

NEMA Standards Publication TS 3.5-1996

*NATIONAL TRANSPORTATION COMMUNICATIONS FOR ITS PROTOCOL (NTCIP)
OBJECT DEFINITIONS FOR ACTUATED TRAFFIC SIGNAL CONTROLLER UNITS*

Version 96.01.07 April 7, 1997

Published by

National Electrical Manufacturers Association
1300 N. 17th Street
Rosslyn, Virginia 22209

© Copyright 1997 by the National Electrical Manufacturers Association. All rights including translation into other languages, reserved under the Universal Copyright Convention, the Berne Convention for the Protection of Literary and Artistic Works, and the International and Pan American Copyright Conventions.

Table of Contents

Foreword	vi
Section 1 General	1
1.1 SCOPE.....	1
1.2 REFERENCES.....	1
1.2.1 Normative References.....	1
1.2.2 Other References	2
1.3 ACTUATED CONTROLLER UNIT TERMS.....	2
1.4 ABBREVIATIONS AND ACRONYMS.....	6
Section 2 OBJECT DEFINITIONS	7
2.1 MIB HEADER.....	7
2.2 PHASE PARAMETERS	7
2.2.1 Maximum Phases.....	7
2.2.2 Phase Table	8
2.2.2.1 Phase Number	8
2.2.2.2 Phase Walk Parameter	9
2.2.2.3 Phase Pedestrian Clear Parameter	9
2.2.2.4 Phase Minimum Green Parameter	9
2.2.2.5 Phase Passage Parameter.....	9
2.2.2.6 Phase Maximum Green 1 Parameter	10
2.2.2.7 Phase Maximum Green 2 Parameter	10
2.2.2.8 Phase Yellow Change Parameter.....	10
2.2.2.9 Phase Red Clear Parameter.....	11
2.2.2.10 Phase Red Revert.....	11
2.2.2.11 Phase Added Initial Parameter.....	11
2.2.2.12 Phase Maximum Initial Parameter.....	12
2.2.2.13 Phase Time Before Reduction Parameter	12
2.2.2.14 Phase Cars Before Reduction Parameter	12
2.2.2.15 Phase Time To Reduce Parameter	13
2.2.2.16 Phase Reduce By	13
2.2.2.17 Phase Minimum Gap Parameter	13
2.2.2.18 Phase Dynamic Max Limit	13
2.2.2.19 Phase Dynamic Max Step	14
2.2.2.20 Phase Startup	14
2.2.2.21 Phase Options	15
2.2.2.22 Phase Ring Parameter	16
2.2.2.23 Phase Concurrency	16
2.2.3 Maximum Phase Groups.....	16
2.2.4 Phase Status Group Table.....	16
2.2.4.1 Phase Status Group Number.....	17
2.2.4.2 Phase Status Group Reds	17
2.2.4.3 Phase Status Group Yellows	18
2.2.4.4 Phase Status Group Greens.....	18
2.2.4.5 Phase Status Group Dont Walks.....	18
2.2.4.6 Phase Status Group Pedestrian clears	19
2.2.4.7 Phase Status Group Walks.....	19
2.2.4.8 Phase Status Group Vehicle Calls.....	19
2.2.4.9 Phase Status Group Pedestrian Calls	20
2.2.4.10 Phase Status Group Phase Ons.....	20
2.2.4.11 Phase Status Group Phase Nexts	20
2.2.5 Phase Control Table.....	21

2.2.5.1 Phase Control Group Number	21
2.2.5.2 Phase Omit Control.....	22
2.2.5.3 Pedestrian Omit Control.....	22
2.2.5.4 Phase Hold Control.....	23
2.2.5.5 Phase Force Off Control	23
2.2.5.6 Vehicle Call Control	24
2.2.5.7 Pedestrian Call Control.....	24
2.3 DETECTOR PARAMETERS	25
2.3.1 Maximum Vehicle Detectors.....	25
2.3.2 Vehicle Detector Parameter Table	25
2.3.2.1 Vehicle Detector Number.....	26
2.3.2.2 Vehicle Detector Options Parameter	26
2.3.2.3 Vehicle Detector Call Phase Parameter	26
2.3.2.4 Vehicle Detector Switch Phase Parameter.....	27
2.3.2.5 Vehicle Detector Delay Parameter	27
2.3.2.6 Vehicle Detector Extend Parameter	27
2.3.2.7 Vehicle Detector Queue Limit.....	27
2.3.2.8 Vehicle Detector No Activity Parameter.....	28
2.3.2.9 Vehicle Detector Maximum Presence Parameter.....	28
2.3.2.10 Vehicle Detector Erratic Counts Parameter.....	28
2.3.2.11 Vehicle Detector Fail Time Parameter.....	28
2.3.2.12 Vehicle Detector Alarms	29
2.3.2.13 Vehicle Detector Reported Alarms	29
2.3.2.14 Vehicle Detector Reset.....	30
2.3.3 Maximum Vehicle Detector Status Groups	30
2.3.4 Vehicle Detector Status Group Table.....	30
2.3.4.1 Detector Status Group Number	31
2.3.4.2 Detector Status Group Active	31
2.3.4.3 Detector Alarm Status.....	31
2.3.5 Volume / Occupancy report.....	32
2.3.5.1 Volume / Occupancy Sequence	32
2.3.5.2 Volume / Occupancy Period	32
2.3.5.3 Active Volume / Occupancy Detectors	32
2.3.5.4 Volume / Occupancy Table.....	32
2.3.5.4.1 Volume data	33
2.3.5.4.2 Occupancy data	33
2.3.6 Maximum Pedestrian Detectors	34
2.3.7 Pedestrian Detector Parameter Table.....	34
2.3.7.1 Pedestrian Detector Number	34
2.3.7.2 Pedestrian Detector Call Phase Parameter.....	35
2.3.7.3 Pedestrian Detector No Activity Parameter	35
2.3.7.4 Pedestrian Detector Maximum Presence Parameter	35
2.3.7.5 Pedestrian Detector Erratic Counts Parameter	35
2.3.7.6 Pedestrian Detector Alarms	36
2.4 UNIT PARAMETERS	36
2.4.1 StartUp Flash Parameter.....	36
2.4.2 Automatic Ped Clear Parameter.....	37
2.4.3 Backup Time Parameter.....	37
2.4.4 Unit Red Revert Parameter	37
2.4.5 Unit Control Status	38
2.4.6 Unit Flash Status	38
2.4.7 Unit Alarm Status 2.....	39
2.4.8 Unit Alarm Status 1.....	39
2.4.9 Short Alarm Status	39
2.4.10 Unit Control.....	40
2.4.11 Maximum Alarm Groups.....	41

2.4.12 Alarm Group Table	41
2.4.12.1 Alarm Group Number.....	41
2.4.12.2 Alarm Group State	41
2.4.13 Maximum Special Function Outputs	42
2.4.14 Special Function Output Table.....	42
2.4.14.1 Special Function Output Number.....	42
2.4.14.2 Special Function Output Control.....	43
2.5 COORDINATION PARAMETERS	43
2.5.1 Coord Operational Mode Parameter	43
2.5.2 Coord Correction Mode Parameters	43
2.5.3 Coord Maximum Mode Parameters	44
2.5.4 Coord Force Mode Parameters.....	44
2.5.5 Maximum Patterns Parameters.....	45
2.5.6 Pattern Table Type.....	45
2.5.7 Pattern Table	45
2.5.7.1 Pattern Number Entry	46
2.5.7.2 Pattern Cycle Time	46
2.5.7.3 Pattern Offset Time Parameter.....	46
2.5.7.4 Pattern Split Number Parameter.....	47
2.5.7.5 Pattern Sequence Parameter	47
2.5.8 Maximum Splits	47
2.5.9 Split Table.....	47
2.5.9.1 Split Number	48
2.5.9.2 Split Phase Number	48
2.5.9.3 Split Time Parameter	48
2.5.9.4 Split Mode Parameter	49
2.5.9.5 Split Coordinated Phase	49
2.5.10 Coordination Pattern Status	49
2.5.11 Local Free Status	50
2.5.12 Coordination Cycle Status.....	50
2.5.13 Coordination Sync Status.....	50
2.5.14 System Pattern Control	51
2.5.15 System Sync Control.....	51
2.6 TIME BASE PARAMETERS	51
2.6.1 Time Base Pattern Sync Parameter.....	51
2.6.2 Maximum Time Base Actions.....	52
2.6.3 Time Base ASC Action Table.....	52
2.6.3.1 Time Base Action Number Entry	52
2.6.3.2 Time Base Action Pattern Parameter	53
2.6.3.3 Time Base Action Auxiliary Function Parameter	53
2.6.3.4 Time Base Action Special Function Parameter	53
2.6.4 Time Base ASC Action Status.....	53
2.7 PREEMPT PARAMETERS	54
2.7.1 Maximum Preempts	54
2.7.2 Preempt Table	54
2.7.2.1 Preempt Number.....	55
2.7.2.2 Preempt Control Parameter	55
2.7.2.3 Preempt Link Parameter	55
2.7.2.4 Preempt Delay Parameter	56
2.7.2.5 Preempt Duration Parameter	56
2.7.2.6 Preempt Minimum Green Parameter.....	56
2.7.2.7 Preempt Minimum Walk Parameter.....	56
2.7.2.8 Preempt Enter Pedestrian Clear Parameter.....	57
2.7.2.9 Preempt Track Green Parameter	57
2.7.2.10 Preempt Minimum Dwell Parameter	57
2.7.2.11 Preempt Maximum Presence Parameter	57

2.7.2.12 Preempt Track Phase Parameter	58
2.7.2.13 Preempt Dwell Phase Parameters.....	58
2.7.2.14 Preempt Dwell Ped Parameters	58
2.7.2.15 Preempt Exit Phase Parameters	58
2.7.2.16 Preempt State	59
2.7.3 Preempt Control Table	59
2.7.3.1 Preempt Control Number.....	60
2.7.3.2 Preempt Control State	60
2.8 RING PARAMETERS	60
2.8.1 Maximum Rings.....	60
2.8.2 Maximum Sequences.....	60
2.8.3 Sequence Table	61
2.8.3.1 Sequence Number	61
2.8.3.2 Sequence Ring Number	61
2.8.3.3 SequenceData	62
2.8.4 Maximum Ring Control Groups.....	62
2.8.5 Ring Control Group Table	62
2.8.5.1 Ring Control Group Number	63
2.8.5.2 Ring Stop Time Control.....	63
2.8.5.3 Ring Force Off Control.....	63
2.8.5.4 Ring Max 2 Control	64
2.8.5.5 Ring Max Inhibit Control.....	64
2.8.5.6 Ring Ped Recycle Control.....	65
2.8.5.7 Ring Red Rest Control.....	65
2.8.5.8 Ring Omit Red Control.....	66
2.9 CHANNEL PARAMETERS	66
2.9.1 Maximum Channels.....	66
2.9.2 Channel Table	66
2.9.2.1 Channel Number.....	67
2.9.2.2 Channel Control Source Parameters.....	67
2.9.2.3 Channel Control Type Parameters	67
2.9.2.4 Channel Flash Parameters	68
2.9.2.5 Channel Dim Parameters	68
2.9.3 Maximum Channel Status Groups	68
2.9.4 Channel Status Group Table.....	69
2.9.4.1 Channel Status Group Number	69
2.9.4.2 Channel Status Group Reds.....	69
2.9.4.3 Channel Status Group Yellows.....	70
2.9.4.4 Channel Status Group Greens	70
2.10 OVERLAP PARAMETERS	70
2.10.1 Maximum Overlaps	70
2.10.2 Overlap Table	71
2.10.2.1 Overlap Number.....	71
2.10.2.2 Overlap Type	71
2.10.2.3 Overlap Included Phase Parameters.....	72
2.10.2.4 Overlap Modifier Phase Parameters.....	72
2.10.2.5 Overlap Trailing Green Parameter.....	72
2.10.2.6 Overlap Trailing Yellow Change Parameter	73
2.10.2.7 Overlap Trailing Red Clear Parameter	73
2.10.3 Maximum Overlap Status Groups	73
2.10.4 Overlap Status Group Table.....	73
2.10.4.1 Overlap Status Group Number	74
2.10.4.2 Overlap Status Group Reds.....	74
2.10.4.3 Overlap Status Group Yellows.....	74
2.10.4.4 Overlap Status Group Greens	75
2.11 TS2 PORT 1 PARAMETERS.....	75

2.11.1 Maximum Port 1 Addresses	75
2.11.2 Port 1 Table	75
2.11.2.1 Port 1 Number.....	76
2.11.2.2 Port 1 Device Present.....	76
2.11.2.3 Port 1 Frame 40 Enable.....	76
2.11.2.4 Port 1 Status	77
2.11.2.5 Port 1 Fault Frame	77
Section 3 Group Definitions	79
3.1 PHASE CONFORMANCE GROUP	79
3.2 DETECTOR CONFORMANCE GROUP	80
3.3 VOLUME OCCUPANCY REPORT CONFORMANCE GROUP.....	80
3.4 UNIT CONFORMANCE GROUP.....	80
3.5 SPECIAL FUNCTION CONFORMANCE GROUP	80
3.6 COORDINATION CONFORMANCE GROUP	81
3.7 TIME BASE CONFORMANCE GROUP	81
3.8 PREEMPT CONFORMANCE GROUP.....	81
3.9 RING CONFORMANCE GROUP	81
3.10 CHANNEL CONFORMANCE GROUP	82
3.11 OVERLAP CONFORMANCE GROUP	82
3.12 TS-2 PORT1 CONFORMANCE GROUP	82
Section 4 CONFORMANCE STATEMENTS	83

Foreword

This publication describes the objects used for managing actuated traffic signal controller units.

The text consists of mandatory requirements which are classified as NEMA Standard Publication and information that is in conformance with NEMA Authorized Engineering Information. Both are defined in the back of each NEMA Standard Publication.

User inputs are solicited by canvass on each NEMA standards proposal for Transportation Management and Associated Control Equipment. This procedure provides guidance before industry action. Every comment received is acknowledged and the action taken reported to the submitter. Comments should be submitted to:

Vice President, Engineering Department
National Electrical Manufacturers Association
1300 N.17th Street
Rosslyn, Virginia 22209

This standards publication was developed by the Transportation Management Systems and Associated Control Devices section of the National Electrical Manufacturers Association. Section approval of the standard does not necessarily imply that all section members voted for its approval or participated in its development. At the time it was approved, the Transportation Management Systems and Associated Control Devices section had the following members.

ADDCO Manufacturing Co.—St. Paul, MN
American Electronic Sign—Spokane, WA
Automatic Signal/Eagle Signal Corporation—Austin, TX
BI Tran Systems, Inc.—Sacramento, CA
Cylink Corporation—Sunnyvale, CA
Eberle Design, Inc.—Phoenix, AZ
Econolite Control Products, Inc.—Anaheim, CA
Fiberoptic Display Systems, Inc.—Smithfield, RI
Gardner-Rowe Systems, Inc.—Walnut Creek, CA
Information Station Specialists, Inc.—Zeeland, MI
Intersection Development Corp.—Fullerton, CA
ITS Product Group—Ormond Beach, FL
McCain Traffic Supply, Inc.—Vista, CA
P B Farradyne Inc.—Rockville, MD
Peek Traffic - Transyt Corp.—Tallahassee, FL
Rockwell Automation—Mayfield Heights, OH
Safetran Traffic Systems, Inc.—Colorado Springs, CO
Skyline Products, Inc.—Colorado Springs, CO
3M Intelligent Transportation Systems—St. Paul, MN
Traffic Sensor Corp.—Corona, CA
Viggen Corp.—McLean, VA
Vultron, Inc.—Rochester Hills, MI

The Joint AASHTO/ITE/NEMA Committee on the NTCIP has recommended this standards publication for adoption to both the American Association of State Highway and Transportation Officials (AASHTO) and the Institute of Transportation Engineers (ITE). After adoption is approved by AASHTO and ITE, future printings of this publication will have a Joint AASHTO/ITE/NEMA Standard cover, and will appear different from this publication. Unless noted, the content of the publications will be the same.

Section 1 General

1.1 SCOPE

This publication defines objects which are specific to actuated signal controllers. It also defines standardized object groups which can be used for conformance statements.

1.2 REFERENCES

For approved errata sheets, contact:

NTCIP Coordinator
National Electrical Manufacturers Association
1300 North 17th Street, Suite 1847
Rosslyn, VA 22209-3801
email: ntcip@nema.org

1.2.1 Normative References

Information on proposed errata, which are under discussion by the relevant NTCIP Working Group, is available on the World Wide Web at <http://www.ntcip.org>.

The following standards (normative references) contain provisions which, through reference in this text, constitute provisions of this Standard. Other documents and standards (other references) are referenced in these documents, which might provide a complete understanding of the entire protocol and the relations between all parts of the protocol. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this Standard are encouraged to investigate the possibility of applying the most recent editions of the standard listed below.

The following standards contain provisions which, through reference in this text, constitute provisions of this Standard. While end users of NTCIP do not need to obtain these documents, they do provide a complete understanding of the protocol. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this Standard are encouraged to investigate the possibility of applying the most recent editions of the standards listed below.

ANSI

11 West 42nd Street, 13th Floor
New York, New York 10036

ISO/IEC 8824-1:1995 *Information Technology—Abstract Syntax Notation One (ASN.1): Specification of Basic Notation*

ISO/IEC 8824-2:1995 *Information Technology—Abstract Syntax Notation One (ASN.1): Information Object Specification*

DDN Network Information Center

14200 Park Meadow Drive
Suite 200
Chantilly, VA 22021

Electronic copies of RFC documents may be obtained using "anonymous FTP" to the host <nic.ddn.mil> or <ds.internic.net>. Printed copies are available from: (800) 365-3642 or (703) 802-4535.

RFC1155 *Structure and Identification of Management Information for TCP/IP-based Internets*. K. McCloghrie; M. Rose; 05/10/1990

RFC1212 *Concise MIB Definitions*. K. McCloghrie; M. Rose; 03/26/1991

1.2.2 Other References

National Electrical Manufacturers Association

1300 North 17th Street, Suite 1847
Rosslyn, VA 22209

TS 2-1992 *Traffic Controller Assemblies*

TS 3.1-1996 *National Transportation Communications for ITS Protocol - Overview*

TS 3.2-1996 *National Transportation Communications for ITS Protocol - Simple
Transportation
Management Framework*

TS 3.3-1996 *National Transportation Communications for ITS Protocol - Class B Profile*

ANSI

11 West 42nd Street, 13th Floor
New York, New York 10036
(212) 642-4900

ISO/IEC 8824-3:1995 *Information Technology—Abstract Syntax Notation One (ASN.1):
Constraint Specification*

ISO/IEC 8824-4:1995 *Information Technology—Abstract Syntax Notation One (ASN.1):
Parameterization of ASN.1 Specifications*

ISO/IEC 8825-1:1995 *Information Technology—ASN.1 Encoding Rules: Specification of Basic
Encoding Rules (BER), Canonical Encoding Rules (CER) and
Distinguished Encoding Rules (DER).*

DDN Network Information Center

14200 Park Meadow Drive
Suite 200
Chantilly, VA 22021

RFC1213 *Management Information Base for Network Management of TCP/IP-
based Internets: MIB-II*. K. McCloghrie; M. Rose; CP/IP-base

RFC1157 *A Simple Network Management Protocol (SNMP)*. M. Schoffstall;
M. Feder; J. Davin; J. Case; 05/10/1990

1.3 ACTUATED CONTROLLER UNIT TERMS

These terms define the nomenclature frequently used in regard to actuated traffic signal control devices. These terms reflect the consensus of the traffic control equipment industry as represented by NEMA and are intended to be in harmony with terminology in current usage.

Actuation: The operation of any type of detector.

Automatic Flash: Automatic programmed flash mode not caused by manual switch activation or fault condition or startup.

Auxiliary Function: A control that may activate auxiliary functions or outputs in an actuated controller unit.

Backup Mode: Control by local TBC or Interconnect based on absence of master or central command.

Barrier: A barrier (compatibility line) is a reference point in the preferred sequence of a multi-ring CU at which all rings are interlocked. Barriers assure there will be no concurrent selection and timing of conflicting phases for traffic movement in different rings. All rings cross the barrier simultaneously to select and time phases on the other side.

Call: A registration of a demand for right-of-way by traffic (vehicles or pedestrians) to a controller unit.

Call, Serviceable Conflicting: A call which:

- a. Occurs on a conflicting phase not having the right-of-way at the time the call is placed.
- b. Occurs on a conflicting phase which is capable of responding to a call.
- c. When occurring on a conflicting phase operating in an occupancy mode, remains present until given its right-of-way.

Channel: Three circuits of a Monitor Device wired to monitor the green, yellow, and red outputs of the associated load switch position in the Terminal & Facilities. Channel 1 is assumed to monitor Load Switch 1, etc.

Check: An output from a controller unit that indicates the existence of unanswered call(s).

Concurrency Group: A group of phases which describes possible timing combinations. A phase within the group shall be able to time concurrently with any other phase from another ring contained in the group. For example, in the typical dual-ring eight phase design, phases 1, 2, 5 and 6 form one concurrency group and phases 3, 4, 7, and 8 form another concurrency group.

Concurrent Timing: A mode of controller unit operation whereby a traffic phase can be selected and timed simultaneously and independently with another traffic phase.

Controller Assembly: A complete electrical device mounted in a cabinet for controlling the operation of a traffic control signal display(s).

Controller Unit: A controller unit is that portion of a controller assembly that is devoted to the selection and timing of signal displays.

Coordination: The control of controller units in a manner to provide a relationship between specific green indications at adjacent intersections in accordance with a time schedule to permit continuous operation of groups of vehicles along the street at a planned speed.

Coordinator: A device or program/routine which provides coordination.

Cycle: The total time to complete one sequence of signalization around an intersection. In an actuated controller unit, a complete cycle is dependent on the presence of calls on all phases. In a pretimed controller unit it is a complete sequence of signal indications.

Cycle Length: The time period in seconds required for one complete cycle.

Detector, Pedestrian: A detector that is responsive to operation by or the presence of a pedestrian.

Detector, System: Any type of vehicle detector used to obtain representative traffic flow information.

Detector, Vehicle: A detector that is responsive to operation by or the presence of a vehicle.

Dial: The cycle timing reference or coordination input activating same. Dial is also frequently used to describe the cycle.

Display Map: A graphic display of the street system being controlled showing the status of the signal indications and the status of the traffic flow conditions.

Dual Entry: Dual entry is a mode of operation (in a multi-ring CU) in which one phase in each ring must be in service. If a call does not exist in a ring when it crosses the barrier, a phase is selected in that ring to be activated by the CU in a predetermined manner.

Dwell: The interval portion of a phase when present timing requirements have been completed.

First Coordinated Phase: The coordinated phase which occurs first within the concurrent group of phases containing the coordinated phase(s) when there are constant calls on all phases.

Flash: Operation where one section in each vehicle signal (yellow or red) is alternately on and off with a one second cycle time and a 50 percent duty cycle.

Fault Monitor State: internal CU diagnostics have determined that the CU device is not in a safe operational state. An output may be asserted to indicate this condition.

Force Off: A command to force the termination of the green interval in the actuated mode or Walk Hold in the nonactuated mode of the associated phase. Termination is subject to the presence of a serviceable conflicting call. The Force Off function shall not be effective during the timing of the Initial, Walk, or Pedestrian Clearance. The Force Off shall only be effective as long as the condition is sustained. If a phase specific Force Off is applied, the Force Off shall not prevent the start of green for that phase.

