ADEC
Technologies

ADEC Protocol Specification

for TLS-Compatible Traffic Data Acquisition Using TDC1, TDC3 and TDC4 Series of Traffic Detectors

Version 2.0

1 PURPOSE ... 3
2 PRINCIPLE ... 3
3 PROTOCOL FORMAT ... 3
3.1 Byte ... 3
3.2 Packet ... 3
3.3 Long Frame, Variable Length ... 3
3.4 Short Frame, 5 Bytes Fixed Length ... 3
3.5 Single Character ... 4
4 COMMUNICATION ... 4
4.1 Data Flow ... 4
4.2 Protocol Primitives ... 4
4.3 The Control Byte ... 5
5 DATA SPECIFICATION ... 6
5.1 Reset Communication and FCB Bit ... 6
5.2 User Data ... 6
5.3 Tick Value ... 7
5.4 Retrieving JPEG Pictures From TDC4 Detectors ... 8
5.5 Traffic Data ... 10
5.6 Detector Status ... 12
6 TLS EXAMPLES ... 13
6.1 Initialization of Communication ... 13
6.2 Poll Traffic Data ... 14
6.3 Example of picture retrieval ... 15
7 SITOS EXAMPLES ... 16
7.1 Reset Detection ... 16
7.2 Traffic Data Inquiry ... 17

1 Purpose

Purpose of this document is to provide information that enables programmers and software engineers to develop data gathering devices that can obtain road traffic information from non-intrusive traffic detectors such as the TDC series from ADEC Technologies. TDC detectors are equipped with a half-duplex RS 485 interface that runs on 9600 baud, 8 characters, even Parity, 1 Stop-bit (9600, 8E1). Any device attempting to query traffic information must adhere to these settings.

2 Principle

The protocol is binary in nature, hence does not allow users to query information using terminal programs such as Windows Hyper Terminal. Half-duplex means that the client (PC, data-logger etc.) must send a query to the detector, that will response with the requested data - only one device at any given time is sending data, in accordance with the transmission rules below. The TDC does not send unsolicited data packets.

3 Protocol Format

3.1 Byte

Following the 8-E-1 rule, each byte is encoded into 11 bits as follows:

Start	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7	Parity	Stop
	LSB							MSB		
0	N	N	N	N	N	N	N	N	P	1

3.2 Packet

Keeping the Byte sequence in mind, each multi-byte packet consists of a sequence of bytes in the following format:

Header	Application Data	Checksum	End byte

Where as:

Header: Identifies protocol type etc.

Application data: includes everything application specific, including detector address (RS 485 ID), parameter values etc.

Checksum = Sum of all byes in Application Data modulo 256.

Multi-byte values are transmitted higher-valued bytes first, for example a 16-bit value of 0x1234 is transmitted as follows:

Byte n-1	Byte n	Byte n+1	Byte n+2
…	0x12	0x34	...

The protocol implements three basic packet types:

- Long Frame
- Short Frame
- Single Character

3.3 Long Frame, Variable Length

Header				Application Data			CHK	EndByte
Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7…m	Byte n - 1	Byte n
0x68h	Length	Length	0x68h	CTRL Byte	ADR Byte	Data	Check-sum	0x16h

3.4 Short Frame, 5 Bytes Fixed Length

Header	Application Data		CHK	EndByte
Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
0x10h	CTRL Byte	ADR Byte	Checksum	0x16h

3.5 Single Character

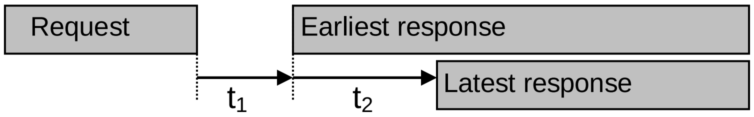
Header
Byte 1
0xE5h

4 Communication

4.1 Data Flow

Besides the format of the bytes and data packets, the following rules apply:

1. Telegrams that don’t meet the above format are discarded
2. After receiving a request, the response must be sent after 33 bits (3.3 ms, t_1) but not later than 10 ms thereafter (t_2):

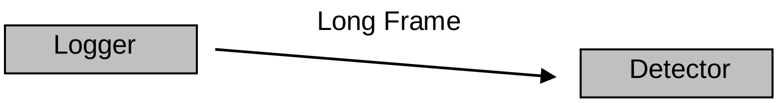


4.2 Protocol Primitives

The PC or data-logger connects to the detector whereby any of the following protocol primitives are used:

