

КОМПЛЕКС АППАРАТУРЫ РЕЧЕВОЙ СВЯЗИ  
«КАРС Топаз»

Руководство по установке ПО подсистемы Коллектора мониторинга  
по протоколу SNMP версии 3.х.х.х

ЦИВР.00530-01 34 01

Листов 18

2015

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

## СОДЕРЖАНИЕ

<b>1 АННОТАЦИЯ.....</b>	<b>3</b>
<b>2 ОБЩИЕ СВЕДЕНИЯ О ПРОГРАММЕ .....</b>	<b>3</b>
<b>3 СИСТЕМНЫЕ ЗАВИСИМОСТИ .....</b>	<b>4</b>
3.1 УСТАНОВКА ИНТЕРПРЕТАТОРА PHP ПОД WINDOWS.....	4
3.2 УСТАНОВКА ИНТЕРПРЕТАТОРА PHP ПОД LINUX (RH, FEDORA).....	4
3.3 УСТАНОВКА ПАКЕТА NET-SNMP ПОД WINDOWS .....	5
3.4 УСТАНОВКА ПАКЕТА NET-SNMP ПОД LINUX .....	5
3.5 УСТАНОВКА ПАКЕТА RSYNC.....	5
<b>4 ОСОБЕННОСТИ ОБНОВЛЕНИЯ ПО .....</b>	<b>6</b>
<b>5 СТРУКТУРА ПРОГРАММЫ .....</b>	<b>8</b>
5.1 ИСПОЛНЯЕМЫЕ МОДУЛИ И БИБЛИОТЕКИ.....	8
5.2 КОНФИГУРАЦИОННЫЕ ФАЙЛЫ.....	8
<b>6 НАСТРОЙКА ПРОГРАММЫ .....</b>	<b>9</b>
6.1 УСТАНОВКА И НАСТРОЙКА ДЕМОНА (СЛУЖБЫ) SNMPTRAPD ПАКЕТА NET-SNMP .....	9
6.2 УСТАНОВКА СЛУЖБЫ.....	10
<b>7 ТОНКАЯ НАСТРОЙКА СТАТУСОВ И ИНДИКАТОРОВ В БАЗЕ ДАННЫХ.....</b>	<b>13</b>
7.1 ОБЩИЕ СВЕДЕНИЯ.....	13
7.2 ПЕРЕМЕННЫЕ МОНИТОРИНГА .....	15
7.3 ИНДИКАТОРЫ .....	16
7.4 ЖУРНАЛ МОНИТОРИНГА .....	18

## 1 Аннотация

Настоящее Руководство содержит сведения о назначении и функциях Коллектора Мониторинга (КМ) по протоколу SNMP; о технических средствах, обеспечивающих выполнение программы; о структуре и настройке программы; а также описание способов проверки, позволяющих дать заключение о работоспособности КМ.

## 2 Общие сведения о программе

КМ является подсистемой СТКУ (Системы Технического Контроля и Управления), отвечающая за актуальность индикаторов состояния системных элементов и элементов замены в БД СТКУ.

В состав КМ входят:

- **исполнительный модуль службы коллектора (snmp-collector[.exe])** – реализует логику сбора статистики о состоянии наблюдаемых объектов;
- **модуль перенаправления событий SNMP trap (process-trap-forward.php)** - предназначен для перенаправления событий от службы snmptrapd к службе коллектора.

