INTELLIGENT TRANSPORT SYSTEMS

TRAFFIC SYSTEMS

SPECIFICATION NO. TSI-SP-084

COMMUNICATIONS PROTOCOL FOR SCHOOL ZONE ALERT SYSTEM

Issue: 1.0
Dated: 08/06/2021

DISCLAIMER AND CONDITIONS FOR USE OF THIS SPECIFICATION

This Specification has been prepared by Transport for NSW (referred to herein as TfNSW) for use, insofar as it is applicable, in the State of New South Wales for equipment supplied under an TfNSW order or contract, or under an order or a contract from another party that is required in writing by TfNSW to use this Specification.

The use of this TfNSW Specification other than by those parties stated above and in the manner stated above is not recommended or authorised by TfNSW. Any such use is entirely the decision of the user alone. TfNSW disclaims all responsibilities and liabilities arising whether directly or indirectly from any such use. TfNSW does not warrant that this Specification is error free, nor does TfNSW warrant the suitability, fitness or otherwise of this Specification for any stated or implied purposes expressed or implied in this Specification or other documents. By using this Specification, the user agrees to indemnify TfNSW against the full amount of all expenses, losses, damages and costs (on a full indemnity basis and whether or not incurred by or awarded against TfNSW) which may be suffered by any person or TfNSW in connection with or arising out of the use of this Specification in any manner.

TfNSW is not under any duty to inform you of any errors in or changes to this Specification.

RECORD OF AMENDMENTS

Issue	Summary	Date	Approved by
1.0	Original, derived from content of ITS-SZAS-SD-022 and ITS-SZAS-SD-023	08/06/2021	Director ITS Engineering

C O N T E N T S

Tables

Figures

1 SCOPE

1.1 Purpose

This document describes the communications protocol to be used by TfNSW for control and monitoring of School Zone Alert Signs.

This protocol is a text-based protocol that is designed to operate over an IP network using both TCP and UDP.

1.2 Scope

This document defines the physical means and protocol messages to be used for communication between the School Zone Alert System (SZAS) Central Management Computer (CMC) and the signs.

Where necessary, the functionality required to implement the protocol is described.

1.3 Definitions and Glossary of Terms

The following definitions and abbreviations shall apply:

Term	Definition
4G	4th generation public mobile communication system
Alert Display	Flashing display elements of an SZAS sign; lanterns and annulus.
CMC	Central Management Computer
CRC	Cyclic Redundancy Check
DNS	Domain Name System
ID	Identification or identification code
IP	Internet Protocol
ITS	Intelligent Transportation Systems
LSB	Least Significant Bit
MSB	Most Significant Bit
RTC	Real-time counter
SZAS	School Zone Alert System
TCP	Transmission Control Protocol
TfNSW	Transport for NSW, a New South Wales government agency
UDP	User Datagram Protocol
VPN	Virtual Private Network

Table 1 – Definitions and Glossary of Terms

1.4 TfNSW Specifications and Documents

[1] TSI-SP-083 - School Zone Alert Signs

2 GENERAL

2.1 Network

A private IP-VPN wide area network is used.

2.2 Data Link Layer

A wireless 4G communications data link layer shall be used.

2.3 Network Layer

The Sign Controller shall use IP (Internet Protocol) as its network layer.

2.4 Transport Layer

2.4.1 The TCP shall have a configurable port.

2.4.2 The TCP shall have a default port of 8007.

2.4.3 The TCP shall be used to transfer the management protocol messages and download firmware.

2.4.4 The UDP shall have a configurable port.

2.4.5 The UDP shall have a default port of 10080.

2.4.6 The UDP shall be used to send the trigger message from the CMC to the Sign Controller.

2.5 Operation

The communications link between the CMC and the Sign Controller shall be used to download new configuration data (such as timetable) and device firmware, and to receive notifications of failures, and logs, for auditing purposes.

2.6 Client and Server

2.6.1 In the SZAS architecture, the Sign Controller shall have the role of Client and the CMC that of Server.

2.6.2 The CMC initiates communications, and defined communication sequences, except as noted below.

(a) When the sign first starts up – sign initiates.

(b) In the case of an alarm – sign initiates.

2.7 System Addressing

2.7.1 Each Sign Controller shall be addressed using an IP address.

2.7.2 The CMC shall be given a hostname and DNS used to resolve this to an IP Address.

2.7.3 IP addresses shall be allocated within the reserved private network ranges of the same class.

Note: All Signs and the CMC will be within a single subnet. The addresses might be 10.0.0.0/255.255.0.0. For the sake of illustration, the CMC address 10.50.108.101 is used in this document.