Free: Operation without coordination control from any source.

Gap Reduction: A feature whereby the Unit Extension or allowed time spacing between successive vehicle actuations on the phase displaying the green in the extensible portion of the interval is reduced.

Group: Any portion of a traffic control network (system) that can be controlled by a common set of timing plans.

Hold: A command that retains the existing Green interval.

Hold-On Line: A signal to an intersection controller commanding it to remain under computer control.

Interconnect: A means of remotely controlling some or all of the functions of a traffic signal.

Intersection status: The knowledge of whether a controlled intersection is on-line and which mode it is currently operating in.

Interval: The part or parts of the signal cycle during which signal indications do not change.

Load Switch Driver Group: The set of three outputs which are used to drive load switch inputs to provide a Green, Yellow, or Red output condition for vehicle signals or Walk, Ped Clear, or Dont Walk output condition for pedestrian signals.

Malfunction Management Unit (MMU): A device used to detect and respond to improper and conflicting signals and improper operating voltages in a traffic controller assembly.

Maximum Green: The maximum green time with an serviceable opposing actuation, which may start during the initial portion.

Minimum Green Interval: The shortest green time of a phase. If a time setting control is designated as Minimum Green, the green time shall be not less than that setting.

Multi-Ring Controller Unit: A multi-ring CU contains two or more interlocked rings which are arranged to time in a preferred sequence and to allow concurrent timing of all rings, subject to barrier restraint.

Nonlocking Memory: A mode of actuated-controller-unit operation which does not require the retention of a call for future utilization by the controller assembly.

Occupancy: A measurement of vehicle presence within a zone of detection, expressed in seconds of time a given point or area is occupied by a vehicle.

Off-line: A controller assembly not under the control of the normal control source.

Offset: The time relationship, expressed in seconds, between the starting point of the first coordinated phase Green and a system reference point.

Omit, Phase: A command that causes omission of a selected phase.

On-line: A controller assembly under the control of the normal control source.

Overlap: A Green indication that allows traffic movement during the green intervals of and clearance intervals between two or more phases.

Passage Time: The time allowed for a vehicle to travel at a selected speed from the detector to the stop line.

Pattern: A unique set of coordination parameters (cycle value, split values, offset value, and sequence).

Pedestrian Clearance Interval: The first clearance interval for the pedestrian signal following the pedestrian WALK indication.

Pedestrian Recycle: A method of placing a recurring demand for pedestrian service on the movement when that movement is not in its Walk interval.

Permissive: A time period, during which the CU is allowed to leave the coordinated phase(s) under coordination control to go to other phases.

Phase Sequence: A predetermined order in which the phases of a cycle occur.

Phase, Active: The indicated phase is currently timing. A phase is always active if it is Green or Yellow (Walk or Pedestrian Clear for Pedestrian Phases). It is also active if it is timing Red Clearance. It may be considered active during Red Dwell.

Phase, Conflicting: Conflicting phases are two or more traffic phases which will cause interfering traffic movements if operated concurrently.

Phase, Nonconflicting: Nonconflicting phases are two or more traffic phases which will not cause interfering traffic movements if operated concurrently.

Phase, Pedestrian: A traffic phase allocated to pedestrian traffic which may provide a right-of-way pedestrian indication either concurrently with one or more vehicular phases, or to the exclusion of all vehicular phases.

Phase, Traffic: Those green, change and clearance intervals in a cycle assigned to any independent movement(s) of traffic.

Phase, Vehicular: A vehicular phase is a phase which is allocated to vehicular traffic movement as timed by the controller unit.

Preemption: The transfer of the normal control of signals to a special signal control mode for the purpose of servicing railroad crossings, emergency vehicle passage, mass transit vehicle passage, and other special tasks, the control of which require terminating normal traffic control to provide the priority needs of the special task.

Preemptor: A device or program/routine which provides preemption.

Progression: The act of various controller units providing specific green indications in accordance with a time schedule to permit continuous operation of groups of vehicles along the street at a planned speed.

Red Clearance Interval: A clearance interval which may follow the yellow change interval during which both the terminating phase and the next phase display Red signal indications.

Red Revert: Provision within the controller unit to assure a minimum Red signal indication in a phase following the Yellow Change interval of that phase .

Rest: The interval portion of a phase when present timing requirements have been completed.

Ring: A ring consists of two or more sequentially timed and individually selected conflicting phases so arranged as to occur in an established order.

Sequence, Interval: The order of appearance of signal indications during successive intervals of a cycle.

Single Entry: Single entry is a mode of operation (in a multi-ring CU) in which a phase in one ring can be selected and timed alone if there is no demand for service in a nonconflicting phase on the parallel ring(s).

Single-Ring Controller Unit: A single-ring CU contains two or more sequentially timed and individually selected conflicting phases so arranged as to occur in an established order.

Special Function: A control that may activate specific functions or outputs in an actuated controller unit.

Split: The segment of the cycle length allocated to each phase or interval that may occur (expressed in seconds). In an actuated controller unit, split is the time in the cycle allocated to a phase.

Standby Mode: An operational state called by master or central command which directs the controller unit to select Pattern, Automatic Flash, or Automatic Free based on local Time Base schedule or Interconnect inputs.

Time Base Control: A means for the automatic selection of modes of operation of traffic signals in a manner prescribed by a predetermined time schedule.

Timing Plan: The Split times for all segments (Phase/Interval) of the coordination cycle.

Volume: The number of vehicles passing a given point per unit of time.

Yellow Change Interval: The first interval following the green interval in which the signal indication for that phase is yellow.

Yield: A command which permits termination of the green interval.

1.4 ABBREVIATIONS AND ACRONYMS

The abbreviations used in this Standard Publication are defined as follows:

BIU—Bus Interface Unit

CA—Controller Assembly

CU—Controller Unit

MMU—Malfunction Management Unit

TBC—Time Base Control

TF—Terminals and Facilities

Section 2 OBJECT DEFINITIONS

This section defines those objects which are specifically used by actuated traffic signal controllers. The objects are defined using the OBJECT-TYPE macro specified in RFC 1212. The text provided from Clause 2.1 through the end of the section (except the clause headings) constitutes the NEMA Standard ASC MIB.

The clauses below present the objects in lexicographical order of their OBJECT IDENTIFIERS which correspond to their physical location within the global naming tree. All of the objects defined in this document reside under the "asc" node of the global naming tree. To aid in object management, the "asc" node has been subdivided into logical categories, each defined by a node under the "asc" node. The individual objects are then located under the appropriate node.

Nodes should not be confused with Conformance Groups, which are defined in Section 3. A Conformance Group is a logical grouping of objects which is used for conformance statements. While Conformance Groups will frequently correspond to the nodal structure, a Conformance Group may contain objects which are not lexicographically ordered. For example, a Schedule Conformance Group may contain both "global" and "asc" specific objects.

Text preceded by a double hyphen in the MIB definitions represent normative text for this standard.

2.1 MIB HEADER

```
ASC_MIB1 DEFINITIONS ::= BEGIN
-- the following OBJECT IDENTIFIERS are used in the ASC MIB:
```

```
IMPORTS
    devices
    FROM TMIB;
```

```
asc OBJECT IDENTIFIER ::= { devices 1 }
```

2.2 PHASE PARAMETERS

```
phase OBJECT IDENTIFIER
::= { asc 1 }
```

-- This node shall contain objects that configure, monitor or control phase functions for this device.

2.2.1 Maximum Phases

```
maxPhases OBJECT-TYPE
    SYNTAX INTEGER (0..255)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The Maximum Number of Phases this Actuated Controller Unit supports. This object
        indicates the maximum rows which shall appear in the phaseTable object."
    ::= { phase 1 }
```

2.2.2 Phase Table

phaseTable OBJECT-TYPE
SYNTAX SEQUENCE OF PhaseEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION
"A table containing Actuated Controller Unit phase parameters. The number of rows in this table is equal to the maxPhases object."
 ::= { phase 2 }

phaseEntry OBJECT-TYPE
SYNTAX PhaseEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION
"Parameters for a specific Actuated Controller Unit phase."
 INDEX { phaseNumber }
 ::= { phaseTable 1 }

PhaseEntry ::= SEQUENCE {
phaseNumber INTEGER,
phaseWalk INTEGER,
phasePedestrianClear INTEGER,
phaseMinimumGreen INTEGER,
phasePassage INTEGER,
phaseMaximum1 INTEGER,
phaseMaximum2 INTEGER,
phaseYellowChange INTEGER,
phaseRedClear INTEGER,
phaseRedRevert INTEGER,
phaseAddedInitial INTEGER,
phaseMaximumInitial INTEGER,
phaseTimeBeforeReduction INTEGER,
phaseCarsBeforeReduction INTEGER,
phaseTimeToReduce INTEGER,
phaseReduceBy INTEGER,
phaseMinimumGap INTEGER,
phaseDynamicMaxLimit INTEGER,
phaseDynamicMaxStep INTEGER,
phaseStartup INTEGER,
phaseOptions INTEGER,
phaseRing INTEGER,
phaseConcurrency OCTET STRING }

2.2.2.1 Phase Number

phaseNumber OBJECT-TYPE
SYNTAX INTEGER (1..255)
ACCESS read-only
STATUS mandatory
DESCRIPTION
"The phase number for objects in this row. This value shall not exceed the maxPhases object value."
 ::= { phaseEntry 1 }

2.2.2.2 Phase Walk Parameter

phaseWalk OBJECT-TYPE
SYNTAX INTEGER (0..255)
ACCESS read-write
STATUS mandatory
DESCRIPTION
"Phase Walk Parameter in seconds. This shall control the amount of time the Walk indication shall be displayed."
REFERENCE
"NEMA TS 2 Clause 3.5.3.1 and 3.5.3.2.2.a"
::= { phaseEntry 2 }

2.2.2.3 Phase Pedestrian Clear Parameter

phasePedestrianClear OBJECT-TYPE
SYNTAX INTEGER (0..255)
ACCESS read-write
STATUS mandatory
DESCRIPTION
"Phase Pedestrian Clear Parameter in seconds. This shall control the duration of the Pedestrian Clearance output (if present) and the flashing period of the Don't Walk output. "
REFERENCE
"NEMA TS 2 Clause 3.5.3.1 and 3.5.3.2.2.b"
::= { phaseEntry 3 }

2.2.2.4 Phase Minimum Green Parameter

phaseMinimumGreen OBJECT-TYPE
SYNTAX INTEGER (0..255)
ACCESS read-write
STATUS mandatory
DESCRIPTION
"Phase Minimum Green Parameter in seconds (NEMA TS 2 range: 1-255 sec). The first timed portion of the Green interval which may be set in consideration of the storage of vehicles between the zone of detection for the approach vehicle detector(s) and the stop line."
REFERENCE
"NEMA TS 2 Clause 3.5.3.1 and 3.5.3.2.1.a.(1)"
::= { phaseEntry 4 }

2.2.2.5 Phase Passage Parameter

phasePassage OBJECT-TYPE
SYNTAX INTEGER (0..255)
ACCESS read-write
STATUS mandatory
DESCRIPTION
"Phase Passage Parameter in tenth seconds (0-25.5 sec). Passage Time, Vehicle Interval, Preset Gap, Vehicle Extension: the extensible portion of the Green shall be a function of vehicle actuations that occur during the Green interval. The phase shall remain in the extensible portion of the Green interval as long as the passage timer is not timed out. The timing of this portion of the green interval shall be reset with each subsequent vehicle actuation and shall not commence to time again until the vehicle actuation is removed."
REFERENCE
"NEMA TS 2 Clause 3.5.3.1 and 3.5.3.2.1.a.(2)"
::= { phaseEntry 5 }

2.2.2.6 Phase Maximum Green 1 Parameter

phaseMaximum1 OBJECT-TYPE
SYNTAX INTEGER (0..255)
ACCESS read-write
STATUS mandatory
DESCRIPTION
"Phase Maximum 1 Parameter in seconds (NEMA TS 2 range: 1-255 sec). This time setting shall determine the maximum length of time this phase may be held Green in the presence of a serviceable conflicting call. In the absence of a serviceable conflicting call the Maximum Green timer shall be held reset unless Max Vehicle Recall is enabled for this phase. This is the default maximum value to use. It may be overridden via an external input, coordMaximumMode or other method."
REFERENCE
"NEMA TS 2 Clause 3.5.3.1, 3.5.3.2.1.a.(3) and 3.5.3.5"
::= { phaseEntry 6 }

2.2.2.7 Phase Maximum Green 2 Parameter

phaseMaximum2 OBJECT-TYPE
SYNTAX INTEGER (0..255)
ACCESS read-write
STATUS mandatory
DESCRIPTION
"Phase Maximum 2 Parameter in seconds (NEMA TS 2 range: 1-255 sec). This time setting shall determine the maximum length of time this phase may be held Green in the presence of a serviceable conflicting call. In the absence of a serviceable conflicting call the Maximum Green timer shall be held reset unless Max Vehicle Recall is enabled for this phase. This may be implemented as the max green timer via an external input, coordMaximumMode or other method."
REFERENCE
"NEMA TS 2 Clause 3.5.3.1, 3.5.3.2.1.a.(3), 3.5.3.5 and 3.5.4.1 (7)"
::= { phaseEntry 7 }

2.2.2.8 Phase Yellow Change Parameter

phaseYellowChange OBJECT-TYPE
SYNTAX INTEGER (0..255)
ACCESS read-write
STATUS mandatory
DESCRIPTION
"Phase Yellow Change Parameter in tenth seconds (NEMA TS 2 range: 3-25.5 sec). Following the Green interval of each phase the CU shall provide a Yellow Change interval which is timed according to the Yellow Change parameter for that phase."
REFERENCE
"NEMA TS 2 Clause 3.5.3.1 and 3.5.3.2.5.a"
::= { phaseEntry 8 }

2.2.2.9 Phase Red Clear Parameter

phaseRedClear OBJECT-TYPE

SYNTAX INTEGER (0..255)

ACCESS read-write

STATUS mandatory

DESCRIPTION

"Phase Red Clearance Parameter in tenth seconds (0-25.5 sec). Following the Yellow Change interval for each phase, the CU shall provide a Red Clearance interval which is timed according to the Red Clearance parameter for that phase."

REFERENCE

"NEMA TS 2 Clause 3.5.3.1 and 3.5.3.2.5.b"

::= { phaseEntry 9 }

2.2.2.10 Phase Red Revert

phaseRedRevert OBJECT-TYPE

SYNTAX INTEGER (0..255)

ACCESS read-write

STATUS optional

DESCRIPTION

"Red revert time parameter in tenth seconds . A minimum Red indication to be timed following the Yellow Change interval and prior to the next display of Green on the same signal output driver group.

The unitRedRevert parameter shall act as a minimum red revert time for all signal displays.

The phaseRedRevert parameter may increase the red revert time for a specific phase. If the phaseRedRevert parameter is less than the unitRedRevert the unitRedRevert time shall be used."

::= { phaseEntry 10 }

2.2.2.11 Phase Added Initial Parameter

phaseAddedInitial OBJECT-TYPE

SYNTAX INTEGER (0..255)

ACCESS read-write

STATUS mandatory

DESCRIPTION

"Phase Added Initial Parameter in tenths of seconds (0-25.5 sec). Added Initial parameter (Seconds / Actuation) shall determine the time by which the variable initial time period will be increased from zero with each vehicle actuation received during the associated phase Yellow and Red intervals."

REFERENCE

"NEMA TS 2 Clause 3.5.3.1 and 3.5.3.2.1.b.(1).(b)"

::= { phaseEntry 11 }

2.2.2.12 Phase Maximum Initial Parameter

phaseMaximumInitial OBJECT-TYPE

SYNTAX INTEGER (0..255)

ACCESS read-write

STATUS mandatory

DESCRIPTION

"Phase Maximum Initial Parameter in seconds (0-255 sec). The maximum value of the variable initial timing period. Variable Initial timing shall equal the lesser of [added initial(seconds / actuation) * number of actuations] or [Max Initial]. The variable initial time shall not be less than Minimum Green."

REFERENCE

"NEMA TS 2 Clause 3.5.3.2.1.b.(1).(c)"

::= { phaseEntry 12 }

2.2.2.13 Phase Time Before Reduction Parameter

phaseTimeBeforeReduction OBJECT-TYPE

SYNTAX INTEGER (0..255)

ACCESS read-write

STATUS mandatory

DESCRIPTION

"Phase Time Before Reduction (TBR) Parameter in seconds (0-255 sec). The Time Before Reduction period shall begin when the phase is Green and there is a serviceable conflicting call. If the serviceable conflicting call is removed before completion of this time (or time to reduce), the timer shall reset. Upon completion of the TBR period or the CarsBeforeReduction (CBR) parameter is satisfied, whichever occurs first, the linear reduction of the allowable gap from the Passage Time shall begin."

REFERENCE

"NEMA TS 2 Clause 3.5.3.1 and 3.5.3.2.1.b.(2)"

::= { phaseEntry 13 }

2.2.2.14 Phase Cars Before Reduction Parameter

phaseCarsBeforeReduction OBJECT-TYPE

SYNTAX INTEGER (0..255)

ACCESS read-write

STATUS optional

DESCRIPTION

"Phase Cars Before Reduction (CBR) Parameter (0-255 vehicles). When the phase is Green and the sum of the cars waiting (vehicle actuations during Yellow & Red intervals) on serviceable conflicting phases equals or exceeds the CBR parameter or the Time Before Reduction (TBR) parameter is satisfied, whichever occurs first, the linear reduction of the allowable gap from the Passage Time shall begin."

::= { phaseEntry 14 }

2.2.2.15 Phase Time To Reduce Parameter

phaseTimeToReduce OBJECT-TYPE

SYNTAX INTEGER (0..255)

ACCESS read-write

STATUS mandatory

DESCRIPTION

"Phase Time To Reduce Parameter in seconds (0-255 sec). This parameter shall control the rate of reduction of the allowable gap between the Passage Time and Minimum Gap setting."

REFERENCE

"NEMA TS 2 Clause 3.5.3.1 and 3.5.3.2.1.b.(2)"

::= { phaseEntry 15 }

2.2.2.16 Phase Reduce By

phaseReduceBy OBJECT-TYPE

SYNTAX INTEGER (0..255)

ACCESS read-write

STATUS optional

DESCRIPTION

"This object may be used for volume density gap reduction as an alternate to the linear reduction defined by NEMA TS 1 and TS 2. It contains the tenths of seconds to reduce the gap by (0.0 - 25.5 seconds). The frequency of reduction shall produce the Minimum Gap after a time equal to the 'phaseTimeToReduce' object."

::= { phaseEntry 16 }

2.2.2.17 Phase Minimum Gap Parameter

phaseMinimumGap OBJECT-TYPE

SYNTAX INTEGER (0..255)

ACCESS read-write

STATUS mandatory

DESCRIPTION

"Phase Minimum Gap Parameter in tenth seconds (0-25.5 sec). The reduction of the allowable gap shall continue until the gap reaches a value equal to or less than the minimum gap as set on the Minimum Gap control after which the allowable gap shall remain fixed at the values set on the Minimum Gap control."

REFERENCE

"NEMA TS 2 Clause 3.5.3.1 and 3.5.3.2.1.b.(2)"

::= { phaseEntry 17 }

2.2.2.18 Phase Dynamic Max Limit

phaseDynamicMaxLimit OBJECT-TYPE

SYNTAX INTEGER (0..255)

ACCESS read-write

STATUS optional

DESCRIPTION

"This object shall determine either the upper or lower limit of the running max in seconds (0-255) during dynamic max operation.

The normal maximum (i.e. Max1, Max2, etc.) shall determine the other limit as follows:

When dynamicMaxLimit is larger than the normal maximum, it shall become the upper limit.

When dynamicMaxLimit is smaller than the normal maximum, it shall become the lower limit.

Setting dynamicMaxLimit greater than zero enables dynamic max operation with the normal maximum used as the initial maximum setting. See dynamicMaxStep for details on dynamic max operation.

Maximum recall or a failed detector that is assigned to the associated phase shall disable dynamic max operation for the phase."

::= { phaseEntry 18 }

2.2.2.19 Phase Dynamic Max Step

phaseDynamicMaxStep OBJECT-TYPE

SYNTAX INTEGER (0..255)

ACCESS read-write

STATUS optional

DESCRIPTION

"This object shall determine the automatic adjustment to the running max in tenth seconds (0-25.5).

When a phase maxes out twice in a row, and on each successive max out thereafter, one dynamic max step value shall be added to the running max until such addition would mean the running max was greater than the larger of normal max or dynamic max limit.

When a phase gaps out twice in a row, and on each successive gap out thereafter, one dynamic max step value shall be subtracted from the running max until such subtraction would mean the running max was less than the smaller of the normal max or the dynamic max limit.

If a phase gaps out in one cycle and maxes out in the next cycle, or vice versa, the running max will not change."

::= { phaseEntry 19 }

2.2.2.20 Phase Startup

phaseStartup OBJECT-TYPE

SYNTAX INTEGER { other (1),
phaseNotOn (2),
greenWalk (3),
greenNoWalk (4),
yellowChange (5),
redClear (6) }

ACCESS read-write

STATUS mandatory

DESCRIPTION

"The Phase Startup parameter is an enumerated integer which selects the startup state for each phase after restoration of a defined power interruption or activation of the external start input. The following entries are defined:

other; this phase initializes in a state not defined by this standard.

phaseNotOn; this phase initializes in a Red state (the phase is not active and no intervals are timing).

greenWalk; this phase initializes at the beginning of the minimum green and walk timing intervals.

greenNoWalk; this phase initializes at the beginning of the minimum green timing interval.

yellowChange; this phase initializes at the beginning of the Yellow Change interval.

redClear; this phase initializes at the beginning of the Red Clearance interval."

