Описание мониторинга параметров устройств под управлением контроллеров серии МСК по протоколу SNMP.

Поддерживаемая версия протокола: SNMPv2.

Поддерживаемые community: "public" для доступа "только для чтения", "private" для доступа на "чтение-запись".

Базовая ветка дерева MIB поддерживаемых устройств: 1.3.6.1.4.1.27142.1.12 (список устройств и их параметров приведен ниже)

Поддержка команд SNMP getnext/walk: выполняются только для подветок ветки 1.3.6.1.4.1.27142.1.12. отдельно.

Поддержка snmp-trap пакетов: отправляются snmp-inform сообщения при появлении аварийных ситуаций (список snmp-inform сообщений приведен ниже).

Сообщения snmp-trap высылаются сразу при возникновении аварийной ситуации. Если удаленный хост, куда отправляются аварии, перестает быть доступным, то после того как он станет доступен, произойдет отправка сообщения ***SystemInitTrap*** для сброса аварий и затем отправятся все текущие аварии. Есть сообщения snmp-trap высылающиеся периодически(каждые 5 минут) — это информирующие сообщения о состоянии напряжения 220В(***SupplyUPhaseTrap***) и температурных датчиках(***ClimateTempStat***).

Настройка отправки snmp-trap пакетов осуществляется в графическом интерфейсе программы конфигурирования и мониторинга mskmon, описание приведено ниже.

Список поддерживаемых устройств и параметров (MIB).

```
--------------------------------------------------------------------------------
--      UGM //!!!
--------------------------------------------------------------------------------
  supply            OBJECT IDENTIFIER ::= { ugm 1 }
  guard             OBJECT IDENTIFIER ::= { ugm 2 }
  climate           OBJECT IDENTIFIER ::= { ugm 3 }
  dshdsl            OBJECT IDENTIFIER ::= { ugm 4 }
  sdfe                      OBJECT IDENTIFIER ::= { ugm 5 }
  ump                       OBJECT IDENTIFIER ::= { ugm 6 }
  kns                       OBJECT IDENTIFIER ::= { ugm 7 }
  idp240            OBJECT IDENTIFIER ::= { ugm 8 }
  idp350            OBJECT IDENTIFIER ::= { ugm 9 }
  akb                       OBJECT IDENTIFIER ::= { ugm 10 }
  load                      OBJECT IDENTIFIER ::= { ugm 11 }
  elmetr            OBJECT IDENTIFIER ::= { ugm 12 }
  idp350v13                 OBJECT IDENTIFIER ::= { ugm 13 }
  puv2                      OBJECT IDENTIFIER ::= { ugm 14 }
  uka                       OBJECT IDENTIFIER ::= { ugm 15 }
  ukn                       OBJECT IDENTIFIER ::= { ugm 16 }


--------------------------------------------------------------------------------
--          Supply //!!!
--


----------------------
--          SUPPLY PARAMS //!!!
--


  supplyParams              OBJECT IDENTIFIER ::= { supply 1 }
  supplyParamList OBJECT IDENTIFIER ::= { supplyParams 1 }
```

```
  supplyUPhase OBJECT-TYPE
SYNTAX      DisplayString
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
          "A supply voltage."
::= { supplyParamList 1 }

  supplyUHZ   OBJECT-TYPE
SYNTAX      DisplayString
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
          "A line frequency."
::= { supplyParamList 2 }

  supplyUDC   OBJECT-TYPE
SYNTAX      DisplayString
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
          "A direct-current supply voltage."
::= { supplyParamList 3 }

  supplyUDCNom   OBJECT-TYPE
SYNTAX      DisplayString
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
          "A direct-current nominal supply voltage."
::= { supplyParamList 4 }
```

```
----------------------
--            SUPPLY TRAPS //!!!
--


 supplyTraps                    OBJECT IDENTIFIER ::= { supply 2 }
 supplyTrapList    OBJECT IDENTIFIER ::= { supplyTraps 1 }


supplyAlarm NOTIFICATION-TYPE
 OBJECTS          { supplyUPhase, supplyUDC }
 STATUS     current
 DESCRIPTION
       "A supplyTrap trap is sent when the
       value of supplyUDC or supplyUPhase is critical."
 ::= { supplyTrapList 1 }


supplyNormal NOTIFICATION-TYPE
 OBJECTS          { supplyUPhase, supplyUDC }
 STATUS     current
 DESCRIPTION
       "A supplyTrap trap is sent when the
       value of supplyUDC or supplyUPhase is normal."
 ::= { supplyTrapList 2 }


supplyUPhaseTrap NOTIFICATION-TYPE
 OBJECTS { supplyUPhase }
 STATUS     current
 DESCRIPTION
       "A supply trap of supplyUPhase sends periodically."
 ::= { supplyTrapList 3 }


supplyNoPhaseAlarm NOTIFICATION-TYPE
 STATUS     current
```

```
    DESCRIPTION
            "A supplyTrap trap is sent when the
            value of supplyUPhase is 0."
    ::= { supplyTrapList 4 }

--------------------------------------------------------------------------
--              Guard //!!!
--


----------------------
--              GUARD PARAMS //!!!
--


   guardParams                  OBJECT IDENTIFIER ::= { guard 1 }
   guardParamList   OBJECT IDENTIFIER ::= { guardParams 1 }


  guardKick OBJECT-TYPE
SYNTAX      INTEGER {
                        normal(0),
                        kick  (1)
    }
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
            "A kick case."
::= { guardParamList 1 }


  guardLock   OBJECT-TYPE
SYNTAX      INTEGER {
                        normal(0),
                        lock  (1)
    }
```

```
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
            "A lock case."
::= { guardParamList 2 }


----------------------
--              GUARD TRAPS //!!!
--


  guardTraps                    OBJECT IDENTIFIER ::= { guard 2 }
  guardTrapList     OBJECT IDENTIFIER ::= { guardTraps 1 }


 guardAlarm NOTIFICATION-TYPE
    OBJECTS { guardKick, guardLock }
    STATUS      current
    DESCRIPTION
            "A guardTrap trap is sent when the
            value of an instance guardLock or guardKick alarm."
    ::= { guardTrapList 1 }


 guardNormal NOTIFICATION-TYPE
    OBJECTS { guardKick, guardLock }
    STATUS      current
    DESCRIPTION
            "A guardTrap trap is sent when the
            value of an instance guardLock or guardKick normal."
    ::= { guardTrapList 2 }


--------------------------------------------------------------------------
--              Climate //!!!
--
```

```
----------------------
--            CLIMATE PARAMS //!!!
--


  climateParams     OBJECT IDENTIFIER ::= { climate 1 }
  climateParamListOBJECT IDENTIFIER ::= { climateParams 1 }


  climateStatus   OBJECT-TYPE
SYNTAX      INTEGER {
                        normal(0),
                        alarm (1)
   }
   MAX-ACCESS  read-only
   STATUS      current
   DESCRIPTION
         "Climate status."
::= { climateParamList 1 }


 climateTempCooler   OBJECT-TYPE
SYNTAX      DisplayString
   MAX-ACCESS  read-write
   STATUS      current
   DESCRIPTION
         "Set-on cooler temperature."
::= { climateParamList 2 }


 climateTempChoke   OBJECT-TYPE
SYNTAX      DisplayString
   MAX-ACCESS  read-write
   STATUS      current
   DESCRIPTION
```

```
        "Set-on choke temperature."
::= { climateParamList 3 }

 climateTempHeater   OBJECT-TYPE
SYNTAX      DisplayString
   MAX-ACCESS  read-write
   STATUS      current
   DESCRIPTION
        "Set-on heater temperature."
::= { climateParamList 4 }

 climateSensTemp      OBJECT-TYPE
SYNTAX      DisplayString
   MAX-ACCESS  read-only
   STATUS      current
   DESCRIPTION
        "All sensors temperature."
::= { climateParamList 5 }

----------------------
--            CLIMATE TRAPS //!!!
--

  climateTraps                  OBJECT IDENTIFIER ::= { climate 2 }
  climateTrapList   OBJECT IDENTIFIER ::= { climateTraps 1 }

 climateAlarm NOTIFICATION-TYPE
   OBJECTS { indexTS, temperature }
   STATUS      current
   DESCRIPTION
        "A climate trap is sent when the
        value of an instance temperature critical."
```

```
  ::= { climateTrapList 1 }

 climateNormal NOTIFICATION-TYPE
  STATUS      current
  DESCRIPTION
        "A climate trap is sent when the
        value of an instance temperature critical."
  ::= { climateTrapList 2 }

 coolerAlarm NOTIFICATION-TYPE
  STATUS      current
  DESCRIPTION
        "A cooler trap is sent when cooler may be broken."
  ::= { climateTrapList 3 }

 coolerNormal NOTIFICATION-TYPE
  STATUS      current
  DESCRIPTION
        "A climate trap is sent when cooler is alright(after may be broken)"
  ::= { climateTrapList 4 }

 climateTempStat NOTIFICATION-TYPE
  OBJECTS { climateSensTemp }
  STATUS      current
  DESCRIPTION
        "A climate trap of sensors temperature is sent periodically."
  ::= { climateTrapList 5 }
----------------------
--            CLIMATE TEMPERATURE //!!!
--

  temperatureSensorTable OBJECT-TYPE
```

```
 SYNTAX      SEQUENCE OF TemperatureSensors
 MAX-ACCESS  not-accessible
 STATUS      current
 DESCRIPTION
       "Contains information."
 ::= { climate 3 }

temperatureSensorEntry OBJECT-TYPE
 SYNTAX      TemperatureSensors
 MAX-ACCESS  not-accessible
 STATUS      current
 DESCRIPTION
       "An entry ."
 INDEX   { indexTS }
 ::= { temperatureSensorTable 1 }

TemperatureSensors ::= SEQUENCE {
    indexTS     INTEGER,
    temperature DisplayString
    }

indexTS OBJECT-TYPE
 SYNTAX      INTEGER (1..8)
 MAX-ACCESS  read-only
 STATUS      current
 DESCRIPTION
       "Number of Temperature Sensor.
          UGM          BEP
       1  KNS-2        IDP-1/AKB
       2  KNS-3        IDP-2/KNS-1
       3  AKB         IDP-3
       4  Case         IDP-4/KNS-2
```

```
          5               IDP-5/
          6  KNS/IDP-4    IDP-6/KNS-3
          7  GVS/IDP-5    IDP-7/
          8  GVS-6        IDP-8/KNS-4
          "
   ::= { temperatureSensorEntry 1 }

  temperature OBJECT-TYPE
   SYNTAX      DisplayString
   MAX-ACCESS  read-only
   STATUS      current
   DESCRIPTION
         "This variable indicates the temperature  ( Celsius )."
   ::= { temperatureSensorEntry 2 }

--------------------------------------------------------------------------
--            2SHDSL_2E1 //!!!
--
----------------------
--            2SHDSL_2E1 PARAM //!!!
--

  dshdslParamTable OBJECT-TYPE
   SYNTAX      SEQUENCE OF DSHDSLPARAMEntry
   MAX-ACCESS  not-accessible
   STATUS      current
   DESCRIPTION
        "Contains information."
   ::= { dshdsl 1 }

  dshdslParamEntry OBJECT-TYPE
   SYNTAX      DSHDSLPARAMEntry
```

```
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "An entry containing information about PARAM ports."
    INDEX   { dshdslIndex  }
    ::= { dshdslParamTable 1 }

   DSHDSLPARAMEntry ::= SEQUENCE {
       dshdslIndex                 INTEGER,
       dshdslExist                 INTEGER,
dshdslPll                          INTEGER
       }

   dshdslIndex  OBJECT-TYPE
    SYNTAX      INTEGER (1..16)
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "A unique value, greater than zero."
    ::= { dshdslParamEntry 1 }

   dshdslExist  OBJECT-TYPE
    SYNTAX      INTEGER {
                                not-exist(0),
                                exist(1)
                                }
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "This variable indicates the Exist Board."
    ::= { dshdslParamEntry 2 }
```

```
  dshdslPll  OBJECT-TYPE
   SYNTAX      INTEGER {
                                    off(0),
                                    shdsl-0(1),
                                    shdsl-1(2),
                                    pcm30-0(3),
                                    pcm30-1(4)
                                    }
   MAX-ACCESS  read-write
   STATUS      current
   DESCRIPTION
         "PLL line current 2SHDSL_2E1s."
   ::= { dshdslParamEntry 3 }


----------------------
--             2SHDSL_2E1 TRAPS //!!!
--
  dshdslTraps                    OBJECT IDENTIFIER ::= { dshdsl 2 }
  dshdslTrapList     OBJECT IDENTIFIER ::= { dshdslTraps 1 }


 dshdslBoardUp NOTIFICATION-TYPE
   OBJECTS { dshdslIndex }
   STATUS      current
   DESCRIPTION
         "A deviceTrap trap is sent when the device exist changes."
   ::= { dshdslTrapList 1 }


 dshdslBoardDown NOTIFICATION-TYPE
   OBJECTS { dshdslIndex }
   STATUS      current
   DESCRIPTION
         "A deviceTrap trap is sent when the device exist changes."
```

::= { dshdslTrapList 2 }

dshdslPcm30Alarm NOTIFICATION-TYPE

OBJECTS { dshdslIndex, dshdslPcm30Index, dshdslPcm30State }

STATUS current

DESCRIPTION

"A pcm30ChangeState_dshdsl trap is sent when the

value of an instance pcm30State_dshdsl changes."

::= { dshdslTrapList 3 }

dshdslPcm30Normal NOTIFICATION-TYPE

OBJECTS { dshdslIndex, dshdslPcm30Index, dshdslPcm30State }

STATUS current

DESCRIPTION

"A pcm30ChangeState_dshdsl trap is sent when the

value of an instance pcm30State_dshdsl changes."

::= { dshdslTrapList 4 }

dshdslShdslAlarm NOTIFICATION-TYPE

OBJECTS { dshdslIndex, dshdslShdslIndex, dshdslShdslState }

STATUS current

DESCRIPTION

"A shdslChangeState_dshdsl trap is sent when the

value of an 3instance shdslState_dshdsl changes."