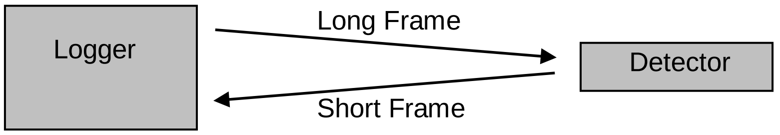
4.2.1 Send / No Reply

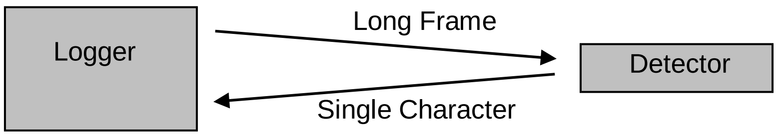
The data logger sends a data packet to the detector, no response is expected - the detector does not respond to confirm the receipt of the packet:



4.2.2 Send / Confirm

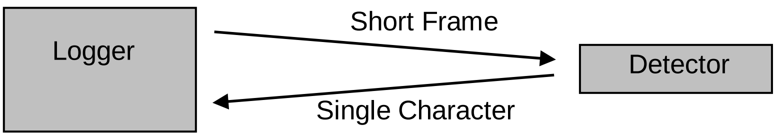
The data logger sends a packet to the detector, the detector replies with a short frame or a single character to confirm the receipt of the packet:



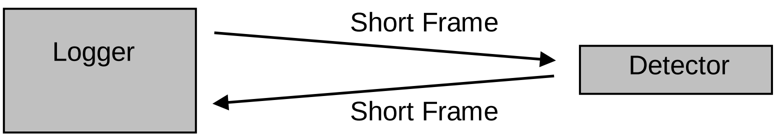


4.2.3 Request / Response

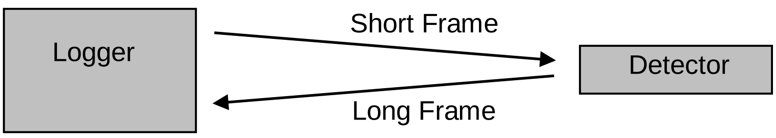
The logger sends a short packet to the detector requesting information, the detector replies with any appropriate response. The request for example is a request for new traffic data, upon which the detector replies that no information has arrived using a single character.



The detector may also response using a Short Frame, to confirm the receipt of a command:



To report traffic data, the detector receives the request for such data and replies with a Long Frame containing the information of one or more vehicle:



A traffic detector never sends traffic data without having received a request to do so.

4.3 The Control Byte

The Control Byte is part of every frame except the single character response. The composition and the meaning of the bits within the control byte depend on the originator of the packet to which the Control Byte belongs:

4.3.1 The Control Byte from the Logger

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
MSB							*LSB*
0	1	FCB	FCV	See below			

- FCB: The Frame Count Bit and Frame Count Valid (FCB, FCV) is used by the peers to indicate whether a packet has been received properly. To indicate to the detector that the previous packet has been properly received, the logger toggles the FCB bit in the subsequent request. If it doesn’t do so, the detector will repeat the previous response. The FCB bit is only used in conjunction with requests for traffic data. The first query to a detector is ‘1’.
- FCV: a value of ‘1’ indicates if the FCB is valid and needs consideration
- Bit 0 - 3: these lower 4 bits specify the nature of the content of the inquiry as follows:
  0: Reset communication and FCB bit, start with anticipated value of ‘1’, see section 5.1 on page 6.
  3: Requests the detector to return the user data
  4: Requests the detector to return the tick value
  8: Requests the detector to return all available information of the recent vehicle(s)
  9: Requests the detector to return its status

4.3.2 The Control Byte from the Detector

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
MSB							*LSB*
0	0	ACD	DFC	See below			

ACD: Access Demand, the detector does not use this function, the value is always 0

DFC: Data flow control, the detector does not use this function, the value is always 0

Bit 0 – 3: these four lower bits specify the type of data contained within the packet, in particular:

4: Tick value

8: Traffic Data

11: Detector Status

It is worth noting that Bit 6 is '1' in packets that originate with the logger, equally, the bit is set to '0' in all packets sent from detectors.

5 Data Specification

Note: The examples below assume the address (RS 485 ID) of the detector is 1 for illustration purposes.

5.1 Reset Communication and FCB Bit

5.1.1 Request

Header	Application Data		CHK	EndByte
Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
0x10h	CTRL Byte	ADR Byte	Checksum	0x16h
0x10h	**0x40h**	**0x01h**	**0x41h**	**0x16h**

5.1.2 Response

Header
Byte 1
0xE5h

The detector replies with 'E5' single-byte response, indicating the data buffer has been emptied and the next FCB is expected to be '1'.

