / целого и плавающего числа является плавающее число. Результатом поразрядной бинарной операции &| всегда является целое число. Результат возведения в степень - всегда плавающее число.

Некоторые математические операции со строками неприменимы. Например, невозможно определить что такое деление или умножение для строк.

Следует отметить особенность выполнения арифметических операций со строками – если первый операнд является строкой, то и результат арифметической операции тоже строка.

```

DIM s AS STRING
f=3.14
a="f="+f
rem a – строковая переменная равная «f=3.14»

```

## Системные переменные

Если имя переменной начинается с символа '@', то такая переменная является системной – ее значение, как правило, доступно только для чтения и результатом чтения является определенное значение, которое зависит от имени переменной:

Имя	Значение
@TickCount	Число миллисекунд, прошедшее с момента запуска Windows
@Time	Число секунд, прошедшее с 1 го января 1970 г
@Hour	Часы текущего времени (0-23)
@Min	Минуты текущего времени (0-59)
@Sec	Секунды текущего времени (0-59)
@Year	Текущий год
@Month	Текущий месяц (1-12)
@Day	Текущий день(1-31)
@WeekDay	Текущий день недели (1-7) 1-вск,2-пон... (с версии 5.29)
@Random	Случайное число в диапазоне от 0 до числа RandomMax, устанавливаемого

	записью в переменную <b>@Random</b> . Например, для генерации случайного числа от 0 до 2000 следует выполнить: <b>@Random=2000</b>
<b>@MessageColor</b>	Цвет фона окна сообщения программы. Цвет окна сообщения задается следующим образом: <b>@MessageColor=0xBBGGRR</b> где BB - интенсивность синего цвета 00-FF (HEX) GG - интенсивность зеленого цвета 00-FF RR - интенсивность красного цвета 00-FF Например: <b>@MessageColor=0xFF</b> - окно сообщения ярко красного цвета <b>@MessageColor=255</b> - то же самое <b>@MessageColor=0</b> - цвет по умолчанию
<b>@PlayFile</b>	Воспроизведения любого звукового файла: <b>@PlayFile="Имя wav файла"</b>
<b>@MapCnt</b>	Количество объектов на карте
<b>@MapIndex</b>	Индекс объекта, к которому можно обратиться через системную переменную <b>@Map</b>
<b>@Map</b>	Объект на карте
<b>@MessageVisible</b>	Проверка видимости окна сообщения программы. 0-окно невидимо.
<b>@Error</b>	Результат последней операции (с файлом или устройством): 0-Успешно 1-Файл не открыт 2-Ошибка записи в файл 3-Ошибка чтения из файла
<b>@MessageButtons</b>	Количество кнопок в окне сообщения программы, может быть числом от 0 до 3. Доступна как для записи, так и для чтения. Тексты надписей на кнопках можно изменить оператором Print #6, .
<b>@MessageResult</b>	Результат закрытия окна сообщения программы. Принимает одно из следующих значений: 0-окно закрыто программно 1-пользователь нажал кнопку 1 2-пользователь нажал кнопку 2 3-пользователь нажал кнопку 3 Проверка закрытия окна выполняется чтением переменной <b>@MessageVisible</b> . Далее можно проверить переменную <b>@MessageResult</b> .
<b>@Log</b>	Выполнение записи в общий журнал программы Текстовые поля заполняются выполнением печати (PRINT) в потоки номер 6, 7 и 8. Переменная только для записи. См. раздел «Выполнение записи в общий журнал программы»
<b>@Format</b>	Задание формата вывода переменных в плавающем формате <b>@format="#"###"</b> <b>@format="00#####"</b>

## Предопределенные константы

Вы можете использовать следующие предопределенные константы:

Имя	Значение
<b>True</b>	1
<b>False</b>	0
<b>MaxInt</b>	2147483647
<b>PI</b>	Число Пи

## Числа

В программе могут использоваться целые знаковые числа размером 4 байта на число. Для задания целых чисел в программе можно использовать следующие форматы:

- **Десятичные** числа – записываются с символом **'D' (DECIMAL)** в конце или без него и содержат символы от '0' до '9'.
- **Шестнадцатеричные** числа - записываются с символом **'H' (HEX)** в конце и содержат символы от '0' до '9' и буквы от 'A' до 'F'. Допускается записывать без символа 'H' в конце, но в этом случае число должно начинаться с префикса '0x' или '0X'.
- **Восьмеричные** числа - записываются с символом **'O' (OCTAL)** в конце и содержат символы от '0' до '7'.
- **Двоичные** числа - записываются с символом **'B' (BINARY)** в конце и содержат символы '0' и '1'.

Для плавающих чисел используется 8-ми байтовое числовое представление плавающего числа. Признаком плавающего числа является наличие точки в числовой записи.

В программе могут использоваться строковые константы, которые должны обрамляться двойными кавычками.

## Функции

В данном разделе описаны встроенные функции языка БЕЙСИК, реализованные в данном интерпретаторе. Функции сгруппированы по назначению.

### Математические функции

Имя	Описание функции и пример использования
Abs(VAR)	Возвращает модуль числа VAR – типа INTEGER или DOUBLE d=-2000.0 a=500.0+Abs(d)
Atn(DBL)	Возвращает арктангенс числа (тип DOUBLE). d=Atn(0.5)
Cos(DBL)	Возвращает плавающее число, косинус плавающего числа DBL: @format="0.#####" print "c=",cos(1.507)
Exp(DBL)	Возвращает число e в степени DBL. (Возвращает тип Double). d=exp(0.9)
Fix(DBL)	Возвращает (INTEGER) целую часть числа DBL print pi-Fix(pi) gem дробная часть числа ПИ
Log(DBL)	Возвращает натуральный логарифм числа DBL. (Возвращает тип Double). d=Log(0.9)
Log10(DBL)	Возвращает десятичный логарифм числа DBL. (Возвращает тип Double). d=Log10(0.9)
Randomize()	Инициализирует генератор случайных чисел. Возвращает 1 (INTEGER) i=Randomize()
Rnd(INT)	Возвращает целое число, содержащее случайное число от 0 до INT. i=Rnd(100)
Sqr(DBL)	Возвращает корень числа DBL d=Sqr(25.0)
Sin(DBL)	Возвращает плавающее число, синус плавающего числа DBL: @format="0.#####" print "s=",sin(1.507)
Tan(DBL)	Возвращает тангенс числа DBL. d=Tan(1.6)

### Функции времени

Имя	Описание функции и пример использования
Date\$(DBL)	Возвращает строку даты из переменной времени DBL print Date\$(Now()) gem вывод строки текущей даты
DateTime\$(DBL)	Возвращает строку даты и времени из переменной времени DBL print DateTime\$(Now()) gem вывод строки текущей даты и времени
DateSerial (INT1, INT2,INT3)	Возвращает дату (DOUBLE), указанную по частям: DateSerial (year, month, day) t=TimeSerial(12,45,30) d=DateSerial(2000,3,18) s=d+t print Time\$(t)

	print Date\$(d) print DateTime\$(s)
Day(DBL)	Возвращает день месяца (INTEGER) из указанной даты DBL. i=Day(Time()) get текущий день месяца
Hour(DBL)	Возвращает количество часов (INTEGER) из параметра DBL. h=Hour(Time()) get текущий час
Minute(DBL)	Возвращает число минут (INTEGER от 0 до 59) содержащимся в параметре DBL m=Minute(Now())
Month(DBL)	Возвращает номер месяца (INTEGER) из указанной даты DBL. i=Month(Time())
MSec(time)	Возвращает количество миллисекунд(INTEGER), содержащихся в параметре DBL s=MSec(Now())
Second(time)	Возвращает количество секунд(INTEGER), содержащихся в параметре DBL s=Second(Now())
TimeSerial(INT1, INT2,INT3)	Возвращает время (DOUBLE), указанное по частям: TimeSerial(hour, minute, second) print Time\$( TimeSerial(12,45,30))
Timer()	Возвращает целое число, содержащее количество секунд, прошедших после полуночи i=Timer()
WeekDay(DBL)	Возвращает день недели (INTEGER), которому соответствует указанная дата. Возможные значения функции: 1- Воскресенье 2- Понедельник 3- Вторник 4- Среда 5- Четверг 6- Пятница 7- Суббота Print WeekDay(Now())
Year(DBL)	Возвращает количество годов (INTEGER), содержащихся в параметре даты DBL year=Year(Time())
Time\$(DBL)	Возвращает строку времени из переменной времени DBL print Time\$(Now()) get вывод строки текущего времени
Now()	Возвращает плавающее число, описывающее текущее время tm=Now()
Time()	Аналог функции Now – получение текущего времени tm=Time()

## Функции преобразования типов

Имя	Описание функции и пример использования
CDbl(VAR)	Возвращает плавающее число, полученное из VAR. Преобразование выполняется из любых типов: dim S as STRING dim D as DOUBLE S="1234.567" D=CDbl(S);
CInt(VAR)	Возвращает целое число, полученное из VAR. Преобразование выполняется из любых типов: dim S as STRING dim I as INTEGER S="1234.567" I=CInt(S);
CStr(VAR)	Возвращает строку, полученную из VAR. Преобразование выполняется из любых типов: s=CStr(237.78); print "s=",s
IsNumeric(VAR)	Возвращает 1, если VAR является числовым значением и 0 если строковым
IsString(VAR)	Возвращает 1, если VAR является строковым значением и 0 если числовым

TypeName(VAR)	Предоставляет информацию о типе переменной VAR. Возвращает одну из следующих строк: "INTEGER" "DOUBLE" "STRING"
VarType(VAR)	Возвращает целое число - тип переменной VAR. Возможные значения функции: 1-INTEGER,2-DOUBLE,3-STRING

## Строковые функции

Имя	Описание функции и пример использования
Asc(STR)	Возвращает целое число – код первого символа в строке s="QWERTY" i=Asc(s) rem код первого символ в строке i=Asc(Mid(s,2,1)) rem код второго символ в строке
InStr (STR1, STR2)	Возвращает номер позиции (INTEGER) первого вхождения строки STR2 в строку STR1 s="1234567890" print "pos=",InStr(s,"789") rem pos=7 print "pos=",InStr(s,"987") rem pos=0
StrComp(STR1, STR2)	Сравнивает две строки. Возвращает (INTEGER): <0, если STR1 < STR2 =0, если STR1 = STR2 >0, если STR1 > STR2
StrIComp(STR1, STR2)	Сравнивает две строки без учета регистра. Возвращает (INTEGER): <0, если STR1 < STR2 =0, если STR1 = STR2 >0, если STR1 > STR2
Chr\$(INT)	Возвращает строку, состоящую из одного символа, соответствующего коду INT: s="." for i=64 to 90 s=s+chr\$(i) next print s
Format(DBL,STR)	Возвращает строку с форматированным выводом плавающего числа: Format(5459.4, "####0.00") REM Возвратит "5,459.40".
Hex(INT)	Возвращает строку, содержащую шестнадцатичное значение числа INT: s=HEX(65535)
LCase(STR)	Возвращает строку, содержащую исходную строку в нижнем регистре
UCase(STR)	Возвращает строку, содержащую исходную строку в верхнем регистре
Left(STR,INT)	Возвращает часть строки STR, начиная с первого символа до указанного номера INT
Len(STR)	Возвращает целое число - длину строки STR. S="New Year" i=Len(S)
LTrim(STR)	Возвращает строку без лидирующих пробелов
Mid(STR,INT, INT)	Возвращает строку, извлечённую из строки STR, начиная с символа в позиции INT1, и содержащую INT2 символов.
Right(STR,INT)	Возвращает строку из INT символов из строки STR, начиная справа. s = "Hello World" MyStr = Right(s, 6); REM Возвратит " World"
RTrim(STR)	Возвращает строку без пробелов справа от строки STR. s=" Hello" s=RTrim(s)
Space(INT)	Возвращает строку, состоящую из INT пробелов. s=Space(10)
Str(VAR)	Возвращает строку, представляющую число VAR. s = Str(459.001)
String(INT,INT)	Возвращает строку, состоящую из INT1 числа символов INT2. s=String(10,65)

	<code>s= String(10,"A")</code>
Trim(STR)	Возвращает строку без лидирующих и замыкающих пробелов в строке STR. Т.е. одновременно выполняет 2 ф-ции: RTrim и LTrim. <code>s=Trim(s)</code>
Val(STR)	Превращает строку в число. (Возвращает тип Double).

