Miami-Dade County Traffic Signal Controller Local Software Functional and Performance Specification

Miami-Dade Traffic Signal Controller Local Software

Functional and Performance Specification

1. Introduction

1.1 Purpose

1.1.1. This specification describes the features and functional requirements for Miami-Dade Traffic Signal Controller Local Software. The controller software must be a Commercially-available Off-the-Shelf (COTS) product designed for operation in County-approved Caltrans Model 2070LX traffic signal controllers. The controller software must be compatible with the central traffic management system software used by the Miami-Dade County Department of Transportation and Public Works (DTPW) Traffic Management Center (TMC).

1.2 Controller software overview

1.2.1. The controller software must be a COTS product from the traffic controller manufacturer designed to run on a Linux operating system and 2070LX hardware platform compliant with MDC traffic signal controller hardware specifications and Caltrans TEES 2009 requirements including all Errata for a 2070LX Traffic Signal Controller unit. In addition, the controller software shall be compatible with the following ATC controller hardware platforms:

1.2.1.1. ATC 2070

1.2.1.2. ATC NEMA TS2 Type 1

1.2.1.3. ATC NEMA TS2 Type 2

1.3 Software attributes

1.3.1. Compatibility and Standards Conformance

1.3.2. The controller software must operate on the Miami-Dade County network and not interfere with other systems. The controller software must be compatible with all 170 type cabinets, including, but not limited to Miami-Dade County MD-552 and MD-660 series controller cabinets.

1.3.3. The controller software must fully support Caltrans TEES 2009 2070LX and ATC 5201 Advanced Transportation Controller Standard v06 (v06.25) controller requirements.

1.3.4. The controller software must provide functionality that meets or exceeds operational characteristics, including National Transportation Communications for ITS (Intelligent Transportation Systems) Protocol (NTCIP) support, as described in National Electrical Manufacturers Association (NEMA) TS-2-2016.

1.3.5. The controller software must support all mandatory traffic signal controller requirements and objects defined in NTCIP 1202 as well as Appendix A, Miami-Dade Traffic Signal Controller NTCIP requirements.

1.3.6. The controller software must not rely upon proprietary objects or protocols for communication.

1.3.7. Controller software must be provided with the entire MIB in electronic and hard copy. Controller software use must include rights to re-use/redistribute the entire MIB including any extensions as necessary to communicate to the Central Software.

1.3.8. The controller software must be fully functional, compatible, and readily loadable onto all Model 2070LX controllers listed on the Miami-Dade County Traffic Signals and Signs Division Qualified Products List.

1.4 Reliability

1.4.1. The controller software must include features that help achieve high reliability, such as modularity, backup, and recovery.

1.5 Support and maintenance

1.5.1. The controller software license must provide Miami-Dade County with proven and tested updates at no cost for as long as the controller hardware is supported by its manufacturer.

2. General and Administrative

2.1 Intersection Information

2.1.1. The controller software must be able to display intersection information including: date and time, phase, operating mode, preemption, and coordination status.

2.1.2. The controller software must allow the user to view version information.

2.2 Software Updates

2.2.1. The controller software must permit authorized users to update the operating system and traffic signal application via USB and laptop computer.

2.2.2. Security and Programming

2.2.3. The controller software must allow programming locally via the display panel, locally via USB, locally via Data Key, locally via a laptop PC by a serial or Ethernet cable, and remotely from a Central ATMS software via NTCIP commands.

2.2.4. The controller software must allow administrators to set multiple user IDs with unique passcodes.

2.2.5. The controller software must provide a minimum of 3 security levels (e.g., Administrator, read-write for configuration settings except user IDs and passcodes, and read-only).

2.2.6. The controller software must reside in non-volatile flash memory.

2.2.7. The controller software must check file compatibility and integrity of updates prior to implementation.

2.3 Copy/Cut/Paste

2.3.1. The controller software must support central software user ability to copy data from one phase, timing plan, detector plan, detector options, pattern, or sequence to another like feature to facilitate programming where values are the same.

2.3.2. The controller software must provide the ability to create and store a backup database that can be used to restore a controller or program a new controller.

2.4 Changing and Saving Controller Settings

2.4.1. The controller software must prompt the user to confirm changes prior to committing them to the controller database for implementation.

