



XT2500

Product Manual

Firmware Version: 1183DA2.4

Revised February 08, 2021

TABLE OF CONTENTS

1. Functional Description	4
1.1. Overview	4
1.2. Mechanical	4
2. Device Setup	5
2.1. Configure Device Via PC (RS232 to USB)	5
3. Interfaces	6
3.1. UART	6
4. Device Configuration	7
4.1. APNs (apn)	8
4.2. OBD Accel and Decel (oad)	9
4.3. OBD RPM and Speed Events (ors)	10
4.4. OTA Prevent (ota)	11
4.5. Bluetooth Enabled (blt)	12
4.6. Buzzer Songs (bzs)	13
4.7. Destination (dst)	13
4.8. Generic Config Value (gcv)	14
4.9. Input Default Polarity (idp)	15
4.10. Motion Sensitivity (mst)	16
4.11. Serial Port Settings (sps)	16
4.12. Timer Duration (tmr)	18
4.13. Packet Creation Recipe (pcr)	19
4.14. Third-Party Config Values (tpv)	19
4.15. GPS Solution Filter (gsf)	21
4.16. Vehicle Bus Configuration (vbs)	22
5. Interpreter	24
5.1. Language	24

5.2. Interpreter Scripting Trigger Block Examples36

5.3. Packet Recipe37

6. Command Language76

6.1. Overview76

6.2. Commands via UDP76

6.3. Verb and Module76

6.4. Object76

6.5. Arguments77

6.6. Device Command Modules79

6.7. California Proposition 65 Warning92

1. FUNCTIONAL DESCRIPTION

1.1. OVERVIEW

The XT2500 series is a plug-n-play universal vehicle tracking unit (VTU) for use in light duty/passenger vehicles. The XT2500 also suits clientele with large fleets of heavy duty trucks with 9-pin/6-pin Deutsch connectors (RP1226 for newer trucks) via a cable harness/adapter. the XT2500 utilizes a cellular modem, a Bluetooth modem, a GPS modem, and an integrated accelerometer to collect information about the vehicle for a variety of applications.

One key benefit of the XT2500 is the flexibility of the device, allowing it to suit an individual client’s desires. The device can be expanded by connecting to an external XT1065 I/O box. The XT1065 I/O box includes additional interfaces such as digital inputs, digital outputs, analog inputs, RS232 interface, and Dallas/Maxim 1-wire interface. The XT2500 is a versatile device that can include an optional Bluetooth interface for ELD applications and optional internal battery and buzzer for driver behavior alerts. The XT2500 devices can be controlled through various channels, ranging from simple system parameters to more complex device interpreter scripts, which offer maximum customization as they are written completely by the customer.

1.2. MECHANICAL



Mechanical	
Dimensions	2.63" x 1.49" x 1.06" (6.7 x 5.3 x 2.7 cm)
Weight	2 oz. (57 g)
Physical Connections	J1962 (with adapters available)
Operating Temperature	-25°C to 70°C (-13°F to 158°F)

2. DEVICE SETUP

2.1. CONFIGURE DEVICE VIA PC (RS232 TO USB)

1. Connect the device to a variable power supply.
2. Set the power supply to output average 12 volts and verify the device is drawing current after it has powered up (the device will automatically switch on when it receives power).
3. Open "Device Manager" on windows and click the drop-down next to "Ports (COM & LPT)".
4. Attach a USB to RS232 coverter (Ex. USB-RS232-0.0) cable to a port on the computer and pins 17 and 18 of the 24-pin connector.



NOTE: A "USB Serial Device" and the associated COM number should appear under "Ports". If nothing shows up, unplug the connection to the Device, flip it over, and plug it back in. Take note of the COM port number.

5. Open Secure CRT and double-click on the session with the correct COM port number. This should connect the device to the terminal.
6. Right-click the session and go to "Properties".
7. Ensure the following is set for the session:
 - a. Connection > Protocol > **Serial**
 - b. Connection > Serial > Baud rate > **115200**
 - c. Terminal > Check **Auto reconnect**
 - d. Terminal > Emulation > **Terminal [VT100]**
8. Click "Okay".



NOTE: The device should now be connected to the terminal and commands can be sent.

3. INTERFACES

3.1. UART

XT2500 devices include a 2-wire asynchronous serial interface (UART) operated through RS232 for communication with a host device. RS232 interface is TIA/EIA-232-F compliant and will accept 0-5V signaling. The UART interface is a 3.0V TTL interface conforming to the ITU-T V.24 recommendation, with CMOS compatible signal levels (0V for low data bit or ON state and 3.0V for high data bit or OFF state). If the

- The default baud rate is 115200 bits/s
- The default frame format is 8N1 (8 data bits, no parity, 1 stop bit)
- The interface does not support flow-control

The UART interface is active and requires no authentication by default. AES-128 challenge/response authentication can be enabled by enabling the console-authentication bit in the firmware flags setting of the device description.

4. DEVICE CONFIGURATION

System parameters are designed to be non-volatile and have the ability to be updated through a serial connection, USB, SMS, UDP and are used for many modules within the device. All system parameters (aka configs, params) have a default value that can always be restored in case the device ends up in an unknown state.

The following two commands will be most relevant when dealing with the following configurations:

- `:rycfg` -- Used to read a system configuration
- `:wycfg` -- Used to write a system configuration

When using either of these commands, you must specify what configuration slot you wish to read/write, otherwise the console will default to the first slot (index[0]). In the case that the desired parameter does not have a factory default setting (configs PCR, PCA, & GFN) the console will not default to the first slot unless it has already been written.

Example:

```
:wycfg tmr 30 0
:OK
```

```
:rycfg dst
| dst[0]: addr:"us.address.com", port:1234 (0x4d2)
: OK
```

4.1. APNs (APN)

The *APN* parameter contains settings for APN gateway, username, and password.

Configuration instances: 4

Command Type	Syntax
SET	Command :wycfg apn[x] "<apn_name>" "<username>" "<password>" Response :OK
READ	Command :rycfg apn[x] Response apn[x]: name:"<apn_name>", user:"<username>", passwd:"<password>" :OK

Parameter	Range	Description
apn_name	64 Char	The name of the APN gateway used for GSM, GPRS, 3G and 4G cellular networks. Examples of APN's are as follows:internet.t-mobile, wap.cingular, internet.mnc012.mcc345.gprs If instances 1 to 3 are configured as "disabled", the given slot will be skipped during APN hunting procedures. NOTE: For 4G AT&T modules, a blank APN configuration triggers a network-pushed APN.
username	32 Char	The username used to access the APN given by the first argument.
password	32 Char	The password used to access the APN given by the first argument.

4.2. OBD ACCEL AND DECEL (OAD)

Configuration instances: 2

Command Type	Syntax
SET	Command :wycfg oad[x] <rate> Response :OK
READ	Command :rycfg oad[x] Response oad[x]: accdec:<rate> (hex_val) :OK

Parameter	Descriptions		
rate	rate is measured in .1 mph		
	Instance	Range	Description
	oad[0]	10 to 200	OBD acceleration threshold.
			Speed rate of change to exceed on acceleration to declare Hard Acceleration Event
	oad[1]	10 to 200	OBD deceleration threshold.
		Speed rate of change to exceed on deceleration to declare Hard Deceleration Event	

4.3. OBD RPM AND SPEED EVENTS (ORS)

Configuration instances: 2

Command Type	Syntax
SET	Command :wycfg ors[x] <threshold> <time sec> Response :OK
READ	Command :rycfg ors[x] Response ors[x]: thresh:<threshold> (hex_val), time:<time sec> (hex_val) :OK

Instance	Description		
ors[0]	RPM Threshold		
	Parameter	Range	Description
	threshold	0 (disable), 100 to 20000	RPM in 1 rpm to exceed
	time sec	1 to 60	Time RPM must be greater than threshold to declare an RPM Exceeded Event
ors[1]	Speed Threshold		
	Parameter	Range	Description
	threshold	0 (disable), 20 to 150	Speed in 1 mph to exceed
	Time sec	1 to 60	Time Speed must be greater than threshold to declare a Speed Exceeded Event

4.4. OTA PREVENT (OTA)

Configuration instances: 1

Command Type	Syntax
SET	Command :wycfg ota <enable_mask> Response :OK
READ	Command :rycfg ota <enable_mask> Response ota: enable:<enable_mask> :OK

Parameter	Description	
enable_mask	Hex	Description
	0x0	OTA is not prevented due to OBD Communications state or TPS Transfer
	0x1	Prevent OTA when OBD Communications are Active (OBD Communications detected)
	0x2	Prevent OTA when TPS Transfer is Active (TPS Transfer is in progress)

4.5. BLUETOOTH ENABLED (BLT)

The *Bluetooth Enabled* configuration sets the Bluetooth communication state.

Configuration instances: 1

Command Type	Syntax
SET	Command :wycfg blt <enable> Response :OK
READ	Command :rycfg blt Response blt: en:<enable> (hex_val) :OK

Param	Description						
enable	Enables Bluetooth:						
	<table> <tr> <th>Value</th><th>Description</th></tr> <tr> <td>0</td><td>Disables bluetooth communications</td></tr> <tr> <td>1</td><td>Enables bluetooth communications</td></tr> </table>	Value	Description	0	Disables bluetooth communications	1	Enables bluetooth communications
Value	Description						
0	Disables bluetooth communications						
1	Enables bluetooth communications						

4.6. BUZZER SONGS (BZR)

The *Buzzer Songs* configuration sets the frequency and duration of up to 8 notes on the device buzzer.

Configuration instances: 10

Command Type	Syntax
SET	Command :wycfg bzs[x] <frequency1> <duration1> ... <frequency8> <duration8> Response :OK
READ	Command :rycfg bzs[x] Response bzs[x]: freq1:<frequency1> (hex_val), dur1:<duration1> (hex_val), ... freq8:<frequency8> (hex_val), dur8:<duration8> (hex_val) :OK

Param	Range	Description
frequencyX	0 to 65535	A frequency that corresponds to the Xth note to be played (Hz); 0 is silent.
durationX	0 (disable), 10-6535	The duration of the Xth note to be played (ms)

4.7. DESTINATION (DST)

The dst[9] instance is utilized for all Device Manager interactions, including device Checkin Messages; configure accordingly.

Configuration instances: 10

Command Type	Syntax
SET	Command :wycfg dst[x] "<ip> <hostname> <local serial>" <port_number> Response :OK
READ	Command :rycfg dst[x] Response dst[x]: addr:"<ip> <hostname> <local serial>" port:<port_number> (hex_val) :OK

Parameter	Range	Description
<ip> <hostname> <local serial>	64 Characters	This id is the location specified by the dst. It can take the form of an IP address (192.0.0.0), a hostname limited to 64 characters, or a local serial port. The valid local serial ports are as follows: "SERIAL_AUX0" "SERIAL_AUX1" "USB"
port_number	1 to 65533	If using an IP address or a hostname, can be used to specify the port on which to connect.