5.2 User Data

The detector is set to specific settings which it does or does not apply depending on whether requested feature are supported or required as given by the setup, type and firmware of the detector:

5.2.1 Request

Header				Application Data				CHK	EndByte
Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8	Byte 9	Byte 10
0x68h	Length	Length	0x68h	CTRL Byte	ADR Byte	Data	Data	Check-sum	0x16h
0x68h	**0x04h**	**0x04h**	**0x68h**	**0x73h**	**0x01h**	**#1**	**#2**	∑	**0x16h**

The bits in the two data bytes #1 and #2 assume the following purpose:

Value Byte #1	Value Byte #2	Description
0x00h	0x00h	Set detection mode to normal (Frontfire)
	Any value ≠ 0x00h	Set detection mode to reverse(i) (Backfire)
0x01h	Any	Enable status information
0x0Eh	0x00h	Reset wrong-way driver flag(ii)
0x0Fh	0x01	Static detector info (iii)
	0x02	Dynamic detector info (iii)
	0x03	Reset sync timer (iii)

i) only supported in the model TDC3-2

ii) if “wrong-way driver auto-reset” flag is switched on (factory default) using the ADEC DetSoft software, the wrong-way driver flag is reset automatically when the next response message containing vehicle data of a normal-driving vehicle is transmitted.

iii) Not implemented unless specified

5.2.2 Response

Header
Byte 1
0xE5h

5.3 Tick Value

This command asks the detector to return the current value of the 24-bit tick value. 1 tick lasts 2.5 ms. This value is only available in the TDC3(4) models with 2+1, 5+1 and 8+1 vehicle classes.

5.3.1 Request

Header	Application Data		CHK	EndByte
Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
0x10h	CTRL Byte	ADR Byte	Checksum	0x16h
0x10h	**0x44h**	**0x01h**	**0x45h**	**0x16h**

5.3.2 Response

Header				Application Data					CHK	EndByte
Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7 - 9			Byte 10	Byte 11
0x68h	Length	Length	0x68h	CTRL Byte	ADR Byte	Data	Data	Data	Check-sum	0x16h
0x68h	**0x05h**	**0x05h**	**0x68h**	**0x04h**	**0x01h**	**#1**	**#2**	**#3**	∑	**0x16h**

Data bytes #1 – #3 contain the tick value.

5.4 Retrieving JPEG Pictures From TDC4 Detectors

Picture data from the TDC4 is transmitted embedded in the TLS “User-Data”, Code 3, packet. Since the transmission of an image can take longer than it takes for 4 vehicles to pass, it is good design practice to keep querying for traffic data during the transmission of the image data.

5.4.1 Request

Header				Application Data				CHK	EndByte
Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8-#	Byte#	Byte#
0x68h	Length	Length	0x68h	CTRL Byte	ADR Byte	Cmd Byte	Data	Check-sum	0x16h
0x68h	**Len**	**Len**	**0x68h**	**0x73h**	**0x01h**	**#**	**#**	**∑**	**0x16h**
Notes							(i)		

(i) The Data-byte is not present unless the Cmd-Byte contains the ‘Execute Command’ (Value 0x50)

CTRL-Byte, see 4.3.1 on page 5, 0x70 (packet sent by logger) + 0x03 (Request user data)

The Cmd-byte is used without data except for use with the Execute Command which requires Byte 8 to contain the actual command to be executed:

User Data: **Cmd Byte from Primary**							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Bits 4-7 contain the value of the command, the following values are possible:				0	0	0	0
1 → Get Status							
2 → Get Data							
5 → Execute Command							
6 → Get Length (of picture, in Bytes)							

5.4.2 Data-Byte

Data-byte is only necessary when the Cmd Byte refers to it through the ‘Execute Command’. The following commands are possible:

Data (Byte 8)	
Value	
0x01	→ Take picture (capture scene with current settings)
0x02	→ Release picture (after transmission or when interrupted etc.)
0x03	→ Reset communication (between detector and camera subsystem)
0x04	→ Reset Camera (causes the camera to perform reset on itself)

Note: As with all other operation-related and device-specific settings, it is necessary to use ADEC installation and commissioning software to configure the camera properties prior taking pictures, such as the desired picture size, whether to automatically capture pictures on certain traffic conditions etc.

The picture is encoded in JPEG format. The detector is first instructed to capture a picture, subsequently, the picture size is queried from the detector and lastly the actual picture data is retrieved. While not mandatory, it is good practice to release the picture after it has been retrieved. The detector response confirms any commands by repeating the command back to the caller, adding any status information to the lower four bits of the command byte (Byte 7 of the user-data packet)

5.4.3 Response message structures

The response message can take various forms, depending on what the query to the detector.

