

```
Rmu DEFINITIONS ::= BEGIN
IMPORTS
IpAddress, Integer32, OBJECT-TYPE, NOTIFICATION-TYPE,
MODULE-IDENTITY, enterprises, OBJECT-IDENTITY
FROM SNMPv2-SMI
OBJECT-GROUP, NOTIFICATION-GROUP
FROM SNMPv2-CONF
DisplayString
FROM SNMPv2-TC;
```

```
tflRemoteMonitoringHeaderMIB MODULE-IDENTITY
LAST-UPDATED "201503111304Z"
ORGANIZATION "TfL"
CONTACT-INFO "Gurmeet Sahotay
```

Transport for London
197 Blackfriars Road
London
SE1 8NJ

Tel +44 203 054 2136 email: Gurmeet.Sahotay@tfl.gov.uk"

```
DESCRIPTION "Introduction
```

The Remote Monitoring Unit is either a separate or integral subsystem of any ITS Equipment (the ITS equipment may be a traffic light Controller, Traffic processing electronics unit and an OHVD processing unit).

This document defines an IP based Remote Monitoring MIB for ITS equipment.. The MIB is broken down into a number of sub MIBs which each provides the Remote Monitoring capabilities for each equipment type in a modular format. The tflRemoteMonitoringITSHeaderMIB is the main MIB which calls each sub MIB.

MIB Rationale

The approach this MIB takes is to expose the current status of the equipment being monitored.

It is less concerned with the history of the equipment. This is a deliberate design decision to keep the RMU as simple as possible:

No need to synchronise logs between RMU and equipment being monitored.

No need to maintain faults and clears on the RMU.

No need for the RMU to discover and store historical information about the equipment being monitored.

Allows the SNMP In-Station to walk the MIB for all relevant information.

For simple cases, allows the use of off-the-shelf Network-Management-Software to act as the SNMP In-Station.

TfL envisages that the SNMP In-Station might work as follows:

When a RMU comes online for the first time, the SNMP In-Station shall walk the MIB tree to discover all information about the unit.

The SNMP In-Station shall periodically poll strategic objects to check the status of the RMU and to confirm that the RMU is reachable. If an object shows something of interest then the SNMP In-Station shall poll further objects for more detail. For example: the MIB provides a high level lamp status object to expose any lamp issue; if there is an issue then a walk of further objects will reveal the specific phase and lamp colour.

The SNMP In-Station can enable INFORMs for specific objects on the RMU. This shall cause the RMU to send a INFORM if the value of that object changes. This allows for immediate fault reporting if desired.

The RMU shall handle events of interest by assigning a Date and Time stamp to the specific objects. For example there is an RMU Last Reset object that exposes when the RMU was most recently reset."

REVISION "201503111304Z"

DESCRIPTION "Changes to the tlcDetectorStatusTable created detectorSetting and detectorStatus to better define the state, fault and setting of each detector to the controller."

REVISION "201503031546Z"

DESCRIPTION "rmuSystemSyncMethod should read/write and rmuConfigInputBits should be Read only All tables modified where where accessible-for-notify used replaced with read only. Required to be accessible if a table walk is requested. OID rmuConfigInputBits and OID rmuConfigInputBits. If not Implemented set to zero, and integer changed from m1.128 to 0..128. secondaryPowerType OID noBackupOption (6) added. Detector Status - added a detectorSetToInhibit

(4) state. for better detector diagnostics" REVISION "201411111048Z" DESCRIPTION "TfL (IM) - Reviewed the Power monitoring functions of the rmu changed the descriptions and not supported functions. Have changed the rmuMainsSupplyStatus to rmuPrimaryPowerStatus and rmuPrimaryPowerSupplyType OIDs Have changed the rmuBatteryStatus which has been replaced with rmuSecondaryPowerStatus and rmuSecondaryPowerType. This still provides similar options for a alternative supply source. TfL- notifications naming convention uses Event and Changed. Will use Event Only notification names updated"

REVISION "201411101545Z"

DESCRIPTION "TfL- ohvdPowerFailure and ohvdHeaterFailure where set to obsolete status, these should be set to current status Telent - level 3 error (SMI Error, tolerated by many implementations) Last update time not correct manually corrected and revisions in reverse order."

REVISION "201411071452Z"

DESCRIPTION "TfL - Warning (level 5), line 455,559,810,1006,1159:
[integer-misuse] use Integer32 instead of INTEGER in SMiv2"
REVISION "201411071446Z"
DESCRIPTION "TfL - Description updated on rmuMainPowerStatus. Siemens -
Updated movaErrorVaueExceeded object, Description of error with the
correct meaning
and naming convention of the error bit. This
has now been corrected and is in-line with MOVA definition. The
redSignIndex and colourConflictPhaseIndex should be accessible-for-
notify"
REVISION "201411041720Z"
DESCRIPTION "Siemens comments - rmuConfigInputBits should be
an Integer 1..128 rmuConfigOutputBits should
be an Integer 1..128 redPhaseName -Should be
a DisplayString Not an Integer"
REVISION "201411041700Z"
DESCRIPTION "TfL - Addtional Information added to the Description of
streamModeSubData"
REVISION "201411041655Z" DESCRIPTION "TfL- movaErrorValueExceeded
Description amended previously incorrect description."
REVISION "201411041314Z"
DESCRIPTION "TfL - Additional description added tflStreamEntry3" REVISION
"201411041237Z"
DESCRIPTION "TfL - inputPortPolarity changed to inputBitPolarity
continuity with naming conventions"
REVISION "201411041023Z"
DESCRIPTION "Telent - rmuMainsPowerStatus - unsupported Value added to
the OID"
REVISION "201411031454Z"
DESCRIPTION "TfL - Imports removed streamAllOut (CHANGE-MY-NAME-MIB) from
Imports error with the editor tool. Amended OID
1.3.6.1.4.1.42773.1.1.21.1.4 naming error inputPortPolarity NOT
inportPortPolarity Ranges added to the Indexes for reg signs and colour
conflicts"
REVISION "201410231606Z"
DESCRIPTION "TfL - Regulatory Sign OID and Notification added to MIB"
REVISION "201410221053Z"
DESCRIPTION "This revision in corporates IM, SWARCO, SIEMENS comments for
V10 documented MIB, Includes descriptions and OID corrections. Detailed
Description with introduction added to the MIB. tlcFaultLogAlarm missing
from MIB Notifications has now been added Edited and added to MIB.
Descriptions to the
MIB objects. General Stream notification added
to tlcNotifications. Added Notification for rmuDeviceCommsand
tlcFaultLogAlarm OIDs" REVISION "201408210938Z"
DESCRIPTION "initial version of this module"

::= { transportforlondon 1 }

transportforlondon OBJECT IDENTIFIER ::= { enterprises 42773 }

rmuMIB OBJECT IDENTIFIER ::= { tflRemoteMonitoringHeaderMIB 1 }

rmuManufacturer OBJECT-TYPE

SYNTAX DisplayString

MAX-ACCESS read-only

STATUS current
DESCRIPTION "This object reveals the manufacturer of the RMU. This is a textual description."
::= { rmuMIB 1 }

rmuEquipmentType OBJECT-TYPE
SYNTAX DisplayString
MAX-ACCESS read-only
STATUS current
DESCRIPTION "This object reveals the equipment type identity of the on street equipment. This is a textual description which provides the authority with the manufacturers Equipment Type of the particular RMU installed."
::= { rmuMIB 2 }

rmuIPAddress OBJECT-TYPE
SYNTAX IPAddress
MAX-ACCESS read-only
STATUS current
DESCRIPTION "This object reveals the current setting of the RMU IP Address"
::= { rmuMIB 3 }

rmuGateway OBJECT-TYPE
SYNTAX IPAddress
MAX-ACCESS read-only
STATUS current
DESCRIPTION "This object reveals the current setting of the default gateway"
::= { rmuMIB 4 }

rmuBroadcastAddress OBJECT-TYPE
SYNTAX IPAddress
MAX-ACCESS read-only
STATUS current
DESCRIPTION "This object reveals the current setting of the RMU broadcast address."
::= { rmuMIB 5 }

rmuNetmask OBJECT-TYPE
SYNTAX IPAddress
MAX-ACCESS read-only
STATUS current
DESCRIPTION "This object reveals the current setting of the RMU network mask"
::= { rmuMIB 6 }

rmuDNSServerPrimary OBJECT-TYPE
SYNTAX IPAddress
MAX-ACCESS read-only
STATUS current
DESCRIPTION "This object reveals the IP address of the Primary DNS server"
::= { rmuMIB 7 }

rmuDNSServerSecondary OBJECT-TYPE
SYNTAX IPAddress

MAX-ACCESS read-only
STATUS current
DESCRIPTION "This object reveals the IP address of the Secondary DNS server"
::= { rmuMIB 8 }

rmuDHCP OBJECT-TYPE
SYNTAX INTEGER { dhcpON(1),
dhcpOFF(2)
}

MAX-ACCESS read-only
STATUS current
DESCRIPTION "This object reveals the status setting of the DHCP setting,"
DEFVAL { dhcpOFF }
::= { rmuMIB 9 }

rmuFSTable OBJECT-TYPE
SYNTAX SEQUENCE OF RmuFSEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION "This object contains a table that lists the RMU firmware and software modules"
::= { rmuMIB 10 }

rmuFSEntry OBJECT-TYPE
SYNTAX RmuFSEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION "Each entry contains the information for a single firmware or software module." INDEX {
rmuFSIndex
}
::= { rmuFSTable 1 }