4.8. GENERIC CONFIG VALUE (GCV)

Configuration instances: 64

Command Type	Syntax
SET	Command :wycfg gcv[x] <unsigned_long> Response :OK
READ	Command :rycfg gcv[x] Response gcv[x]: value:<unsigned_long> (hex_val) :OK

Parameter	Range	Description
unsigned_long	0 to 4494967295	A generic configuration value accessible within a script.

4.9. INPUT DEFAULT POLARITY (IDP)

- idp[0] -> PUD on user input 1
- idp[1] -> PUD on user input 2
- idp[2] -> PUD on user input 3
- idp[3] -> PUD on user input 4

Configuration instances: 4

Command Type	Syntax
SET	Command :wycfg idp[x] <pullup_disable> Response :OK
READ	Command :rycfg idp[x] Response idp[x]: polrty:<pullup_disable> (hex_val) :OK

Parameter	Range	Description
Input active level	0 to 1	Sets the corresponding user input "active" level by engaging an external pullup ('0') or pulldown ('1') on corresponding that user input. An idp setting of '1' means the user input is configured active high, so the user input is pulled down when not driven by external influences. An idp setting of '0' means the user input is configured active low, so the user input is pulled up when not driven by external influences.

4.10. MOTION SENSITIVITY (MST)

Configuration instances: 1

Command Type	Syntax
SET	Command :wycfg mst <motion_sensitivity_value> Response :OK
READ	Command :rycfg mst Response mst: motion:<motion_sensitivity_value> (hex_val) :OK

Parameter	Range	Description
motion_sensitivity_value	1 to 2000	Configure accelerometer motion sensitivity in thousandths of a unit of gravity.

4.11. SERIAL PORT SETTINGS (SPS)

- sps[0] -> Aux port 0
- sps[1] -> Aux port 1

Configuration instances: 2

Command Type	Syntax
SET	Command :wycfg sps[x] <baudrate> <dps_value> <mode_value> <console_print> Response :OK
READ	Command :rycfg sps[x] Response sps[x]: baud:<baudrate> (hex_val), dps:<dps_value> (hex_val), mode:<mode_value> (hex_val), conprt:<console_print> (hex_val) :OK

Parameter	Range	Description								
baudrate	N/A	Acceptable baudrates: • 115200 bit/sec • 57600 bit/sec • 38400 bit/sec • 19200 bit/sec • 9600 bit/sec • 4800 bit/sec • 2400 bit/sec • 1200 bit/sec								
dps_value	3 Char	Acceptable dps values: 8N1								
Mode_value	N/A	Acceptable mode values: <table><tr><th>Value</th><th>Description</th></tr><tr><td>0</td><td>Console</td></tr><tr><td>3</td><td>Garmin</td></tr><tr><td>6</td><td>Aux Passthru -- Only one serial port can be an Aux Passthru at any given time.</td></tr></table>	Value	Description	0	Console	3	Garmin	6	Aux Passthru -- Only one serial port can be an Aux Passthru at any given time.
Value	Description									
0	Console									
3	Garmin									
6	Aux Passthru -- Only one serial port can be an Aux Passthru at any given time.									
Console_print	N/A	Scripted console print from BuildAndSend() format <table><tr><th>Value</th><th>Description</th></tr><tr><td>0</td><td>(ASCII)</td></tr><tr><td>1</td><td>(Binary)</td></tr></table>	Value	Description	0	(ASCII)	1	(Binary)		
Value	Description									
0	(ASCII)									
1	(Binary)									

4.12. TIMER DURATION (TMR)

Configuration instances: 32

Command Type	Syntax
SET	Command :wycfg tmr[x] <Duration> <auto_start> Response :OK
READ	Command :rycfg tmr[x] Response tmr[x]: sec:<Duration> (hex_val), flags:<auto_start> (hex_val) :OK

Parameter	Range	Description						
Duration	0 to 4294967295	The time that is set as the duration of the timer (s)						
auto_start	N/A	Determines if the timer is set to start automatically or needs to be manually started.						
		<table><tr><th>Index</th><th>Description</th></tr><tr><td>0</td><td>Disables autostart</td></tr><tr><td>1</td><td>Enables autostart</td></tr></table>	Index	Description	0	Disables autostart	1	Enables autostart
Index	Description							
0	Disables autostart							
1	Enables autostart							

4.13. PACKET CREATION RECIPE (PCR)

Configuration instances: 128

Command Type	Syntax
SET	Command :wycfg pcr[x] "<recipe_hex_string>" Response :OK
READ	Command :rycfg pcr[x] Response pcr[x]: resipe_hex_string:"<recipe_hex_string>" :OK



NOTE: This parameter does not have a factory default setting.

Parameter	Description
recipe_hex_string	Refer to Packet Creation Recipe for configuration details

4.14. THIRD-PARTY CONFIG VALUES (TPV)

Configuration instances: 1

Command Type	Syntax
SET	Command :wycfg tpv <dst_index> <checkin_freq> <pid_list_1> <pid_list_2> <Flag_value> Response :OK
READ	Command :rycfg tpv Response tpv: index:<dst_index> (hex_val), checkin:<checkin_freq> (hex_val), pidlst1:<pid_list1>, pidlst2:<pid_list2>, flag:<Flag_value> (hex_val) :OK

Parameter	Range	Description
dst_index	0 to 9	Destination config index that points to the Xirgo Gateway Server

Parameter	Range	Description																																				
checkin_freq	0 to 255	Time in hours for periodic check to gateway server																																				
Pid_list_1	0x0 to 0xFFFFFFFF	Bit list for supported pids 1 to 32																																				
		<table><tr><th>Hex</th><th>Description</th></tr><tr><td>0x8000000</td><td>Odometer (miles) (ppid 1 0x74)</td></tr><tr><td>0x4000000</td><td>Oil Life (%) (ppid 2 0x75)</td></tr><tr><td>0x2000000</td><td>TPMS LF (psi) (ppid 3 0x76)</td></tr><tr><td>0x1000000</td><td>TPMS RF (psi) (ppid 4 0x77)</td></tr><tr><td>0x0800000</td><td>TPMS LR (psi) (ppid 5 0x78)</td></tr><tr><td>0x0400000</td><td>TPMS RR (psi) (ppid 6 0x8E)</td></tr><tr><td>0x0200000</td><td>TPMS LRI (psi) (ppid 7 0x90)</td></tr><tr><td>0x0100000</td><td>TPMS RRI (psi) (ppid 8 0x90)</td></tr><tr><td>0x0080000</td><td>TPMS SPR (psi) (ppid 9 0x91)</td></tr><tr><td>0x0040000</td><td>Seatbelt D (ascii) (ppid 10 0x92)</td></tr><tr><td>0x0020000</td><td>Seatbelt P (ascii) (ppid 11 0x9c)</td></tr><tr><td>0x0010000</td><td>Fuel Level (%) (0x5A)</td></tr><tr><td>0x0008000</td><td>Airbag Lamp (ascii) (ppid 12 0x9d)</td></tr><tr><td>0x0004000</td><td>PRNDL (ascii) (ppid 13 0x9e)</td></tr><tr><td>0x0002000</td><td>Parking Brake State (ascii) (ppid 14 0x9F)</td></tr><tr><td>0x0001000</td><td>Parking Brake Lamp (ascii) (ppid 15 0xA0)</td></tr><tr><td>0x0000800</td><td>0x000001 (reserved)</td></tr></table>	Hex	Description	0x8000000	Odometer (miles) (ppid 1 0x74)	0x4000000	Oil Life (%) (ppid 2 0x75)	0x2000000	TPMS LF (psi) (ppid 3 0x76)	0x1000000	TPMS RF (psi) (ppid 4 0x77)	0x0800000	TPMS LR (psi) (ppid 5 0x78)	0x0400000	TPMS RR (psi) (ppid 6 0x8E)	0x0200000	TPMS LRI (psi) (ppid 7 0x90)	0x0100000	TPMS RRI (psi) (ppid 8 0x90)	0x0080000	TPMS SPR (psi) (ppid 9 0x91)	0x0040000	Seatbelt D (ascii) (ppid 10 0x92)	0x0020000	Seatbelt P (ascii) (ppid 11 0x9c)	0x0010000	Fuel Level (%) (0x5A)	0x0008000	Airbag Lamp (ascii) (ppid 12 0x9d)	0x0004000	PRNDL (ascii) (ppid 13 0x9e)	0x0002000	Parking Brake State (ascii) (ppid 14 0x9F)	0x0001000	Parking Brake Lamp (ascii) (ppid 15 0xA0)	0x0000800	0x000001 (reserved)
Hex	Description																																					
0x8000000	Odometer (miles) (ppid 1 0x74)																																					
0x4000000	Oil Life (%) (ppid 2 0x75)																																					
0x2000000	TPMS LF (psi) (ppid 3 0x76)																																					
0x1000000	TPMS RF (psi) (ppid 4 0x77)																																					
0x0800000	TPMS LR (psi) (ppid 5 0x78)																																					
0x0400000	TPMS RR (psi) (ppid 6 0x8E)																																					
0x0200000	TPMS LRI (psi) (ppid 7 0x90)																																					
0x0100000	TPMS RRI (psi) (ppid 8 0x90)																																					
0x0080000	TPMS SPR (psi) (ppid 9 0x91)																																					
0x0040000	Seatbelt D (ascii) (ppid 10 0x92)																																					
0x0020000	Seatbelt P (ascii) (ppid 11 0x9c)																																					
0x0010000	Fuel Level (%) (0x5A)																																					
0x0008000	Airbag Lamp (ascii) (ppid 12 0x9d)																																					
0x0004000	PRNDL (ascii) (ppid 13 0x9e)																																					
0x0002000	Parking Brake State (ascii) (ppid 14 0x9F)																																					
0x0001000	Parking Brake Lamp (ascii) (ppid 15 0xA0)																																					
0x0000800	0x000001 (reserved)																																					
Pid_list_2	0x0 to 0xFFFFFFFF	Bit list for supported PIDs (32 to 32) <reserved>																																				
Flag	0 to 255	Reserved																																				

4.15. GPS SOLUTION FILTER (GSF)

Configuration instances: 1

Command Type	Syntax
SET	Command :wycfg gsf <year> <DOP> <HACC> <speed_mask> Response :OK
READ	Command :rycfg gsf Response gsf: year:<year> (hex_val), HDOP:<DOP> (hex_val), HACC:<HACC> (hex_val), mask:<speed_mask> (hex_val) :OK