## Нестандартные функции

Имя	Описание функции и пример использования
Opros(INT, INT,INT, INT,INT)	Выполнение СОС опроса. Возвращает переменную типа INTEGER <code>Opros(luch,adr,cmd,data,pogog)</code> Результат INT: INT<0 - не удалось выполнить опрос или нет ответа -1 - нет потока связанного с лучем -2 - нет доступа к потоку опроса -3 - не удастся выбрать луч... -4 - нет ответа по СОС INT>=0 – успешный ответ получен INT&255 – Байт1 ответа (INT/256)&255 – Байт2 ответа Пример: <code>res=Opros(1,61,255,0,-100)</code> <code>if res&gt;=0 Print "Ответ1=",res&amp;255, "Ответ2=",((res/256)&amp;255)</code>

## Функции работы с цветом

Имя	Описание функции и пример использования
RGB(INT, INT,INT)	Получение цвета по трем составляющим. Возвращает переменную типа INTEGER <code>RGB(red,green,blue)</code> , где red=0...255 green=0...255 blue=0...255 Пример: <code>redcolor=RGB(255,0,0)</code> <code>greencolor=RGB(0,255,0)</code> <code>bluecolor= RGB(0, 0,255)</code>

## Функции работы со звуком

Имя	Описание функции и пример использования
PlaySound(STR)	Проигрывание звукового файла STR. Если файл существует, то возвращается 1 (INTEGER) и он проигрывается. Если файл не существует, то возвращается ноль. Данная функция добавляет имя файла в очередь воспроизведения звуков и возвращает управление еще до завершения воспроизведения указанного звукового файла. Поддерживаются только WAV файлы. Пример: <code>i=PlaySound("wave\alarm.wav")</code>

INT – тип INTEGER  
DBL – тип DOUBLE  
STR – тип STRING  
VAR – любой тип

## Допустимые арифметические операции

В выражениях допускаются следующие арифметические операции (в порядке возрастания приоритета):

Символ	Операция
<b>Низкоприоритетные операции</b>	
<	Меньше (результат Да-1, Нет -0)
>	Больше (результат Да-1, Нет -0)
=	Равно (результат Да-1, Нет -0)
<>	Не равно (результат Да-1, Нет -0)
>=	Больше или равно (результат Да-1, Нет -0)
<=	Меньше или равно (результат Да-1, Нет -0)
AND	Логическое AND (результат Да-1, Нет -0)
OR	Логическое OR (результат Да-1, Нет -0)
XOR	Логическое XOR (результат Да-1, Нет -0)
<b>Среднеприоритетные операции</b>	
-	Вычитание
+	Сложение
&	Поразрядное AND (целочисленная операция)
	Поразрядное OR (целочисленная операция)
<b>Высокоприоритетные операции</b>	
*	Умножение
/	Деление
%	Остаток от деления
<b>Сверхвысокоприоритетные операции</b>	
^	Возведение в степень
<b>Унарные операции (наиболее приоритетные)</b>	
+	Положительное число (можно не указывать)
-	Отрицательное число
NOT	Логическое НЕ (результат Да-1, Нет -0)

ВНИМАНИЕ: Для изменения приоритета операций используйте круглые скобки.

## Выражения

В программе можно использовать сложные выражения, состоящие из символов операций, чисел, любых переменных и круглых скобок. Порядок выполнения расчета может быть изменен круглыми скобками.

Пример:

```
var1= var2+var3 <> var4+var5^2
```

## Оператор IF THEN ELSE ENDIF

Общий вид оператора:

```
IF выражение THEN операторы  
ELSE операторы  
ENDIF
```

**ВНИМАНИЕ:** Оператор ELSE в операторе IF допускается не использовать. Но каждому оператору IF в программе должен соответствовать оператор окончания – ENDIF. Если ELSE используется, то он относится к ближайшему оператору IF.

Уровень вложенности операторов IF – до 32.

Пример:

```
IF (a+b)>3 THEN b=0; a=0;  
ELSE a=1; b=2;  
ENDIF
```

```
IF Biu1.F1 THEN a=1;
  IF n+Dir<>16 THEN a=0
ENDIF
ELSE
a=-1
ENDIF
```

## Оператор присвоения значения переменной

Общий вид оператора:

Переменная = выражение

Оператор выполняет присвоение переменной значение выражения. В состав выражения могут входить любые переменные программы.

Пример:

```
A=@TickCount+Z^3-56*Test
```

## Оператор GOTO метка

Переход на метку.

Пример:

```
IF v&4 GOTO MyLabel
ENDIF
END
MyLabel;;
  Print "v=",v
END
```

## Оператор GOSUB метка

Вызов подпрограммы расположенной после метки. В конце подпрограммы располагается оператор возврата RETURN. Количество операторов RETURN в подпрограмме неограниченно. Уровень вложенности GOSUB не более 16.

Пример:

```
GOSUB MySub
END
MySub;;
  Print "v=",v
RETURN
```

## Оператор RETURN

RETURN завершает подпрограмму.

## Оператор FOR,TO, STEP

Общий вид оператора:

Оператор FOR переменная=выражение TO выражение STEP выражение  
 Оператор FOR позволяет организовать циклическое выполнение кода программы. Завершается оператором NEXT. Уровень вложенности циклов не более 16.  
 Шаг STEP допускается не указывать – в этом случае шаг равен единице.  
 Оператор FOR в качестве индекса использует целочисленную переменную.

Пример:

```
FOR i=0 TO 10
PRINT "i=",i
NEXT
```

## Оператор NEXT

Завершает ближайший цикл FOR.

## Оператор WHILE WEND

Общий вид оператора:

```
WHILE выражение
операторы
WEND
```

Операторы выполняются пока выражение истинно.  
 Уровень вложенности операторов WHILE – до 16.

Пример:

```
i=0
WHILE i<10
PRINT "i=",i
i=i+1
WEND
```

## Оператор PRINT

Оператор используется для отладочной печати и для вывода в predetermined окна программы (потоки).

Общий вид оператора:

```
PRINT #N,параметр 1, параметр 2, ...
```

Здесь

#N – номер потока

Параметр – Строка в двойных кавычках или выражение

В следующей таблице приведены номера потоков:

Номер потока	Пояснение
#0	Вывод в отладочное окно программы. Допускается #0 не указывать.
#1	Вывод в специальное окно сообщения, открываемое при печати в этот поток. Для закрытия окна сообщения необходимо напечатать пустую строку.
#2	Вывод в окно ввода целых чисел.
#3	Вывод в статусную строку программы.
#4	Печать в файл (дописывание в файл)
#5	Печать в файл (с удалением в момент создания)
#6	Вывод текстов на кнопках в окне сообщений. Текст состоит из трех записей, заканчивающихся запятой. Print #6,"Кнопка1, Yes,No,"
#7	Поле имя записи в журнал не более 63 символом (усекается если больше)
#8	Поле подимя записи в журнал не более 47 символов

#9	Поле текст записи в журнал не более 31 символа
#10 и далее	Не используются с оператором PRINT

Пример:

```
PRINT #0,"Переменная A=",a," куб=",a^3
rem *****далее то же самое*****
PRINT "Переменная A=",a," куб=",a^3
rem *****Открыть окно сообщения*****
PRINT #1,"Дежурному электрику срочно"
PRINT #1,"включить автомат номер ",A
rem *****Заккрыть окно сообщения*****
PRINT #1,""
```

Внешний вид окна сообщений:

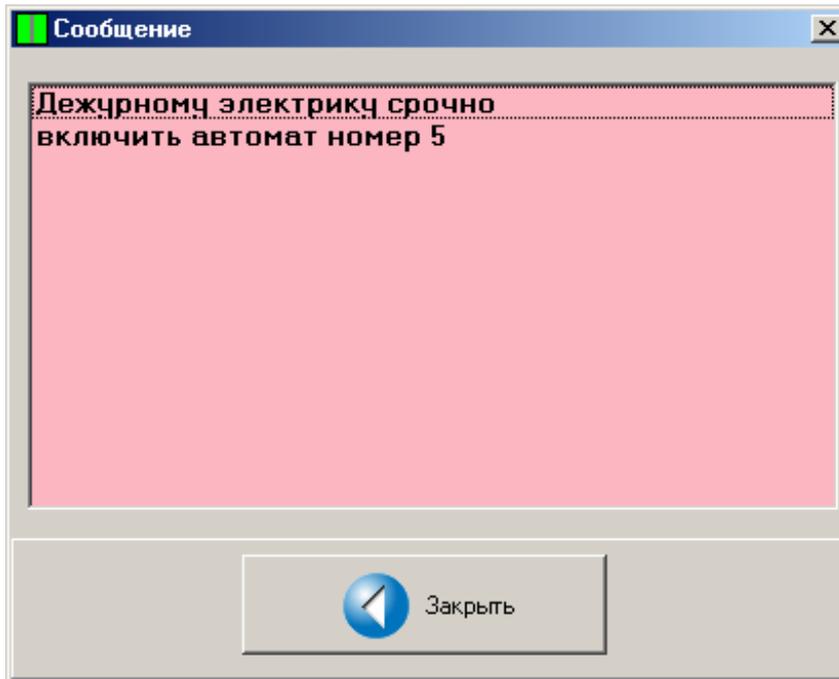


Рисунок – Пример окна сообщений

Пример:

```
A=2001
PRINT #3,"Сообщение",A
```

## Оператор END

Завершает выполнение программы. Может встречаться неоднократно в тексте программы.

Пример:

```
IF Upit>280 then END
...
END
```

## Оператор REM

Вставление комментария в программу. Любые символы до конца строки после REM будут пропущены.

Пример:

```
REM ***** Программа управления *****
REM ***** 22.07.2001 *****
```

## Оператор OPEN

Общий вид оператора:

```
OPEN #N AS FILE "filename"
```

Здесь

**#N** – номер потока

**filename** – Строка (имя файла) в двойных кавычках

Оператор OPEN открывает файл **filename** для записи или чтения.