RmuFSEntry ::= SEQUENCE {
rmuFSIndex
Integer32,
rmuFSModule
DisplayString,
rmuFSDescription
DisplayString,
rmuFSModuleBuildno
DisplayString,
rmuFSMD5Checksum
DisplayString
}

rmuFSIndex OBJECT-TYPE
SYNTAX Integer32 (1..256)
MAX-ACCESS read-only
STATUS current
DESCRIPTION "This object is an index number which increments from 1 onwards each entry is unique table entry." ::= { rmuFSEntry 1 }

rmuFSModule OBJECT-TYPE
SYNTAX DisplayString

MAX-ACCESS read-only
STATUS current
DESCRIPTION "This object is a textual name that identifies the firmware / software module."
::= { rmuFSEntry 2 }

rmuFSDescription OBJECT-TYPE
SYNTAX DisplayString
MAX-ACCESS read-only
STATUS current
DESCRIPTION "This object is a textual description of the firmware / software module functionality."
::= { rmuFSEntry 3 }

rmuFSModuleBuildno OBJECT-TYPE
SYNTAX DisplayString
MAX-ACCESS read-only
STATUS current
DESCRIPTION "This object is that textual field that uniquely identifies the firmware / software build. This is defined by the manufacturer. For example: this could be a release number or an issue date." ::= { rmuFSEntry 4 }

rmuFSMD5Checksum OBJECT-TYPE
SYNTAX DisplayString (SIZE(32)) MAX-ACCESS read-only
STATUS current
DESCRIPTION "This object reveals the textual field to provide the firmware / software MD5 Check Sum."
::= { rmuFSEntry 5 }

rmuTimeOfDay OBJECT-TYPE
SYNTAX DisplayString (SIZE(15))
UNITS "YYYYMMDDHHmmssZ"
MAX-ACCESS read-write
STATUS current
DESCRIPTION "This object reveals the date and time of the RMUs clock irrespective of whether it is synchronised or not. Z indicates Zulu or GMT. It can also be used to set the date and time."
::= { rmuMIB 11 }

rmuSystemSyncMethod OBJECT-TYPE
SYNTAX INTEGER {
notSet(1),
mainsSync(2),
sntp(3), ntp(4),
gps(5),
radio(6),
other(7)
}
MAX-ACCESS read-write
STATUS current
DESCRIPTION "This object reveals the method by which the RMUs clock is synchronised. This can be set."
DEFVAL { notSet }
::= { rmuMIB 12 }

rmuConfigSiteID OBJECT-TYPE

SYNTAX DisplayString

MAX-ACCESS read-write

STATUS current

DESCRIPTION "This object reveals the site ID. This is a textual description which enables the authority to uniquely identify the site e.g. the traffic signal site." ::= { rmuMIB 13 }

rmuConfigSiteAddress OBJECT-TYPE

SYNTAX DisplayString MAX-ACCESS read-write

STATUS current

DESCRIPTION "This object reveals the site address. This is a textual description which enables the authority to enter the location of the site e.g. street names, postcode, grid reference." ::= { rmuMIB 14 }

rmuConfigInStationAddress OBJECT-TYPE

SYNTAX DisplayString

MAX-ACCESS read-write

STATUS current

DESCRIPTION "This object is the address of the In-station to which the RMU sends INFORMs. This can be set.

This can be an IP address or a hostname and can include a port number preceded by a colon e.g. 192.168.1.1:12345."

::= { rmuMIB 15 }

rmuConfigNTPAddress1 OBJECT-TYPE SYNTAX DisplayString

MAX-ACCESS read-write

STATUS current

DESCRIPTION "This object reveals the Primary NTP server. This can be an IP address or host name."

::= { rmuMIB 16 }

rmuConfigNTPAddress2 OBJECT-TYPE

SYNTAX DisplayString

MAX-ACCESS read-write

STATUS current

DESCRIPTION "This object reveals the Secondary NTP server.

This can be an IP address or host name."

::= { rmuMIB 17 }

rmuConfigInformTableUpdated OBJECT-TYPE

SYNTAX DisplayString (SIZE(15))

UNITS "YYYYMMDDHHmmssZ"

MAX-ACCESS read-only

STATUS current

DESCRIPTION "This object reveals the date and time when the config Inform Table was last changed. Z indicates Zulu or GMT."

::= { rmuMIB 18 }

rmuConfigInformTable OBJECT-TYPE

SYNTAX SEQUENCE OF RmuConfigInformEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION "The table must contain one entry for each INFORM specified under the rmNotifications sub-tree

of this MIB. The table should list the entries

with the OIDs in numerical order."
 ::= { rmuMIB 19 }

rmuConfigInformEntry OBJECT-TYPE

SYNTAX RmuConfigInformEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION "Each entry identifies an INFORM by its Object Identifier and whether it is enabled or disabled." INDEX {

configInformIndex

} ::= { rmuConfigInformTable 1 }

RmuConfigInformEntry ::= SEQUENCE {

configInformIndex

Integer32,

configInformIdentifier

OBJECT IDENTIFIER,

configInformEnableStatus

INTEGER

}

configInformIndex OBJECT-TYPE

SYNTAX Integer32 (1..256)

MAX-ACCESS read-only

STATUS current

DESCRIPTION "This object is an index number which increments from 1 onwards each entry is unique table entry" ::= { rmuConfigInformEntry 1 }

configInformIdentifier OBJECT-TYPE

SYNTAX OBJECT IDENTIFIER

MAX-ACCESS read-only

STATUS current

DESCRIPTION "This object is the OID of the INFORM."

REFERENCE "rmNotifications"

::= { rmuConfigInformEntry 2 }

configInformEnableStatus OBJECT-TYPE

SYNTAX INTEGER { disable(1),

enable(2)

}

MAX-ACCESS read-write

STATUS current

DESCRIPTION "This object enables or disables the sending of any INFORMs. If set to disable then the RMU will not send any INFORMs. If set to enable then the

RMU will send INFORMs as configured in the rmuConfigInformTable. Note: this object is set to disable by default;

the In-station should ensure the rmuInstationAddress is correct before setting this object to enable." DEFVAL { disable }

::= { rmuConfigInformEntry 3 }

rmuConfigInputBits OBJECT-TYPE

SYNTAX Integer32 (0..128)

MAX-ACCESS read-only

STATUS current DESCRIPTION "This OID reveals the number of logical inputs available to the RMU. The number might vary by device. However this MIB supports up to 128 inputs. If not Implemented set to zero."

```
::= { rmuMIB 20 }
```

rmuConfigInputBitStatusTable OBJECT-TYPE

SYNTAX SEQUENCE OF RmuConfigInputBitStatusEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION "This table contains a list of all available input bits, textual details for each bit, configuration settings and current state."

```
::= { rmuMIB 21 }
```

rmuConfigInputBitStatusEntry OBJECT-TYPE

SYNTAX RmuConfigInputBitStatusEntry

MAX-ACCESS not-accessible

STATUS current DESCRIPTION "Each identifies a single logical input bit."

INDEX {

inputBitIndex

}

```
::= { rmuConfigInputBitStatusTable 1 }
```

RmuConfigInputBitStatusEntry ::= SEQUENCE {

inputBitIndex

Integer32,

inputBitDescription

DisplayString,

inputSetInform

INTEGER,

inputBitPolarity

INTEGER,

inputState

INTEGER

}

inputBitIndex OBJECT-TYPE

SYNTAX Integer32 (1..128)

MAX-ACCESS read-only

STATUS current

DESCRIPTION "This object is an index number which increments from 1 onwards, each entry is unique table entry the First entry shall start from the Least Significant bit."

```
::= { rmuConfigInputBitStatusEntry 1 }
```

inputBitDescription OBJECT-TYPE

SYNTAX DisplayString (SIZE(255))

MAX-ACCESS read-write

STATUS current

DESCRIPTION "This object returns a textual description that uniquely identifies the input bit."

```
::= { rmuConfigInputBitStatusEntry 2 }
```

inputSetInform OBJECT-TYPE

SYNTAX INTEGER {

disableINFORM(1),

enableINFORM(2)

}

MAX-ACCESS read-write

STATUS current

DESCRIPTION "This object identifies if the input is of interest to the In-station."

If this object is enabled then the RMU will generate an rnuInputStateChanged INFORM on each change of this input state. The INFORM shall contain the matching inputBitIndex, inputBitDescription and inputState.

Note: the above applies only if the rmuInputStateChanged INFORMs are enabled in the rmuConfigInformTable." ::= { rmuConfigInputBitStatusEntry 3 }

inputBitPolarity OBJECT-TYPE

SYNTAX INTEGER {
 normal(1),
 polarityInverted(2)
}

MAX-ACCESS read-write

STATUS current

DESCRIPTION "This object returns the input bit polarity. Normal implies TRUE = logic 1. Polarity inverted implies TRUE = Logic 0. This is configurable from the In-station." DEFVAL { normal }

::= { rmuConfigInputBitStatusEntry 4 }

inputState OBJECT-TYPE

SYNTAX INTEGER {
 logicTrue(1),
 logicFalse(2)
}

MAX-ACCESS read-only

STATUS current

DESCRIPTION "This object returns the input bit state of each input bit."

::= { rmuConfigInputBitStatusEntry 5 }

rmuConfigOutputBits OBJECT-TYPE

SYNTAX Integer32 (0..128)

MAX-ACCESS read-only

STATUS current

DESCRIPTION "This OID reveals the number of logical outputs available to the RMU. The number might vary by device. However this MIB supports up to 128 outputs. If not Implemented set to zero."

::= { rmuMIB 22 }

rmuConfigOutputBitStatusTable OBJECT-TYPE

SYNTAX SEQUENCE OF RmuConfigOutputBitStatusEntry MAX-ACCESS not-accessible

STATUS current

DESCRIPTION "This table contains a list of all available output bits, textual details for each bit, configuration settings and current state."

::= { rmuMIB 23 }

rmuConfigOutputBitStatusEntry OBJECT-TYPE

SYNTAX RmuConfigOutputBitStatusEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION "Each identifies a single logical output bit." INDEX {
 outputBitIndex
}

::= { rmuConfigOutputBitStatusTable 1 }

```

RmuConfigOutputBitStatusEntry ::= SEQUENCE {
    outputBitIndex
        Integer32,
    outputBitDescription   DisplayString,
    outputSetInform
        INTEGER,
    outputBitPolarity
        INTEGER,
    outputState
        INTEGER
}

```

```

outputBitIndex OBJECT-TYPE
SYNTAX Integer32 (1..128)
MAX-ACCESS accessible-for-notify
STATUS current
DESCRIPTION "This object returns the output Bit Index. The
    Bit numbering shall start from 1 up to 128"
::= { rmuConfigOutputBitStatusEntry 1 }

```

```

outputBitDescription OBJECT-TYPE
SYNTAX DisplayString (SIZE(64))
MAX-ACCESS read-write
STATUS current
DESCRIPTION "This object returns a textual description that uniquely
    identifies the output bit."
::= { rmuConfigOutputBitStatusEntry 2 }

```

```

outputSetInform OBJECT-TYPE
SYNTAX INTEGER {  disableINFORM( 1 ),
    enableINFORM( 2 )
}
MAX-ACCESS read-write
STATUS current
DESCRIPTION "This object identifies if the output is of interest to the
    In-station.