Header				Application Data				CHK	EndByte
Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8-#	Byte#	Byte#
0x68h	Length	Length	0x68h	CTRL Byte	ADR Byte	Cmd Byte	Data	Check-sum	0x16h
0x68h	**Len**	**Len**	**0x68h**	**0x03h**	**0x01h**	**#**	**#**	**∑**	**0x16h**
Notes							(i)		

The Cmd Byte repeats the command received by the detector, Bit0-Bit3 always contain the status of the detector:

User Data: **Cmd Byte from Detector**							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Bits 4-7 contain the value of the command that has been sent to the detector in the previous request				offline	busy	error	pic

offline	→	The camera is currently not available
busy	→	The camera is currently busy, repeat request later
error	→	An error has occurred
pic	→	A picture is currently stored in the camera ready for retrieval

5.4.4 Response Data Byte

5.4.4.1 Response to *Get Status* Request

Data (Byte 8): Trigger Source(s)							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	0	Queue	Wrong-Way Driver	0	0	0	0

A value of '1' of either bit indicates that the corresponding event will trigger the detector to capture a picture automatically

5.4.4.2 Response to *Get Length* Request

Data (Byte 8-11): Picture Size [Bytes]			
Byte 8	Byte 9	Byte 10	Byte 11
Bit 24-31	Bit 16-23	Bit 8-15	Bit 0-7

This request is sent to retrieve the number of bytes the JPEG image contains. The response contains a 32-bit number indicating the size of the picture. Note that the request normally has to be sent twice or more as the first query causes the detector to calculate the length, the subsequent query will return the proper size (in bytes) of the image. The 'busy' flag is returned instead.

5.4.4.3 Response to *Get Data* Request

Data (Byte 8-n): JPEG Picture Data						
Byte 8	Byte 9	Byte 10	Byte 11	Byte 12	…	Byte n
Packet # High	Packet # Low	Data Len High	Data Len Low	Image Data		

- Packet # (Hi/Lo) contain the sequence of the packet. The first packet contains '0x00 0x00'
- Data Len (Hi/Lo) contain the actual number of bytes in "Image Data", for standard 64-byte image data per packet, the value is '0x00 0x40', the last byte is therefore Byte 51. Note that the last frame may contain fewer than 64 bytes net
- Image Data is the JPEG-formatted picture data. The response contains no Image Data and the 'busy' flag set if no image data is available

5.4.4.4 Other Responses

The responses to the Execute Command requests do not contain additional data. The response contains mere confirmation and the current value of the status flags, see first section of this chapter.

5.5 Traffic Data

The logger obtains traffic information from the detector using this request. The detector responses with ‘E5’ if no new vehicles have passed since the last request, or with the detailed information of each of the vehicles that have passed since the past request. The maximum number of vehicles that the detector can store is four. Any additional vehicles will cause the data of the oldest vehicle to be replaced with the data of the most recent vehicle (FIFO).

5.5.1 Request

Header	Application Data		CHK	EndByte
Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
0x10h	CTRL Byte	ADR Byte	Checksum	0x16h
0x10h	**0x58h**	**0x01h**	**0x59h**	**0x16h**

5.5.2 Response – No data

If there’s no new vehicle information, the detector responds acknowledging the receipt of the request. The logger then knows that there’s no new vehicle information and no status change has occurred either.

Header
Byte 1
0xE5h

5.5.3 Response – Pending Status and No Traffic Data

If a status is pending the detector reports the status as follows:

Header				Application Data			CHK	EndByte
Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8	Byte 9
0x68h	Length	Length	0x68h	CTRL Byte	ADR Byte	Data	Check-sum	0x16h
0x68h	**0x03h**	**0x03h**	**0x68h**	**0x08h**	**0x01h**	**Status**	∑	**0x16h**

See section 5.6.2 on page 12 for details on description of bits in the status byte.