Для открытия файла на запись используется потоки #4 и #5.

Запись в файл выполняется при помощи печати в 4-й поток (PRINT #4,...). После каждого оператора PRINT в файл записывается перевод строки и возврат каретки (каждый PRINT-отдельная строка). Вывод в файл происходит в текстовом режиме. Для вывода в файл в двоичном режиме следует использовать оператор OUTPUT.

Оператор OPEN может открывать файл для записи на 4-м или 5-и потоке.

В файл открытый на 4-м потоке новые строки всегда дописывается в конец.

В файл открытый на 5-м потоке каждое открытие удаляет все предыдущее содержимое файла.

**ВНИМАНИЕ!** Одновременно можно открыть два файла на запись.

Для закрытия файла используется оператор CLOSE.

Для чтения из файла используются потоки #6...#9. На чтение может быть открыто одновременно до 4-х файлов. Чтение из файлов возможно только в двоичном режиме. Для ввода из файлов используется оператор INPUT.

Пример:

```
OPEN #5 AS FILE "123.txt"  
A=2006  
PRINT #5,"Сообщение ",A  
CLOSE #5
```

## Оператор CLOSE

Общий вид оператора:

```
CLOSE #N
```

Здесь

**#N** – номер потока

Пример:

```
CLOSE #5
```

Закрывает открытый ранее файл.

## Оператор INPUT

Ввод данных из файла.

Общий вид оператора:

```
INPUT #N,переменная
```

Здесь

**#N** – номер потока (#6,#7,#8 или #9)

**переменная** – имя переменной, куда будет прочитан один байт из файла

Ввод данных выполняется побайтно, где прочитанное число является положительным и принимает значения в диапазоне 0...255.

Пример:

```
OPEN #8 AS FILE "MY.DAT"  
FOR I=1 TO 10  
INPUT #8,A  
PRINT A  
NEXT  
CLOSE #8
```

Чтение 10 байт из файла «MY.DAT».

ПРИМЕЧАНИЕ:

Для проверки выполнения реального чтения используется системная переменная

@Error, принимающая следующие значения :

0-Успешно

1-Файл не открыт

2-Ошибка записи в файл

3-Ошибка чтения из файла

Значение переменной можно контролировать после выполнения любой операции с файлом.

Пример:

```
OPEN #8 AS FILE "MY.DAT"  
IF @Error>0 THEN Print "Файл не открыт..."  
ENDIF  
INPUT #8,A  
IF @Error>0 THEN Print "Не удалось прочитать..."  
ENDIF
```

## Оператор OUTPUT

Ввод данных из файла.

Общий вид оператора:

```
OUTPUT #N,выражение
```

Здесь

#N – номер потока (#6,#7,#8 или #9)

**выражение** – любое допустимое выражение.

Вывод данных выполняется побайтно – в файл будет записан один байт результата выражения. Если значение результата выражения превышает число 255, то в файл будет записан ТОЛЬКО младший байт (положительное число в диапазоне 0...255).

Пример:

```
OPEN #5 AS FILE "MY.DAT"  
IF @Error>0 THEN PRINT "File is not open";END  
ENDIF  
FOR i=0 to 255  
OUTPUT #5,255-i  
NEXT  
CLOSE #5  
END
```

Вывод 256 байт в файл «MY.DAT».

ПРИМЕЧАНИЕ:

Для проверки выполнения реальной записи используется системная переменная

@Error, принимающая следующие значения :

0-Успешно

1-Файл не открыт

2-Ошибка записи в файл

3-Ошибка чтения из файла

Значение переменной можно контролировать после выполнения любой операции с файлом.

## Оператор RUN

Запуск внешнего приложения или функции из DLL библиотеки

Общий вид оператора:

```
RUN "filename"  
RUN "funcname" FROM "dllfilename"
```

здесь «filename» - имя исполняемого файла

«dllfilename» - имя файла динамической библиотеки

«funcname» - имя функции в динамической библиотеке

Пример:

```

REM Запуск приложения
RUN "C:\Program Files\WinAmp\WinAmp.exe"
REM Вызов функции из библиотеки
RUN "@ShowResultSq" FROM "Bpdd.dll"

```

## Переменные объектов карты

Каждый объект на карте может иметь собственное имя для программирования. Имя задается в окне свойств объекта следующим образом:

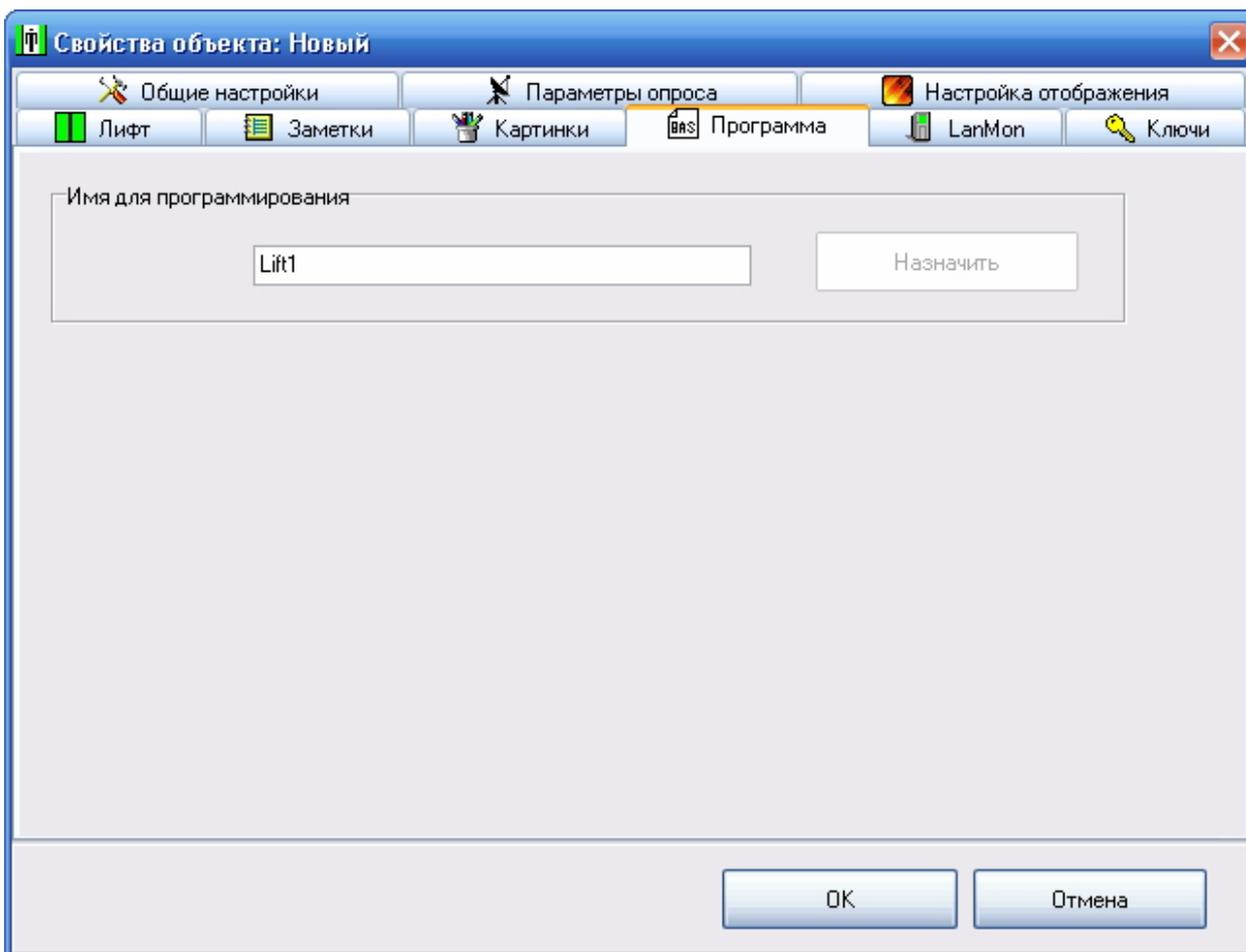


Рисунок – Назначение имени объекта для программирования

Имя для программирования должно быть уникальным текстовым идентификатором, начинающимся с латинской буквы. Имя объекта нечувствительно к регистру.

После имени объекта должен стоять символ точка и далее имя поля объекта. Имя поля также нечувствительно к регистру.

Для генерации уникального имени рекомендуется воспользоваться кнопкой «Назначить».

Ниже приведена таблица полей, которые есть у любого объекта на карте:

Поле переменной	Тип	Пояснение
<b>Luch</b>	INTEGER	Номер СОС луча
<b>Adr</b>	INTEGER	СОС адрес
<b>PulseWidth</b>	INTEGER	Ширина импульса
<b>Porog</b>	INTEGER	Порог приема 0...255 (128 - нулевой порог)
<b>Attem</b>	INTEGER	Количество попыток обмена
<b>Type</b>	INTEGER	Тип объекта: 0 //лифт (БЗЛ)

		1 //датчик температуры 2 //охранный датчик 3 //параметр лифта 4 //Блок Информационно Управляющий лифтовой 5 //Блок Голосовой Связи БГС-П 6 //Блок аналоговых датчиков 7 //Блок Информационно Управляющий 8 //Блок ККД 9 //Индикатор 10 //Аналоговый параметр 11 //Переключатель 12 //Блок передачи данных дуплексный БПДД 13 //Измеритель уровня ИУ-1 14 //Указатель извещатель ручной речевой цифровой
<b>Temperature</b>	INTEGER	Температура объекта
<b>Ohrana</b>	INTEGER	Состояние охраны объекта
<b>Quality</b>	INTEGER	Качество обмена 0-100 % (только для чтения)
<b>UsePIN</b>	INTEGER	Подключенные сигналы БДКЛ бит 0 - состояние РТО бит 1 - состояние РОД бит 2 - состояние РКД бит 3 - использовать сигнал скорости для движения лифта
<b>Sound</b>	INTEGER	Звук пункта (1-разрешен или 0-запрещен)
<b>Disabled</b>	INTEGER	Объект выключен (1 – выключен 0 –включен) Состояние хранится в Attrib
<b>Status</b>	INTEGER	Состояние объекта. Отдельные биты определяют состояние объекта: бит 0 - Call - Вызов из лифта или подвала, ЭЩ бит 1 - CallMO - Вызов из МО бит 2 - Ohrana - состояние охраны 1 или 2-5 бит 3 – Restart - перезапуск бит 4 – Faza - состояние сигнала Faza бит 5 – Sve - состояние сигнала Svet бит 6 – StopPress- нажата кнопка Стоп бит 7 – LiftMove – лифт двигается бит 8 – LiftOpen - Дверь кабины лифта открыта бит 9 – LiftBreak-Обнаружены неисправности лифта бит 10 – Bzl - Срабатывание защитного устройства БЗЛ бит 11 – BzlErr-Ошибки обмена с БЗЛ бит 12 – Temp-произошло превышение порога температуры бит 13 – Gradient-произошло превыш. градиента температуры бит 14 – NoAnswer - нет ответа от блока (контр.связи) бит 15 – BzlReset - произошел сброс лифтового контроллера бит 16 – NoMove - долго нет движения лифта бит 17 - пассажир в кабине лифта
<b>Attrib</b>	INTEGER	Атрибуты объекта. Отдельные биты определяют настройки и состояние объекта: бит 0 - NeedFull - нужен полный опрос бит 1- ABit1- не используется бит 2- ABit2- не используется бит 3- UseBzl - использовать БЗЛ(для лифта) бит 4- InvOhr - инвертировать датчик охраны бит 5- HaveOhr - датчик охраны подключен бит 6- HaveTemp - датчик темп. подключен бит 7- Disabled - объект выключен бит 8- Test - выполнять Тест-контроль (есть ПУ лифта) бит 9- TestMO - выполнять Тест-контроль МП (есть ПУ МП) бит 10 – Passenger - использовать датчик пассажира бит 11 – FourTemp - четыре датчика температуры бит 12 – Auto - автофункция бит 13 – Rotate - повернуть датчик бит 14 – NameUnVisible - Не отображать название объекта