2.4.2. The controller software must highlight changes to settings prior to committing them to the controller database for implementation.

2.4.3. The controller software must prompt the user prior to activating changes.

2.4.4. The controller software must have capability to upload and download controller database via USB and data key.

3. Signal Control

3.1 Display

3.1.1. Display menu must provide access to control all programmable features.

3.1.2. The controller software must include a help system that provides information regarding the currently active menu or selected parameter.

3.2 Phase Sequence and Rings

3.2.1. The controller software must support programming up to sixteen phases for vehicles.

3.2.2. The controller software must support programming up to sixteen phases for pedestrians.

3.2.3. The controller software must support up to sixteen configurable phase sequences.

3.2.4. The controller software must support up to four rings.

3.2.5. The controller software must be able to turn on and turn off exclusive pedestrian phases by TOD/DOW/Temporary time base schedule commands which will time and display pedestrian indications with vehicle movements remaining in all red.

3.3 Input & Output Assignments

3.3.1. I/O mapping and pin assignment must meet standard C1 connector and terminal requirements with the exceptions for Miami-Dade noted below.

3.3.1.1. Pin 49 - Spare is Reserved

3.3.1.2. Pin 50 - Spare is Reserved

3.3.1.3. Pin 53 - Spare is Reserved

3.3.1.4. Pin 54 - Spare is Reserved

3.3.1.5. Pin 66 - Pretime is Reserved

3.3.1.6. Pin 68 - Spare Input is Reserved

3.3.1.7. Pin 91 - Audible Ped is Audible Ped for late night mode

3.3.2. The controller software must allow authorized users to assign all inputs and outputs.

3.3.3. The controller software must support flashing yellow arrow operation.

3.3.4. The controller must support Pedestrian Hybrid Beacon operation.

3.4 Programmable Logic

3.4.1. The controller software must support logic commands including AND, OR, NOR and NAND, and latching operations.

3.4.2. Logic gates must support delay and extension timing for all inputs and outputs.

3.5 Startup Operations

3.5.1. The controller software startup conditions are configurable to include a user defined all-red or flash interval.

3.5.2. The controller software must permit users to assign startup phases, flash duration, all red duration, side street green, main street green, and vehicle and pedestrian calls on startup.

3.6 Phase Timing Parameters

3.6.1. The controller software must permit unique timing parameters and allow users to define minimum and maximum green time and secondary maximum green time, yellow clearance time, and red clearance time for any phase.

3.6.2. The controller software must allow users to define secondary maximum green time for any phase.

3.6.3. The signal controller software must permit users to define the amount of time necessary between detector actuations to cause a gap out for any phase.

3.6.4. The signal controller software must permit users to define the duration of Flashing Don't Walk time for any phase and must allow the users to include the clearance time if pedestrian operation is negative to overlaps.

3.6.5. The controller software must support rest in walk, advance and delay pedestrian walk, flashing don't walk time and green time.

3.6.6. The controller software must support gap reduction and dynamic max limit defined by user

3.6.7. The controller software must support conditional service and phase conditional re-service for any phase.

3.6.8. The controller software shall be capable of providing a Leading Pedestrian Interval

3.7 Phase Option Parameters

3.7.1. The controller software must support phase omit per TOD/DOW.

3.7.2. The controller software must support minimum and maximum recall for any active phase per TOD/DOW.

3.7.3. The controller software must prevent a yellow trap via detector switching or similar option.

3.8 Overlaps

3.8.1. The controller software must support programming up to sixteen overlaps for vehicles.

3.8.2. The controller software must support programming up to sixteen overlaps for pedestrians. The controller software must support negative (or minus) pedestrian overlap which allows the controller to serve a pedestrian phase without starting the conflicting overlap until the programmed pedestrian phase has completed to solid don't walk. If the overlap is already on when a pedestrian phase should be serviced, the pedestrian phase will be delayed until the conflicting overlap is terminated.

3.8.3. The controller software must be able to time green, yellow, and all red times.

3.8.4. The signal controller software must permit users to define compatible and incompatible phases with overlaps.

3.9 Detectors

3.9.1. The controller software must support a minimum of 64 vehicle detectors and 16 pedestrian detectors.

3.9.2. The signal controller software must provide a minimum of 3 unique sets of detectors parameters.

3.9.3. The signal controller software must allow users to assign attributes to each vehicle and pedestrian detectors and use it as counting detectors.

