Modular Communication Protocol 'X' Overview

This document describes a modular communication protocol, which is used by our devices to send location, status and other information to an application server. Protocol 'X' is supported by all our devices, and supports all available features. The modular protocol format allows the packet content to be customised to include only the modules desired by the user, and those modules can be included in reports as and when appropriate.

Packet level hand-shaking is identical to our earlier protocols, K, M and V. Many of the data fields share a common format with those protocols, although note that some have been revised to increase range or precision.

Packets are sent using TCP or UDP sockets, directly to a nominated IP address (or hostname) and port number (see IPAD and PORT parameters).

TCP Mode
We recommend the use of TCP mode for better diagnostics in the event of communication problems. Successful receipt of each packet should be acknowledged by the host server using a single byte, numerical value 06 (NOT ascii '06'), hereafter referred to as an <ACK>.

The host server should respond to LOGIN packets and reports with <ACK>. Once the <ACK> has been received, the *device* will proceed to send a further packet and wait again for <ACK> from the host server. When all data has been sent, the *device* will go into idle mode until further data is queued for sending. The TCP socket will be left open, as this reduces data usage significantly. The host server may close the TCP socket at any time, in which case the *device* will detect the state and re-open as and when necessary.

In TCP mode, there is an option to send device IMEI with every packet or to send a single LOGIN packet only once after the socket is opened and thereafter send packets without the LOGIN data. Thereafter, the login packet is not resent unless requested by the host. Please refer to the MODE command for details of communication protocol and LOGIN options.

UDP Mode
In UDP mode the device IMEI (LOGIN data) must be sent with every packet, as there is no concept of an open socket.

Data Delivery
Reports are sent as they are queued in real-time. If there are communication service issues, reports will be stored and therefore packets may contain several reports each, up to a maximum packet size of 1024 bytes. The data report storage is implemented as a FIFO (first in first out) buffer, so that reports are always delivered in chronological order (oldest first).

Acknowledgment
Elapsed time between sending the packet and receiving the <ACK> is to be kept to a minimum (typically < 1 second). If no <ACK> is received within a specified timeout, the *device* will retain the data and re-try after a short delay (typically 60 seconds). The <ACK> timeout default is 15 seconds, but is user configurable. The requirement for an <ACK> can be disabled by setting TCPT to zero, but this is not recommended. Please refer to the Astra *Device Configuration Guide - Generic Command Reference* document for details of TCPT and other device configurations.

Note that the ACK confirms *successful receipt* of a packet, not successful parsing or anything else. We recommend that all packets are responded with an ACK, regardless of parsing errors. If no ACK is received, the *device* will continue to send the same report(s) and the communications will be blocked. LOGIN packets may be denied (NAK or no response) based on the IMEI being recognised, but after successful LOGIN, we recommend that every packet received should be responded with ACK.

OTA commands can be delivered to the *device* whilst the socket is open. Format is the same as for RS232 mode commands, although in TCP/UDP/SMS mode, multiple commands can be delivered in a single packet. Successfully processed commands will be acknowledged as below:

$DIST,OK$IGNM,OK

Please refer to Astra Device Configuration Guide - Generic Command Reference document for more details of configuration parameters and Astra $ command protocols.

LOGIN PACKET:

When MODE is set to 6, the *device* will send a LOGIN packet to each time a new TCP socket is opened. Thereafter, the LOGIN packet will not be resent unless specifically requested (using $LOGN command) or if/when the socket gets closed and has to be re-opened. The LOGIN packet is ASCII format, variable length with semi-colon delimiters, as follows:

```
$ASTRA;<model>;<IMEI>;<VIN>;<firmware-version>;<protocol>;<hw-version>
```

where:

<model>	device model number, e.g. “AT240V8”
<IMEI>	device IMEI number, 15 digit unique identifier
<VIN>	Vehicle identification number from FMS interface
<firmware-version>	device firmware version number, e.g. 6.0.9.0
<protocol>	a single character to identify the protocol used in the binary report packets that will follow the LOGON, in this case 0x58 or ascii ‘X’
<hw-version>	device hardware version

example:

```
$ASTRA;AT240V8;352218031234567;W1294000023443;7.0.1.1;X;8.0
```

PACKET CONTENT:

Protocol X is selected using the $PROT configuration parameter:

```
$PROT,<protocol>,<mask>
```