		бит 15 – InvROD - инвертировать сигнал РОД бит 16 – CRCSOS - протокол обмена CRC бит 17 - записать протокол срабатывания УБ бит 18 - записать протокол срабатывания УКСЛ бит 19 - записывать сигналы лифта бит 20 - электронный лифт КМЗ бит b21+ b22 -разговорный канал лифта: 0 - CHAN_LIFT    1 - CHAN_MP 2 - CHAN_PODVAL 3 - CHAN_E бит 23 - наступил перегрев эд. лифта - нужно остановить бит 24 - лифт остановлен-перегрев бит 25 - БЗЛ старого типа бит 26 - инвертировать сигнал РТО бит 27 - электронный лифт ПКЛ-32 бит 28 - у блока цифровой звук
<b>PictureIndex</b>	INTEGER	Индекс отображаемой картинки
<b>Name</b>	STRING	Имя блока
<b>Hint</b>	STRING	Подсказка блока
<b>SubName</b>	STRING	Подимья блока
<b>Template</b>	STRING	Формат отображения
<b>ImageListFile</b>	STRING	Имя файла набора картинок
<b>VoiceFile</b>	STRING	Звуковой файл объекта
<b>Caption</b>	STRING	Подпись объекта
<b>DisplacementX</b>	INTEGER	Смещение подписи вправо или влево
<b>DisplacementY</b>	INTEGER	Смещение подписи вверх или вниз
<b>MicVolume</b>	INTEGER	Уровень воспроизведения с микрофона
<b>AuxVolume</b>	INTEGER	Уровень аудиовхода
<b>MicVolumeMP</b>	INTEGER	Уровень микрофона MP
<b>AuxVolumeMP</b>	INTEGER	Уровень аудиовхода MP
<b>LogUB</b>	INTEGER	Сохранять в журнале данные УБ
<b>LogUKSL</b>	INTEGER	Сохранять в журнале данные УКСЛ
<b>ProgName</b>	STRING	Имя для программирования (до 31 символа)
<b>TextNorma</b>	STRING	Текст нормального состояния (до 31 символа)
<b>TextAlarm</b>	STRING	Текст аварийного состояния (до 31 символа)
<b>Top</b>	INTEGER	координата на карте сверху
<b>Left</b>	INTEGER	координата на карте слева
<b>Map</b>	INTEGER	номер карты - отображать на этой карте
<b>A3</b>	INTEGER	адрес A3 переменной в LanMon
<b>A4</b>	INTEGER	адрес A4 переменной в LanMon
<b>Bytes0 – Bytes51</b>	INTEGER	51 байт специального назначения. Значение каждого байта зависит от типа объекта.

ВНИМАНИЕ: Не все поля объекта являются значимыми. Например, поле адреса «Adt» в программируемом переключателе не используется.

### Дополнительные поля БИУ (СОС-95 БИУ-Ф)

Поле переменной	Тип	Пояснение
<b>F1</b>	INTEGER	Фаза 1
<b>F2</b>	INTEGER	Фаза 2
<b>F3</b>	INTEGER	Фаза 3
<b>F4</b>	INTEGER	Фаза 4
<b>F5</b>	INTEGER	Фаза 5
<b>F6</b>	INTEGER	Фаза 6
<b>UPR1</b>	INTEGER	Управление каналом 1
<b>UPR2</b>	INTEGER	Управление каналом 2
<b>Vp</b>	INTEGER	Наличие питания в схеме управления
<b>A</b>	INTEGER	перекос фаз

C1	INTEGER	текущее состояние управления канала 1 (прочитано)
C2	INTEGER	текущее состояние управления канала 2 (прочитано)
T	INTEGER	наличие температуры
Temp	INTEGER	температура непосредственно с датчика
Cnt1	INTEGER	счетчик фазы 1
Cnt2	INTEGER	счетчик фазы 2
Cnt3	INTEGER	счетчик фазы 3
Cnt4	INTEGER	счетчик фазы 4
Cnt5	INTEGER	счетчик фазы 5
Cnt6	INTEGER	счетчик фазы 6

### Дополнительные поля БАД

Поле переменной	Тип	Пояснение
Par1	INTEGER	Параметр 1
Par2	INTEGER	Параметр 2
Par3	INTEGER	Параметр 3
Par4	INTEGER	Параметр 4
Par5	INTEGER	Параметр 5
Par6	INTEGER	Параметр 6
Par7	INTEGER	Параметр 7
Par8	INTEGER	Параметр 8

### Дополнительные поля ККД

Поле переменной	Тип	Пояснение
K1	INTEGER	Шлейф 1
K2	INTEGER	Шлейф 2
K3	INTEGER	Шлейф 3
K4	INTEGER	Шлейф 4
K5	INTEGER	Шлейф 5
K6	INTEGER	Шлейф 6
K7	INTEGER	Шлейф 7
K8	INTEGER	Шлейф 8
Mask1	INTEGER	маска 1
Mask2	INTEGER	маска 2
Mask3	INTEGER	маска 3
Mask4	INTEGER	маска 4
Mask5	INTEGER	маска 5
Mask6	INTEGER	маска 6
Mask7	INTEGER	маска 7
Mask8	INTEGER	маска 8
Temp	INTEGER	температура непосредственно с датчика
Temp1	INTEGER	дополнительная температура 1
Temp2	INTEGER	дополнительная температура 2
Temp3	INTEGER	дополнительная температура 3
Temp4	INTEGER	дополнительная температура 4

### Дополнительные поля Индикатора

Поле переменной	Тип	Пояснение
Value	INTEGER	Состояние индикатора (0 или 1)

## Дополнительные поля Аналогового Параметра

Поле переменной	Тип	Пояснение
Value	INTEGER	Значение в индикаторе (целое число)
ParNum	INTEGER	Номер параметра источника
K	FLOAT	Коэффициент
Zero	INTEGER	Значение нуля датчика (целое число)
FloatValue	FLOAT	Значение в формате float
DataValid	INTEGER	Данные достоверны 1-Да 0-Нет

## Дополнительные поля Управляемого переключателя (Тумблер)

Поле переменной	Тип	Пояснение
State	INTEGER	Состояние переключателя (0 или 1)

ВНИМАНИЕ: Для того чтобы Управляемый переключатель и Индикатора работали только под управлением программы, следует удалить имя в свойствах объекта.

## Дополнительные поля БПДД

Поле переменной	Тип	Пояснение
Reset	INTEGER	Сброс внутренних программных буферов БПДД. Необходимо выполнить один раз перед каждым сеансом обмена. Переменная доступна только для записи. Результат выполнения записи в эту переменную проверяется через системную переменную @Eggor: 0- сброс выполнен 4- нет доступа к БПДД (дальнейший обмен не будет выполнен) (только запись)
TimeOut	INTEGER	Максимальное время в миллисекундах для чтения одного символа из БПДД (по умолчанию=100).
InBuf	INTEGER	Количество символов во внутреннем приемном буфере БПДД. Доступно только для чтения. Результат выполнения чтения этой переменной проверяется через системную переменную @Eggor: 0- полученные данные достоверны 4- нет доступа к БПДД (полученное значение недостоверно) (только чтение)
Byte	INTEGER	Переменная для доступа к внутренним буферам приемника и передатчика БПДД. Переменная доступна для чтения и записи. При записи, записываемый символ выводится в последовательный интерфейс через внутренний буфер БПДД. При чтении происходит извлечение принятого символа из внутреннего буфера БПДД или ожидание символа в течении TimeOut миллисекунд. Результат доступа к переменной проверяется через системную переменную @Eggor: 0- выполнено успешно 4- нет доступа к БПДД 5-не удалось послать символ 6-не удалось принять символ

## Дополнительные поля блока ИУ-1

Поле переменной	Тип	Пояснение
Level1	INTEGER	Уровень 1
Level2	INTEGER	Уровень 2
Level3	INTEGER	Уровень 3
Level4	INTEGER	Уровень 4

Temp	INTEGER	Температура непосредственно с датчика
------	---------	---------------------------------------

### Дополнительные поля блока УИР-РЦ

Поле переменной	Тип	Пояснение
Zona	INTEGER	Номер зоны
ZonaType	INTEGER	Тип зоны
SpeakOproS	INTEGER	Разрешить выполнение опроса во время переговоров

### Дополнительные поля блока БИУ-Л

Поле переменной	Тип	Пояснение
MP	INTEGER	Состояние управления 1- МП включен 0-выключен
Faza	INTEGER	Состояние выхода МП 1- есть фаза 0-нет фазы (только чтение)
CheckFaza	INTEGER	Тип БИУ: 1 – БИУ-ЛМ 0 –БИУ-Л
Connected	INTEGER	Подключен ли БИУ-ЛМ: 1-Да, 0-Нет (только чтение)

### Работа с неименованными объектами на карте

Используя специальные системные переменные можно получить доступ к любым объектам на карте. Все объекты представлены в виде единого списка.

Общее количество визуальных объектов на карте доступно для чтения в системной переменной **@MapCnt** . Для записи эта переменная недоступна.

Для получения доступа к объекту на карте необходимо указать индекс объекта в списке. Для этого используется системная переменная **@MapIndex** , которая доступна как для записи, так и для чтения. Следует помнить, что системная переменная **@MapIndex** должна находиться в пределах 0... **@MapCnt-1** .

Для доступа собственно к объекту используется системная переменная **@Map**, которая может иметь любые поля, которые описаны для объектов ранее.