5.5.4 Response With Queue Status

If a queue status is pending the detector sends traffic data as follows:

The response to the first request after a queue is detected returns

- speed = 0
- class = 32 (TDC3-2 / TDC4-2) or class = 6 (TDC3-3 / TDC4-3, TDC3-5 / TDC4-5, TDC3-8 / TDC4-8)
- occupancy = time between request and detection of queue
- time gap = time between beginning of queue and event immediately preceding queue detection
- counter = value of the last vehicle

Request during queue returns

- speed = 0
- class = 32 (TDC3-2 / TDC4-2); class = 6 (TDC3-3 / TDC4-3, TDC3-5 / TDC4-5, TDC3-8 / TDC4-8)
- occupancy = time since last request occurred
- time gap = 0 sec
- counter = value of the last vehicle

Request after queue is finish

- speed = vehicle speed (subject to significant inaccuracies)
- class = 32 (TDC3-2 / TDC4-2); class = 6 or truck (TDC3-3 / TDC4-3, TDC3-5 / TDC4-5, TDC3-8 / TDC4-8)
- occupancy = time since last request occurred
- time gap = 0 sec
- counter = value of the last vehicle plus 1

5.5.5 Response – Traffic data

Header				Application Data			CHK	EndByte
Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7 … n-2	Byte n-1	Byte n
0x68h	Length	Length	0x68h	CTRL Byte	ADR Byte	Data	Check-sum	0x16h
0x68h	**n-6**	**n-6**	**0x68h**	**0x08h**	**0x01h**	**Data**	∑	**0x16h**

The data section consists of three subsections (vehicles 2, 3 and 4 are optional data):

Data				
Byte 7	Byte 8 – 11	Byte 12 – *k*	…	Bytes n-*k*-2 … n-2
Status	Lifetime Vehicle Counter (32-bit)	Vehicle Information	…	Vehicle Information
0x0Bh	**0 – 2^{32}**	**Vehicle 1**	…	**Vehicle 4**
Notes:		(i)	(ii)	(ii)

i) The vehicle data comprises 6, 7 or 11 bytes per vehicle as a function of the different optional features. Note that TDC3-2 and the TDC4-2 detector models TLS Length field is set to 0 if 'On', it is omitted when 'Off'. The TDC1-PIR **always** provides vehicle length information.

TLS Length	Off	On	On
Lane-Changing vehicle detection	Off	Off	On
Vehicle data length *k*, per vehicle [bytes]	6	7	11

ii) Data for more than one vehicle is only sent if more data is in the buffer.

Vehicle 1 Information									
Byte 12	Byte 13	Byte 14	Byte 15	Byte 16	Byte 17	Byte 18	Byte 19	Byte 20-21	Byte 22
Speed	Class & Lane Position	Occupancy in units of [10 ms]		Time Gap to previous vehicle in [10 ms]		Length in [0.1 m]	reserve	Time Stamp [2.5 ms]	reserve
0…0xFFh (0 … 255)	**0x01h … 0x3Fh**	**0 … 0xFFFFh (0 … 655.35 s)**		**0 … 0xFFFFh (0 … 655.35 s)**		**0 … 0xFFh (0…25.5 m)**	**0**	**0 … 0xEA60h (0 … 150 s)**	**0**
Notes:	(iv)					(i)	(i, iii)	(i, iii)	(i, iii)

iii) These three bytes is a time-stamp value sync'ed between Master and Slaves and is only added in the frame if the lane-changing vehicle detection feature is turned on.

iv) Only available on 2+1, 5+1 and 8+1 models with lane-changing feature turned on. The upper two bits of the vehicle class indicate the lane position of the vehicle in driving direction, as follows:

Lane Information & Vehicle Class Information							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Lane Information		Vehicle Class Information					
0	0	→ Middle of the lane					
0	1	→ Left side of the lane					
1	0	→ Right side of lane					

5.5.5.1 Vehicle Class Information (according to TLS)

<table>
<tr><th colspan="2">Vehicle type</th><th>TDC3-8 / TDC4-8</th><th>TDC3-5 / TDC4-5</th><th>TDC3-3 / TDC4-3</th><th>TDC3-2 / TDC4-2</th><th>TDC1</th></tr>
<tr><td>Motorcycle</td><td></td><td>10</td><td rowspan="3">1</td><td rowspan="3">32</td><td rowspan="4">32</td><td rowspan="4">1</td></tr>
<tr><td>Car</td><td></td><td>7</td></tr>
<tr><td>Van</td><td></td><td>11</td></tr>
<tr><td>Unknown</td><td></td><td>6</td><td>6</td><td>6</td></tr>
<tr><td>Car w/ trailer</td><td></td><td>2</td><td>2</td><td rowspan="6">33</td><td rowspan="6">33</td><td rowspan="3">3</td></tr>
<tr><td>Bus</td><td></td><td>5</td><td>5</td></tr>
<tr><td>Truck</td><td></td><td>3</td><td>3</td></tr>
<tr><td>Truck w/ Trailer</td><td></td><td>8</td><td rowspan="2">4</td><td rowspan="2">4</td></tr>
<tr><td>Tractor w/ semi-trailer</td><td></td><td>9</td></tr>
</table>

(i)

i) The TDC1-PIR supports three classes by length: class 1: <5.6 m, class 3: 5.6 m – 12.2 m, class 4: >12.2 m

5.6 Detector Status

The detector status is normally sent along with the traffic information. If no traffic information is present, the detector sends the status only (see section 5.5.3 on page 10). Please note that the bits in the Status Byte have different meanings for different families of TDC detectors.