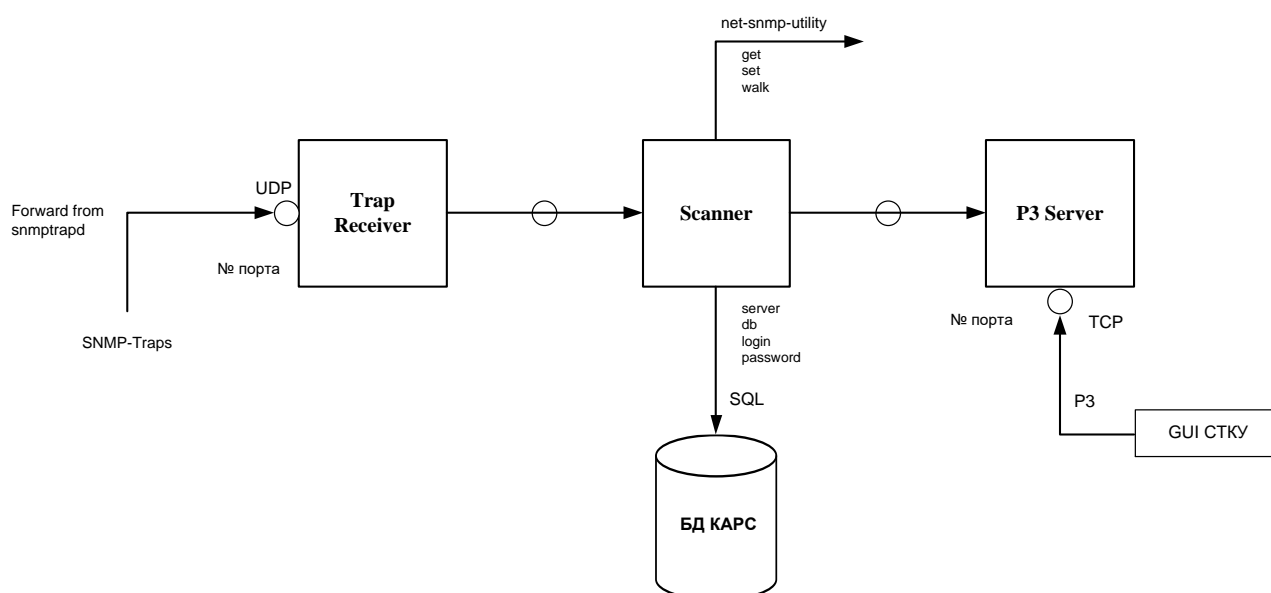


Рисунок 1 – Архитектура SNMP-collector-a

### 3 Системные зависимости

Для работы подсистемы необходимо, чтобы были установлены следующие дополнительные пакеты:

- Интерпретатор PHP 5.x.x или более новый (установлен и доступен в системных путях).
- Пакет поддержки протокола SNMP net-snmp 5.7.x или более новый (установлен и доступен в системных путях).
- Должен быть обеспечен доступ к БД СТКУ (см. Инструкцию по СТКУ).
- Runtime пакеты Qt 4.8.1-5 (см. Инструкцию по СТКУ).

#### 3.1 Установка интерпретатора php под Windows

1. Необходимо распаковать содержимое архива **php-win-5-4-11.zip** в отдельный каталог. Этот архив располагается в каталоге \dependencies. Для определенности скажем пусть это каталог «с:\php».

2. Добавить этот каталог в переменную окружения PATH для всех пользователей, с тем, чтобы исполняемые модули были доступны из любого окружения.

3. Проверить корректность установки, например, выполнив из командной строки команду: «**php -v**». В случае верной установки получим ответ:

```
>php -v
PHP 5.4.11 (cli) (built: Jan 16 2013 20:26:43)
Copyright (c) 1997-2013 The PHP Group
Zend Engine v2.4.0, Copyright (c) 1998-2013 Zend Technologies
```

#### 3.2 Установка интерпретатора php под Linux (RH, Fedora)

1. Выполнить от пользователя root команду «**yum install php**».

2. Проверить корректность установки, выполнив команду «**php -v**». В случае верной установки получим ответ:

```
[]# php -v
PHP 5.4.16 (cli) (built: Jun 6 2013 09:20:50)
Copyright (c) 1997-2013 The PHP Group
Zend Engine v2.4.0, Copyright (c) 1998-2013 Zend Technologies
```

В случае использования других сборок Linux, отличных от RH, необходимо осуществить установку при помощи других систем контроля пакетов, отличных от yum или собрать интерпретатор из исходных кодов, используя инструкцию по сборки интерпретатора php.

### 3.3 Установка пакета **net-snmp** под Windows

1. Необходимо распаковать содержимое архива **net-snmp-win-5-7-2.zip** в корень диска C:. Этот архив располагается в каталоге \dependencies.

2. Добавить этот каталог «**c:\usr\bin\**» в переменную окружения PATH для всех пользователей, с тем, чтобы исполняемые модули были доступны из любого окружения.

3. Проверить корректность установки, например, выполнив из командной строки команду: «**snmpget -V**». В случае верной установки получим ответ:

```
>snmpget -V
NET-SNMP version: 5.7.2
```

4. Установить службы, выполнив команды от имени администратора:

```
«snmpd.exe -register -quiet -Ln»
«snmptrapd.exe -register -quiet -Ln»
```

5. Перезапустить устройство.

### 3.4 Установка пакета **net-snmp** под Linux

1. Выполнить от пользователя root команду «**yum install net-snmp**». Для дистрибутива RH этого достаточно. Для дистрибутивов Fedora 17-20 пакеты net-snmp семантически разделены. Здесь нам необходимо установить кроме net-snmp ещё пакет net-snmp-utils. Устанавливаем его командой «**yum install net-snmp-utils**».

2. Проверить корректность установки, выполнив, например, команду «**snmpget -V**». В случае верной установки получим ответ:

```
[]# snmpget -V
NET-SNMP version: 5.7.2
```

В случае использования других сборок Linux, отличных от RH, необходимо осуществить установку при помощи других систем контроля пакетов, отличных от **yum** или собрать пакет из исходных кодов, используя инструкцию по сборке пакета **net-snmp**.

### 3.5 Установка пакета **rsync**

Для успешной работы коллектора в режиме резервирования необходимо установить пакеты rsync и rsync-daemon командой «**yum install rsync rsync-daemon**». Тонкая настройка необходимых свойств описана в документе «Описание взаимодействия по протоколу P5.doc». Тонкую настройку и проверку её корректности должен осуществлять скрипт установки. На данный момент использование протокола P5 возможно только под ОС LINUX.

