# SPAT MIB Support Document

July 23, 2012

*Authors:* Dustin DeVoe (Econolite)

## Overview

This document describes the Signal Phase and Timing (SPaT) support in ASC/3 for Battelle research group. The ASC/3 supports NTCIP 1202 with 16 phases and 16 overlaps.

## Scope of Operation

To limit the inherent complexities of the controller during the initial concept phase, we have agreed to support only NTCIP 1202 defined controller programming.

|  |  |  |
| --- | --- | --- |
| **General NTCIP Operation** | **Implemented** | **Basic Testing Complete** |
| Fixed Time Vehicle Only | Yes | Yes |
| Fixed Time with Ped | Yes | Yes |
| Actuated Free | Yes | Yes |
| Actuated Free with Ped | Yes | Yes |
| Coordinated | Yes | Yes |
| Floating Force-Off/Fixed | Yes | Yes |
| Transition Dwell/Smooth/Add | Yes | Yes |
| Pattern Recalls Min/Max/Ped | Yes | Yes |
| Overlaps | Partial | Partial |
| Included | Yes | Yes |
| Lag Green, Yellow, Red | No | No |
| Modifier (aka Not Included) | No | No |

## SNMP Objects For SPaT MIB

Please refer to the Battelle SPaT MIB documentation for more details.

Objects

1.3.6.1.4.1.1206.3.47.1 spatTimeToChangeTable NODE (0)

1.3.6.1.4.1.1206.3.47.1.1 spatTimeToChangeEntry NODE (1)

1.3.6.1.4.1.1206.3.47.1.1.1 spatTimeToChangePhaseNumber LEAF INTEGER

1.3.6.1.4.1.1206.3.47.1.1.2 spatVehMinTimeToChange LEAF INTEGER

1.3.6.1.4.1.1206.3.47.1.1.3 spatVehMaxTimeToChange LEAF INTEGER

1.3.6.1.4.1.1206.3.47.1.1.4 spatPedMinTimeToChange LEAF INTEGER

1.3.6.1.4.1.1206.3.47.1.1.5 spatPedMaxTimeToChange LEAF INTEGER

1.3.6.1.4.1.1206.3.47.2 spatOvlpTimeToChangeTable NODE (0)

1.3.6.1.4.1.1206.3.47.2.1 spatOvlpTimeToChangeEntry NODE (1)

1.3.6.1.4.1.1206.3.47.2.1.1 spatTimeToChangeOvlpNumber LEAF INTEGER

1.3.6.1.4.1.1206.3.47.2.1.2 spatOvlpMinTimeToChange LEAF INTEGER

1.3.6.1.4.1.1206.3.47.2.1.3 spatOvlpMaxTimeToChange LEAF INTEGER

1.3.6.1.4.1.1206.3.47.3 spatDiscontinuousChangeFlag LEAF INTEGER

1.3.6.1.4.1.1206.3.47.4 spatFlashingOutputPhaseStatus LEAF INTEGER

1.3.6.1.4.1.1206.3.47.5 spatFlashingOutputOverlapStatus LEAF INTEGER

1.3.6.1.4.1.1206.3.47.6 spatIntersectionStatus LEAF INTEGER

## NTCIP-Based 100ms Broadcast Interface

Byte-Map Structure of the Broadcast Message, Version #2.