#### ПРИМЕР:

Приведенная ниже программа выводит в отладочное окно список всех объектов карты и делает текстовую расшифровку типа объекта.

```

PRINT "Всего объектов на карте:",@Mapcnt
FOR i=0 TO @Mapcnt-1
  @MapIndex=i
  PRINT "Объект",@MapIndex," Тип=", @Map.Type
  IF @Map.Type=0 THEN PRINT "БДКЛ"; ENDIF
  IF @Map.Type=1 THEN PRINT "Датчик температуры"; ENDIF
  IF @Map.Type=2 THEN PRINT "Охранный датчик"; ENDIF
  IF @Map.Type=3 THEN PRINT "Параметр лифта"; ENDIF
  IF @Map.Type=4 THEN PRINT "Блок Инф. Управляющий лифтовой"; ENDIF
  IF @Map.Type=5 THEN PRINT "Блок Голосовой Связи БГС-П"; ENDIF
  IF @Map.Type=6 THEN PRINT "Блок аналоговых датчиков"; ENDIF
  IF @Map.Type=7 THEN PRINT "Блок Информационно Управ."; ENDIF
  IF @Map.Type=8 THEN PRINT "Блок ККД"; ENDIF
  IF @Map.Type=9 THEN PRINT "Индикатор"; ENDIF
  IF @Map.Type=10 THEN PRINT "Аналоговый параметр"; ENDIF
  IF @Map.Type=11 THEN PRINT "Переключатель"; ENDIF
NEXT
END

```

Так как в случае использования системной переменной **@Map** известен только индекс объекта, то в операторе PRINT допускается использовать дополнительные поля:

**@Map.Name** – название объекта на карте

**@Map.SubName** – подимя объекта на карте. (Например, расположение охранного шлейфа).

#### ПРИМЕР:

Ниже приведена программа, которая измеряет напряжение +110 вольт для всех включенных блоков БДКЛ

```
REM Эта программа выводит напряжение +110 вольт для всех
REM включенных блоков БДКЛ
PRINT "Всего объектов на карте:",@Mapcnt
cnt=0
FOR i=0 TO @Mapcnt-1
@MapIndex=i
IF @Map.Type=0 THEN GOSUB LiftFound
ENDIF
NEXT
PRINT "Всего БДКЛ: ",cnt
END
```

```
LiftFound:;
REM БДКЛ может быть выключен
IF @Map.Disabled THEN RETURN
ENDIF
rem убедимся что это БДКЛ а не БДК
if @Map.Attrib&8 = 0 THEN RETURN
REM рассчитаем напряжение +110 В
u=@Map.Bytes3*@Map.Bytes20/256
cnt=cnt+1
PRINT cnt,". U=",u," ",@Map.Name
REM опять измерить напряжение
@Map.Attrib=@Map.Attrib|1
RETURN
```

## Ввод чисел в программу

Для ввода числа в программе используется следующий код:

**переменная=@InputInteger**

Эту функцию нужно использовать очень осторожно, т.к. выполнение опроса приостанавливается на время ввода числа пользователем.

Дополнительно нужно задать текст сообщения в открывающемся окне, используя печать в поток номер #2. Пример:

```
PRINT #2,"Введите параметр (0-1000)"
var=@InputInteger
```

Вид окна ввода числа:

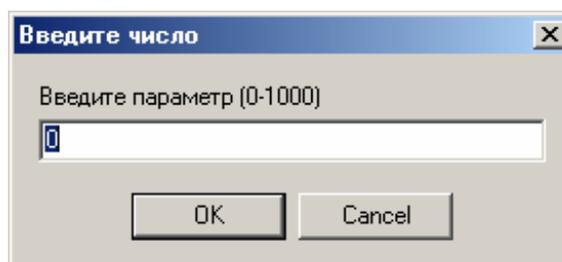


Рисунок – Окно ввода числа в программу

## Выполнение записи в общий журнал программы

В некоторых случаях требуется выполнить занесение в общий журнал программы дополнительной информации. Это можно выполнить следующим образом:

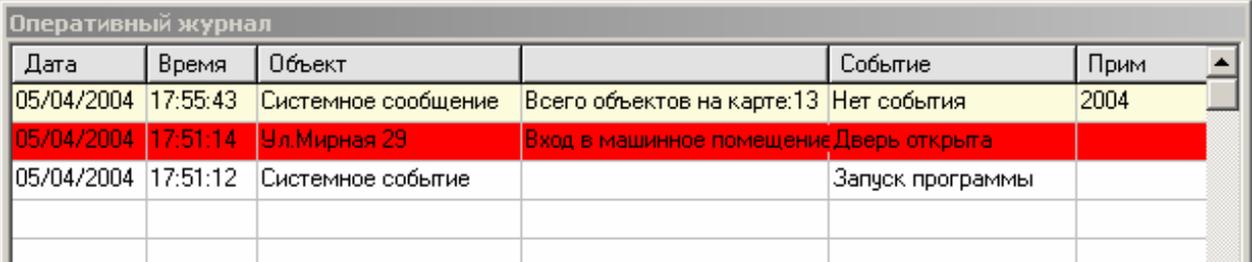
1. Записать в поток #7 оператором PRINT имя объекта (не более 63 символов)
2. Записать в поток #8 оператором PRINT подвйма объекта (не более 47 символов)
3. Записать в поток #9 оператором PRINT текстовую информацию (не более 31 символов)
4. Собственно выполнить запись в журнал записью в системную переменную @Log  
Дата и время записываются автоматически.

### ПРИМЕР:

Ниже приведена программа, которая выполняет запись в общий журнал программы:

```
PRINT #7,"Системное сообщение"  
PRINT #8,"Всего объектов на карте:",@Marcnt  
PRINT #9,"Нет события"  
@Log=2004  
END
```

На рисунке показан результат выполнения однократного выполнения программы – появление записи в оперативном журнале.



Дата	Время	Объект	Событие	Прим
05/04/2004	17:55:43	Системное сообщение	Всего объектов на карте:13 Нет события	2004
05/04/2004	17:51:14	Ул.Мирная 29	Вход в машинное помещение Дверь открыта	
05/04/2004	17:51:12	Системное событие	Запуск программы	

Рисунок – Оперативный журнал

Следует обратить внимание, что в поле «Прим» отображается целое число, записанное в переменную @Log.

## Вызов окна программы

Для вызова окна программы нажмите Ctrl+P (Program). Общий вид окна программы:

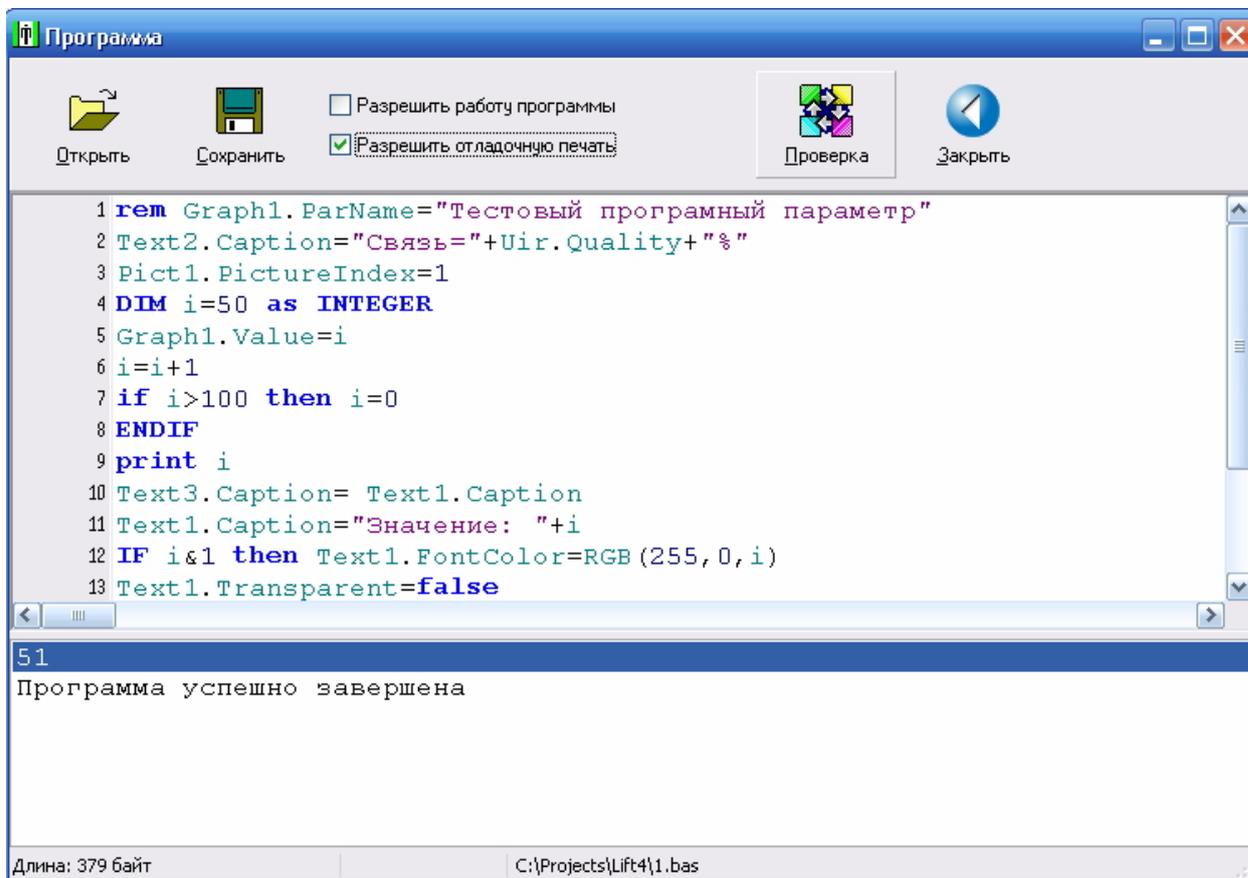


Рисунок – Основное окно программы

Верхняя часть окна предназначена для редактирования текста программы.

Нижняя часть окна называется отладочным окном и предназначено для вывода отладочной информации.

В окне редактирования программа отображается с синтаксической раскраской текста.

## Редактирование программы

Перед редактированием программы сбросьте «переключатель» «Разрешить работу программы» и установите переключатель «Разрешить отладочную печать».