::= { dshdslTrapList 5 }

dshdslShdslNormal NOTIFICATION-TYPE

OBJECTS { dshdslIndex, dshdslShdslIndex, dshdslShdslState }

STATUS current

DESCRIPTION

"A shdslChangeState_dshdsl trap is sent when the

value of an 3instance shdslState_dshdsl changes."

```
    ::= { dshdslTrapList 6 }


----------------------
--              2SHDSL_2E1 PCM30 //!!!
--


   dshdslPcm30Table OBJECT-TYPE
    SYNTAX      SEQUENCE OF DSHDSLPCM30Entry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
          "Contains information."
    ::= { dshdsl 3 }


   dshdslPcm30Entry OBJECT-TYPE
    SYNTAX      DSHDSLPCM30Entry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
          "An entry containing information about PCM30 ports."
    INDEX   { dshdslPcm30Index }
    ::= { dshdslPcm30Table 1 }


   DSHDSLPCM30Entry ::= SEQUENCE {
       dshdslPcm30Index                INTEGER,
       dshdslPcm30State                DisplayString,
dshdslPcm30Status           INTEGER,
dshdslPcm30Code                 INTEGER,
dshdslPcm30Transp               INTEGER,
dshdslPcm30Loop             INTEGER,
dshdslPcm30Nvp                  INTEGER,
dshdslPcm30Pcs                  INTEGER,
```

```
dshdslPcm30Psc                  INTEGER,
dshdslPcm30E3                   INTEGER,
dshdslPcm30E5                   INTEGER,
dshdslPcm30Aus                  INTEGER
    }

  dshdslPcm30Index OBJECT-TYPE
   SYNTAX     INTEGER (1..32)
   MAX-ACCESS  read-only
   STATUS     current
   DESCRIPTION
        "A unique value, greater than zero."
   ::= { dshdslPcm30Entry 1 }

  dshdslPcm30State OBJECT-TYPE
   SYNTAX     DisplayString
   MAX-ACCESS  accessible-for-notify
   STATUS     current
   DESCRIPTION
        "This variable indicates the Line Status of theinterface.
        It contains 'no input stream' (NVP), 'emergency indication
        signal' (SIA), 'cycle synchronization lost' (PCS), E-3, E-5,
        'far end emergency' (AUS), '(far end) supercycle synchronization
        lost' (PSCS(U)). The ifPCM15CurrentState is a bit map represented
        as a sum, therefore, it can represent multiple failures (alarms).
        ifPCM15NoAlarm must be set if and only if no other flag is set.
        The various bitpositions are:
        0   NoAlarm
        1    NVP
        2    SIA
        4    PCS
        8    E-3
```

```
        16    E-5
        32    AUS
        64    PSCS
        128   PSCSU
        "
  ::= { dshdslPcm30Entry 2 }

dshdslPcm30Status  OBJECT-TYPE
  SYNTAX      INTEGER {
                                off(0),
                                on(1)
                                }
  MAX-ACCESS  read-write
  STATUS      current
  DESCRIPTION
        "Status of PCM30 of 2SHDSL_2E1 (off/on)."
  ::= { dshdslPcm30Entry 3 }

 dshdslPcm30Code  OBJECT-TYPE
  SYNTAX      INTEGER {
                                ami(0),
                                hdb3(1)
                                }
  MAX-ACCESS  read-write
  STATUS      current
  DESCRIPTION
        "Current encoding of PCM30 of 2SHDSL_2E1 (ami/hdb3)."
  ::= { dshdslPcm30Entry 4 }

 dshdslPcm30Transp  OBJECT-TYPE
  SYNTAX      INTEGER {
                                off(0),
```

on(1)

}

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"State of transparency of PCM30 of 2SHDSL_2E1 (off/on)."

::= { dshdslPcm30Entry 5 }

dshdslPcm30Loop OBJECT-TYPE

SYNTAX INTEGER {

no(0),

near(1),

far(2),

dual(3)

}

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"Value of loop of PCM30 of 2SHDSL_2E1 (no/far/near/dual)."

::= { dshdslPcm30Entry 6 }

dshdslPcm30Nvp OBJECT-TYPE

SYNTAX INTEGER

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Value of nvp of PCM30 of 2SHDSL_2E1."

::= { dshdslPcm30Entry 7 }

dshdslPcm30Pcs OBJECT-TYPE

SYNTAX INTEGER

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Value of pcs of PCM30 of 2SHDSL_2E1."

::= { dshdslPcm30Entry 8 }

dshdslPcm30Psc OBJECT-TYPE

SYNTAX INTEGER

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Value of psc of PCM30 of 2SHDSL_2E1."

::= { dshdslPcm30Entry 9 }

dshdslPcm30E3 OBJECT-TYPE

SYNTAX INTEGER

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Value of e3 of PCM30 of 2SHDSL_2E1."

::= { dshdslPcm30Entry 10 }

dshdslPcm30E5 OBJECT-TYPE

SYNTAX INTEGER

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Value of e5 of PCM30 of 2SHDSL_2E1."

::= { dshdslPcm30Entry 11 }

dshdslPcm30Aus OBJECT-TYPE

SYNTAX INTEGER

MAX-ACCESS read-only

```
    STATUS      current
    DESCRIPTION
            "Value of aus of PCM30 of 2SHDSL_2E1."
    ::= { dshdslPcm30Entry 12 }


----------------------
--              2SHDSL_2E1 SHDSL //!!!
--


   dshdslShdslTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF DSHDSLSHDSLEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
          "Contains information."
    ::= { dshdsl 4 }

   dshdslShdslEntry OBJECT-TYPE
    SYNTAX      DSHDSLSHDSLEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
           "An entry containing information about SHDSL ports."
    INDEX   { dshdslShdslIndex }
    ::= { dshdslShdslTable 1 }

   DSHDSLSHDSLEntry ::= SEQUENCE {
       dshdslShdslIndex                    INTEGER,
       dshdslShdslState                    DisplayString,
dshdslShdslStatus               INTEGER,
       dshdslShdslDescr                    DisplayString,
dshdslShdslRange                INTEGER,
```

```
dshdslShdslMode                 INTEGER,
dshdslShdslBackoff                      INTEGER,
dshdslShdslChainFull                    INTEGER,
dshdslShdslChainLen                     INTEGER
    }

  dshdslShdslIndex OBJECT-TYPE
   SYNTAX      INTEGER (1..32)
   MAX-ACCESS  read-only
   STATUS      current
   DESCRIPTION
        "A unique value, greater than zero."
   ::= { dshdslShdslEntry 1 }

  dshdslShdslState OBJECT-TYPE
   SYNTAX      DisplayString
   MAX-ACCESS  accessible-for-notify
   STATUS      current
   DESCRIPTION
        "This variable indicates the Line Status of theinterface.
        It contains 'no input stream' (NVP), 'chain shdsl is broken' (ChainBroken)
        The various bitpositions are:
        0    NoAlarm
        1    NVP
        2    ChainBroken
        "
   ::= { dshdslShdslEntry 2 }

  dshdslShdslStatus  OBJECT-TYPE
   SYNTAX      INTEGER {
                                off(0),
                                on(1)
```

```
                                }
 MAX-ACCESS  read-write
 STATUS      current
 DESCRIPTION
        "Status of shdsl of 2SHDSL_2E1 (off/on)."
 ::= { dshdslShdslEntry 3 }

dshdslShdslDescr  OBJECT-TYPE
 SYNTAX      DisplayString
 MAX-ACCESS  read-write
 STATUS      current
 DESCRIPTION
        "Description of place where is shdsl of 2SHDSL_2E1."
 ::= { dshdslShdslEntry 4 }

dshdslShdslRange  OBJECT-TYPE
 SYNTAX      INTEGER
 MAX-ACCESS  read-write
 STATUS      current
 DESCRIPTION
        "Current range of shdsl of 2SHDSL_2E1."
 ::= { dshdslShdslEntry 5 }

dshdslShdslMode  OBJECT-TYPE
 SYNTAX      INTEGER {
                                lt(0),
                                mt(1)
                                }
 MAX-ACCESS  read-write
 STATUS      current
 DESCRIPTION
        "Current mode of shdsl of 2SHDSL_2E1 (lt/mt)."
```

::= { dshdslShdslEntry 6 }

dshdslShdslBackoff OBJECT-TYPE

SYNTAX INTEGER

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"Value of backoff of shdsl of 2SHDSL_2E1."

::= { dshdslShdslEntry 7 }

dshdslShdslChainFull OBJECT-TYPE

SYNTAX INTEGER

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Indicates entirety of chain of shdsl of 2SHDSL_2E1."

::= { dshdslShdslEntry 8 }

dshdslShdslChainLen OBJECT-TYPE

SYNTAX INTEGER

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Length of chain of shdsl of 2SHDSL_2E1."

::= { dshdslShdslEntry 9 }

-- 2SHDSL_2E1 SHDSL SOCIF //!!!

--

dshdslSocifTable OBJECT-TYPE

```
    SYNTAX      SEQUENCE OF DSHDSLSOCIFEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
          "Contains information."
    ::= { dshdsl 5 }

  dshdslSocifEntry OBJECT-TYPE
    SYNTAX      DSHDSLSOCIFEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
          "An entry containing information about SHDSL'S SOCIF ports."
    INDEX   { dshdslSocifIndex }
    ::= { dshdslSocifTable 1 }

  DSHDSLSOCIFEntry ::= SEQUENCE {
      dshdslSocifIndex                INTEGER,
dshdslSocifStatus             DisplayString
      }

  dshdslSocifIndex OBJECT-TYPE
    SYNTAX      INTEGER (1..512)
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
          "A unique value, greater than zero."
    ::= { dshdslSocifEntry 1 }

  dshdslSocifStatus  OBJECT-TYPE
    SYNTAX      DisplayString
    MAX-ACCESS  read-only
```

```
    STATUS      current
    DESCRIPTION
            "Parameters status of socif of shdsl of 2SHDSL_2E1."
    ::= { dshdslSocifEntry 2 }


-------------------------------------------------------------------------
--              2SHDSL_4E1 (SDFE) //!!!
--


----------------
--              SDFE PARAM //!!!
--


   sdfeParamTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF SDFEPARAMEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
          "Contains information."
    ::= { sdfe 1 }


   sdfeParamEntry OBJECT-TYPE
    SYNTAX      SDFEPARAMEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
          "An entry containing information about PARAM ports."
    INDEX   { sdfeIndex }
    ::= { sdfeParamTable 1 }


   SDFEPARAMEntry ::= SEQUENCE {
       sdfeIndex        INTEGER,
```

```
    sdfeExist         INTEGER
    }

  sdfeIndex OBJECT-TYPE
   SYNTAX      INTEGER (1..16)
   MAX-ACCESS  read-only
   STATUS      current
   DESCRIPTION
        "A unique value, greater than zero."
   ::= { sdfeParamEntry 1 }

  sdfeExist OBJECT-TYPE
   SYNTAX      INTEGER {
                                not-exist(0),
                                exist(1)
                                }
   MAX-ACCESS  read-only
   STATUS      current
   DESCRIPTION
        "This variable indicates the Exist Board."
   ::= { sdfeParamEntry 2 }

----------------------
--            SDFE TRAPS //!!!
--
  sdfeTraps                     OBJECT IDENTIFIER ::= { sdfe 2 }
  sdfeTrapList                  OBJECT IDENTIFIER ::= { sdfeTraps 1 }

 sdfeBoardUp NOTIFICATION-TYPE
   OBJECTS { sdfeIndex }
   STATUS      current
   DESCRIPTION
```

"A deviceTrap trap is sent when the device exist changes."

::= { sdfeTrapList 1 }

sdfeBoardDown NOTIFICATION-TYPE

OBJECTS { sdfeIndex }

STATUS current

DESCRIPTION

"A deviceTrap trap is sent when the device exist changes."

::= { sdfeTrapList 2 }

sdfePcm30Alarm NOTIFICATION-TYPE

OBJECTS { sdfeIndex, sdfePcm30Index, sdfePcm30State }

STATUS current

DESCRIPTION

"A pcm30ChangeState_sdfe trap is sent when the

value of an instance pcm30State_sdfe changes."

::= { sdfeTrapList 3 }

sdfePcm30Normal NOTIFICATION-TYPE

OBJECTS { sdfeIndex, sdfePcm30Index, sdfePcm30State }

STATUS current

DESCRIPTION

"A pcm30ChangeState_sdfe trap is sent when the

value of an instance pcm30State_sdfe changes."

::= { sdfeTrapList 4 }

sdfeShdslAlarm NOTIFICATION-TYPE

OBJECTS { sdfeIndex, sdfeShdslIndex, sdfeShdslState }

STATUS current

DESCRIPTION

"A shdslChangeState_sdfe trap is sent when the

value of an 3instance shdslState_sdfe changes."

::= { sdfeTrapList 5 }

sdfeShdslNormal NOTIFICATION-TYPE

OBJECTS { sdfeIndex, sdfeShdslIndex, sdfeShdslState }

STATUS current

DESCRIPTION

"A shdslChangeState_sdfe trap is sent when the

value of an 3instance shdslState_sdfe changes."

::= { sdfeTrapList 6 }

-- SDFE PCM30 //!!!

--

sdfePcm30Table OBJECT-TYPE

SYNTAX SEQUENCE OF SDFEPCM30Entry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"Contains information."

::= { sdfe 3 }

sdfePcm30Entry OBJECT-TYPE

SYNTAX SDFEPCM30Entry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"An entry containing information about PCM30 ports."