## 4 Особенности обновления ПО

При обновлении ПО коллектора необходимо выполнить следующие действия.

1. Сделать резервную копию БД. Используйте пункт меню **«Управление БД»** → **«Резервное копирование»** (подробнее см. пункт 15 «Резервирование и восстановление БД» инструкции «Руководство оператора СТКУ. ЦИВР.00531-01 34 01»).

2. Отключить все СТКУ, подключенные к данной БД. Используйте пункт меню **«Питание»** → **«Выход из Графического интерфейса»** или **«Завершение работы системы»** (подробнее см. пункт 17 «Управление питанием» инструкции «Руководство оператора СТКУ. ЦИВР.00531-01 34 01»).

3. Включить логирование. Для включения логирования откройте файл настройки **snmp-collector.conf**. Значение в строке **LogLevel** должно быть указано не менее 10.

```
[MAIN]
LogLevel=10
```

4. После первого запуска обновленного ПО необходимо убедиться, что патч применен успешно. Для проверки необходимо открыть нужный (дата, время) лог-файл, пример см. ниже.

//Расположение лог-файла - /opt/ksrs-tech/snmp/Logs/2020\_01\_25/15\_snmp-collector.log

```
15:54:25.573: open new log file
15:54:25.573: SYSTEM STARTED [version=3.0.2 build 63, application=SNMP collector for control
system, logLevel=13, logAddr=127.0.0.1:514]
15:54:25.573: P3 SERVER ON PORT <39001> - SUCCESSFULLY STARTED
15:54:25.574: P5: Start (port 44777)
15:54:25.574: P5: CHANGE STATUS [psNONE->psINIT]
15:54:27.074: P5: CHANGE STATUS [psINIT->psMAIN]
15:54:27.075: SNMP TRAP RECEIVER[int] ON PORT <162> - SUCCESSFULLY STARTED
15:54:27.079: SNMPS: PhpRequester: PHP processor code successfully extracted
15:54:27.286: SNMPS: DB: Start database statistic request - OK
15:54:27.286: P5: p5SYNC: CHANGE MODE [smNone -> smMain]
15:54:27.286: P5: p5SYNC: Try Start BK
[db=/var/ksrs/base/borg/AERO.FDB,bk=/var/ksrs/base/borg/backup/p5operative.fbk.tmp]
15:54:27.288: P5: p5SYNC: process [backup] - started
15:54:27.288: P5: p5SYNC: CHANGE STATUS [smMain~{ smsIdle } -> smsWaitDBBackup]
15:54:27.288: SNMP SCANNER ON <DB=127.0.0.1:/var/ksrs/base/borg/AERO.FDB> - SUCCESSFULLY
STARTED
15:54:30.171: P7 server: P7 open stream - OK [lAddr=0.0.0.0, lPort=50001]
15:54:30.171: P7(P1) SERVER - SUCCESSFULLY STARTED
15:54:30.172: SNMPS: PhpRequester: PHP Requester STDOUT:
<
READY
>
15:54:30.172: SNMPS: PhpRequester: PHP Requester - started
15:54:30.172: P5: p5SYNC: process [backup] - finished [exitCode=1]
15:54:30.173: P5: p5SYNC: process [backup] - error:
gbak: ERROR:cannot open backup file /var/ksrs/base/borg/backup/p5operative.fbk.tmp
gbak:Exiting before completion due to errors

15:54:30.173: P5: p5SYNC: Database backup [/var/ksrs/base/borg/backup/p5operative.fbk.tmp] - FAIL[1]
```

15:54:30.173: P5: p5SYNC: CHANGE STATUS [smMain~{smsWaitDBBackup} -> smsIdle]  
15:54:30.227: SNMPS: DB: Info: try patch database [from 129 to 166]  
15:54:30.227: SNMPS: DB: Start patch 130 [size=5865, compressed=yes]->  
15:54:30.228: SNMPS: DB: Found 39 sql requests  
15:54:30.228: SNMPS: DB: Run sql request 1:  
15:54:30.228: SNMPS: DB: sql:

// Патч 166 успешно применяется

15:54:35.527: SNMPS: DB: Patch 166 successfully applied

5. Выключить логирование.

6. Запустить СТКУ.