2.8 CMC Address

2.8.1 The CMC shall be allocated a static IP address.

2.8.2 The CMC shall be addressed by the Sign Controller using DNS hostname or an IP address.

2.9 Sign Controller Address

2.9.1 Each sign is manually allocated (a SIM card (the SIM card has a pre-assigned static IP address).

2.9.2 UDP trigger packets shall be addressed to the IP address allocated to the sign.

2.9.3 The IP address identifies a sign and hence its sign ID. Each time the sign connects with an IP address, based on that IP Address a Sign ID is associated by the CMC.

3 APPLICATION LAYER

3.1 Data

3.1.1 The Sign Controller shall use as its application layer a message interpreter listening on each TCP connection.

3.1.2 The Sign Controller request and response messages shall be structured using three (3) character long descriptors.

3.1.3 The messages shall be transferred between nodes using a TCP/IP connection.

3.2 Transparency

The protocol shall be implemented to be independent of the communications media.

3.3 Error Free Communications

The Sign Controller shall rely on the underlying TCP protocol to ensure error free delivery of data.

3.4 General Message Exchange

3.4.1 The CMC shall initiate communications with the Sign Controller via the UDP trigger message.

3.4.2 Upon reception of the UDP trigger message, the Sign Controller shall establish a TCP connection with the CMC.

3.4.3 Upon a successful establishment of the TCP connection, the Sign Controller shall respond with a message containing its sign ID and status word.

3.4.4 The CMC shall then control all further exchanges. Refer to the Sequence diagram below for more details.

3.4.5 All message exchanges between the Sign Controller and the CMC shall be enclosed within a ‘<’ start of message and ‘>’ end of message characters as shown in the sequence diagrams below.