INDEX { sdfePcm30Index }

::= { sdfePcm30Table 1 }

SDFEPCM30Entry ::= SEQUENCE {

```
        sdfePcm30Index                  INTEGER,
        sdfePcm30State                  DisplayString,
    sdfePcm30Status             INTEGER,
    sdfePcm30Code               INTEGER,
    sdfePcm30Transp             INTEGER,
    sdfePcm30Loop               INTEGER,
    sdfePcm30Nvp                INTEGER,
    sdfePcm30Pcs                INTEGER,
    sdfePcm30Psc                INTEGER,
    sdfePcm30E3                 INTEGER,
    sdfePcm30E5                 INTEGER,
    sdfePcm30Aus                INTEGER
        }

   sdfePcm30Index OBJECT-TYPE
    SYNTAX      INTEGER (1..64)
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
          "A unique value, greater than zero."
    ::= { sdfePcm30Entry 1 }

   sdfePcm30State OBJECT-TYPE
    SYNTAX      DisplayString
    MAX-ACCESS  accessible-for-notify
    STATUS      current
    DESCRIPTION
          "This variable indicates the Line Status of the interface."
    ::= { sdfePcm30Entry 2 }

   sdfePcm30Status  OBJECT-TYPE
    SYNTAX      INTEGER {
```

```
                                off(0),
                                on(1)
                                }
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "Status of PCM30 of 2SHDSL_2E1 (off/on)."
    ::= { sdfePcm30Entry 3 }

sdfePcm30Code  OBJECT-TYPE
    SYNTAX      INTEGER {
                                ami(0),
                                hdb3(1)
                                }
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "Current encoding of PCM30 of 2SHDSL_4E1 (ami/hdb3)."
    ::= { sdfePcm30Entry 4 }

  sdfePcm30Transp  OBJECT-TYPE
    SYNTAX      INTEGER {
                                off(0),
                                on(1)
                                }
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "State of transparency of PCM30 of 2SHDSL_4E1 (off/on)."
    ::= { sdfePcm30Entry 5 }

  sdfePcm30Loop  OBJECT-TYPE
```

SYNTAX INTEGER {

no(0),

near(1),

far(2),

dual(3)

}

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"Value of loop of PCM30 of 2SHDSL_4E1 (no/far/near/dual)."

::= { sdfePcm30Entry 6 }

sdfePcm30Nvp OBJECT-TYPE

SYNTAX INTEGER

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Value of nvp of PCM30 of 2SHDSL_4E1."

::= { sdfePcm30Entry 7 }

sdfePcm30Pcs OBJECT-TYPE

SYNTAX INTEGER

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Value of pcs of PCM30 of 2SHDSL_4E1."

::= { sdfePcm30Entry 8 }

sdfePcm30Psc OBJECT-TYPE

SYNTAX INTEGER

MAX-ACCESS read-only

STATUS current

```
    DESCRIPTION
            "Value of psc of PCM30 of 2SHDSL_4E1."
    ::= { sdfePcm30Entry 9 }

   sdfePcm30E3  OBJECT-TYPE
    SYNTAX      INTEGER
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
            "Value of e3 of PCM30 of 2SHDSL_4E1."
    ::= { sdfePcm30Entry 10 }

   sdfePcm30E5  OBJECT-TYPE
    SYNTAX      INTEGER
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
            "Value of e5 of PCM30 of 2SHDSL_4E1."
    ::= { sdfePcm30Entry 11 }

   sdfePcm30Aus  OBJECT-TYPE
    SYNTAX      INTEGER
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
            "Value of aus of PCM30 of 2SHDSL_4E1."
    ::= { sdfePcm30Entry 12 }


----------------
--              SDFE SHDSL //!!!
--
```

```
  sdfeShdslTable OBJECT-TYPE
   SYNTAX      SEQUENCE OF SDFESHDSLEntry
   MAX-ACCESS  not-accessible
   STATUS      current
   DESCRIPTION
        "Contains information."
   ::= { sdfe 4 }

  sdfeShdslEntry OBJECT-TYPE
   SYNTAX      SDFESHDSLEntry
   MAX-ACCESS  not-accessible
   STATUS      current
   DESCRIPTION
         "An entry containing information about SHDSL ports."
   INDEX   { sdfeShdslIndex }
   ::= { sdfeShdslTable 1 }

  SDFESHDSLEntry ::= SEQUENCE {
      sdfeShdslIndex                    INTEGER,
      sdfeShdslState                    DisplayString,
      sdfeShdslStatus                   INTEGER,
      sdfeShdslDescr                    DisplayString,
sdfeShdslRange                  INTEGER,
sdfeShdslMode                   INTEGER,
sdfeShdslChainFull              INTEGER,
sdfeShdslChainLen          INTEGER
      }

  sdfeShdslIndex OBJECT-TYPE
   SYNTAX      INTEGER (1..32)
   MAX-ACCESS  read-only
   STATUS      current
```

DESCRIPTION

"A unique value, greater than zero."

::= { sdfeShdslEntry 1 }

sdfeShdslState OBJECT-TYPE

SYNTAX DisplayString

MAX-ACCESS accessible-for-notify

STATUS current

DESCRIPTION

"This variable indicates the Line Status of theinterface.

It contains 'no input stream' (NVP), 'chain shdsl is broken' (ChainBroken)

The various bitpositions are:

0 NoAlarm

1 NVP

2 ChainBroken

"

::= { sdfeShdslEntry 2 }

sdfeShdslStatus OBJECT-TYPE

SYNTAX INTEGER {

off(0),

on(1)

}

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"Status of shdsl of 2SHDSL_4E1 (off/on)."

::= { sdfeShdslEntry 3 }

sdfeShdslDescr OBJECT-TYPE

SYNTAX DisplayString

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"Description of place where is shdsl of 2SHDSL_4E1."

::= { sdfeShdslEntry 4 }

sdfeShdslRange OBJECT-TYPE

SYNTAX INTEGER

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"Current range of shdsl of 2SHDSL_4E1."

::= { sdfeShdslEntry 5 }

sdfeShdslMode OBJECT-TYPE

SYNTAX INTEGER {

lt(0),

mt(1)

}

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"Current mode of shdsl of 2SHDSL_4E1 (lt/mt)."

::= { sdfeShdslEntry 6 }

sdfeShdslChainFull OBJECT-TYPE

SYNTAX INTEGER

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Indicates entirety of chain of shdsl of 2SHDSL_4E1."

::= { sdfeShdslEntry 7 }

```
  sdfeShdslChainLen  OBJECT-TYPE
   SYNTAX      INTEGER
   MAX-ACCESS  read-only
   STATUS      current
   DESCRIPTION
         "Length of chain of shdsl of 2SHDSL_4E1."
   ::= { sdfeShdslEntry 8 }


-------------------------------------
--            2SHDSL_4E1 SDFE SOCIF //!!!
--

  sdfeSocifTable OBJECT-TYPE
   SYNTAX      SEQUENCE OF SDFESOCIFEntry
   MAX-ACCESS  not-accessible
   STATUS      current
   DESCRIPTION
        "Contains information."
   ::= { sdfe 5 }

  sdfeSocifEntry OBJECT-TYPE
   SYNTAX      SDFESOCIFEntry
   MAX-ACCESS  not-accessible
   STATUS      current
   DESCRIPTION
         "An entry containing information about SHDSL'S SOCIF ports."
   INDEX   { sdfeSocifIndex }
   ::= { sdfeSocifTable 1 }

  SDFESOCIFEntry ::= SEQUENCE {
     sdfeSocifIndex                          INTEGER,
sdfeSocifStatus                      DisplayString
```

```
        }

    sdfeSocifIndex OBJECT-TYPE
     SYNTAX      INTEGER (1..512)
     MAX-ACCESS  read-only
     STATUS      current
     DESCRIPTION
           "A unique value, greater than zero."
     ::= { sdfeSocifEntry 1 }

    sdfeSocifStatus  OBJECT-TYPE
     SYNTAX      DisplayString
     MAX-ACCESS  read-only
     STATUS      current
     DESCRIPTION
           "Parameters status of socif of sdfe of 2SHDSL_4E1."
     ::= { sdfeSocifEntry 2 }

----------------
--              SDFE ETDM //!!!
--

    sdfeEtdmTable OBJECT-TYPE
     SYNTAX      SEQUENCE OF SDFEETDMEntry
     MAX-ACCESS  not-accessible
     STATUS      current
     DESCRIPTION
           "Contains information."
     ::= { sdfe 6 }

    sdfeEtdmEntry OBJECT-TYPE
     SYNTAX      SDFEETDMEntry
```

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"An entry containing information about SHDSL ports."

INDEX { sdfeIndex }

::= { sdfeEtdmTable 1 }

SDFEETDMEntry ::= SEQUENCE {

sdfeEtdmStatus INTEGER,

sdfeEtdmRange INTEGER

}

sdfeEtdmStatus OBJECT-TYPE

SYNTAX INTEGER {

off(0),

on(1)

}

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"Status of etdm of 2SHDSL_4E1 (off/on)."

::= { sdfeEtdmEntry 1 }

sdfeEtdmRange OBJECT-TYPE

SYNTAX INTEGER

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"Current range of etdm of 2SHDSL_4E1."

::= { sdfeEtdmEntry 2 }

```
--            SDFE PLL //!!!
--


  sdfePllTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF SDFEPLLEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
          "Contains information."
    ::= { sdfe 7 }


  sdfePllEntry OBJECT-TYPE
    SYNTAX      SDFEPLLEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
          "An entry containing information about PLL ports."
    INDEX   { sdfeIndex }
    ::= { sdfePllTable 1 }


  SDFEPLLEntry ::= SEQUENCE {
      sdfePllSource0                    INTEGER,
      sdfePllSource1                    INTEGER,
sdfePllSource2                  INTEGER,
sdfePllSource3                  INTEGER,
sdfePllCntSrc                   INTEGER,
sdfePllCurPllSrc          DisplayString,
sdfePllCurPhase           INTEGER,
sdfePllFlowPhase          INTEGER,
sdfePllCodeDac                  INTEGER,
sdfePllStepPll                  INTEGER
      }
```

```
sdfePllSource0  OBJECT-TYPE
 SYNTAX      INTEGER {
                                off(0),
                                shdsl-0(1),
                                shdsl-1(2),
                                pcm30-0(3),
                                pcm30-1(4),
                                pcm30-2(5),
                                pcm30-3(6)
                                }
 MAX-ACCESS  read-write
 STATUS      current
 DESCRIPTION
       "Source0 of pll of 2SHDSL_4E1 (off/shdsl_0/shdsl_1/pcm30_0/pcm30_1/pcm30_2/pcm30_3)."
::= { sdfePllEntry 1 }

sdfePllSource1  OBJECT-TYPE
 SYNTAX      INTEGER {
                                off(0),
                                shdsl-0(1),
                                shdsl-1(2),
                                pcm30-0(3),
                                pcm30-1(4),
                                pcm30-2(5),
                                pcm30-3(6)
                                }
 MAX-ACCESS  read-write
 STATUS      current
 DESCRIPTION
       "Source1 of pll of 2SHDSL_4E1 (off/shdsl_0/shdsl_1/pcm30_0/pcm30_1/pcm30_2/pcm30_3)."
::= { sdfePllEntry 2 }
```

```
sdfePllSource2  OBJECT-TYPE
 SYNTAX      INTEGER {
                                off(0),
                                shdsl-0(1),
                                shdsl-1(2),
                                pcm30-0(3),
                                pcm30-1(4),
                                pcm30-2(5),
                                pcm30-3(6)
                                }
 MAX-ACCESS  read-write
 STATUS      current
 DESCRIPTION
       "Source2 of pll of 2SHDSL_4E1 (off/shdsl_0/shdsl_1/pcm30_0/pcm30_1/pcm30_2/pcm30_3)."
::= { sdfePllEntry 3 }

sdfePllSource3  OBJECT-TYPE
 SYNTAX      INTEGER {
                                off(0),
                                shdsl-0(1),
                                shdsl-1(2),
                                pcm30-0(3),
                                pcm30-1(4),
                                pcm30-2(5),
                                pcm30-3(6)
                                }
 MAX-ACCESS  read-write
 STATUS      current
 DESCRIPTION
       "Source3 of pll of 2SHDSL_4E1 (off/shdsl_0/shdsl_1/pcm30_0/pcm30_1/pcm30_2/pcm30_3)."
::= { sdfePllEntry 4 }
```

```
sdfePllCntSrc  OBJECT-TYPE
 SYNTAX      INTEGER
 MAX-ACCESS  read-only
 STATUS      current
 DESCRIPTION
       "CNT_SRC of pll of 2SHDSL_4E1."
 ::= { sdfePllEntry 5 }

sdfePllCurPllSrc  OBJECT-TYPE
 SYNTAX      DisplayString
 MAX-ACCESS  read-only
 STATUS      current
 DESCRIPTION
       "Current source of pll of 2SHDSL_4E1."
 ::= { sdfePllEntry 6 }

sdfePllCurPhase  OBJECT-TYPE
 SYNTAX      INTEGER
 MAX-ACCESS  read-only
 STATUS      current
 DESCRIPTION
       "Current phase of pll of 2SHDSL_4E1."
 ::= { sdfePllEntry 7 }

sdfePllFlowPhase  OBJECT-TYPE
 SYNTAX      INTEGER
 MAX-ACCESS  read-only
 STATUS      current
 DESCRIPTION
       "Flow phase of pll of 2SHDSL_4E1."
 ::= { sdfePllEntry 8 }
```

```
  sdfePllCodeDac  OBJECT-TYPE
   SYNTAX      INTEGER
   MAX-ACCESS  read-only
   STATUS      current
   DESCRIPTION
         "CODE_DAC of pll of 2SHDSL_4E1."
   ::= { sdfePllEntry 9 }

  sdfePllStepPll  OBJECT-TYPE
   SYNTAX      INTEGER
   MAX-ACCESS  read-only
   STATUS      current
   DESCRIPTION
         "Step of pll of 2SHDSL_4E1."
   ::= { sdfePllEntry 10 }




--------------------------------------------------------------------------------
--             UMP //!!!
--


----------------
--             UMP PARAM //!!!
--


  umpParamTable OBJECT-TYPE
   SYNTAX      SEQUENCE OF UMPPARAMEntry
   MAX-ACCESS  not-accessible
   STATUS      current
   DESCRIPTION
```

```
        "Contains information."
 ::= { ump 1 }

umpParamEntry OBJECT-TYPE
 SYNTAX      UMPPARAMEntry
 MAX-ACCESS  not-accessible
 STATUS      current
 DESCRIPTION
        "An entry containing information about PARAM ports."
 INDEX   { umpIndex }
 ::= { umpParamTable 1 }

UMPPARAMEntry ::= SEQUENCE {
    umpIndex       INTEGER,
    umpFlowState           DisplayString,
    umpExist       INTEGER,
    umpNvp                 INTEGER,
    umpPcs                 INTEGER,
    umpAus                 INTEGER
    }

umpIndex OBJECT-TYPE
 SYNTAX      INTEGER (1..16)
 MAX-ACCESS  read-only
 STATUS      current
 DESCRIPTION
        "A unique value, greater than zero."
 ::= { umpParamEntry 1 }

umpFlowState OBJECT-TYPE
 SYNTAX      DisplayString
 MAX-ACCESS  accessible-for-notify
```

STATUS current

DESCRIPTION

"This variable indicates the Line Status of theinterface.