3.10 Coordination

3.10.1. Coordination functions to control intersection cycle lengths, system offset relationships, and phase split percentages are provided as a standard feature, with no need for additional modules or software.

3.10.2. Coordination, Time of Day, and external alternate sequence can alter the standard phase by selecting a configurable sequence.

3.10.3. Offset correction/transition mode and maximum number of cycles defined by user.

3.10.4. The controller software provides configurable minimum values for green, walk, pedestrian clearance, yellow clearance, red clearance and overlap. These values cannot be changed by the controller.

3.11 Date and Time Functions

3.11.1. The controller software must include a clock that keeps track of the current date and time. Ensure the clock can be configured to a local time based on the time zone offset and be remotely set.

3.11.2. Controller software must support scheduling based on time of day and external triggering. The scheduler must permit the user to define the months, days of week, and date of month for assigned day plans.

3.11.3. The scheduler must permit users to define and use time-of-day for functions including, but not limited to timings, coordination patterns, detector plans, automatic flash, and special functions.

3.11.4. The controller software must be able to adjust daylight saving time automatically.

3.12 Preemption and Priority

3.12.1. The controller software must provide control to enable or disable preemption by phase. The controller software must include a user configurable duration and sequence that the preemption signal is active prior to initiating the signal preemption. Preemption duration must be configurable.

3.12.1.1. The controller software must permit the user to define an all red state prior to entering preemption.

3.12.1.2. The controller software must allow the user to define preemption exit settings, including phases for exiting preemption and the recovery period.

3.12.1.3. The preemption priority must allow “nested” preemption.

3.12.2. Emergency Preemption

3.12.2.1. The controller software must support a minimum of eight preemption sequences without requiring additional software or hardware.

3.12.3. Transit/Bus Priority

3.12.3.1. The controller software must support a minimum of ten bus or other low-priority preemption sequences without requiring additional software or hardware.

3.12.3.2. The controller software must provide a minimum of 4 set of TSP strategy parameters and 16 TSP strategies.

3.12.3.3. The controller software must be compatible with all South Dade Busway intersections.

3.13 Controller Event Logs

3.13.1. The controller software must include an event log buffer capable of storing a minimum of 24 hours of time and date-stamped high resolution events or alarms. Once logged, events remain in the buffer until cleared or the log buffer capacity is exceeded. When capacity is reached, events are over-written by the first-in first-out method.

3.13.2. The controller software must permit users to collect events and alarm data remotely and export data to USB in a readable format.

3.14 Communication

3.14.1. The controller software must support communication via Ethernet and Serial interfaces. Ethernet interfaces must be able to provide ICMP Echo Replies to ICMP Echo Requests.

3.15 Front Panel Interface

3.15.1. The controller software must provide a menu-based front panel user interface using the controller front panel display and keypad controls.

3.15.2. The front panel interface must permit authorized users to read, enter, modify, and save edit all programmable controller settings.

3.15.3. The controller software must provide different users security levels thought front panel

3.15.4. The front panel must return to a user-definable state after a user-specified amount of time with no front panel activity.

3.15.5. The controller software must permit authorized users to update the database via laptop computer or thought the communication system.

4. Peer-to-Peer Operation

4.1 General

4.1.1. The controller software must provide Peer-to-Peer operation that allows controllers to share information over an Ethernet network between intersections. The controller software must allow programming of peer-to-peer functions, including the identification of peer controllers by IP address.

4.1.2. The provided data must include phases, plan, flash, TSP and signal status, detectors, logic gates, and input/output pins.

5. Central System Compatibility

5.1 Miami-Dade County Central Traffic Management System Software

5.1.1. The controller software must be compatible with Miami-Dade Central Traffic Management System Software.

5.1.2. The controller software must support adaptive traffic control and connected vehicle applications.

5.2 NTCIP-Compliant Central Management Systems

5.2.1. The controller software must be NTCIP compliant in accordance with these specifications.

MIAMI-DADE TRAFFIC SIGNAL CONTROLLER NTCIP REQUIREMENTS
APPENDIX A

PART 1 GENERAL

This document defines National Transportation Communications for ITS Protocol (NTCIP) requirements for actuated traffic signal controller (ASC) units. Please note that if the group is mandatory, then all objects within that group listed as mandatory shall be supported. If the ASC does not support the functionality associated with a specific object or group of objects, yet still meets Florida Department of Transportation (FDOT) and Miami-Dade County specifications, then the device must respond with a noSuchName error response when requests are made for those objects.