Parameter	Range	Description										
year	1000 - 5000	Any solution with a reported year before this will be filtered out										
DOP	0 to 200	Any solution with an HDOP (or PDOP on some systems) above this value will be filtered out. Units 0.1DOP. “20” means 2.0										
HACC	0 to 255	Horizontal accuracy. Not applicable (ignored) for XT2500										
speed_mask	0x0 to 0xFFFFFFFF	speed filter mask allows for filtering speed (clamps to 0) under various conditions										
<table><tr><th>Hex</th><th>Description</th></tr><tr><td>0x01</td><td>QUALITY: If speed < 10MPH and num_sats < 7, speed is clamped to 0. If speed < 5MPH and num_sats < 9, speed is clamped to 0.</td></tr><tr><td>0x02</td><td>MOTION: If motion state is stopped then the speed is clamped to 0</td></tr><tr><td>0x04</td><td>MOVEMENT: If movement state is stopped then the speed is clamped to 0</td></tr><tr><td>0x08</td><td>IGNITION: If ignition state is off then the speed is clamped to 0</td></tr></table>			Hex	Description	0x01	QUALITY: If speed < 10MPH and num_sats < 7, speed is clamped to 0. If speed < 5MPH and num_sats < 9, speed is clamped to 0.	0x02	MOTION: If motion state is stopped then the speed is clamped to 0	0x04	MOVEMENT: If movement state is stopped then the speed is clamped to 0	0x08	IGNITION: If ignition state is off then the speed is clamped to 0
Hex	Description											
0x01	QUALITY: If speed < 10MPH and num_sats < 7, speed is clamped to 0. If speed < 5MPH and num_sats < 9, speed is clamped to 0.											
0x02	MOTION: If motion state is stopped then the speed is clamped to 0											
0x04	MOVEMENT: If movement state is stopped then the speed is clamped to 0											
0x08	IGNITION: If ignition state is off then the speed is clamped to 0											

4.16. VEHICLE BUS CONFIGURATION (VBS)

Configuration instances: 1

Command Type	Syntax
SET	Command :wycfg vbs <detect_mask> <start_mask> <mode> Response :OK
READ	Command :rycfg vbs Response gsfc: detect:<detect_mask> (hex_val), start:<start_mask> (hex_val), mode:<mode> (hex_val) :OK

Parameter	Range	Description																		
detect_mask	0x0 to 0xFF	8 bits, hex mask. (Default: 0xFF)																		
		<table><tr><th>Hex</th><th>Description</th></tr><tr><td>0x01</td><td>Query Vin first time ignition on a vehicle</td></tr><tr><td>0x02</td><td>N/A</td></tr><tr><td>0x04</td><td>Big hammer trip end.</td></tr></table>	Hex	Description	0x01	Query Vin first time ignition on a vehicle	0x02	N/A	0x04	Big hammer trip end.										
		Hex	Description																	
		0x01	Query Vin first time ignition on a vehicle																	
		0x02	N/A																	
0x04	Big hammer trip end.																			
start_mask	0x0 to 0xFFFF	16 bits, hex mask. (Default: 0xFFFF)																		
		<table><tr><th>Hex</th><th>Description</th></tr><tr><td>0x0001</td><td>Query Fuel level on trip start</td></tr><tr><td>0x0002</td><td>Query DTC on trip start</td></tr><tr><td>0x0004</td><td>N/A</td></tr><tr><td>0x0008</td><td>OBd-II backoff (Listens for obd-II requests that are coming from external sources)</td></tr><tr><td>0x0010</td><td>Proprietary backoff (Listens for obd-II requests that are coming from proprietary sources)</td></tr><tr><td>0x0020</td><td>Diagsession backoff (Listens for diagnostic session start requests)</td></tr><tr><td>0x0040</td><td>TP backoff (Listens for diagnostic tester present or “keep alive” requests)</td></tr><tr><td>0x0080</td><td>Security backoff (Listens for secure diagnostic session start request)</td></tr></table>	Hex	Description	0x0001	Query Fuel level on trip start	0x0002	Query DTC on trip start	0x0004	N/A	0x0008	OBd-II backoff (Listens for obd-II requests that are coming from external sources)	0x0010	Proprietary backoff (Listens for obd-II requests that are coming from proprietary sources)	0x0020	Diagsession backoff (Listens for diagnostic session start requests)	0x0040	TP backoff (Listens for diagnostic tester present or “keep alive” requests)	0x0080	Security backoff (Listens for secure diagnostic session start request)
		Hex	Description																	
		0x0001	Query Fuel level on trip start																	
		0x0002	Query DTC on trip start																	
		0x0004	N/A																	
		0x0008	OBd-II backoff (Listens for obd-II requests that are coming from external sources)																	
		0x0010	Proprietary backoff (Listens for obd-II requests that are coming from proprietary sources)																	
		0x0020	Diagsession backoff (Listens for diagnostic session start requests)																	
		0x0040	TP backoff (Listens for diagnostic tester present or “keep alive” requests)																	
0x0080	Security backoff (Listens for secure diagnostic session start request)																			
mode	0 to 255	See table below for what values to set.																		

VBUS Mode Value	Description	Use Case
0	OBD-II	Use this mode when you are sure you will be connected to a passenger vehicle. Supported transports per commlink: Commlink0: J1850, Kline, HS Can
1	PPID Mode	Use this mode when you are sure you will be connected to a passenger vehicle, and want to enable proprietary PID collection. Supported transports per commlink: Commlink0: HS Can Commlink1: MS Can, SW Can Commlink2: Kline
2	Heavy Duty Mode	Use this mode when you are sure you will be connected to a heavy-duty truck. Supported transports per commlink: Commlink0: J1939, HS Can (Vin collection) Commlink1: J1939, J1708
3	J1708 Only Mode	Use this mode when you are sure you will be connected to a heavy-duty truck with the 6 pin adapter cable. Supported transports per commlink: Commlink0: J1708
129	Auto-Detect Mode	Use this mode when you are not sure if you will be connected to a heavy-duty truck or a passenger vehicle. Supported transports per commlink: Commlink0: J1850, Kline, HS Can, J1939 Commlink1: J1939, J1708
255	Disable	To disable all OBD comms

5. INTERPRETER

5.1. LANGUAGE

5.1.1. Interpreter Language Basics

Overview

Certain Xirgo products contain functionality to run user-created scripts. This allows functionality to be tuned and changed without needing Firmware updates. The device runs an interpreter which ‘interprets’ byte code that has been compiled from a human readable script. This document describes the language/syntax used for this script.

Byte-Code Characteristics

Blocks	Description
Script	Each Script Block can contain unlimited Trigger Blocks (limited only by memory reserved for byte code on device itself).
Trigger	Each Trigger Block has one Trigger Block Test. A test is usually made of one comparison, but the language also allows to use more than one comparison in which the test is true if ANY of the comparisons are true or if ALL the comparisons are true. A Trigger Block Test may contain up to 2 comparisons. NOTE: Each Trigger Block can contain unlimited Conditional Action Blocks.
Conditional Action	Each Conditional Action Block may optionally contain one Conditional Action Block Test. A Conditional Action Block Test may contain up to 5 comparisons (Using same “any” or “all” logic as described for Trigger Block Test). NOTE: Each Conditional Action Block must contain one Action Block.
Action	Each Action Block can contain 1 – 3 Actions

How Trigger Block Tests Work

Each Trigger Block is entered when the Trigger Block Test becomes true. The Trigger Block is not entered WHILE the test is true, only the moment it becomes true.

Example: If you want to trigger actions when vehicle speed goes above 80 KPH:

```
trigger when Gt(GPSSpeed, 800) [Debounce(0, 0)]
...
```

The interpreter will enter the trigger block the moment the vehicle speed increases above 80 KPH. It will not continue to enter the trigger block during subsequent evaluations where the speed remains above 80 KPH. Once the speed drops below 80 (for at least one evaluation), then the trigger block will be entered again next time the speed increases above 80 KPH.



NOTE: if the speed oscillates between 80.0 and 80.1 kmh it is possible to cause the actions to be executed as frequently as the speed oscillates. In order to avoid this, make use of the debounce specifiers.

How Conditional Action Block Tests Work

Unlike Trigger Block Tests, Conditional Action Block Tests allow the action to be performed WHILE the test is true. Continuing with the example above, let's say we want to further limit our actions to only execute the moment speed goes above 80 kmh AND the vehicle heading is within 10 degrees of North:

```
trigger when Gt(GPSSpeed, 800) [Debounce(0, 0)]
    conduct any InRange(GPSHeading, 3500, 3600) [Debounce(0, 0)]
                InRange(GPSHeading, 0, 100) [Debounce(0, 0)]
    actions
        ...
```

5.1.2. Compiling scripts with ILC

1. Type the script (as noted above) with your text editor and save as a text file with .txt file extension.
2. *Recommended: Put both ilc.exe and your script <name>.txt in the same folder.*
3. Start your preferred command line terminal (cmd.exe or powershell).
4. Change directory to the folder where your script is located.

Example command prompt:

```
cd USERPROFILE\Documents\XT6300\Scripts for cmd.exe or cd
env:USERPROFILE\Documents\XT6300\Scripts for powershell
```

5. Run ilc.exe from the command line: ilc.exe <name>.txt (If "Interpreter script format OK" appears at the end, your script was successfully compiled).

Example Command Prompt

```
0x40 SMALL_CONST_VALUE (value: 0)
TOK_NUMBER: "22"
0x56 SMALL_CONST_VALUE (value: 22)
TOK_NUMBER: "0"
0x40 SMALL_CONST_VALUE (value: 0)
```

```
TOK_NUMBER: "2"
0x42 SMALL_CONST_VALUE (value: 2)
0xe0 INTERP_END

binary script size: 134
file size: 142
file name: network_testing_script.bin
Interpreter script format OK
ilc v2.16
```



NOTE: Using Windows to drag the .txt file onto the ilc.exe as a means of execution will always result in a .bin file being generated, even if the compiler failed. It is recommended instead to always run ilc.exe from a command line interface.

5.1.3. Upload and Run Scripts

1. Connect to the device using standard USB-to-Serial or USB-to-RS232 cabling.
2. Open the terminal emulator program (ZOC or similar) and connect to active COM port.
3. Ensure connectivity by issuing the command :qti.
4. Issue command to start script upload: :grscr x <port_index> where *port_index* is one of the following:

Index	Description
0	USB Port
1	RS232 1
2	RS232 2

5. Upload your script file (<name>.bin) using your terminal emulator program via Xmodem.



NOTE: your compiled script must be less than 16 KB in order to be loaded onto the device.

```
:OK
:grscr x 0
:OK
Erasing FLASH memory
Waiting for Xmodem Start (Ctrl-D twice to cancel)

Starting xmodem transfer. Press Ctrl+C to cancel.
Transferring V_Script_Nov_102015_JB.bin...
  100%    572 bytes  572 bytes/sec 00:00:01      0 Errors

Interpreter Script Download Successful
Interpreter Script Stored in ChipFlash
```

5.1.4. Interpreter Commands

All language elements presented in this section are case sensitive.

Definitions

Keyword	Definition
Comments	Comments start with the pound (#) character and continue until the end of the line.
Numbers	Numbers can be specified in decimal (e.g., 0, 1, 100, 256, -359) or in hex when the value is non-negative (e.g., 0x0, 0x1, 0x64, 0x100)
Debounce Specify	Debounce(<hi>, <lo>) - where hi and lo are specified in seconds (max 15). The 'Debounce Specify' element is only used following certain tests (see below). When a debounce is specified, it means that a test is true only AFTER the comparison is true for hi seconds and false AFTER the comparison is false for lo seconds.