REFERENCE

"NEMA TS 2 Clause 3.5.5.1 and 3.5.5.12"

::= { phaseEntry 20 }

2.2.2.21 Phase Options

phaseOptions OBJECT-TYPE

SYNTAX INTEGER (0..65535)

ACCESS read-write

STATUS mandatory

DESCRIPTION

"Optional phase functions (0 = False/Disabled, 1 = True/Enabled)

Bit Description

- 0 Enabled Phase - provide a means to define whether this phase is used in the current configuration. A disabled phase shall not provide any outputs nor respond to any phase inputs.
- 1 Automatic Flash Entry Phase - When Automatic Flash is called, the CU shall service the Entry Phase(s), clear to an All Red, then initiate flashing operation. Support is optional.
REFERENCE NEMA TS 2 Clause 3.9.1.2.1
- 2 Automatic Flash Exit Phase - The CU shall move immediately to the beginning of the phase(s) programmed as Exit Phase(s) when Automatic Flash terminates. Support is optional
REFERENCE NEMA TS 2 Clause 3.9.1.2.1
- 3 Non-Actuated 1 - when set to 1 causes a phase to respond to the Call To Non-Actuated 1 input (if present) or other method. Support is optional
REFERENCE NEMA TS 2 Clause 3.5.5.5.8
- 4 Non-Actuated 2 - when set to 1 causes a phase to respond to the Call To Non-Actuated 2 input (if present) or other method. Support is optional
REFERENCE NEMA TS 2 Clause 3.5.5.5.8
- 5 Non Lock Detector Memory - when set to 0 will cause the call to be locked at the beginning of the yellow interval. When set to 1 call locking will depend on the detectorOptions object.
REFERENCE NEMA TS 2 Clause 3.5.3.4
- 6 Min. Vehicle Recall - when set to 1 causes recurring demand for vehicle service on the phase when that phase is not in its Green interval.
REFERENCE NEMA TS 2 Clause 3.5.3.6
- 7 Max Vehicle Recall - when set to 1 causes a call on a phase such that the timing of the Green interval for that phase shall be extended to Maximum Green time.
REFERENCE NEMA TS 2 Clause 3.5.3.5
- 8 Ped. Recall - when set to 1 causes a recurring pedestrian demand which shall function in the same manner as an external pedestrian call except that it shall not recycle the pedestrian service until a conflicting phase is serviced
REFERENCE NEMA TS 2 Clause 3.5.3.7
- 9 Soft Vehicle Recall - when set to 1 causes a call on a phase when all conflicting phases are in green dwell or red dwell and there are no serviceable conflicting calls. Support is optional.
- 10 Dual Entry Phase - in multi-ring configurations when set to 1 causes the phase to become active upon entry into a concurrency group (crossing a barrier) when no calls exist in its ring within its concurrency group.
REFERENCE NEMA TS 2 Clause 3.5.5.3
- 11 Simultaneous Gap Disable - in multi-ring configurations when set to 1 disables a gapped out phase from reverting to the extensible portion. Support is optional
REFERENCE NEMA TS 2 Clause 3.5.5.3
- 12 Guaranteed Passage - when set to 1 enables an actuated phase operating in volume density mode (using gap reduction) to retain the right of way for the unexpired portion of the Passage time following the decision to terminate the green due to a reduced gap. Support is optional
- 13 Actuated Rest In Walk - when set to 1 causes an actuated phase to rest in Walk when there is no serviceable conflicting call at the end of Walk Timing.

- 14 Conditional Service Enable - in multi-ring configurations when set to 1 causes a gapped/maxed phase to conditionally service a preceding actuated vehicle phase when sufficient time remains before max time out of the phase(s) not prepared to terminate. Support is optional.
REFERENCE NEMA TS 2 Clause 3.5.3.9
- 15 AddedInitialCalculation - If set (1) the CU shall compare counts from all associated AddedInitial detectors and use the largest count value for the calculations. If clear (0) the CU shall sum all associated AddedInitial detector counts and use this sum for the calculations. The ability to modify the setting of this bit is optional."

::= { phaseEntry 21 }

2.2.2.22 Phase Ring Parameter

phaseRing OBJECT-TYPE

SYNTAX INTEGER (1..255)

ACCESS read-write

STATUS mandatory

DESCRIPTION

"Phase ring number (1..maxRings) that identified the ring which contains the associated phase. This value must not exceed the maxRings object value."

::= { phaseEntry 22 }

2.2.2.23 Phase Concurrency

phaseConcurrency OBJECT-TYPE

SYNTAX OCTET STRING

ACCESS read-write

STATUS mandatory

DESCRIPTION

"Each octet contains a phase number (binary value) that may run concurrently with the associated phase. Phases that are contained in the same ring may NOT run concurrently."

::= { phaseEntry 23 }

2.2.3 Maximum Phase Groups

maxPhaseGroups OBJECT-TYPE

SYNTAX INTEGER (1..255)

ACCESS read-only

STATUS mandatory

DESCRIPTION

"The Maximum Number of Phase Groups (8 Phases per group) this Actuated Controller Unit supports. This value is equal to TRUNCATE $[(\text{maxPhases} + 7) / 8]$. This object indicates the maximum rows which shall appear in the phaseStatusGroupTable and phaseControlGroupTable."

::= { phase 3 }

2.2.4 Phase Status Group Table

phaseStatusGroupTable OBJECT-TYPE

SYNTAX SEQUENCE OF PhaseStatusGroupEntry

ACCESS not-accessible

STATUS mandatory

DESCRIPTION

"A table containing Actuated Controller Unit Phase Output (Red, Yellow, & Green) and Call (vehicle & pedestrian) status in groups of eight Phases. The number of rows in this table is equal to the maxPhaseGroups object."

::= { phase 4 }

phaseStatusGroupEntry OBJECT-TYPE
SYNTAX PhaseStatusGroupEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION
"Red, Yellow, & Green Output Status and Vehicle and Pedestrian Call for eight Actuated
Controller Unit Phases."
INDEX { phaseStatusGroupNumber }
 ::= { phaseStatusGroupTable 1 }

PhaseStatusGroupEntry ::= SEQUENCE {
phaseStatusGroupNumber INTEGER,
phaseStatusGroupReds INTEGER,
phaseStatusGroupYellows INTEGER,
phaseStatusGroupGreens INTEGER,
phaseStatusGroupDontWalks INTEGER,
phaseStatusGroupPedClears INTEGER,
phaseStatusGroupWalks INTEGER,
phaseStatusGroupVehCalls INTEGER,
phaseStatusGroupPedCalls INTEGER,
phaseStatusGroupPhaseOns INTEGER,
phaseStatusGroupPhaseNexts INTEGER }

2.2.4.1 Phase Status Group Number

phaseStatusGroupNumber OBJECT-TYPE
SYNTAX INTEGER (0..255)
ACCESS read-only
STATUS mandatory
DESCRIPTION
"The Phase StatusGroup number for objects in this row. This value shall not exceed the
maxPhaseGroups object value."
 ::= { phaseStatusGroupEntry 1 }

2.2.4.2 Phase Status Group Reds

phaseStatusGroupReds OBJECT-TYPE
SYNTAX INTEGER (0..255)
ACCESS read-only
STATUS mandatory
DESCRIPTION
"Phase Red Output Status Mask, when a bit = 1, the Phase Red is currently active. When a bit
= 0, the Phase Red is NOT currently active.
Bit 7 = Phase number = (phaseStatusGroupNumber * 8)
Bit 6 = Phase number = (phaseStatusGroupNumber * 8) - 1
Bit 5 = Phase number = (phaseStatusGroupNumber * 8) - 2
Bit 4 = Phase number = (phaseStatusGroupNumber * 8) - 3
Bit 3 = Phase number = (phaseStatusGroupNumber * 8) - 4
Bit 2 = Phase number = (phaseStatusGroupNumber * 8) - 5
Bit 1 = Phase number = (phaseStatusGroupNumber * 8) - 6
Bit 0 = Phase number = (phaseStatusGroupNumber * 8) - 7"
 ::= { phaseStatusGroupEntry 2 }

2.2.4.3 Phase Status Group Yellows

phaseStatusGroupYellows OBJECT-TYPE

SYNTAX INTEGER (0..255)

ACCESS read-only

STATUS mandatory

DESCRIPTION

"Phase Yellow Output Status Mask, when a bit = 1, the Phase Yellow is currently active.

When a bit = 0, the Phase Yellow is NOT currently active.

Bit 7 = Phase number = (phaseStatusGroupNumber * 8)

Bit 6 = Phase number = (phaseStatusGroupNumber * 8) - 1

Bit 5 = Phase number = (phaseStatusGroupNumber * 8) - 2

Bit 4 = Phase number = (phaseStatusGroupNumber * 8) - 3

Bit 3 = Phase number = (phaseStatusGroupNumber * 8) - 4

Bit 2 = Phase number = (phaseStatusGroupNumber * 8) - 5

Bit 1 = Phase number = (phaseStatusGroupNumber * 8) - 6

Bit 0 = Phase number = (phaseStatusGroupNumber * 8) - 7"

::= { phaseStatusGroupEntry 3 }

2.2.4.4 Phase Status Group Greens

phaseStatusGroupGreens OBJECT-TYPE

SYNTAX INTEGER (0..255)

ACCESS read-only

STATUS mandatory

DESCRIPTION

"Phase Green Output Status Mask, when a bit = 1, the Phase Green is currently active.

When a bit = 0, the Phase Green is NOT currently active.

Bit 7 = Phase number = (phaseStatusGroupNumber * 8)

Bit 6 = Phase number = (phaseStatusGroupNumber * 8) - 1

Bit 5 = Phase number = (phaseStatusGroupNumber * 8) - 2

Bit 4 = Phase number = (phaseStatusGroupNumber * 8) - 3

Bit 3 = Phase number = (phaseStatusGroupNumber * 8) - 4

Bit 2 = Phase number = (phaseStatusGroupNumber * 8) - 5

Bit 1 = Phase number = (phaseStatusGroupNumber * 8) - 6

Bit 0 = Phase number = (phaseStatusGroupNumber * 8) - 7"

::= { phaseStatusGroupEntry 4 }

2.2.4.5 Phase Status Group Dont Walks

phaseStatusGroupDontWalks OBJECT-TYPE

SYNTAX INTEGER (0..255)

ACCESS read-only

STATUS mandatory

DESCRIPTION

"Phase Dont Walk Output Status Mask, when a bit = 1, the Phase Dont Walk is currently active. When a bit = 0, the Phase Dont Walk is NOT currently active.

Bit 7 = Phase number = (phaseStatusGroupNumber * 8)

Bit 6 = Phase number = (phaseStatusGroupNumber * 8) - 1

Bit 5 = Phase number = (phaseStatusGroupNumber * 8) - 2

Bit 4 = Phase number = (phaseStatusGroupNumber * 8) - 3

Bit 3 = Phase number = (phaseStatusGroupNumber * 8) - 4

Bit 2 = Phase number = (phaseStatusGroupNumber * 8) - 5

Bit 1 = Phase number = (phaseStatusGroupNumber * 8) - 6

Bit 0 = Phase number = (phaseStatusGroupNumber * 8) - 7"

::= { phaseStatusGroupEntry 5 }

2.2.4.6 Phase Status Group Pedestrian clears

phaseStatusGroupPedClears OBJECT-TYPE

SYNTAX INTEGER (0..255)

ACCESS read-only

STATUS mandatory

DESCRIPTION

"Phase Ped. Clear Output Status Mask, when a bit = 1, the Phase Ped. Clear is currently active. When a bit = 0, the Phase Ped. Clear is NOT currently active.

Bit 7 = Phase number = (phaseStatusGroupNumber * 8)

Bit 6 = Phase number = (phaseStatusGroupNumber * 8) - 1

Bit 5 = Phase number = (phaseStatusGroupNumber * 8) - 2

Bit 4 = Phase number = (phaseStatusGroupNumber * 8) - 3

Bit 3 = Phase number = (phaseStatusGroupNumber * 8) - 4

Bit 2 = Phase number = (phaseStatusGroupNumber * 8) - 5

Bit 1 = Phase number = (phaseStatusGroupNumber * 8) - 6

Bit 0 = Phase number = (phaseStatusGroupNumber * 8) - 7"

::= { phaseStatusGroupEntry 6 }

2.2.4.7 Phase Status Group Walks

phaseStatusGroupWalks OBJECT-TYPE

SYNTAX INTEGER (0..255)

ACCESS read-only

STATUS mandatory

DESCRIPTION

"Phase Walk Output Status Mask, when a bit = 1, the Phase Walk is currently active. When a bit = 0, the Phase Walk is NOT currently active.

Bit 7 = Phase number = (phaseStatusGroupNumber * 8)

Bit 6 = Phase number = (phaseStatusGroupNumber * 8) - 1

Bit 5 = Phase number = (phaseStatusGroupNumber * 8) - 2

Bit 4 = Phase number = (phaseStatusGroupNumber * 8) - 3

Bit 3 = Phase number = (phaseStatusGroupNumber * 8) - 4

Bit 2 = Phase number = (phaseStatusGroupNumber * 8) - 5

Bit 1 = Phase number = (phaseStatusGroupNumber * 8) - 6

Bit 0 = Phase number = (phaseStatusGroupNumber * 8) - 7"

::= { phaseStatusGroupEntry 7 }

2.2.4.8 Phase Status Group Vehicle Calls

phaseStatusGroupVehCalls OBJECT-TYPE

SYNTAX INTEGER (0..255)

ACCESS read-only

STATUS mandatory

DESCRIPTION

"Phase Vehicle Call Status Mask, when a bit = 1, the Phase vehicle currently has a call for service. When a bit = 0, the Phase vehicle currently does NOT have a call for service.

Bit 7 = Phase number = (phaseStatusGroupNumber * 8)

Bit 6 = Phase number = (phaseStatusGroupNumber * 8) - 1

Bit 5 = Phase number = (phaseStatusGroupNumber * 8) - 2

Bit 4 = Phase number = (phaseStatusGroupNumber * 8) - 3

Bit 3 = Phase number = (phaseStatusGroupNumber * 8) - 4

Bit 2 = Phase number = (phaseStatusGroupNumber * 8) - 5

Bit 1 = Phase number = (phaseStatusGroupNumber * 8) - 6

Bit 0 = Phase number = (phaseStatusGroupNumber * 8) - 7"

::= { phaseStatusGroupEntry 8 }

2.2.4.9 Phase Status Group Pedestrian Calls

phaseStatusGroupPedCalls OBJECT-TYPE

SYNTAX INTEGER (0..255)

ACCESS read-only

STATUS mandatory

DESCRIPTION

"Phase Pedestrian Call Status Mask, when a bit = 1, the Phase pedestrian currently has a call for service. When a bit = 0, the Phase pedestrian currently does NOT have a call for service.

Bit 7 = Phase number = (phaseStatusGroupNumber * 8)

Bit 6 = Phase number = (phaseStatusGroupNumber * 8) - 1

Bit 5 = Phase number = (phaseStatusGroupNumber * 8) - 2

Bit 4 = Phase number = (phaseStatusGroupNumber * 8) - 3

Bit 3 = Phase number = (phaseStatusGroupNumber * 8) - 4

Bit 2 = Phase number = (phaseStatusGroupNumber * 8) - 5

Bit 1 = Phase number = (phaseStatusGroupNumber * 8) - 6

Bit 0 = Phase number = (phaseStatusGroupNumber * 8) - 7"

::= { phaseStatusGroupEntry 9 }

2.2.4.10 Phase Status Group Phase Ons

phaseStatusGroupPhaseOns OBJECT-TYPE

SYNTAX INTEGER (0..255)

ACCESS read-only

STATUS mandatory

DESCRIPTION

"Phase On Status Mask, when a bit = 1, the Phase is currently active. When a bit = 0, the Phase currently is NOT active. The phase is ON during the Green, Yellow, & Red Clearance intervals of that phase. It shall be permissible for this status to be True (bit=1) during the Red Dwell state.

Bit 7 = Phase number = (phaseStatusGroupNumber * 8)

Bit 6 = Phase number = (phaseStatusGroupNumber * 8) - 1

Bit 5 = Phase number = (phaseStatusGroupNumber * 8) - 2

Bit 4 = Phase number = (phaseStatusGroupNumber * 8) - 3

Bit 3 = Phase number = (phaseStatusGroupNumber * 8) - 4

Bit 2 = Phase number = (phaseStatusGroupNumber * 8) - 5

Bit 1 = Phase number = (phaseStatusGroupNumber * 8) - 6

Bit 0 = Phase number = (phaseStatusGroupNumber * 8) - 7"

::= { phaseStatusGroupEntry 10 }

2.2.4.11 Phase Status Group Phase Nexts

phaseStatusGroupPhaseNexts OBJECT-TYPE

SYNTAX INTEGER (0..255)

ACCESS read-only

STATUS mandatory

DESCRIPTION

"Phase Next Status Mask, when a bit = 1, the Phase currently is committed to be NEXT in sequence & remains present until the phase becomes active (On/Timing). When a bit = 0, the Phase currently is NOT committed to be NEXT in sequence. The phase next to be serviced shall be determined at the end of the green interval of the terminating phase; except that if the decision cannot be made at the end of the Green interval, it shall not be made until after the end of all Vehicle Change & Clearance intervals.

Bit 7 = Phase number = (phaseStatusGroupNumber * 8)
 Bit 6 = Phase number = (phaseStatusGroupNumber * 8) - 1
 Bit 5 = Phase number = (phaseStatusGroupNumber * 8) - 2
 Bit 4 = Phase number = (phaseStatusGroupNumber * 8) - 3
 Bit 3 = Phase number = (phaseStatusGroupNumber * 8) - 4
 Bit 2 = Phase number = (phaseStatusGroupNumber * 8) - 5
 Bit 1 = Phase number = (phaseStatusGroupNumber * 8) - 6
 Bit 0 = Phase number = (phaseStatusGroupNumber * 8) - 7"

::= { phaseStatusGroupEntry 11 }

2.2.5 Phase Control Table

phaseControlGroupTable OBJECT-TYPE

SYNTAX SEQUENCE OF PhaseControlGroupEntry

ACCESS not-accessible

STATUS optional

DESCRIPTION

"A table containing Actuated Controller Unit Phase Control in groups of eight phases. The number of rows in this table is equal to the maxPhaseGroups object.

This table is optional for Actuated Controller Units conforming to this specification. If implemented then all objects in this table shall be implemented."

::= { phase 5 }

phaseControlGroupEntry OBJECT-TYPE

SYNTAX PhaseControlGroupEntry

ACCESS not-accessible

STATUS mandatory

DESCRIPTION

"Phase Control for eight Actuated Controller Unit phases."

INDEX { phaseControlGroupNumber }

::= { phaseControlGroupTable 1 }

PhaseControlGroupEntry ::= SEQUENCE {

phaseControlGroupNumber	INTEGER,
phaseControlGroupPhaseOmit	INTEGER,
phaseControlGroupPedOmit	INTEGER,
phaseControlGroupHold	INTEGER,
phaseControlGroupForceOff	INTEGER,
phaseControlGroupVehCall	INTEGER,
phaseControlGroupPedCall	INTEGER }

2.2.5.1 Phase Control Group Number

phaseControlGroupNumber OBJECT-TYPE

SYNTAX INTEGER (1..255)

ACCESS read-only

STATUS mandatory

DESCRIPTION

"The Phase Control Group number for objects in this row. This value shall not exceed the maxPhaseGroups object value."

::= { phaseControlGroupEntry 1 }

2.2.5.2 Phase Omit Control

phaseControlGroupPhaseOmit OBJECT-TYPE

SYNTAX INTEGER (0..255)

ACCESS read-write

STATUS mandatory

DESCRIPTION

"This object is used to allow a remote entity to omit phases from being serviced in the device. When a bit = 1, the device shall activate the System Phase Omit control for that phase. When a bit = 0, the device shall not activate the System Phase Omit control for that phase.

Bit 7 = Phase number = (phaseControlGroupNumber * 8)

Bit 6 = Phase number = (phaseControlGroupNumber * 8) - 1

Bit 5 = Phase number = (phaseControlGroupNumber * 8) - 2

Bit 4 = Phase number = (phaseControlGroupNumber * 8) - 3

Bit 3 = Phase number = (phaseControlGroupNumber * 8) - 4

Bit 2 = Phase number = (phaseControlGroupNumber * 8) - 5

Bit 1 = Phase number = (phaseControlGroupNumber * 8) - 6

Bit 0 = Phase number = (phaseControlGroupNumber * 8) - 7

The device shall reset this object to ZERO when in BACKUP Mode. A write to this object shall reset the Backup timer to ZERO."

REFERENCE

"NEMA TS 2 Clause 3.5.3.11.2"

::= { phaseControlGroupEntry 2 }

2.2.5.3 Pedestrian Omit Control

phaseControlGroupPedOmit OBJECT-TYPE

SYNTAX INTEGER (0..255)

ACCESS read-write

STATUS mandatory

DESCRIPTION

"This object is used to allow a remote entity to omit peds from being serviced in the device. When a bit = 1, the device shall activate the System Ped Omit control for that phase. When a bit = 0, the device shall not activate the System Ped Omit control for that phase.

Bit 7 = Phase number = (phaseControlGroupNumber * 8)

Bit 6 = Phase number = (phaseControlGroupNumber * 8) - 1

Bit 5 = Phase number = (phaseControlGroupNumber * 8) - 2

Bit 4 = Phase number = (phaseControlGroupNumber * 8) - 3

Bit 3 = Phase number = (phaseControlGroupNumber * 8) - 4

Bit 2 = Phase number = (phaseControlGroupNumber * 8) - 5

Bit 1 = Phase number = (phaseControlGroupNumber * 8) - 6

Bit 0 = Phase number = (phaseControlGroupNumber * 8) - 7

The device shall reset this object to ZERO when in BACKUP Mode. A write to this object shall reset the Backup timer to ZERO."

REFERENCE

"NEMA TS 2 Clause 3.5.3.11.3"

::= { phaseControlGroupEntry 3 }

2.2.5.4 Phase Hold Control

phaseControlGroupHold OBJECT-TYPE

SYNTAX INTEGER (0..255)

ACCESS read-write

STATUS mandatory

DESCRIPTION

"This object is used to allow a remote entity to hold phases in the device. When a bit = 1, the device shall activate the System Phase Hold control for that phase. When a bit = 0, the device shall not activate the System Phase Hold control for that phase.

Bit 7 = Phase number = (phaseControlGroupNumber * 8)

Bit 6 = Phase number = (phaseControlGroupNumber * 8) - 1

Bit 5 = Phase number = (phaseControlGroupNumber * 8) - 2

Bit 4 = Phase number = (phaseControlGroupNumber * 8) - 3

Bit 3 = Phase number = (phaseControlGroupNumber * 8) - 4

Bit 2 = Phase number = (phaseControlGroupNumber * 8) - 5

Bit 1 = Phase number = (phaseControlGroupNumber * 8) - 6

Bit 0 = Phase number = (phaseControlGroupNumber * 8) - 7

The device shall reset this object to ZERO when in BACKUP Mode. A write to this object shall reset the Backup timer to ZERO."

REFERENCE

"NEMA TS 2 Clause 3.5.3.11.1"

::= { phaseControlGroupEntry 4 }

2.2.5.5 Phase Force Off Control

phaseControlGroupForceOff OBJECT-TYPE

SYNTAX INTEGER (0..255)

ACCESS read-write

STATUS optional

DESCRIPTION

"This object is used to apply force offs on a per phase basis. When a bit = 1, the device shall activate the System Phase Force Off control for that phase. When a bit = 0, the device shall not activate the System Phase Force Off control for that phase. When the phase green terminates, the associated bit shall be reset to 0.

Bit 7 = Phase number = (phaseControlGroupNumber * 8)

Bit 6 = Phase number = (phaseControlGroupNumber * 8) - 1

Bit 5 = Phase number = (phaseControlGroupNumber * 8) - 2

Bit 4 = Phase number = (phaseControlGroupNumber * 8) - 3

Bit 3 = Phase number = (phaseControlGroupNumber * 8) - 4

Bit 2 = Phase number = (phaseControlGroupNumber * 8) - 5

Bit 1 = Phase number = (phaseControlGroupNumber * 8) - 6

Bit 0 = Phase number = (phaseControlGroupNumber * 8) - 7

The device shall reset this object to ZERO when in BACKUP Mode. A write to this object shall reset the Backup timer to ZERO."

::= { phaseControlGroupEntry 5 }

2.2.5.6 Vehicle Call Control

phaseControlGroupVehCall OBJECT-TYPE

SYNTAX INTEGER (0..255)

ACCESS read-write

STATUS mandatory

DESCRIPTION

"This object is used to allow a remote entity to place calls for vehicle service in the device. When a bit = 1, the device shall place a call for vehicle service on that phase. When a bit = 0, the device shall not place a call for vehicle service on that phase.