PART 2 NTCIP REQUIREMENTS

Table 1: Abbreviations used in this document

Abbreviation	Description
M	*Mandatory*
O	*Optional*
D	*Deprecated*

Table 2: Supplemental Traffic Signal Controller NTCIP Requirements Table

Conformance Group / Object Name	Reference	Miami-Dade Requirement	Additional Notes and Requirements
Phase Parameters	**NTCIP 1202v02.19 2.2**	**M Group**	
maxPhases	2.2.1	M	
phaseTable	2.2.2	M	
phaseEntry		M	
phaseNumber	2.2.2.1	M	
phaseWalk	2.2.2.2	M	
phasePedestrianClear	2.2.2.3	M	
phaseMinimumGreen	2.2.2.4	M	
phasePassage	2.2.2.5	M	
phaseMaximum1	2.2.2.6	M	
phaseMaximum2	2.2.2.7	M	
phaseYellowChange	2.2.2.8	M	
phaseRedClear	2.2.2.9	M	

Conformance Group / Object Name	Reference	Miami-Dade Requirement	Additional Notes and Requirements
phaseRedRevert	2.2.2.10	M	
phaseAddedInitial	2.2.2.11	M	
phaseMaximumInitial	2.2.2.12	M	
phaseTimeBeforeReduction	2.2.2.13	M	
phaseCarsBeforeReduction	2.2.2.14	M	
phaseTimeToReduce	2.2.2.15	M	
phaseReduceBy	2.2.2.16	M	
phaseMinimumGap	2.2.2.17	M	
phaseDynamicMaxLimit	2.2.2.18	M	
phaseDynamicMaxStep	2.2.2.19	M	
phaseStartup	2.2.2.20	M	
phaseOptions	2.2.2.21	M	
phaseRing	2.2.2.22	M	
phaseConcurrency	2.2.2.23	M	
maxPhaseGroups	2.2.3	M	
phaseStatusGroupTable	2.2.4	M	
phaseStatusGroupEntry		M	
phaseStatusGroupNumber	2.2.4.1	M	
phaseStatusGroupReds	2.2.4.2	M	
phaseStatusGroupYellows	2.2.4.3	M	
phaseStatusGroupGreens	2.2.4.4	M	
phaseStatusGroupDontWalks	2.2.4.5	M	
phaseStatusGroupPedClears	2.2.4.6	M	
phaseStatusGroupWalks	2.2.4.7	M	
phaseStatusGroupVehCalls	2.2.4.8	M	
phaseStatusGroupPedCalls	2.2.4.9	M	
phaseStatusGroupPhaseOns	2.2.4.10	M	
phaseStatusGroupPhaseNexts	2.2.4.11	M	
phaseControlGroupTable	2.2.5	M	
phaseControlGroupEntry		M	
phaseControlGroupNumber	2.2.5.1	M	
phaseControlGroupPhaseOmit	2.2.5.2	M	
phaseControlGroupPedOmit	2.2.5.3	M	
phaseControlGroupHold	2.2.5.4	M	