## 5 Структура программы

### 5.1 Исполняемые модули и библиотеки

**snmp-collector[.exe]** – исполняемый модуль службы КМ;

**process-trap-forward.php** – модуль расширения логики обработки оповещений службы snmptrapd, доставку событий к службе КМ;

**/sqldrivers/qsqllibased4.dll** – драйвер доступа к БД СТКУ (для систем на ОС Windows).

### 5.2 Конфигурационные файлы

**snmp-collector.conf** – настройка параметров службы КМ;

**snmptrapd.conf** – настройка параметров службы приёма snmp-trap (snmptrapd).

**lemz-snmp-collector.service** – настройка запуска службы коллектора для подсистемы systemd Linux.



## 6 Настройка программы

Для успешного запуска ПО КМ необходимо:

- 1) наличие x86 PC с установленной ОС: Linux (ядро  $\geq 3.6.10$ ) или Windows (XP, Vista, 7 или 8);
- 2) должны быть разрешены системные зависимости (см. пункт 3 настоящей инструкции).

### 6.1 Установка и настройка демона (службы) snmptrapd пакета NET-SNMP

1. Необходимо убедиться, в том, что **snmptrapd.conf** имеется в системе. Для систем под **Windows** он располагается «c:\usr\etc\snmp\snmptrapd.conf». Для систем под **Linux**, чаще всего здесь: «/etc/snmp/snmptrapd.conf».

2. Отредактировать файл для **snmptrapd.conf**.

Его содержание для систем под **Windows** должно быть следующим:

```
ignoreAuthFailure yes
disableAuthorization yes
snmpTrapdAddr UDP:162
traphandle default c:/php/php.exe c:/usr/bin/process-trap-forward.php
```

Содержимое под **Linux** – следующим:

```
ignoreAuthFailure yes
disableAuthorization yes
snmpTrapdAddr UDP:162
traphandle default /usr/bin/php /usr/bin/process-trap-forward.php
```

Содержимое файла под **Linux** отличается путем к файлу-обработчику trap-ов **process-trap-forward.php**. Этот файл необходимо взять из архива под **Windows**.

3. Установить системную службу **snmptrapd**.

Под **Windows** необходимо в командной строке выполнить команду «>**snmptrapd -register -Ln -On -n**». После этого возникнет сообщение об успешной установке службы и в списке системных служб можно обнаружить следующую запись: «**Net-SNMP trap Handler**».

Под **Linux** для автоматического запуска службы необходимо воспользоваться одной из сессий управления системными службами:

1. Скрипт автозапуска запуска <**rc.local**> для систем **Linux(RH)**.
2. Возможности подсистемы **systemd** (в современных системах **Linux(Fedora 17-20)**).

Подробно остановимся на установке службы в подсистему **systemd**. После установки пакета **net-snmp** скрипты автозапуска в этих системах уже размещены в подсистеме **systemd**. Они располагаются в файле **/usr/lib/systemd/system/snmptrapd.service**. По умолчанию его содержание таково:

```
[Unit]
Description=Simple Network Management Protocol (SNMP) Trap Daemon.
After=syslog.target network.target

[Service]
Type=notify
```

```
Environment=OPTIONS="-Lsd"
EnvironmentFile=-/etc/sysconfig/snmptrapd
ExecStart=/usr/sbin/snmptrapd $OPTIONS -f
ExecReload=/bin/kill -HUP $MAINPID
```

```
[Install]
WantedBy=multi-user.target
```

Необходимо откорректировать его следующим образом:

```
[Unit]
Description=Simple Network Management Protocol (SNMP) Trap Daemon.
After=syslog.target network.target
```

```
[Service]
Type=notify
Environment=OPTIONS="-Ln"
EnvironmentFile=-/etc/sysconfig/snmptrapd
ExecStart=/usr/sbin/snmptrapd $OPTIONS -f -On -n
ExecReload=/bin/kill -HUP $MAINPID
```

```
[Install]
WantedBy=multi-user.target
```

Чтобы актуализировать изменения необходимо выполнить команду «**systemctl daemon-reload**». Эта команда заставляет подсистему перечитать конфигурационные файлы доступных служб. Чтобы назначить автозапуск службы, необходимо выполнить команду «**systemctl enable snmptrapd**». Эта команда приводит к возникновению символической ссылки «**/etc/systemd/system/multi-user.target.wants/snmptrapd.service**». Это ссылка указывает подсистеме о необходимости запустить данный скрипт как службу при старте операционной системы. При необходимости запуска, остановки, проверки статуса можно воспользоваться командой «**service snmptrapd start|stop|status**».

## 6.2 Установка службы

1. Скопировать файл **snmp-collector[.exe]** в отдельный каталог.

2. Под **Windows** скопировать в подкаталог **/sqldrivers** библиотеку **qsqlibased4.dll**. Под **Linux** устанавливается специальный пакет (см. **инструкцию СТКУ по доступу к БД**).