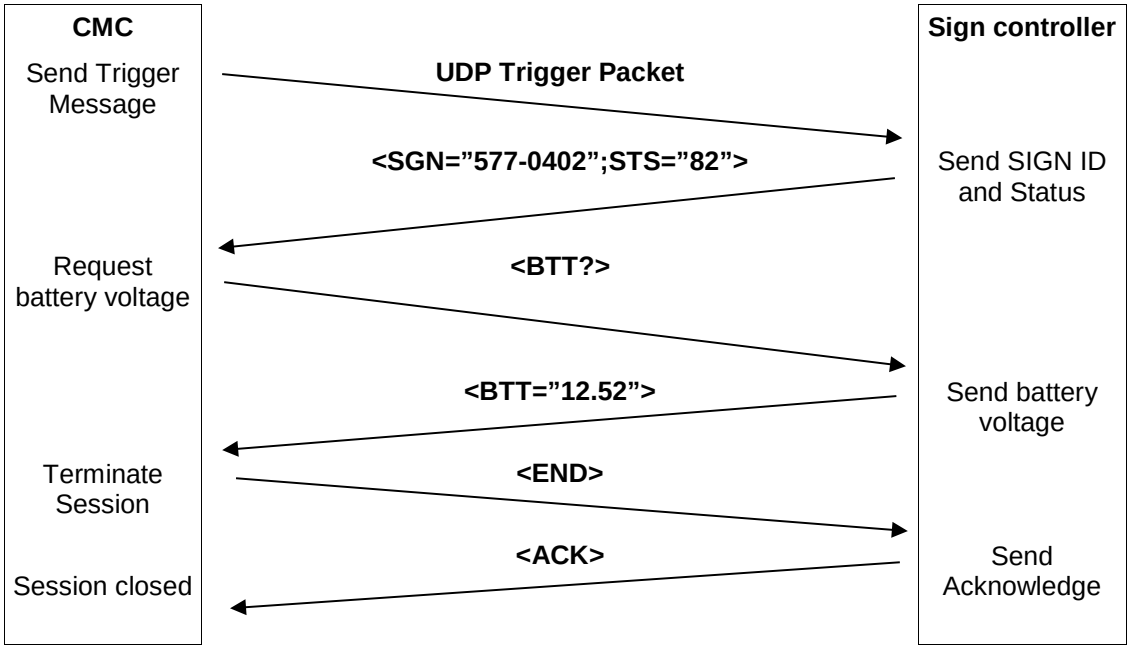

Figure 1 – CMC initiated communications sequence

3.4.6 The Sign Controller shall only initiate communications with the CMC, when it has an alarm or event to report.

3.4.7 The Sign Controller shall send its sign ID and the status word when it has an alarm or event to report.

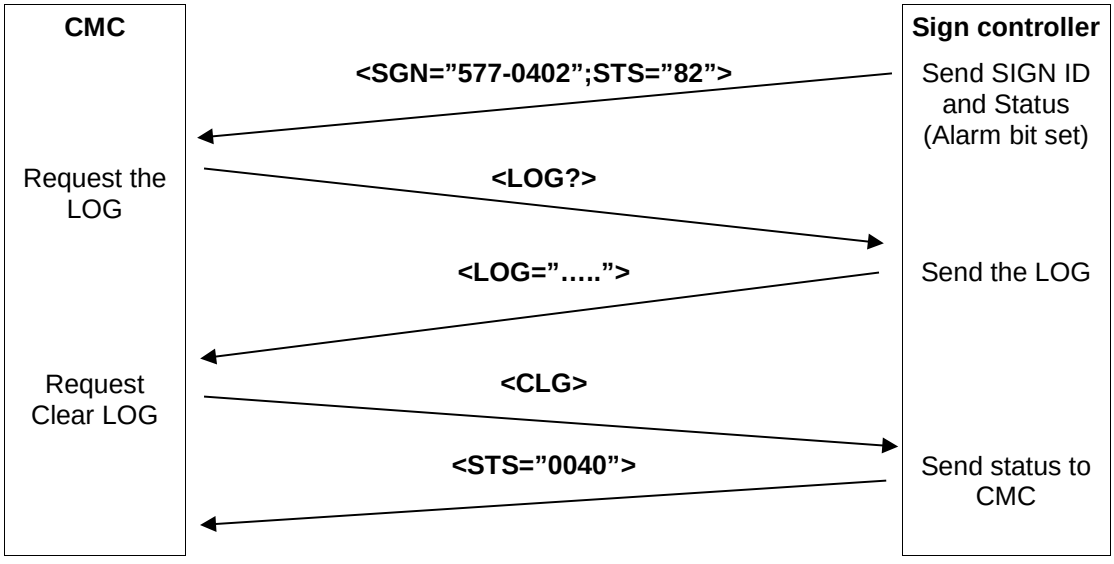

Figure 2 – Sign controller-initiated communications sequence

3.4.8 A response message shall only be sent once all requests within a message have been processed.

3.5 Message Format

3.5.1 The messages shall exchange the defined CMC requests and Sign controller responses, as defined in Appendix A.

3.5.2 Multiple requests and responses shall either be:

a) included in a message separated by a semicolon character ";"; or

b) sent individually

3.5.3 Multiple responses shall be returned in the same order as the requests within a message.

3.5.4 The Sign Controller shall respond with a "REJ" response if it contains white spaces, including TAB, carriage returns and line feeds.

3.5.5 The Sign Controller shall respond with a "REJ" response when any character is received that is not part of the defined message.

3.6 Time Format

3.6.1 SZAS message timing operates in "ticks".

3.6.2 Each tick is one second in duration.

3.6.3 The SZAS CMC clock operates in Epoch time (seconds since midnight 1/1/1970).

4 SZAS MESSAGE FORMATS

The various types of messages and their format are described below.

4.1 Request Message

4.1.1 Each request message shall conform to one of the defined entries in the tables given in Appendix A.

4.1.2 If a message contains several requests, (by using the semicolon separator described in 3.5.2 a), then all the requests therein shall be in the same message class. The message classes are as follows:

a) Configuration messages (refer Appendix A.1);

b) Telemetry messages (refer Appendix A.2);

c) Commands (refer Appendix A.3)

4.1.3 The request message shall either consist of the descriptor tag by itself or followed by an argument.

4.1.4 The argument shall either be a question mark character '?' which implies that the value of that variable is requested, or an equals character '=' which is used to set a variable's value to the data after the equals character.

For example:

CLG	Clear log entries that have been confirmed sent
DTE?	Request current ticks
DTE="SSSSSSSS"	Set current ticks

4.2 Response Message

4.2.1 The Response Message shall be determined according to the request message, as given in Appendix A.

4.2.2 If multiple requests were sent then the response shall be placed in the same order as the requests, except for correctly executed requests that do not require a response.

4.2.3 The response message shall be one of the following:

(a) The tag of the message followed by an equals sign, followed by the requested response parameter/data, for messages with a defined reply.

(b) A single <ACK> response message, for messages that have no defined reply.

(c) The tag of the message followed by a hash character (#), if the request fails such that the sign cannot execute the request.

(d) A reject <REJ> message if the sign received a corrupted or invalid request. In this event, the sign not enact any requests with that message.

5 SIGN CONTROLLER INFORMATION VALUES

The following sections identify and describe the status and alarm variables to be provided by the Sign Controller.

5.1 Status Flags

5.1.1 The 16-bit status word shall provide the status and state of alarms and events of the Sign Controller.

5.1.2 The status word shall be sent to the CMC when requested by the status request command (STS).

5.1.3 The description of the status word is provided in Appendix B.1.

5.2 Display Error byte

5.2.1 The DISPERR bit in the status flags shall indicate whether a fault has occurred in one of the displays.

5.2.2 The specific sign element that is faulty shall be provided in the Display Error byte of the DEF tag.

5.2.3 When one of the bits is set (1), it indicates a fault has occurred in the corresponding display element, as defined in Appendix B.2.

