# SPAT MIB Support Document

July 23, 2012

*Authors:* Dustin DeVoe

**REVISION HISTORY**

|  |  |  |
| --- | --- | --- |
| Date | Description | Protocol Version |
| January 25, 2012 | Original | 1 |
| February 9, 2012 | added sections Intersection Status Byte, Flashing Output Status Bytes, IP Address, UDP Port, Broadcast Message - example | 1 |
| February 14, 2012 | Corrected flashing phase and overlap status objects, removed broadcast example. Added logging section. | 1 |
| February 21, 2012 | Added SNMP Objects and new Message Sequence Counter | 1 |
| February 22, 2012 | Updated SNMP Objects list per coding implementation | 1 |
| March 7,  2012 | Added description for logging on the ASC/3 firmware. Fixed SNMP Objects table. | 1 |
| April 5, 2012 | Added current Action Plan to the SPAT broadcast. | 1 |
| June 19, 2012 | Added new timestamp with system seconds and milliseconds of day | 1 |
| July 23, 2012 | Add Pedestrian calling information | 2 |

## 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

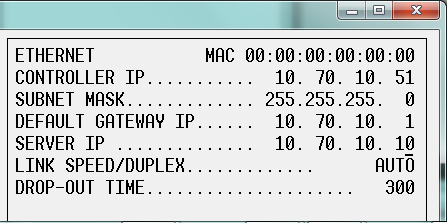
|  |  |  |
| --- | --- | --- |
| **Bit #** | **Feature** | **Description of Bit (1 = SET)** |
| 0 | Manual Control Enable Active | Set if Manual Control Enable operation has been activated. |
| 1 | Stop Time (all rings) Active | Set only if the controller has been commanded to stop timing on ALL RINGS. |
| 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. |

## Broadcast Configuration

## IP Address

To set the controller to send the push packet to a particular IP address, use the following instructions:

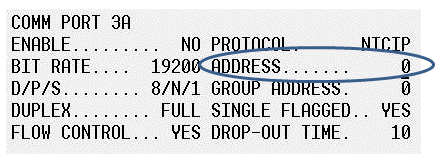
1. Press Main Menu (MM) on the Front Panel
2. Press 1 (Configuration)
3. Press 5 (Communication)
4. Press 1 (Ethernet)
5. Move cursor down to the row for **SERVER IP** and specify your ***Destination IP*** as seen below



1. Restart the application for the push packet thread to take the new settings

## UDP Port

The Destination Port for the UDP packet may be selected in MM-1-5-3: Communications Port 3A Address as circled in the screen shot below. As long as the value is GREATER THAN 24, the controller will attempt to open a socket on that port number for the SPaT broadcast.



## Enabling SPaT Operation

In order to enable the SPaT objects in the ASC/3 and enable the push packet (100ms SPaT MIB broadcast), the user must send an NTCIP SET asc3ViiMessageEnable to a value of 2 or 6. This signifies the Battelle message format.

|  |  |
| --- | --- |
| **SNMP Set Value** | **Resulting Broadcast Message Description** |
| 2 | Original SPAT proposal as of June 2012 |
| 6 | Original SPAT plus additional Pedestrian Information |

In the NEMA MIB Tree, the enable is listed as follows:

1.3.6.1.4.1.1206.3.5.2.9.44.1 asc3ViiMessageEnable LEAF INTEGER

The actual OID value sent in an SNMP message would be **1.3.6.1.4.1.1206.3.5.2.9.44.1.0** to address asc3ViiMessageEnable object.

*NOTE:* an extra .0 is added onto the end to signify that this LEAF is not part of a table and does not require an index.