Bit 7 = Phase number = (phaseControlGroupNumber * 8)

Bit 6 = Phase number = (phaseControlGroupNumber * 8) - 1

Bit 5 = Phase number = (phaseControlGroupNumber * 8) - 2

Bit 4 = Phase number = (phaseControlGroupNumber * 8) - 3

Bit 3 = Phase number = (phaseControlGroupNumber * 8) - 4

Bit 2 = Phase number = (phaseControlGroupNumber * 8) - 5

Bit 1 = Phase number = (phaseControlGroupNumber * 8) - 6

Bit 0 = Phase number = (phaseControlGroupNumber * 8) - 7

The device shall reset this object to ZERO when in BACKUP Mode. A write to this object shall reset the Backup timer to ZERO."

::= { phaseControlGroupEntry 6 }

2.2.5.7 Pedestrian Call Control

phaseControlGroupPedCall OBJECT-TYPE

SYNTAX INTEGER (0..255)

ACCESS read-write

STATUS mandatory

DESCRIPTION

"This object is used to allow a remote entity to place calls for ped service in the device. When a bit = 1, the device shall place a call for ped service on that phase. When a bit = 0, the device shall not place a call for ped service on that phase.

Bit 7 = Phase number = (phaseControlGroupNumber * 8)

Bit 6 = Phase number = (phaseControlGroupNumber * 8) - 1

Bit 5 = Phase number = (phaseControlGroupNumber * 8) - 2

Bit 4 = Phase number = (phaseControlGroupNumber * 8) - 3

Bit 3 = Phase number = (phaseControlGroupNumber * 8) - 4

Bit 2 = Phase number = (phaseControlGroupNumber * 8) - 5

Bit 1 = Phase number = (phaseControlGroupNumber * 8) - 6

Bit 0 = Phase number = (phaseControlGroupNumber * 8) - 7

The device shall reset this object to ZERO when in BACKUP Mode. A write to this object shall reset the Backup timer to ZERO."

::= { phaseControlGroupEntry 7 }

2.3 DETECTOR PARAMETERS

detector OBJECT IDENTIFIER

::= { asc 2 }

-- This defines a node for supporting detector objects.

2.3.1 Maximum Vehicle Detectors

maxVehicleDetectors OBJECT-TYPE

SYNTAX INTEGER (0..255)

ACCESS read-only

STATUS mandatory

DESCRIPTION

"The Maximum Number of Vehicle Detectors this Actuated Controller Unit supports. This object indicates the maximum rows which shall appear in the vehicleDetectorTable object."

::= { detector 1 }

2.3.2 Vehicle Detector Parameter Table

vehicleDetectorTable OBJECT-TYPE

SYNTAX SEQUENCE OF VehicleDetectorEntry

ACCESS not-accessible

STATUS mandatory

DESCRIPTION

"A table containing Actuated Controller Unit vehicle detector parameters. The number of rows in this table is equal to the maxVehicleDetectors object."

::= { detector 2 }

vehicleDetectorEntry OBJECT-TYPE

SYNTAX VehicleDetectorEntry

ACCESS not-accessible

STATUS mandatory

DESCRIPTION

"Parameters for a specific Actuated Controller Unit detector."

INDEX { vehicleDetectorNumber }

::= { vehicleDetectorTable 1 }

VehicleDetectorEntry ::= SEQUENCE {
vehicleDetectorNumber INTEGER ,
vehicleDetectorOptions INTEGER ,
vehicleDetectorCallPhase INTEGER ,
vehicleDetectorSwitchPhase INTEGER ,
vehicleDetectorDelay INTEGER ,
vehicleDetectorExtend INTEGER ,
vehicleDetectorQueueLimit INTEGER ,
vehicleDetectorNoActivity INTEGER ,
vehicleDetectorMaxPresence INTEGER ,
vehicleDetectorErraticCounts INTEGER ,
vehicleDetectorFailTime INTEGER ,
vehicleDetectorAlarms INTEGER ,
vehicleDetectorReportedAlarms INTEGER ,
vehicleDetectorReset INTEGER }

2.3.2.1 Vehicle Detector Number

vehicleDetectorNumber OBJECT-TYPE
SYNTAX INTEGER (1..255)
ACCESS read-only
STATUS mandatory
DESCRIPTION

"The vehicle detector number for objects in this row. The value shall not exceed the maxVehicleDetectors object value."

::= { vehicleDetectorEntry 1 }

2.3.2.2 Vehicle Detector Options Parameter

vehicleDetectorOptions OBJECT-TYPE
SYNTAX INTEGER (0..255)
ACCESS read-write
STATUS mandatory
DESCRIPTION

"Vehicle Detector Options Parameter as follows:

Bit	Function
7	Call - if set (1) the CU shall place a demand for vehicular service on the assigned phase when the phase is not timing the green interval.
6	Queue - if set (1) the CU shall extend the green interval of the assigned phase until a gap occurs (no actuation) or until the green has been active longer than the vehicleDetectorQueueLimit time. This is optional.
5	AddedInitial - if set (1) the CU shall accumulate detector actuation counts for use in the added initial calculations. Counts shall be accumulated from the beginning of the yellow interval to the beginning of the green interval.
4	Passage - if set (1) the CU shall maintain a reset to the associated phase passage timer for the duration of the detector actuation when the phase is green.
3	Red Lock Call - if set (1) the detector will lock a call to the assigned phase if an actuation occurs while the phase is not timing Green or Yellow. This mode is optional.
2	Yellow Lock Call - if set (1) the detector will lock a call to the assigned phase if an actuation occurs while the phase is not timing Green.
1	Occupancy Detector - if set (1) the detector collects data for the associated detector occupancy object(s). This capability may not be supported on all detector inputs to a device.
0	Volume Detector - if set (1) the detector collects data for the associated detector volume object(s). This capability may not be supported on all detector inputs to a device."

::= { vehicleDetectorEntry 2 }

2.3.2.3 Vehicle Detector Call Phase Parameter

vehicleDetectorCallPhase OBJECT-TYPE
SYNTAX INTEGER (0..255)
ACCESS read-write
STATUS mandatory
DESCRIPTION

"This object contains assigned phase number for the detector input associated with this row. The associated detector call capability is enabled when this object is set to a non-zero value. The value shall not exceed the value of maxPhases"

REFERENCE

"NEMA TS 2 Clause 3.5.5.5.4 and 3.5.5.5.5"

::= { vehicleDetectorEntry 4 }

2.3.2.4 Vehicle Detector Switch Phase Parameter

vehicleDetectorSwitchPhase OBJECT-TYPE
SYNTAX INTEGER (0..255)
ACCESS read-write
STATUS mandatory
DESCRIPTION
"Detector Switch Phase Parameter (i.e., Phase Number). The phase to which a vehicle detector actuation shall be switched when the assigned phase is Yellow or Red and the Switch Phase is Green"
REFERENCE
"NEMA TS 2 Clause 3.5.5.5.4.c"
::= { vehicleDetectorEntry 5 }

2.3.2.5 Vehicle Detector Delay Parameter

vehicleDetectorDelay OBJECT-TYPE
SYNTAX INTEGER (0..65535)
ACCESS read-write
STATUS mandatory
DESCRIPTION
"Detector Delay Parameter in tenth seconds (0–255.0 sec). The period a detector actuation (input recognition) shall be delayed when the phase is not Green"
REFERENCE
"NEMA TS 2 Clause 3.5.5.5.4.a"
::= { vehicleDetectorEntry 6 }

2.3.2.6 Vehicle Detector Extend Parameter

vehicleDetectorExtend OBJECT-TYPE
SYNTAX INTEGER (0..255)
ACCESS read-write
STATUS mandatory
DESCRIPTION
"Detector Extend Parameter in tenth seconds (0–25.5 sec). The period a vehicle detector actuation (input duration) shall be extended from the point of termination , when the phase is Green"
REFERENCE
"NEMA TS 2 Clause 3.5.5.5.4.b"
::= { vehicleDetectorEntry 7 }

2.3.2.7 Vehicle Detector Queue Limit

vehicleDetectorQueueLimit OBJECT-TYPE
SYNTAX INTEGER (0..255)
ACCESS read-write
STATUS optional
DESCRIPTION
"Detector Queue Limit parameter in seconds (0-255 sec). The length of time that an actuation from a queue detector may continue into the phase green. This time begins when the phase becomes green and when it expires any associated detector inputs shall be ignored. This time may be shorter due to other overriding device parameters (i.e. Maximum time, Force Off's, ...)."
::= { vehicleDetectorEntry 8 }

2.3.2.8 Vehicle Detector No Activity Parameter

vehicleDetectorNoActivity OBJECT-TYPE

SYNTAX INTEGER (0..255)

ACCESS read-write

STATUS mandatory

DESCRIPTION

"Detector No Activity diagnostic Parameter in minutes (0–255 min.) . If an active detector does not exhibit an actuation in the specified period, it is considered a fault by the diagnostics and the detector is classified as Failed. A value of 0 for this object shall disable this diagnostic for this detector."

REFERENCE

"NEMA TS 2 Clause 3.9.3.1.4.1"

::= { vehicleDetectorEntry 9 }

2.3.2.9 Vehicle Detector Maximum Presence Parameter

vehicleDetectorMaxPresence OBJECT-TYPE

SYNTAX INTEGER (0..255)

ACCESS read-write

STATUS mandatory

DESCRIPTION

"Detector Maximum Presence diagnostic Parameter in minutes (0-255 min.). If an active detector exhibits continuous detection for too long a period, it is considered a fault by the diagnostics and the detector is classified as Failed. A value of 0 for this object shall disable this diagnostic for this detector."

REFERENCE

"NEMA TS 2 Clause 3.9.3.1.4.2"

::= { vehicleDetectorEntry 10 }

2.3.2.10 Vehicle Detector Erratic Counts Parameter

vehicleDetectorErraticCounts OBJECT-TYPE

SYNTAX INTEGER (0..255)

ACCESS read-write

STATUS mandatory

DESCRIPTION

"Detector Erratic Counts diagnostic Parameter in counts/minute (0-255 cpm). If an active detector exhibits excessive actuations, it is considered a fault by the diagnostics and the detector is classified as Failed. A value of 0 for this object shall disable this diagnostic for this detector."

REFERENCE

"NEMA TS 2 Clause 3.9.3.1.4.3"

::= { vehicleDetectorEntry 11 }

2.3.2.11 Vehicle Detector Fail Time Parameter

vehicleDetectorFailTime OBJECT-TYPE

SYNTAX INTEGER (0..255)

ACCESS read-write

STATUS optional

DESCRIPTION

"Detector Fail Time in seconds (0..255). If a detector diagnostic indicates that the associated detector input is failed, then a call shall be placed on the associated phase during all non-green intervals.

When each green interval begins the call shall be maintained for the length of time specified by this object and then removed.

If the value of this object equals the maximum value (255) then a constant call shall be placed on the associated phase (max recall).

If the value of this object equals zero then no call shall be placed on the associated phase for any interval (no recall).

Compliant devices may support a limited capability for this object (i.e. only max recall or max recall and no recall). At a minimum the max recall setting must be supported."

::= { vehicleDetectorEntry 12 }

2.3.2.12 Vehicle Detector Alarms

vehicleDetectorAlarms OBJECT-TYPE

SYNTAX INTEGER (0..255)

ACCESS read-only

STATUS mandatory

DESCRIPTION

"This object shall return indications of detector alarms. Detector Alarms are indicated as follows:

- | Bit | Definition |
|-----|--|
| 0 | No Activity Fault: This detector has been flagged as non-operational due to lower than expected activity by the CU detector diagnostic. |
| 1 | Max Presence Fault: This detector has been flagged as non-operational due to a presence indicator that exceeded the maximum expected time by the CU detector diagnostic. |
| 2 | Erratic Output Fault: This detector has been flagged as non-operational due to erratic outputs (excessive counts) by the CU detector diagnostic. |
| 3 | Communications Fault: Communications to the device (if present) have failed. |
| 4 | Configuration Fault: Detector is assigned but is not supported. |
| 5-6 | Reserved. |
| 7 | Other Fault: The detector has failed due to some other cause. |

Once set a bit shall maintain its state as long as the condition exists. The bit shall clear when the condition no longer exists."

::= { vehicleDetectorEntry 13 }

2.3.2.13 Vehicle Detector Reported Alarms

vehicleDetectorReportedAlarms OBJECT-TYPE

SYNTAX INTEGER (0..255)

ACCESS read-only

STATUS optional

DESCRIPTION

"This object shall return detector device reported alarms (via some communications mechanism). Inductive Loop Detector Alarms are indicated as follows:

- | Bit | Definition |
|-----|--|
| 0 | Other |
| 1 | Watchdog Fault: This detector has been flagged as non-operational due to a watchdog error. |
| 2 | Open Loop Fault: This detector has been flagged as non-operational due to an open loop (broken wire). |
| 3 | Shorted Loop Fault: This detector has been flagged as non-operational due to a shorted loop wire. |
| 4 | Excessive Change Fault: This detector has been flagged as non-operational due to an inductance change that exceeded expected values. |
| 5-7 | Reserved |

Once set a bit shall maintain its state as long as the condition exists. The bit shall clear when the condition no longer exists."

::= { vehicleDetectorEntry 14 }

2.3.2.14 Vehicle Detector Reset

vehicleDetectorReset OBJECT-TYPE

SYNTAX INTEGER (0..1)

ACCESS read-write

STATUS mandatory

DESCRIPTION

"This object when set to TRUE (non-zero) shall cause the CU to command the associated detector to reset. This object shall automatically return to FALSE (zero) after the CU has issued the reset command.

NOTE: this may affect other detector (detector channels) that are physically attached to a common reset line."

::= { vehicleDetectorEntry 15 }

2.3.3 Maximum Vehicle Detector Status Groups

maxVehicleDetectorStatusGroups OBJECT-TYPE

SYNTAX INTEGER (0..255)

ACCESS read-only

STATUS mandatory

DESCRIPTION

"The maximum number of detector status groups (8 detectors per group) this device supports. This value is equal to TRUNCATE [(maxVehicleDetectors + 7) / 8]. This object indicates the maximum number of rows which shall appear in the vehicleDetectorStatusGroupTable object."

::= { detector 3 }

2.3.4 Vehicle Detector Status Group Table

vehicleDetectorStatusGroupTable OBJECT-TYPE

SYNTAX SEQUENCE OF VehicleDetectorStatusGroupEntry

ACCESS not-accessible

STATUS mandatory

DESCRIPTION

"A table containing detector status in groups of eight detectors. The number of rows in this table is equal to the maxVehicleDetectorStatusGroups object."

::= { detector 4 }

vehicleDetectorStatusGroupEntry OBJECT-TYPE

SYNTAX VehicleDetectorStatusGroupEntry

ACCESS not-accessible

STATUS mandatory

DESCRIPTION

"A group (row) of detector status."

INDEX { vehicleDetectorNumber }

::= { vehicleDetectorStatusGroupTable 1 }

VehicleDetectorStatusGroupEntry ::= SEQUENCE {

vehicleDetectorStatusGroupNumber

INTEGER ,

vehicleDetectorStatusGroupActive

INTEGER ,

vehicleDetectorStatusGroupAlarms

INTEGER }

2.3.4.1 Detector Status Group Number

vehicleDetectorStatusGroupNumber OBJECT-TYPE

SYNTAX INTEGER (1..255)

ACCESS read-only

STATUS mandatory

DESCRIPTION

"The detector status group number for objects in this row. This value shall not exceed the maxVehicleDetectorStatusGroups object value."

::= { vehicleDetectorStatusGroupEntry 1 }

2.3.4.2 Detector Status Group Active

vehicleDetectorStatusGroupActive OBJECT-TYPE

SYNTAX INTEGER (0..255)

ACCESS read-only

STATUS mandatory

DESCRIPTION

"This object shall return the detection status of each detector associated with the group. Each detector shall be represented as ON (detect) or OFF (no-detect) by individual bits in this object. If a detector is ON then the associated bit shall be set (1). If a detector is OFF then the associated bit shall be clear (0).

Bit Description

7 Detector number = (vehicleDetectorStatusGroupNumber * 8)

6 Detector number = (vehicleDetectorStatusGroupNumber * 8) - 1

5 Detector number = (vehicleDetectorStatusGroupNumber * 8) - 2

4 Detector number = (vehicleDetectorStatusGroupNumber * 8) - 3

3 Detector number = (vehicleDetectorStatusGroupNumber * 8) - 4

2 Detector number = (vehicleDetectorStatusGroupNumber * 8) - 5

1 Detector number = (vehicleDetectorStatusGroupNumber * 8) - 6

0 Detector number = (vehicleDetectorStatusGroupNumber * 8) - 7"

::= { vehicleDetectorStatusGroupEntry 2 }

2.3.4.3 Detector Alarm Status

vehicleDetectorStatusGroupAlarms OBJECT-TYPE

SYNTAX INTEGER (0..255)

ACCESS read-only

STATUS mandatory

DESCRIPTION

"This object shall return the alarm status of the detectors associated with the group. Each detector alarm status shall be represented as ON or OFF by individual bits in this object. If any detector alarm (defined in the vehicleDetectorAlarm object) is active the associated bit shall be set (1). If a detector alarm is not active the associated bit shall be clear (0).

Bit Description

7 Detector number = (vehicleDetectorStatusGroupNumber * 8)

6 Detector number = (vehicleDetectorStatusGroupNumber * 8) - 1

5 Detector number = (vehicleDetectorStatusGroupNumber * 8) - 2

4 Detector number = (vehicleDetectorStatusGroupNumber * 8) - 3

3 Detector number = (vehicleDetectorStatusGroupNumber * 8) - 4

2 Detector number = (vehicleDetectorStatusGroupNumber * 8) - 5

1 Detector number = (vehicleDetectorStatusGroupNumber * 8) - 6

0 Detector number = (vehicleDetectorStatusGroupNumber * 8) - 7"

::= { vehicleDetectorStatusGroupEntry 3 }

2.3.5 Volume / Occupancy report

volumeOccupancyReport OBJECT IDENTIFIER
::= { detector 5 }

-- This node contains the objects necessary to support volume / occupancy reporting .

2.3.5.1 Volume / Occupancy Sequence

volumeOccupancySequence OBJECT-TYPE
SYNTAX INTEGER (0..255)
ACCESS read-only
STATUS mandatory
DESCRIPTION

"This object defines a Sequence Number for Volume / Occupancy data collection. This object is used to detect duplicate or missing reports. The value cycles within the limits of 0 to 255. This object is incremented by one at the expiration of the volumeOccupancyPeriod time."

::= { volumeOccupancyReport 1 }

2.3.5.2 Volume / Occupancy Period

volumeOccupancyPeriod OBJECT-TYPE
SYNTAX INTEGER (0..255)
ACCESS read-write
STATUS mandatory
DESCRIPTION

"This object defines the number of seconds (0-255) that comprise the volume / occupancy collection period. When the collection period expires the device shall increment the volumeOccupancySequence, update the volumeOccupancyTable entries and reset the volume occupancy timer."

::= { volumeOccupancyReport 2 }

2.3.5.3 Active Volume / Occupancy Detectors

activeVolumeOccupancyDetectors OBJECT-TYPE
SYNTAX INTEGER (0..255)
ACCESS read-only
STATUS mandatory
DESCRIPTION

"The number of detectors in this device. This object indicates how many rows are in the volumeOccupancyTable object. There shall be a row for every detector that is collecting volume or occupancy data (refer to detectorOptions in the detectorTable)."

::= { volumeOccupancyReport 3 }

2.3.5.4 Volume / Occupancy Table

volumeOccupancyTable OBJECT-TYPE
SYNTAX SEQUENCE OF VolumeOccupancyEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION

"A table containing Detector Volume and Occupancy data collected. The number of rows in this table is equal to the activeVolumeOccupancyDetectors object."

::= { volumeOccupancyReport 4 }

volumeOccupancyEntry OBJECT-TYPE
SYNTAX VolumeOccupancyEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION
"The Volume and Occupancy data collected for one of the detectors in the device."
INDEX { vehicleDetectorNumber }
 ::= { volumeOccupancyTable 1 }

VolumeOccupancyEntry ::= SEQUENCE {
detectorVolume INTEGER,
detectorOccupancy INTEGER }

2.3.5.4.1 Volume data

detectorVolume OBJECT-TYPE
SYNTAX INTEGER (0..255)
ACCESS read-only
STATUS mandatory
DESCRIPTION
"Detector Volume data collected over the Volume / Occupancy Period. This value shall range from 0 to 254 indicating the volume of traffic crossing the associated detectorNumber during the collection period.

The value 255 shall indicate volume overflow."
 ::= { volumeOccupancyEntry 1 }

2.3.5.4.2 Occupancy data

detectorOccupancy OBJECT-TYPE
SYNTAX INTEGER (0..255)
ACCESS read-only
STATUS mandatory
DESCRIPTION
"Detector Occupancy data collected over the Volume / Occupancy Period or Detector Unit Diagnostic Information. The value of the object shall indicate occupancy or detector diagnostic information as follows:

Range	Meaning
0-200	Detector Occupancy in 0.5% Increments
201-209	Reserved
210	Max Presence Fault
211	No Activity Fault
212	Open loop Fault
213	Shorted loop Fault
214	Excessive Change Fault
215	Reserved
216	Watchdog Fault
217	Erratic Output Fault
218-255	Reserved

Faults shall be indicated for all collection periods during which a fault is detected if either occupancy data or volume data is being collected. The highest numbered fault shall be presented if more than one fault is active (i.e. indicate OpenLoop rather than NoActivity)."
 ::= { volumeOccupancyEntry 2 }

2.3.6 Maximum Pedestrian Detectors

maxPedestrianDetectors OBJECT-TYPE

SYNTAX INTEGER (0..255)

ACCESS read-only

STATUS mandatory

DESCRIPTION

"The Maximum Number of Pedestrian Detectors this Actuated Controller Unit supports. This object indicates the maximum rows which shall appear in the pedestrianDetectorTable object."

::= { detector 6 }

2.3.7 Pedestrian Detector Parameter Table

pedestrianDetectorTable OBJECT-TYPE

SYNTAX SEQUENCE OF PedestrianDetectorEntry

ACCESS not-accessible

STATUS mandatory

DESCRIPTION

"A table containing Actuated Controller Unit pedestrian detector parameters. The number of rows in this table is equal to the maxPedestrianDetectors object."

::= { detector 7 }

pedestrianDetectorEntry OBJECT-TYPE

SYNTAX PedestrianDetectorEntry

ACCESS not-accessible

STATUS mandatory

DESCRIPTION

"Parameters for a specific Actuated Controller Unit pedestrian detector."

INDEX { pedestrianDetectorNumber }

::= { pedestrianDetectorTable 1 }

PedestrianDetectorEntry ::= SEQUENCE {

pedestrianDetectorNumber INTEGER,

pedestrianDetectorCallPhase INTEGER,

pedestrianDetectorNoActivity INTEGER,

pedestrianDetectorMaxPresence INTEGER,

pedestrianDetectorErraticCounts INTEGER,

pedestrianDetectorAlarms INTEGER }

2.3.7.1 Pedestrian Detector Number

pedestrianDetectorNumber OBJECT-TYPE

SYNTAX INTEGER (1..255)

ACCESS read-only

STATUS mandatory

DESCRIPTION

"The pedestrianDetector number for objects in this row. The value shall not exceed the maxPedestrianDetectors object value."

::= { pedestrianDetectorEntry 1 }

2.3.7.2 Pedestrian Detector Call Phase Parameter

pedestrianDetectorCallPhase OBJECT-TYPE

SYNTAX INTEGER (0..255)

ACCESS read-write

STATUS mandatory

DESCRIPTION

"This object contains assigned phase number for the pedestrian detector input associated with this row. The associated detector call capability is enabled when this object is set to a non-zero value. The value shall not exceed the value of maxPhases."