5.3 RTC Synchronization

5.3.1 When the Sign Controller receives a RTC synchronisation request (SYN) from the CMC, the Sign Controller shall request the value of RTC (in ticks) from the CMC by sending a <DTE?> message.

5.3.2 The Sign Controller shall measure the total round trip time (in RTC ticks) (RTT_M) taken from issuing its RTC value request <DTE?> to receiving the RTC value (T_{CMC}) from the CMC. ($RTT_M = T_{CMC} - RTC_{Sign}$)

5.3.3 The Sign Controller shall adjust its RTC only if:

The RTT_M is less than or equal to acceptable round trip time RTT_A that was specified in the RTC synchronisation request; or

The current Sign Controller's RTC is greater than (T_{CMC} + 3/2 × RTT_M); or

The current Sign Controller's RTC is less than (T_{CMC} - ½ × RTT_M).

5.3.4 If the Sign Controller is required to adjust its RTC then the Sign Controller shall set its RTC to: T_{CMC} + (½ × RTT_M)

5.3.5 The Sign Controller shall complete the RTC synchronisation process by responding with the measured round trip time (RTT_M) and the difference between the new Sign Controller RTC and the previous Sign Controller RTC. If the RTC is not adjusted then the difference value is 0.

5.3.6 If the magnitude of the adjustment is greater than 999 ticks, then it shall be reported as 999 or -999 ticks, depending upon the sign of the adjustment.

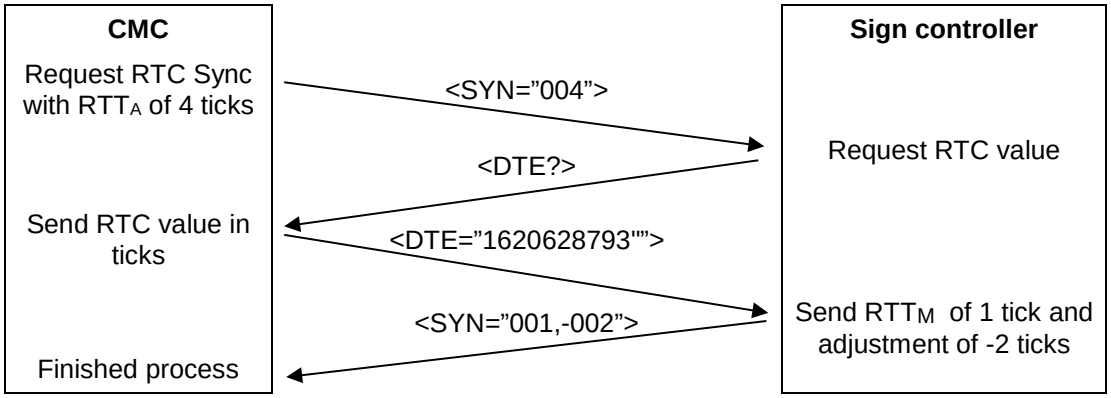

Figure 3 – RTC synchronisation process sequence example

5.4 Sign Logs and Alarm Reporting

5.4.1 Events (see TSI-SP-083 [1]) shall be reported to the CMC in the format given in Appendix D.

5.4.2 If an alarm occurs, the following process shall be carried out (see Figure 2):-

a) The Sign Controller shall send its Sign ID and the status to CMC;

b) If the ALARM bit is set in the Status word, CMC asks for the sign log file.

c) The Sign Controller responds, sending its current log;

d) CMC issues the Clear log request (CLG)

e) The Sign Controller clears all its log entries.

f) If no other alarms have occurred during the above process, the Sign Controller shall clear the ALARM bit in the Status word, and respond with the current status of the Sign Controller.

APPENDIX A MESSAGES

The lists in this appendix cover all messages in the SZAS protocol. However, some messages are aimed at the mode of operation of older signs. The applicability of individual messages to the current sign design is given in the current sign specification, TSI-SP-083 [1].