3. Запустить приложение. После чего в терминале будет отображен примерно следующий вывод:

```
>snmp-collector
16:57.223: open new log file
16:57.223: SYSTEM STARTED [version=3.0.0 (beta 1) build 4, application=SNMP collector
for control system]
16:57.223: Configuration file <.../snmp-collector.conf> created
16:57.223: Init default: MAIN/LogLevel=8
16:57.223: Init default: DB/Sever=127.0.0.1
16:57.223: Init default: DB/File=d:/lemz/db/aero.fdb
16:57.223: Init default: DB/Login=SYSDBA
16:57.223: Init default: DB/Password=masterkey
48:17.411: Init default: TRAP/ForwardPort=39009
48:17.412: Init default: P3/Port=39001
16:57.239: P3 SERVER ON PORT <39001> - SUCCESSFULLY STARTED
16:57.239: SNMP TRAP RECEIVER ON PORT <39009> - SUCCESSFULLY STARTED
```

4. Остановить приложение нажатием клавиш **[Ctrl]+[C]**.

5. Теперь конфигурационный файл (snmp-collector.conf) с параметрами по умолчанию создан. Далее необходимо отредактировать его параметры:

#### 5.1. Раздел [MAIN]

Параметр **LogLevel** – уровень логирования (значение от 0-7, а бит 3 указывает на необходимость логирования в файл). Если необходимо полностью отключить логирование, то установите данное значение в 0. Чем значение больше (не считая 3-й бит), тем уровень детализации логов больше.

Параметр **Packer=/usr/bin/xz** – путь к архиватору.

Параметр **PackerOptions=-9kef** – опция архиватора.

#### 5.2. Раздел [DB]

Параметр **Sever** – IP-адрес (или имя) сервера СУБД. Для КМ этот параметр ДОЛЖЕН быть **127.0.0.1**. Подключение к удаленному СУБД возможен только для тестовых целей.

Параметр **File** – файл базы данных (БД КСРС) на сервере СУБД.

Параметр **Login** – имя пользователя на сервере СУБД.

Параметр **Password** – пароль пользователя на сервере СУБД.

#### 5.3. Раздел [P3]

Параметр **Port** – порт для подключения графического интерфейса СТКУ (изменять не рекомендуется).

#### 5.4. Раздел [TRAP]

Параметр **ForwardPort** – указывает на UDP порт, по которому будут приниматься сообщения типа TRAP, об асинхронных изменения переменных мониторинга на оборудовании. Значение по умолчанию равно 39009. В этом случае необходимо, чтобы TRAP-ы предварительно обрабатывались службой snmptrapd (см. **раздел 6.1 пункт 2**). Однако коллектор способен самостоятельно обрабатывать TRAP-ы. Для этого необходимо, чтобы этот параметр имел значение 162. В этом случае служба snmptrapd ДОЛЖНА быть отключена.

#### 5.5. Раздел [P7]

Параметр **Enable** – разрешает (запрещает – по умолчанию) запуск подсистемы автоконфигурирования оборудования КАРС. Может принимать значения 0 или 1 (разрешено).

#### 5.6. Раздел [P5]

Параметр **Port** – порт для подключения точек резервирования. Выбирается из общих соображений.

Параметр **Points** – содержит список адресов и портов подключения к коллекторам, объединенным в группу резервирования, разделенных запятой. Например, **Points=192.168.12.1:41777,192.168.12.3:41777**.

Параметр **Group** – номер группы резервирования. Все коллекторы, резервирующие одну базу данных, должны быть объединены в группу с одинаковым номером.

Параметр **Priority** – номер приоритета в группе. Значение применяется при выборе ведущего коллектора. Чем меньше значение, тем выше приоритет.

Параметр **KeepAliveInterval** – интервал времени, измеряемый в миллисекундах, через который коллектор будет высылать в группе резервирования анонсы готовности.

Параметр **KeepAliveTimeout** – время, в течение которого отсутствие анонсов будет считаться недоступностью коллектора.

Параметр **KeepAliveRepeat** – количество повторов посылок анонсов при получении ответов.

Параметр **DBSyncMethod** – тип метода синхронизации БД между коллекторами. На данный момент поддерживается только один тип синхронизации RSYNC (значение 0).

6. Сервер **snmp-collector** имеет возможность быть зарегистрированным как системная служба. Для дополнительной информации необходимо выполнить команду:  
**«snmp-collector --help»**

Для систем под **Windows** необходимо и достаточно выполнить команду:  
**«snmp-collector --install»**

Для систем под **Linux** для установки службы необходимо взять файл **«lemz-snmp-collector.service»** следующего содержания:

```
[Unit]
Description=System SNMP collector Service

[Service]
Type=simple
ExecStart=/opt/ksrs/snmp-collector/snmp-collector --run

[Install]
WantedBy=multi-user.target
```

Отредактировать переменную «ExecStart» исходя из пути расположения модуля. После чего выполнить команду **«systemctl daemon-reload»**. Чтобы назначить автозапуск службы, необходимо выполнить команду **«systemctl enable lemz-snmp-collector»**. Эта команда приводит к возникновению символической ссылки «/etc/systemd/system/multi-user.target.wants/lemz-snmp-collector.service». Это ссылка указывает подсистеме о необходимости запустить данный скрипт как службу при старте операционной системы.