NOTE: NOTE: A test debounce MUST be specified any time the first argument in the test is either a system value or a special function; it's the script designer's responsibility to input appropriate values for hi and lo.

Keywords

Keyword	Description
trigger	Denotes the start of a Trigger Block
conduct	Denotes the start of a conditional Action Block
when	Denotes the start of a test when only one comparison is used
any	Use when grouping multiple comparisons; test is true when ANY of the comparisons are true
all	Use when grouping multiple comparisons; ALL comparisons must be true for test to be true
always	Use when there are no tests for a Conditional Action Block. The actions inside a Conditional Action Block will ALWAYS run when the Trigger Block's test(s) are true.

Keyword	Description
actions	Denotes the start of an Action Block
run	Denotes the start of a single Action

Transforms

Script Language	Description
BitNot(<a>)	Bitwise NOT (one's compliment) of argument a
BitAnd(<a>,)	Bitwise AND of arguments a and b
BitOr(<a>,)	Bitwise OR of arguments a and b
BitXor(<a>,)	Bitwise XOR of arguments a and b
BitShift(<a>,)	Shift the contents of argument a by the value of argument b; i.e., if b is negative the shift is Left, and if B is positive the shift is Right
Add(<a>,)	Arithmetic addition of arguments a and b
Subtract(<a>,)	Arithmetic subtraction of argument b from a
Multiply(<a>,)	Arithmetic multiplication of arguments a and b

Tests

Script Language	Description
InRange(<a>, , <c>)	True when argument a is between argument b and argument c (inclusive)
NinRange(<a>, , <c>)	True when a is less than b or a is greater than c; b must be less than a
Eq(<a>,)	True when a equals b
Neq(<a>,)	True when a is not equal to b
Gt(<a>,)	True when a is greater than b
Lt(<a>,)	True when a is less than b
GtEq(<a>,)	True when a is greater than/equal to b
LtEq(<a>,)	True when a is less than/equal to b

Test items <a>, , and <c> below should be one of the following:

- System Value
- Event
- Special Function
- Numbers (constants)

System Values

Script Language	Description
UnixTime	Maps directly to UnixTime packet field (ID: 0x06)
GPSLat	Maps directly to Latitude packet field (ID: 0x07)
GPSLon	Maps directly to Longitude packet field (ID: 0x08)
GPSAlt	Maps directly to Altitude packet field (ID: 0x09)
GPSHeading	Maps directly to Heading packet field (ID: 0x0a)
GPSSpeed	Maps directly to GpsSpeed packet field (ID: 0x0b)
Inputs	Maps directly to InputStates packet field (ID: 0x0c)
Outputs	Maps directly to OutputStates packet field (ID: 0x0d)
GSPTripOdom1	Maps directly to GpsTripOdom1 packet field (ID: 0x10)
OdomDelta	
GPSHDOP	Maps directly to Hdop packet field (ID: 0x12)
GPSNumSats	Maps directly to NumSats packet field (ID: 0x13)
InternVoltage	Maps directly to InternalBattVolts packet field (ID: 0x16)
BattVoltage	Maps directly to VehicleBattVolts_1byte packet field (ID: 0x17)
GPSLifetimeOdom	GPSOdom Maps directly to GpsLifetimeOdom packet field (ID: 0x18)
GPSOdom	
OBDLifetimeOdom	Maps directly to ObdLifetimeOdom packet field (ID: 0x2f)
OdomDiff	
WakeReason	Maps directly to WakeReason packet field (ID: 0x53) (see Wake Reason Mask for mask definition)

Events

Script Language	Description
TimerExpired(<index>)	Evaluates to 1 when timer is expired; otherwise 0 <i>Range: 0 to 31</i>
UserEventIsActive(<index>)	Evaluates to 1 when a user event has been injected into interpreter; otherwise 0. Use :xrmsg <index> to inject a message into interpreter . <i>Range: 0 to 255</i>
AccelEventIsActive(<index>)	Evaluates to 1 when an accelerometer event is detected; otherwise 0 <i>Range: 0 to 7</i>
DriverIdRead(<index>)	Evaluates to 1 when driver id is detected on 1-wire channel <index>, otherwise 0 <i>Range: 0 to 1</i>
SystemEventIsActive(<index>)	Evaluates to 1 when system event <index> is active, otherwise 0 <i>Range: 0 to 20</i>

Script Language	Description	
	Value	Description
	0	Reset
	1	Wakeup
	2	BootloaderUpgrade
	3	FWUpgrade
	4	ScriptUpgrade
	5	ParamSetUpgrade
	6	OverlayUpgrade (not implemented)
	7	ManualConfigChange
	8	CellRegistrationChange
	9	IPChange
	10	SMSReceived
	11	SMSSendOK
	12	SMSSendFailure
	13	UDPReceived
	14	UDPSendOK
	15	UDPSendFailure
	16	PacketStorageEmpty
	17	PacketStorageFull
	18	PdpConnectionReset (not implemented)
	19	Device Powerup
	20	OTA Reject (not implemented)
ObdDtcEventIsActive(0)	Evaluates to 1 when an OBD DTC event is detected, otherwise 0.	
AccelCrashEventIsActive(0)	Evaluates to 1 when an accelerometer crash event is detected, otherwise 0.	

Special Functions

Script Language	Description
SystemVal(<packet field id>)	Evaluates to value stored in packet field Packet field ID: See Packet Recipes for available IDs
UserVar16(<index>)	Evaluates to value stored in 16bit user variable <i>Index range: 0 to 15</i>
UserVar32(<index>)	Evaluates to value stored in 32bit user variable <i>Index Range: 0 to 15</i>
FlagsSet(<index>)	Evaluates to 1 when flag is set, otherwise 0 <i>Index Range: 0 to 31</i>

Script Language	Description																																				
GeofenceState(<index>)	<table> <tr> <th>Index</th><th>Description</th></tr> <tr> <td>-1</td><td><index> is not configured</td></tr> <tr> <td>0</td><td>Outside fence</td></tr> <tr> <td>1</td><td>Inside fence</td></tr> </table> <i>Index range: 0 to 49</i>	Index	Description	-1	<index> is not configured	0	Outside fence	1	Inside fence																												
Index	Description																																				
-1	<index> is not configured																																				
0	Outside fence																																				
1	Inside fence																																				
SystemState(<index>)	Evaluates to 1 when true, otherwise 0 <i>Index range: 0 to 16</i> <table> <tr> <th>Index</th><th>Description</th></tr> <tr> <td>0</td><td>Valid Registration</td></tr> <tr> <td>1</td><td>Valid IP</td></tr> <tr> <td>2</td><td>Valid GPS</td></tr> <tr> <td>3</td><td>Valid Script</td></tr> <tr> <td>4</td><td>DM Session Active</td></tr> <tr> <td>5</td><td>Accel Oriented</td></tr> <tr> <td>6</td><td>Bluetooth Discoverable</td></tr> <tr> <td>7</td><td>Bluetooth Paired</td></tr> <tr> <td>8</td><td>(Not implemented)</td></tr> <tr> <td>9</td><td>Bluetooth Connected</td></tr> <tr> <td>10</td><td>Bluetooth Powered</td></tr> <tr> <td>11</td><td>Cellular Powered</td></tr> <tr> <td>12</td><td>GPS Powered</td></tr> <tr> <td>13</td><td>Garmin Powered</td></tr> <tr> <td>14</td><td>Motion Via Accel</td></tr> <tr> <td>15</td><td>Motion Via GPS</td></tr> <tr> <td>16</td><td>External Power</td></tr> </table>	Index	Description	0	Valid Registration	1	Valid IP	2	Valid GPS	3	Valid Script	4	DM Session Active	5	Accel Oriented	6	Bluetooth Discoverable	7	Bluetooth Paired	8	(Not implemented)	9	Bluetooth Connected	10	Bluetooth Powered	11	Cellular Powered	12	GPS Powered	13	Garmin Powered	14	Motion Via Accel	15	Motion Via GPS	16	External Power
Index	Description																																				
0	Valid Registration																																				
1	Valid IP																																				
2	Valid GPS																																				
3	Valid Script																																				
4	DM Session Active																																				
5	Accel Oriented																																				
6	Bluetooth Discoverable																																				
7	Bluetooth Paired																																				
8	(Not implemented)																																				
9	Bluetooth Connected																																				
10	Bluetooth Powered																																				
11	Cellular Powered																																				
12	GPS Powered																																				
13	Garmin Powered																																				
14	Motion Via Accel																																				
15	Motion Via GPS																																				
16	External Power																																				
GenericConfig(<index>)	Evaluates to value stored in gcv[<index>] config parameters. <i>Index range: 0 to 63</i>																																				
PacketsPending(<index>)	Evaluates to number of unsent packets stored in the log <table> <tr> <th>Index</th><th>Description</th></tr> <tr> <td>0</td><td>Normal Packet Space</td></tr> <tr> <td>1</td><td>UDR Packet Space</td></tr> </table>	Index	Description	0	Normal Packet Space	1	UDR Packet Space																														
Index	Description																																				
0	Normal Packet Space																																				
1	UDR Packet Space																																				
NoAckRxRetryCount(0)	Evaluates to number of retried failed ACK packets																																				
AuxPassthruScratchpadSize(0)	Evaluates to aux passthru scratchpad size (in bytes).																																				
AuxPassthruScratchpadByte(<index>)	Evaluates to the byte value at index of the aux passthru scratchpad.																																				

Actions

Script Language	Description		
ResetDevice()	N/A		
ResetModem()	N/A		
ResetGPS()	N/A		
AdjustUserVar (<type>, <index>, <adjust_amount>)	Type range: 0 to 2		
	Type	Description	
	0	8bit variable	
	1	16bit variable	
	2	32bit variable	
	Index range:		
	Index	Description	
	0 to 31	8bit variables	
	0 to 15	16bit variables	
	0 to 15	32bit variables	
	Adjust_amount range:		
	Adjust Amount	Description	
	-128 to 127	8bit variables	
	-32768 to 32767	16bit variables	
	-2147483648 to 2147483647	32bit variables	
	NOTE: user variables are 'clamped' at min and max. Meaning no matter the size of adjustment the value will never go lower than the minimum or higher than the maximum (it will not roll over).		
SetFlag(<flag_index>)	Index range: 0 to 31		
ClearFlag(<flag_index>)	Index range: 0 to 31		
StartTimer(<timer_index>)	Index range: 0 to 31		
StopTimer(<timer_index>)	Index range: 0 to 31		
ResetTimer(<timer_index>)	Index range: 0 to 31		
BuildAndSendMsg (<packet_id>, <reason_code>, <destination_id>, <ack>)	Parameter	Range	Description
	Packet_id	0 to 254	Should be an ID configured with :wycfg pcr[x]
	reason_code	0 to 255	Assigns the reason_code to the index in the script. Any reason_code less than 0 or greater than 255 will wrap.
	destination_id	0 to 9	Should be one of the destinations configured with :wycfg dst[x] ...