5.6.1 Request

Header	Application Data		CHK	EndByte
Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
0x10h	CTRL Byte	ADR Byte	Checksum	0x16h
0x10h	**0x49h**	**0x01h**	**0x4Ah**	**0x16h**

5.6.2 Response

Header				Application Data			CHK	EndByte
Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8	Byte 9
0x68h	Length	Length	0x68h	CTRL Byte	ADR Byte	Status	Check-sum	0x16h
0x68h	**0x03h**	**0x03h**	**0x68h**	**0x0Bh**	**0x01h**	**Status**	∑	**0x16h**

5.6.3 TDC3 Series Status Byte

The **status-bit** is encoded as follows, a value of '1' indicates *active*:

Status Byte							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
HW fault	Sync fault	Queue	Wrong-way driver	Ultrasonic notification	IR 2 notification	IR 1 notification	Radar notification

Notes: (i)

i) **Important**: wrong-way driver notification may be available for transmission after the actual vehicle event of the wrong-way driving vehicle has already occurred, this means that the vehicle data of the wrong-way driver may not show the wrong-way driver flag yet. Subsequent request will show the updated wrong-way driver flag, see 5.5.3 on page 10.

5.6.4 TDC1-PIR Series Status Byte

The **status-bit** is encoded as follows, a value of '1' indicates *active*:

Status Byte							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
HW fault		Queue	Wrong-way driver		Low supply voltage	Thermo notification	IR notification
Notes			(i)		(ii)		

i) **Important:** wrong-way driver notification may be available for transmission after the actual vehicle event of the wrong-way driving vehicle has already occurred, the means that the vehicle data of the wrong-way driver may not show the wrong-way driver flag yet. Subsequent request will show the updated wrong-way driver flag, see 5.5.3 on page 10.

ii) This bit is set to '1' when the supply voltage drops below the minimum allowable value (see TDC1 installation manual)

6 TLS Examples

The following examples illustrate the data exchange between the client (Logger, TLS out-station, computer) and the detector. In each example, is assumed that

- The detector's unique RS 485 ID (=address) is 1
- All data is shown in hexadecimal format.

While it is not necessarily evident from the brief examples below, the FCB bit must be toggled with each subsequent requests. Not doing so indicates to the detector that the previous response has not been received and the detector then attempts to re-sending the same data as in the previous response. Consult 4.3.1 on page 5 for a detailed description on where the FCB bit is and how it needs to be handled.

6.1 Initialization of Communication

Detectors may or may not require communication reset to take place before traffic data can be requested. Therefore it is good design practice to always initialize the communication to each detector that is connected to the RS 485 segment before attempting to obtain traffic and status information:

6.1.1 Request: Get Status

Byte #	1	2	3	4	5
Description	TLS msg type	CTRL Byte	ADR Byte	Checksum	TLS end-of-msg
Value	**0x10**	**0x49**	**0x01**	**0x4A**	**0x16**

CTRL-Byte, see 4.3.1 on page 5, 0x40 (data from logger) + 0x09 (get status)

6.1.2 Response: Status Clear Via Long TLS Frame

Byte #	1	2	3	4	5	6	7	8	9
Description	Long frame header				CTRL	ADR	Status	Chk	TLS EOM
Value	**0x68**	**0x03**	**0x03**	**0x68**	**0x0B**	**0x01**	**0x00**	**0x0C**	**0x16**

CTRL-Byte, see 4.3.2 on page 6, 0x0B (detector status)

6.1.3 Response: Status Not Clear Via Long TLS Frame

Byte #	1	2	3	4	5	6	7	8	9
Description	Long frame header				CTRL	ADR	Status	Chk	TLS EOM
Value	**0x68**	**0x03**	**0x03**	**0x68**	**0x0B**	**0x01**	**0x08**	**0x14**	**0x16**

CTRL-Byte, see 4.3.2 on page 6, 0x0B (detector status)

Status, see 5.6 on page 12, 0x08 (ultrasonic fault, maybe commissioning hasn't completed successfully)

6.1.4 Request: Reset Communication

Byte #	1	2	3	4	5
Description	TLS msg type	CTRL Byte	ADR Byte	Checksum	TLS end-of-msg
Value	**0x10**	**0x40**	**0x01**	**0x41**	**0x16**

6.1.5 Response: Confirm Reset of Communication

Byte #	1
Description	TLS Single Char
Value	**0xE5**

6.2 Poll Traffic Data

After 0xE5 has been received above, the detector is now ready to receive requests for traffic data. Traffic data is requested as follows (excerpt from DetSoft Protocol Sniffer).