Where: <protocol> should be set to 16 for protocol X
<mask> defines the modules required in bitmask format

Protocol X modules are described in an accompanying excel file, which can be used to calculate the required mask setting, as follows:

	A	B	C	D	E	F	G	H	I	J
1	ASTRA TELEMATICS MODULAR PROTOCOL 'X'				MODULE MASK:	4497339	HIGHLIGHTED MODULES HAVE BEEN IMPLEMENTED AS OF			
2	Version 1.15, 15/12/2016									
3	MODULE#	MASK	enable?	MODULE OPTIONS:	data fields	included	source	format	units	bytes
31										
32	1	1	1	DEVICE POWER:	external input voltage	every report		integer	volts/0.2	1
33					internal back-up battery level			integer	%	1
34									module total:	2
35										
36	2	2	1	GPS DATA:	gps time/date, last known good	every report		integer	julian sec	4
37					gps latitude			integer	WGS84*10e6	4
38					gps longitude			integer	WGS84*10e6	4
39					gps current speed			integer	2 kmh per bit	1
40					gps max speed since last report			integer	2 kmh per bit	1
41					gps heading			integer	2 deg per bit	1
42					gps altitude			integer	20m per bit	1
43					gps journey distance			integer	100m per bit	2
44									module total:	18
45										
46	3	4	0	DIGITALS:	current digital I/O states	every report		bitfield		2
47					digital I/O changes mask			bitfield		2
48									module total:	4
49										
50	4	8	1	ANALOGUES:	ADC1 input current value	every report		integer	raw ADC	2
51					ADC2 input current value			integer	raw ADC	2
52									module total:	4
53										
54	5	16	1	DRIVER BEHAVIOUR:	accel X max since last report	whilst ignition is on (in-journey)	accelerometer	integer	m/s/s*10	1
55					accel X min since last report		accelerometer	integer	m/s/s*10	1
56					accel Y max since last report		accelerometer	integer	m/s/s*10	1
57					accel Y min since last report		accelerometer	integer	m/s/s*10	1
58					accel Z max since last report		accelerometer	integer	m/s/s*10	1
59					accel Z min since last report		accelerometer	integer	m/s/s*10	1
60					idle time		gps	integer	seconds	2
61									module total:	8
62										
63	6	32	1	SIGNAL QUALITY:	GSM RSSI + GPS/GNSS no. of SATS	every report		integer	see comment	1
64										

Calculate the mask by putting a '1' in the ***enable?*** column of the excel sheet for all modules required. Enter '0' for modules not required. The associated protocol mask setting will be calculated at the top of the document, as can be seen in cell F1, of the above example, showing a mask value of 4497339.

Enabling a given module in the mask tells the device to include that module as and when appropriate. Some modules will be included in every report, whilst others will be included only when the data is relevant. The ***included*** column gives guidance on when each module will be present in the report:

MODULE OPTIONS:	data fields	included	source	format	units	bytes
DIGITALS:	current digital I/O states	every report		bitfield		2
	digital I/O changes mask			bitfield		2
					module total:	4
ANALOGUES:	ADC1 input current value	every report		integer	raw ADC	2
	ADC2 input current value			integer	raw ADC	2
					module total:	4
DRIVER BEHAVIOUR:	accel X max since last report	whilst ignition is on (in-journey)	accelerometer	integer	m/s/s*10	1
	accel X min since last report		accelerometer	integer	m/s/s*10	1
	accel Y max since last report		accelerometer	integer	m/s/s*10	1
	accel Y min since last report		accelerometer	integer	m/s/s*10	1
	accel Z max since last report		accelerometer	integer	m/s/s*10	1
	accel Z min since last report		accelerometer	integer	m/s/s*10	1
	idle time		gps	integer	seconds	2
					module total:	8

Details of data format and content can be found in the comments for certain cells. Hold your mouse over any cell with a red triangle at the top-right corner to view the comment.