## Вставка объектовых переменных в программу

Для вставки переменной объекта используйте комбинацию клавиш Ctrl+ПРОБЕЛ. После этого открывается окно, в котором можно выбрать объект и поле:

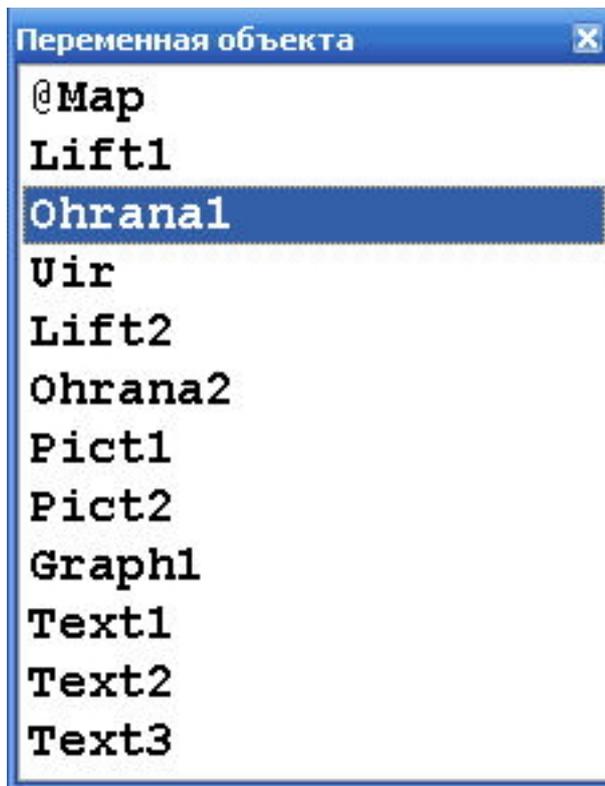


Рисунок – Вставка имени переменной объекта

Для выбора нажмите "ENTER" или используйте двойной клик мышкой. Для отмены нажмите кнопку "ESC". После выбора объекта появляется окно свойств:

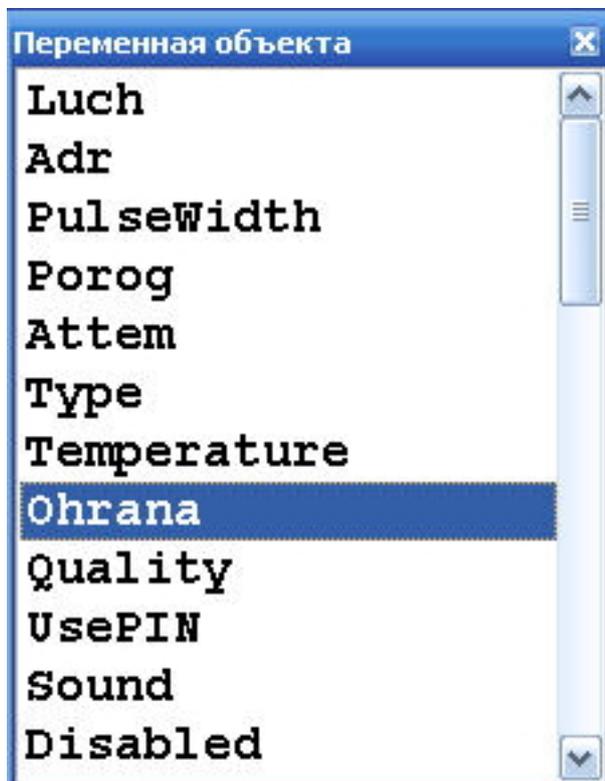


Рисунок – Вставка поля переменной объекта

Выберите нужное свойство объекта и нажмите кнопку «ENTER»

## Дополнительные возможности редактора программы

Вызвав контекстное меню редактора программы

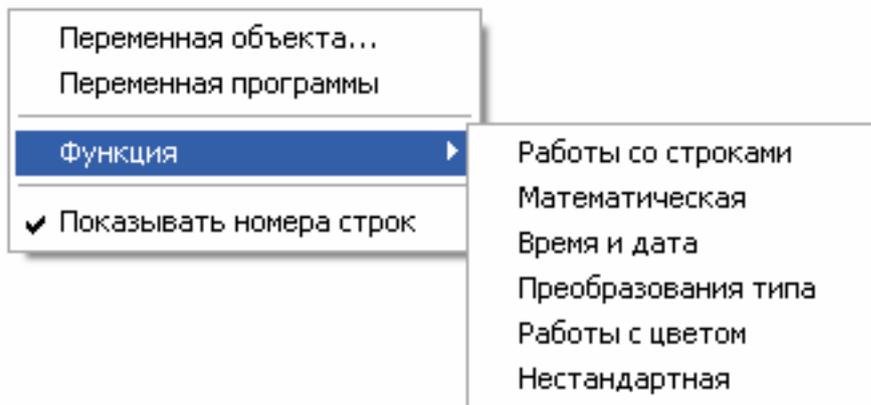


Рисунок – Контекстное меню редактора программы

можно выполнить следующие дополнительные действия:

1. Переменная объекта – вставить переменную объекта (описано выше)
2. Переменная программы - вставить уже известную переменную программы
3. Вставить поддерживаемую функцию (описание функции не вставляется в окно редактирования)

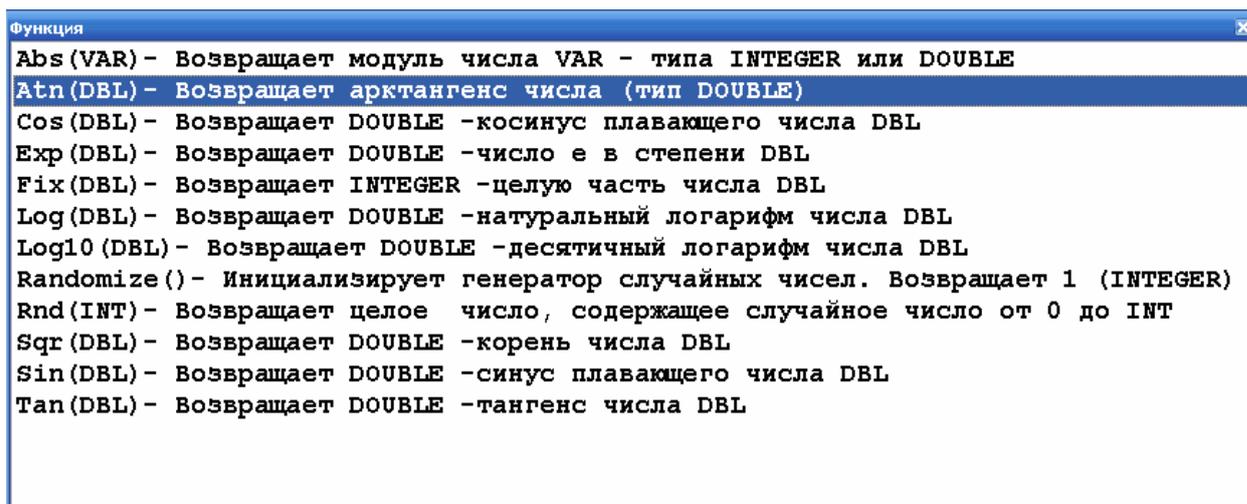


Рисунок – Вставка функции из списка

Для вставки функции выберите нужную функцию и нажмите кнопку «ENTER» или выполните двойной клик мышкой по нужной строке – в окне редактирования будет вставлено имя функции.

## Проверка работы программы

Для проверки программы нажмите кнопку «Проверка». Программа будет выполнена однократно и в нижнем отладочном окне появится сообщение о завершении работы программы.

**ВНИМАНИЕ.** Перепись программы из окна редактирования в исполняемую область происходит в момент нажатия кнопки «Проверка». Если вы изменяете программу в момент ее работы, то для внесения изменений следует выполнить проверку вновь отредактированной программы.

## Выполнение программы

Программа выполняется после полного цикла опроса всех объектов, если установлен переключатель «Разрешить работу программы».

## Сохранение и загрузка программы

Имя программы и признак работы программы сохраняются в файле карты. Сама программа хранится в отдельном файле с расширением BAS. При нажатии на кнопку «Проверка» программа сохраняется в файле с текущим именем. Для смены имени сохраните программу под другим именем, нажав на кнопку «Сохранить». Для загрузки программы из другого файла нажмите «Открыть» и укажите новую программу.

## Постоянная работа программы

Для постоянной работы программы установите переключатель «Разрешить работу программы» и сбросьте переключатель «Разрешить отладочной печати».

Программа выполняется каждый раз после полного цикла опроса всех объектов (с периодом около 1 секунды). Для того чтобы программа загружалась и начинала выполняться автоматически, необходимо сохранить текущую карту в момент постоянной работы программы.

## Дополнительные возможности отладочного окна

В отладочном окне доступно контекстное меню, которое позволяет выполнить следующие дополнительные действия:

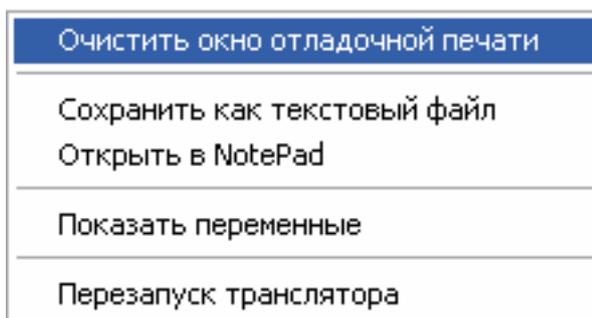


Рисунок – Контекстное меню окна отладки

1. Очистить окно отладочной печати. Выполняется очистка текста в окне отладочной печати.
2. Сохранить как текстовый файл. Весь текст будет сохранен в виде текстового файла.
3. Открыть в Notepad. Текст из отладочного окна будет перенесен во встроенный редактор Windows.
4. Показать переменные. В отладочном окне будет выведен список всех переменных с указанием имени, типа и значения каждой переменной.
5. Перезапуск транслятора. Выполняется удаление всех переменных транслятора. Требуется выполнять перезапуск транслятора если вы хотите чтобы выполнились операторы DIM программы.

## Дополнительные переменные области Bytes

Каждый объект на карте имеет 36 дополнительных переменных, к которым имеется программный доступ. Доступ к байтам выполняется с использованием следующих переменных – от **Bytes0** до **Bytes35**. Ниже приведены таблицы значений области Bytes в зависимости от типа объекта.

### Блок диспетчерского контроля (Блок диспетчерского контроля лифтовой)

Переменная	Имя	Значение
------------	-----	----------

Bytes0	Type	тип лифта (хранится на диске): 0 - Тип 0.65 и 0.71 м/с 1 - Тип 1.0 м/с 143,144 2 - Тип 1.0 м/с 366 3 - Тип УЛЖ 10,17 4 - Тип ШУЛК 5 - Тип УЛ 6 и далее – дополнительные типы лифтов
Bytes1	* State	Состояние защитного устройства БДКЛ: 0 - Все в порядке Остальные значения- сработка защитного устройства: 1 - Нет движения на бол. скорости 2 - Нет движения на мал. скорости 3 - Устройство безопасности 4 - Ошибка фаз ABC 5 - Движение без двигателя 6 - Команда из диспетчерской
Bytes2	X	состояние расширенных сигналов (точное название зависит от типа лифта): бит 0 - Вне зоны точной остановки бит 1 - Неисправность привода дверей бит 2 - Авария по сигналу SIG 0 бит 3 - Авария по сигналу SIG 1 бит 4 - Авария по сигналу SIG 2 бит 5 - Авария по сигналу SIG 3 бит 6 - Авария по сигналу SIG 4 бит 7 - Неисправность С3
Bytes3	* Upit	Код напряжение питания (+110) из АЦП Пересчет в напряжение питания выполняется по формуле: $U_v = \text{код} * K_{110} / 256$ K110 доступен в Bytes20 (см. далее)
Bytes4	* Ui	Код напряжение на РКД из АЦП
Bytes5	Temp	Температура, необработанное значение
Bytes6	* Acnt	Счетчик фазы А (число от 0 до 39)
Bytes7	* BCnt	Счетчик фазы В (число от 0 до 39)
Bytes8	* CCnt	Счетчик фазы С (число от 0 до 39)
Bytes9	* Order	Порядок фаз: 2- норма 0 или 1 – ошибка
Bytes10	* F	Частота оптического датчика деленная на два (Гц)
Bytes11	ResetType	тип сброса лифтового контроллера
Bytes12	OhrBtn	состояние охран и кнопок из контроллера связи: бит 0 – BTN2 – вызов из лифта бит 1 – BTN3 – вызов из подвала бит 2 – BTN4 - вызов из электрощитовой бит 3 – OHR1 – охранный шлейф 1 бит 4 – OHR2 – охранный шлейф 2 бит 5 – OHR3 – охранный шлейф 3 бит 6 – OHR4 – охранный шлейф 4 бит 7 – OHR5 – охранный шлейф 5
Bytes13	BgsState	состояние контроллера связи: бит 0 – BTN1 – вызов из машинного помещения бит 1 - Обмен с лифтовым контроллером (1-есть 0-нет) бит 2 – произошел сброс от WDT бит 3 – произошел сброс от Reset бит 4 – не используется бит 5 – Сигнал FAZA бит 6 – Сигнал SVET бит 7 – Сигнал OHR6 (контроль работоспособности шлейфов)