Each row in the table is a request or a response. The arrow after the time-stamp indicates whether it is a request or a response:

→ request

← response

Format of the time-stamp:

HH:MM:SS:LLL → hours, minutes, seconds, milliseconds

03:12:31:218 → 10 58 01 59 16	Request to ID 1 to send traffic data. FCB is not set here
03:12:31:250 ← 68 03 03 68 08 01 08 11 16	no traffic data but ultrasound alarm is still active
03:12:38:453 → 10 78 01 79 16	Request to ID 1 to send traffic data. FCB is set here
03:12:38:531 ← E5	traffic data buffer is empty, and no alarm is active
03:13:11:453 → 10 58 01 59 16	Request to ID 1 to send traffic data. FCB is not set here
03:13:11:500 ← 68 0E 0E 68 08 01 00 00 00 00 04 4E 08 03 65 1C 68 FE 4D 16	Response from detector ID 1 with traffic data (CTRL-Byte is 8): status shows no alarms, lifetime vehicle counter shows 0x00, 0x00, 0x00, 0x04 → 4 vehicles, followed by 0x4E: vehicle speed = 78 km/h 0x08: vehicle class, 8 on TDC3-8 means Truck with Trailer 0x03, 0x65: occupancy = 0x0365 → 869d → 8.69 seconds 0x1c 0x68: time gap = 0x1c68 → 7272 → 72.72 seconds 0xFE: length = 0xFE = 254 → 25.4 meters

6.3 Example of picture retrieval

03:53:07:937 → 68 04 04 68 73 04 50 01 C8 16	0x68 04 04 68 TLS long-frame header 0x73 7 from primary, FCB set, user data (Code 3) 0x04 RS 485 ID of detector 0x50 "Execute Command" 0x01 Command = 1: Take Picture
03:53:07:968 ← 68 04 04 68 03 04 50 01 58 16	0x68 04 04 68 TLS long-frame header 0x03 TLS Code 3, msg originating w/ detector 0x04 RS 485 ID of sender 0x50 "Execute Command" 0x01 Command = 1: Take Picture
03:53:08:031 → 68 03 03 68 53 04 60 B7 16	0x68 03 03 68 TLS long-frame header 0x53 5 from primary, FCB not set, user data 0x04 RS 485 ID of detector 0x60 Get Length (of picture data)
03:53:08:062 ← 68 03 03 68 03 04 65 6C 16	0x68 03 03 68 TLS long-frame header 0x03 Code 3 response 0x04 RS 485 ID of sender 0x65 GetLength, picture ready, camera busy
03:53:08:109 → 68 03 03 68 73 04 60 D7 16	< query length again >
03:53:08:140 ← 68 07 07 68 03 04 61 00 00 31 8C 25 16	0x61 Get Length, picture ready Length: 0x0000318c = 12,684 bytes
03:53:08:187 -> 68 03 03 68 53 04 20 77 16	0x20 Get Data
03:53:08:296 ← 68 47 47 68 03 04 21 00 00 00 40 FF D8 FF FE 00 24 D9 28 16 C2 00 00 00 00 00 00 00 00 00 00 00 00 00 00 F0 00 40 01 15 00 32 12 0B 51 04 51 04 00 00 FF DB 00 84 00 05 04 04 05 04 03 05 05 04 05 06 06 05 06 08 0D 09 08 07 46 16	0x21 Get data, picture ready 0x00 00 00 40 …. 09 08 07 Picture data, frame 0x0000
03:53:08:359 → 68 03 03 68 73 04 20 97 16	
03:53:08:468 ← 68 47 47 68 03 04 21 00 01 00 40 07 08 10 0B 0C 0A 0D 13 11 14 14 13 11 12 12 15 18 1E 1A 15 16 1D 17 12 12 1A 24 1B 1D 1F 20 22 22 22 14 19 25 28 25 21 27 1E 21 22 20 01 06 06 06 08 07 08 0F 09 09 0F 20 16 12 16 20 20 20 20 ED 16	0x21 Get data, picture ready 0x00 01 00 40 …. 20 20 20 Picture data, frame 0x0001

….. transmission of picture data frames ….