```

If this object is enabled then the RMU will generate an rnuOutputStateChanged INFORM on each change of this input state. The INFORM shall contain the matching outputBitIndex, outputBitDescription and inputState.

Note: the above applies only if the rmuOutputStateChanged INFORMs are enabled in the rmuConfigInformTable." ::= { rmuConfigOutputBitStatusEntry 3 }

```

outputBitPolarity OBJECT-TYPE SYNTAX INTEGER {
    normalLogic( 1 ),
    polarityInvertedLogic( 2 )
}
MAX-ACCESS read-write
STATUS current
DESCRIPTION "This object returns the output bit polarity. Normal implies
    TRUE = logic 1. Polarity inverted implies TRUE = Logic 0. This is
    configurable from the In-station."
::= { rmuConfigOutputBitStatusEntry 4 }

```

outputState OBJECT-TYPE

```
SYNTAX INTEGER {  
    truelogic( 1 ),  
    falselogic( 2 )  
}
```

MAX-ACCESS read-write

STATUS current

DESCRIPTION "This object returns the output bit state of each output bit." ::= { rmuConfigOutputBitStatusEntry 5 }

rmuLastResetStatus OBJECT-TYPE

```
SYNTAX DisplayString (SIZE(15))
```

UNITS "YYYYMMDDHHmmssZ"

MAX-ACCESS read-only

STATUS current

DESCRIPTION "This object returns the date and time of when the RMUs monitoring software was last restarted. This could be from a hardware reset or a soft restart. Z indicates Zulu time or GMT" ::= { rmuMIB 24 }

rmuETLogOnStatus OBJECT-TYPE

```
SYNTAX INTEGER {  
    engineerLoggedOff( 1 ),  
    engineerLoggedOn( 2 ),  
    notSupported( 3 )  
}
```

MAX-ACCESS read-only

STATUS current

DESCRIPTION "This object reveals whether a local engineer is logged onto the RMU as detected by the RMU. For example this could be a login from a local serial handset or from a device on the same network subnet." ::= { rmuMIB 25 }

rmuPrimaryPowerStatus OBJECT-TYPE

```
SYNTAX INTEGER {  
    primarySupplyOk( 1 ),  
    primarySupplyFault( 2 )  
}
```

MAX-ACCESS read-only

STATUS current

DESCRIPTION "This object reveals the status of the RMUs mains power supply this may be mains direct supply or via the controller. If RMU is not able to support this monitoring function or a secondary supply then a primaryPowerOk shall be returned." ::= { rmuMIB 26 }

rmuPrimaryPowerType OBJECT-TYPE

```
SYNTAX INTEGER {  
    mainsDirect( 1 ),  
    mainsViaController( 2 ),  
    otherPrimary( 3 )  
}
```

MAX-ACCESS read-only

STATUS current

DESCRIPTION "This object reveals the status of the RMUs power supply source type." ::= { rmuMIB 27 }

rmuSecondaryPowerStatus OBJECT-TYPE

```
SYNTAX INTEGER {  
    secondarySupplyOk( 1 ),  
    secondarySupplyFault( 2 ),    secondarySupplyNotSupported( 3 )  
}
```

MAX-ACCESS read-only

STATUS current

DESCRIPTION "This object reveals the RMUs Secondary power status. Note that there are options for RMUs that do not have a secondary power status"

::= { rmuMIB 28 }

rmuSecondaryPowerType OBJECT-TYPE

```
SYNTAX INTEGER {  
    rmuBatteryBackup( 1 ),  
    controllerBatteryBackup( 2 ),    alternativeMainsSourceBackup( 3 ),  
    solarBackup( 4 ),  
    otherBackup( 5 ),  
    noBackup( 6 )  
}
```

MAX-ACCESS read-only

STATUS current

DESCRIPTION "This object reveals the status of the RMUs Secondary power supply source type."

::= { rmuMIB 29 }

rmuDeviceCommsStatus OBJECT-TYPE

```
SYNTAX INTEGER {  
    commsOk( 1 ),    commsFault( 2 ),  
    commsNotApplicable( 3 )  
}
```

MAX-ACCESS read-only

STATUS current

DESCRIPTION "This object reveals the status of communications between the RMU and the traffic control equipment being monitoring. This is applicable to where

there is a physical connection e.g. serial or network communications."

::= { rmuMIB 30 }

tlcMIB OBJECT IDENTIFIER ::= { tflRemoteMonitoringHeaderMIB 2 }

tlcStatus OBJECT-TYPE

```
SYNTAX INTEGER {  
    normal( 1 ),  
    fault( 2 )  
}
```

MAX-ACCESS read-only

STATUS current

DESCRIPTION "This object reveals the status of any traffic controller fault. eg which includes general controller, lamp and detector faults."

::= { tlcMIB 1 }

tlcDoorOpen OBJECT-TYPE

```
SYNTAX INTEGER {  
    doorClosed( 1 ),  
    doorOpened( 2 ),  
    notAvailable( 3 )  
}
```

```

}
MAX-ACCESS read-only
STATUS current
DESCRIPTION "This object reveals the current status of controller  door."
 ::= { tlcMIB 2 }

tlcManualPanelOpen OBJECT-TYPE
SYNTAX INTEGER {
    manualPanelClosed( 1 ),
    manualPanelOpened( 2 ),
    notAvailable( 3 )
}
MAX-ACCESS read-only
STATUS current DESCRIPTION "This object reveals the current status of
controller  manual panel."
 ::= { tlcMIB 3 }

tlcDimmingStatus OBJECT-TYPE
SYNTAX INTEGER {
    lampDimmingEnabled( 1 ),
    lampDimmingDisabled( 2 )
}
MAX-ACCESS read-only
STATUS current
DESCRIPTION "This object reveals the status of the traffic control
equipment dimming functionality if enabled." ::= { tlcMIB 4 }

tlcDimmingState OBJECT-TYPE
SYNTAX INTEGER {
    lampsNormal( 1 ),
    lampsDimmed( 2 )
}
MAX-ACCESS read-only
STATUS current
DESCRIPTION "This object reveals the status of a traffic controller
dimming state."
 ::= { tlcMIB 5 }

tlcDimmingfaults OBJECT-TYPE
SYNTAX INTEGER {
    dimmingNormal( 1 ),
    dimmingFault( 2 )
}
MAX-ACCESS read-only
STATUS current
DESCRIPTION "This object reveals a status of traffic controller  dimming
fault."
 ::= { tlcMIB 6 }

tlcStuckStageStatus OBJECT-TYPE
SYNTAX INTEGER {
    normal( 1 ),
    fault( 2 )
}
MAX-ACCESS read-only
STATUS current
DESCRIPTION "This object reveals a status of a stuck stage." ::= { tlcMIB

```

7 }

tlcStuckStage OBJECT-TYPE

SYNTAX Integer32

MAX-ACCESS read-only STATUS current

DESCRIPTION "When there is a stage stuck fault this object shall return the stage number. Otherwise it shall return 0."

::= { tlcMIB 8 }

tlcStreamTable OBJECT-TYPE

SYNTAX SEQUENCE OF TlcStreamEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION "This object contains a list of the Traffic Light Controller Status for each Stream."

::= { tlcMIB 9 }

tlcStreamEntry OBJECT-TYPE

SYNTAX TlcStreamEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION "Each entry identifies the mode, function and status of each stream."

INDEX {

streamIndex

} ::= { tlcStreamTable 1 }

TlcStreamEntry ::= SEQUENCE {

streamIndex

Integer32,

streamMode

INTEGER,

streamModeSubData

DisplayString,

streamFunction

INTEGER,

streamAllOut

INTEGER,

movControlStatus

INTEGER,

movErrorValue

Integer32,

movErrorValueExceeded

INTEGER

}

streamIndex OBJECT-TYPE

SYNTAX Integer32 (1..256)

MAX-ACCESS read-only

STATUS current

DESCRIPTION "This object shall report the traffic stream number, of the traffic controller starting from 1. Note controllers that use stream 0 should reference

1 for stream 0 and 2 for stream 1, etc...."

::= { tlcStreamEntry 1 }

streamMode OBJECT-TYPE SYNTAX INTEGER {

```

unknownMode( 1 ),
signalsOff( 2 ),
startUpMode( 3 ),
fixedModeSimpleOrLinked( 4 ), fixedModeToCurrentMax( 5 ),
vehicleActuated( 6 ), cablelessLinkingNonBaseTime( 7 ),
cablelessLinkingBaseTime( 8 ), partTimeOperation( 9 ),
hurryCall( 10 ),
utcMode( 11 ),
movaMode( 12 ),
scoot( 13 ),
manualMode( 14 ),
manualStepOn( 15 ),
wardenControl( 16 ), publicServiceVehiclePriority( 17 ),
lightRailTransitMode( 18 ), otherApprovedControlMethod( 32 )
}

```

MAX-ACCESS read-only

STATUS current

DESCRIPTION "This object shall report the current method of control, of the traffic controller."

::= { tlcStreamEntry 2 }

streamModeSubData OBJECT-TYPE

SYNTAX DisplayString (SIZE(15))

MAX-ACCESS read-only

STATUS current

DESCRIPTION "This object shall report the mode Sub set of data with regard to the current mode, of the traffic controller. Mode Sub Data will be set at 0 when

not required i.e. UTC, Manual when there is no other mode Sub Data.