DIGITALS:	current digital I/O states			bitfield		2
	digital I/O changes mask			bitfield		2
					module total:	4
ANALOGUES:	ADC1 input current value			integer	raw ADC	2
	ADC2 input current value			integer	raw ADC	2
					module total:	4
DRIVER BEHAVIOUR:	accel X max since last report			integer	m/s/s*10	1
	accel X min since last report			integer	m/s/s*10	1
	accel Y max since last report			integer	m/s/s*10	1
	accel Y min since last report			integer	m/s/s*10	1
	accel Z max since last report			integer	m/s/s*10	1
	accel Z min since last report		accelerometer	integer	m/s/s*10	1
	idle time		gps	integer	seconds	2
					module total:	8
SIGNAL QUALITY:	GSM RSSI + GPS/GNSS no. of SATS	every report		integer	see comment	1

state of digital inputs (low byte) and outputs (high byte):
bit 0 - digital input 1
bit 1 - digital input 2
bit 2 - digital input 3
bit 3 - digital input 4
bit 4 - digital input 5
bit 5 - digital input 6
etc.
bit 8 - digital output 1
bit 9 - digital output 2
bit 10 - digital output 3
bit 11 - digital output 4
bit 12 - digital output 5
etc.

PARSING THE PACKET

Following the LOGIN packet, each packet received will contain the one *packet header* + a number of *reports*. One packet may contain any number of *reports*, up to a maximum packet length of 1024 bytes. The *packet header* specifies the *packet length* and *number of reports in packet* to assist in parsing.

Each *report* may then contain any of the modules enabled in the *module mask*. The exact modules present are identified in the *report module mask* for each *report*. The *report module mask* is the key to correct parsing.

REPORT HEADER:	report sequence number	every report		integer	count	1	
	report module mask			bitfield		6	
	report (RTC) time			integer	julian sec	4	
	report reason flags			bitfield		4	
	report status flags			bitfield		2	
					module total:	**17**	

For extra guidance on parsing the protocol X reports, please ask for our example code, which is available on request by email to *support@astratelematics.com*

MODULE AVAILABILITY AND CONTENT

This document presents and overview of protocol X and guidance on using the detailed definition, which is provided by the accompanying excel file. Note that new modules may be added from time to time, hence please refer to the latest version of *Protocol X Description.xlsx* for details of modules, content and format.

CHECKSUM

Each packet ends in a 2 byte checksum (16 bit CCITT), which applies to all preceding bytes in the packet. Please refer to our example code for guidance on calculating the checksum.

TCP KEEPALIVE

Our TCP keepalive option allows a simple PING-PONG to be sent at a specified interval, which serves not simply to keep the socket alive (networks close idle sockets, typically after 10 minutes or so), but to make it clear when a socket is closed. Often we can get a socket open status from the GSM module, when its not actually open, and we only find out for sure when we try to send some data. Please refer to the Astra *Device Configuration Guide - Generic Command Reference* document for details of how to enable the TCP keepalive using the $TCPT command.

PING is defined as a single byte of value 02

PONG is defined as a single byte of value 03

When this option is enabled, the astra device will send a PING at the interval defined in $TCPT, and will expect a PONG in response. Failure to receive the expected response will result in closure of the (apparently) open socket and re-initiation of the connection.

Likewise, the platform server may send a PING to the astra at any time to verify the TCP socket connection, in which case, the astra will reply with PONG.

RECOMMENDED SUPPORT AND DIAGNOSTIC FEATURES

1. Command handling

The ability to send a generic command (user specified) to a device over the TCP socket and see the response from the device.

As above, but with the ability to select bulk (many) devices based on an IMEI list, device type, firmware version or group.

2. Alerts

The ability to anticipate issues in advance and handle them, possibly before the end-users even notice. Alerts will be a massive benefit in this, as we have seen over the years with platforms that have this capability, compared to clients / platforms that don't have alerts. Ideally, these alerts should trigger an email to specified at the time of the alert, but as a minimum, there should have a log file which is accessible to the support team.

As a basic, we suggest:

- Device not communicating – no data / connection / reports received for 2 x STIM minutes
- Device lost power – device with External Voltage zero (or < 6.0V) for 1 hour
- Missing data – missing data from a device based on MSN
- Device rebooted unexpectedly – RESTART report which does not follow FOTA or initial installation

The above will assist to ease support workload.

3. Data Content

The following fields of Protocol X data are essential for diagnostics and effective tech. support, and at the very least should be parsed out into the database to allow alerts and reports as diagnostics tools

REPORT HEADER	Reason	(Restart, External Power Event as a minimum)
	Status	(Ignition, GPS Invalid, Stored / live report)
DEVICE POWER	External Voltage	