It contains 'no input stream' (NVP), 'emergency indication

signal' (SIA), 'cycle synchronization lost' (PCS), E-3, E-5,

'far end emergency' (AUS), '(far end) supercycle synchronization

lost' (PSCS(U)). The ifPCM15CurrentState is a bit map represented

as a sum, therefore, it can represent multiple failures (alarms).

ifPCM15NoAlarm must be set if and only if no other flag is set.

The various bitpositions are:

0 NoAlarm

1 NVP

2 SIA

4 PCS

8 E-3

16 E-5

32 AUS

64 PSCS

128 PSCSU

"

::= { umpParamEntry 2 }

umpExist OBJECT-TYPE

SYNTAX INTEGER {

not-exist(0),

exist(1)

}

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"This variable indicates the Exist Board."

::= { umpParamEntry 3 }

```
  umpNvp OBJECT-TYPE
   SYNTAX      INTEGER
   MAX-ACCESS  read-only
   STATUS      current
   DESCRIPTION
         "This variable indicates the NVP."
   ::= { umpParamEntry 4 }

  umpPcs OBJECT-TYPE
   SYNTAX      INTEGER
   MAX-ACCESS  read-only
   STATUS      current
   DESCRIPTION
         "This variable indicates the PCS."
   ::= { umpParamEntry 5 }

  umpAus OBJECT-TYPE
   SYNTAX      INTEGER
   MAX-ACCESS  read-only
   STATUS      current
   DESCRIPTION
         "This variable indicates the AUS."
   ::= { umpParamEntry 6 }

----------------------
--            UMP TRAPS //!!!
--
  umpTraps                    OBJECT IDENTIFIER ::= { ump 2 }
  umpTrapList                 OBJECT IDENTIFIER ::= { umpTraps 1 }

 umpBoardUp NOTIFICATION-TYPE
```

```
    OBJECTS { umpIndex }
    STATUS      current
    DESCRIPTION
            "A deviceTrap trap is sent when the device exist changes."
    ::= { umpTrapList 1 }

umpBoardDown NOTIFICATION-TYPE
    OBJECTS { umpIndex }
    STATUS      current
    DESCRIPTION
            "A deviceTrap trap is sent when the device exist changes."
    ::= { umpTrapList 2 }

 umpFlowAlarm NOTIFICATION-TYPE
    OBJECTS { umpIndex, umpFlowState }
    STATUS      current
    DESCRIPTION
            "A flowChangeState_ump trap is sent when the
            value of an instance flowState_ump changes."
    ::= { umpTrapList 3 }

 umpFlowNormal NOTIFICATION-TYPE
    OBJECTS { umpIndex, umpFlowState }
    STATUS      current
    DESCRIPTION
            "A flowChangeState_ump trap is sent when the
            value of an instance flowState_ump changes."
    ::= { umpTrapList 4 }

--------------------------------------------------------------------------
--              KNS //!!!
--
```

```
----------------
--            KNS PARAM //!!!
--


  knsParamTable OBJECT-TYPE
   SYNTAX      SEQUENCE OF KNSPARAMEntry
   MAX-ACCESS  not-accessible
   STATUS      current
   DESCRIPTION
        "Contains information."
   ::= { kns 1 }


  knsParamEntry OBJECT-TYPE
   SYNTAX      KNSPARAMEntry
   MAX-ACCESS  not-accessible
   STATUS      current
   DESCRIPTION
         "An entry containing information about PARAM ports."
   INDEX   { knsIndex }
   ::= { knsParamTable 1 }


  KNSPARAMEntry ::= SEQUENCE {
      knsIndex        INTEGER,
      knsExist        INTEGER,
knsStatus INTEGER,
knsT                  DisplayString,
knsState  DisplayString
      }


  knsIndex OBJECT-TYPE
   SYNTAX      INTEGER (1..8)
```

```
 MAX-ACCESS  read-only
 STATUS      current
 DESCRIPTION
       "A unique value, greater than zero."
 ::= { knsParamEntry 1 }

knsExist OBJECT-TYPE
 SYNTAX      INTEGER {
                                not-exist(0),
                                exist(1)
                                }
 MAX-ACCESS  read-only
 STATUS      current
 DESCRIPTION
       "This variable indicates the Exist Board."
 ::= { knsParamEntry 2 }

knsStatus OBJECT-TYPE
 SYNTAX      INTEGER {
                                off(0),
                                on(1)
                                }
 MAX-ACCESS  read-write
 STATUS      current
 DESCRIPTION
       "This variable indicates the state of Board (off/on)."
 ::= { knsParamEntry 3 }

knsT OBJECT-TYPE
 SYNTAX      DisplayString
 MAX-ACCESS  read-only
 STATUS      current
```

```
  DESCRIPTION
        "This variable indicates the Temperature."
  ::= { knsParamEntry 4 }

 knsState OBJECT-TYPE
  SYNTAX      DisplayString
  MAX-ACCESS  accessible-for-notify
  STATUS      current
  DESCRIPTION
        "This variable indicates the Line Status of the interface.
        ALARM is sent when any of alarm is occurred.
        ALARM SUPPLY is sent when alarm supply is occurred.
        LIMITATION is sent when limitation is occured.
        OVERHEAT is sent when overheat is occured."
  ::= { knsParamEntry 5 }



------------------------
--            KNS TRAPS //!!!
--
  knsTraps                    OBJECT IDENTIFIER ::= { kns 2 }
  knsTrapList                 OBJECT IDENTIFIER ::= { knsTraps 1 }

 knsBoardUp NOTIFICATION-TYPE
  OBJECTS { knsIndex }
  STATUS      current
  DESCRIPTION
        "A deviceTrap trap is sent when the device exist changes."
  ::= { knsTrapList 1 }

 knsBoardDown NOTIFICATION-TYPE
  OBJECTS { knsIndex }
```

```
    STATUS      current
    DESCRIPTION
            "A deviceTrap trap is sent when the device exist changes."
    ::= { knsTrapList 2 }

  knsAlarm NOTIFICATION-TYPE
    OBJECTS { knsIndex, knsState}
    STATUS      current
    DESCRIPTION
            "A knsAlarm trap is sent when the
            value of an instance knsState changes alarm."
    ::= { knsTrapList 3 }

  knsNormal NOTIFICATION-TYPE
    OBJECTS { knsIndex, knsState }
    STATUS      current
    DESCRIPTION
            "A knsNormal trap is sent when the
            value of an instance knsState changes normal."
    ::= { knsTrapList 4 }


--------------------------------------------------------------------------
--              IDP240 //!!!
--


----------------
--              IDP240 PARAM //!!!
--


    idp240ParamTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF IDP240PARAMEntry
    MAX-ACCESS  not-accessible
```

```
    STATUS      current
    DESCRIPTION
        "Contains information."
    ::= { idp240 1 }

  idp240ParamEntry OBJECT-TYPE
    SYNTAX      IDP240PARAMEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "An entry containing information about PARAM ports."
    INDEX   { idp240Index }
    ::= { idp240ParamTable 1 }

   IDP240PARAMEntry ::= SEQUENCE {
       idp240Index             INTEGER,
       idp240Exist             INTEGER,
idp240Status                   INTEGER,
       idp240T                 DisplayString,
idp240TCrit                    DisplayString,
       idp240UOut              DisplayString,
idp240UNom                     DisplayString,
       idp240IOut              DisplayString,
idp240INom                     DisplayString,
       idp240Iutechki                  DisplayString,
idp240Jumpers                  INTEGER,
idp240JumpersSet  INTEGER,
idp240INomSet                  DisplayString,
       idp240UkzSet                    DisplayString,
idp240IxxSet                   DisplayString,
       idp240UMaxSet                   DisplayString,
idp240IMaxSet                  DisplayString,
```

```
idp240State                    DisplayString
      }

  idp240Index OBJECT-TYPE
   SYNTAX     INTEGER (1..8)
   MAX-ACCESS  read-only
   STATUS     current
   DESCRIPTION
        "A unique value, greater than zero."
   ::= { idp240ParamEntry 1 }

  idp240Exist OBJECT-TYPE
   SYNTAX     INTEGER {
                                not-exist(0),
                                exist(1)
                                }
   MAX-ACCESS  read-only
   STATUS     current
   DESCRIPTION
        "This variable indicates the Exist Board."
   ::= { idp240ParamEntry 2 }

 idp240Status OBJECT-TYPE
   SYNTAX     INTEGER {
            off(0),
            on(1)
}
   MAX-ACCESS  read-write
   STATUS     current
   DESCRIPTION
        "A textual string containing interface status."
   ::= { idp240ParamEntry 3 }
```

```
idp240T OBJECT-TYPE
  SYNTAX      DisplayString
  MAX-ACCESS  read-only
  STATUS      current
  DESCRIPTION
        "Temperature line current IDP240s."
  ::= { idp240ParamEntry 4 }

idp240TCrit OBJECT-TYPE
  SYNTAX      DisplayString
  MAX-ACCESS  read-write
  STATUS      current
  DESCRIPTION
        "Critcal temperature line current IDP240s."
  ::= { idp240ParamEntry 5 }

idp240UOut OBJECT-TYPE
  SYNTAX      DisplayString
  MAX-ACCESS  read-only
  STATUS      current
  DESCRIPTION
        "U_OUT line current IDP240s."
  ::= { idp240ParamEntry 6 }

idp240UNom OBJECT-TYPE
  SYNTAX      DisplayString
  MAX-ACCESS  read-only
  STATUS      current
  DESCRIPTION
        "U_NOM line current IDP240s."
  ::= { idp240ParamEntry 7 }
```

```
idp240IOut OBJECT-TYPE
  SYNTAX      DisplayString
  MAX-ACCESS  read-only
  STATUS      current
  DESCRIPTION
        "I_OUT line current IDP240s."
  ::= { idp240ParamEntry 8 }

idp240INom OBJECT-TYPE
  SYNTAX      DisplayString
  MAX-ACCESS  read-only
  STATUS      current
  DESCRIPTION
        "I_NOM line current IDP240s."
  ::= { idp240ParamEntry 9 }

idp240Iutechki OBJECT-TYPE
  SYNTAX      DisplayString
  MAX-ACCESS  read-only
  STATUS      current
  DESCRIPTION
        "I-ullage line current IDP240s."
  ::= { idp240ParamEntry 10 }

idp240Jumpers OBJECT-TYPE
  SYNTAX      INTEGER
  MAX-ACCESS  read-only
  STATUS      current
  DESCRIPTION
        "Jumpers line current IDP240s."
  ::= { idp240ParamEntry 11 }
```

```
idp240JumpersSet OBJECT-TYPE
  SYNTAX      INTEGER
  MAX-ACCESS  read-write
  STATUS      current
  DESCRIPTION
        "Jumpers-set line current IDP240s."
  ::= { idp240ParamEntry 12 }

idp240INomSet OBJECT-TYPE
  SYNTAX      DisplayString
  MAX-ACCESS  read-write
  STATUS      current
  DESCRIPTION
        "I_NOM_SET line current IDP240s."
  ::= { idp240ParamEntry 13 }

idp240UkzSet OBJECT-TYPE
  SYNTAX      DisplayString
  MAX-ACCESS  read-write
  STATUS      current
  DESCRIPTION
        "U_KZ_SET line current IDP240s."
  ::= { idp240ParamEntry 14 }

idp240IxxSet OBJECT-TYPE
  SYNTAX      DisplayString
  MAX-ACCESS  read-write
  STATUS      current
  DESCRIPTION
        "I_XX_SET line current IDP240s."
  ::= { idp240ParamEntry 15 }
```

```
idp240UMaxSet OBJECT-TYPE
  SYNTAX      DisplayString
  MAX-ACCESS  read-write
  STATUS      current
  DESCRIPTION
        "I_MAX_SET line current IDP240s."
  ::= { idp240ParamEntry 16 }

idp240IMaxSet OBJECT-TYPE
  SYNTAX      DisplayString
  MAX-ACCESS  read-write
  STATUS      current
  DESCRIPTION
        "I_MAX_SET line current IDP240s."
  ::= { idp240ParamEntry 17 }

idp240State OBJECT-TYPE
  SYNTAX      DisplayString
  MAX-ACCESS  accessible-for-notify
  STATUS      current
  DESCRIPTION
        "This variable indicates the Line Status of the interface.
        ALARM is sent when any of alarm is occurred.
        SHORT-CIRCUIT is sent when short-circuit is occurred.
        LIMITATION is sent when limitation is occured.
        LINE-BREAK is sent when line-break is occured.
        LEAKAGE is sent when I-LEAKAGE is occured.
        REJECT is sent when REJECT is occured.
  SHORT-CIRCUIT-I-MAX is sent when short-circuit and limitation I-max is occured."
  ::= { idp240ParamEntry 18 }
```

```
----------------------

--            IDP240 TRAPS //!!!

--

  idp240Traps                   OBJECT IDENTIFIER ::= { idp240 2 }

  idp240TrapList                OBJECT IDENTIFIER ::= { idp240Traps 1 }


 idp240BoardUp NOTIFICATION-TYPE

   OBJECTS { idp240Index }

   STATUS      current

   DESCRIPTION

         "A deviceTrap trap is sent when the device exist changes."

   ::= { idp240TrapList 1 }


 idp240BoardDown NOTIFICATION-TYPE

   OBJECTS { idp240Index }

   STATUS      current

   DESCRIPTION

         "A deviceTrap trap is sent when the device exist changes."

   ::= { idp240TrapList 2 }


 idp240Alarm NOTIFICATION-TYPE

   OBJECTS { idp240Index, idp240State }

   STATUS      current

   DESCRIPTION

         "A idp240Alarm trap is sent when the

         value of an instance idp240State changes alarm."

   ::= { idp240TrapList 3 }


 idp240Normal NOTIFICATION-TYPE

   OBJECTS { idp240Index, idp240State }

   STATUS      current

   DESCRIPTION
```

```
        "A idp240Normal trap is sent when the
        value of an instance idp240State changes normal."
    ::= { idp240TrapList 4 }

--------------------------------------------------------------------------
--          IDP350 //!!!
--

----------------
--          IDP350 PARAM //!!!
--

  idp350ParamTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF IDP350PARAMEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "Contains information."
    ::= { idp350 1 }

  idp350ParamEntry OBJECT-TYPE
    SYNTAX      IDP350PARAMEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "An entry containing information about PARAM ports."
    INDEX   { idp350Index }
    ::= { idp350ParamTable 1 }

  IDP350PARAMEntry ::= SEQUENCE {
      idp350Index               INTEGER,
      idp350Exist               INTEGER,
```

```
        idp350Status            INTEGER,
        idp350T                 DisplayString,
        idp350UOut              DisplayString,
        idp350IOut              DisplayString,
idp350INom                      DisplayString,
        idp350Iutechki                  DisplayString,
idp350Jumpers                   INTEGER,
idp350JumpersSet  INTEGER,
idp350UNomSet                   DisplayString,
idp350INomSet                   DisplayString,
        idp350UkzSet                    DisplayString,
idp350IxxSet                    DisplayString,
        idp350UMaxSet                   DisplayString,
idp350IMaxSet                   DisplayString,
idp350IUte4Set                  DisplayString,
idp350State                     DisplayString
        }

   idp350Index OBJECT-TYPE
    SYNTAX      INTEGER (1..8)
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
          "A unique value, greater than zero."
    ::= { idp350ParamEntry 1 }

   idp350Exist OBJECT-TYPE
    SYNTAX      INTEGER {
                                not-exist(0),
                                exist(1)
                                }
    MAX-ACCESS  read-only
```

```
    STATUS      current
    DESCRIPTION
            "This variable indicates the Exist Board."
    ::= { idp350ParamEntry 2 }

    idp350Status OBJECT-TYPE
    SYNTAX      INTEGER {
                off(0),
                on(1)
}
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
            "A textual string containing interface status."
    ::= { idp350ParamEntry 3 }

   idp350T OBJECT-TYPE
    SYNTAX      DisplayString
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
            "Temperature line current IDP350s."
    ::= { idp350ParamEntry 4 }

   idp350UOut OBJECT-TYPE
    SYNTAX      DisplayString
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
            "U_OUT line current IDP350s."
    ::= { idp350ParamEntry 5 }
```

```
idp350IOut OBJECT-TYPE
  SYNTAX      DisplayString
  MAX-ACCESS  read-only
  STATUS      current
  DESCRIPTION
        "I_OUT line current IDP350s."
  ::= { idp350ParamEntry 6 }

idp350INom OBJECT-TYPE
  SYNTAX      DisplayString
  MAX-ACCESS  read-only
  STATUS      current
  DESCRIPTION
        "I_NOM line current IDP350s."
  ::= { idp350ParamEntry 7 }

idp350Iutechki OBJECT-TYPE
  SYNTAX      DisplayString
  MAX-ACCESS  read-only
  STATUS      current
  DESCRIPTION
        "I-ullage line current IDP350s."
  ::= { idp350ParamEntry 8 }

idp350Jumpers OBJECT-TYPE
  SYNTAX      INTEGER
  MAX-ACCESS  read-only
  STATUS      current
  DESCRIPTION
        "Jumpers line current IDP350s."
  ::= { idp350ParamEntry 9 }
```

```
idp350JumpersSet OBJECT-TYPE
  SYNTAX      INTEGER
  MAX-ACCESS  read-write
  STATUS      current
  DESCRIPTION
        "Jumpers-set line current IDP350s."
  ::= { idp350ParamEntry 10 }

idp350UNomSet OBJECT-TYPE
  SYNTAX      DisplayString
  MAX-ACCESS  read-write
  STATUS      current
  DESCRIPTION
        "I_NOM_SET line current IDP350s."
  ::= { idp350ParamEntry 11 }

idp350INomSet OBJECT-TYPE
  SYNTAX      DisplayString
  MAX-ACCESS  read-write
  STATUS      current
  DESCRIPTION
        "I_NOM_SET line current IDP350s."
  ::= { idp350ParamEntry 12 }

idp350UkzSet OBJECT-TYPE
  SYNTAX      DisplayString
  MAX-ACCESS  read-write
  STATUS      current
  DESCRIPTION
        "I_KZ_SET line current IDP350s."
  ::= { idp350ParamEntry 13 }
```

```
idp350IxxSet OBJECT-TYPE
  SYNTAX      DisplayString
  MAX-ACCESS  read-write
  STATUS      current
  DESCRIPTION
        "I_XX_SET line current IDP350s."
  ::= { idp350ParamEntry 14 }

idp350UMaxSet OBJECT-TYPE
  SYNTAX      DisplayString
  MAX-ACCESS  read-write
  STATUS      current
  DESCRIPTION
        "I_MAX_SET line current IDP350s."
  ::= { idp350ParamEntry 15 }

idp350IMaxSet OBJECT-TYPE
  SYNTAX      DisplayString
  MAX-ACCESS  read-write
  STATUS      current
  DESCRIPTION
        "I_MAX_SET line current IDP350s."
  ::= { idp350ParamEntry 16 }

idp350IUte4Set OBJECT-TYPE
  SYNTAX      DisplayString
  MAX-ACCESS  read-write
  STATUS      current
  DESCRIPTION
        "I_UTE4_SET line current IDP350s."
  ::= { idp350ParamEntry 17 }
```

```
 idp350State OBJECT-TYPE
   SYNTAX      DisplayString
   MAX-ACCESS  accessible-for-notify
   STATUS      current
   DESCRIPTION
         "This variable indicates the Line Status of the interface.
         ALARM is sent when any of alarm is occurred.
         SHORT-CIRCUIT is sent when short-circuit is occurred.
         LIMITATION is sent when limitation is occured.
         LINE-BREAK is sent when line-break is occured.
         LEAKAGE is sent when I-LEAKAGE is occured.
         REJECT1 is sent when REJECT is occured.
   REJECT2 is sent when REJECT is occured."
   ::= { idp350ParamEntry 18 }



----------------------
--            IDP350 TRAPS //!!!
--
  idp350Traps                              OBJECT IDENTIFIER ::= { idp350 2 }
  idp350TrapList                 OBJECT IDENTIFIER ::= { idp350Traps 1 }

 idp350BoardUp NOTIFICATION-TYPE
   OBJECTS { idp350Index }
   STATUS      current
   DESCRIPTION
         "A deviceTrap trap is sent when the device exist changes."
   ::= { idp350TrapList 1 }

 idp350BoardDown NOTIFICATION-TYPE
   OBJECTS { idp350Index }
   STATUS      current
```

```
  DESCRIPTION
        "A deviceTrap trap is sent when the device exist changes."
  ::= { idp350TrapList 2 }

 idp350Alarm NOTIFICATION-TYPE
  OBJECTS { idp350Index, idp350State }
  STATUS      current
  DESCRIPTION
        "A idp350Alarm trap is sent when the
        value of an instance idp350State changes alarm."
  ::= { idp350TrapList 3 }

 idp350Normal NOTIFICATION-TYPE
  OBJECTS { idp350Index, idp350State }
  STATUS      current
  DESCRIPTION
        "A idp350Normal trap is sent when the
        value of an instance idp350State changes normal."
  ::= { idp350TrapList 4 }



--**********************
--       AKB
--**********************

  akbTable OBJECT-TYPE
  SYNTAX      SEQUENCE OF AKBEntry
  MAX-ACCESS  not-accessible
  STATUS      current
  DESCRIPTION
        "A list of AKB interface entries.  The number of entries is
        given by the value of ifNumber."
```