Examples: VA Max Set Value 2 FT to current max set Value 3 CLF (Non Base Time) Plan Value 4 MOVA Dataset Value 2.

streamMode | streamModeSubData

INTEGER | DisplayString (SIZE(15))

unknownMode(1) | 0

signalsOff(2) | 0

startUpMode(3) | 0

fixedModeSimpleOrLinked(4) | 0

fixedModeToCurrentMax(5) | Max set running 0=Normal Max 1=1st Alternate Max 2=2nd Alternate Max etc

vehicleActuated(6) | Max set running 0=Normal
Max 1=1st Alternate Max 2=2nd Alternate Max etc

cablelessLinkingNonBaseTime(7) | CLF Plan Number 0=Plan 0 1=Plan 1
2=Plan 2 etc

cablelessLinkingBaseTime(8) | CLF Plan Number 0=Plan 0 1=Plan 1 2=Plan 2 etc

partTimeOperation(9) | 0

hurryCall(10) | Which Hurry Call is currently active (Unit Number)

utcMode(11) | 0

movMode(12) | Active Mova Dataset number (1 to 4)

scoot(13) | 0

manualMode(14) | 0

manualStepOn(15) | 0

wardenControl(16) | 0

publicServiceVehiclePriority(17) | Which Priority is currently active (Unit Number)

lightRailTransitMode(18) | Which LRT is currently active (Unit Number)

otherApprovedControlMethod(32) | 0"
::= { tlcStreamEntry 3 }

streamFunction OBJECT-TYPE

SYNTAX INTEGER {

unknown(1), junctionController(2),
junctionControllerWithPedestrianFacilities(3), pelican(4),
toucan(5),
puffins(6), junctionControllerWithLightRailTransitFacilities(7),
wigwag(8),
pegasus(9),
pedestrianCrossing(10)
}

MAX-ACCESS read-write

STATUS current

DESCRIPTION "This object reveals the controller function for that stream."

::= { tlcStreamEntry 4 }

streamAllOut OBJECT-TYPE

SYNTAX INTEGER {

normal(1),
fail(2)
}

MAX-ACCESS read-only STATUS current

DESCRIPTION "This object is set to fail when the controller detects an all out fault on this stream. The object is set to normal otherwise including when the signals are scheduled to be off."

::= { tlcStreamEntry 5 }

movControlStatus OBJECT-TYPE

SYNTAX INTEGER {

```

    normal( 1 ),
    fault( 2 )
}
MAX-ACCESS read-only
STATUS current
DESCRIPTION "This object shall reveal the MOVA Control fault status bit,
if the notification is set then the object status shall be sent via a
Inform when
the state changes."
::= { tlcStreamEntry 6 }

movErrorValue OBJECT-TYPE SYNTAX Integer32
MAX-ACCESS read-only
STATUS current
DESCRIPTION "This object shall reveal the MOVA Error Value (ERRVAL). The
INTEGER Value is to be set to the MOVA Error Value Example 1, 5, 20, 21,
etc..."
::= { tlcStreamEntry 7 }

movErrorValueExceeded OBJECT-TYPE
SYNTAX INTEGER {
    normalERRVALLessThan20( 1 ), failedERRVALGreaterThanOrEqual20( 2 )
}
MAX-ACCESS read-only
STATUS current
DESCRIPTION "This object shall reveal the overall MOVA enabled state with
regard to the Error Count being less than or more than 20."
::= { tlcStreamEntry 8 }

tlcSpecialConditioningStatus OBJECT-TYPE SYNTAX INTEGER {
    normal( 1 ),
    fault( 2 )
}
MAX-ACCESS read-only
STATUS current
DESCRIPTION "The object reveals the status of the special conditioning
fault flag."
::= { tlcMIB 10 }

tlcLMUStatus OBJECT-TYPE
SYNTAX INTEGER {
    normal( 1 ),
    faultOneOrMoreLMUs( 2 )
}
MAX-ACCESS read-only
STATUS current
DESCRIPTION "This object shall reveal a LMU (Lamp Monitoring Unit) any
checks that if failed trigger the RLM actions or suspend monitoring
failure detection
by the controller,"
::= { tlcMIB 11 }

tlcAllOutStatus OBJECT-TYPE SYNTAX INTEGER {
    normal( 1 ),
    fault( 2 )
}
MAX-ACCESS read-only

```

```

STATUS current
DESCRIPTION "This object reveals the status of a 'signals all out' on the
traffic site."
::= { tlcMIB 12 }

tlcLampStatus OBJECT-TYPE
SYNTAX INTEGER {
    normal( 1 ),
    fault( 2 )
}
MAX-ACCESS read-only
STATUS current
DESCRIPTION "This object reveals the status of any lamp failure." ::= {
tlcMIB 13 }

tlcAnyLampStatusTable OBJECT-TYPE
SYNTAX SEQUENCE OF TlcAnyLampStatusEntry
MAX-ACCESS not-accessible
STATUS current DESCRIPTION "When there is a lamp fault this table exposes
the state of each colour on each phase. The In-station can read the
table to determine which lamps are faulty"
::= { tlcMIB 14 }

tlcAnyLampStatusEntry OBJECT-TYPE
SYNTAX TlcAnyLampStatusEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION "This entry identifies the lamps status, colour
and phase along with either the number of lamps missing or lamp wattage
missing"
INDEX {
    anyPhaseIndex,
    anyColourIndex
}
::= { tlcAnyLampStatusTable 1 }

TlcAnyLampStatusEntry ::= SEQUENCE {
    anyPhaseIndex
        Integer32,
    anyColourIndex Integer32,
    anyPhaseName
        DisplayString,
    anyColourName
        DisplayString,
    anyLampStatus
        INTEGER,
    anyNumberOfFaultyLamps
        Integer32,
    anyLampWattageMissing
        DisplayString
}

anyPhaseIndex OBJECT-TYPE
SYNTAX Integer32 (1..256)
MAX-ACCESS read-only
STATUS current
DESCRIPTION "This object reveals the phase index, which increments from 1

```

onwards each entry is a unique table entry." ::= { tlcAnyLampStatusEntry 1
}

anyColourIndex OBJECT-TYPE

SYNTAX Integer32 (1..256)

MAX-ACCESS read-only

STATUS current

DESCRIPTION "This object reveals the colour index, which increments from
1 onwards each entry is a unique table entry." ::= { tlcAnyLampStatusEntry
2 }

anyPhaseName OBJECT-TYPE

SYNTAX DisplayString

MAX-ACCESS read-only

STATUS current

DESCRIPTION "This object reveals the phase colour identification. For
Example: 'Phase A'"

::= { tlcAnyLampStatusEntry 3 }

anyColourName OBJECT-TYPE

SYNTAX DisplayString

MAX-ACCESS read-only

STATUS current

DESCRIPTION "This object returns the phase colour identification, the
value returned shall be as per the controller specification for each
phase programmed on the controller

Example Red, Amber or Green"

::= { tlcAnyLampStatusEntry 4 }

anyLampStatus OBJECT-TYPE

SYNTAX INTEGER {

normal(1), fault(2)

}

MAX-ACCESS read-only

STATUS current

DESCRIPTION "This is the entry contains the status of the actual phase
affected by a lamp fault."

::= { tlcAnyLampStatusEntry 5 }

anyNumberOfFaultyLamps OBJECT-TYPE

SYNTAX Integer32

MAX-ACCESS read-only

STATUS current

DESCRIPTION "This object returns the number of lamps missing. Note:-
Where manufacturer unable to report this data type a '65536' shall be
returned and anyLampWattageMissing shall be given."

::= { tlcAnyLampStatusEntry 6 }

anyLampWattageMissing OBJECT-TYPE

SYNTAX DisplayString

MAX-ACCESS read-only

STATUS current DESCRIPTION "This object returns the missing wattage per
phase. Example: '0 Watt' shall indicate the phase is
not affected by a lamp fault. Example: '50 Watts' for one lamp missing,
'100 Watts' for two lamps missing, Etc.....

Note: - Where manufacturer is unable to report

this data type a '65536' shall be returned and anyNumberOfFaultyLamps shall be given."

::= { tlcAnyLampStatusEntry 7 }

tlcRedLampStatus OBJECT-TYPE

SYNTAX INTEGER {

normal(1),

fault(2)

}

MAX-ACCESS read-only

STATUS current

DESCRIPTION "This object shall reveal any Red lamp failure detection by the controller."

::= { tlcMIB 15 }

tlcRedLampMonitoringTable OBJECT-TYPE

SYNTAX SEQUENCE OF TlcRedLampMonitoringEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION "When there is a red lamp fault this table exposes the state on each phase. The In-station can read the table to determine which lamps are faulty"

::= { tlcMIB 16 }

tlcRedLampMonitoringEntry OBJECT-TYPE

SYNTAX TlcRedLampMonitoringEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION "This entry identifies the red lamp phase and red lamp status"

INDEX {

redPhaseIndex

}

::= { tlcRedLampMonitoringTable 1 }

TlcRedLampMonitoringEntry ::= SEQUENCE {

redPhaseIndex

Integer32,

redPhaseName

DisplayString,

redLampStatus

INTEGER

}

redPhaseIndex OBJECT-TYPE

SYNTAX Integer32 (1..256)

MAX-ACCESS read-only

STATUS current

DESCRIPTION "This object reveals the phase index, which increments from 1 onwards each entry is a unique table entry." ::= {

tlcRedLampMonitoringEntry 1 }

redPhaseName OBJECT-TYPE

SYNTAX DisplayString

MAX-ACCESS read-only

STATUS current

DESCRIPTION "This object reveals the controller red phase identification. For Example: 'Phase A'"

```
::= { tlcRedLampMonitoringEntry 2 }
```

```
redLampStatus OBJECT-TYPE
```

```
SYNTAX INTEGER { normalNoFaultDetected( 0 ), singleRedLampFailure( 1 ),  
twoOrMoreRedLampFailures( 2 ), redLampFaultUnknown( 3 )  
}
```

```
MAX-ACCESS read-only
```

```
STATUS current
```

```
DESCRIPTION "This object reveals the status of the red lamps  
on this phase. If red lamps are not monitored  
then the value returned will be redLampFaultUnknown." ::= {  
tlcRedLampMonitoringEntry 3 }
```

```
tlcColourConflictStatus OBJECT-TYPE
```

```
SYNTAX INTEGER {  
normal( 1 ),  
fault( 2 )  
}
```

```
MAX-ACCESS read-only
```

```
STATUS current
```

```
DESCRIPTION "This object reveals the status of colour conflict faults as  
detected by the controller.
```

Controller implementations vary. In general this object is enabled when the status of the lamps detected by the controller voltage sensors does not match those requested by the controller.