Bytes14	Rele	состояние реле БГС-ККД: бит 0 - Сигнал U_12 бит 1 - Сигнал U_13 бит 2 - Сигнал U_34 бит 3 - Сигнал U_M/S бит 4 - Сигнал U_LOCAL бит 5 - Сигнал U_LIFT
Bytes15	LiftState	состояние сигналов лифта
Bytes16	Uline	Напряжение в линии
Bytes17	* LiftState1	еще состояние сигналов лифта: Бит 0 – OUT Бит 1 – BLOCK Бит 2 – SIG0 Бит 3 – SIG1 Бит 4 – SIG2 Бит 5 – SIG3 Бит 6 – SIG4 Бит 7– FSIG
Bytes18	R	сопротивление цепи безопасности ВНИМАНИЕ – Не используется
Bytes19	UbUksl	байт 2 из команды 10,11
Bytes20	K110	коэффициент измерения U110 (хранится на диске)
Bytes21	CallCnt	счетчик вызова
Bytes22	CallCntMO	счетчик вызова машинного отделения
Bytes23	Temp1	дополнительная температура 1
Bytes24	Temp2	дополнительная температура 2
Bytes25	Temp3	дополнительная температура 3
Bytes26	Temp4	дополнительная температура 4
Bytes27	TempCnt	температурный счетчик
Bytes28	CntMO	счетчик сколько раз подряд идут вызовы из МО
Bytes29	LastOhrBtn	предыдущее состояние охр
Bytes30	LastX	предыдущее состояние аварий лифта
Bytes31		не используется
Bytes32- Bytes35	LogParam (4 байта)	последняя запись об УБ(УКСЛ) в журнале

\* - эти поля не обновляются автоматически. Для обновления необходимо каждый раз устанавливать специальный бит NeedFull (бит 0) в атрибутах объекта следующим образом:

```
temp= Lift1.Attrib&1
IF temp=0 THEN Lift1.Attrib= Lift1.Attrib+1
ENDIF
```

## Параметр лифта

Переменная	Имя	Значение
Bytes0	Value	значение параметра
Bytes1		не используется
Bytes2, Bytes3	IntValue (2 байта)	двухбайтовый параметр
Bytes4	BytesCnt	количество байтов: 0-бит 1-байт 2-слово
Bytes5	BitNum	номер бита (0-7)
Bytes6	Adr	номер стандартного параметра
Bytes7	NeedRead	нужно читать

## БИУ-Л

Переменная	Имя	Значение
Bytes0	MP	состояние управления 1- МП включен 0-выключен
Bytes1	Faza	состояние выхода МП 1- есть фаза 0-нет фазы

## БГС-П

Переменная	Имя	Значение
Bytes0	Podval	Расположение: 0 – электрощитовая, не 0- подвал
Bytes1	CallCnt	Счетчик вызова (для имитации длинного гудка)

## Охранный шлейф

Переменная	Имя	Значение
Bytes0	OhranaNum	Номер охранного шлейфа

## Датчик температуры

Переменная	Имя	Значение
Bytes0	TempNum	Номер температурного датчика
Bytes1	HighTempCnt	счетчик времени высокой температуры (для выключения при перегреве двигателя)
Bytes2		не используется
Bytes3		не используется
Bytes4-Bytes7	LastTemp	предыдущая температура
Bytes8-Bytes11	LogTemperature	температура объекта записанная в журнал

## Блок аналоговых датчиков

Переменная	Имя	Значение
Bytes0	Par0	Значение параметра 0: Беззнаковое целое – два байта –младший байт
Bytes1	(два байта)	по младшему адресу
Bytes2	Par1	Значение параметра 0: Беззнаковое целое – два байта –младший байт
Bytes3	(два байта)	по младшему адресу
Bytes4	Par2	Значение параметра 0: Беззнаковое целое – два байта –младший байт
Bytes5	(два байта)	по младшему адресу
Bytes6	Par3	Значение параметра 0: Беззнаковое целое – два байта –младший байт
Bytes7	(два байта)	по младшему адресу
Bytes8	Par4	Значение параметра 0: Беззнаковое целое – два байта –младший байт
Bytes9	(два байта)	по младшему адресу
Bytes10	Par5	Значение параметра 0: Беззнаковое целое – два байта –младший байт
Bytes11	(два байта)	по младшему адресу
Bytes12	Par6	Значение параметра 0: Беззнаковое целое – два байта –младший байт
Bytes13	(два байта)	по младшему адресу
Bytes14	Par7	Значение параметра 0: Беззнаковое целое – два байта –младший байт
Bytes15	(два байта)	по младшему адресу

## БИУ СОС-95

Переменная	Имя	Значение
Bytes0	UPR1	управление каналом 1
Bytes1	UPR2	управление каналом 2
Bytes2	F0	Фаза 0
Bytes3	F1	Фаза 1
Bytes4	F2	Фаза 2
Bytes5	F3	Фаза 3
Bytes6	F4	Фаза 4
Bytes7	F5	Фаза 5
Bytes8	Vp	Наличие питания в схеме управления
Bytes9	A	перекос фаз
Bytes10	C1	текущее состояние управления канала 1 (прочитано)

Bytes11	C2	текущее состояние управления канала 2 (прочитано)
Bytes12	T	наличие температуры
Bytes13	Temp	температура непосредственно с датчика
Bytes14	Cnt0	счетчик фазы 0
Bytes15	Cnt1	счетчик фазы 1
Bytes16	Cnt2	счетчик фазы 2
Bytes17	Cnt3	счетчик фазы 3
Bytes18	Cnt4	счетчик фазы 4
Bytes19	Cnt5	счетчик фазы 5

## ККД СОС-95

Переменная	Имя	Значение
Bytes0-Bytes7		не используются
Bytes 8- Bytes 15	K[8]	состояние охранных контуров 0 - 7
Bytes16	Temp	температура непосредственно с датчика
Bytes17	Temp1	дополнительная температура 1
Bytes18	Temp2	дополнительная температура 2
Bytes19	Temp3	дополнительная температура 3
Bytes20	Temp4	дополнительная температура 4
Bytes22	TempCnt	температурный счетчик
Bytes22	Byte	байт состояния всех контуров

## Индикатор

Переменная	Имя	Значение
Bytes0	ParNum	номер параметра
Bytes1	Value	состояние индикатора

## Аналоговый параметр

Переменная	Имя	Значение
Bytes0	ParNum	номер параметра
Bytes1	OnlyPositive	запретить отрицательные значения
Bytes2		не используется
Bytes3		не используется
Bytes4-Bytes7	K (четыре байта)	коэффициент отображения (тип float)
Bytes8-Bytes11	Zero (четыре байта)	значение нуля датчика (код)
Bytes12- Bytes15	Change (четыре байта)	Изменение для записи в журнал (код)
Bytes16- Bytes19	Value (четыре байта)	значение параметра
Bytes20- Bytes23	LogValue (четыре байта)	последнее значение параметра в журнале

## Тумблер

Переменная	Имя	Значение
Bytes0	State	состояние тумблера

## Программирование БПДД

Блок передачи данных дуплексный позволяет выполнить прием и передачу байтов внешнему устройству по последовательному интерфейсу. Для этого следует задать блоку программное имя (например **Bpdd**) и использовать специальные переменные, связанные с блоком БПДД.

## Сброс

Перед выполнением операции приема/передачи с использованием БПДД необходимо выполнить сброс БПДД. При выполнении сброса очищаются внутренние буферы приема и передачи, а также выполняется настройка БПДД на прием четных команд по интерфейсу СОС-95. Рекомендуется выполнять **Reset** перед очередной логической командой обмена по последовательному интерфейсу. Возможна ситуация когда сброса выполнить невозможно – например неисправность блока БПДД, обрыв линии связи до БПДД и др. Поэтому выполнение операции сброса необходимо контролировать через системную переменную **@Error**. Возможные значения переменной **@Error** после выполнения сброса БПДД:

0 – сброс выполнен успешно

4 – нет доступа к БПДД (дальнейший обмен не будет выполнен)

### ПРИМЕР:

Приведенная ниже программа выполняет сброс БПДД и проверяет выполнение сброса:

```
Bpdd.Reset=1  
IF @Error>0 THEN PRINT "Не удалось выполнить сброс БПДД";END  
ENDIF
```

## Посылка символа

Посылка байта в последовательный интерфейс выполняется через переменную **Byte** блока БПДД. Запись в эту переменную вызывает запись байта в передающий буфер БПДД. Если буфер пуст, то передача выполняется сразу. Результат записи следует контролировать через системную переменную **@Error** :

0- послано успешно

4- нет доступа к БПДД

5-не удалось послать символ

### ПРИМЕР:

Приведенная ниже программа выполняет сброс БПДД, проверяет выполнение сброса и посылает символ 05:

```
Bpdd.Reset=1  
IF @Error>0 THEN PRINT "Не удалось выполнить сброс БПДД";END  
ENDIF  
Bpdd.Byte=5  
IF @Error>0 THEN PRINT "Не удалось послать символ";END  
ENDIF
```

## Прием символа

Прием байта из последовательного интерфейса выполняется через переменную **Byte** блока БПДД. При чтении происходит извлечение принятого символа из внутреннего буфера БПДД или ожидание символа в течении **TimeOut** миллисекунд.

Результат чтения следует контролировать через системную переменную **@Error** :

0- принято успешно

4- нет доступа к БПДД

6-не удалось принять символ

### ПРИМЕР:

Приведенная ниже программа выполняет сброс БПДД, проверяет выполнение сброса, посылает символ 05 и принимает ответный байт:

```
Bpdd.Reset=1  
IF @Error>0 THEN PRINT "Не удалось выполнить сброс БПДД";END  
ENDIF  
Bpdd.Byte=5  
IF @Error>0 THEN PRINT "Не удалось послать символ";END  
ENDIF  
var= Bpdd.Byte  
IF @Error>0 THEN PRINT "Не удалось принять символ";END
```

```
ENDIF
PRINT "Получен символ ",var
END
```

## Проверка приемного буфера

Количество символов во внутреннем приемном буфере БПДД можно получить через системную переменную **InBuf**. Переменная доступна только для чтения. Результат выполнения чтения этой переменной проверяется через системную переменную **@Error**:

0- полученные данные достоверны

4- нет доступа к БПДД (полученное значение недостоверно)

**ПРИМЕР:**

```
cnt=Vpdd.InBuf
IF @Error>0 THEN PRINT "Нет доступа к БПДД";END
ENDIF
PRINT "В приемном буфере",cnt, " байт"
```

## Установка таймаута

Таймаут приема одного символа из последовательного интерфейса доступен для установки и для чтения в переменной **TimeOut**. Значение таймаута задается в миллисекундах.