## 7 Тонкая настройка статусов и индикаторов в базе данных

### 7.1 Общие сведения

Структура БД КСРС содержит специальные поля в таблицах системных элементов (TB\_SYS\_ELEMENTS) и элементов замены (TB\_REMOVABLE\_ELEMENTS): STATUS и STATUS\_DESC. Значения этих полей заполняет КМ при помощи вызова процедур БД КСРС, обозначенных сигнатурой SNMP\_SET.

Поле STATUS таблицы TB\_SYS\_ELEMENTS заполняется, исходя из составляющих системный элемент статусов таблицы TB\_REMOVABLE\_ELEMENTS, по формуле: статус системного элемента – это максимальное по величине значение статусов его элементов замены. Значения статусов системных элементов определяются так:

STATUS	STATUS_DESC
0	Normal (Норма)
1	Warning notify (Предупреждение)
2	Warning alarm (Предаварийное состояние)
3	Fail (Авария)
4	Absent (Нет ответа о статусе)

Для специфической обработки событий от конкретного оборудования, исходя из типа системного элемента, в таблице TB\_SNMP\_LOGIC должна быть объявлена соответствующая процедура. Это процедура реализуется в линейном коде без подпрограмм на языке php.

Этот код может быть написан, например, так:

```
//.1.3.6.1.4.1.120312.1.4.1.1.0
//Global status:
//0 - OK
//>0 - error
$gStatus = NULL;
if (isset($IN_DATA['.1.3.6.1.4.1.120312.1.4.1.1.0']))
    $gStatus = $IN_DATA['.1.3.6.1.4.1.120312.1.4.1.1.0'];
if (isset($gStatus))
{
    $old_status=$gStatus['RE_STATUS'];
    $status = 4;
    if (is_numeric($gStatus['VALUE']))
        $status = ($gStatus['VALUE']=='0') ? 0 : 3 ;
    if ($old_status!=$status)
    {
        $des_status = ($status==0) ? 'Normal' : 'Fail code '.$gStatus['VALUE'];
        returnREStatus($gStatus['RE'],$status,$des_status);
    }
}

//.1.3.6.1.4.1.120312.1.4.2.1.8.X
//LK status (X=0..3 - position):
//INTEGER VALUE=0..499
//0 - OK
//>0 - error
for ($i=0;$i<4;$i++)
{
    $gStatus = NULL;
    if (isset($IN_DATA['.1.3.6.1.4.1.120312.1.4.2.1.8.'.$i]))
        $gStatus = $IN_DATA['.1.3.6.1.4.1.120312.1.4.2.1.8.'.$i];
    if (isset($gStatus))
    {
        $old_status=$gStatus['RE_STATUS'];
        $status = 4;
        if (is_numeric($gStatus['VALUE']))
```

```

    $status = ($gStatus['VALUE']=='0') ? 0 : 3 ;
    if ($old_status!=$status)
    {
        $des_status = ($status==0) ? 'Normal' : 'Fail code '.$gStatus['VALUE'];
        returnREStatus($gStatus['RE'],$status,$des_status);
    }
}
}

```

Данный фрагмент кода реализует обработку изменения SNMP-переменной с OID=.1.3.6.1.4.1.120312.1.4.1.1.0. Список записей \$IN\_DATA[<OID>] содержит структуры вида:

**array('VALUE' => '%1', 'RE' => '%2', 'RE\_STATUS' => '%3')**

где 'VALUE' – новое значение переменной, 'RE' – уникальный идентификатор элемента замены и 'RE\_STATUS' – предыдущее значение кода статуса элемента замены. Для того чтобы вернуть в КМ значение нового статуса элемента замены необходимо вызвать заранее определённую функцию returnREStatus (<идентификатор элемента замены>,<новый статус>,<новое описание статуса>). Эту функцию желательно вызывать только в случае изменения статуса или его описания.

Если статус элемента замены изменен, то статус системного элемента будет автоматически учтен и подключенные к КМ СТКУ оперативно получают эту информацию.

Существует также таблица TB\_SNMP\_COMMUNITY, предназначенная для тонкой настройки параметров доступа к стеку протокола SNMP для данного системного элемента. Ниже приведено описание полей этой таблицы.

Поле	Тип	Описание
SE_ID	INTEGER	Идентификатор системного элемента.
COMMUNITY	VARCHAR(64)	Кодовое слово для протокола SNMP 2с (по умолчанию – public).
PASSWD	VARCHAR(64)	Пароль к кодовому слову (по умолчанию – пустая строка).
V3AUTH_PROTOCOL	VARCHAR(10)	Метод кодирования запроса авторизации (по умолчанию – MD5).
V3LEVEL	VARCHAR(12)	Уровень защиты. Возможные варианты: noAuthNoPriv authNoPriv authPriv. (по умолчанию – noAuthNoPriv).
V3CONTEXT	VARCHAR(64)	Имя контекста авторизации (по умолчанию – пустая строка).
V3PRIV_PROTOCOL	VARCHAR(10)	Протокол обеспечения приватности (по умолчанию – DES).
V3USER_NAME	VARCHAR(64)	Приватное имя пользователя (по умолчанию – пустая строка).
V3PRIV_PASSWD	VARCHAR(64)	Пароль приватности для режима authPriv (по умолчанию – пустая строка).