A.1 Configuration Messages

Tag	Description	Use
ADN	Get Sign Controller number. Up to 32 characters 0-9, a-z, A-Z,'.','-','_','/','\'. This is a fixed number coded into the controller.	CMC request: ADN? Sign response: ADN="80000136" Existing practice uses 8 digits. The first 4 digits are static for a manufacturer/protocol combination, and assigned by TfNSW (e.g. 8000). The next 4 digits are assigned by the manufacturer, in sequence of manufacture release. (e.g. 0136)
BVL	Get/Set battery voltage threshold. Battery voltage threshold is provided in volts and in DD.DD format.	Set battery voltage threshold in sign: CMC request: BVL="10.21" Get battery voltage threshold from sign: CMC request: BVL? Sign response: BVL="10.21"
CMC	Get/Set the CMC IP address or hostname remotely	Set CMC IP in sign CMC request: CMC=”010.109.010.001” Get CMC IP from sign: CMC request: CMC? Sign response: CMC="010.109.010.001"
CTD	Get/Set the wait period (in RTC ticks) before attempting a reconnection. Up to a 4-digit number.	Set reconnect wait period in sign: CMC request: CTD="6000" Get reconnect wait period from sign: CMC request: CTD? Sign response: CTD="6000"
DRT	Get/Set Alert Display electrical current hysteresis for display alarm trigger.	Set DRT in the sign; CMC request: DRT="XXX" Get DRT from the sign: CMC request: DRT? Sign response: DRT =”XXX”

Tag	Description	Use
ECT	Get/Set Alert Displays electrical current alarm threshold. Provided in mA in DDDD format, separated with commas. The order of the thresholds is Left Lantern, Right Lantern and Annulus.	Set electrical current threshold in sign CMC request: ECT="0435,0435,1237” Get electrical current threshold from sign: CMC request: ECT? Sign response: ECT="0435,0435,1237”
FPN	Get/Set Sign Controller flash pattern. Flash pattern is up to 32 bytes long. Each byte is structured as follows: Bit 0 – Left lantern, Bit 1 – Right lantern, Bit 2 – Annulus. The sequence of the pattern is from LSB to MSB. This is decimal encoded. If a bit is set in a byte it indicates that Alert Display is activated for that flash pattern stage.	Set flash pattern in sign CMC request: FPN ="5555511000002266666"" Get flash pattern from Sign: CMC request: FPN? Sign response: FPN="5555511000002266666""
FWV	Get the firmware version in operation. Firmware versions are sent as 32 characters in 0-9, a-z, A-Z,'.','-','_','/','\'.	CMC request: FWV? Sign response: FWV=”1.00”
ITT	Set sign call-home daily schedule times This is aimed at a sign connected to the CMC by satellite. The format is ITT=”ENA,T1,T2,T3,T4,T5,T6” ENA = number of times per day the sign shall call home. Tn = UTC time as 4 digits from 0000 to 2359. (HHMM format). To align times, the sign shall assume CMC timestamp equals epoch time. Sign reject <REJ> to occur if ENA is not 4, 5 or 6, or if used times are not in ascending sequence, or if less than 5 minutes apart.	Set call schedule: CMC request: ITT=”ENA,T1,T2,T3,T4,T5,T6” Get call schedule: CMC request: ITT? Sign response: ITT=”ENA,T1,T2,T3,T4,T5,T6” T1 applies first, unused time slots shall have an entry Example: ITT=”4,0600,0800,1400,1600,0000,0000”
MID	Get parameters of the sign’s modem :- Response format: MID=”’modem chipset manufacturer’, ‘factory serial number’, ’hardware version’, ’software version’, ’IMEI’, ‘IMSI’”. If any parameters cannot be obtained then ‘N/A’ shall be returned	CMC request: MID? Sign response: MID=“’Mfr’,’Ser’,’hw’,’sw’,’imei’,’imsi’” .
PWM	Get/Set PWM duty cycle for flashing display element Default value is 100. Note: suits when displays all same type (e.g. Lanterns only)	Set PWM in the sign: CMC request: PWM="XXX" Get PWM from the sign: CMC request: PWM? Sign response PWM =”XXX”
SGN	Get/Set get sign ID. Sign ID is up to 32 characters in 0-9, a-z, A-Z,'.','-','/','\'	Set sign ID in sign CMC request: SGN="ABC1234" Get sign ID from sign: CMC request: SGN? Sign response: SGN="ABC1234"