The SET message should look something like this

30 2d 02 01 00 04 06 70 75 62 6c 69 63 a3 20 02 01 00 02 01 00 02 01 00 30 15 30 13 *06 0e 2b 06 01 04 01 89 36 03 05 02 09 2c 01 00* *02 01 02*

Where *06 0e 2b 06 01 04 01 89 36 03 05 02 09 2c 01 00*specifies the OID

06 0e                                                                             OID type (06) , 14 bytes (0e)

2b 06 01 04 01 89 36 03 05 02 09 2c 01 001.3.6.1.4.1.1206.3.5.2.9.44.1.0

And where *02 01 02* is the SET Data

*02 01 02* Integer type (02), 1 byte (01) and data value 2 (02)

Below is the actual MIB Definition of asc3ViiMessageEnable as implemented in the ASC/3:

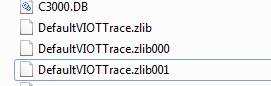
asc3ViiMessageEnable OBJECT-TYPE  
 SYNTAX INTEGER (0..255)  
 ACCESS read-write  
 STATUS mandatory  
 DESCRIPTION  
 "This object is a bit-map of enables related to   
 Connected Vehicle applications and will be used   
 to enable/disable the broadcast of the Signal  
 Phase and Timing message transport.  
   
 Currently only two are implemented:  
 bit 0: Vehicle Infrastructure Integration (VII)   
 UDP Subnet Broadcast on configurable port  
 bit 1: Battelle research for NTCIP broadcast (SPAT)  
 "  
::= { asc3Vii 1 }

## ASC/3 Error Logging

Virtual Controller releases will have the logging turned on by default in order to improve fault reporting. You will see the “RECORDING” message on the I/O panel of the virtual controller, as can be seen in the following screen capture…



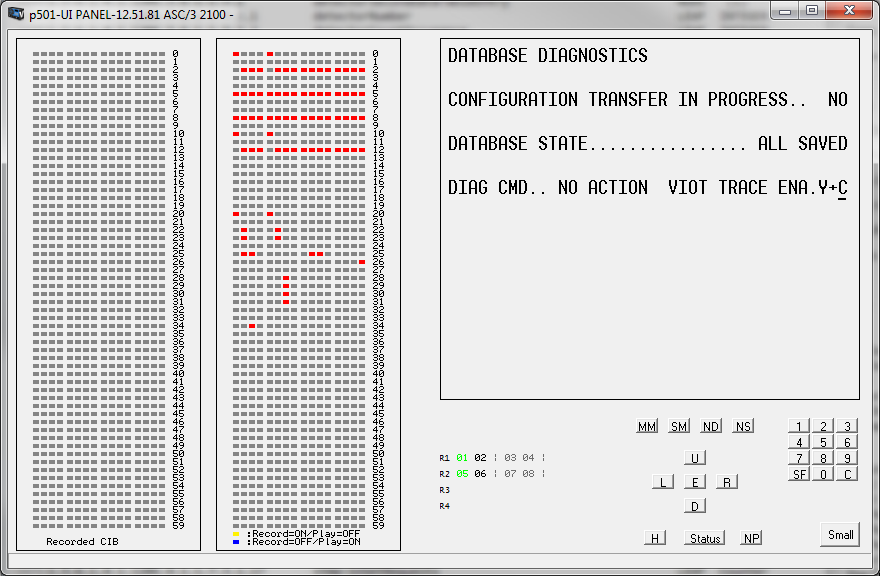
When you find an issue you would like to share with use.  Exit the virtual controller and grab the “DefaultVIOTTrace.zlib” file and include that with a brief description citing clock values.  **Note:** Prior traces in the directory will not be overwritten as they will be renamed with each start of the controller.



*In this screen shot of the file directory, the latest run is circled in blue and the older captures are shown with the orange box around them.*