::= { pedestrianDetectorEntry 2 }

2.3.7.3 Pedestrian Detector No Activity Parameter

pedestrianDetectorNoActivity OBJECT-TYPE

SYNTAX INTEGER (0..255)

ACCESS read-write

STATUS mandatory

DESCRIPTION

"Pedestrian Detector No Activity diagnostic Parameter in minutes (0–255 min.) . If an active detector does not exhibit an actuation in the specified period, it is considered a fault by the diagnostics and the detector is classified as Failed. A value of 0 for this object shall disable this diagnostic for this detector."

REFERENCE

"NEMA TS 2 Clause 3.9.3.1.4.1"

::= { pedestrianDetectorEntry 3 }

2.3.7.4 Pedestrian Detector Maximum Presence Parameter

pedestrianDetectorMaxPresence OBJECT-TYPE

SYNTAX INTEGER (0..255)

ACCESS read-write

STATUS mandatory

DESCRIPTION

"Pedestrian Detector Maximum Presence diagnostic Parameter in minutes (0-255 min.). If an active detector exhibits continuous detection for too long a period, it is considered a fault by the diagnostics and the detector is classified as Failed. A value of 0 for this object shall disable this diagnostic for this detector."

REFERENCE

"NEMA TS 2 Clause 3.9.3.1.4.2"

::= { pedestrianDetectorEntry 4 }

2.3.7.5 Pedestrian Detector Erratic Counts Parameter

pedestrianDetectorErraticCounts OBJECT-TYPE

SYNTAX INTEGER (0..255)

ACCESS read-write

STATUS mandatory

DESCRIPTION

"Pedestrian Detector Erratic Counts diagnostic Parameter in counts/minute (0-255 cpm). If an active detector exhibits excessive actuations, it is considered a fault by the diagnostics and the detector is classified as Failed. A value of 0 for this object shall disable this diagnostic for this detector."

REFERENCE

"NEMA TS 2 Clause 3.9.3.1.4.3"

::= { pedestrianDetectorEntry 5 }

2.3.7.6 Pedestrian Detector Alarms

pedestrianDetectorAlarms OBJECT-TYPE

SYNTAX INTEGER (0..255)

ACCESS read-only

STATUS mandatory

DESCRIPTION

"This object shall return indications of detector alarms. Detector Alarms are indicated as follows:

Bit Definition

0 No Activity Fault: This detector has been flagged as non-operational due to lower than expected activity by the CU detector diagnostic.

1 Max Presence Fault: This detector has been flagged as non-operational due to a presence indicator that exceeded the maximum expected time by the CU detector diagnostic.

2 Erratic Output Fault: This detector has been flagged as non-operational due to erratic outputs (excessive counts) by the CU detector diagnostic.

3 Communications Fault: Communications to the device (if present) have failed.

4 Configuration Fault: Detector is assigned but is not supported.

5-6 Reserved.

7 Other Fault: The detector has failed due to some other cause.

Once set a bit shall maintain its state as long as the condition exists. The bit shall clear when the condition no longer exists."

::= { pedestrianDetectorEntry 6 }

2.4 UNIT PARAMETERS

unit OBJECT IDENTIFIER

::= { asc 3 }

--"This defines a node for supporting unit objects."

2.4.1 StartUp Flash Parameter

unitStartUpFlash OBJECT-TYPE

SYNTAX INTEGER (0..255)

ACCESS read-write

STATUS mandatory

DESCRIPTION

"Unit Start up Flash time parameter in seconds (0 to 255 sec). The period/state (Start-Up Flash occurs when power is restored following a device defined power interruption. During the Start-Up Flash state, the Fault Monitor and Voltage Monitor outputs shall be inactive (if present)."

REFERENCE

"NEMA TS 2 Clause 3.9.1.1"

::= { unit 1 }

2.4.2 Automatic Ped Clear Parameter

unitAutoPedestrianClear OBJECT-TYPE
SYNTAX INTEGER { disable(1),
enable (2) }
ACCESS read-write
STATUS mandatory
DESCRIPTION

"Unit Automatic Ped Clear parameter (1 = False/Disable 2=True/Enable). When enabled, the CU shall time the Pedestrian Clearance interval when Manual Control Enable is active and prevent the Pedestrian Clearance interval from being terminated by the Interval Advance input."

REFERENCE
"NEMA TS 2 Clause 3.5.3.10"

::= { unit 2 }

2.4.3 Backup Time Parameter

unitBackupTime OBJECT-TYPE
SYNTAX INTEGER (0..65535)
ACCESS read-write
STATUS mandatory
DESCRIPTION

"The Backup Time in seconds (0-65535). When one of the defined system control parameters is SET, the backup timer is reset and times the unitBackupTime interval. If the unitBackupTime interval expires without a SET operation to one of the system control parameters, then the CU shall revert to Backup Mode.

A value of zero (0) shall disable this feature.

The system control parameters are:

phaseControlGroupPhaseOmit, phaseControlGroupPedOmit, phaseControlGroupHold, phaseControlGroupForceOff, phaseControlGroupVehCall, phaseControlGroupPedCall, unitControl, systemPatternControl, systemSyncControl, preemptControlState, ringControlGroupStopTime, ringControlGroupForceOff, ringControlGroupMax2, ringControlGroupMaxInhibit, ringControlGroupPedRecycle, ringControlGroupRedRest, ringControlGroupOmitRedClear and unitControl."

::= { unit 3 }

2.4.4 Unit Red Revert Parameter

unitRedRevert OBJECT-TYPE
SYNTAX INTEGER (0..255)
ACCESS read-write
STATUS mandatory
DESCRIPTION

"The red revert in tenth seconds (0.0 - 25.5 seconds). This value shall provide the minimum red revert time for all phases (i.e. if it is greater than a phaseRedRevert object value, then this value shall be used as the red revert time for the affected phase).

This object provides a minimum Red indication following the Yellow Change interval and prior to the next display of Green on the same signal output driver group."

::= { unit 4 }

2.4.5 Unit Control Status

unitControlStatus OBJECT-TYPE

SYNTAX INTEGER { other (1),
systemControl (2),
systemStandby (3),
backupMode(4),
manual (5),
timebase (6),
interconnect (7),
interconnectBackup (8)}

ACCESS read-only

STATUS mandatory

DESCRIPTION

"The Control Mode for Pattern, Flash, or Free at the device:

systemControl; control by master or central commands.

systemStandby; control by local based on master or central command to use local control.

backupMode; Backup Mode (see Terms).

Manual; control by entry other than zero in coordOperationalMode.

timebase; control by the local Time Base.

interconnect; control by the local Interconnect inputs.

interconnectBackup; control by local TBC due to invalid Interconnect inputs or loss of sync.

other; control by a source other than those listed above.

The value of this object is ignored when in BACKUP Mode.

A write to this object shall reset the BACKUP timer to ZERO."

::= { unit 5 }

2.4.6 Unit Flash Status

unitFlashStatus OBJECT-TYPE

SYNTAX INTEGER { other(1),
notFlash(2),
automatic(3),
localManual(4),
faultMonitor(5),
mmu(6),
startup(7) ,
preempt (8)}.

ACCESS read-only

STATUS mandatory

DESCRIPTION

"The Flash modes:

notFlash; the CU is not in Flash

automatic; the CU is currently in an Automatic Flash state.

localManual; the Controller Unit Local Flash input is active, MMU Flash input is not active, and Flash is not commanded by the Master.

faultMonitor; the CU is currently in a Fault Monitor State.

mmu; the Controller Unit MMU Flash input is active and the CU is not in Start-Up Flash.

startup; the CU is currently timing the Start-Up Flash period.

preempt; the CU is currently timing the preempt Flash.

other; the CU is in flash for some other reason."

::= { unit 6 }

2.4.7 Unit Alarm Status 2

unitAlarmStatus2 OBJECT-TYPE
SYNTAX INTEGER (0..255)
ACCESS read-only
STATUS mandatory
DESCRIPTION

"Device Alarm Mask 2 (0 = False, 1 = True) as follows:

- Bit 7 - Reserved.
- Bit 6 - Reserved.
- Bit 5 - Reserved.
- Bit 4 - Stop Time - When either CU Stop Time Input becomes active.
- Bit 3 - External Start - When the CU External Start becomes active.
- Bit 2 - Response Fault - When any NEMA TS2 Port 1 response frame fault occurs.
- Bit 1 - Low Battery - When any battery voltage falls below the required level.
- Bit 0 - Power Restart - When power returns after a power interruption.

Once set, a bit shall maintain it's state as long as the condition exists."

::= { unit 7 }

2.4.8 Unit Alarm Status 1

unitAlarmStatus1 OBJECT-TYPE
SYNTAX INTEGER (0..255)
ACCESS read-only
STATUS mandatory
DESCRIPTION

"Device Alarm Mask 1 (0 = False, 1 = True) as follows:

- Bit 7 - CoordActive - When coordination is active and not preempted or overridden.
- Bit 6 - Local Free - When any of the Controller Unit inputs and/or programming cause it to not respond to coordination control.
- Bit 5 - Local Flash - When the Controller Unit Local Flash input becomes active, MMU Flash input is not active, and Flash is not commanded by the system.
- Bit 4 - MMU Flash - When the Controller Unit MMU Flash input remains active for a period of time exceeding the Start-Up Flash time.
- Bit 3 - Cycle Fail - When a local Controller Unit is operating in the non-coordinated mode, whether the result of a Cycle Fault or Free being the current normal mode, and cycling diagnostics indicate that a serviceable call exists that has not been serviced for two cycles.
- Bit 2 - Coord Fail - When a Coord Fault is in effect and a Cycle Fault occurs again within two cycles of the coordination retry.
- Bit 1 - Coord Fault - When a Cycle Fault is in effect and the serviceable call has been serviced within two cycles after the Cycle Fault.
- Bit 0 - Cycle Fault - When the Controller Unit is operating in the coordinated mode and cycling diagnostics indicate that a serviceable call exists that has not been serviced for two cycles.

Once set, a bit shall maintain it's state as long as the condition exists."

::= { unit 8 }

2.4.9 Short Alarm Status

shortAlarmStatus OBJECT-TYPE
SYNTAX INTEGER (0..255)
ACCESS read-only
STATUS mandatory
DESCRIPTION

"Short Alarm Mask (0 = False, 1 = True) as follows:

- Bit 7 - Critical Alarm; When the Stop Time input is active.
 - Bit 6 - Non-Critical Alarm; When an physical alarm input is active.
 - Bit 5 - Detector Fault; When any detectorAlarm fault occurs.
 - Bit 4 - Coordination Alarm; When the CU is not running the called pattern without offset correction within three cycles of the command. An offset correction requiring less than three cycles due to cycle overrun caused by servicing a pedestrian call shall not cause a Coordination Alarm.
 - Bit 3 - Local Override; When any external input or CU programming has prevented the device from responding to a system pattern command.
 - Bit 2 - Local Cycle Zero; When running coordinated and the local coord cycle timer has passed through zero.
 - Bit 1 - T&F Flash; When either the Local Flash or MMU Flash input becomes active.
 - Bit 0 - Preempt - When any of the CU Preempt inputs become active.
- Once set, a bit shall maintain it's state as long as the condition exists."

::= { unit 9 }

2.4.10 Unit Control

unitControl OBJECT-TYPE

SYNTAX INTEGER (0..255)

ACCESS read-write

STATUS mandatory

DESCRIPTION

"This object is used to allow a remote entity to activate unit functions in the device (0 = False / Disabled, 1 = True / Enabled) as follows:

Bit 7 = Dimming Enable - when set to 1, causes channel dimming to operate as configured.

REFERENCE NEMA TS 2 Clause 3.9.2

Bit 6 = Interconnect - when set to 1, shall cause the interconnect inputs to operate at a higher priority than the timebase control (TBC On Line).

REFERENCE NEMA TS 2 Clause 3.6.2.3 and 3.8.3

Bit 5 = Walk Rest Modifier - when set to 1, causes non-actuated phases to remain in the timed-out Walk state (rest in Walk) in the absence of a serviceable conflicting call.

REFERENCE NEMA TS 2 Clause 3.5.5.5.13

Bit 4 = Call to Non-Actuated 2 - when set to 1, causes any phase(s) appropriately programmed in the phaseOptions object to operate in the Non-Actuated Mode.

REFERENCE NEMA TS 2 Clause 3.5.5.5.8

Bit 3 = Call to Non-Actuated 1 - when set to 1, causes any phase(s) appropriately programmed in the phaseOptions object to operate in the Non-Actuated Mode.

REFERENCE NEMA TS 2 Clause 3.5.5.5.8

Bit 2 = External Minimum Recall - when set to 1, causes a recurring demand on all vehicle phases for a minimum vehicle service.

REFERENCE NEMA TS 2 Clause 3.5.5.5.9

Bit 1 = Reserved

Bit 0 = Reserved

When a bit = 1, the device shall activate the Unit control. When a bit = 0, the device shall not activate the Unit control.

The device shall reset this object to ZERO when in BACKUP Mode. A write to this object shall reset the BACKUP timer."

::= { unit 10 }

2.4.11 Maximum Alarm Groups

maxAlarmGroups OBJECT-TYPE
SYNTAX INTEGER(0..255)
ACCESS read-only
STATUS optional
DESCRIPTION

"This object contains the maximum number of alarm groups (8 alarm inputs per group) this device supports. This object indicates the maximum rows which shall appear in the alarmGroupTable object."

::= { unit 11 }

2.4.12 Alarm Group Table

alarmGroupTable OBJECT-TYPE
SYNTAX SEQUENCE OF AlarmGroupEntry
ACCESS not-accessible
STATUS optional
DESCRIPTION

"This table contains alarm input status in groups of eight inputs. The number of rows in this table is equal to the maxAlarmGroups object."

::= { unit 12 }

alarmGroupEntry OBJECT-TYPE
SYNTAX AlarmGroupEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION
"Status for eight alarm inputs."

INDEX { alarmGroupNumber }

::= { alarmGroupTable 1 }

AlarmGroupEntry ::= SEQUENCE {
alarmGroupNumber INTEGER,
alarmGroupState INTEGER}

2.4.12.1 Alarm Group Number

alarmGroupNumber OBJECT-TYPE
SYNTAX INTEGER (0..255)
ACCESS read-only
STATUS mandatory
DESCRIPTION

"The alarm group number for objects in this row. This value shall not exceed the maxAlarmGroups object value."

::= { alarmGroupEntry 1 }

2.4.12.2 Alarm Group State

alarmGroupState OBJECT-TYPE
SYNTAX INTEGER (0..255)
ACCESS read-only
STATUS mandatory
DESCRIPTION

"Alarm input state bit field. When a bit = 1, the associated physical alarm input is active. When a bit = 0, the associated alarm input is NOT active."

Bit 7 = Alarm Input number = (alarmGroupNumber * 8)
Bit 6 = Alarm Input number = (alarmGroupNumber * 8) -1
Bit 5 = Alarm Input number = (alarmGroupNumber * 8) -2
Bit 4 = Alarm Input number = (alarmGroupNumber * 8) -3
Bit 3 = Alarm Input number = (alarmGroupNumber * 8) -4
Bit 2 = Alarm Input number = (alarmGroupNumber * 8) -5
Bit 1 = Alarm Input number = (alarmGroupNumber * 8) -6
Bit 0 = Alarm Input number = (alarmGroupNumber * 8) -7"

::= {alarmGroupEntry 2 }

2.4.13 Maximum Special Function Outputs

maxSpecialFunctionOutputs OBJECT-TYPE

SYNTAX INTEGER (0..255)

ACCESS read-only

STATUS mandatory

DESCRIPTION

"The Maximum Number of Special Functions this Actuated Controller Unit supports."

::= { unit 13 }

2.4.14 Special Function Output Table

specialFunctionOutputTable OBJECT-TYPE

SYNTAX SEQUENCE OF SpecialFunctionOutputEntry

ACCESS not-accessible

STATUS optional

DESCRIPTION

"A table containing Actuated Controller Unit special function output objects. The number of rows in this table is equal to the maxSpecialFunctionOutputs object."

::= { unit 14 }

specialFunctionOutputEntry OBJECT-TYPE

SYNTAX SpecialFunctionOutputEntry

ACCESS not-accessible

STATUS mandatory

DESCRIPTION

"Control for Actuated Controller Unit system special functions."

INDEX { specialFunctionOutputNumber }

::= { specialFunctionOutputTable 1 }

SpecialFunctionOutputEntry ::= SEQUENCE {

specialFunctionOutputNumber

INTEGER,

specialFunctionOutputState

INTEGER }

2.4.14.1 Special Function Output Number

specialFunctionOutputNumber OBJECT-TYPE

SYNTAX INTEGER (0..255)

ACCESS read-write

STATUS mandatory

DESCRIPTION

"The special function output number associated with the specialFunctionOutputState object in this row. This value shall not exceed the maxSpecialFunctionOutputs object value."

::= { specialFunctionOutputEntry 1 }

2.4.14.2 Special Function Output Control

specialFunctionOutputState OBJECT-TYPE

SYNTAX INTEGER (0..1)

ACCESS read-write

STATUS mandatory

DESCRIPTION

"The special function output (logical or physical) on the device may be controlled by this object. When this object is non-zero then the associated special function output signal shall be ON. When this object is zero then the associated special function output signal shall be OFF. A read of this object shall reflect the current state of the special function output."

::= { specialFunctionOutputEntry 2 }

2.5 COORDINATION PARAMETERS

coord OBJECT IDENTIFIER

::= { asc 4 }

-- The coord node contains objects that support coordination configuration, status and control functions for -- the device.

2.5.1 Coord Operational Mode Parameter

coordOperationalMode OBJECT-TYPE

SYNTAX INTEGER (0..255)

ACCESS read-write

STATUS mandatory

DESCRIPTION

"This object defines the operational mode for coordination. The possible modes are:

Value Description

0 Automatic - this mode provides for coord operation, free, and flash to be determined automatically by the possible sources (i.e. Interconnect, Time Base, or System Commands).

1-253 Manual Pattern - these modes provides for Coord operation running this pattern. This selection of pattern overrides all other pattern commands.

254 Manual Free - this mode provides for Free operation without coordination or Automatic Flash from any source.

255 Manual Flash - this mode provides for Automatic Flash without coordination or Free from any source."

REFERENCE

"NEMA TS 2 Clause 3.6.2.4"

::= { coord 1 }

2.5.2 Coord Correction Mode Parameters

coordCorrectionMode OBJECT-TYPE

SYNTAX INTEGER { other (1),
dwell (2),
shortway (3),
addOnly (4) }

ACCESS read-write

STATUS mandatory

DESCRIPTION

"This object defines the Coord Correction Mode. The possible modes are:

other; the coordinator establishes a new offset by a mechanism not defined in this standard.

dwel; when changing offset, the coordinator shall establish a new offset by dwelling in the coord phase(s) until the desired offset is reached.
shortway (Smooth); when changing offset, the coordinator shall establish a new offset by adding or subtracting to/from the timings in a manner that limits the cycle change. This operation is performed in a device specific manner.
addOnly; when changing offset, the coordinator shall establish a new offset by adding to the timings in a manner that limits the cycle change. This operation is performed in a device specific manner."

::= { coord 2 }

2.5.3 Coord Maximum Mode Parameters

coordMaximumMode OBJECT-TYPE

SYNTAX INTEGER { other (1),
maximum1 (2),
maximum2 (3),
maxInhibit (4) }

ACCESS read-write

STATUS mandatory

DESCRIPTION

"This object defines the Coord Maximum Mode. The possible modes are:
other; the maximum mode is determined by some other mechanism not defined in this standard.
maximum1; the internal Maximum 1 Timing shall be effective while coordination is running a pattern.
maximum2; the internal Maximum 2 Timing shall be effective while coordination is running a pattern.
maxInhibit; the internal Maximum Timing shall be inhibited while coordination is running a pattern."

::= { coord 3 }

2.5.4 Coord Force Mode Parameters

coordForceMode OBJECT-TYPE

SYNTAX INTEGER { other(1),
floating (2),
fixed (3) }

ACCESS read-write

STATUS mandatory

DESCRIPTION

"This object defines the Pattern Force Mode. The possible modes are:
other; the CU implements a mechanism not defined in this standard.
floating; each phase will be forced the split time after it becomes active. This allows unused split time to revert to the coord phase.
fixed; each phase will be forced at a fixed position in the cycle. This allows unused split time to revert to the following phase."

::= { coord 4 }

2.5.5 Maximum Patterns Parameters

maxPatterns OBJECT-TYPE

SYNTAX INTEGER (0..253)

ACCESS read-only

STATUS mandatory

DESCRIPTION

"The maximum number of Patterns this Actuated Controller Unit supports. This object indicates how many rows are in the patternTable object. Pattern 254 is always Flash mode and pattern 255 is always Free mode."

::= { coord 5 }

2.5.6 Pattern Table Type

patternTableType OBJECT-TYPE

SYNTAX INTEGER { other (1),
patterns (2),
offset3 (3),
offset5 (4) }

ACCESS read-only

STATUS mandatory

DESCRIPTION

"This object provides information about any special organizational structure required for the pattern table. The defined structures are as follows:

- other - The pattern table setup is not described in this standard, refer to device manual.
- patterns - Each row of the pattern table represents a unique pattern and has no dependencies on other rows.
- offset3 - The pattern table is organized into plans which have three offsets. Each plan uses three consecutive rows. Only the offsetTime value may vary between each of the three rows. Plan 1 is contained in rows 1, 2 and 3, Plan 2 is contained in rows 4, 5 and 6, Plan 3 is in rows 7, 8 and 9, etc..
- offset5 - The pattern table is organized into plans which have five offsets. Each plan occupies five consecutive rows. Only the offsetTime value may vary between each of the rows. Plan 1 is contained in rows 1, 2, 3, 4 and 5, Plan 2 is contained in rows 6, 7, 8, 9 and 10, Plan 3 is contained in rows 11, 12, 13, 14 and 15, etc..."

REFERENCE

"NEMA TS 2 Clause 3.6.2.1 and 3.6.2.2"

::= { coord 6 }

2.5.7 Pattern Table

patternTable OBJECT-TYPE

SYNTAX SEQUENCE OF PatternEntry

ACCESS not-accessible

STATUS mandatory

DESCRIPTION

"A table containing Actuated Controller Unit coordination Pattern parameters. The number of rows in this table is equal to the maxPatterns object."