Tag	Description	Use
STD	Get/Set the wait period for closing a connection due to inactivity (in ticks). Up to a 4-digit number.	Set session inactivity period in sign CMC request: STD="0030" Get the session inactivity period from sign: CMC request: STD? Sign response: STD="0030"
TMO	Get/Set the communications time-out duration (in ticks). If no communications has occurred with the CMC for this duration (in ticks) an alarm shall be raised. If set to 0, the period is infinite. Up to a 6-digit number.	Set Time-out in sign CMC request: TMO="600000" Get Time-out from sign: CMC request: TMO? Sign response: TMO="600000"
TTB	Get/Set the timetable on the Sign Controller. It is in ASCII Hex format. The timetable is structured in the following manner: T = Number of timetable segments (1-byte) S_1 = segment 1 start time (4-bytes) N_1 = segment 1 number of days (1-byte) C_1= segment 1 calendar (Variable) See Appendix C for a detailed explanation.	Set timetable in sign: CMC request: TTB=”$TS_1N_1C_1S_2N_2C_2S_3N_3C_3$” Get timetable from sign: CMC request: TTB? Sign response: TTB=”$TS_1N_1C_1S_2N_2C_2S_3N_3C_3$” Example: TTB="036019bd503a0ff0ffc3ff0ffc3ff0ffc3ff0ffc3f607cabe098ffc3ff0ffc3ff0ffc3ff0ffc3ff0ffc3ff00000000ffc3fc0ffc3ff0ffc3ff0ffc3ff0ffc3ff615b6b504a0ffc3ff0ffc3ff0ffc3ff0ffc3ff0ffc3ff0ff"
TTC	Initiate sign to calculate the timetable checksum and return with the encoded checksum value. Checksum is calculated using the CRC-CCITT (XModem) method.	Get timetable checksum from sign: CMC request: TTC? Sign response: TTC="797D”
TTO	Get/Set operating durations. There will be three operating durations (in ticks) specified which are: S1=Duration of the 1st timetabled operation S2=Duration between 1st and 2nd timetabled operation S3=Duration of the 2nd timetabled operation	Set operating durations in sign: CMC request: TTO=”S1, S2, S3” Example: TTO=”5400,18000,5400” Get operating durations from sign: CMC request: TTO? Sign response: TTO=”S1, S2, S3”
TTV	Set/Get timetable version. Timetable version is up to 32 characters 0-9, a-z, A-Z,'.','-','_','/','\'.	Set timetable version in sign CMC request: TTV="XXXXX" Get timetable version from sign CMC request: TTV? Sign response: TTV=”XXXXX”

Table 2 – Configuration messages

A.2 Telemetry Messages

Tag	Description	Use
BTT	Get battery voltage from sign. Battery voltage is provided in DD.DD format.	CMC request: BTT? Sign response: BTT="12.36"
CEL	Retrieve Serving Cell Information from the mobile network Shall be in csv format and include serving cell and frequency band.	CMC request: CEL? Sign response: CEL=" 4G,4436,22……,0E8259F,29,38,NOCONN"
DER	Get display error byte. Display error byte is a decimal representation of the byte. See Appendix B.2	CMC request: DER? Sign response: DER=”1”
DMP	Initiate sign to dump trace data as string in intel hex format.	CMC request: DMP? Sign response: DMP="<intelhex_format>"
DTE	Get/Set the RTC on the Sign Controller. RTC is up to 10 digits long and less than 2^32-1.	Set RTC in sign: CMC request: DTE="1394078114" Get RTC value from sign: CMC request: DTE? Sign response: DTE="1394078114" Get RTC from CMC (part of time synchronisation sequence): Sign request: DTE? CMC response: DTE="1394078114" Sign request: SYN=”27.999”
ESC	Get display element electrical current in mA. Electrical current is to be provided in DDDD format. The order of sampled current is ‘Left Lantern, Right Lantern, Annulus’	CMC request: ESC? Sign response: ESC="0435,0429,1327"
GPS	Retrieve GPS location, in NMEA format	CMC request: GPS? Sign response: GPS = "LAT,LONG,HDOP”
LOG	Get sign log. The format of log be sent is detailed in Appendix D.	CMC request: LOG? Sign response: LOG="……."
TMP	Get sign temperature. To be provided in °Centigrade in signed DDD.D format.	CMC request: TMP? Sign response: TMP="-3.1"
RSS	Get the sign Received Signal Strength. Provided as a signed value in dBm up to three digits, whole numbers only.	CMC request: RSS? Sign response: RSS="-65"
STS	Get sign status. Status is the decimal representation of the status word. See Appendix B.1.	CMC request: STS? Sign response: STS=”25”
SVN	Retrieve more detailed firmware version info	CMC request: SVN? Sign response: SVN="SVN1244".