All controllers support a green-green conflict fault; some provide further faults including absence of requested lamps.

The In-station should read the tlcColourConflictTable to get details of the conflicting and/or absent phases."

```
::= { tlcMIB 17 }
```

```
tlcColourConflictTable OBJECT-TYPE
```

```
SYNTAX SEQUENCE OF TlcColourConflictEntry
```

```
MAX-ACCESS not-accessible STATUS current
```

```
DESCRIPTION "When there is a conflict fault this table exposes the state  
of each colour on each phase. The In-station can read the table to  
determine which lamps conflict." ::= { tlcMIB 18 }
```

```
tlcColourConflictEntry OBJECT-TYPE
```

```
SYNTAX TlcColourConflictEntry
```

```
MAX-ACCESS not-accessible
```

```
STATUS current
```

```
DESCRIPTION "Each identifies the colour conflict status for  
any affect phases"
```

```
INDEX {  
colourConflictPhaseIndex  
}
```

```
::= { tlcColourConflictTable 1 }
```

```
TlcColourConflictEntry ::= SEQUENCE {
```

```
colourConflictPhaseIndex
```

```
Integer32,
```

```
colourConflictPhaseName
```

```
DisplayString,
```

```
greenConflictStatus
```

```
INTEGER,  
amberConflictStatus  INTEGER,  
redConflictStatus  
  INTEGER  
}
```

colourConflictPhaseIndex OBJECT-TYPE

SYNTAX Integer32 (1..256)

MAX-ACCESS read-only

STATUS current

DESCRIPTION "Increment from 1"

::= { tlcColourConflictEntry 1 }

colourConflictPhaseName OBJECT-TYPE

SYNTAX DisplayString

MAX-ACCESS read-only

STATUS current

DESCRIPTION "A textual description that uniquely identifies
the phase e.g. Phase A"

::= { tlcColourConflictEntry 2 }

greenConflictStatus OBJECT-TYPE

SYNTAX INTEGER {

noFault(1),

lampOn(2),

lampConflict(3),

lampAbsent(4)

}

MAX-ACCESS read-only STATUS current

DESCRIPTION "This object reveals the status of the green lamp on this
phase as: - requested by the controller.

- detected by the controller voltage sensors.

There are 3 fault conditions:

lampOn indicates: - another lamp conflicts with
the green on this phase. - the controller requested this phase. - the
controller detected this phase.

lampConflict indicates: - the green on this phase conflicts with another
lamp. - the controller

DID NOT request this phase. - the controller detected this phase.

lampAbsent indicates: - the controller requested this phase. - the
controller DID NOT detect this phase."

::= { tlcColourConflictEntry 3 }

amberConflictStatus OBJECT-TYPE

SYNTAX INTEGER {

noFault(1),

lampOn(2),

lampConflict(3),

lampAbsent(4)

}

MAX-ACCESS read-only

STATUS current

DESCRIPTION "This object reveals the status of the amber lamp on this
phase as: - requested by the controller.

- detected by the controller voltage sensors.

Note: not all controllers support this functionality. In this case the value returned should be noFault.

There are 3 fault conditions:

lampOn indicates: - another lamp conflicts with the amber on this phase. - the controller requested this phase. - the controller detected this phase.

lampConflict indicates: - the amber on this phase conflicts with another lamp. - the controller DID NOT request this phase. - the controller detected this phase.

lampAbsent indicates: - the controller requested this phase. - the controller DID NOT detect this phase."

::= { tlcColourConflictEntry 4 }

redConflictStatus OBJECT-TYPE

SYNTAX INTEGER {

noFault(1),

lampOn(2),

lampConflict(3),

lampAbsent(4)

}

MAX-ACCESS read-only

STATUS current

DESCRIPTION "This object reveals the status of the red lamp

on this phase as: - requested by the controller.

- detected by the controller voltage sensors.

Note: not all controllers support this functionality. In this case the value returned should be noFault.

There are 3 fault conditions:

lampOn indicates: - another lamp conflicts with the red on this phase. - the controller requested this phase. - the controller detected this phase.

lampConflict indicates: - the red on this phase conflicts with another lamp. - the controller

DID NOT request this phase. - the controller detected this phase.

lampAbsent indicates: - the controller requested this phase. - the controller DID NOT detect this phase."

::= { tlcColourConflictEntry 5 }

tlcDetectorFault OBJECT-TYPE

SYNTAX INTEGER {

normal(1),

fault(2)

}

MAX-ACCESS read-only

STATUS current DESCRIPTION "This object reveals the status of any general detector fault and is only cleared when all detectors are operational and cleared on the controller

i.e. DFM cleared."
 ::= { tlcMIB 19 }

tlcDetectorStatusTable OBJECT-TYPE
SYNTAX SEQUENCE OF TlcDetectorStatusEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION "This table contains a detailed list of the detector fault status providing the Authority with information on which detector has a fault, which detector has failed and its fault type."
 ::= { tlcMIB 20 }

tlcDetectorStatusEntry OBJECT-TYPE
SYNTAX TlcDetectorStatusEntry
MAX-ACCESS not-accessible
STATUS current DESCRIPTION "Each entry identifies the name type and status of each detector"
INDEX {
 detectorIndex
}
 ::= { tlcDetectorStatusTable 1 }

TlcDetectorStatusEntry ::= SEQUENCE {
 detectorIndex
 Integer32,
 detectorName
 DisplayString,
 detectorSetting
 INTEGER,
 detectorStatus
 INTEGER,
 detectorType
 DisplayString
}

detectorIndex OBJECT-TYPE
SYNTAX Integer32 (1..256)
MAX-ACCESS read-only
STATUS current
DESCRIPTION "This object returns detector index value which shall start from 1"
 ::= { tlcDetectorStatusEntry 1 }

detectorName OBJECT-TYPE
SYNTAX DisplayString
MAX-ACCESS read-only STATUS current
DESCRIPTION "This object returns the detector name, the value returned shall information below, For example 'NVD', 'PBU', ETC....."
 ::= { tlcDetectorStatusEntry 2 }

detectorSetting OBJECT-TYPE
SYNTAX INTEGER {
 normal(1),
 inhibited(2)
}
MAX-ACCESS read-only

```

STATUS current
DESCRIPTION "This object reveals the individual detector control setting
within the controller."
::= { tlcDetectorStatusEntry 3 }

detectorStatus OBJECT-TYPE
SYNTAX INTEGER {
    normal( 1 ),
    permanentDetect( 2 ),
    permanentNoDetect( 3 )
} MAX-ACCESS read-only
STATUS current
DESCRIPTION "This object reveals the individual detector fault type.
1,normal is reported if the detection is responding to activations
within the detector timing thresholds. 2, permanentDetect is reported if
the detector
is in this state and exceeds the timing threshold. 3, noPermanentDetect
is reported if the detector
is in this state and exceeds the timing threshold." ::= {
tlcDetectorStatusEntry 4 }

detectorType OBJECT-TYPE
SYNTAX DisplayString
MAX-ACCESS read-write
STATUS current
DESCRIPTION "This object returns the Detector Type Name example given
below this field is readable and writeable.

Note1: This is intended for the operational team
to set these parameters.

Examples:- 'MVD' 'PCaTS' 'Infra Red Detector'
'Push Button' 'Speed detector' 'XYZ loop' 'Bus Priority Detector' 'Hurry
call' 'Other' 'Loops' 'Magnetometer' 'Vehicle Video image Detection'
'Pedestrian Detector' 'Pedestrian Video Detector' 'Not Defined' 'Not Set'
Etc....

Note2: If the detector has been set to inhibit
then the description of the detector and reason
can be set here."
::= { tlcDetectorStatusEntry 5 }

tlcFaultLogAlarm OBJECT-TYPE SYNTAX INTEGER {
    noFaultInLog( 1 ),
    faultInLog( 2 )
}
MAX-ACCESS read-only
STATUS current
DESCRIPTION "This object reveals the status of the controller fault log
alarm, this indicates there is a fault within the controller fault log."
::= { tlcMIB 21 }

tlcRegSignStatus OBJECT-TYPE
SYNTAX INTEGER {
    regSignsOK( 1 ),
    regSignsFaultDetected( 2 )
}

```

```

MAX-ACCESS read-only
STATUS current
DESCRIPTION "This object shall reveal any Reg Sign failure detection by
the controller."
::= { tlcMIB 22 }

tlcRegSignTable OBJECT-TYPE SYNTAX SEQUENCE OF TlcRegSignEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION "This table reveals in detail the status of Regualtory Signs"
::= { tlcMIB 23 }

tlcRegSignEntry OBJECT-TYPE
SYNTAX TlcRegSignEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION "This entry identifies the reg sign status and
a description field which allows the manufacturer to identify the
external phase / sensor ..etc..." INDEX {
    regSignIndex
}
::= { tlcRegSignTable 1 }

TlcRegSignEntry ::= SEQUENCE {
    regSignIndex
        Integer32,
    regSignMonitorType
        DisplayString,
    regSignStatus
        INTEGER
}
    regSignIndex OBJECT-TYPE
SYNTAX Integer32 (1..256)
MAX-ACCESS read-only
STATUS current
DESCRIPTION "This object returns the reg sign index value which shall
start from 1"
::= { tlcRegSignEntry 1 }

regSignMonitorType OBJECT-TYPE
SYNTAX DisplayString
MAX-ACCESS read-write
STATUS current
DESCRIPTION "This object reveals the identity of the monitored
phase/sensor. This is a textual description field which provides the
manufacturer to inputa unique description. ie phase,sensor, etc... of the
monitored Reg Sign."
::= { tlcRegSignEntry 2 }

regSignStatus OBJECT-TYPE
SYNTAX INTEGER {
    regSignNormal( 1 ), regSignFault( 2 )
}
MAX-ACCESS read-only
STATUS current
DESCRIPTION "This reveals the status of the Reg Sign/s monitored with
respect to the Phase/Sensor..etc"

```

::= { tlcRegSignEntry 3 }

ohvdMIB OBJECT IDENTIFIER ::= { tflRemoteMonitoringHeaderMIB 3 }

ohvdManufacturerName OBJECT-TYPE

SYNTAX DisplayString (SIZE(255))

MAX-ACCESS read-only

STATUS current

DESCRIPTION "This object reveals the manufacturer of the OHVD. This is a textual description."