Выполнять вышеописанную настройку необходимо в 2-х случаях: если применяется протокол SNMPv3 или в SNMPv2 применяется отличный от public community.

## 7.2 Переменные мониторинга

Основой (наравне с логикой, заложенной в таблице TB\_SNMP\_LOGIC) системы мониторинга в БД КСРС являются наборы прототипов SNMP-переменных, которые хранятся в таблице TB\_SNMP\_PARAMS. Таблица имеет следующую структуру:

Поле	Тип	Описание
PARAM_TYPE_ID	INTEGER	Идентификатор типа переменной (первичный ключ).
OID	VARCHAR(128)	SNMP OID описатель в численном формате.
SE_TYPE_ID	INTEGER	Код типа системного элемента.
RE_TYPE_ID	INTEGER	Код типа элемента замены.
OID_BY_PLACE	SMALLINT	Если значение не равно 0, то при генерации реальной переменной OID будет построен с учетом значения PLACE у элемента замена: $OID+'.'+(PLACE+OID\_BY\_PLACE-1)$ .
SCAN_STEP	INTEGER	Период сканирования переменной (в секундах). Если значение равно 0, то переменная сканируется единожды при старте, а изменения должны доставляться при помощи событий inform или trap.
IS_TRAP_OID	SMALLINT	Если значение равно 1, то переменная является описателем события trap (inform) и не будет сканироваться никогда. В случае получения указанного трапа будет осуществлен перезапрос всех переменных элемента замены. Не следует декларировать переменные трапов, если процедура полного пересчёта значений переменных не требуется.
METHOD	VARCHAR(16)	Метод сканирования переменной. Формат: <code>&lt;method&gt;[_&lt;version&gt;]</code> . Метод, на момент написания документа, – только get. Если версия не указана, то запрос get формируется в версии 2с.
TIMEOUT	INTEGER	Величина таймаута (в секундах) после которого запрос переменной считается неуспешным.
RETRY_COUNT	INTEGER	Необходимое количество попыток в случае таймаута.
DESCRIPTION	VARCHAR(128)	Символьное описание переменной.

Данная таблица является источником данных для другой таблицы TB\_SNMP\_PARAM\_STATUSES. Элементы этой таблицы создаются автоматически при возникновении в системе очередного элемента замены. Таблица имеет следующую структуру:

Поле	Тип	Описание
PARAM_ID	INTEGER	Идентификатор переменной.
RE_ID	INTEGER	Идентификатор элемента замены.
LOG_ID	BIGINT	Идентификатор события в системном журнале.
OID	VARCHAR(128)	См. TB_SNMP_PARAMS
SCAN_STEP	INTEGER	См. TB_SNMP_PARAMS



LAST_SCAN_MOMENT	TIMESTAMP	Астрономическое время момента последнего сканирования переменной.
IS_TRAP_OID	SMALLINT	См. TB_SNMP_PARAMS
METHOD	VARCHAR(16)	См. TB_SNMP_PARAMS
TIMEOUT	INTEGER	См. TB_SNMP_PARAMS
RETRY_COUNT	INTEGER	См. TB_SNMP_PARAMS

Таким образом, значения TB\_SNMP\_PARAMS заводятся при регистрации типа системного элемента и комплектов элементов замены. Остальные действия по получению значений переменных должны осуществляться автоматически КМ.

### 7.3 Индикаторы

На любую совокупность переменных (из таблицы TB\_SNMP\_PARAMS) можно «прикрепить» индикатор, который в дальнейшем будет отображаться в СТКУ отдельно.

Рассмотрим конфигурирование индикаторов на примере. Пусть необходимо переменную №4 (в таблице TB\_SNMP\_PARAMS переменная имеет атрибут PARAM\_TYPE\_ID, равный 4 и имеет OID= .1.3.6.1.4.1.120312.3.1.2.0) вывести отдельным индикатором для системного элемента типа «ЦТРС». Эта переменная отображает статус оператора терминала ЦТРС и может принимать следующие значения:

Значение	Описание	Статус индикатора
0	Графический интерфейс (сервисное приложение) ЦТРС отсутствует	4 - авария недоступности терминала.
1	Оператор не занят соединениями (не имеет активных разговоров)	0 - нормальное состояние.
2	Оператор занят разговором в радиозфире	0 - нормальное состояние.
3	Оператор занят разговором по наземной связи	0 - нормальное состояние.
4	Оператор заблокировал ЦТРС (отсутствует и не может принять вызов)	1 - предупреждение начального уровня.