Table 3 – Telemetry messages

A.3 CMC command messages

Tag	Description	Use
CLG	Clear Sign log in sign.	CMC request: CLG Sign response: STS="25"
END	Terminate and acknowledge end of session.	CMC request: END
RBT	Initiate sign to reboot with the current firmware.	CMC request: RBT Sign response: <ACK> , then reboot
SCK	Initiate sign self-check. After self-check the sign returns 1 for passed or 0 for failed.	CMC request: SCK Sign response: SCK=”0”
SOP	Get/Set the state of operation of the Alert Displays. Either off (0) or on (1).	Set state of display operation in sign CMC request: SOP=”0” Get state of display operation from sign: CMC request: SOP? Sign response: SOP="0"
SYN	Request Sign controller to initiate time synchronisation. Provide round-trip time (RTT) in DDD ticks and the adjustment in DDD or –DDD ticks.	Time Synchronize the sign with the CMC CMC request: SYN=”45” Sign request DTE? CMC response: DTE="58752257" Sign response: SYN=”27.999”
TFL	Initiate sign test flash cycle for the specified number of ticks, using the specified test pattern. The test flash cycle is two digits long (ranging from 01 to 99).	CMC request: TFL=”30,Test_Flash_Pattern” Sign response: TFL# if test flash cycle not performed
UFW	Update firmware of sign. Sign to download, update, and resume operation. Detail message content and process is expected to vary to suit the sign model. The process used shall include a checksum.	CMC request: UFW="XYZ" A URL may be sent for the site to use to source its new firmware, or the firmware may be encoded in the message itself.

Table 4 – CMC command messages

A.4 Other messages

Tag	Description	Use
ACK	Acknowledgement from sign to CMC when no other response is defined.	Sign response: ACK
REJ	Reject message from sign to CMC.	Sign response: REJ
;	Used as a tag separator. Not to be used anywhere else in a message	Example: BTT?;TMP?

Table 5 – Other messages

APPENDIX B STATUS AND DISPLAY ELEMENT

B.1 Status Word

Bit	Name	Description	Status Indication
0	ALARM	Alarm state	0 No alarms 1 Alarm occurred
1	DISPERR	Display error state	0 Normal 1 A fault has occurred in the displays
2	CFGERR	Configuration error state	0 Normal 1 Configuration is corrupt (reset when rebooted)
3	FWDL	Firmware download	0 Normal 1 Firmware downloading
4	FWDLER	Firmware download error	0 Normal 1 Error occurred during firmware download
5	BTTLW	Battery low voltage	0 Normal 1 Battery voltage is below threshold
6	FLH	Flashing state	0 One or more alert displays are flashing 1 No alert displays are flashing
7	SOP	State of operation (for alert displays)	0 State of Operation “off” 1 State of Operation “on”
8	DOORSTS	Door open state	0 Door is closed 1 Door is open
9-15	-	-	*undefined, for future use*

Table 6 – Status Word

B.2 Display Error Byte

Bit	Display Element
0	Left Lantern
1	Right Lantern
2	Annulus

Table 7 – Display Error Byte

APPENDIX C TIMETABLE

C.1 Timetable structure

The timetable is sent in ASCII Hex, structured in the following manner:

Description	Length	Comment
Number of timetable segments	1-byte	Segments such as a school term
Absolute start time for the first timetabled operation (in ticks)	4-bytes	The time of day of this absolute time is to be used as the daily start time for the whole of the segment by repeatedly adding 24 hours.
Number of calendar entries in first timetable segment	1-byte	Number of days in this timetable segment
Calendar for the first timetable segment	Variable	Each day's timetabled operation are represented by 2-bits which are packed into a byte from LSB to MSB. See Table 9 below.
Absolute start time for the second timetabled operation (in ticks)	4-bytes	32-bit RTC value (CMC epoch time)
Number of calendar entries in second timetable segment	1-byte	Number of days in this timetable segment
Calendar for the second timetable segment	Variable	2-bits for each day's calendar operation. See Table 9 – below.

Table 8 – Timetable structure

C.2 Operation period enablement bits

Each day's worth of calendar entry comprises two bits that indicate which timetabled operations are required.

Bits	Description
00	Both timetabled operations are off
01	1st timetabled operation is ON and 2nd timetabled operation OFF
10	1st timetabled operation is OFF and 2nd timetabled operation is ON
11	Both timetabled operations are on

Table 9 – Operation period enablement bits

C.3 Timetable Example

The timetable message below is broken down in the table that follows:

<TTB="036019bd503a0ff0ffc3ff0ffc3ff0ffc3ff0ffc3f607cabe098ffc3ff0ffc3ff0ffc3ff0ffc3ff0ffc3ff00000000ffc3fc0ffc3ff0ffc3ff0ffc3ff0ffc3ff 615b6b504a0ffc3ff0ffc3ff0ffc3ff0ffc3ff0ffc3ff0ff">