::= { coord 7 }

patternEntry OBJECT-TYPE
SYNTAX PatternEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION
"Parameters for a specific Actuated Controller Unit pattern."
INDEX { patternNumber }
 ::= { patternTable 1 }

PatternEntry ::= SEQUENCE {
 patternNumber INTEGER,
 patternCycleTime INTEGER,
 patternOffsetTime INTEGER,
 patternSplitNumber INTEGER,
 patternSequenceNumber INTEGER }

2.5.7.1 Pattern Number Entry

patternNumber OBJECT-TYPE
SYNTAX INTEGER (0..253)
ACCESS read-only
STATUS mandatory
DESCRIPTION
"The pattern number for objects in this row. This value shall not exceed the maxPatterns object value."
 ::= { patternEntry 1 }

2.5.7.2 Pattern Cycle Time

patternCycleTime OBJECT-TYPE
SYNTAX INTEGER (0..255)
ACCESS read-write
STATUS mandatory
DESCRIPTION
"The patternCycleTime object specifies the length of the pattern cycle in seconds (NEMA TS 2 range: 30-255). A pattern cycle time less than adequate to service the minimum requirements of all phases shall result in Free mode. If the pattern cycle time is zero and the associated split table (if any) contains values greater than zero then the CU shall utilize the split time values as maximum values for each phase."
REFERENCE
"NEMA TS 2 Clause 3.6.2.1.1"
 ::= { patternEntry 2 }

2.5.7.3 Pattern Offset Time Parameter

patternOffsetTime OBJECT-TYPE
SYNTAX INTEGER (0..255)
ACCESS read-write
STATUS mandatory
DESCRIPTION
"The patternOffsetTime defines by how many seconds (NEMA TS 2 range: 0-254) the local time zero shall lag the system time zero (synchronization pulse) for this pattern.. An offset value equal to or greater than the cycle time shall result in Free being the operational mode. While this condition exists, the Local Free bit of unitAlarmStatus and the LocalOverride bit of shortAlarmStatus shall be set to one (1)."
REFERENCE
"NEMA TS 2 Clause 3.6.2.2"

::= { patternEntry 3 }

2.5.7.4 Pattern Split Number Parameter

patternSplitNumber OBJECT-TYPE
SYNTAX INTEGER (0..255)
ACCESS read-write
STATUS mandatory
DESCRIPTION

"This object is used to locate information in the splitTable to use for this pattern. This value shall not exceed the maxSplits object value."

::= { patternEntry 4 }

2.5.7.5 Pattern Sequence Number Parameter

patternSequenceNumber OBJECT-TYPE
SYNTAX INTEGER (0..255)
ACCESS read-write
STATUS mandatory
DESCRIPTION

"This object is used to locate information in the sequenceTable to use with this pattern. This value shall not exceed the maxSequences object value."

::= { patternEntry 5 }

2.5.8 Maximum Splits

maxSplits OBJECT-TYPE
SYNTAX INTEGER (0..255)
ACCESS read-only
STATUS mandatory
DESCRIPTION

"The maximum number of Split Plans this Actuated Controller Unit supports. This object indicates how many Split plans are in the splitTable object."

::= { coord 8 }

2.5.9 Split Table

splitTable OBJECT-TYPE
SYNTAX SEQUENCE OF SplitEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION

"A table containing Actuated Controller Unit coordination split parameters. The number of rows in this table is equal to maxSplits."

::= { coord 9 }

splitEntry OBJECT-TYPE
SYNTAX SplitEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION

"Split type Parameters for a specific Actuated Controller Unit phase."

INDEX { splitNumber, splitPhase }

::= { splitTable 1 }

```
SplitEntry ::= SEQUENCE {  
    splitNumber      INTEGER,  
    splitPhase       INTEGER,  
    splitTime        INTEGER,  
    splitMode        INTEGER,  
    splitCoordPhase  INTEGER }
```

2.5.9.1 Split Number

```
splitNumber OBJECT-TYPE  
    SYNTAX  INTEGER (1..255)  
    ACCESS  read-only  
    STATUS  mandatory  
    DESCRIPTION  
        "The object defines which rows of the split table comprise a split group. All rows that have  
        the same splitNumber are in the same split group. The value of this object shall not exceed  
        the maxSplits object value."  
::= { splitEntry 1 }
```

2.5.9.2 Split Phase Number

```
splitPhase OBJECT-TYPE  
    SYNTAX  INTEGER (1..255)  
    ACCESS  read-only  
    STATUS  mandatory  
    DESCRIPTION  
        "The phase number for objects in this row. The value of this object shall not exceed the  
        maxPhases object value."  
::= { splitEntry 2 }
```

2.5.9.3 Split Time Parameter

```
splitTime OBJECT-TYPE  
    SYNTAX  INTEGER (0..255)  
    ACCESS  read-write  
    STATUS  mandatory  
    DESCRIPTION  
        "The time in seconds the splitPhase is allowed to receive (i.e. before a Force Off is applied)  
        when constant demands exist on all phases. In floating coordForceMode, this is always the  
        maximum time a non-coordinated phase is allowed to receive. In fixed coordForceMode, the  
        actual allowed time may be longer if a previous phase gapped out.  
        The splitTime includes all phase clearance times for the associated phase. The split time  
        shall be longer than the sum of the phase minimum service requirements (Minimum Green,  
        Yellow Change, Red Clearance, Walk, Pedestrian Clearance, etc.) for the phase. When the  
        time is NOT adequate to service the minimum service requirements of the phase, Free  
        Mode shall be the result.  
        If the cycleTime entry of the associated patternTable entry is zero (i.e. the device is in Free  
        Mode), then the value of this object shall be applied as a maximum time for the associated  
        phase.  
        If the critical path through the phase diagram is less than the cycleTime entry of the  
        associated patternTable entry, all extra time is allotted to the coordination phase in each  
        ring.  
        If the critical path through the phase diagram is greater than the cycleTime entry of the  
        associated patternTable entry (and the cycleTime is not zero) the device shall operate in the  
        Free Mode.
```

While the Free Mode condition exists, the Local Override bit of shortAlarm shall be set to one (1)."

REFERENCE

"NEMA TS 2 Clause 3.6.2.1.2"

::= { splitEntry 3 }

2.5.9.4 Split Mode Parameter

splitMode OBJECT-TYPE

SYNTAX INTEGER { other(1),
none (2),
minimumVehicleRecall (3),
maximumVehicleRecall (4),
pedestrianRecall (5),
maximumVehicleAndPedestrianRecall (6),
phaseOmitted (7) }

ACCESS read-write

STATUS mandatory

DESCRIPTION

"This object defines operational characteristics of the phase. The following options are available:

other; the operation is not specified in this standard

none; no split mode control.

minimumVehicleRecall; this phase operates with a minimum vehicle recall.

maximumVehicleRecall; this phase operates with a maximum vehicle recall.

pedestrianRecall; this phase operates with a pedestrian recall.

maximumVehicleAndPedestrianRecall; this phase operates with a maximum vehicle & pedestrian recall.

phaseOmitted; this phase is omitted."

::= { splitEntry 4 }

2.5.9.5 Split Coordinated Phase

splitCoordPhase OBJECT-TYPE

SYNTAX INTEGER (0..1)

ACCESS read-write

STATUS mandatory

DESCRIPTION

"To select the associated phase as a coordinated phase this object shall be set to TRUE (non zero)."

::= { splitEntry 5 }

2.5.10 Coordination Pattern Status

coordPatternStatus OBJECT-TYPE

SYNTAX INTEGER (0..255)

ACCESS read-only

STATUS mandatory

DESCRIPTION

"This object defines the running coordination pattern / mode in the device. The possible values are:

Value Description

0 Not used

1-253 Pattern, indicates the currently running pattern

254 Free, indicates Free operation without coordination.

255 Flash, indicates Automatic Flash without coordination."

::= { coord 10 }

2.5.11 Local Free Status

localFreeStatus OBJECT-TYPE

SYNTAX INTEGER { other(1),
notFree(2),
commandFree(3),
transitionFree(4),
inputFree(5),
coordFree(6),
badPlan(7),
badCycleTime(8),
splitOverrun (9),
invalidOffset (10),
failed(11) }

ACCESS read-only

STATUS mandatory

DESCRIPTION

"The Free modes:

notFree - The unit is not running in free mode.

commandFree - the current pattern command is the Free mode pattern.

transitionFree - the CU has a pattern command but is cycling to a point to begin coordination.

inputFree - one of the CU inputs cause it to not respond to coordination.

coordFree - the CU programming for the called pattern is to run Free.

badPlan - Free - the called pattern is invalid.

badCycleTimeFree - the specified cycle time is not adequate to service the all phase minimum service requirements.

splitOverrun - Free - the sum of the critical path splitTime's exceed the programmed patternCycleTime value..

invalidOffset - Free - the programmed patternOffsetTime value exceeds the programmed patternCycleTime value.

failedFree - cycling diagnostics have called for Free.

other - Some other condition has caused the device to run in free mode."

::= { coord 11 }

2.5.12 Coordination Cycle Status

coordCycleStatus OBJECT-TYPE

SYNTAX INTEGER (0..65535)

ACCESS read-only

STATUS mandatory

DESCRIPTION

"The Local Cycle Timer position for the running pattern in the device. Value 0 to 510. Count down from current cycle's Cycle Length to ZERO. This value may exceed the cycleTime value during correction cycle's (cycleTime + dwell)."

::= { coord 12 }

2.5.13 Coordination Sync Status

coordSyncStatus OBJECT-TYPE

SYNTAX INTEGER (0..65536)

ACCESS read-only

STATUS mandatory

DESCRIPTION

"The Sync Cycle Timer position for the running pattern in the device. Value 0 to 510. It represents the time that has elapsed since system time zero."

::= { coord 13 }

2.5.14 System Pattern Control

systemPatternControl OBJECT-TYPE

SYNTAX INTEGER (0..255)

ACCESS read-write

STATUS mandatory

DESCRIPTION

"This object is used to establish the Called System Pattern / Mode for the device. The possible values are:

Value	Description
0	Standby - the system relinquishes control of the device.
1-253	Pattern - these values indicate the system commanded pattern
254	Free - this value indicates a call for Free
255	Flash - this value indicates a call for Automatic Flash

If an unsupported / invalid pattern is called, Free shall be the operational mode. The value of this object is ignored in BACKUP mode.

A write to this object shall reset the BACKUP timer to ZERO."

::= { coord 14 }

2.5.15 System Sync Control

systemSyncControl OBJECT-TYPE

SYNTAX INTEGER (0..255)

ACCESS read-write

STATUS mandatory

DESCRIPTION

"This object is used to set the current System Sync Cycle Timer in the Unit to a Value 0 to 255. The device shall recognize a write to this object as a command to set the timer equal to the value in the object. The Sync Cycle Timer counts up from 0. When the value in the object is 255, the System Sync Cycle Timer shall be referenced to the local Time Base in accordance with its programming.

This CU must maintain an accuracy of 0.1 seconds based on the receipt of the SET packet.

The value of this object is ignored when in BACKUP Mode.

A write to this object shall reset the BACKUP timer to ZERO."

::= { coord 15 }

2.6 TIME BASE PARAMETERS

timebase OBJECT IDENTIFIER

::= { asc 5 }

-- "This object is an identifier used to group all objects for support of timebase functions. If a device -- implements timebase functions then these objects shall be supported."

2.6.1 Time Base Pattern Sync Parameter

timebasePatternSync OBJECT-TYPE

SYNTAX INTEGER (0..65535)

ACCESS read-write

STATUS mandatory

DESCRIPTION

"Pattern Sync Reference in minutes past midnight. When the value is 0xFFFF, the controller unit shall use the Action time as the Sync Reference for that pattern."

REFERENCE

"NEMA TS 2 Clause 3.8.2"

::= { timebase 1 }

2.6.2 Maximum Time Base Actions

maxTimebaseAscActions OBJECT-TYPE

SYNTAX INTEGER (0..255)

ACCESS read-only

STATUS mandatory

DESCRIPTION

"The Maximum Number of Actions this device supports. This object indicates the maximum rows which shall appear in the timebaseAscActionTable object."

::= { timebase 2 }

2.6.3 Time Base ASC Action Table

timebaseAscActionTable OBJECT-TYPE

SYNTAX SEQUENCE OF TimebaseAscActionEntry

ACCESS not-accessible

STATUS mandatory

DESCRIPTION

"A table containing Actuated Controller Unit Time Base action parameters. The number of rows in this table is equal to the maxTimebaseAscActions object."

::= { timebase 3 }

timebaseAscActionEntry OBJECT-TYPE

SYNTAX TimebaseAscActionEntry

ACCESS not-accessible

STATUS mandatory

DESCRIPTION

"Action Parameters for a Actuated Controller Unit Time Base Program."

INDEX { timebaseAscActionNumber }

::= { timebaseAscActionTable 1 }

TimebaseAscActionEntry ::= SEQUENCE {

timebaseAscActionNumber INTEGER,

timebaseActionPattern INTEGER,

timebaseActionAuxillaryFunction INTEGER,

timebaseActionSpecialFunction INTEGER }

2.6.3.1 Time Base Action Number Entry

timebaseAscActionNumber OBJECT-TYPE

SYNTAX INTEGER (0..255)

ACCESS read-only

STATUS mandatory

DESCRIPTION

"The time base Action number for objects in this row. This value shall not exceed the maxTimebaseAscActions object value."

::= { timebaseAscActionEntry 1 }

2.6.3.2 Time Base Action Pattern Parameter

timebaseActionPattern OBJECT-TYPE
SYNTAX INTEGER (0..255)
ACCESS read-write
STATUS mandatory
DESCRIPTION

"The Pattern that shall be active when this Action is active. The value shall not exceed the value of maxPatterns. A pattern of zero indicates that no pattern is being selected."

::= { timebaseAscActionEntry 2 }

2.6.3.3 Time Base Action Auxiliary Function Parameter

timebaseActionAuxillaryFunction OBJECT-TYPE
SYNTAX INTEGER (0..255)
ACCESS read-write
STATUS mandatory
DESCRIPTION

"The Auxiliary functions that shall be active when this Action is active.

Bit Function
0 Auxiliary function 1 enabled if set (non-zero), disabled if clear (zero).
1 Auxiliary function 2 enabled if set (non-zero), disabled if clear (zero).
2 Auxiliary function 3 enabled if set (non-zero), disabled if clear (zero).
3-7 Reserved"

::= { timebaseAscActionEntry 3 }

2.6.3.4 Time Base Action Special Function Parameter

timebaseActionSpecialFunction OBJECT-TYPE
SYNTAX INTEGER (0..255)
ACCESS read-write
STATUS mandatory
DESCRIPTION

"The Special Functions that shall be active when this Action is active.

Bit 7 - Special Function 8 (0 =False / Disabled, 1 = True / Enabled)
Bit 6 - Special Function 7 (0 =False / Disabled, 1 = True / Enabled)
Bit 5 - Special Function 6 (0 =False / Disabled, 1 = True / Enabled)
Bit 4 - Special Function 5 (0 =False / Disabled, 1 = True / Enabled)
Bit 3 - Special Function 4 (0 =False / Disabled, 1 = True / Enabled)
Bit 2 - Special Function 3 (0 =False / Disabled, 1 = True / Enabled)
Bit 1 - Special Function 2 (0 =False / Disabled, 1 = True / Enabled)
Bit 0 - Special Function 1 (0 =False / Disabled, 1 = True / Enabled)"

::= { timebaseAscActionEntry 4 }

2.6.4 Time Base ASC Action Status

timebaseAscActionStatus OBJECT-TYPE
SYNTAX INTEGER(0..255)
ACCESS read-only
STATUS mandatory
DESCRIPTION

"This object indicates the current time base Action Table row that will be used when the CU is in Time Base operation. A value of zero indicates that no time base Action is selected."

::= { timebase 4 }

2.7 PREEMPT PARAMETERS

preempt OBJECT IDENTIFIER

::= { asc 6 }

-- The preempt node contains objects that support preempt input functions for the device.

2.7.1 Maximum Preempts

maxPreempts OBJECT-TYPE

SYNTAX INTEGER (0..255)

ACCESS read-only

STATUS mandatory

DESCRIPTION

"The Maximum Number of Preempts this Actuated Controller Unit supports. This object indicates the maximum rows which shall appear in the preemptTable object."

REFERENCE

"NEMA TS 2 Clause 3.7"

::= { preempt 1 }

2.7.2 Preempt Table

preemptTable OBJECT-TYPE

SYNTAX SEQUENCE OF PreemptEntry

ACCESS not-accessible

STATUS mandatory

DESCRIPTION

"A table containing Actuated Controller Unit preemption parameters. The number of rows in this table is equal to the maxPreempts object."

::={ preempt 2 }

preemptEntry OBJECT-TYPE

SYNTAX PreemptEntry

ACCESS not-accessible

STATUS mandatory

DESCRIPTION

"Parameters for a specific Actuated Controller Unit preemptor."

INDEX { preemptNumber }

::={ preemptTable 1 }

PreemptEntry ::= SEQUENCE {

preemptNumber	INTEGER,
preemptControl	INTEGER,
preemptLink	INTEGER,
preemptDelay	INTEGER,
preemptMinimumDuration	INTEGER,
preemptMinimumGreen	INTEGER,
preemptMinimumWalk	INTEGER,
preemptEnterPedClear	INTEGER,
preemptTrackGreen	INTEGER,
preemptDwellGreen	INTEGER,
preemptMaximumPresence	INTEGER,
preemptTrackPhase	OCTET STRING,
preemptDwellPhase	OCTET STRING,
preemptDwellPed	OCTET STRING,
preemptExitPhase	OCTET STRING,
preemptState	INTEGER }

2.7.2.1 Preempt Number

preemptNumber OBJECT-TYPE
SYNTAX INTEGER (0..255)
ACCESS read-only
STATUS mandatory
DESCRIPTION
"The preempt number for objects in this row. The value shall not exceed the maxPreempts object value."
 ::= { preemptEntry 1 }

2.7.2.2 Preempt Control Parameter

preemptControl OBJECT-TYPE
SYNTAX INTEGER (0..255)
ACCESS read-write
STATUS mandatory
DESCRIPTION
"Preempt Miscellaneous Control Parameter Mask (0 = False / Disabled, 1 = True / Enabled) as follows:
Bit Function
7-4
3 Flash Dwell - the CU shall cause the phases listed in the preemptDwellPhase object to flash Yellow during the Dwell phase. All active phases not listed in the preemptDwellPhase shall flash Red. If any conflicting phase numbers are listed in the preemptDwellPhase then all active phases shall flash Red. This control is optional.
2 Preempt Override preemptNumber + 1 - provide a means to define whether this preempt shall NOT override the next higher numbered Preempt. When set (1) this preempt shall NOT override the next higher numbered preempt. This parameter shall be ignored when preemptNumber equals maxPreempts.
1 Preempt Override Flash - provide a means to define whether this preempt shall NOT override Automatic Flash. When set (1) this preempt shall not override Automatic Flash.
0 Non-Locking Memory - provide a means to enable an operation which does not require detector memory. When set (1) a preempt sequence shall not occur if the preempt input terminates prior to expiration of the preemptDelay time."
REFERENCE
"NEMA TS 2 Clause 3.7.2.1 and 3.7.2.2"
 ::= { preemptEntry 2 }

2.7.2.3 Preempt Link Parameter

preemptLink OBJECT-TYPE
SYNTAX INTEGER (0..255)
ACCESS read-write
STATUS mandatory
DESCRIPTION
"This object provides a means to define a higher priority preempt to be combined (linked) with this preempt. At the end of Dwell time, the linked preempt shall receive an automatic call which shall be maintained as long as the demand for this preempt is active. Any value that is not a higher priority preempt or a valid preempt shall be ignored. The value shall not exceed the maxPreempts object value."
 ::= { preemptEntry 3 }

2.7.2.4 Preempt Delay Parameter

preemptDelay OBJECT-TYPE
SYNTAX INTEGER (0..65535)
ACCESS read-write
STATUS mandatory
DESCRIPTION
"Preempt Delay Time in seconds (0-600 sec). This value determines the time the preempt input shall be active prior to initiating any preempt sequence. A non-locking preempt input which is removed prior to the completion of this time shall not cause a preempt sequence to occur."
 ::= { preemptEntry 4 }

2.7.2.5 Preempt Duration Parameter

preemptMinimumDuration OBJECT-TYPE
SYNTAX INTEGER (0..65535)
ACCESS read-write
STATUS mandatory
DESCRIPTION
"Preempt Minimum Duration Time in seconds (0..65535). This value determines the minimum time during which the preempt is active. Duration begins timing at the end of Preempt Delay (if non zero) and will prevent an exit from the Dwell state until this time has elapsed."
 ::= { preemptEntry 5 }

2.7.2.6 Preempt Minimum Green Parameter

preemptMinimumGreen OBJECT-TYPE
SYNTAX INTEGER (0..255)
ACCESS read-write
STATUS optional
DESCRIPTION
"Preempt Minimum Green Time in seconds (0-255 sec). A preempt initiated transition shall not cause the termination of an existing Green prior to its display for lesser of the phase's Minimum Green time or this period."
 ::= { preemptEntry 6 }

2.7.2.7 Preempt Minimum Walk Parameter

preemptMinimumWalk OBJECT-TYPE
SYNTAX INTEGER (0..255)
ACCESS read-write
STATUS optional
DESCRIPTION
"Preempt Minimum Walk Time in seconds (0-255 sec). A preempt initiated transition shall not cause the termination of an existing Walk prior to its display for the lesser of the phase's Minimum Walk time or this period."
 ::= { preemptEntry 7 }

2.7.2.8 Preempt Enter Pedestrian Clear Parameter

preemptEnterPedClear OBJECT-TYPE

SYNTAX INTEGER (0..255)

ACCESS read-write

STATUS optional

DESCRIPTION

"Enter Ped ClearTime in seconds (0-255 sec). This parameter controls the ped clear timing for a normal Walk signal terminated by a preempt initiated transition. A preempt initiated transition shall not cause the termination of a Pedestrian Clearance prior to its display for the lesser of the phase's Pedestrian Clearance time or this period."

::= { preemptEntry 8 }

2.7.2.9 Preempt Track Green Parameter

preemptTrackGreen OBJECT-TYPE

SYNTAX INTEGER (0..255)

ACCESS read-write

STATUS mandatory

DESCRIPTION

"Track Clear Green Time in seconds (0-255 sec). This parameter controls the green timing for the track clearance movement. The phase(s) active during the Track Green interval are enabled in preemptTrackPhase object."

::= { preemptEntry 9 }

2.7.2.10 Preempt Minimum Dwell Parameter

preemptDwellGreen OBJECT-TYPE

SYNTAX INTEGER (0..255)

ACCESS read-write

STATUS mandatory

DESCRIPTION

"Minimum Dwell Time in seconds (1-255 sec). This parameter controls the minimum timing for the dwell movement. The phase(s) active during the Dwell interval are enabled in preemptDwellPhase object.

The Dwell interval shall not terminate prior to the completion of Preempt Duration Time, Preempt Dwell Time, & the call is no longer present."

::= { preemptEntry 10 }

2.7.2.11 Preempt Maximum Presence Parameter

preemptMaximumPresence OBJECT-TYPE

SYNTAX INTEGER (0..65535)

ACCESS read-write

STATUS mandatory

DESCRIPTION

"Preempt Maximum Presence time in seconds (0-65535 sec). This value determines the maximum time which a preempt call may remain active and be considered valid. When the preempt call has been active for this time period, the CU shall return to normal operation.

This preempt call shall be considered invalid until such time as a change in state occurs (no longer active). When set to zero the preempt maximum presence time is disabled."

::= { preemptEntry 11 }

2.7.2.12 Preempt Track Phase Parameter

preemptTrackPhase OBJECT-TYPE

SYNTAX OCTET STRING

ACCESS read-write

STATUS mandatory

DESCRIPTION

"Each octet within the octet string contains a phaseNumber(binary value) that shall be active during the Preempt Track Green interval."

::= { preemptEntry 12 }

2.7.2.13 Preempt Dwell Phase Parameters

preemptDwellPhase OBJECT-TYPE

SYNTAX OCTET STRING

ACCESS read-write

STATUS mandatory

DESCRIPTION

"Each octet within the octet string contains a phaseNumber (binary value)that is allowed during the Preempt Dwell interval."