Для заведения индикатора необходимо проделать следующие операции:

1. Зарегистрировать индикатор в таблице TB\_SNMP\_INDICATOR\_LIST:

Поле	Тип	Описание
IND_TYPE_ID	INTEGER	Идентификатор типа индикатора (первичный ключ). Уникальное значение по порядку от единицы.
NAME	VARCHAR(128)	Название типа индикатора.
LOGIC	BLOB SUB_TYPE 1 SEGMENT SIZE 4096	Логика обработки индикатора.

Заполняем поля, например, так:

Поле	Значение
IND_TYPE_ID	1
NAME	Статус оператора ЦТРС
LOGIC	// .1.3.6.1.4.1.120312.3.1.2.0



```

//Status of terminal operator:
//0 - no operator present
//1 - operator free
//2 - operator on radio
//3 - operator on ground
//4 - operator blocked
$gStatus = NULL;
if (isset($IN_DATA['.1.3.6.1.4.1.120312.3.1.2.0']))
    $gStatus = $IN_DATA['.1.3.6.1.4.1.120312.3.1.2.0'];
if (isset($gStatus))
{
    $status = 4;
    $des_status = 'Terminal isn\'t available for the operator';
    if (is_numeric($gStatus['VALUE']))
    {
        $status = 0;
        if ($gStatus['VALUE']==1)
        {
            $des_status = 'Operator is ready for talk';
        }else
        {
            if ($gStatus['VALUE']==2)
            {
                $des_status = 'Operator is used RADIO';
            }else
            {
                if ($gStatus['VALUE']==3)
                {
                    $des_status = 'Operator is used GGS(PSTN)';
                }else
                {
                    if ($gStatus['VALUE']==4)
                    {
                        $status = 1;
                        $des_status = 'Terminal blocked by operator';
                    }else
                    {
                        $status = 4;
                    }
                }
            }
        }
    }
}
returnINDStatus($gStatus['IND'],$status,$des_status);
}

```

Данный фрагмент кода реализует обработку изменения SNMP-переменной с OID=.1.3.6.1.4.1.120312.3.1.2.0 для изменения индикатора статуса оператора ЦТРС. Список записей \$IN\_DATA[<OID>] содержит структуры вида:

**array('VALUE' => '%2', 'IND' => %3, 'IND\_STATUS' => %4, 'IND\_VALUE' => '%5')**

где 'VALUE' – новое значение переменной, 'IND' – уникальный идентификатор индикатора, 'IND\_STATUS' – предыдущее значение кода статуса индикатора и 'IND\_VALUE' – предыдущее значение индикатора. Для того чтобы вернуть в КМ значение нового индикатора необходимо вызвать функцию returnINDStatus (<идентификатор индикатора>,<новый статус индикатора>,<новое значение индикатора>). Эту функцию желательно вызывать только в случае изменения статуса или значения индикатора.

2. Заполнить список типов переменных, по которым вычисляется индикатор в таблице TB\_SNMP\_INDICATOR\_PARAMS:

Поле	Тип	Описание
IND_TYPE_ID	INTEGER	Идентификатор типа индикатора (внешний ключ на таблицу TB_SNMP_INDICATOR_LIST).

PARAM_TYPE_ID	INTEGER	Идентификатор типа переменной (внешний ключ на таблицу TB_SNMP_PARAMS).
---------------	---------	---

Например, так:

Поле	Значение
IND_TYPE_ID	1
PARAM_TYPE_ID	4

Теперь вновь введенные ЦТПС будут иметь описанный нами индикатор.

## 7.4 Журнал мониторинга

Основным хранилищем журнала мониторинга является таблица TB\_SNMP\_EVENT\_LOG:

Поле	Тип	Описание
LOG_ID	BIGINT	Уникальный идентификатор записи в журнал.
RE_ID	INTEGER	Идентификатор элемента замены, относительно которого произведена запись в журнал.
SE_ID	INTEGER	Идентификатор системного элемента.
MOMENT	TIMESTAMP	Момент времени, когда запись была произведена.
LOG_TYPE	SMALLINT	Код тип события. На данный момент определены следующие типы событий: 0 - изменен статус элемента замены; 1 - изменена некоторая SNMP переменная, 2 – изменено значение некоторого индикатора.
DESCRIPTION	VARCHAR(4096)	Символьное описание события.

Все изменения статусов элементов замены и индикаторов сохраняются в таблице TB\_SYS\_MONITORING\_LOG:

Поле	Тип	Описание
LOG_ID	BIGINT	Уникальный идентификатор записи в журнал.
SE_ID	INTEGER	Идентификатор системного элемента.
RE_ID	INTEGER	Идентификатор элемента замены, относительно которого произведена запись в журнал.
IND_ID	INTEGER	Идентификатор индикатора.
MOMENT	TIMESTAMP	Момент времени изменения статуса.
STATUS	INTEGER	Новое значение статуса.
DESCRIPTION	VARCHAR(2048)	Описание статуса.