Description	Hex (transmitted)	Value (Decimal)	Comment
Number of timetable segments (normally 3)	03	3	-
Segment 1, start time of first operation of day	6019bd50	1612299600 (ticks) CMC epoch time	Local time= 8am Wed 3rd Feb 2021 Each day in segment 1 starts 8:00am. Durations then given by TTO setting.
Segment 1, Number of days	3a	58	Calendar is sent in whole bytes. Calendar will be 30 hex digits, 15 bytes calculated from Roundup (58/4).
Segment 1 calendar	0ff0ffc3ff0ffc3ff 0ffc3ff0ffc3f	-	Binary Calendar = 111111110000111111111100001111 111111000011111111110000111111 111100001111111111000011111111 110000111111111110000111111
Segment 2, start time of first operation of day	607cabe0	1618783200 (ticks)	Local time = 8am Mon 19th Apr 2021
Segment 2, Number of days	98	152	Calendar will be 76 hex digits, 38 bytes.
Segment 2 calendar	ffc3ff0ffc3ff0ffc 3ff0ffc3ff0ffc3ff 00000000ffc3f c0ffc3ff0ffc3ff0 ffc3ff0ffc3ff	-	Binary Calendar = 111111111100001111111111000011 111111110000…(omitted data) …111111111100001111111110000 111111111100001111111111
Segment 3, start time of first operation of day	615b6b50	1633381200 (ticks)	Local time = 8am Tue 5th Oct 2021
Segment 3, Number of days	4a	74	Calendar will be 38 hex digits, 17 bytes.
Segment 3 calendar	0ffc3ff0ffc3ff0ff c3ff0ffc3ff0ffc3 ff0ff		Binary Calendar = 111111111100001111111111000011 111111110000111111111100001111 111111000011111111110000111111 111100001111111111000011111111 110000111111111110000111111111

Table 10 – Timetable example detail

As per the example shown above, calendar segment 1 has 58 days of calendar entries, in 15 bytes. The first 12 days of entries in binary are 00001111 11111100 00111111. The calendar entry's relation to day number and byte number is shown below. See Table 9 for enablement bits explanation.

Bit Significance	←MSB				←Increasing Significance							LSB
Enablement bits	00	00	11	11	11	11	11	00	00	11	11	11
Day number	12	11	10	9	8	7	6	5	4	3	2	1
Byte number	13				14				15			

Table 11 – Calendar entries

APPENDIX D LOG FILE MESSAGE

D.1 Composition of message

Each log file entry of the message shall comprise a cyclic sequence number (000 to 999), timestamp (RTC value in ticks), and a 2 digit event code field. A colon ":" shall separate each of the fields:

- Sequence Number: Timestamp: Event Code: Event Data

Event data shall be appended if the event code requires event data fields. Multiple log entries shall be separated by a comma character, ",".

Note: White spaces including TAB, carriage return and line feeds are ignored by the CMC system in the log file.

D.2 Event codes

The log entry event codes and associated event data required is tabled below.

Event code	Logged Event Description	Event data Required
01	Unexpected initialisation of the sign	None
02	Sign housing door opened	None
03	Closure of sign housing door (recovery from open)	None
04	Configuration error	None
05	Firmware download error	None
06	Battery voltage below minimum threshold	Battery voltage in DD.DD format.
07	Recovered from battery below minimum voltage	None
08	Failure of any of the Alert Displays	Display error byte in DD format, then electrical current for that display in DDDD format.
09	Recovered from Alert Display failure	None
10	CMC communications timeout	None
11	Commencement of timetabled operation	None
12	Cessation of timetabled operation	None
13	Locally triggered test flash request	None
14	Trace data available	None
15	Controller initialised due unresponsive modem	None
16	Mains power failure	None
17	Recovery from mains power failure	None
18	Battery needs replacement	None
99	Debugging Aid	Customised

Table 12 – Event codes

D.3 Example Log File Entries

Example log Entry	Explanation
001: 1394078114:01,	Sign Controller initialised
002: 1394078114:02,	Housing door opened
003: 1394078114:03,	Closure of housing door (recovered from door open)
004: 1394078114:04,	Configuration error
005: 1394078114:05,	Firmware download error
009: 1394078114:06:10.92,	Battery voltage below threshold at 10.92 V
010: 1394078114:07,	Battery voltage above threshold (recovered from below threshold)
014: 1394078114:08:02:0200,	Alert Display fault in right conspicuity device with the measured current of 200 mA
015: 1394078114:09	Recovered from Alert Display fault
016: 1394078114:10,	CMC communications timeout
017: 1394078114:11,	Commencement of timetabled operation
018: 1394078114:12,	Cessation of timetabled operation
019: 1394078114:13,	Test Flash triggered locally by opening door (some signs)
020: 1394078114:14,	Trace data available (some signs). Segger SystemView Trace data available for retrieval with <DMP?> command. Tracing is paused until existing data is retrieved.
021: 1394078114:15,	Controller initialised due unresponsive modem (some signs) Reset done if internal communication not established in 10 attempts, and time is at least 10 minutes outside flashing cycle times.
022: 1394078114:99: 0000:5aea5173,	Used for sending debug data (some signs) Recently used for sending the controller current timestamp during the time synchronisation issue troubleshooting.

Table 13 – Example log file entries