::= { preemptEntry 13 }

2.7.2.14 Preempt Dwell Ped Parameters

preemptDwellPed OBJECT-TYPE

SYNTAX OCTET STRING

ACCESS read-write

STATUS optional

DESCRIPTION

"Each octet within the octet string contains a phaseNumber (binary value)indicating a pedestrian movement that is allowed during the Preempt Dwell interval."

::= { preemptEntry 14 }

2.7.2.15 Preempt Exit Phase Parameters

preemptExitPhase OBJECT-TYPE

SYNTAX OCTET STRING

ACCESS read-write

STATUS mandatory

DESCRIPTION

"Each octet within the octet string contains a phaseNumber (binary value)that shall be active following Preempt."

::= { preemptEntry 15 }

2.7.2.16 Preempt State

preemptState OBJECT-TYPE

SYNTAX INTEGER {
 other (1),
 notActive (2),
 notActiveWithCall (3),
 entryStarted (4),
 trackService (5),
 dwell (6),
 linkActive (7),
 exitStarted (8),
 maxPresence (9) }

ACCESS read-only

STATUS optional

DESCRIPTION

"Preempt State provides status on which state the associated preempt is in. The states are as follows:

State	Description
notActive	preempt input is not active, this preempt is not active.
notActiveWithCall	preempt input is active, preempt service has not started.
entryStarted	preempt service is timing the entry intervals.
trackService	preempt service is timing the track intervals.
dwell	preempt service is timing the dwell intervals.
linkActive	preempt service is performing linked operation.
exitStarted	preempt service is timing the exit intervals.
maxPresence	preempt input has exceeded maxPresence time
other	preempt service is not specified in this standard."

::= { preemptEntry 16 }

2.7.3 Preempt Control Table

preemptControlTable OBJECT-TYPE

SYNTAX SEQUENCE OF PreemptControlEntry

ACCESS not-accessible

STATUS optional

DESCRIPTION

"This table contains the control objects that allow the preempts to be activated remotely. There shall be one control object for each preempt input supported by the device. The number of rows in this table shall be equal to maxPreempts."

::= { preempt 3 }

preemptControlEntry OBJECT-TYPE

SYNTAX PreemptControlEntry

ACCESS not-accessible

STATUS mandatory

DESCRIPTION

"Control objects for each preempt input. These objects allow the system to activate preempt functions remotely."

INDEX { preemptControlNumber }

::= { preemptControlTable 1 }

PreemptControlEntry ::= SEQUENCE {

preemptControlNumber

preemptControlState

INTEGER,

INTEGER }

2.7.3.1 Preempt Control Number

preemptControlNumber OBJECT-TYPE
SYNTAX INTEGER (0..255)
ACCESS read-only
STATUS mandatory
DESCRIPTION

"This object shall indicate the preempt input number controlled by the associated preemptControlState object in this row."

::= { preemptControlEntry 1 }

2.7.3.2 Preempt Control State

preemptControlState OBJECT-TYPE
SYNTAX INTEGER (0..1)
ACCESS read-only
STATUS mandatory
DESCRIPTION

"This object when set to ON (non-zero) shall cause the associated preempt actions to occur unless the actions have already been started by the physical preempt input. The preempt shall remain active as long as this object is ON or the physical preempt input is ON. This object when set to OFF (zero) shall cause the physical preempt input to control the associated preempt actions.

The value of this object is ignored in BACKUP mode. A write to this object shall reset the BACKUP timer."

::= { preemptControlEntry 2 }

2.8 RING PARAMETERS

ring OBJECT IDENTIFIER
::= { asc 7 }

--The ring node contains objects that support ring configuration, status and control functions in the device.

2.8.1 Maximum Rings

maxRings OBJECT-TYPE
SYNTAX INTEGER (0..255)
ACCESS read-only
STATUS mandatory
DESCRIPTION

"The value of this object shall specify the maximum number of rings this device supports."

::= { ring 1 }

2.8.2 Maximum Sequences

maxSequences OBJECT-TYPE
SYNTAX INTEGER (0..255)
ACCESS read-only
STATUS mandatory
DESCRIPTION

"The value of this object shall specify the maximum number of sequence plans this device supports."

::= { ring 2 }

2.8.3 Sequence Table

sequenceTable OBJECT-TYPE
SYNTAX SEQUENCE OF SequenceEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION
"This table contains all the sequence plans for the controller. A sequence plan shall consist of one row for each ring that the CU supports. Each row defines the phase service order for that ring."
 ::= { ring 3 }

sequenceEntry OBJECT-TYPE
SYNTAX SequenceEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION
"Phase Sequence Parameters for an Actuated Controller Unit."
INDEX { sequenceNumber, sequenceRingNumber }
 ::= { sequenceTable 1 }

SequenceEntry ::= SEQUENCE {
sequenceNumber INTEGER,
sequenceRingNumber INTEGER,
sequenceData OCTET STRING }

2.8.3.1 Sequence Number

sequenceNumber OBJECT-TYPE
SYNTAX INTEGER (0..255)
ACCESS read-only
STATUS mandatory
DESCRIPTION
"This number identifies a sequence plan. Each row of the table contains the phase sequence for a ring. A sequence plan shall consist of one row for each ring that defines the phase sequences for that ring."
 ::= { sequenceEntry 1 }

2.8.3.2 Sequence Ring Number

sequenceRingNumber OBJECT-TYPE
SYNTAX INTEGER (0..255)
ACCESS read-write
STATUS mandatory
DESCRIPTION
"This number identifies the ring number this phase sequence applies to."
 ::= { sequenceEntry 2 }

2.8.3.3 SequenceData

sequenceData OBJECT-TYPE

SYNTAX OCTET STRING

ACCESS read-write

STATUS mandatory

DESCRIPTION

"Each octet is a Phase Number (binary value) within the associated ring number. The phase number value shall not exceed the maxPhases object value. The order of phase numbers determines the phase sequence for the ring. The phase numbers shall not be ordered in a manner that would violate the devices compatibility group."

::= { sequenceEntry 3 }

2.8.4 Maximum Ring Control Groups

maxRingControlGroups OBJECT-TYPE

SYNTAX INTEGER (0..255)

ACCESS read-only

STATUS mandatory

DESCRIPTION

"The maximum number of Ring Control Groups (8 rings per group) this Actuated Controller Unit supports. This value is equal to TRUNCATE[(maxRings + 7) / 8]. This object indicates the maximum rows which shall appear in the ringControlGroupTable object."

::= { ring 4 }

2.8.5 Ring Control Group Table

ringControlGroupTable OBJECT-TYPE

SYNTAX SEQUENCE OF RingControlGroupEntry

ACCESS not-accessible

STATUS mandatory

DESCRIPTION

"A table containing Actuated Controller Unit Ring Control in groups of eight rings. The number of rows in this table is equal to the maxRingControlGroups object."

::= { ring 5 }

ringControlGroupEntry OBJECT-TYPE

SYNTAX RingControlGroupEntry

ACCESS not-accessible

STATUS mandatory

DESCRIPTION

"Ring Control for eight Actuated Controller Unit rings."

INDEX { ringControlGroupNumber }

::= { ringControlGroupTable 1 }

RingControlGroupEntry ::= SEQUENCE {

ringControlGroupNumber	INTEGER,
ringControlGroupStopTime	INTEGER,
ringControlGroupForceOff	INTEGER,
ringControlGroupMax2	INTEGER,
ringControlGroupMaxInhibit	INTEGER,
ringControlGroupPedRecycle	INTEGER,
ringControlGroupRedRest	INTEGER,
ringControlGroupOmitRedClear	INTEGER }

2.8.5.1 Ring Control Group Number

ringControlGroupNumber OBJECT-TYPE

SYNTAX INTEGER (0..255)

ACCESS read-only

STATUS mandatory

DESCRIPTION

"The Ring Control Group number for objects in this row. This value shall not exceed the maxRingControlGroups object value."

::= { ringControlGroupEntry 1 }

2.8.5.2 Ring Stop Time Control

ringControlGroupStopTime OBJECT-TYPE

SYNTAX INTEGER (0..255)

ACCESS read-write

STATUS mandatory

DESCRIPTION

"This object is used to allow a remote entity to stop timing in the device. When a bit = 1, the device shall activate the System Stop Time control for that ring.

Bit 7 = Ring number = (ringControlGroupNumber * 8)

Bit 6 = Ring number = (ringControlGroupNumber * 8) - 1

Bit 5 = Ring number = (ringControlGroupNumber * 8) - 2

Bit 4 = Ring number = (ringControlGroupNumber * 8) - 3

Bit 3 = Ring number = (ringControlGroupNumber * 8) - 4

Bit 2 = Ring number = (ringControlGroupNumber * 8) - 5

Bit 1 = Ring number = (ringControlGroupNumber * 8) - 6

Bit 0 = Ring number = (ringControlGroupNumber * 8) - 7

The value of this object is ignored in BACKUP Mode. A write to this object shall reset the Backup timer."

REFERENCE

"NEMA TS 2 Clause 3.5.4.1.6"

::= { ringControlGroupEntry 2 }

2.8.5.3 Ring Force Off Control

ringControlGroupForceOff OBJECT-TYPE

SYNTAX INTEGER (0..255)

ACCESS read-write

STATUS mandatory

DESCRIPTION

"This object is used to allow a remote entity to terminate phases via a force off command in the device. When a bit = 1, the device shall activate the System Force Off control for that ring.

Bit 7 = Ring number = (ringControlGroupNumber * 8)

Bit 6 = Ring number = (ringControlGroupNumber * 8) - 1

Bit 5 = Ring number = (ringControlGroupNumber * 8) - 2

Bit 4 = Ring number = (ringControlGroupNumber * 8) - 3

Bit 3 = Ring number = (ringControlGroupNumber * 8) - 4

Bit 2 = Ring number = (ringControlGroupNumber * 8) - 5

Bit 1 = Ring number = (ringControlGroupNumber * 8) - 6

Bit 0 = Ring number = (ringControlGroupNumber * 8) - 7

The value of this object is ignored in BACKUP Mode. A write to this object shall reset the Backup timer."

REFERENCE

"NEMA TS 2 Clause 3.5.4.1.1"

::= { ringControlGroupEntry 3 }

2.8.5.4 Ring Max 2 Control

ringControlGroupMax2 OBJECT-TYPE

SYNTAX INTEGER (0..255)

ACCESS read-write

STATUS optional

DESCRIPTION

"This object is used to allow a remote entity to request maximum 2 timings in the device.

When a bit = 1, the device shall activate the System Maximum 2 control for that ring.

Bit 7 = Ring number = (ringControlGroupNumber * 8)

Bit 6 = Ring number = (ringControlGroupNumber * 8) - 1

Bit 5 = Ring number = (ringControlGroupNumber * 8) - 2

Bit 4 = Ring number = (ringControlGroupNumber * 8) - 3

Bit 3 = Ring number = (ringControlGroupNumber * 8) - 4

Bit 2 = Ring number = (ringControlGroupNumber * 8) - 5

Bit 1 = Ring number = (ringControlGroupNumber * 8) - 6

Bit 0 = Ring number = (ringControlGroupNumber * 8) - 7

The value of this object is ignored in BACKUP Mode. A write to this object shall reset the Backup timer."

REFERENCE

"NEMA TS 2 Clause 3.5.4.1.7"

::= { ringControlGroupEntry 4 }

2.8.5.5 Ring Max Inhibit Control

ringControlGroupMaxInhibit OBJECT-TYPE

SYNTAX INTEGER (0..255)

ACCESS read-write

STATUS optional

DESCRIPTION

"This object is used to allow a remote entity to request internal maximum timings be inhibited in the device. When a bit = 1, the device shall activate the System Max Inhibit control for that ring.

Bit 7 = Ring number = (ringControlGroupNumber * 8)

Bit 6 = Ring number = (ringControlGroupNumber * 8) - 1

Bit 5 = Ring number = (ringControlGroupNumber * 8) - 2

Bit 4 = Ring number = (ringControlGroupNumber * 8) - 3

Bit 3 = Ring number = (ringControlGroupNumber * 8) - 4

Bit 2 = Ring number = (ringControlGroupNumber * 8) - 5

Bit 1 = Ring number = (ringControlGroupNumber * 8) - 6

Bit 0 = Ring number = (ringControlGroupNumber * 8) - 7

The value of this object is ignored in BACKUP Mode. A write to this object shall reset the Backup timer."

REFERENCE

"NEMA TS 2 Clause 3.5.4.1.3"

::= { ringControlGroupEntry 5 }

2.8.5.6 Ring Ped Recycle Control

ringControlGroupPedRecycle OBJECT-TYPE

SYNTAX INTEGER (0..255)

ACCESS read-write

STATUS mandatory

DESCRIPTION

"This object is used to allow a remote entity to request a pedestrian recycle in the device. When a bit = 1, the device shall activate the System Ped Recycle control for that ring.

- Bit 7 = Ring number = (ringControlGroupNumber * 8)
- Bit 6 = Ring number = (ringControlGroupNumber * 8) - 1
- Bit 5 = Ring number = (ringControlGroupNumber * 8) - 2
- Bit 4 = Ring number = (ringControlGroupNumber * 8) - 3
- Bit 3 = Ring number = (ringControlGroupNumber * 8) - 4
- Bit 2 = Ring number = (ringControlGroupNumber * 8) - 5
- Bit 1 = Ring number = (ringControlGroupNumber * 8) - 6
- Bit 0 = Ring number = (ringControlGroupNumber * 8) - 7

The value of this object is ignored in BACKUP Mode. A write to this object shall reset the Backup timer."

REFERENCE

"NEMA TS 2 Clause 3.5.4.1.5"

::= { ringControlGroupEntry 6 }

2.8.5.7 Ring Red Rest Control

ringControlGroupRedRest OBJECT-TYPE

SYNTAX INTEGER (0..255)

ACCESS read-write

STATUS optional

DESCRIPTION

"This object is used to allow a remote entity to request red rest in the device. When a bit = 1, the device shall activate the System Red Rest control for that ring.

- Bit 7 = Ring number = (ringControlGroupNumber * 8)
- Bit 6 = Ring number = (ringControlGroupNumber * 8) - 1
- Bit 5 = Ring number = (ringControlGroupNumber * 8) - 2
- Bit 4 = Ring number = (ringControlGroupNumber * 8) - 3
- Bit 3 = Ring number = (ringControlGroupNumber * 8) - 4
- Bit 2 = Ring number = (ringControlGroupNumber * 8) - 5
- Bit 1 = Ring number = (ringControlGroupNumber * 8) - 6
- Bit 0 = Ring number = (ringControlGroupNumber * 8) - 7

The value of this object is ignored when in BACKUP Mode. A write to this object shall reset the Backup timer."

REFERENCE

"NEMA TS 2 Clause 3.5.4.1.2"

::= { ringControlGroupEntry 7 }

2.8.5.8 Ring Omit Red Control

ringControlGroupOmitRedClear OBJECT-TYPE

SYNTAX INTEGER (0..255)

ACCESS read-write

STATUS optional

DESCRIPTION

"This object is used to allow a remote entity to omit red clearances in the device. When a bit = 1, the device shall activate the System Omit Red Clear control for that ring.

Bit 7 = Ring number = (ringControlGroupNumber * 8)

Bit 6 = Ring number = (ringControlGroupNumber * 8) - 1

Bit 5 = Ring number = (ringControlGroupNumber * 8) - 2

Bit 4 = Ring number = (ringControlGroupNumber * 8) - 3

Bit 3 = Ring number = (ringControlGroupNumber * 8) - 4

Bit 2 = Ring number = (ringControlGroupNumber * 8) - 5

Bit 1 = Ring number = (ringControlGroupNumber * 8) - 6

Bit 0 = Ring number = (ringControlGroupNumber * 8) - 7

The value of this object is ignored in BACKUP Mode. A write to this object shall reset the Backup timer."

REFERENCE

"NEMA TS 2 Clause 3.5.4.1.4"

::= { ringControlGroupEntry 8 }

2.9 CHANNEL PARAMETERS

channel OBJECT IDENTIFIER

::= { asc 8 }

--This defines a node for supporting channel objects.

2.9.1 Maximum Channels

maxChannels OBJECT-TYPE

SYNTAX INTEGER (0..255)

ACCESS read-only

STATUS mandatory

DESCRIPTION

"The Maximum Number of Channels this Actuated Controller Unit supports. This object indicates the maximum rows which shall appear in the channelTable object."

::= { channel 1 }

2.9.2 Channel Table

channelTable OBJECT-TYPE

SYNTAX SEQUENCE OF ChannelEntry

ACCESS not-accessible

STATUS mandatory

DESCRIPTION

"A table containing Actuated Controller Unit channel parameters. The number of rows in this table is equal to the maxChannels object."

::= { channel 2 }

channelEntry OBJECT-TYPE
SYNTAX ChannelEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION
"Parameters for a specific Actuated Controller Unit channel."
INDEX { channelNumber }
 ::= { channelTable 1 }

ChannelEntry ::= SEQUENCE {
channelNumber INTEGER,
channelControlSource INTEGER,
channelControlType INTEGER,
channelFlash INTEGER,
channelDim INTEGER }

2.9.2.1 Channel Number

channelNumber OBJECT-TYPE
SYNTAX INTEGER (0..255)
ACCESS read-only
STATUS mandatory
DESCRIPTION
"The channel number for objects in this row. This value shall not exceed the maxChannels object value."
 ::= { channelEntry 1 }

2.9.2.2 Channel Control Source Parameters

channelControlSource OBJECT-TYPE
SYNTAX INTEGER (0..255)
ACCESS read-only
STATUS mandatory
DESCRIPTION
"This object defines the channel control source (which Phase or Overlap). The value shall not exceed maxPhases or maxOverlaps as determined by channelControlType object:
Value 00 = No Control (Not In Use)
Value 01 = Phase 01 or Overlap A
Value 02 = Phase 02 or Overlap B
||
Value 15 = Phase 15 or Overlap O
Value 16 = Phase 16 or Overlap P
etc."
 ::= { channelEntry 2 }

2.9.2.3 Channel Control Type Parameters

channelControlType OBJECT-TYPE
SYNTAX INTEGER { other (1),
phaseVehicle (2),
phasePedestrian (3),
overlap (4) }
ACCESS read-only
STATUS mandatory
DESCRIPTION
"This object defines the channel control type (Vehicle Phase, Pedestrian Phase, or Overlap):
phaseVehicle - The channel controls a vehicle phase display.

phase Pedestrian - The channel controls a pedestrian phase display.
overlap - The channel controls an overlap display.
other - The channel controls an other type of display."

::= { channelEntry 3 }

2.9.2.4 Channel Flash Parameters

channelFlash OBJECT-TYPE

SYNTAX INTEGER (0..255)

ACCESS read-only

STATUS mandatory

DESCRIPTION

"This object defines the channel state during Automatic Flash.

Bit Function

7-4 Reserved

3 Flash Alternate Half Hertz (Bit 0 = Off / Disabled & 1 = On / Enabled)

2 Flash Red (Bit 0 = Off / Red Dark & 1 = On / Flash Red)

1 Flash Yellow (Bit 0 = Off / Yellow Dark & 1 = On / Flash Yellow)

0 Reserved"

::= { channelEntry 4 }

2.9.2.5 Channel Dim Parameters

channelDim OBJECT-TYPE

SYNTAX INTEGER (0..255)

ACCESS read-only

STATUS mandatory

DESCRIPTION

"This object defines the channel state during Dimming. Dimming shall be accomplished by the elimination of alternate one-half segments from the AC sinusoid applied to the field terminals.

Bit Function

7-4 Reserved

3 Dim Alternate Half Line Cycle (Bit 0 = Off / + half cycle & 1 = On / - half cycle)

2 Dim Red - Bit 0 = Off / Red Not Dimmed & 1 = On / Dimmed Red

1 Dim Yellow - Bit 0 = Off / Yellow Not Dimmed & 1 = On / Dimmed Yellow

0 Dim Green - Bit 0 = Off / Green Not Dimmed & 1 = On / Dimmed Green"

::= { channelEntry 5 }

2.9.3 Maximum Channel Status Groups

maxChannelStatusGroups OBJECT-TYPE

SYNTAX INTEGER (0..255)

ACCESS read-only

STATUS mandatory

DESCRIPTION

"The maximum number of Channel Status Groups (8 channels per group) this Actuated Controller Unit supports. This value is equal to TRUNCATE [(maxChannels + 7) / 8]. This object indicates the maximum rows which shall appear in the channelStatusGroupTable object."

::= { channel 3 }

2.9.4 Channel Status Group Table

channelStatusGroupTable OBJECT-TYPE
SYNTAX SEQUENCE OF ChannelStatusGroupEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION
"A table containing Actuated Controller Unit channel output (Red, Yellow, & Green) status in groups of eight channels. The number of rows in this table is equal to the maxChannelStatusGroups object."
 ::= { channel 4 }

channelStatusGroupEntry OBJECT-TYPE
SYNTAX ChannelStatusGroupEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION
"Red, Yellow, & Green Output Status for eight Actuated Controller Unit channels."
INDEX { channelStatusGroupNumber }
 ::= { channelStatusGroupTable 1 }

ChannelStatusGroupEntry ::= SEQUENCE {
channelStatusGroupNumber INTEGER,
channelStatusGroupReds INTEGER,
channelStatusGroupYellows INTEGER,
channelStatusGroupGreens INTEGER }

2.9.4.1 Channel Status Group Number

channelStatusGroupNumber OBJECT-TYPE
SYNTAX INTEGER (0..255)
ACCESS read-only
STATUS mandatory
DESCRIPTION
"The channelStatusGroup number for objects in this row. This value shall not exceed the maxChannelStatusGroups object value."
 ::= { channelStatusGroupEntry 1 }

2.9.4.2 Channel Status Group Reds

channelStatusGroupReds OBJECT-TYPE
SYNTAX INTEGER (0..255)
ACCESS read-only
STATUS mandatory
DESCRIPTION
"Channel Red Output Status Mask, when a bit = 1, the Channel Red is currently active. When a bit = 0, the Channel Red is NOT currently active.
Bit 7 = Channel number = (channelStatusGroupNumber * 8)
Bit 6 = Channel number = (channelStatusGroupNumber * 8) - 1
Bit 5 = Channel number = (channelStatusGroupNumber * 8) - 2
Bit 4 = Channel number = (channelStatusGroupNumber * 8) - 3
Bit 3 = Channel number = (channelStatusGroupNumber * 8) - 4
Bit 2 = Channel number = (channelStatusGroupNumber * 8) - 5
Bit 1 = Channel number = (channelStatusGroupNumber * 8) - 6
Bit 0 = Channel number = (channelStatusGroupNumber * 8) - 7"
 ::= { channelStatusGroupEntry 2 }

2.9.4.3 Channel Status Group Yellows

channelStatusGroupYellows OBJECT-TYPE

SYNTAX INTEGER (0..255)

ACCESS read-only

STATUS mandatory

DESCRIPTION

"Channel Yellow Output Status Mask, when a bit = 1, the Channel Yellow is currently active.

When a bit = 0, the Channel Yellow is NOT currently active.