Script Language	Description																						
	<table><tr><th>Parameter</th><th>Range</th><th>Description</th></tr><tr><td>Ack range</td><td>0 to 2</td><td>See Ack table below</td></tr></table> Ack table: <table><tr><th>Range</th><th>Description</th></tr><tr><td>0</td><td>No acknowledgement needed</td></tr><tr><td>1</td><td>Resend until acknowledged</td></tr><tr><td>2</td><td>Priority packet: no storage, no acknowledgement</td></tr></table> See footnotes 1-3 below table.	Parameter	Range	Description	Ack range	0 to 2	See Ack table below	Range	Description	0	No acknowledgement needed	1	Resend until acknowledged	2	Priority packet: no storage, no acknowledgement								
Parameter	Range	Description																					
Ack range	0 to 2	See Ack table below																					
Range	Description																						
0	No acknowledgement needed																						
1	Resend until acknowledged																						
2	Priority packet: no storage, no acknowledgement																						
EnterSleep(<wake_mask>, <wake_minutes>)	Set the wake reason mask <table><tr><th>Value (Hex)</th><th>Wake Reason</th></tr><tr><td>0x0000</td><td>SMS or UDP command received</td></tr><tr><td>0x0001</td><td>Input 0 (Ignition) transitioned high</td></tr><tr><td>0x0002</td><td>Input 1 transitioned high</td></tr><tr><td>0x0004</td><td>Input 2 transitioned high</td></tr><tr><td>0x0008</td><td>Input 3 transitioned high</td></tr><tr><td>0x0010</td><td>Input 4 transitioned high</td></tr><tr><td>0x0020</td><td>Configured sleep timeout expired</td></tr><tr><td>0x0040</td><td>Vibration detected</td></tr><tr><td>0x0080</td><td>Main supply voltage exceeds configured threshold value</td></tr><tr><td>0x0100</td><td>Main supply voltage fell below 6.0VDC (device unplugged)</td></tr></table>	Value (Hex)	Wake Reason	0x0000	SMS or UDP command received	0x0001	Input 0 (Ignition) transitioned high	0x0002	Input 1 transitioned high	0x0004	Input 2 transitioned high	0x0008	Input 3 transitioned high	0x0010	Input 4 transitioned high	0x0020	Configured sleep timeout expired	0x0040	Vibration detected	0x0080	Main supply voltage exceeds configured threshold value	0x0100	Main supply voltage fell below 6.0VDC (device unplugged)
Value (Hex)	Wake Reason																						
0x0000	SMS or UDP command received																						
0x0001	Input 0 (Ignition) transitioned high																						
0x0002	Input 1 transitioned high																						
0x0004	Input 2 transitioned high																						
0x0008	Input 3 transitioned high																						
0x0010	Input 4 transitioned high																						
0x0020	Configured sleep timeout expired																						
0x0040	Vibration detected																						
0x0080	Main supply voltage exceeds configured threshold value																						
0x0100	Main supply voltage fell below 6.0VDC (device unplugged)																						
SendGarminMsg(<msg_id>, <flags>)	<table><tr><th>Range</th><th>Description</th></tr><tr><td>1</td><td>Unit ID Request (Garmin ESN Request)</td></tr><tr><td>2</td><td>Product ID Request</td></tr><tr><td>4</td><td>Ping</td></tr><tr><td>5</td><td>FMI Enable</td></tr><tr><td>6</td><td>Set Auto Arrival to defaults (30 sec, 100 meters)</td></tr><tr><td>7</td><td>Disable Status Message</td></tr><tr><td>8</td><td>Disable ETA Message</td></tr><tr><td>9</td><td>Send user-defined payload (should be triggered with GarminUserPayloadReceived(0))</td></tr></table> Flags range: 0 to 1 <table><tr><th>Range</th><th>Description</th></tr><tr><td>1</td><td>Save ACK/NAK</td></tr></table>	Range	Description	1	Unit ID Request (Garmin ESN Request)	2	Product ID Request	4	Ping	5	FMI Enable	6	Set Auto Arrival to defaults (30 sec, 100 meters)	7	Disable Status Message	8	Disable ETA Message	9	Send user-defined payload (should be triggered with GarminUserPayloadReceived(0))	Range	Description	1	Save ACK/NAK
Range	Description																						
1	Unit ID Request (Garmin ESN Request)																						
2	Product ID Request																						
4	Ping																						
5	FMI Enable																						
6	Set Auto Arrival to defaults (30 sec, 100 meters)																						
7	Disable Status Message																						
8	Disable ETA Message																						
9	Send user-defined payload (should be triggered with GarminUserPayloadReceived(0))																						
Range	Description																						
1	Save ACK/NAK																						
SetOutput(<output_	Index range: 0 to 2																						

Script Language	Description
index>)	
ClearOutput (<output_index>)	<i>Index range: 0 to 2</i>
PulseOutput (<output_index>, <seconds_on>)	<i>Index range: 0 to 2</i> <i>seconds_on range: 1 to 65535</i>
FlashOutput (<output_index>, <blink_rate>)	<i>Index range: 0 to 2</i> <i>Blink_rate range: 1 to 100 Hz</i>
ClearDriverIds()	N/A
CheckInNow()	N/A
BuzzerPlay(<song_ index> <volume> <loops>)	<i>Song_index range: 0 to 9</i> <i>Volume range: 0 to 100</i> <i>Loops clamped to range: 0 to 255; 0 is continuous play</i>
BuzzerStop()	N/A
ResetObdHarshAccel Cnt()	N/A
ResetObdHarshBrake Cnt()	N/A
AuxPassthruScratchp adLoad()	Load the Aux Passthru scratchpad with the most recently-received packet

Actions Table Footnotes

1. *BuildAndSendMsg* example: using command `:wycfg pcr[0] "01050104030708"`. The item index 0 in `pcr[0]` is NOT the `packet_id`. This should be considered the `slot_id` (of which there are only 128). The `packet_id` in this example is the first hex byte of the hex string "01", meaning `packet_id` = 1. The parameter set may be coordinated so the recipe for `packet_id` x is stored at `slot_id` x. However, this limits your `packet_id` range to 0 - 127.
2. *BuildAndSendMsg* Coordinated `slot_id` and `packet_id` examples:
 - `:wycfg pcr[0] "00050104020708"`
 - `:wycfg pcr[1] "01050104020708"`
 - ...
 - `:wycfg pcr[127] "7f050104020708"`
3. *BuildAndSendMsg* recommendation: Due to the design of the entire XT6300 system, it is recommended customers do not exceed a message generation rate of one every five seconds for long durations (one

every ten seconds for CATm devices). The throughput of the cellular design is the bottleneck. Messages will always be stored to flash and sent out eventually, unless the duration at such a rate causes the message storage space to be filled completely. If a customer is looking for continuous real time data, these are the limitations they should be made aware of.

4. *SaveDataToFlash()* *Note:* Persist data saves normally happen automatically every 60 seconds. This command is intended to augment that. This command has a progressive spam filter that does not allow the command to be executed in quick procession. The required time to wait between saves is one second, doubling with every proceeding save; it will not extend above 60 seconds. 60 seconds after the last successful save this is reduced back to 1 second. Calling this function while the spam filter is active does not perform the save.

5.1.5. Wake Reason Masks

Script Language	Description																						
EnterDeepSleep (<wake_mask>, <wake_minutes>)	Set the wake reason mask <table> <tr> <th>Value (Hex)</th><th>Wake Reason</th></tr> <tr> <td>0x0001</td><td>Input 0 (Ignition) transitioned high</td></tr> <tr> <td>0x0002</td><td>Input 1 transitioned high</td></tr> <tr> <td>0x0004</td><td>Input 2 transitioned high</td></tr> <tr> <td>0x0008</td><td>Input 3 transitioned high</td></tr> <tr> <td>0x0010</td><td>Input 4 transitioned high</td></tr> <tr> <td>0x0020</td><td>Configured sleep timeout expired</td></tr> <tr> <td>0x0040</td><td>Vibration detected</td></tr> <tr> <td>0x0080</td><td>Main supply voltage exceeds configured threshold value</td></tr> <tr> <td>0x0100</td><td>Main supply voltage fell below 6.0VDC (device unplugged)</td></tr> </table>	Value (Hex)	Wake Reason	0x0001	Input 0 (Ignition) transitioned high	0x0002	Input 1 transitioned high	0x0004	Input 2 transitioned high	0x0008	Input 3 transitioned high	0x0010	Input 4 transitioned high	0x0020	Configured sleep timeout expired	0x0040	Vibration detected	0x0080	Main supply voltage exceeds configured threshold value	0x0100	Main supply voltage fell below 6.0VDC (device unplugged)		
Value (Hex)	Wake Reason																						
0x0001	Input 0 (Ignition) transitioned high																						
0x0002	Input 1 transitioned high																						
0x0004	Input 2 transitioned high																						
0x0008	Input 3 transitioned high																						
0x0010	Input 4 transitioned high																						
0x0020	Configured sleep timeout expired																						
0x0040	Vibration detected																						
0x0080	Main supply voltage exceeds configured threshold value																						
0x0100	Main supply voltage fell below 6.0VDC (device unplugged)																						
EnterSleep(<wake_mask>,<wake_minutes>)	Set the wake reason mask <table> <tr> <th>Value (Hex)</th><th>Wake Reason</th></tr> <tr> <td>0x0000</td><td>SMS or UDP command received</td></tr> <tr> <td>0x0001</td><td>Input 0 (Ignition) transitioned high</td></tr> <tr> <td>0x0002</td><td>Input 1 transitioned high</td></tr> <tr> <td>0x0004</td><td>Input 2 transitioned high</td></tr> <tr> <td>0x0008</td><td>Input 3 transitioned high</td></tr> <tr> <td>0x0010</td><td>Input 4 transitioned high</td></tr> <tr> <td>0x0020</td><td>Configured sleep timeout expired</td></tr> <tr> <td>0x0040</td><td>Vibration detected</td></tr> <tr> <td>0x0080</td><td>Main supply voltage exceeds configured threshold value</td></tr> <tr> <td>0x0100</td><td>Main supply voltage fell below 6.0VDC (device unplugged)</td></tr> </table>	Value (Hex)	Wake Reason	0x0000	SMS or UDP command received	0x0001	Input 0 (Ignition) transitioned high	0x0002	Input 1 transitioned high	0x0004	Input 2 transitioned high	0x0008	Input 3 transitioned high	0x0010	Input 4 transitioned high	0x0020	Configured sleep timeout expired	0x0040	Vibration detected	0x0080	Main supply voltage exceeds configured threshold value	0x0100	Main supply voltage fell below 6.0VDC (device unplugged)
Value (Hex)	Wake Reason																						
0x0000	SMS or UDP command received																						
0x0001	Input 0 (Ignition) transitioned high																						
0x0002	Input 1 transitioned high																						
0x0004	Input 2 transitioned high																						
0x0008	Input 3 transitioned high																						
0x0010	Input 4 transitioned high																						
0x0020	Configured sleep timeout expired																						
0x0040	Vibration detected																						
0x0080	Main supply voltage exceeds configured threshold value																						
0x0100	Main supply voltage fell below 6.0VDC (device unplugged)																						

5.2. INTERPRETER SCRIPTING TRIGGER BLOCK EXAMPLES

5.2.1. Packet Send Example

This example shows an alternate message being sent every two minutes. If flag[0] is set, then send packet 1, otherwise, send packet 0. Always reset the timer and toggle the flag.

```
trigger when Eq(TimerExpired(0), 1)
  conduct when Eq(FlagIsSet(0), 1) [Debounce(0, 0)]
    actions
      run BuildAndSendMsg(1, 0, 1, 0)
      run ClearFlag(0)
  conduct when Neq(FlagIsSet(0), 1) [Debounce(5, 0)]
    # NOTE: debounce Hi for 5 seconds, to keep these actions from executing
    # due to
    # the previous CondAct block clearing the flag.
    actions
      run BuildAndSendMsg(0, 0, 1, 0)
      run SetFlag(0)
  conduct always
    actions
      run ResetTimer(0)
```

5.2.2. Build and Send Message

A message is created and sent when you send a *user event* to the interpreter. The *user event* can be sent via :xrmsg <index>.