*byte 0: DynObj13 response byte (0xcd)*

*byte 1: number of phase/overlap blocks below (16)*

*bytes 2-14:*

*0x01 (phase#) (1 byte)*

*VehMinTimeToChange.1 (2 bytes)*

*VehMaxTimeToChange.1 (2 bytes)*

*PedMinTimeToChange.1 (2 bytes)*

*PedMaxTimeToChange.1 (2 bytes)*

*OvlpMinTimeToChange.1 (2 bytes)*

*OvlpMaxTimeToChange.1 (2 bytes)*

*...*

*< repeat for each phase and overlap – bytes 15-196 >*

*...*

*bytes 197-209:*

*0x10 (phase#) (1 byte)*

*VehMinTimeToChange.16 (2 bytes)*

*VehMaxTimeToChange.16 (2 bytes)*

*PedMinTimeToChange.16 (2 bytes)*

*PedMaxTimeToChange .16 (2 bytes)*

*OvlpMinTimeToChange .16 (2 bytes)*

*OvlpMaxTimeToChange .16 (2 bytes)*

*bytes 210-215:*

*PhaseStatusReds (2 bytes bit-mapped for phases 1-16)*

*PhaseStatusYellows (2 bytes bit-mapped for phases 1-16)*

*PhaseStatusGreens (2 bytes bit-mapped for phases 1-16)*

*bytes 216-221:*

*PhaseStatusDontWalks (2 bytes bit-mapped for phases 1-16)*

*PhaseStatusPedClears (2 bytes bit-mapped for phases 1-16)*

*PhaseStatusWalks (2 bytes bit-mapped for phases 1-16)*

*bytes 222-227:*

*OverlapStatusReds (2 bytes bit-mapped for overlaps 1-16)*

*OverlapStatusYellows (2 bytes bit-mapped for overlaps 1-16)*

*OverlapStatusGreens (2 bytes bit-mapped for overlaps 1-16)*

*bytes 228-229:*

*FlashingOutputPhaseStatus (2 bytes bit-mapped for phases 1-16)*

*bytes 230-231:*

*FlashingOutputOverlapStatus (2 bytes bit-mapped for overlaps 1-16)*

*byte 232:*

*IntersectionStatus (1 byte) (bit-coded byte)*

*Byte 233:*

*TimebaseAscActionStatus (1 byte) (current action plan)*

*byte 234:*

*DiscontinuousChangeFlag (1 byte) (upper 5 bits are msg version #2, 0b00010XXX)*

*byte 235:*

*MessageSequenceCounter (1 byte) (lower byte of up-time deciseconds)*

*Byte 236-238:*

*SystemSeconds (3 byte) (sys-clock seconds in day 0-84600)*

*Byte 239-240:*

*SystemMilliSeconds (2 byte) (sys-clock milliseconds 0-999)*

*Byte 241-242:*

*PedestrianDirectCallStatus (2 byte) (bit-mapped phases 1-16)*

*Byte 243-244:*

*PedestrianLatchedCallStatus (2 byte) (bit-mapped phases 1-16)*

## Flashing Output Status Words

Two bit-mapped words (2 bytes) indicate which phases and overlaps are currently flashing. These, used in conjunction with the existing Green, Yellow and Red status bytes will provide enough information to determine a flashing color on a movement. For example, a flashing green phase or a FYA (Flashing Yellow Arrow) overlap.

These words will only be valid during normal or programmed flash operation. If the cabinet has switched to relay-flash, the controller will not be able to provide an accurate representation of these output states. If the controller can detect such a condition, the expected status information has not yet been defined.

## Intersection Status Byte

|  |  |  |
| --- | --- | --- |
| **Bit #** | **Feature** | **Description of Bit (1 = SET)** |
| 0 | Manual Control Enable Active | Set if Manual Control Enable operation has been activated. |
| 1 | Stop Time (all rings) Active | Set only if the controller has been commanded to stop timing on ALL RINGS. |
| 2 | Fault Flash Active | Set if, for any reason, the controller has dropped CVM due to a Failure condition. Failure conditions include MMU faults such as conflict or short yellow, Preempt faults such as Interlock or Gate Down failures, communications faults such as SDLC (TS-2 type 1) problems. |
| 3 | Preempt Active | Set if ANY of the preempt runs is active; it will not be set if there is a call for a preempt run but that run has not been activated for whatever reason. |
| 4 | TSP Active | Set if ANY of the TSP runs is active; it will not be set if there is a call for a TSP run but that run has not been activated for whatever reason. |
| 5 | Coordination Active (IN STEP) | Set if the controller is currently running an IN-STEP coordination pattern. |
| 6 | Coordination-in-Transition (DWELL, ADD, SUBTRACT) | Set whenever the controller is trying to get a coordination pattern IN-STEP. The controller may be using one of three methods- DWELL, ADD, SUBTRACT. |
| 7 | Programmed Flash Active | Set if the controller is in flash other than fault flashes. Example of programmed flash include scheduled, Preempt, remote, or auto flash. |