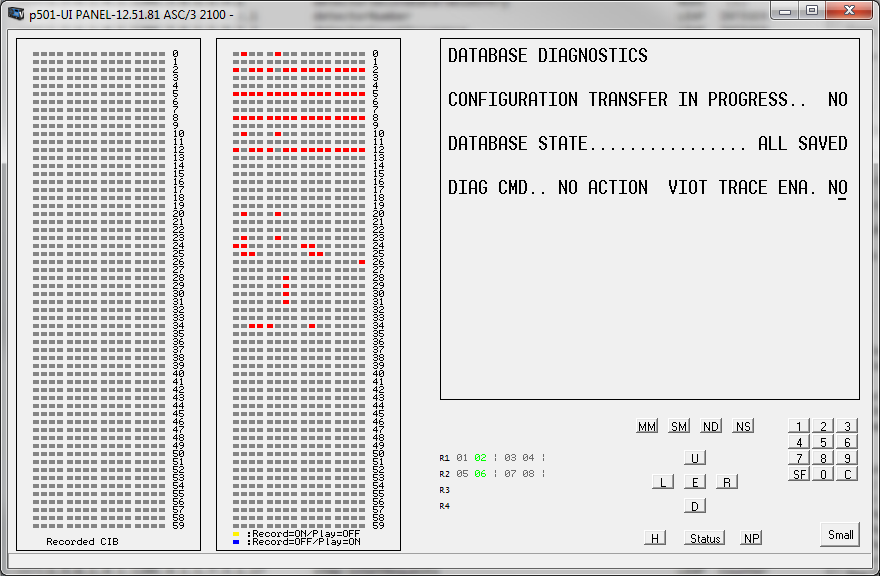
### Enable Logging on ASC/3 Controller (*not tested, use upon request only*)

On the ASC/3 Controller you have to manually enable it through the front panel screens. Following the instructions below a new capture will be created with each power-cycle of the controller.



**Start Capturing**

* On front panel go to Main Menu-9-3-1 (Advanced Database Diagnostics) and hit Special Function 3 times
* Select **Y+C** so the capture log will be reset each time you restart the controller
* Power Cycle ASC/3



**Stop Capturing and Grab Log**

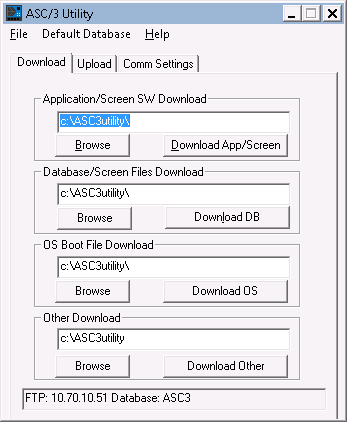
* On front panel go to Main Menu-9-3-1 (Advanced Database Diagnostics) and hit Special Function 3 times
* Select **NO** so the capture log will be reset each time you restart the controller
* Use ASC/3 Utility to Upload the Log
* Grab the CIBCOB.cap file from your selected Upload target directory and share with Dustin

## ASC/3 Controller Upgrade

The ASC/3 has a windows application for updating firmware and uploading/downloading logs and configuration. You will need to retrieve this utility and use the **Comm Settings** tab to set the IP address of your controller on the network and the **Upload** tab to retrieve the database or log files. The **Download** tab contains all the options for sending files to the controller such as a new traffic application or configuration database.

Download Latest Utility from:

[ftp://visitor:ecpi2ecpi@ftp.econolite.com/ASC3\_Utility-3.11.exe](ftp://visitor:ecpi2ecpi@ftp.econolite.com/ASC3_Utility-3.11.exe )



*In this screen shot of the update utility, the button circled in blue will update the controller with the latest traffic application and screens contained in the directory provided. The button circled in orange will download the configuration database to the controller.*

## ASC/3 FTP

There is an FTP client built within the ASC/3 firmware that will allow full access to the file system using a standard ftp client with the credentials Username:”econolite” and Password:”ecpi2ecpi”. It is recommended to be in binary mode if that is optional in your client. Navigate to the Set1 directory for access to the latest database configuration files.

ftp 10.1.15.67

Connected to 10.1.15.67.

220 Tornado-vxWorks (VxWorks5.5.1) FTP server ready

User (10.1.15.67:(none)): econolite

331 Password required

Password: [ecpi2ecpi]

230 User logged in

ftp> cd set1

250 Changed directory to "/set1"

ftp> ls

200 Port set okay

150 Opening BINARY mode data connection

. . . . [end of example] . . . .