NOTE: see actions[] array in src/interpreter/interpreter.c to find valid action indices and number of arguments each action should take.

```
trigger when Eq(UserEventIsActive(0xff), 1)
```

```

conduct always
  actions
    run BuildAndSendMsg(4, 32, 1, 0)

```

5.2.3. Set User Flag 3 if OBD Reports PTO On

PTO status resides in the bit 1 position of Fld_ObdRunStatus (0x2c). The example below uses the BitAnd() operator with a hex mask isolating the desired bit, comparing it to 0 for rising edge trigger.

This operation works from the inside out, parenthetically. SystemVal (0x2c) grabs the value stored in Fld_ObdRunStatus. BitAnd() then takes that value “ANDed” with its second argument of 0x02, which isolates the bit 1 position.

When bit 1 of Fld_ObdRunStatus (PTO state) is high, the BitAnd() resolves to a 1. Neq() then takes that and compares it with its second argument of 0. When they don’t equal each other, the trigger is executed. The Debounce in this example is disabled.

```

trigger when Neq(BitAnd(SystemVal(0x2c), 0x02), 0) [Debounce(0, 0)]
  conduct always
    actions
      run SetFlag(3)

```

5.3. PACKET RECIPE

5.3.1. Fields

ID	Name	Description						
0x01	PacketID	Packet recipes can be labeled from 0-255, but there are only 128 recipe slots						
		<table><tr><th>Bytes</th><th>Units</th><th>Resolution</th><th>Range</th></tr><tr><td>1</td><td>N/A</td><td>1</td><td>0 to 255</td></tr></table>	Bytes	Units	Resolution	Range	1	N/A
Bytes	Units	Resolution	Range					
1	N/A	1	0 to 255					
0x02	FmCustomHeader	value TBD (currently zero)						
		<table><tr><th>Bytes</th><th>Units</th><th>Resolution</th><th>Range</th></tr><tr><td>1</td><td>N/A</td><td>1</td><td>0 to 255</td></tr></table>	Bytes	Units	Resolution	Range	1	N/A
Bytes	Units	Resolution	Range					
1	N/A	1	0 to 255					

ID	Name	Description			
0x03	DeviceId	Unsigned integer representing numeric ESN			
		Bytes	Units	Resolution	Range
		4	N/A	1	100000000 to 999999999
0x04	ReasonCode	Unsigned integer (any reason code < 0 or > 255 will wrap)			
		Bytes	Units	Resolution	Range
		1	N/A	1	0 to 255
0x05	PacketSerialNum	Unsigned integer			
		Bytes	Units	Resolution	Range
		2	N/A	1	0 to 65535
0x06	UnixTime	Unsigned integer representing number of seconds since Unix Epoch			
		Bytes	Units	Resolution	Range
		4	N/A	1	0x0 to 0xffffffff
0x07	Latitude	Signed integer decimal value of 4byte hex string divided by 1000000 (useful range -180.0 to 180.0)			
		Bytes	Units	Resolution	Range
		4	Degrees	1.00E-06	-2147.483648 to 2147.483647
0x08	Longitude	Signed integer decimal value of 4-byte hex string divided by 1000000 (useful range -90.0 to 90.0)			
		Bytes	Units	Resolution	Range
		4	Degrees	1.00E-06	-2147.483648 to 2147.483647
0x09	Altitude	Signed integer			
		Bytes	Units	Resolution	Range
		2	m	0.1	-3276.8 to

ID	Name	Description			
		Bytes	Units	Resolution	Range
					3276.7
0x0a	Heading	Signed integer (useful range 0.0 to 360.0)			
		Bytes	Units	Resolution	Range
		2	Degrees	1.00E-01	-3276.8 to 3276.7
0x0b	GpsSpeed	Unsigned integer			
		Bytes	Units	Resolution	Range
		1	kmh	1	0 to 255
0x0c	InputStates	See Input State Bits table below			
		Bytes	Units	Resolution	Range
		1	N/A	1	0x0 to 0xff
0x0d	OutputStates	Bitfield: see Output State bits Table below			
		Bytes	Units	Resolution	Range
		1	N/A	1	0x00 to 0x1f
0x0e	DriverIdCode1	Unsigned integer representing unique iButton ID			
		Bytes	Units	Resolution	Range
		4	N/A	1	0 to 4294967295
0x0f	DriverIdCode2	Unsigned integer representing unique iButton ID			
		Bytes	Units	Resolution	Range
		4	N/A	1	0 to 4294967295
0x10	GSPTripOdom1	See Odometer Details Table below			
		Bytes	Units	Resolution	Range
		4	m	1	0 to 4294967295
0x11	Flags	Bitfield			

ID	Name	Description			
		Bytes	Units	Resolution	Range
		4	N/A	1	0 to 4294967295
0x12	Hdop	Unsigned integer			
		Bytes	Units	Resolution	Range
		1	DOP	0.1	0.0 to 25.5
0x13	NumSats				
		Bytes	Units	Resolution	Range
		1	N/A	1	0 to 255
0x14	ReceiverSigStr	See Signal Quality Measure Table below			
		Bytes	Units	Resolution	Range
		2	dBm	1.23	-113 to -75
0x15	CellularCarrierId				
		Bytes	Units	Resolution	Range
		2	N/A	1	0x0 to 0xffff
0x16	InternalBattVolts	Possibly using a 2-byte unsigned integer representing mV (0 to 65535)			
		Bytes	Units	Resolution	Range
		1	V	0.1	0.0 to 25.5
0x17	VehicleBattVolts_1byte	If value is greater than 25.5v, it will cap at 25.5v			
		Bytes	Units	Resolution	Range
		1	V	0.1	0.0 to 25.5
0x18	GpsLifetimeOdom	Unsigned integer; see Odometer Details Table below			
		Bytes	Units	Resolution	Range
		4	m	1	0 to 4294967295
0x19	AccelStartDateTime	Unsigned integer representing time from Unix Epoch (s)			
		Bytes	Units	Resolution	Range
		4	N/A	1	0x0 to 0xffffffff

ID	Name	Description			
0x1a	AccelStartLat	Signed integer decimal value of 4-byte hex string divided by 1000000 (useful range -180.0 to 180.0)			
		Bytes	Units	Resolution	Range
		4	Degrees	1.00E-06	-2147.483648 to 2147.483647
0x1b	AccelStartLong	Signed integer decimal value of 4-byte hex string divided by 1000000 (useful range -90.0 to 90.0)			
		Bytes	Units	Resolution	Range
		4	Degrees	1.00E-06	-2147.483648 to 2147.483647
0x1c	AccelStartSpeed	Bytes	Units	Resolution	Range
		1	kmh	1	0 to 255
0x1d	AccelStartHeading	Bytes	Units	Resolution	Range
		2	Degrees	0.1	0 to 3599
0x1e	MaxAccel	Bytes	Units	Resolution	Range
		2	mG	1	0 to 32767
0x1f	AccelEventDuration	Bytes	Units	Resolution	Range
		1	Seconds	0.1	0 to 255
0x20	AccelEndDateTime	Unsigned integer representing time from Unix Epoch (s)			
		Bytes	Units	Resolution	Range
		4	N/A	1	0x0 to 0xffffffff
0x21	AccelEndLat	Signed integer decimal value of 4-byte hex string divided by 1000000 (useful range -180.0 to 180.0)			
		Bytes	Units	Resolution	Range
		4	Degrees	1.00E-06	-2147.483648 to 2147.483647

ID	Name	Description								
0x22	AccelEndLong	Signed integer decimal value of 4-byte hex string divided by 1000000 (useful range -90.0 to 90.0)								
		<table><tr><th>Bytes</th><th>Units</th><th>Resolution</th><th>Range</th></tr><tr><td>4</td><td>Degrees</td><td>1.00E-06</td><td>-2147.483648 to 2147.483647</td></tr></table>	Bytes	Units	Resolution	Range	4	Degrees	1.00E-06	-2147.483648 to 2147.483647
Bytes	Units	Resolution	Range							
4	Degrees	1.00E-06	-2147.483648 to 2147.483647							
0x23	AccelEndSpeed	<table><tr><th>Bytes</th><th>Units</th><th>Resolution</th><th>Range</th></tr><tr><td>1</td><td>kmh</td><td>1</td><td>0 to 255</td></tr></table>	Bytes	Units	Resolution	Range	1	kmh	1	0 to 255
Bytes	Units	Resolution	Range							
1	kmh	1	0 to 255							
0x24	AccelEndHeading	<table><tr><th>Bytes</th><th>Units</th><th>Resolution</th><th>Range</th></tr><tr><td>2</td><td>Degrees</td><td>0.1</td><td>0 to 3599</td></tr></table>	Bytes	Units	Resolution	Range	2	Degrees	0.1	0 to 3599
Bytes	Units	Resolution	Range							
2	Degrees	0.1	0 to 3599							
0x25	Not implemented	Available for future use								
0x26	GarminPacketSize	Size of entire Garmin packet (includes <DLE><pid><size><payload><sum><DLE><ETX> and DLE stuffing)								
		<table><tr><th>Bytes</th><th>Units</th><th>Resolution</th><th>Range</th></tr><tr><td>2</td><td>N/A</td><td>N/A</td><td>6 to 516</td></tr></table>	Bytes	Units	Resolution	Range	2	N/A	N/A	6 to 516
Bytes	Units	Resolution	Range							
2	N/A	N/A	6 to 516							
0x27	GarminPacket	Garmin Packet bytes: <DLE><pid><size><payload><sum><DLE><ETX> and DLE stuffing								
		<table><tr><th>Bytes</th><th>Units</th><th>Resolution</th><th>Range</th></tr><tr><td>Variable</td><td>N/A</td><td>N/A</td><td>N/A</td></tr></table>	Bytes	Units	Resolution	Range	Variable	N/A	N/A	N/A
Bytes	Units	Resolution	Range							
Variable	N/A	N/A	N/A							
0x28	BluetoothPayloadSize *	Unsigned integer								
		<table><tr><th>Bytes</th><th>Units</th><th>Resolution</th><th>Range</th></tr><tr><td>2</td><td>N/A</td><td>N/A</td><td>N/A</td></tr></table>	Bytes	Units	Resolution	Range	2	N/A	N/A	N/A
Bytes	Units	Resolution	Range							
2	N/A	N/A	N/A							
0x29	BluetoothPayload *	Bytes								
		<table><tr><th>Bytes</th><th>Units</th><th>Resolution</th><th>Range</th></tr><tr><td>Variable</td><td>N/A</td><td>N/A</td><td>N/A</td></tr></table>	Bytes	Units	Resolution	Range	Variable	N/A	N/A	N/A
Bytes	Units	Resolution	Range							
Variable	N/A	N/A	N/A							
0x2a	CommandPayloadSize *	<table><tr><th>Bytes</th><th>Units</th><th>Resolution</th><th>Range</th></tr><tr><td>2</td><td>N/A</td><td>N/A</td><td>N/A</td></tr></table>	Bytes	Units	Resolution	Range	2	N/A	N/A	N/A
Bytes	Units	Resolution	Range							
2	N/A	N/A	N/A							