::= { ohvdMIB 1 }

ohvdEquipmentType OBJECT-TYPE

SYNTAX DisplayString (SIZE(255))

MAX-ACCESS read-only

STATUS current

DESCRIPTION "This object reveals the equipment type identity of the on street equipment. This is a textual description which provides the authority with

the manufacturers Equipment Type of the particular OHVD installed."

::= { ohvdMIB 2 }

ohvdSystemStatus OBJECT-TYPE

SYNTAX INTEGER {

normal(1),

fault(2)

}

MAX-ACCESS read-only

STATUS current

DESCRIPTION "This object shall reveals the system status of the OHVD."

::= { ohvdMIB 3 }

ohvdPowerFailure OBJECT-TYPE

SYNTAX INTEGER {

normal(1),

fault(2)

}

MAX-ACCESS read-only

STATUS current DESCRIPTION "This object reveal the power status of the OHVD system."

::= { ohvdMIB 4 }

ohvdHeaterFailure OBJECT-TYPE

SYNTAX INTEGER {

normal(1),

fault(2)

}

MAX-ACCESS read-only

STATUS current

DESCRIPTION "This object reveals the heater status of the OHVD system"

::= { ohvdMIB 5 }

ohvdFailureNonCritical OBJECT-TYPE

SYNTAX INTEGER {

normal(1),

fault(2)

```

    }
MAX-ACCESS read-only
STATUS current
DESCRIPTION "This object reveals a Non Critical Failure by
    the OHVD system."
::= { ohvdMIB 6 }

ohvdCriticalFail OBJECT-TYPE
SYNTAX INTEGER {
    normal( 1 ),
    fault( 2 )
}
MAX-ACCESS read-only
STATUS current
DESCRIPTION "This object reveals a Critical Failure by the
    OHVD system."
::= { ohvdMIB 7 }

ohvdBeamFault OBJECT-TYPE
SYNTAX INTEGER {
    normal( 1 ),
    fault( 2 )
}
MAX-ACCESS read-only
STATUS current
DESCRIPTION "This object reveals thestatus of the OHVD beams." ::= {
ohvdMIB 8 }

ohvdBeamStatusTable OBJECT-TYPE
SYNTAX SEQUENCE OF OhvdBeamStatusEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION "This table shall reveal the OHVD Beam faults for  each
individual beam"
::= { ohvdMIB 11 }

ohvdBeamStatusEntry OBJECT-TYPE
SYNTAX OhvdBeamStatusEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION "This is the entry contains the beam status of
    each beam"
INDEX {
    beamIndex
}
::= { ohvdBeamStatusTable 1 }

OhvdBeamStatusEntry ::= SEQUENCE {
    beamIndex
        Integer32,
    beamNumber
        DisplayString,
    beamStatus
        INTEGER
}

beamIndex OBJECT-TYPE

```

SYNTAX Integer32 (1..16)

MAX-ACCESS read-only

STATUS current

DESCRIPTION "This object is an index number which increments from 1 onwards each entry is unique table entry" ::= { ohvdBeamStatusEntry 1 }

beamNumber OBJECT-TYPE

SYNTAX DisplayString

MAX-ACCESS read-only

STATUS current

DESCRIPTION "This object reveals the Beam number this shall
OID shall support up to 16 beams. Each beam shall be uniquely identified
i.e. 'Beam 1', 'Beam 2', 'Beam 3', etc.... 'Beam 16'"
::= { ohvdBeamStatusEntry 2 }

beamStatus OBJECT-TYPE

SYNTAX INTEGER {

normal(1),

fault(2)

}

MAX-ACCESS read-only

STATUS current

DESCRIPTION "This object reveals the beam fault status for
each beam."

::= { ohvdBeamStatusEntry 3 }

ohvdDetectorFault OBJECT-TYPE

SYNTAX INTEGER { normal(1),

fault(2)

}

MAX-ACCESS read-only

STATUS current

DESCRIPTION "This object reveals the status of any detector." ::= {
ohvdMIB 12 }

ohvdDetectorStatusTable OBJECT-TYPE

SYNTAX SEQUENCE OF OhvdDetectorStatusEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION "This Object shall reveal the status of each detector in a
Table format"

::= { ohvdMIB 13 }

ohvdDetectorStatusEntry OBJECT-TYPE

SYNTAX OhvdDetectorStatusEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION "This is the Entry containing the details of all the Loop
number, Fault type and fault status of each loop the table shall support
up to 16

loops"

INDEX {

ohvdDetIndex

}

::= { ohvdDetectorStatusTable 1 }

OhvdDetectorStatusEntry ::= SEQUENCE {

```
ohvdDetIndex
    Integer32,
ohvdDetNumber
    DisplayString,
ohvdDetFaultType
    INTEGER,
ohvdDetStatus
    INTEGER
}
```

```
ohvdDetIndex OBJECT-TYPE
SYNTAX Integer32 (1..16)
MAX-ACCESS read-only
STATUS current
DESCRIPTION "This object is an index number which increments from 1
onwards each entry is unique table entry" ::= { ohvdDetectorStatusEntry 1
}
```

```
ohvdDetNumber OBJECT-TYPE
SYNTAX DisplayString
MAX-ACCESS read-only
STATUS current DESCRIPTION "This object reveals the detector number this
shall OID shall support up to 16 detectors"
::= { ohvdDetectorStatusEntry 2 }
```

```
ohvdDetFaultType OBJECT-TYPE
SYNTAX INTEGER {
    normal( 1 ),
    loopPermanentDetect( 2 ), loopPermanentNoDetect( 3 )
}
MAX-ACCESS read-only
STATUS current
DESCRIPTION "This object shall reveals the fault type of each detector"
::= { ohvdDetectorStatusEntry 3 }
```

```
ohvdDetStatus OBJECT-TYPE
SYNTAX INTEGER {
    loopNormal( 1 ),
    loopFaulty( 2 )
}
MAX-ACCESS read-only
STATUS current DESCRIPTION "This object shall reveal the Loop status when
the permanent detect or no detect is set in OID ohvdDetFaultType."
::= { ohvdDetectorStatusEntry 4 }
```

```
ohvdSignFault OBJECT-TYPE
SYNTAX INTEGER {
    normal( 1 ),
    fault( 2 )
}
MAX-ACCESS read-only
STATUS current
DESCRIPTION "This object reveal the status of any sign on the OHVD
system."
::= { ohvdMIB 14 }
```

```
ohvdSignStatusTable OBJECT-TYPE
```

SYNTAX SEQUENCE OF OhvdSignStatusEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION "This table reveals the status of each sign."

::= { ohvdMIB 15 }

ohvdSignStatusEntry OBJECT-TYPE SYNTAX OhvdSignStatusEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION "This is the entry contains the details of each sign and its status."

INDEX {

signIndex

}

::= { ohvdSignStatusTable 1 }

OhvdSignStatusEntry ::= SEQUENCE {

signIndex

Integer32,

signNumber

INTEGER,

signDimmerStatus

INTEGER,

signCommStatus

INTEGER,

signStatus

INTEGER

}

signIndex OBJECT-TYPE

SYNTAX Integer32 (1..16)

MAX-ACCESS read-only

STATUS current

DESCRIPTION "This object is an index number which increments from 1 onwards each entry is unique table entry." ::= { ohvdSignStatusEntry 1 }

signNumber OBJECT-TYPE

SYNTAX INTEGER {

sign1(1),

sign2(2),

sign3(3),

sign4(4),

sign5(5),

sign6(6),

sign7(7),

sign8(8),

sign9(9),

sign10(10),

sign11(11),

sign12(12),

sign13(13),

sign14(14),

sign15(15),

sign16(16)

}

MAX-ACCESS read-only

STATUS current

DESCRIPTION "This object reveals the sign number this shall OID shall

support up to 16 signs"
::= { ohvdSignStatusEntry 2 }

signDimmerStatus OBJECT-TYPE

SYNTAX INTEGER {
 dimmerNormal(1),
 dimmed(2),
 dimmerFaulty(3)
}

MAX-ACCESS read-only

STATUS current

DESCRIPTION "This object reveals the status of a dimming to
the sign, including a fault condition."

::= { ohvdSignStatusEntry 3 }

signCommStatus OBJECT-TYPE

SYNTAX INTEGER {
 commsNormal(1),
 commsFaulty(2)
}

MAX-ACCESS read-only

STATUS current

DESCRIPTION "This object hall reveals a sign to controller communications
fault status."

::= { ohvdSignStatusEntry 4 }

signStatus OBJECT-TYPE

SYNTAX INTEGER {
 permanentDetect(1),
 noDetect(2)
}

MAX-ACCESS read-only

STATUS current

DESCRIPTION "This object reveals the sign fault status type." ::= {
ohvdSignStatusEntry 5 }

ohvdEventLogStatus OBJECT-TYPE

SYNTAX INTEGER {
 lessThan75PercentFull(1), greaterThanOrEqual75PercentFull(2)
}

MAX-ACCESS read-only

STATUS current

DESCRIPTION "This object reveals the OHVD Log storage capacity." ::= {
ohvdMIB 16 }

ohvdEventLogTable OBJECT-TYPE

SYNTAX SEQUENCE OF OhvdEventLogEntry MAX-ACCESS not-accessible

STATUS current

DESCRIPTION "This Object contains a detailed list of the Over Height
Vehicle Detections. The start of the table shall show the newest log
entry first follow

by every subsequent entry."