03:53:42:343 → 68 03 03 68 73 04 20 97 16	
03:53:42:453 ← 68 47 47 68 03 04 21 00 C5 00 40 5D D0 A6 32 BC F1 D6 91 D9 36 70 CB C7 4E 6A 39 58 EC 05 91 63 C1 65 E9 EB 51 28 5E 49 75 E7 9E B4 B9 5A E8 08 8B 2A 14 96 75 C9 F7 A8 59 23 62 72 57 8F 7A 2C C6 93 29 CC 88 1B 01 C7 E7 55 25 82 16	0x21 Get data, picture ready 0x00 C5 00 40 …. E7 55 25 Picture data, frame 0x00C5
03:53:42:515 → 68 03 03 68 53 04 20 77 16	
03:53:42:562 ← 68 13 13 68 03 04 21 00 C6 00 0C 00 0E AB F9 D3 B3 03 FF FF FF FF D9 0A 16	0x21 Get data, picture ready 0x00 C6 00 40 …. FF FF D9 Picture data, frame 0x00C6
03:53:42:656 → 68 04 04 68 73 04 50 02 C9 16	0x50 Execute command 0x02 Release picture
03:53:42:687 ← 68 04 04 68 03 04 50 02 59 16	0x50 Execute command 0x02 Release picture

7 SiTOS Examples

If the detector is configured to operate in SiTOS-mode, the behavior described below characterizes the request/respond protocol and the content of the long-frame packages.

The detectors must be configured with “Output TLS Length” and “Provide Lane Position” both turned on.

7.1 Reset Detection

When the detector resets, it must not respond to Code 8 request.

The TLS station must issue a Code 0 first for the detector to response to Code 8 requests.

The detector acknowledges Code 0 with E5.

The first Code 8 request after the Code 0 must return the status flags even if no flag is set, or the status flags together with vehicle data if there are vehicles in the buffer.

```
02:00:43:671 -> 10 58 03 5B 16
02:00:43:703 <- E5
02:00:44:671 -> 10 78 03 7B 16
02:00:44:687 <- E5

< Detector resets, TLS station continues to send Code 8 requests >

02:01:06:562 -> 10 58 03 5B 16
02:01:06:968 -> 10 58 03 5B 16
02:01:07:375 -> 10 58 03 5B 16
02:01:07:796 -> 10 58 03 5B 16
02:01:08:250 -> 10 58 03 5B 16
02:01:08:671 -> 10 58 03 5B 16
02:01:09:140 -> 10 40 03 43 16                -> TLS Station initializes communication, FCB=0
02:01:09:156 <- E5                            -> Detector acknowledges with 'E5'
02:01:09:562 -> 10 78 03 7B 16                -> 1st traffic data request Code 8, FCB=1, FCV=1
02:01:09:593 <- 68 03 03 68 00 03 00 03 16    -> Code 0 Response includes status even if no vehicles in buffer and all flags are clear
02:01:09:625 -> 10 58 03 5B 16
02:01:09:656 <- E5
```

7.2 Traffic Data Inquiry

Traffic data requests is transmitted via Code 8

The detector responses via E5 if no traffic data is in buffer and no status flag is set.

The detector responds with Code 0 and the status-byte only if one or more of the status flags is set, even if no vehicle data are in the vehicle data buffer

If the detector has new vehicles in the buffer since the last request, it sends the vehicle data to the TLS station via Code 0 (never Code 8).

The response must include in any circumstance the TLS-length, the time-stamp and the 2 reserve bytes. Detector models (TDC 3-2 for example) that natively do not support TLS length or time-stamp for lane-changing vehicles compose and transmit same message with these specific bytes filled with '00' (zero).

```
02:18:18:453 -> 10 78 03 7B 16
02:18:18:468 <- E5
02:18:19:453 -> 10 58 03 5B 16
02:18:19:500 <- 68 12 12 68 00 03 00 00 00 00 86 4E 08 03 65 FC 9A FE 00 86 54 00 B5 16
02:18:20:453 -> 10 78 03 7B 16
02:18:20:484 <- E5
02:18:21:453 -> 10 58 03 5B 16
02:18:21:468 <- E5

< error occurs in ultrasonic subsystem >

02:18:22:453 -> 10 78 03 7B 16
02:18:22:487 <- 68 03 03 68 00 03 08 03 16    -> status change is reported even if no vehicle data in buffer
02:18:23:453 -> 10 58 03 5B 16
02:18:23:467 <- E5
02:18:24:453 -> 10 78 03 7B 16
```

00	CTRL Byte
00, 08	flags
00 00 00 86	vehicle counter
4E	speed
08	class
03 65	occupancy
FC 9A	time-gap
FE	TLS length
00	reserve
86 54	time-stamp
00 B5 16	reserve, checksum, end-char