Bit 7 = Channel number = (channelStatusGroupNumber * 8)

Bit 6 = Channel number = (channelStatusGroupNumber * 8) - 1

Bit 5 = Channel number = (channelStatusGroupNumber * 8) - 2

Bit 4 = Channel number = (channelStatusGroupNumber * 8) - 3

Bit 3 = Channel number = (channelStatusGroupNumber * 8) - 4

Bit 2 = Channel number = (channelStatusGroupNumber * 8) - 5

Bit 1 = Channel number = (channelStatusGroupNumber * 8) - 6

Bit 0 = Channel number = (channelStatusGroupNumber * 8) - 7"

::= { channelStatusGroupEntry 3 }

2.9.4.4 Channel Status Group Greens

channelStatusGroupGreens OBJECT-TYPE

SYNTAX INTEGER (0..255)

ACCESS read-only

STATUS mandatory

DESCRIPTION

"Channel Green Output Status Mask, when a bit = 1, the Channel Green is currently active.

When a bit = 0, the Channel Green is NOT currently active.

Bit 7 = Channel number = (channelStatusGroupNumber * 8)

Bit 6 = Channel number = (channelStatusGroupNumber * 8) - 1

Bit 5 = Channel number = (channelStatusGroupNumber * 8) - 2

Bit 4 = Channel number = (channelStatusGroupNumber * 8) - 3

Bit 3 = Channel number = (channelStatusGroupNumber * 8) - 4

Bit 2 = Channel number = (channelStatusGroupNumber * 8) - 5

Bit 1 = Channel number = (channelStatusGroupNumber * 8) - 6

Bit 0 = Channel number = (channelStatusGroupNumber * 8) - 7"

::= { channelStatusGroupEntry 4 }

2.10 OVERLAP PARAMETERS

overlap OBJECT IDENTIFIER

::= { asc 9 }

--"This node contains objects that configure, monitor and control overlap functions."

2.10.1 Maximum Overlaps

maxOverlaps OBJECT-TYPE

SYNTAX INTEGER (0..255)

ACCESS read-only

STATUS mandatory

DESCRIPTION

"The Maximum Number of Overlaps this Actuated Controller Unit supports. This object indicates the maximum number of rows which shall appear in the overlapTable object."

::= { overlap 1 }

2.10.2 Overlap Table

overlapTable OBJECT-TYPE
SYNTAX SEQUENCE OF OverlapEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION
"A table containing Actuated Controller Unit overlap parameters. The number of rows in this table is equal to the maxOverlaps object."
 ::= { overlap 2 }

overlapEntry OBJECT-TYPE
SYNTAX OverlapEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION
"Parameters for a specific Actuated Controller Unit overlap."
INDEX { overlapNumber }
 ::= { overlapTable 1 }

OverlapEntry ::= SEQUENCE {
overlapNumber INTEGER,
overlapType INTEGER,
overlapIncludedPhases OCTET STRING,
overlapModifierPhases OCTET STRING,
overlapTrailGreen INTEGER,
overlapTrailYellow INTEGER,
overlapTrailRed INTEGER }

2.10.2.1 Overlap Number

overlapNumber OBJECT-TYPE
SYNTAX INTEGER (0..255)
ACCESS read-only
STATUS mandatory
DESCRIPTION
"The overlap number for objects in this row. The value shall not exceed the maxOverlaps object. The value maps to the Overlap as follows: 1 = Overlap A, 2 = Overlap B etc."
 ::= { overlapEntry 1 }

2.10.2.2 Overlap Type

overlapType OBJECT-TYPE
SYNTAX INTEGER { other(1),
normal (2),
minusGreenYellow (3) }
ACCESS read-write
STATUS mandatory
DESCRIPTION
"The type of overlap operation for this row. The types are as follows:
normal - The overlap output shall be controlled by the overlapIncludedPhases when this type is selected. The overlap output shall be green in the following situations:
(1) when an overlap included phase is green.
(2) when an overlap included phase is yellow (or red clearance) and an overlap included phase is next.

The overlap output shall be yellow when an included phase is yellow and an

overlap included phase is not next.

The overlap output shall be red whenever the overlap green and yellow are not ON.

minusGreenYellow - The overlap output shall be controlled by the overlapIncludedPhases and the overlapModifierPhases if this type is selected.

The overlap output shall be green in the following situations:

(1) when an overlap included phase is green and an overlap modifier phase is NOT green.

(2) when an overlap included phase is yellow (or red clearance) and an overlap included phase is next and an overlap modifier phase is NOT green.

The overlap output shall be yellow when an overlap included phase is yellow and an overlap modifier phase is NOT yellow and an overlap included phase is not next.

The overlap output shall be red whenever the overlap green and yellow are not ON.

other - The overlap operates in another mode than those described above."

::= { overlapEntry 2 }

2.10.2.3 Overlap Included Phase Parameters

overlapIncludedPhases OBJECT-TYPE

SYNTAX OCTET STRING

ACCESS read-write

STATUS mandatory

DESCRIPTION

"Each octet is a Phase (number) that shall be an included phase for the overlap. The phase number value shall not exceed the maxPhases object value. When an included phase output is green or when the CU is cycling between included phases, the overlap output shall be green."

::= { overlapEntry 3 }

2.10.2.4 Overlap Modifier Phase Parameters

overlapModifierPhases OBJECT-TYPE

SYNTAX OCTET STRING

ACCESS read-write

STATUS mandatory

DESCRIPTION

"Each octet is a Phase (number) that shall be a modifier phase for the overlap. The phase number value shall not exceed the maxPhases object value. The function of the modifier phase(s) is defined by the overlapType selected."

::= { overlapEntry 4 }

2.10.2.5 Overlap Trailing Green Parameter

overlapTrailGreen OBJECT-TYPE

SYNTAX INTEGER (0..255)

ACCESS read-write

STATUS mandatory

DESCRIPTION

"Overlap Trailing Green Parameter in seconds (0-255 sec). When this value is greater than zero and the overlap green would normally terminate, the overlap green shall be extended by this additional time."

::= { overlapEntry 5 }

2.10.2.6 Overlap Trailing Yellow Change Parameter

overlapTrailYellow OBJECT-TYPE
SYNTAX INTEGER (0..255)
ACCESS read-write
STATUS mandatory
DESCRIPTION
"Overlap Trailing Yellow Change Parameter in tenth seconds (NEMA range: 3.0-25.5) seconds. When the overlap green has been extended (Trailing Green), this value shall determine the current length of the Yellow Change interval for the overlap."
::= { overlapEntry 6 }

2.10.2.7 Overlap Trailing Red Clear Parameter

overlapTrailRed OBJECT-TYPE
SYNTAX INTEGER (0..255)
ACCESS read-write
STATUS mandatory
DESCRIPTION
"Overlap Trailing Red Clear Parameter in tenth seconds (0-25.5 sec). When the overlap green has been extended (Trailing Green), this value shall determine the current length of the Red Clearance interval for the overlap."
::= { overlapEntry 7 }

2.10.3 Maximum Overlap Status Groups

maxOverlapStatusGroups OBJECT-TYPE
SYNTAX INTEGER (0..255)
ACCESS read-only
STATUS mandatory
DESCRIPTION
"The Maximum Number of Overlap Status Groups (8 overlaps per group) this Actuated Controller Unit supports. This value is equal to TRUNCATE [(maxOverlaps + 7) / 8]. This object indicates the maximum rows which shall appear in the overlapStatusGroupTable object."
::= { overlap 3 }

2.10.4 Overlap Status Group Table

overlapStatusGroupTable OBJECT-TYPE
SYNTAX SEQUENCE OF OverlapStatusGroupEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION
"A table containing Actuated Controller Unit overlap output (Red, Yellow, & Green) status in groups of eight overlaps. The number of rows in this table is equal to the maxOverlapStatusGroups object."
::= { overlap 4 }

overlapStatusGroupEntry OBJECT-TYPE
SYNTAX OverlapStatusGroupEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION
"Red, Yellow, & Green Output Status for eight Actuated Controller Unit overlaps."
INDEX { overlapStatusGroupNumber }
::= { overlapStatusGroupTable 1 }

```
OverlapStatusGroupEntry ::= SEQUENCE {  
    overlapStatusGroupNumber      INTEGER,  
    overlapStatusGroupReds        INTEGER,  
    overlapStatusGroupYellows     INTEGER,  
    overlapStatusGroupGreens      INTEGER }
```

2.10.4.1 Overlap Status Group Number

```
overlapStatusGroupNumber OBJECT-TYPE  
    SYNTAX  INTEGER (0..255)  
    ACCESS  read-only  
    STATUS  mandatory  
    DESCRIPTION  
        "The overlap StatusGroup number for objects in this row. This value shall not exceed the  
        maxOverlapStatusGroups object value."  
 ::= { overlapStatusGroupEntry 1 }
```

2.10.4.2 Overlap Status Group Reds

```
overlapStatusGroupReds OBJECT-TYPE  
    SYNTAX  INTEGER (0..255)  
    ACCESS  read-only  
    STATUS  mandatory  
    DESCRIPTION  
        "Overlap Red Output Status Mask, when a bit = 1, the Overlap Red is currently active. When  
        a bit = 0, the Overlap Red is NOT currently active.  
        Bit 7 = Overlap number = (overlapStatusGroupNumber * 8)  
        Bit 6 = Overlap number = (overlapStatusGroupNumber * 8) - 1  
        Bit 5 = Overlap number = (overlapStatusGroupNumber * 8) - 2  
        Bit 4 = Overlap number = (overlapStatusGroupNumber * 8) - 3  
        Bit 3 = Overlap number = (overlapStatusGroupNumber * 8) - 4  
        Bit 2 = Overlap number = (overlapStatusGroupNumber * 8) - 5  
        Bit 1 = Overlap number = (overlapStatusGroupNumber * 8) - 6  
        Bit 0 = Overlap number = (overlapStatusGroupNumber * 8) - 7"  
 ::= { overlapStatusGroupEntry 2 }
```

2.10.4.3 Overlap Status Group Yellows

```
overlapStatusGroupYellows OBJECT-TYPE  
    SYNTAX  INTEGER (0..255)  
    ACCESS  read-only  
    STATUS  mandatory  
    DESCRIPTION  
        "Overlap Yellow Output Status Mask, when a bit = 1, the Overlap Yellow is currently active.  
        When a bit = 0, the Overlap Yellow is NOT currently active.  
        Bit 7 = Overlap number = (overlapStatusGroupNumber * 8)  
        Bit 6 = Overlap number = (overlapStatusGroupNumber * 8) - 1  
        Bit 5 = Overlap number = (overlapStatusGroupNumber * 8) - 2  
        Bit 4 = Overlap number = (overlapStatusGroupNumber * 8) - 3  
        Bit 3 = Overlap number = (overlapStatusGroupNumber * 8) - 4  
        Bit 2 = Overlap number = (overlapStatusGroupNumber * 8) - 5  
        Bit 1 = Overlap number = (overlapStatusGroupNumber * 8) - 6  
        Bit 0 = Overlap number = (overlapStatusGroupNumber * 8) - 7"  
 ::= { overlapStatusGroupEntry 3 }
```

2.10.4.4 Overlap Status Group Greens

overlapStatusGroupGreens OBJECT-TYPE

SYNTAX INTEGER (0..255)

ACCESS read-only

STATUS mandatory

DESCRIPTION

"Overlap Green Output Status Mask, when a bit = 1, the Overlap Green is currently active.

When a bit = 0, the Overlap Green is NOT currently active.

Bit 7 = Overlap number = (overlapStatusGroupNumber * 8)

Bit 6 = Overlap number = (overlapStatusGroupNumber * 8) - 1

Bit 5 = Overlap number = (overlapStatusGroupNumber * 8) - 2

Bit 4 = Overlap number = (overlapStatusGroupNumber * 8) - 3

Bit 3 = Overlap number = (overlapStatusGroupNumber * 8) - 4

Bit 2 = Overlap number = (overlapStatusGroupNumber * 8) - 5

Bit 1 = Overlap number = (overlapStatusGroupNumber * 8) - 6

Bit 0 = Overlap number = (overlapStatusGroupNumber * 8) - 7"

::= { overlapStatusGroupEntry 4 }

2.11 TS2 PORT 1 PARAMETERS

ts2port1 OBJECT IDENTIFIER

::= { asc 10 }

-- This object is an identifier used to group all objects for support of TS 2 port1 activities.

2.11.1 Maximum Port 1 Addresses

maxPort1Addresses OBJECT-TYPE

SYNTAX INTEGER (0..255)

ACCESS read-only

STATUS mandatory

DESCRIPTION

"The Maximum Number of Port 1 addresses this Actuated Controller Unit supports. This object indicates the maximum rows which shall appear in the port1Table object."

::= { ts2port1 1 }

2.11.2 Port 1 Table

port1Table OBJECT-TYPE

SYNTAX SEQUENCE OF Port1Entry

ACCESS not-accessible

STATUS mandatory

DESCRIPTION

"A table containing Actuated Controller Unit port 1 parameters. The number of rows in this table is equal to the maxPort1Addresses object."

::= { ts2port1 2 }

port1Entry OBJECT-TYPE

SYNTAX Port1Entry

ACCESS not-accessible

STATUS mandatory

DESCRIPTION

"This object defines a conceptual row in the port 1 Table."

INDEX { port1Number }

::= { port1Table 1 }

```
Port1Entry ::= SEQUENCE {  
    port1Number          INTEGER,  
    port1DevicePresent   INTEGER,  
    port1Frame40Enable   INTEGER,  
    port1Status          INTEGER,  
    port1FaultFrame      INTEGER }
```

2.11.2.1 Port 1 Number

```
port1Number OBJECT-TYPE  
    SYNTAX INTEGER (0..255)  
    ACCESS read-only  
    STATUS mandatory  
    DESCRIPTION  
        "The Port 1 address for objects in this row. This value shall not exceed the  
        maxPort1Addresses object value."  
 ::= { port1Entry 1 }
```

2.11.2.2 Port 1 Device Present

```
port1DevicePresent OBJECT-TYPE  
    SYNTAX INTEGER (0..1)  
    ACCESS read-only  
    STATUS mandatory  
    DESCRIPTION  
        "The presence or absence of a device for this port 1 address. Command Frames shall be  
        transmitted only to those devices that are present as determined by this programming. If the  
        object is TRUE (non-zero) then the device is present. If the object is FALSE (zero) then the  
        device is not present."  
    REFERENCE  
        "NEMA TS 2 Clause 3.3.1.4"  
 ::= { port1Entry 2 }
```

2.11.2.3 Port 1 Frame 40 Enable

```
port1Frame40Enable OBJECT-TYPE  
    SYNTAX INTEGER (0..1)  
    ACCESS read-only  
    STATUS mandatory  
    DESCRIPTION  
        "To enable or disable Frame 40 messages to the device at this port 1 address. Frame 40 is  
        used to poll the secondary stations for a secondary to secondary message exchange.  
        Command 40 series frames shall be transmitted only to those devices that are enabled, as  
        determined by this programming.  
        TRUE (non-zero) - Enable frame 40 messages for this device.  
        FALSE (zero) - Disable frame 40 messages for this device."  
    REFERENCE  
        "NEMA TS 2 Clause 3.3.1.4.1"  
 ::= { port1Entry 3 }
```

2.11.2.4 Port 1 Status

port1Status OBJECT-TYPE

SYNTAX INTEGER {
 other (1),
 online (2),
 responseFault (3) }

ACCESS read-write

STATUS mandatory

DESCRIPTION

"This object indicates the communications status with the associated device:

online - This indicates that at least five of the most recent 10 response transfers were received correctly

responseFault - This indicates that more than 5 of the most recent 10 response transfers were received incorrectly.

other - This indicates that some other communications fault has been detected."

::= { port1Entry 4 }

2.11.2.5 Port 1 Fault Frame

port1FaultFrame OBJECT-TYPE

SYNTAX INTEGER (0..255)

ACCESS read-only

STATUS mandatory

DESCRIPTION

"This object indicates the frame number that caused the most recent fault."

::= { port1Entry 5 }

END

Section 3 Group Definitions

A Conformance Group is defined in NEMA TS 3.2 Clause 3.3.5.

Conformance Groups are defined as either mandatory or optional. If a Conformance Group is mandatory, all of the objects with STATUS "mandatory" that are part of the Conformance Group shall be present for a device to claim conformance to the Group. If a Conformance Group is optional, all of the objects that are part of the Conformance Group with the STATUS "mandatory" shall be present if the device supports the Conformance Group. Objects with the STATUS "optional" may be supported.

When a table is included in a Conformance Group, all objects contained in the table are included by reference. This is because a table is defined as a SEQUENCE OF {SEQUENCE}. Thus, all objects listed in the sequence are defined as an integral part of the table. Tables are defined as either mandatory or optional. If a table is mandatory, all of the objects with STATUS "mandatory" shall be present. If a table is optional, all of the objects with the STATUS "mandatory" shall be present if the device supports the table. Objects in the table with the STATUS "optional" may be supported.

**Table 3-1
OBJECT SUPPORT REQUIREMENTS**

Object Status	Table Status	Group Status	Object Support
mandatory	mandatory	mandatory	mandatory
mandatory	mandatory	optional	mandatory, if group is supported
mandatory	optional	mandatory	mandatory, if table is supported
mandatory	optional	optional	mandatory, if both the group and table are supported
optional	mandatory	mandatory	optional
optional	mandatory	optional	optional
optional	optional	mandatory	optional
optional	optional	optional	optional

The Conformance Group definitions for actuated signal controllers are defined in this section. An actuated signal controller has multiple functions; thus, Conformance Groups are defined for each function.

3.1 PHASE CONFORMANCE GROUP

The Phase Conformance Group consists of the phase timing parameters, and phase status variables. The Phase Group shall consist of the following objects:

Object or Group Name	Reference
maxPhases	TS 3.5
phaseTable	TS 3.5
maxPhaseGroups	TS 3.5
phaseStatusGroupTable	TS 3.5

3.2 DETECTOR CONFORMANCE GROUP

The Detector Conformance Group consists of the detector configuration parameters, and the detector status variables. The Detector Group consists of the following objects:

Object or Group Name	Reference
maxVehicleDetectors	TS 3.5
vehicleDetectorTable	TS 3.5
maxVehicleDetectorStatusGroups	TS 3.5
vehicleDetectorStatusGroupTable	TS 3.5
maxPedestrianDetectors	TS 3.5
pedestrianDetectorTable	TS 3.5

3.3 VOLUME OCCUPANCY REPORT CONFORMANCE GROUP

The Volume Occupancy Report Group shall consist of the following objects:

Object or Group Name	Reference
volumeOccupancySequence	TS 3.5
volumeOccupancyPeriod	TS 3.5
activeVolumeOccupancyDetectors	TS 3.5
volumeOccupancyTable	TS 3.5

3.4 UNIT CONFORMANCE GROUP

The Unit Group consists of objects related to overall controller configuration and state. The Unit Group shall consist of the following objects:

Object or Group Name	Reference
unitStartupFlash	TS 3.5
unitBackupTime	TS 3.5
unitRedRevert	TS 3.5
unitControlStatus	TS 3.5
unitFlashStatus	TS 3.5
unitAlarmStatus2	TS 3.5
unitAlarmStatus1	TS 3.5
shortAlarmStatus	TS 3.5
unitControl	TS 3.5
maxAlarmGroups	TS 3.5
alarmGroupTable	TS 3.5

3.5 SPECIAL FUNCTION CONFORMANCE GROUP

The Special Function Conformance Group consists of those objects related to control of special function outputs.

Object or Group Name	Reference
maxSpecialFunctionOutputs	TS 3.5
specialFunctionOutputTable	TS 3.5

3.6 COORDINATION CONFORMANCE GROUP

The Coordination Group consists of those objects related to signal coordination. The Coordination Group shall consist of the following objects:

Object or Group Name	Reference
coordOperationalMode	TS 3.5
coordCorrectionMode	TS 3.5
coordMaximumMode	TS 3.5
coordForceMode	TS 3.5
maxPatterns	TS 3.5
patternTableType	TS 3.5
patternTable	TS 3.5
maxSplits	TS 3.5
splitTable	TS 3.5
coordPatternStatus	TS 3.5
localFreeStatus	TS 3.5
coordCycleStatus	TS 3.5
coordSyncStatus	TS 3.5
systemPatternControl	TS 3.5
systemSyncControl	TS 3.5

3.7 TIME BASE CONFORMANCE GROUP

The Time Base Group consists of the asc specific objects related to time base operation. The Time Base Group shall consist of the following objects:

Object or Group Name	Reference
Time Management Conf. Group	TS 3.4
timebasePatternSync	TS 3.5
maxTimebaseAscActions	TS 3.5
timebaseAscActionTable	TS 3.5
timebaseAscActionStatus	TS 3.5

3.8 PREEMPT CONFORMANCE GROUP

The Preempt Group consists of the preempt configuration parameters, and the preempt status variables. The Preempt Conformance Group shall consist of the following objects:

Object or Group Name	Reference
maxPreempts	TS 3.5
preemptTable	TS 3.5

3.9 RING CONFORMANCE GROUP

The Ring Group consists of the ring configuration parameters, status variables. The Ring Conformance Group shall consist of the following objects:

Object or Group Name	Reference
maxRings	TS 3.5
maxSequences	TS 3.5
sequenceTable	TS 3.5

3.10 CHANNEL CONFORMANCE GROUP

The Channel Group consists of the channel configuration parameters, and status variables. The Channel Group shall consist of the following objects:

Object or Group Name	Reference
maxChannels	TS 3.5
channelTable	TS 3.5
maxChannelStatusGroups	TS 3.5
channelStatusGroupTable	TS 3.5

3.11 OVERLAP CONFORMANCE GROUP

The Overlap Group consists of the overlap configuration parameters, and status variables. The Overlap Group shall consist of the following objects:

Object or Group Name	Reference
maxOverlaps	TS 3.5
overlapTable	TS 3.5
maxOverlapStatusGroups	TS 3.5
overlapStatusGroupTable	TS 3.5

3.12 TS-2 PORT1 CONFORMANCE GROUP

The TS-2 Port1 Conformance Group consists of the TS-2 port 1 configuration parameters, status variables, and control objects. The TS-2 Group shall consist of the following objects:

Object or Group Name	Reference
maxPort1Addresses	TS 3.5
port1Table	TS 3.5

Section 4 CONFORMANCE STATEMENTS

Actuated Signal Controller (ASC) devices shall adhere to the conformance requirements specified in Table 4-1 as a minimum to claim compliance to this standard. Additional objects or groups may be supported without being non-compliant with ASC objects or NTCIP.

Minimum and maximum ranges of objects that differ from the values of the object's SYNTAX field may be enforced by an application running on a device.

A device which enforces range limits within the bounds specified by the values of the object's SYNTAX field shall not be categorized as being non-compliant with ASC objects or NTCIP.

A device which supports a subset of objects with enumerated values shall not be categorized as being non-compliant with ASC objects or NTCIP.

**Table 4-1
CONFORMANCE TABLE**

Conformance Group	Reference	Conformance Requirement
Configuration	TS 3.4	mandatory
Database Management	TS 3.4	optional
Time Management	TS 3.4	optional
Timebase Event Schedule	TS 3.4	optional
Report	TS 3.4	optional
STMP	TS 3.4	optional
PMPP	TS 3.4	optional
Phase	TS 3.5	mandatory
Detector	TS 3.5	mandatory
Volume Occupancy Report	TS 3.5	optional
Unit	TS 3.5	optional
Special Function	TS 3.5	optional
Coordination	TS 3.5	optional
Time Base	TS 3.5	optional
Preempt	TS 3.5	optional
Ring	TS 3.5	optional
Channel	TS 3.5	optional
Overlap	TS 3.5	optional
TS 2 Port 1	TS 3.5	optional