::= { ohvdMIB 17 }

ohvdEventLogEntry OBJECT-TYPE

SYNTAX OhvdEventLogEntry

MAX-ACCESS not-accessible

```

STATUS current
DESCRIPTION "This is the Entry containing the details of all the Data and
times of each Over Height Vehicle Detection"
INDEX {
    ohvdIndex
}
 ::= { ohvdEventLogTable 1 }

OhvdEventLogEntry ::= SEQUENCE {
    ohvdIndex
        Integer32,
    ohvdDateTime DisplayString
}

ohvdIndex OBJECT-TYPE
SYNTAX Integer32 (1..65536)
MAX-ACCESS read-only
STATUS current
DESCRIPTION "This object is a index number which increments from 1
onwards each entry is unique table entry." ::= { ohvdEventLogEntry 1 }

ohvdDateTime OBJECT-TYPE
SYNTAX DisplayString (SIZE(15))
UNITS "YYMMDDHHmmssZ"
MAX-ACCESS read-only
STATUS current
DESCRIPTION "This object reveals the date and time in Numerical format of
the Over Height Vehicle Detection activations

    Example: Year / Month/ Day / Hrs / Mins / Secs
    / Zulu (YYMMDDHHmmssZ) 140626120000Z"
 ::= { ohvdEventLogEntry 2 }

ohvdEventLogResetStatus OBJECT-TYPE SYNTAX INTEGER {
    logCleared( 1 ),
    logResetInProgress( 2 ),
    resetLog( 3 ),
    errorInLog( 4 )
}
MAX-ACCESS read-write
STATUS current
DESCRIPTION "The Event log may be cleared by setting the resetLog, while
the log is being cleared thelogResetInProgress shall be reported and once
the log is cleared
    the logCleared shall be reported. If an error occurs a errorInLog shall
be reported. otherwise logCleared should be reported."
 ::= { ohvdMIB 18 }

rmuHeaderMIBConformance OBJECT IDENTIFIER ::= {
    tflRemoteMonitoringHeaderMIB 4 }

rmuHeaderMIBGroups OBJECT IDENTIFIER ::= { rmuHeaderMIBConformance 1 }
    rmuAllObjects OBJECT-GROUP
OBJECTS {
    rmuManufacturer,
    rmuEquipmentType,
    rmuIPAddress,

```

rmuGateway,
rmuBroadcastAddress,
rmuNetmask,
rmuDNSServerPrimary,
rmuDNSServerSecondary,
rmuDHCP,
rmuTimeOfDay,
rmuSystemSyncMethod,
rmuConfigSiteID,
rmuConfigSiteAddress,
rmuConfigInStationAddress,
rmuConfigNTPAddress1,
rmuConfigNTPAddress2,
rmuConfigInformTableUpdated,
rmuConfigInputBits,
rmuConfigOutputBits,
rmuLastResetStatus,
rmuETLogOnStatus, rmuPrimaryPowerStatus,
rmuPrimaryPowerType,
rmuSecondaryPowerStatus,
rmuSecondaryPowerType,
rmuDeviceCommsStatus,
tlcStatus,
tlcDoorOpen,
tlcManualPanelOpen,
tlcDimmingStatus,
tlcDimmingState,
tlcDimmingfaults,
tlcStuckStageStatus,
tlcStuckStage,
tlcSpecialConditioningStatus,
tlcLMUStatus,
tlcAllOutStatus,
tlcLampStatus,
tlcRedLampStatus,
tlcColourConflictStatus,
tlcDetectorFault,
tlcFaultLogAlarm,
tlcRegSignStatus,
ohvdManufacturerName, ohvdEquipmentType,
ohvdSystemStatus,
ohvdPowerFailure,
ohvdHeaterFailure,
ohvdFailureNonCritical,
ohvdCriticalFail,
ohvdBeamFault,
ohvdDetectorFault,
ohvdSignFault,
ohvdEventLogStatus,
ohvdEventLogResetStatus,
rmuFSIndex,
rmuFSModule,
rmuFSDescription,
rmuFSModuleBuildno,
rmuFSMD5Checksum,
configInformIndex,
configInformIdentifier,

```

configInformEnableStatus,
inputBitIndex,
inputBitDescription,
inputSetInform,
inputBitPolarity,  inputState,
outputBitIndex,
outputBitDescription,
outputSetInform,
outputBitPolarity,
outputState,
streamIndex,
streamMode,
streamModeSubData,
streamFunction,
streamAllOut,
movaControlStatus,
movaErrorValue,
movaErrorValueExceeded,
anyPhaseIndex,
anyColourIndex,
anyPhaseName,
anyColourName,
anyLampStatus,
anyNumberOfFaultyLamps,
anyLampWattageMissing,
redPhaseIndex,
redPhaseName,
redLampStatus,  colourConflictPhaseIndex,
colourConflictPhaseName,
greenConflictStatus,
amberConflictStatus,
redConflictStatus,
detectorIndex,
detectorName,
detectorStatus,
detectorType,
regSignIndex,
regSignMonitorType,
regSignStatus,
beamIndex,
beamNumber,
beamStatus,
ohvdDetIndex,
ohvdDetNumber,
ohvdDetFaultType,
ohvdDetStatus,
signIndex,
signNumber,
signDimmerStatus,
signCommStatus,
signStatus,
ohvdIndex,  ohvdDateTime,
detectorSetting
}

```

STATUS current

DESCRIPTION "This automatically created object group contains all those objects that do not belong to any other OBJECT-GROUP"

```
::= { rmuHeaderMIBGroups 1 }
```

```
rmuAllNotifications NOTIFICATION-GROUP
```

```
NOTIFICATIONS {
```

```
    rmuFSEvent,  
    rmuConfigEvent,  
    rmuInputStateEvent,  
    rmuOutputStateEvent,  
    rmuResetEvent,  
    rmuETLogEvent,  
    rmuSecondaryPowerStatusEvent, rmuPrimaryPowerStatusEvent,  
    rmuDeviceCommsEvent,  
    tlcStatusEvent, tlcDoorStatusEvent,  
    tlcDimmingStatusEvent,  
    tlcStuckStageEvent,  
    tlcAllOutOnAStreamEvent,  
    tlcMOVASStatusEvent,  
    tlcStreamEvent, tlcSpecialConditioningStatusEvent,  
    tlcLMUEvent,  
    tlcAllOutEvent,  
    tlcLampEvent,  
    tlcAnyLampTableEvent,  
    tlcRedLampEvent,  
    tlcRedLampTableEvent,  
    tlcDetectorFaultEvent,  
    tlcDetectorTableEvent,  
    tlcFaultLogAlarmEvent,  
    tlcColourConflictEvent,  
    tlcRegSignEvent,  
    tlcRegSignTableEvent, ohvdSystemEvent,  
    ohvdPowerEvent,  
    ohvdHeaterEvent,  
    ohvdNonCriticalEvent,  
    ohvdCriticalEvent,  
    ohvdBeamEvent,  
    ohvdBeamTableEvent,  
    ohvdDetectorEvent,  
    ohvdDetectorTableEvent,  
    ohvdSignEvent,  
    ohvdSignTableEvent,  
    ohvdLogFullEvent,  
    ohvdLogResetEvent  
}
```

```
STATUS current
```

```
DESCRIPTION "This automatically created notification group contains all  
notifications that do not belong  
to any other NOTIFICATION-GROUP"
```

```
::= { rmuHeaderMIBGroups 2 }
```

```
rmNotifications OBJECT-IDENTITY
```

```
STATUS current
```

```
DESCRIPTION "This OID defines the Notifications for the RMU MIB."
```

```
::= { tflRemoteMonitoringHeaderMIB 5 }
```

```
rmuNotifications OBJECT-IDENTITY
```

```
STATUS current
```

```
DESCRIPTION "Notifications for RMU"
```

```
::= { rmNotifications 1 }
```

rmuFSEvent NOTIFICATION-TYPE

OBJECTS {
 rmuFSIndex,
 rmuFSModule,
 rmuFSModuleBuildno
}

STATUS current

DESCRIPTION "This notification is sent when the firmware/ software is updated the following items shall be returned within the table rmuFSIndex
- Number in the table rmuFSModule - Module Description rmuFSModuleBuildNo
- Module Build No or version number"
::= { rmuNotifications 1 }

rmuConfigEvent NOTIFICATION-TYPE

OBJECTS {
 rmuConfigInformTableUpdated
}

STATUS current

DESCRIPTION "This Notification indicates the configuration INFORM table has been changed this is sent when
an INFORM has been enabled or disabled"
::= { rmuNotifications 2 }

rmuInputStateEvent NOTIFICATION-TYPE

OBJECTS {
 inputBitIndex,
 inputBitDescription,
 inputState
}

STATUS current

DESCRIPTION "This notification is sent if the input state polarity changes."
::= { rmuNotifications 3 }

rmuOutputStateEvent NOTIFICATION-TYPE

OBJECTS {
 outputBitIndex,
 outputBitDescription,
 outputState
}

STATUS current

DESCRIPTION "This notification is sent if the output state polarity changes"
::= { rmuNotifications 4 }

rmuResetEvent NOTIFICATION-TYPE

OBJECTS {
 rmuLastResetStatus
}

STATUS current

DESCRIPTION "This notification is sent when the last rest of the rmu occurred or when the power was recycled
to the unit"
::= { rmuNotifications 5 }

rmuETLogEvent NOTIFICATION-TYPE

OBJECTS {

```

    rmuETLogOnStatus
  }
STATUS current
DESCRIPTION "This notification is sent if the engineers terminal is
logged on or logged OFF or ON"
::= { rmuNotifications 6 }

rmuSecondaryPowerStatusEvent NOTIFICATION-TYPE
OBJECTS {
  rmuSecondaryPowerStatus
}
STATUS current
DESCRIPTION "This notification is returned if the battery status changes
and should only be sent if a battery
back or other alternative power is provided."
::= { rmuNotifications 7 }

rmuPrimaryPowerStatusEvent NOTIFICATION-TYPE
OBJECTS {
  rmuPrimaryPowerStatus
}
STATUS current
DESCRIPTION "This notification is sent if a change in the rmu mains power
supply occurs. Note that this is
only possible if the RMU has an alternative power supply or backup
battery."
::= { rmuNotifications 8 }

rmuDeviceCommsEvent NOTIFICATION-TYPE
OBJECTS {
  rmuDeviceCommsStatus
}
STATUS current
DESCRIPTION "This notification is sent when the rmu device Communications
to the traffic control equipment
is either connected or disconnected. If the rmu
is integral to the traffic control equipment
then this notification is not applicable."
::= { rmuNotifications 9 }

tlcNotifications OBJECT-IDENTITY
STATUS current
DESCRIPTION "Notifications generated from the traffic Light Controller"
::= { rmNotifications 2 }

tlcStatusEvent NOTIFICATION-TYPE
OBJECTS {
  tlcStatus }
STATUS current
DESCRIPTION "This notification is return if the tlc status changes. i.e.
when a fault occurs or when the
fault is cleared."
::= { tlcNotifications 1 }

tlcDoorStatusEvent NOTIFICATION-TYPE
OBJECTS {
  tlcDoorOpen,