ID	Name	Description
0x2b	CommandPayload *	Bytes Units Resolution Range
		Variable N/A N/A N/A
0x2c	OBDRunStates	Bitfield; see OBD Run States Bits Table below
		Bytes Units Resolution Range
		1 N/A N/A 0 to 7
0x2d	OBDCommsState	See OBD Comms State Table below
		Bytes Units Resolution Range
		1 N/A N/A 0 to 3
0x2e	ObdDerivedTripOdom	Unsigned integer; see Odometer Details Table below
		Bytes Units Resolution Range
		4 m 1 0 to 4294967295
0x2f	ObdLifetimeOdom	Unsigned integer; see Odometer Details Table below
		Bytes Units Resolution Range
		4 m 1 0 to 4294967295
0x30	TempSensor0	Bytes Units Resolution Range
		1 N/A N/A N/A
0x31	TempSensor1	Bytes Units Resolution Range
		1 N/A N/A N/A
0x32	UserVar8[0]	8-bit signed integer
		Bytes Units Resolution Range
		1 N/A 1 -128 to 127
0x33	UserVar8[1]	8-bit signed integer
		Bytes Units Resolution Range
		1 N/A 1 -128 to 127

ID	Name	Description						
0x34	UserVar8[2]	8-bit signed integer						
		<table><tr><th>Bytes</th><th>Units</th><th>Resolution</th><th>Range</th></tr><tr><td>1</td><td>N/A</td><td>1</td><td>-128 to 127</td></tr></table>	Bytes	Units	Resolution	Range	1	N/A
Bytes	Units	Resolution	Range					
1	N/A	1	-128 to 127					
0x35	UserVar8[3]	8-bit signed integer						
		<table><tr><th>Bytes</th><th>Units</th><th>Resolution</th><th>Range</th></tr><tr><td>1</td><td>N/A</td><td>1</td><td>-128 to 127</td></tr></table>	Bytes	Units	Resolution	Range	1	N/A
Bytes	Units	Resolution	Range					
1	N/A	1	-128 to 127					
0x36	UserVar8[4]	8-bit signed integer						
		<table><tr><th>Bytes</th><th>Units</th><th>Resolution</th><th>Range</th></tr><tr><td>1</td><td>N/A</td><td>1</td><td>-128 to 127</td></tr></table>	Bytes	Units	Resolution	Range	1	N/A
Bytes	Units	Resolution	Range					
1	N/A	1	-128 to 127					
0x37	UserVar8[5]	8-bit signed integer						
		<table><tr><th>Bytes</th><th>Units</th><th>Resolution</th><th>Range</th></tr><tr><td>1</td><td>N/A</td><td>1</td><td>-128 to 127</td></tr></table>	Bytes	Units	Resolution	Range	1	N/A
Bytes	Units	Resolution	Range					
1	N/A	1	-128 to 127					
0x38	UserVar8[6]	8-bit signed integer						
		<table><tr><th>Bytes</th><th>Units</th><th>Resolution</th><th>Range</th></tr><tr><td>1</td><td>N/A</td><td>1</td><td>-128 to 127</td></tr></table>	Bytes	Units	Resolution	Range	1	N/A
Bytes	Units	Resolution	Range					
1	N/A	1	-128 to 127					
0x39	UserVar8[7]	8-bit signed integer						
		<table><tr><th>Bytes</th><th>Units</th><th>Resolution</th><th>Range</th></tr><tr><td>1</td><td>N/A</td><td>1</td><td>-128 to 127</td></tr></table>	Bytes	Units	Resolution	Range	1	N/A
Bytes	Units	Resolution	Range					
1	N/A	1	-128 to 127					
0x3a	UserVar8[8]	8-bit signed integer						
		<table><tr><th>Bytes</th><th>Units</th><th>Resolution</th><th>Range</th></tr><tr><td>1</td><td>N/A</td><td>1</td><td>-128 to 127</td></tr></table>	Bytes	Units	Resolution	Range	1	N/A
Bytes	Units	Resolution	Range					
1	N/A	1	-128 to 127					
0x3b	UserVar8[9]	8-bit signed integer						
		<table><tr><th>Bytes</th><th>Units</th><th>Resolution</th><th>Range</th></tr><tr><td>1</td><td>N/A</td><td>1</td><td>-128 to 127</td></tr></table>	Bytes	Units	Resolution	Range	1	N/A
Bytes	Units	Resolution	Range					
1	N/A	1	-128 to 127					

ID	Name	Description						
0x3c	UserVar8[10]	8-bit signed integer						
		<table><tr><th>Bytes</th><th>Units</th><th>Resolution</th><th>Range</th></tr><tr><td>1</td><td>N/A</td><td>1</td><td>-128 to 127</td></tr></table>	Bytes	Units	Resolution	Range	1	N/A
Bytes	Units	Resolution	Range					
1	N/A	1	-128 to 127					
0x3d	UserVar8[11]	8-bit signed integer						
		<table><tr><th>Bytes</th><th>Units</th><th>Resolution</th><th>Range</th></tr><tr><td>1</td><td>N/A</td><td>1</td><td>-128 to 127</td></tr></table>	Bytes	Units	Resolution	Range	1	N/A
Bytes	Units	Resolution	Range					
1	N/A	1	-128 to 127					
0x3e	UserVar8[12]	8-bit signed integer						
		<table><tr><th>Bytes</th><th>Units</th><th>Resolution</th><th>Range</th></tr><tr><td>1</td><td>N/A</td><td>1</td><td>-128 to 127</td></tr></table>	Bytes	Units	Resolution	Range	1	N/A
Bytes	Units	Resolution	Range					
1	N/A	1	-128 to 127					
0x3f	UserVar8[13]	8-bit signed integer						
		<table><tr><th>Bytes</th><th>Units</th><th>Resolution</th><th>Range</th></tr><tr><td>1</td><td>N/A</td><td>1</td><td>-128 to 127</td></tr></table>	Bytes	Units	Resolution	Range	1	N/A
Bytes	Units	Resolution	Range					
1	N/A	1	-128 to 127					
0x40	UserVar8[14]	8-bit signed integer						
		<table><tr><th>Bytes</th><th>Units</th><th>Resolution</th><th>Range</th></tr><tr><td>1</td><td>N/A</td><td>1</td><td>-128 to 127</td></tr></table>	Bytes	Units	Resolution	Range	1	N/A
Bytes	Units	Resolution	Range					
1	N/A	1	-128 to 127					
0x41	UserVar8[15]	8-bit signed integer						
		<table><tr><th>Bytes</th><th>Units</th><th>Resolution</th><th>Range</th></tr><tr><td>1</td><td>N/A</td><td>1</td><td>-128 to 127</td></tr></table>	Bytes	Units	Resolution	Range	1	N/A
Bytes	Units	Resolution	Range					
1	N/A	1	-128 to 127					
0x42	UserVar16[0]	16-bit signed integer						
		<table><tr><th>Bytes</th><th>Units</th><th>Resolution</th><th>Range</th></tr><tr><td>2</td><td>N/A</td><td>1</td><td>-32768 to 32767</td></tr></table>	Bytes	Units	Resolution	Range	2	N/A
Bytes	Units	Resolution	Range					
2	N/A	1	-32768 to 32767					
0x43	UserVar16[1]	16-bit signed integer						
		<table><tr><th>Bytes</th><th>Units</th><th>Resolution</th><th>Range</th></tr><tr><td>2</td><td>N/A</td><td>1</td><td>-32768 to 32767</td></tr></table>	Bytes	Units	Resolution	Range	2	N/A
Bytes	Units	Resolution	Range					
2	N/A	1	-32768 to 32767					

ID	Name	Description						
0x44	UserVar16[2]	16-bit signed integer						
		<table><tr><th>Bytes</th><th>Units</th><th>Resolution</th><th>Range</th></tr><tr><td>2</td><td>N/A</td><td>1</td><td>-32768 to 32767</td></tr></table>	Bytes	Units	Resolution	Range	2	N/A
Bytes	Units	Resolution	Range					
2	N/A	1	-32768 to 32767					
0x45	UserVar16[3]	16-bit signed integer						
		<table><tr><th>Bytes</th><th>Units</th><th>Resolution</th><th>Range</th></tr><tr><td>2</td><td>N/A</td><td>1</td><td>-32768 to 32767</td></tr></table>	Bytes	Units	Resolution	Range	2	N/A
Bytes	Units	Resolution	Range					
2	N/A	1	-32768 to 32767					
0x46	UserVar16[4]	16-bit signed integer						
		<table><tr><th>Bytes</th><th>Units</th><th>Resolution</th><th>Range</th></tr><tr><td>2</td><td>N/A</td><td>1</td><td>-32768 to 32767</td></tr></table>	Bytes	Units	Resolution	Range	2	N/A
Bytes	Units	Resolution	Range					
2	N/A	1	-32768 to 32767					
0x47	UserVar16[5]	16-bit signed integer						
		<table><tr><th>Bytes</th><th>Units</th><th>Resolution</th><th>Range</th></tr><tr><td>2</td><td>N/A</td><td>1</td><td>-32768 to 32767</td></tr></table>	Bytes	Units	Resolution	Range	2	N/A
Bytes	Units	Resolution	Range					
2	N/A	1	-32768 to 32767					
0x48	UserVar16[6]	16-bit signed integer						
		<table><tr><th>Bytes</th><th>Units</th><th>Resolution</th><th>Range</th></tr><tr><td>2</td><td>N/A</td><td>1</td><td>-32768 to 32767</td></tr></table>	Bytes	Units	Resolution	Range	2	N/A
Bytes	Units	Resolution	Range					
2	N/A	1	-32768 to 32767					
0x49	UserVar16[7]	16-bit signed integer						
		<table><tr><th>Bytes</th><th>Units</th><th>Resolution</th><th>Range</th></tr><tr><td>2</td><td>N/A</td><td>1</td><td>-32768 to 32767</td></tr></table>	Bytes	Units	Resolution	Range	2	N/A
Bytes	Units	Resolution	Range					
2	N/A	1	-32768 to 32767					
0x4a	UserVar32[0]	32-bit signed integer						
		<table><tr><th>Bytes</th><th>Units</th><th>Resolution</th><th>Range</th></tr><tr><td>4</td><td>N/A</td><td>1</td><td>-2147483648 to 2147483647</td></tr></table>	Bytes	Units	Resolution	Range	4	N/A
Bytes	Units	Resolution	Range					
4	N/A	1	-2147483648 to 2147483647					
0x4b	UserVar32[1]	32-bit signed integer						
		<table><tr><th>Bytes</th><th>Units</th><th>Resolution</th><th>Range</th></tr><tr><td>4</td><td>N/A</td><td>1</td><td>-2147483648 to 2147483647</td></tr></table>	Bytes	Units	Resolution	Range	4	N/A
Bytes	Units	Resolution	Range					
4	N/A	1	-2147483648 to 2147483647					