::= { akb 1 }

akbEntry OBJECT-TYPE

SYNTAX AKBEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"An entry containing management information applicable to a

particular AKB interface."

INDEX { akbIndex }

::= { akbTable 1 }

AKBEntry ::= SEQUENCE {

akbIndex InterfaceIndex,

akbExist INTEGER,

akbStatus INTEGER,

akbU1 DisplayString,

akbU2 DisplayString,

akbU3 DisplayString,

akbU4 DisplayString,

akbU5 DisplayString,

akbI DisplayString,

akbT DisplayString,

akbUCharg DisplayString,

akbTKoeff DisplayString,

akbIMax DisplayString

}

akbIndex OBJECT-TYPE

SYNTAX InterfaceIndex

MAX-ACCESS read-only

STATUS current

```
  DESCRIPTION
        "A unique value, greater than zero, for each interface.  It
        is recommended that values are assigned contiguously
        starting from 1.  The value for each interface sub-layer
        must remain constant at least from one re-initialization of
        the entity's network management system to the next re-
        initialization."
  ::= { akbEntry 1 }

 akbExist OBJECT-TYPE
  SYNTAX      INTEGER {
                                not-exist(0),
                                exist(1)
                                }
  MAX-ACCESS  read-only
  STATUS      current
  DESCRIPTION
        "This variable indicates the Exist AKB."
  ::= { akbEntry 2 }

 akbStatus OBJECT-TYPE
  SYNTAX      INTEGER {
            off(0),
            on(1)
}
  MAX-ACCESS  read-write
  STATUS      current
  DESCRIPTION
        "A textual string containing interface status."
  ::= { akbEntry 3 }

 akbU1 OBJECT-TYPE
```

```
  SYNTAX      DisplayString
  MAX-ACCESS  read-only
  STATUS      current
  DESCRIPTION
        "U1 line current AKBs."
  ::= { akbEntry 4 }

akbU2 OBJECT-TYPE
  SYNTAX      DisplayString
  MAX-ACCESS  read-only
  STATUS      current
  DESCRIPTION
        "U2 line current AKBs."
  ::= { akbEntry 5 }

akbU3 OBJECT-TYPE
  SYNTAX      DisplayString
  MAX-ACCESS  read-only
  STATUS      current
  DESCRIPTION
        "U3 line current AKBs."
  ::= { akbEntry 6 }

akbU4 OBJECT-TYPE
  SYNTAX      DisplayString
  MAX-ACCESS  read-only
  STATUS      current
  DESCRIPTION
        "U4 line current AKBs."
  ::= { akbEntry 7 }

akbU5 OBJECT-TYPE
```

```
  SYNTAX      DisplayString
  MAX-ACCESS  read-only
  STATUS      current
  DESCRIPTION
        "U5 line current AKBs."
  ::= { akbEntry 8 }

akbI OBJECT-TYPE
  SYNTAX      DisplayString
  MAX-ACCESS  read-only
  STATUS      current
  DESCRIPTION
        "I line current AKBs."
  ::= { akbEntry 9 }

akbT OBJECT-TYPE
  SYNTAX      DisplayString
  MAX-ACCESS  read-only
  STATUS      current
  DESCRIPTION
        "T line current AKBs."
  ::= { akbEntry 10 }

akbUCharg OBJECT-TYPE
  SYNTAX      DisplayString
  MAX-ACCESS  read-write
  STATUS      current
  DESCRIPTION
        "U_CHARG line current AKBs."
  ::= { akbEntry 11 }

akbTKoeff OBJECT-TYPE
```

```
    SYNTAX      DisplayString
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
            "T_KOEFF line current AKBs."
    ::= { akbEntry 12 }

  akbIMax OBJECT-TYPE
    SYNTAX      DisplayString
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
            "I_MAX line current AKBs."
    ::= { akbEntry 13 }

----------------------
--              AKB TRAPS
--
    akbTraps                    OBJECT IDENTIFIER ::= { akb 2 }
    akbTrapList                 OBJECT IDENTIFIER ::= { akbTraps 1 }

  akbUp NOTIFICATION-TYPE
    OBJECTS { akbIndex }
    STATUS      current
    DESCRIPTION
            "A deviceTrap trap is sent when the device exist changes."
    ::= { akbTrapList 1 }

  akbDown NOTIFICATION-TYPE
    OBJECTS { akbIndex }
    STATUS      current
    DESCRIPTION
```

```
        "A deviceTrap trap is sent when the device exist changes."
    ::= { akbTrapList 2 }



--**********************
--        LOAD
--**********************


   loadParams                    OBJECT IDENTIFIER ::= { load 1 }
   loadParamList     OBJECT IDENTIFIER ::= { loadParams 1 }

  loadStatus OBJECT-TYPE
SYNTAX      INTEGER
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "A counter of elmetr."
::= { loadParamList 1 }

  loadI   OBJECT-TYPE
SYNTAX      INTEGER
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "A coefficient of elmetr."
::= { loadParamList 2 }

  loadAlarm1 OBJECT-TYPE
SYNTAX      INTEGER {
                    normal(0),
```

```
                    alarm(1)
    }
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Alarm of first load."
::= { loadParamList 3 }

  loadAlarm2   OBJECT-TYPE
SYNTAX      INTEGER {
                    normal(0),
                    alarm(1)
    }
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Alarm of second load."
::= { loadParamList 4 }

  loadAlarm3 OBJECT-TYPE
SYNTAX      INTEGER{
                    normal(0),
                    alarm(1)
    }
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Alarm of third load."
::= { loadParamList 5 }
```

```
--          LOAD TRAPS //!!!
--


  loadTraps                   OBJECT IDENTIFIER ::= { load 2 }
  loadTrapList                OBJECT IDENTIFIER ::= { loadTraps 1 }

 loadAlarm NOTIFICATION-TYPE
   OBJECTS { loadAlarm1, loadAlarm2, loadAlarm3 }
   STATUS      current
   DESCRIPTION
         "A loadTrap trap is sent when the
         value of an instance loadAlarm1 or loadAlarm2 or loadAlarm3 alarm."
   ::= { loadTrapList 1 }

 loadNormal NOTIFICATION-TYPE
   OBJECTS { loadAlarm1, loadAlarm2, loadAlarm3 }
   STATUS      current
   DESCRIPTION
         "A loadTrap trap is sent when the
         value of an instance loadAlarm1 or loadAlarm2 or loadAlarm3 normal."
   ::= { loadTrapList 2 }



--------------------------------------------------------------------------
--          ElMetr //!!!
--


----------------------
--          ElMetr PARAMS //!!!
--


  elmetrParams                OBJECT IDENTIFIER ::= { elmetr 1 }
```

```
  elmetrParamList  OBJECT IDENTIFIER ::= { elmetrParams 1 }


 elmetrCounter OBJECT-TYPE
SYNTAX      INTEGER
   MAX-ACCESS  read-write
   STATUS      current
   DESCRIPTION
         "A counter of elmetr."
::= { elmetrParamList 1 }

 elmetrCoeff   OBJECT-TYPE
SYNTAX      INTEGER
   MAX-ACCESS  read-write
   STATUS      current
   DESCRIPTION
         "A coefficient of elmetr."
::= { elmetrParamList 2 }



--------------------------------------------------------------------------------
--            IDP350v13 //!!!
--


----------------
--            IDP350v13 PARAM //!!!
--

  idp350v13ParamTable OBJECT-TYPE
   SYNTAX      SEQUENCE OF IDP350V13PARAMEntry
   MAX-ACCESS  not-accessible
   STATUS      current
   DESCRIPTION
```

```
        "Contains information."
 ::= { idp350v13 1 }

idp350v13ParamEntry OBJECT-TYPE
 SYNTAX      IDP350V13PARAMEntry
 MAX-ACCESS  not-accessible
 STATUS      current
 DESCRIPTION
        "An entry containing information about PARAM ports."
 INDEX   { idp350v13Index }
 ::= { idp350v13ParamTable 1 }

IDP350V13PARAMEntry ::= SEQUENCE {
    idp350v13Index                  INTEGER,
    idp350v13Exist                  INTEGER,
    idp350v13State                  DisplayString,
    idp350v13Status           INTEGER,
    idp350v13UL            INTEGER,
    idp350v13UH            INTEGER,
    idp350v13T             DisplayString,
    idp350v13UOut          DisplayString,
    idp350v13IOut          DisplayString,
    idp350v13Iutechki      DisplayString,
    idp350v13Jumpers       INTEGER,
    idp350v13JumpersSet    INTEGER,
    idp350v13UNomSet       DisplayString,
    idp350v13INomSet       DisplayString,
    idp350v13IxxSet        DisplayString,
    idp350v13RUte4Set      DisplayString

}
```

```
idp350v13Index OBJECT-TYPE
 SYNTAX      INTEGER (1..8)
 MAX-ACCESS  read-only
 STATUS      current
 DESCRIPTION
       "A unique value, greater than zero."
 ::= { idp350v13ParamEntry 1 }

idp350v13Exist OBJECT-TYPE
 SYNTAX      INTEGER {
                                 not-exist(0),
                                 exist(1)
                                 }
 MAX-ACCESS  read-only
 STATUS      current
 DESCRIPTION
       "This variable indicates the Exist Board."
 ::= { idp350v13ParamEntry 2 }

idp350v13State OBJECT-TYPE
 SYNTAX      DisplayString
 MAX-ACCESS  accessible-for-notify
 STATUS      current
 DESCRIPTION
       "This variable indicates the Line Status of the interface.
       ALARM is sent when any of alarm is occurred.
       SHORT-CIRCUIT is sent when short-circuit is occurred.
       LIMITATION is sent when limitation is occured.
       LINE-BREAK is sent when line-break is occured.
       LEAKAGE is sent when I-LEAKAGE is occured.
       REJECT1 is sent when REJECT is occured.
       REJECT2 is sent when REJECT is occured."
```