```

```

    tlcManualPanelOpen
  }
STATUS current
DESCRIPTION "This notification is returned if the main door
  or manual panel is opened and only if supported
  by the controller."
::= { tlcNotifications 2 }

tlcDimmingStatusEvent NOTIFICATION-TYPE
OBJECTS {
  tlcDimmingState,
  tlcDimmingStatus,
  tlcDimmingfaults
}
STATUS current DESCRIPTION "This notification shall be return if the
tlcDimmingState is enabled and the tlcDimmingfault reports a
  fault."
::= { tlcNotifications 3 }

tlcStuckStageEvent NOTIFICATION-TYPE
OBJECTS {
  tlcStuckStageStatus,
  tlcStuckStage
}
STATUS current
DESCRIPTION "This notification is returned when the stuck stage status is
in fault and also the actual stuck
  stage shall be returned."
::= { tlcNotifications 5 }

tlcAllOutOnAStreamEvent NOTIFICATION-TYPE
OBJECTS {
  streamIndex,
  streamAllOut
}
STATUS current
DESCRIPTION "This notification is return when a signal 'All Out' is
detected"
::= { tlcNotifications 6 }

tlcMOVAStatusEvent NOTIFICATION-TYPE
OBJECTS {
  streamIndex,
  movaControlStatus,
  movaErrorValue,
  movaErrorValueExceeded
}
STATUS current
DESCRIPTION "This notification is sent when there is a MOVA fault
condition, the notification shall also
  return the MOVA error value exceeded and the
  actual value according to each Stream."
::= { tlcNotifications 7 }

tlcStreamEvent NOTIFICATION-TYPE
OBJECTS {
  streamIndex,

```

```

    streamMode,
    streamModeSubData,
    streamFunction
}
STATUS current DESCRIPTION "This notification reveals additional stream
event data. For example the mode of operation, operational functions to
the current method of control and plans related to each stream."
::= { tlcNotifications 8 }

tlcSpecialConditioningStatusEvent NOTIFICATION-TYPE
OBJECTS {
    tlcSpecialConditioningStatus
}
STATUS current
DESCRIPTION "This notification is returned when the special conditioning
status is at fault."
::= { tlcNotifications 9 }

tlcLMUEvent NOTIFICATION-TYPE
OBJECTS {
    tlcLMUStatus
}
STATUS current
DESCRIPTION "This notification is returned when the Lamp Monitor Unit/s
is in fault." ::= { tlcNotifications 10 }

tlcAllOutEvent NOTIFICATION-TYPE
OBJECTS {
    tlcAllOutStatus
}
STATUS current
DESCRIPTION "This notification is returned when there is a
'All Out' detected."
::= { tlcNotifications 11 }

tlcLampEvent NOTIFICATION-TYPE
OBJECTS {
    tlcLampStatus
}
STATUS current
DESCRIPTION "This notification is returned when any lamp out
is detected."
::= { tlcNotifications 12 }

tlcAnyLampTableEvent NOTIFICATION-TYPE
OBJECTS {
    anyPhaseName,
    anyColourName,
    anyLampStatus,
    anyPhaseIndex,
    anyColourIndex
}
STATUS current DESCRIPTION "This notification is returned when any lamp
out
is detected. It details the number of fault lamps or the missing wattage
and there corresponding phase and colour"
::= { tlcNotifications 13 }

```

tlcRedLampEvent NOTIFICATION-TYPE

OBJECTS {
 tlcRedLampStatus
}

STATUS current

DESCRIPTION "This notification is returned when a Red lamp
out is detected."

::= { tlcNotifications 14 }

tlcRedLampTableEvent NOTIFICATION-TYPE

OBJECTS {
 redPhaseIndex,
 redPhaseName,
 redLampStatus
}

STATUS current

DESCRIPTION "This notification is returned when a Red lamp out is
detected. It details the phase and status i.e. 1st Red lamp or second Red
lamp out conditions." ::= { tlcNotifications 15 }

tlcDetectorFaultEvent NOTIFICATION-TYPE

OBJECTS {
 tlcDetectorFault
}

STATUS current

DESCRIPTION "This notification is returned when a detector fault
condition is detected."

::= { tlcNotifications 19 }

tlcDetectorTableEvent NOTIFICATION-TYPE

OBJECTS {
 detectorIndex,
 detectorName,
 detectorStatus
}

STATUS current

DESCRIPTION "This notification is returned when a detector fault is
detected. It details the detector name
and its status of the fault condition type."

::= { tlcNotifications 20 }

tlcFaultLogAlarmEvent NOTIFICATION-TYPE

OBJECTS {
 tlcFaultLogAlarm
}

STATUS current

DESCRIPTION "This notification is returned when there is a status is the
controller fault log."

::= { tlcNotifications 21 }

tlcColourConflictEvent NOTIFICATION-TYPE

OBJECTS {
 colourConflictPhaseIndex,
 colourConflictPhaseName,
 greenConflictStatus,
 tlcColourConflictStatus
}

```

STATUS current
DESCRIPTION "The RMU sends this notification on change of the
tclColourConflictStatus. It must include the
first phase where greenConflictStatus is not noFault."
::= { tlcNotifications 22 }

tlcRegSignEvent NOTIFICATION-TYPE
OBJECTS {
    tlcRegSignStatus
}
STATUS current
DESCRIPTION "This notification is returned when any Reg Sign lamp out is
detected."
::= { tlcNotifications 24 }

tlcRegSignTableEvent NOTIFICATION-TYPE
OBJECTS {
    regSignIndex,
    regSignMonitorType,
    regSignStatus
}
STATUS current
DESCRIPTION "Please replace this text with your description." ::= {
tlcNotifications 25 }

ohvdNotifications OBJECT-IDENTITY
STATUS current
DESCRIPTION "Notifications for Over Height Vehicle Detector" ::= {
rmNotifications 3 }

ohvdSystemEvent NOTIFICATION-TYPE
OBJECTS {
    ohvdSystemStatus }
STATUS current
DESCRIPTION "This notification is return if the ohvd system status
changes. i.e. when a fault occurs or when the fault is cleared."
::= { ohvdNotifications 1 }

ohvdPowerEvent NOTIFICATION-TYPE
OBJECTS {
    ohvdPowerFailure
}
STATUS current
DESCRIPTION "This notification is sent if a change in the ohvd power.
Note this message can only be send if
there is an alternative power supply."
::= { ohvdNotifications 2 }

ohvdHeaterEvent NOTIFICATION-TYPE
OBJECTS {
    ohvdHeaterFailure
}
STATUS current
DESCRIPTION "This notification is sent if there is a fault condition
detected with the heater."
::= { ohvdNotifications 3 }

```

ohvdNonCriticalEvent NOTIFICATION-TYPE

OBJECTS {
 ohvdFailureNonCritical
}

STATUS current

DESCRIPTION "This notification is sent if there is a non critical event
fault condition detected."

::= { ohvdNotifications 4 }

ohvdCriticalEvent NOTIFICATION-TYPE

OBJECTS {
 ohvdCriticalFail
}

STATUS current

DESCRIPTION "This notification is sent if there is a critical event fault
condition detected."

::= { ohvdNotifications 5 }

ohvdBeamEvent NOTIFICATION-TYPE

OBJECTS {
 ohvdBeamFault
}

STATUS current DESCRIPTION "This notification is sent if there is any beam
fault condition detected. The following notification should be walked
ohvdBeamTableEvent to reveal
which beams are faulty."

::= { ohvdNotifications 6 }

ohvdBeamTableEvent NOTIFICATION-TYPE

OBJECTS {
 beamIndex,
 beamNumber,
 beamStatus
}

STATUS current

DESCRIPTION "This notification is sent if there is beam fault condition
detected. The beam number identity
number and its status is returned in a table format."

::= { ohvdNotifications 7 }

ohvdDetectorEvent NOTIFICATION-TYPE

OBJECTS {
 ohvdDetectorFault
} STATUS current

DESCRIPTION "This notification is sent if there is any detector (ie.
Loop) condition detected. The following notification should be walked
ohvdDetectorTableEvent to reveal which detectors are faulty."

::= { ohvdNotifications 8 }

ohvdDetectorTableEvent NOTIFICATION-TYPE

OBJECTS {
 ohvdDetIndex,
 ohvdDetNumber,
 ohvdDetStatus
}

STATUS current

DESCRIPTION "This notification is sent if there is detector fault

```

condition detected. The beam number identity number and its status is
returned in a table format."
::= { ohvdNotifications 9 }

ohvdSignEvent NOTIFICATION-TYPE
OBJECTS { ohvdSignFault
}
STATUS current
DESCRIPTION "This notification is sent if there is any sign fault
condition detected. The following notification should be walked
ohvdSignTableEvent to reveal
which detectors are faulty."
::= { ohvdNotifications 10 }

ohvdSignTableEvent NOTIFICATION-TYPE
OBJECTS {
signIndex,
signNumber,
signDimmerStatus,
signCommStatus,
signStatus
}
STATUS current
DESCRIPTION "This notification is sent if there is sign fault condition
detected. The sign number identity number, its status is returned in a
table format." ::= { ohvdNotifications 11 }
ohvdLogFullEvent NOTIFICATION-TYPE
OBJECTS {
ohvdEventLogStatus
}
STATUS current
DESCRIPTION "This notification is sent if the log is greater than 75%
full is detected."
::= { ohvdNotifications 14 }

ohvdLogResetEvent NOTIFICATION-TYPE
OBJECTS {
ohvdEventLogResetStatus
}
STATUS current
DESCRIPTION "Please replace this text with your description." ::= {
ohvdNotifications 15 }

transportforlondonRegids OBJECT IDENTIFIER ::= { transportforlondon 2 }
transportforlondonCommon OBJECT IDENTIFIER ::= { transportforlondon 3 }
transportforlondonProducts OBJECT IDENTIFIER ::= { transportforlondon 4 }
transportforlondonExperimental OBJECT IDENTIFIER ::= { transportforlondon
5 }

END

```