Conformance Group / Object Name	Reference	Miami-Dade Requirement	Additional Notes and Requirements
phaseControlGroupForceOff	2.2.5.5	M	
phaseControlGroupVehCall	2.2.5.6	M	
phaseControlGroupPedCall	2.2.5.7	M	
Detector Parameters	**2.3**	**M Group**	
maxVehicleDetectors	2.3.1	M	
vehicleDetectorTable	2.3.2	M	
vehicleDetectorEntry		M	
vehicleDetectorNumber	2.3.2.1	M	
vehicleDetectorOptions	2.3.2.2	M	
vehicleDetectorCallPhase	2.3.2.3	M	
vehicleDetectorSwitchPhase	2.3.2.4	M	
vehicleDetectorDelay	2.3.2.5	M	
vehicleDetectorExtend	2.3.2.6	M	
vehicleDetectorQueueLimit	2.3.2.7	M	
vehicleDetectorNoActivity	2.3.2.8	M	
vehicleDetectorMaxPresence	2.3.2.9	M	
vehicleDetectorErraticCounts	2.3.2.10	M	
vehicleDetectorFailTime	2.3.2.11	M	
vehicleDetectorAlarms	2.3.2.12	M	
vehicleDetectorReportedAlarms	2.3.2.13	M	
vehicleDetectorReset	2.3.2.14	M	
maxVehicleDetectorStatusGroups	2.2.3	M	
vehicleDetectorStatusGroupTable	2.3.4	M	
vehicleDetectorStatusGroupEntry		M	
vehicleDetectorStatusGroupNumber	2.3.4.1	M	
vehicleDetectorStatusGroupActive	2.3.4.2	M	
vehicleDetectorStatusGroupAlarms	2.3.4.3	M	
Volume/Occupancy Report	2.3.5	M	
volumeOccupancySequence	2.3.5.1	M	
volumeOccupancyPeriod	2.3.5.2	M	
activeVolumeOccupancyDetectors	2.3.5.3	M	
volumeOccupancyTable	2.3.5.4	M	
volumeOccupancyEntry		M	
detectorVolume	2.3.5.4.1	M	

Conformance Group / Object Name	Reference	Miami-Dade Requirement	Additional Notes and Requirements
detectorOccupancy	2.3.5.4.2	M	
maxPedestrianDetectors	2.3.6	M	
pedestrianDetectorTable	2.3.7	M	
pedestrianDetectorEntry		M	
pedestrianDetectorNumber	2.3.7.1	M	
pedestrianDetectorCallPhase	2.3.7.2	M	
pedestrianDetectorNoActivity	2.3.7.3	M	
pedestrianDetectorMaxPresence	2.3.7.4	M	
pedestrianDetectorErraticCounts	2.3.7.5	M	
pedestrianDetectorAlarms	2.3.7.6	M	
Unit Parameters	**2.4**	**M Group**	
unitStartUpFlash	2.4.1	M	
unitAutoPedestrianClear	2.4.2	M	
unitBackupTime	2.4.3	M	
unitRedRevert	2.4.4	M	
unitControlStatus	2.4.5	M	
unitFlashStatus	2.4.6	M	
unitAlarmStatus2	2.4.7	M	
unitAlarmStatus1	2.4.8	M	
shortAlarmStatus	2.4.9	M	
unitControl	2.4.10	M	
maxAlarmGroups	2.4.11	M	
alarmGroupTable	2.4.12	M	
alarmGroupEntry		M	
alarmGroupNumber	2.4.12.1	M	
alarmGroupState	2.4.12.2	M	
maxSpecialFunctionOutputs	2.4.13	M	
specialFunctionOutputTable	2.4.14	M	
specialFunctionOutputEntry		M	
specialFunctionOutputNumber	2.4.14.1	M	
specialFunctionOutputState	2.4.14.1	D	
specialFunctionOutputControl	2.4.14.3	M	
specialFunctionOutputStatus	2.4.14.4	M	
Coordination Parameters	**2.5**	**M Group**	

Conformance Group / Object Name	Reference	Miami-Dade Requirement	Additional Notes and Requirements
coordOperationalMode	2.5.1	M	
coordCorrectionMode	2.5.2	M	
coordMaximumMode	2.5.3	M	
other(1)		O	
maximum1(2)		M	
maximum2(3)		M	
maxinhibit(4)		M	
coordForceMode	2.5.4	M	
other(1)		M	
floating(2)		M	
fixed(3)		M	
maxPatterns	2.5.5	M	
patternTableType	2.5.6	M	
other(1)		O	
patterns(2)		M	
offset3(3)		O	
offset5(4)		O	
patternTable	2.5.7	M	
patternEntry		M	
patternNumber	2.5.7.1	M	
patternCycleTime	2.5.7.2	M	
patternOffsetTime	2.5.7.3	M	
patternSplitNumber	2.5.7.4	M	
patternSequenceNumber	2.5.7.5	M	
maxSplits	2.5.8	M	
splitTable	2.5.9	M	
splitEntry		M	
splitNumber	2.5.9.1	M	
splitPhase	2.5.9.2	M	
splitTime	2.5.9.3	M	
splitMode	2.5.9.4	M	
other(1)		O	
none(2)		M	
minimumVehicleRecall(3)		M	