::= { idp350v13ParamEntry 3 }

idp350v13Status OBJECT-TYPE

SYNTAX INTEGER {

off(0),

on(1)

}

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"A textual string containing interface status."

::= { idp350v13ParamEntry 4 }

idp350v13UL OBJECT-TYPE

SYNTAX INTEGER {

off(0),

on(1)

}

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"UL line current IDP350s."

::= { idp350v13ParamEntry 5 }

idp350v13UH OBJECT-TYPE

SYNTAX INTEGER {

off(0),

on(1)

}

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"UH line current IDP350s."

::= { idp350v13ParamEntry 6 }

idp350v13T OBJECT-TYPE

SYNTAX DisplayString

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Temperature line current IDP350s."

::= { idp350v13ParamEntry 7 }

idp350v13UOut OBJECT-TYPE

SYNTAX DisplayString

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"U_OUT line current IDP350s."

::= { idp350v13ParamEntry 8 }

idp350v13IOut OBJECT-TYPE

SYNTAX DisplayString

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"I_OUT line current IDP350s."

::= { idp350v13ParamEntry 9 }

idp350v13Iutechki OBJECT-TYPE

SYNTAX DisplayString

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"I-ullage line current IDP350s."

::= { idp350v13ParamEntry 10 }

idp350v13Jumpers OBJECT-TYPE

SYNTAX INTEGER

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Jumpers line current IDP350s."

::= { idp350v13ParamEntry 11 }

idp350v13JumpersSet OBJECT-TYPE

SYNTAX INTEGER

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"Jumpers-set line current IDP350s."

::= { idp350v13ParamEntry 12 }

idp350v13UNomSet OBJECT-TYPE

SYNTAX DisplayString

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"I_NOM_SET line current IDP350s."

::= { idp350v13ParamEntry 13 }

idp350v13INomSet OBJECT-TYPE

SYNTAX DisplayString

MAX-ACCESS read-write

STATUS current

DESCRIPTION

```
        "I_NOM_SET line current IDP350s."
  ::= { idp350v13ParamEntry 14 }

 idp350v13IxxSet OBJECT-TYPE
  SYNTAX      DisplayString
  MAX-ACCESS  read-write
  STATUS      current
  DESCRIPTION
        "I_XX_SET line current IDP350s."
  ::= { idp350v13ParamEntry 15 }

 idp350v13RUte4Set OBJECT-TYPE
  SYNTAX      DisplayString
  MAX-ACCESS  read-write
  STATUS      current
  DESCRIPTION
        "R_UTE4_SET line current IDP350s."
  ::= { idp350v13ParamEntry 16 }



----------------------
--            idp350v13 TRAPS //!!!
--
  idp350v13Traps                    OBJECT IDENTIFIER ::= { idp350v13 2 }
  idp350v13TrapList                 OBJECT IDENTIFIER ::= { idp350v13Traps 1 }

 idp350v13BoardUp NOTIFICATION-TYPE
  OBJECTS { idp350v13Index }
  STATUS      current
  DESCRIPTION
        "A deviceTrap trap is sent when the device exist changes."
  ::= { idp350v13TrapList 1 }
```

```
 idp350v13BoardDown NOTIFICATION-TYPE
   OBJECTS { idp350v13Index }
   STATUS      current
   DESCRIPTION
         "A deviceTrap trap is sent when the device exist changes."
   ::= { idp350v13TrapList 2 }

 idp350v13Alarm NOTIFICATION-TYPE
   OBJECTS { idp350v13Index, idp350v13State }
   STATUS      current
   DESCRIPTION
         "A idp350v13Alarm trap is sent when the
         value of an instance idp350v13State changes alarm."
   ::= { idp350v13TrapList 3 }

 idp350v13Normal NOTIFICATION-TYPE
   OBJECTS { idp350v13Index, idp350v13State }
   STATUS      current
   DESCRIPTION
         "A idp350v13Normal trap is sent when the
         value of an instance idp350v13State changes normal."
   ::= { idp350v13TrapList 4 }


--------------------------------------------------------------------------
--            PUV2 //!!!
--


----------------
--            PUV2 PARAM //!!!
--
```

```
  puv2ParamTable OBJECT-TYPE
   SYNTAX      SEQUENCE OF PUV2PARAMEntry
   MAX-ACCESS  not-accessible
   STATUS      current
   DESCRIPTION
        "Contains information."
   ::= { puv2 1 }

  puv2ParamEntry OBJECT-TYPE
   SYNTAX      PUV2PARAMEntry
   MAX-ACCESS  not-accessible
   STATUS      current
   DESCRIPTION
         "An entry containing information about PARAM ports."
   INDEX   { puv2Index }
   ::= { puv2ParamTable 1 }

  PUV2PARAMEntry ::= SEQUENCE {
      puv2Index             INTEGER,
      puv2Exist             INTEGER,
      puv2State             DisplayString,
      puv2Status1           INTEGER,
puv2Status2                 INTEGER,
puv2Mode1                   INTEGER,
puv2Mode2                   INTEGER,
puv2SpeedMode1              INTEGER,
puv2SpeedMode2              INTEGER,
puv2I1                      INTEGER,
puv2I2                      INTEGER,
puv2Speed1                  INTEGER,
puv2Speed2                  INTEGER
      }
```

```
puv2Index OBJECT-TYPE
 SYNTAX      INTEGER (1..8)
 MAX-ACCESS  read-only
 STATUS      current
 DESCRIPTION
       "A unique value, greater than zero."
::= { puv2ParamEntry 1 }

puv2Exist OBJECT-TYPE
 SYNTAX      INTEGER {
                                not-exist(0),
                                exist(1)
                                }
 MAX-ACCESS  read-only
 STATUS      current
 DESCRIPTION
       "This variable indicates the Exist Board."
::= { puv2ParamEntry 2 }

puv2State OBJECT-TYPE
 SYNTAX      DisplayString
 MAX-ACCESS  accessible-for-notify
 STATUS      current
 DESCRIPTION
       "This variable indicates the Line Status of the interface.
       ALARM is sent when wedging is occurred."
::= { puv2ParamEntry 3 }

puv2Status1 OBJECT-TYPE
 SYNTAX      INTEGER {
           on(0),
```

```
            off(1)
}
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
            "This parameter containing first fan status."
    ::= { puv2ParamEntry 4 }

   puv2Status2 OBJECT-TYPE
    SYNTAX      INTEGER {
            on(0),
            off(1)
}
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
            "This parameter containing second fan status."
    ::= { puv2ParamEntry 5 }

   puv2Mode1 OBJECT-TYPE
    SYNTAX      INTEGER {
            auto(0),
            manual(1)
}
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
            "This parameter containing first fan mode.
    auto - program switches this fan
    manual - user switches this fan(program don't)"
    ::= { puv2ParamEntry 6 }
```

```
  puv2Mode2 OBJECT-TYPE
   SYNTAX      INTEGER {
               auto(0),
               manual(1)
}
   MAX-ACCESS  read-write
   STATUS      current
   DESCRIPTION
         "This parameter containing second fan mode.
   auto - program switches this fan
   manual - user switches this fan(program don't)"
   ::= { puv2ParamEntry 7 }

  puv2SpeedMode1 OBJECT-TYPE
   SYNTAX      INTEGER (0..9)
   MAX-ACCESS  read-write
   STATUS      current
   DESCRIPTION
         "This parameter containing first fan speed mode."
   ::= { puv2ParamEntry 8 }

  puv2SpeedMode2 OBJECT-TYPE
   SYNTAX      INTEGER (0..9)
   MAX-ACCESS  read-write
   STATUS      current
   DESCRIPTION
         "This parameter containing second fan speed mode. "
   ::= { puv2ParamEntry 9 }

  puv2I1 OBJECT-TYPE
   SYNTAX      INTEGER (0..2000)
   MAX-ACCESS  read-only
```

```
  STATUS      current
  DESCRIPTION
        "This parameter containing first fan current (mA)."
  ::= { puv2ParamEntry 10 }

 puv2I2 OBJECT-TYPE
  SYNTAX      INTEGER (0..2000)
  MAX-ACCESS  read-only
  STATUS      current
  DESCRIPTION
        "This parameter containing second fan current (mA)."
  ::= { puv2ParamEntry 11 }

 puv2Speed1 OBJECT-TYPE
  SYNTAX      INTEGER (0..3500)
  MAX-ACCESS  read-only
  STATUS      current
  DESCRIPTION
        "This parameter containing first fan speed (rpm)."
  ::= { puv2ParamEntry 12 }

 puv2Speed2 OBJECT-TYPE
  SYNTAX      INTEGER (0..3500)
  MAX-ACCESS  read-only
  STATUS      current
  DESCRIPTION
        "This parameter containing second fan speed (rpm). "
  ::= { puv2ParamEntry 13 }

----------------------
--            puv2 TRAPS //!!!
--
```

```
 puv2Traps                      OBJECT IDENTIFIER ::= { puv2 2 }

 puv2TrapList                   OBJECT IDENTIFIER ::= { puv2Traps 1 }


puv2BoardUp NOTIFICATION-TYPE

  OBJECTS { puv2Index }

  STATUS      current

  DESCRIPTION

        "A deviceTrap trap is sent when the device exist changes."

  ::= { puv2TrapList 1 }


puv2BoardDown NOTIFICATION-TYPE

  OBJECTS { puv2Index }

  STATUS      current

  DESCRIPTION

        "A deviceTrap trap is sent when the device exist changes."

  ::= { puv2TrapList 2 }


puv2Alarm NOTIFICATION-TYPE

  OBJECTS { puv2Index, puv2State }

  STATUS      current

  DESCRIPTION

        "A puv2Alarm trap is sent when the

        value of an instance puv2State changes alarm."

  ::= { puv2TrapList 3 }


puv2Normal NOTIFICATION-TYPE

  OBJECTS { puv2Index, puv2State }

  STATUS      current

  DESCRIPTION

        "A puv2Normal trap is sent when the

        value of an instance puv2State changes normal."

  ::= { puv2TrapList 4 }
```

```
--------------------------------------------------------------------------
--            UKA //!!!
--


----------------
--            UKA PARAM //!!!
--


  ukaParamTable OBJECT-TYPE
   SYNTAX      SEQUENCE OF UKAPARAMEntry
   MAX-ACCESS  not-accessible
   STATUS      current
   DESCRIPTION
        "Contains information."
   ::= { uka 1 }

  ukaParamEntry OBJECT-TYPE
   SYNTAX      UKAPARAMEntry
   MAX-ACCESS  not-accessible
   STATUS      current
   DESCRIPTION
        "An entry containing information about PARAM ports."
   INDEX   { ukaIndex }
   ::= { ukaParamTable 1 }

  UKAPARAMEntry ::= SEQUENCE {
      ukaIndex                INTEGER,
      ukaExist                INTEGER,
      ukaState                DisplayString,
ukaBCount                     INTEGER,
ukaU1                         DisplayString,
```

```
ukaU2                   DisplayString,
ukaU3                   DisplayString,
ukaU4                   DisplayString,
ukaU5                   DisplayString,
ukaI                    DisplayString,
ukaCurBreaker           INTEGER,
ukaSwCtrlSwitch         INTEGER,
ukaI2cAddress           INTEGER,
ukaReboot               INTEGER
    }

  ukaIndex OBJECT-TYPE
   SYNTAX     INTEGER (1..4)
   MAX-ACCESS  read-only
   STATUS     current
   DESCRIPTION
        "A unique value, greater than zero."
   ::= { ukaParamEntry 1 }

  ukaExist OBJECT-TYPE
   SYNTAX     INTEGER {
                        not-exist(0),
                        exist(1)
                        }
   MAX-ACCESS  read-only
   STATUS     current
   DESCRIPTION
        "This variable indicates the Exist Board."
   ::= { ukaParamEntry 2 }

  ukaState OBJECT-TYPE
   SYNTAX     DisplayString
```