**ПРИМЕР:**

```
Vpdd.TimeOut=2000
```

## Программа БПДД

На карте БПДД отображается в виде специальной картинке. При нажатии левой кнопкой мышки на картинку вызывается специальная программа на языке БЕЙСИК, которая дает дополнительные возможности по получению данных из внешних устройств. Например, эта программа выполняет чтение некоторых данных из внешнего теплосчетчика по последовательному интерфейсу, записывает принятые данные в файл и вызывает внешнюю программу или функцию из динамической библиотеки. Внешнее приложение считывает полученные данные из файла и отображает все необходимые пользователю данные. Таким образом выполняется интеграция стороннего оборудования в СЛДКС.

Для создания дополнительной программы БПДД следует просто поместить необходимую программу на языке БЕЙСИК в каталог СЛДКС и в свойствах БПДД указать имя файла программы. Альтернативным вариантом БЕЙСИК программы является использование специального командного файла **VPDD**. Для этого вместо имени программы на языке БЕЙСИК (\*.bas) необходимо указать командный файл с расширением файла «**VPD**». Командный файл может содержать следующие строки:

```
;сообщение, выдаваемое в случае ошибки
MESSAGE <TEXT>
```

```
;сброс приемных буферов БПДД
RESET
```

```
;послать байт
SEND <BYTE>
```

```
;ожидать байт
WAIT <BYTE>
```

```
;принять N байт в файл FILENAME
TOFILE < FILENAME > <N>
```

```
;послать все байты из файла
FROMFILE <FILENAME>
```

```
;запустить приложение FILENAME
RUN <FILENAME>
```

**;выполнить функцию FUNC из библиотеки FILENAME**  
**RUN <FILENAME> <FUNC>**

**;установка таймаута приема в N мс**  
**TIMEOUT <N>**

**;завершение командного файла**  
**END**

Здесь <FILENAME> <FUNC> - тексты без кавычек, символы пробел внутри текста должны быть заменены на символ подчеркивания.

<BYTE> <N> - десятичные числа

Пример:

Данный пример предполагает, что выход БПДД замкнут на вход. Первоначально выдается байт 01, затем проверяется, что получен байт 01. Далее посылаются 4 байта, которые будут записаны в файл "1.TMP". В конце запускается приложение WinAmp. Если на любом этапе произойдет ошибка, то выполнение командного файла будет прервано и пользователю будет выдано сообщение «Не\_удается\_прочитать\_данные\_...»:

```
;сообщение в случае какой-либо ошибки  
MESSAGE Не_удается_прочитать_данные_...  
;сброс приемных буферов БПДД  
RESET  
;установка таймаута приема байта  
TIMEOUT 250  
;послать байт 01  
SEND 01  
;ожидать байт 01  
WAIT 01  
;послать байт 02  
SEND 02  
SEND 03  
SEND 255  
SEND 128  
;принять 4 байта в файл 1.TMP  
TOFILE 1.TMP 4  
;запустить приложение  
RUN C:\Program_Files\WinAmp\WinAmp.exe  
;завершение командного файла  
END
```

## Программная работа с графиками

Если график не связан ни с каким параметром, то можно организовать вывод на график данных из программы. Для этого следует выполнить следующие действия:

1. Добавьте на карту график и задайте все необходимые параметры отображения графика. Задайте в свойствах графика программное имя по которому можно будет обращаться к графику. На рисунке ниже показано, что графику задано программное имя **Graph1**:

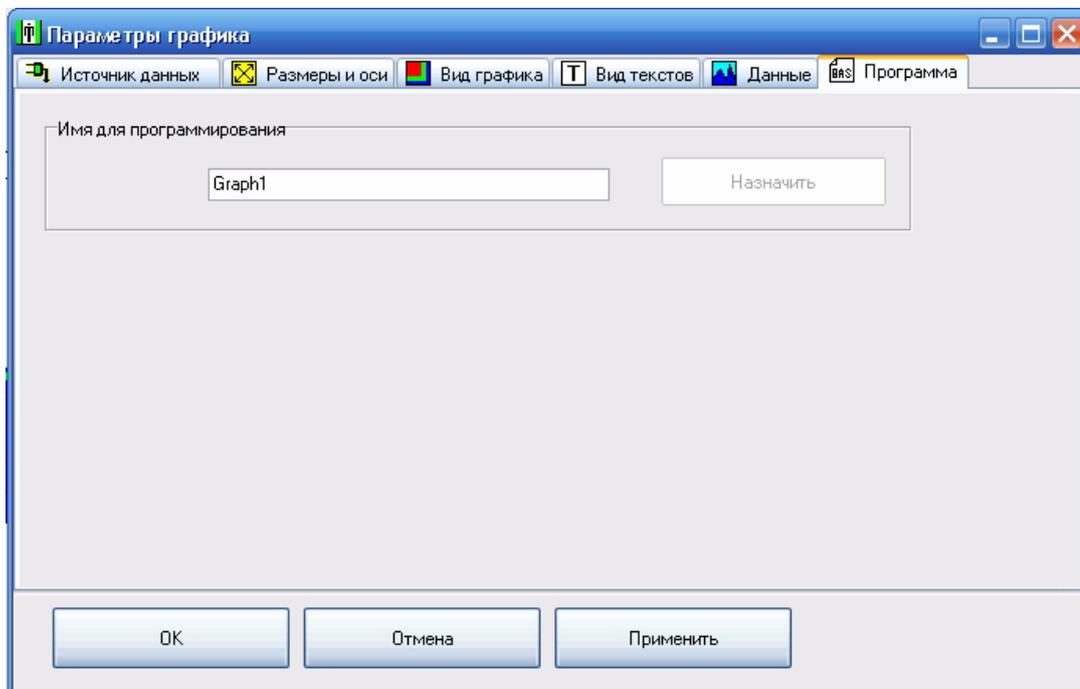


Рисунок – Назначение программного имени графику

2. Далее можно задавать(или читать) значение, отображаемое на графике через свойство графика Value.

Пример:

```
DIM i=50 as INTEGER  
Graph1.Value=i  
i=i+1  
if i>100 then i=0  
ENDIF  
END
```

На рисунке ниже показано как выглядит Graph1 на карте:

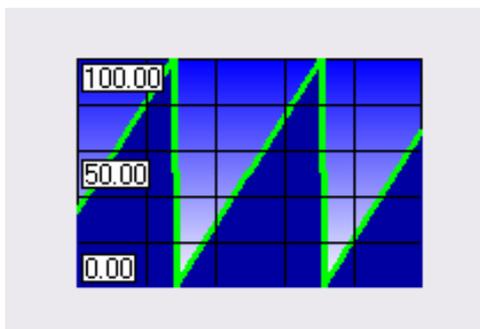


Рисунок – Пример отображения графика

3. Также можно задавать (и получать) текст названия параметра графика:  
Пример:

Graph1.ParName="Тестовый программный параметр"

## Программная работа с текстом

На карте можно создать объект типа «Текст» (на вкладке «Отображение» при создании объектов). У данного объекта может быть задано программное имя в свойствах объекта:

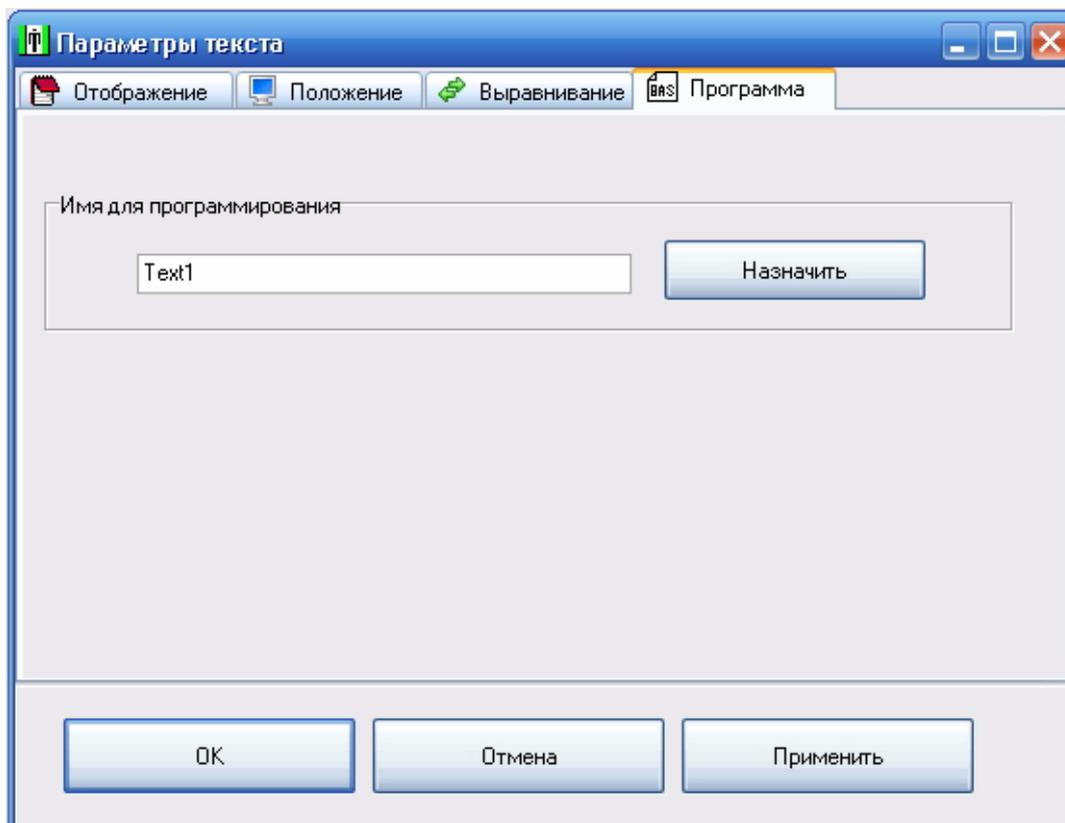


Рисунок – Назначение программного имени тексту

После этого в БЕЙСИК – программе доступны следующие свойства указанного текста:

<b>Caption</b>	Текстовая надпись, отображаемая в этом элементе
<b>Color</b>	Цвет фона текста
<b>Transparent</b>	– прозрачность текста: 0 – отображать фон текста, ненулевое значение – не отображать фон текста
<b>FontColor</b>	Цвет текста
<b>Visible</b>	Признак видимости текста

Пример:

```
DIM i=50 as INTEGER
Graph1.Value=i
i=i+1
if i>100 then i=0
ENDIF
print i
Text1.Caption="Значение: "+i
IF i&1 then Text1.FontColor=255
Text1.Transparent=false
else Text1.FontColor=0
Text1.Transparent=true
ENDIF
END
```