Conformance Group / Object Name	Reference	Miami-Dade Requirement	Additional Notes and Requirements
maximumVehicleRecall(4)		M	
pedestrianRecall(5)		M	
maximumVehicleAndPedestrainRecall(6)		M	
phaseOmitted(7)		M	
splitCoordPhase	2.5.9.5	M	
coordPatternStatus	2.5.10	M	
localFreeStatus	2.5.11	M	
coordCycleStatus	2.5.12	M	
coordSyncStatus	2.5.13	M	
systemPatternControl	2.5.14	M	
systemSyncControl	2.5.15	M	
Time Base Parameters	**2.6**	**M Group**	
timebaseAscPatternSync	2.6.1	M	
maxTimebaseAscActions	2.6.2	M	
timebaseAscActionTable	2.6.3	M	
timebaseAscActionEntry		M	
timebaseAscActionNumber	2.6.3.1	M	
timebaseAscPattern	2.6.3.2	M	
timebaseAscAuxiliaryFunction	2.6.3.3	M	
timebaseAscSpecialFunction	2.6.3.4	M	
timebaseAscActionStatus	2.6.4	M	
Preempt Parameters	**2.7**	**M Group**	
maxPreempts	2.7.1	M	
preemptTable	2.7.2	M	
preemptEntry		M	
preemptNumber	2.7.2.1	M	
preemptControl	2.7.2.2	M	
preemptLink	2.7.2.3	M	
preemptDelay	2.7.2.4	M	
preemptMinimumDuration	2.7.2.5	M	
preemptMinimumGreen	2.7.2.6	M	
preemptMinimumWalk	2.7.2.7	M	
preemptEnterPedClear	2.7.2.8	M	
preemptTrackGreen	2.7.2.9	M	

Conformance Group / Object Name	Reference	Miami-Dade Requirement	Additional Notes and Requirements
preemptDwellGreen	2.7.2.10	M	
preemptMaximumPresence	2.7.2.11	M	
preemptTrackPhase	2.7.2.12	M	
preemptDwellPhase	2.7.2.13	M	
preemptDwellPed	2.7.2.14	M	
preemptExitPhase	2.7.2.15	M	
preemptState	2.7.2.16	M	
preemptTrackOverlap	2.7.2.17	M	
preemptDwellOverlap	2.7.2.18	M	
preemptCyclingPhase	2.7.2.19	M	
preemptCyclingPed	2.7.2.20	M	
preemptCyclingOverlap	2.7.2.21	M	
preemptEnterYellowChange	2.7.2.22	M	
preemptEnterRedClear	2.7.2.23	M	
preemptTrackYellowChange	2.7.2.24	M	
preemptTrackRedClear	2.7.2.25	M	
preemptControlTable	2.7.3	M	
preemptControlEntry		M	
preemptControlNumber	2.7.3.1	M	
preemptControlState	2.7.3.2	M	
Ring Parameters	**2.8**	**M Group**	
maxRings	2.8.1	M	
maxSequences	2.8.2	M	
sequenceTable	2.8.3	M	
sequenceEntry		M	
sequenceNumber	2.8.3.1	M	
sequenceRingNumber	2.8.3.2	M	
sequenceData	2.8.3.3	M	
maxRingControlGroups	2.8.4	M	
ringControlGroupTable	2.8.5	M	
ringControlGroupEntry		M	
ringControlGroupNumber	2.8.5.1	M	
ringControlGroupStopTime	2.8.5.2	M	
ringControlGroupForceOff	2.8.5.3	M	