MAX-ACCESS accessible-for-notify

STATUS   current

DESCRIPTION

"This variable indicates the Line Status of the interface.

ALARM U1(or U2 ... U5) is sent when U1(or U2 ... U5) is lesser when Umin."

::= { ukaParamEntry 3 }

ukaBCount OBJECT-TYPE

SYNTAX   INTEGER {

four(4),

five(5)

}

MAX-ACCESS read-write

STATUS   current

DESCRIPTION

"This variable indicates count of accumulators."

::= { ukaParamEntry 4 }

ukaU1 OBJECT-TYPE

SYNTAX   DisplayString

MAX-ACCESS read-only

STATUS   current

DESCRIPTION

"This variable indicates the voltage(V) of first accumulator."

::= { ukaParamEntry 5 }

ukaU2 OBJECT-TYPE

SYNTAX   DisplayString

MAX-ACCESS read-only

STATUS   current

DESCRIPTION

"This variable indicates the voltage(V) of second accumulator."

::= { ukaParamEntry 6 }

ukaU3 OBJECT-TYPE

SYNTAX DisplayString

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"This variable indicates the voltage(V) of fird accumulator."

::= { ukaParamEntry 7 }

ukaU4 OBJECT-TYPE

SYNTAX DisplayString

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"This variable indicates the voltage(V) of fourth accumulator."

::= { ukaParamEntry 8 }

ukaU5 OBJECT-TYPE

SYNTAX DisplayString

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"This variable indicates the voltage(V) of fifth accumulator."

::= { ukaParamEntry 9 }

ukaI OBJECT-TYPE

SYNTAX DisplayString

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"This variable indicates current(A)."

::= { ukaParamEntry 10 }

ukaCurBreaker OBJECT-TYPE

SYNTAX INTEGER {

off(0),

on(1)

}

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"This variable indicates state of curcuit breaker."

::= { ukaParamEntry 11 }

ukaSwCtrlSwitch OBJECT-TYPE

SYNTAX INTEGER {

off(0),

on(1)

}

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"This variable indicates state of software controlable switch(power MOSFET)."

::= { ukaParamEntry 12 }

ukaI2cAddress OBJECT-TYPE

SYNTAX INTEGER {

uka3(66),

uka4(67)

}

MAX-ACCESS read-write

STATUS current

```
  DESCRIPTION
        "This variable sets i2cAddress."
  ::= { ukaParamEntry 13 }

 ukaReboot OBJECT-TYPE
  SYNTAX     INTEGER {
                              reboot(1)
                              }
  MAX-ACCESS  read-write
  STATUS     current
  DESCRIPTION
        "This variable reboots uka."
  ::= { ukaParamEntry 14 }

----------------------
--            uka TRAPS //!!!
--
  ukaTraps                          OBJECT IDENTIFIER ::= { uka 2 }
  ukaTrapList                       OBJECT IDENTIFIER ::= { ukaTraps 1 }

 ukaBoardUp NOTIFICATION-TYPE
  OBJECTS { ukaIndex }
  STATUS     current
  DESCRIPTION
        "A deviceTrap trap is sent when the device exist changes."
  ::= { ukaTrapList 1 }

 ukaBoardDown NOTIFICATION-TYPE
  OBJECTS { ukaIndex }
  STATUS     current
  DESCRIPTION
        "A deviceTrap trap is sent when the device exist changes."
```

```
  ::= { ukaTrapList 2 }

 ukaAlarm NOTIFICATION-TYPE
  OBJECTS { ukaIndex, ukaState }
  STATUS      current
  DESCRIPTION
        "A ukaAlarm trap is sent when the
        value of an instance ukaState changes alarm."
  ::= { ukaTrapList 3 }

 ukaNormal NOTIFICATION-TYPE
  OBJECTS { ukaIndex, ukaState }
  STATUS      current
  DESCRIPTION
        "A ukaNormal trap is sent when the
        value of an instance ukaState changes normal."
  ::= { ukaTrapList 4 }



--------------------------------------------------------------------------------
--            UKN //!!!
--


----------------
--            UKN PARAM //!!!
--


  uknParamTable OBJECT-TYPE
   SYNTAX      SEQUENCE OF UKNPARAMEntry
   MAX-ACCESS  not-accessible
   STATUS      current
   DESCRIPTION
```

"Contains information."

::= { ukn 1 }

uknParamEntry OBJECT-TYPE

SYNTAX UKNPARAMEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"An entry containing information about PARAM ports."

INDEX { uknIndex }

::= { uknParamTable 1 }

UKNPARAMEntry ::= SEQUENCE {

uknIndex INTEGER,

uknExist INTEGER,

uknState DisplayString,

uknU DisplayString,

uknI DisplayString,

uknCurBreaker INTEGER,

uknSwCtrlSwitch INTEGER,

uknWithKeyMode INTEGER,

uknUChargeMin DisplayString,

uknTimeSOff INTEGER,

uknI2cAddress INTEGER,

uknReboot INTEGER

}

uknIndex OBJECT-TYPE

SYNTAX INTEGER (1..4)

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"A unique value, greater than zero."

::= { uknParamEntry 1 }

uknExist OBJECT-TYPE

SYNTAX INTEGER {

not-exist(0),

exist(1)

}

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"This variable indicates the Exist Board."

::= { uknParamEntry 2 }

uknState OBJECT-TYPE

SYNTAX DisplayString

MAX-ACCESS accessible-for-notify

STATUS current

DESCRIPTION

"This variable indicates the Line Status of the interface."

::= { uknParamEntry 3 }

uknU OBJECT-TYPE

SYNTAX DisplayString

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"This variable indicates the voltage(V) of fifth accumulator."

::= { uknParamEntry 4 }

uknI OBJECT-TYPE

SYNTAX DisplayString

```
 MAX-ACCESS  read-only
 STATUS      current
 DESCRIPTION
       "This variable indicates current(A)."
 ::= { uknParamEntry 5 }

uknCurBreaker OBJECT-TYPE
 SYNTAX      INTEGER {
                off(0),
                on(1)
                }
 MAX-ACCESS  read-only
 STATUS      current
 DESCRIPTION
       "This variable indicates state of curcuit breaker."
 ::= { uknParamEntry 6 }

uknSwCtrlSwitch OBJECT-TYPE
 SYNTAX      INTEGER {
                off(0),
                on(1)
                }
 MAX-ACCESS  read-write
 STATUS      current
 DESCRIPTION
       "This variable indicates state of software controlable switch(power MOSFET)."
 ::= { uknParamEntry 7 }

uknWithKeyMode OBJECT-TYPE
 SYNTAX      INTEGER {
                off(0),
                on(1)
```

```
                }
 MAX-ACCESS  read-write
 STATUS      current
 DESCRIPTION
        "This variable indicates if MSK controls software controlable switch (power MOSFET)."
 ::= { uknParamEntry 8 }

uknUChargeMin OBJECT-TYPE
 SYNTAX      DisplayString
 MAX-ACCESS  read-write
 STATUS      current
 DESCRIPTION
        "This variable indicates when MSK should switch off software controlable switch (power MOSFET)."
 ::= { uknParamEntry 9 }

uknTimeSOff OBJECT-TYPE
 SYNTAX      INTEGER
 MAX-ACCESS  read-write
 STATUS      current
 DESCRIPTION
        "This variable indicates when MSK should switch off software controlable switch (power MOSFET)
        after UChargeMin had happend."
 ::= { uknParamEntry 10 }

uknI2cAddress OBJECT-TYPE
 SYNTAX      INTEGER {
                ukn1(64),
                ukn2(65)
                }
 MAX-ACCESS  read-write
 STATUS      current
 DESCRIPTION
```

```
        "This variable sets i2cAddress."
    ::= { uknParamEntry 11 }

  uknReboot OBJECT-TYPE
    SYNTAX      INTEGER {
                    reboot(1)
                    }
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "This variable reboots ukn."
    ::= { uknParamEntry 12 }

----------------------
--              ukn TRAPS //!!!
--
  uknTraps                              OBJECT IDENTIFIER ::= { ukn 2 }
  uknTrapList                           OBJECT IDENTIFIER ::= { uknTraps 1 }

  uknBoardUp NOTIFICATION-TYPE
    OBJECTS { uknIndex }
    STATUS      current
    DESCRIPTION
        "A deviceTrap trap is sent when the device exist changes."
    ::= { uknTrapList 1 }

  uknBoardDown NOTIFICATION-TYPE
    OBJECTS { uknIndex }
    STATUS      current
    DESCRIPTION
        "A deviceTrap trap is sent when the device exist changes."
    ::= { uknTrapList 2 }
```

Описание информационных сообщений, передаваемых по протоколу SNMP.

Наименование аварии	Описание	Дополнительные параметры
SystemInitTrap ***1.3.6.1.4.1.27142.1.12.1.2.1.1***	Старт системы. Используется для сброса аварий после перезагрузки системы или при установлении связи с сервером после ее потери.	
Supply		
SupplyAlarm ***1.3.6.1.4.1.27142.1.12.1.2.1.1*** — установка аварии ***SupplyNormal*** ***1.3.6.1.4.1.27142.1.12.1.2.1.2*** — сброс аварии	Напряжение сети выходит за границы допустимых значений (85-265 В).	***SupplyUPhase*** — Напряжение сети (85-265 В)
SupplyNoPhaseAlarm ***1.3.6.1.4.1.27142.1.12.1.2.1.3*** — установка аварии ***SupplyNormal*** ***1.3.6.1.4.1.27142.1.12.1.2.1.2*** — сброс аварии	Пропало напряжение сети.	
Guard		
GuardAlarm ***1.3.6.1.4.1.27142.1.12.2.2.1.1*** — установка аварии ***GuardNormal*** ***1.3.6.1.4.1.27142.1.12.2.2.1.2*** — сброс аварии	Изменение состояния охранной системы	***GuardKick*** — удар ***GuardLock*** — вскрытие периметра
Climate		
ClimateAlarm ***1.3.6.1.4.1.27142.1.12.3.2.1.1*** — установка аварии ***ClimateNormal*** ***1.3.6.1.4.1.27142.1.12.3.2.1.2*** — сброс аварии	Превышение устанавливаемого порога одним из датчиков температуры	***IndexTS*** — Номер датчика ***Temperature*** — температура
2SHDSL-2E1-Eth		
DshdslBoardUp ***1.3.6.1.4.1.27142.1.12.4.2.1.1*** ***DshdslBoardDown*** ***1.3.6.1.4.1.27142.1.12.4.2.1.2***	Изъята плата 2shdsl-2e1 Установлена плата 2shdsl-2e1	***DshdslIndex*** — Номер платы
DshdslPcm30Alarm ***1.3.6.1.4.1.27142.1.12.4.2.1.3***	Изменилось состояние потока E1	***DshdslIndex*** — номер платы