Conformance Group / Object Name	Reference	Miami-Dade Requirement	Additional Notes and Requirements
ringControlGroupMax2	2.8.5.4	M	
ringControlGroupMaxInhibit	2.8.5.5	M	
ringControlGroupPedRecycle	2.8.5.6	M	
ringControlGroupRedRest	2.8.5.7	M	
ringControlGroupOmitRedClear	2.8.5.8	M	
ringStatusTable	2.8.6	M	
ringStatusEntry		M	
ringStatus	2.8.6.1	M	
Channel Parameters	**2.9**	**M Group**	
maxChannels	2.9.1	M	
channelTable	2.9.2	M	
channelEntry		M	
channelNumber	2.9.2.1	M	
channelControlSource	2.9.2.2	M	
channelControlType	2.9.2.3	M	
other(1)		O	
phaseVehicle(2)		M	
phasePedestrian(3)		M	
overlap(4)		M	
channelFlash	2.9.2.4	M	
channelDim	2.9.2.5	M	
maxChannelStatusGroups	2.9.3	M	
channelStatusGroupTable	2.9.4	M	
channelStatusGroupEntry		M	
channelStatusGroupNumber	2.9.4.1	M	
channelStatusGroupReds	2.9.4.2	M	
channelStatusGroupYellows	2.9.4.3	M	
channelStatusGroupGreens	2.9.4.4	M	
Overlap Parameters	**2.10**	**M Group**	
maxOverlaps	2.10.1	M	
overlapTable	2.10.2	M	
overlapEntry		M	
overlapNumber	2.10.2.1	M	
overlapType	2.10.2.2	M	

Conformance Group / Object Name	Reference	Miami-Dade Requirement	Additional Notes and Requirements
overlapIncludedPhases	2.10.2.3	M	
overlapModifierPhases	2.10.2.4	M	
overlapTrailGreen	2.10.2.5	M	
overlapTrailYellow	2.10.2.6	M	
overlapTrailRed	2.10.2.7	M	
maxOverlapStatusGroups	2.10.3	M	
overlapStatusGroupTable	2.10.4	M	
overlapStatusGroupEntry		M	
overlapStatusGroupNumber	2.10.4.1	M	
overlapStatusGroupReds	2.10.4.2	M	
overlapStatusGroupYellows	2.10.4.3	M	
overlapStatusGroupGreens	2.10.4.4	M	
TS2 Port 1 Parameters	**2.11**	**O Group**	**Note: None of the objects under 2.11 are required.**
maxPort1Addresses	2.11.1	M	
port1Table	2.11.2	M	
port1Entry		M	
port1Number	2.11.2.1	M	
port1DevicePresent	2.11.2.2	M	
port1Frame40Enable	2.11.2.3	M	
port1Status	2.11.2.4	M	
port1FaultFrame	2.11.2.5	M	
ASC Block Objects	**2.12**	**O Group**	**Note: None of the objects under 2.12 are required.**
ascBlockGetControl	2.12.1	M	
ascBlockData	2.12.2	M	
ascBlockErrorStatus	2.12.3	M	
GLOBAL CONFIGURATION NODE	**NTCIP 1201v03.15r 2.2**	**M Group**	
globalSetIDParmeter	2.2.1	M	
globalMaxModules	2.2.2	M	
globalModuleTable	2.2.3	M	
moduleTableEntry		M	

Conformance Group / Object Name	Reference	Miami-Dade Requirement	Additional Notes and Requirements
moduleNumber	2.2.3.1	M	
moduleDeviceNode	2.2.3.2	M	
moduleMake	2.2.3.3	M	
moduleModel	2.2.3.4	M	
moduleVersion	2.2.3.5	M	
moduleType	2.2.3.6	M	
controller-baseStandards	2.2.4	M	
GLOBAL DATABASE MANAGEMENT NODE	**2.3**	**M Group**	
dbCreateTransaction	2.3.1	M	
dbVerifyStatus	2.3.6	M	
dbVerifyError	2.3.7	M	
GLOBAL TIME MANAGEMENT NODE	**2.4**	**M Group**	
globalTme	2.4.1	M	
globalDayLightSaving	2.4.2	M	
timebase	2.4.3	M	
maxTimeBaseScheduleEntries	2.4.3.1	M	
timeBaseScheduleTable	2.4.3.2	M	
timeBaseScheduleEntry		M	
timeBaseScheduleNumber	2.4.3.2.1	M	
timeBaseScheduleMonth	2.4.3.2.2	M	
timeBaseScheduleDay	2.4.3.2.3	M	
timeBaseScheduleDate	2.4.3.2.4	M	
timeBaseScheduleDayPlan	2.4.3.2.5	M	
timeBaseScheduleTable-status	2.4.3.3	M	
maxDayPlans	2.4.4.1	M	
maxDayPlanEvents	2.4.4.2	M	
timeBaseDayPlanTable	2.4.4.3	M	
timeBaseDayPlanEntry		M	
dayPlanNumber	2.4.4.3.1	M	
dayPlanEventNumber	2.4.4.3.2	M	
dayPlanHour	2.4.4.3.3	M	
dayPlanMinute	2.4.4.3.4	M	
dayPlanActionNumberOID	2.4.4.3.5	M	

Conformance Group / Object Name	Reference	Miami-Dade Requirement	Additional Notes and Requirements
dayPlanStatus	2.4.4.4	M	
globalTimeDifferential	2.4.5	D	
controller-standardTimeZone	2.4.6	M	
controller-localTime	2.4.7	M	
SNMP Group	**RFC1213**	**M Group**	
snmpInPkts	snmp.1	M	
snmpOutPkts	snmp.2	M	
snmpInBadVersions	snmp.3	M	
snmpInBadCommunityNames	snmp.4	M	
snmpInBadCommunityUses	snmp.5	M	
snmpInASNParseErrs	snmp.6	M	
snmpInTooBigs	snmp.8	M	
snmpInNoSuchNames	snmp.9	M	
snmpInBadValues	snmp.10	M	
snmpInReadOnlys	snmp.11	M	
snmpInGenErrs	snmp.12	M	
snmpInTotalReqVars	snmp.13	M	
snmpInTotalSetVars	snmp.14	M	
snmpInGetRequests	snmp.15	M	
snmpInGetNexts	snmp.16	M	
snmpInSetRequests	snmp.17	M	
snmpInGetResponses	snmp.18	M	
snmpInTraps	snmp.19	O	
snmpOutTooBigs	snmp.20	O	
snmpOutNoSuchNames	snmp.21	M	
snmpOutBadValues	snmp.22	M	
snmpOutGenErrs	snmp.24	M	
snmpOutGetRequests	snmp.25	M	
snmpOutGetNexts	snmp.26	M	
snmpOutSetRequests	snmp.27	M	
snmpOutGetResponses	snmp.28	M	
snmpOutTraps	snmp.29	O	
snmpEnableAuthenTraps	snmp.30	O	
System Group	**RFC1213**	**M Group**	

Conformance Group / Object Name	Reference	Miami-Dade Requirement	Additional Notes and Requirements
sysDescr	system 1	M	
sysObjectID	system 2	M	
sysUpTime	system 3	M	
sysContact	system 4	M	
sysName	system 5	M	
sysLocation	system 6	M	
sysServices	system 7	M	
STMP Group	**NTCIP 1103-A.5**	**M Group**	
Dynamic Object Definition	A.5.1	M	
dynObjDef		M	
dynObjEntry		M	
dynObjNumber		M	
dynObjIndex		M	
dynObjVariable		M	
Dynamic Object Configuration	A.5.3	M	
dynObjConfigTable		M	
dynObjConfigEntry		M	
dynObjConfigOwner		M	
dynObjConfigStatus		M	
STMP Statistics	A.5.4	M	
stmp-InPkts	.1	M	
stmp-OutPkts	.2	M	
stmp-InParseErrs	.6	M	
stmp-InTooBigs	.8	M	
stmp-InNoSuchNames	.9	M	
stmp-InBadValues	.10	M	
stmp-InReadOnlys	.11	M	
stmp-InGenErrs	.12	M	
stmp-InGetRequests	.15	M	
stmp-InGetNexts	.16	M	
stmp-InSetRequests	.17	M	
stmp-InGetResponses	.18	M	
stmp-OutTooBigs	.20	M	

Conformance Group / Object Name	Reference	Miami-Dade Requirement	Additional Notes and Requirements
stmp-OutNoSuchNames	.21	M	
stmp-OutBadValues	.22	M	
stmp-OutReadOnly	.23	M	
stmp-OutGenError	.24	M	
stmp-OutGetRequests	.25	M	
stmp-OutGetNexts	.26	M	
stmp-OutSetRequests	.27	M	
stmp-OutGetResponses	.28	M	
stmp-OutTrapResponses	.29	M	
stmp-InSetRequestsNoReply	.31	M	
stmp-InSetResponses	.32	M	
stmp-InErrorResponses	.33	M	
stmp-OutSetRequestsNoReply	.34	M	
stmp-OutSetResponses	.35	M	
stmp-OutErrorResponses	.35	M	
STMP Configuration	A.5.5	M	
dynamicObjectPersistence	.1	M	
dynamicObjectTable-ConfigID	.2	M	