— установка аварии ***DshdslPcm30Normal*** ***1.3.6.1.4.1.27142.1.12.4.2.1.4*** — сброс аварии		***DshdslPcm30Index*** — номер потока ***DshdslPcm30State*** — состояние потока PMC30: *NVP* — отсутствует входной поток; *PCS* — потеря цикловой синхронизации; *AUS* — авария удаленной стороны
DshdslShdslAlarm ***1.3.6.1.4.1.27142.1.12.4.2.1.5*** — установка аварии ***DshdslShdslNormal*** ***1.3.6.1.4.1.27142.1.12.4.2.1.6*** — сброс аварии	Изменилось состояние потока SHDSL	***DshdslIndex*** — номер платы ***DshdslShdslIndex*** — номер потока ***DshdslShdslState*** — состояние SHDSL потока: *Chain_Broken* — разрыв цепочки.
2SHDSL-4E1-Eth (SDFE)		
SdfeBoardUp ***1.3.6.1.4.1.27142.1.12.5.2.1.1***	Изъята плата 2shdsl-4e1-eth	***SdfeIndex*** — номер платы
SdfeBoardDown ***1.3.6.1.4.1.27142.1.12.5.2.1.2***	Установлена плата 2shdsl-4e1-eth	
SdfelPcm30Alarm ***1.3.6.1.4.1.27142.1.12.5.2.1.3*** — установка аварии ***SdfePcm30Normal*** ***1.3.6.1.4.1.27142.1.12.5.2.1.4*** — сброс аварии	Изменилось состояние потока E1	***SdfeIndex*** — номер платы ***SdfePcm30Index*** — номер потока ***SdfePcm30State*** — состояние потока PMC30: *NVP* — отсутствует входной поток; *PCS* — потеря цикловой синхронизации; *AUS* — авария удаленной стороны
SdfeShdslAlarm ***1.3.6.1.4.1.27142.1.12.5.2.1.5*** — установка аварии ***SdfeShdslNormal*** ***1.3.6.1.4.1.27142.1.12.5.2.1.6*** — сброс аварии	Изменилось состояние потока SHDSL	***SdfeIndex*** — номер платы ***SdfeShdslIndex*** — номер потока ***SdfeShdslState*** — состояние SHDSL потока: *Chain_Broken* — разрыв цепочки.
УМП		

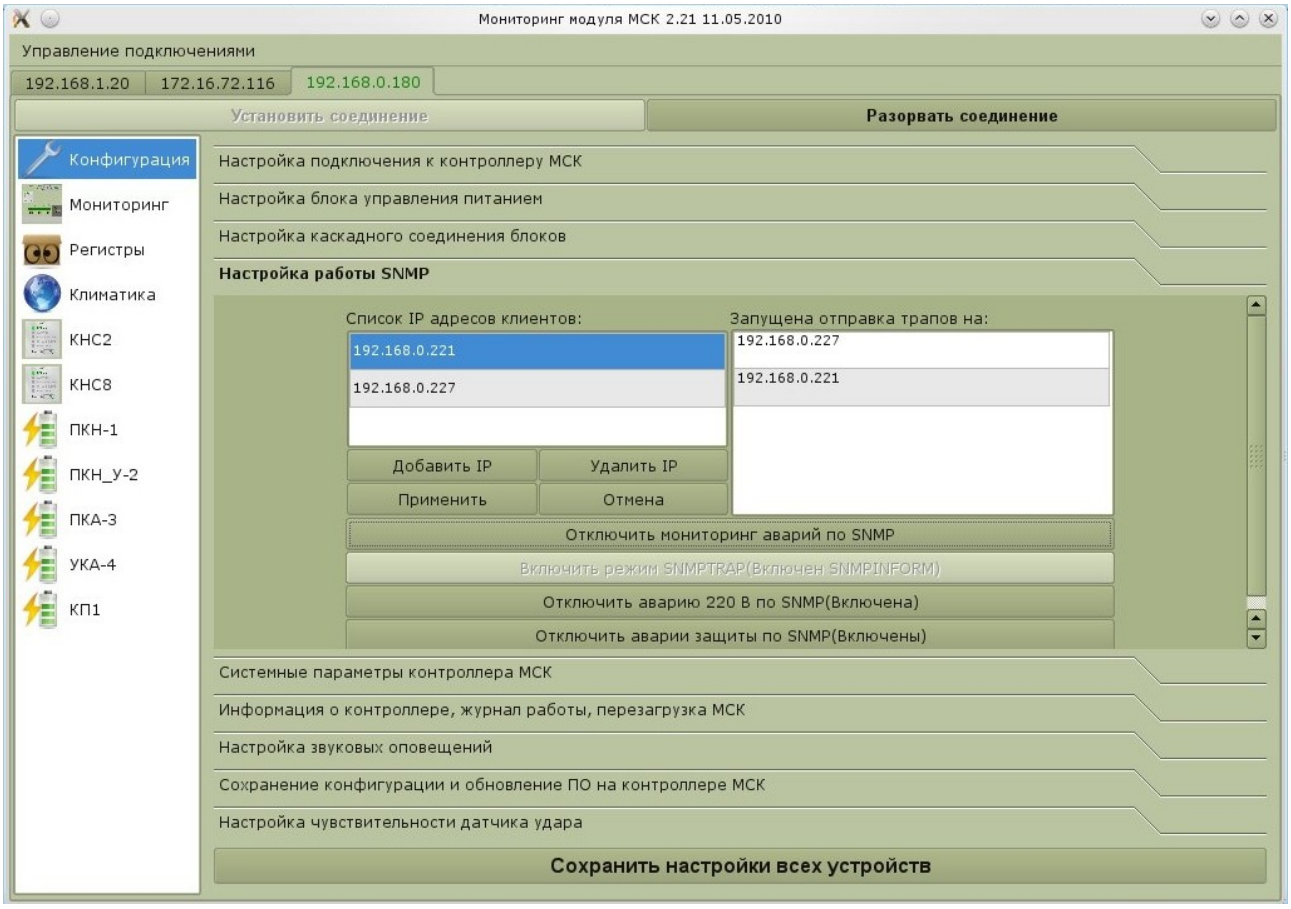
UmpBoardUp ***1.3.6.1.4.1.27142.1.12.6.2.1.1***	Изъята плата УМП	***UmpIndex*** — Номер платы
UmpBoardDown ***1.3.6.1.4.1.27142.1.12.6.2.1.2***	Установлена плата УМП	
UmpFlowAlarm ***1.3.6.1.4.1.27142.1.12.6.2.1.3*** — установка аварии ***UmpFlowNormal*** ***1.3.6.1.4.1.27142.1.12.6.2.1.4*** — сброс аварии	Изменилось состояние потока E1	***UmpIndex*** — номер платы
		UmpFlowIndex — номер потока
		UmpFlowState — состояние потока PMC30
ИДП-240		
Idp240BoardUp ***1.3.6.1.4.1.27142.1.12.8.2.1.1***	Изъята плата ИДП-240	***Idp240Index*** — номер платы
Idp240BoardDown ***1.3.6.1.4.1.27142.1.12.8.2.1.2***	Установлена плата ИДП-240	
Idp240Alarm ***1.3.6.1.4.1.27142.1.12.8.2.1.3*** — установка аварии ***Idp240Normal*** ***1.3.6.1.4.1.27142.1.12.8.2.1.4*** — сброс аварии	Изменилось состояние ИДП-240	***Idp240Index*** — номер платы ***Idp240Status*** — состояние ИДП-240: *ALARM* — общая авария; *SHORT-CIRCUIT* — короткое замыкание, напряжение на выходе ниже заданного порога; *LINE-BREAK* — обрыв, ток нагрузки ниже заданного порога; *LEAKAGE* — утечка, ток утечки выше заданного порога; *REJECT* — отказ, блок неисправен; *SHORT-CIRCUIT-I-MAX* — короткое замыкание в нагрузке.
ИДП-350		
Id3350BoardUp ***1.3.6.1.4.1.27142.1.12.9.2.1.1***	Изъята плата ИДП-350	***Idp350Index*** — номер платы
Idp350BoardDown ***1.3.6.1.4.1.27142.1.12.9.2.1.2***	Установлена плата ИДП-350	
Idp350Alarm ***1.3.6.1.4.1.27142.1.12.9.2.1.3*** — установка аварии	Изменилось состояние ИДП-350	***Idp350Index*** — номер платы

Idp350Normal 1.3.6.1.4.1.27142.1.12.9.2.1.4 — сброс аварии		***Idp350Status*** — состояние ИДП-350: *ALARM* — общая авария ИДП350; *SHORT-CIRCUIT* — короткое замыкание, напряжение на выходе ниже заданного порога; *LIMITATION* — ограничение, напряжение ниже заданного, ток равен току ограничения; *LINE-BREAK* — обрыв, ток нагрузки ниже заданного порога; *LEAKAGE* — утечка, ток утечки выше заданного порога; *REJECT1* — отказ1, отключен(нет напряжения и тока); *REJECT2* — отказ2, напряжение или ток больше нормы.
ИДП-350 v.1.3		
Idз350v13BoardUp 1.3.6.1.4.1.27142.1.12.13.2.1.1 ***Idp350v13BoardDown 1.3.6.1.4.1.27142.1.12.13.2.1.2***	Изъята плата ИДП-350v13 Установлена плата ИДП-350v13	***Idp350v13Index*** — номер платы
Idp350v13Alarm 1.3.6.1.4.1.27142.1.12.13.2.1.3 — установка аварии ***Idp350v13Normal 1.3.6.1.4.1.27142.1.12.13.2.1.4*** — сброс аварии	Изменилось состояние ИДП-350v13	***Idp350v13Index*** — номер платы ***Idp350v13Status*** — состояние ИДП-350v13: *ALARM* — общая авария ИДП350; *SHORT-CIRCUIT* — короткое замыкание, напряжение на выходе ниже заданного порога; *LIMITATION* — ограничение, напряжение ниже заданного, ток равен току ограничения; *LINE-BREAK* — обрыв, ток нагрузки ниже заданного порога;

		LEAKAGE — утечка, сопротивление утечки ниже заданного порога; *REJECT1* — отказ1, отключен(нет напряжения и тока); *REJECT2* — отказ2, напряжение или ток больше нормы.
КНС		
KnsBoardUp ***1.3.6.1.4.1.27142.1.12.7.2.1.1***	Изъята плата КНС	***KnsIndex*** — номер платы
KnsBoardDown ***1.3.6.1.4.1.27142.1.12.7.2.1.2***	Установлена плата КНС	
KnsAlarm ***1.3.6.1.4.1.27142.1.12.7.2.1.3*** — установка аварии ***KnsNormal*** ***1.3.6.1.4.1.27142.1.12.7.2.1.4*** — сброс аварии	Изменилось состояние КНС	***KnsIndex*** — номер платы ***KnsStatus*** — состояние КНС: *ALARM* — общая авария КНС; *ALARM SUPPLY* — авария сети, нет входного напряжения; *LIMITATION* — ограничение, ток нагрузки >= максимального; *OVERHEAT* — перегрев
ПУВ2		
Puv2BoardUp ***1.3.6.1.4.1.27142.1.12.14.2.1.1***	Изъято устройство/плата ПУВ2	***Puv2Index*** — номер устройства/платы
Puv2BoardDown ***1.3.6.1.4.1.27142.1.12.14.2.1.2***	Установлено устройство/плата ПУВ2	
Puv2Alarm ***1.3.6.1.4.1.27142.1.12.14.2.1.3*** — установка аварии ***Puv2Normal*** ***1.3.6.1.4.1.27142.1.12.14.2.1.4*** — сброс аварии	Изменилось состояние ПУВ2	***Puv2Index*** — номер устройства/платы ***Puv2State*** — состояние ПУВ2: *ALARM1* — произошло заклинивание первого вентилятора; *ALARM2* — произошло заклинивание второго вентилятора.
УКА		
UkaBoardUp	Изъято устройство/плата УКА	***UkaIndex*** — номер

1.3.6.1.4.1.27142.1.12.15.2.1.1		устройства/платы
UkaBoardDown ***1.3.6.1.4.1.27142.1.12.15.2.1.2***	Установлено устройство/плата УКА	
UkaAlarm ***1.3.6.1.4.1.27142.1.12.15.2.1.3*** — установка аварии ***UkaNormal*** ***1.3.6.1.4.1.27142.1.12.15.2.1.4*** — сброс аварии	Изменилось состояние УКА	***UkaIndex*** — номер устройства/платы
		UkaState — состояние УКА: *U1* — U1 меньше Umin; *U2* — U2 меньше Umin; *U3* — U3 меньше Umin; *U4* — U4 меньше Umin; *U5* — U5 меньше Umin
УКН/ПКН/ПКН_У		
UknBoardUp ***1.3.6.1.4.1.27142.1.12.16.2.1.1***	Изъято устройство/плата УКН/ПКН/ПКН-У	***UknIndex*** — номер устройства/платы
UknBoardDown ***1.3.6.1.4.1.27142.1.12.16.2.1.2***	Установлено устройство/плата УКН/ПКН/ПКН-У	
Напряжение		
SupplyUPhaseTrap ***1.3.6.1.4.1.27142.1.12.1.2.1.4***	Выслано значение напряжения	***SupplyUPhase*** — значения напряжения
Температура		
ClimateTempStat ***1.3.6.1.4.1.27142.1.12.3.2.1.5***	Выслано значение температур.	***climateSensTemp*** — значения температур всех датчиков.

Настройка отправки snmp-trap пакетов с помощью программы mskmon.



Здесь можно включить/отключить и настроить отправку аварий МСК по протоколу SNMP.

Для настройки мониторинга аварий МСК по протоколу SNMP выбираем вкладку ***«Настройки работы SNMP»*** (Рис.**13**). Там необходимо настроить следующие параметры:

1. IP адреса SNMP клиентов (менеджеров)

2. Нажать кнопку ***«Применить»*** или ***«Отмена»***, если передумали задавать IP адреса SNMP клиентов (менеджеров)

3. Нажать кнопку ***«Сохранить настройки всех устройств»****(Если не нажать, то настройки не сохранятся после перезагрузки)*

4. По усмотрению оператора можно отключить ненужные виды аварий, нажав соответствующие кнопки и сменить вид аварий (SNMPINFORM(по умолчанию) — с гарантированной доставкой, SNMPTRAP — без гарантированной доставки)

5. Нажать кнопку ***«Включить мониторинг аварий по SNMP»***

6. Нажать кнопку ***«Сохранить настройки всех устройств»*** еще раз !!! (чтобы сохранилось состояние самого мониторинга аварий по SNMP)

IP адреса клиентов(менеджеров) SNMP, изменяются и при «включенном мониторинге аварий по SNMP». После изменений IP адресов и последующего применения следует нажать кнопку ***«Сохранить настройки всех устройств».***