ID	Name	Description								
0x4c	UserVar32[2]	32-bit signed integer								
		<table><tr><th>Bytes</th><th>Units</th><th>Resolution</th><th>Range</th></tr><tr><td>4</td><td>N/A</td><td>1</td><td>-2147483648 to 2147483647</td></tr></table>	Bytes	Units	Resolution	Range	4	N/A	1	-2147483648 to 2147483647
		Bytes	Units	Resolution	Range					
4	N/A	1	-2147483648 to 2147483647							
0x4d	UserVar32[3]	32-bit signed integer								
		<table><tr><th>Bytes</th><th>Units</th><th>Resolution</th><th>Range</th></tr><tr><td>4</td><td>N/A</td><td>1</td><td>-2147483648 to 2147483647</td></tr></table>	Bytes	Units	Resolution	Range	4	N/A	1	-2147483648 to 2147483647
		Bytes	Units	Resolution	Range					
4	N/A	1	-2147483648 to 2147483647							
0x4e	UserVar32[4]	32-bit signed integer								
		<table><tr><th>Bytes</th><th>Units</th><th>Resolution</th><th>Range</th></tr><tr><td>4</td><td>N/A</td><td>1</td><td>-2147483648 to 2147483647</td></tr></table>	Bytes	Units	Resolution	Range	4	N/A	1	-2147483648 to 2147483647
		Bytes	Units	Resolution	Range					
4	N/A	1	-2147483648 to 2147483647							
0x4f	UserVar32[5]	32-bit signed integer								
		<table><tr><th>Bytes</th><th>Units</th><th>Resolution</th><th>Range</th></tr><tr><td>4</td><td>N/A</td><td>1</td><td>-2147483648 to 2147483647</td></tr></table>	Bytes	Units	Resolution	Range	4	N/A	1	-2147483648 to 2147483647
		Bytes	Units	Resolution	Range					
4	N/A	1	-2147483648 to 2147483647							
0x50	UserVar32[6]	32-bit signed integer								
		<table><tr><th>Bytes</th><th>Units</th><th>Resolution</th><th>Range</th></tr><tr><td>4</td><td>N/A</td><td>1</td><td>-2147483648 to 2147483647</td></tr></table>	Bytes	Units	Resolution	Range	4	N/A	1	-2147483648 to 2147483647
		Bytes	Units	Resolution	Range					
4	N/A	1	-2147483648 to 2147483647							
0x51	UserVar32[7]	32-bit signed integer								
		<table><tr><th>Bytes</th><th>Units</th><th>Resolution</th><th>Range</th></tr><tr><td>4</td><td>N/A</td><td>1</td><td>-2147483648 to 2147483647</td></tr></table>	Bytes	Units	Resolution	Range	4	N/A	1	-2147483648 to 2147483647
		Bytes	Units	Resolution	Range					
4	N/A	1	-2147483648 to 2147483647							
0x52	SystemStates	Bitfield; see System State Bits Table below								
		<table><tr><th>Bytes</th><th>Units</th><th>Resolution</th><th>Range</th></tr><tr><td>4</td><td>N/A</td><td>1</td><td>0 to 4294967295</td></tr></table>	Bytes	Units	Resolution	Range	4	N/A	1	0 to 4294967295
		Bytes	Units	Resolution	Range					
4	N/A	1	0 to 4294967295							

ID	Name	Description								
0x53	WakeReason	Hex byte value; see Wake Reason Values Table								
		<table><tr><th>Bytes</th><th>Units</th><th>Resolution</th><th>Range</th></tr><tr><td>1</td><td>N/A</td><td>1</td><td>0 to 255</td></tr></table>	Bytes	Units	Resolution	Range	1	N/A	1	0 to 255
Bytes	Units	Resolution	Range							
1	N/A	1	0 to 255							
0x54	ObdTrueOdometer	32-bit unsigned integer; see Odometer Details Table below								
		<table><tr><th>Bytes</th><th>Units</th><th>Resolution</th><th>Range</th></tr><tr><td>4</td><td>m</td><td>1</td><td>0 to 858993459</td></tr></table>	Bytes	Units	Resolution	Range	4	m	1	0 to 858993459
Bytes	Units	Resolution	Range							
4	m	1	0 to 858993459							
0x55	ObdTotFuelUsed	32-bit unsigned integer								
		<table><tr><th>Bytes</th><th>Units</th><th>Resolution</th><th>Range</th></tr><tr><td>4</td><td>liters</td><td>1</td><td>0 to 4294967295</td></tr></table>	Bytes	Units	Resolution	Range	4	liters	1	0 to 4294967295
Bytes	Units	Resolution	Range							
4	liters	1	0 to 4294967295							
0x56	ObdTotEngHours	32-bit unsigned integer								
		<table><tr><th>Bytes</th><th>Units</th><th>Resolution</th><th>Range</th></tr><tr><td>4</td><td>hours</td><td>1</td><td>0 to 4294967295</td></tr></table>	Bytes	Units	Resolution	Range	4	hours	1	0 to 4294967295
Bytes	Units	Resolution	Range							
4	hours	1	0 to 4294967295							
0x57	ObdVehicleSpeed	16-bit unsigned integer								
		<table><tr><th>Bytes</th><th>Units</th><th>Resolution</th><th>Range</th></tr><tr><td>2</td><td>kmh</td><td>0.1</td><td>0 to 2550</td></tr></table>	Bytes	Units	Resolution	Range	2	kmh	0.1	0 to 2550
Bytes	Units	Resolution	Range							
2	kmh	0.1	0 to 2550							
0x58	ObdEngRpm	16-bit unsigned integer								
		<table><tr><th>Bytes</th><th>Units</th><th>Resolution</th><th>Range</th></tr><tr><td>2</td><td>rpm</td><td>1</td><td>0 to 65535</td></tr></table>	Bytes	Units	Resolution	Range	2	rpm	1	0 to 65535
Bytes	Units	Resolution	Range							
2	rpm	1	0 to 65535							
0x59	ObdEngCoolantTemp	16-bit signed integer								
		<table><tr><th>Bytes</th><th>Units</th><th>Resolution</th><th>Range</th></tr><tr><td>2</td><td>Celsius</td><td>1</td><td>-40 to 215</td></tr></table>	Bytes	Units	Resolution	Range	2	Celsius	1	-40 to 215
Bytes	Units	Resolution	Range							
2	Celsius	1	-40 to 215							
0x5a	ObdFuelLevelPct	16-bit unsigned integer								
		<table><tr><th>Bytes</th><th>Units</th><th>Resolution</th><th>Range</th></tr><tr><td>2</td><td>%</td><td>0.1</td><td>0 to 1000</td></tr></table>	Bytes	Units	Resolution	Range	2	%	0.1	0 to 1000
Bytes	Units	Resolution	Range							
2	%	0.1	0 to 1000							

ID	Name	Description			
0x5b	ObdTotDrivingSec	32-bit unsigned integer			
		Bytes	Units	Resolution	Range
		4	s	1	0 to 4294967295
0x5c	ObdTotCruiseSec	32-bit unsigned integer			
		Bytes	Units	Resolution	Range
		4	s	1	0 to 4294967295
0x5d	ObdTotIdleSec	32-bit unsigned integer			
		Bytes	Units	Resolution	Range
		4	s	1	0 to 4294967295
0x5e	ObdTotIdleFuel	32-bit unsigned integer			
		Bytes	Units	Resolution	Range
		4	liters	1	0 to 4294967295
0x5f	ObdHarshBreakTotCnt	32-bit unsigned integer			
		Bytes	Units	Resolution	Range
		4		1	0 to 4294967295
0x60	ObdSpeedExceedTotCnt	32-bit unsigned integer			
		Bytes	Units	Resolution	Range
		4		1	0 to 4294967295
0x61	ObdRPMExceedTotCnt	32-bit unsigned integer			
		Bytes	Units	Resolution	Range
		4		1	0 to 4294967295
0x62	ObdHarchAccelTotCnt	32-bit unsigned integer			
		Bytes	Units	Resolution	Range
		4		1	0 to 4294967295

ID	Name	Description								
0x63	Not implemented	Available for future use								
		<table><tr><th>Bytes</th><th>Units</th><th>Resolution</th><th>Range</th></tr><tr><td>1</td><td>N/A</td><td>N/A</td><td>N/A</td></tr></table>	Bytes	Units	Resolution	Range	1	N/A	N/A	N/A
Bytes	Units	Resolution	Range							
1	N/A	N/A	N/A							
0x64	BluetoothName *									
		<table><tr><th>Bytes</th><th>Units</th><th>Resolution</th><th>Range</th></tr><tr><td>1</td><td>N/A</td><td>N/A</td><td>N/A</td></tr></table>	Bytes	Units	Resolution	Range	1	N/A	N/A	N/A
Bytes	Units	Resolution	Range							
1	N/A	N/A	N/A							
0x65	ObdVIN									
		<table><tr><th>Bytes</th><th>Units</th><th>Resolution</th><th>Range</th></tr><tr><td>17</td><td>ascii</td><td>N/A</td><td>N/A</td></tr></table>	Bytes	Units	Resolution	Range	17	ascii	N/A	N/A
Bytes	Units	Resolution	Range							
17	ascii	N/A	N/A							
0x66	ExternADC0_1byte	Value caps at 25.5v								
		<table><tr><th>Bytes</th><th>Units</th><th>Resolution</th><th>Range</th></tr><tr><td>1</td><td>v</td><td>0.1</td><td>0.0 to 25.5</td></tr></table>	Bytes	Units	Resolution	Range	1	v	0.1	0.0 to 25.5
Bytes	Units	Resolution	Range							
1	v	0.1	0.0 to 25.5							
0x67	Reserved *									
		<table><tr><th>Bytes</th><th>Units</th><th>Resolution</th><th>Range</th></tr><tr><td>N/A</td><td>N/A</td><td>N/A</td><td>N/A</td></tr></table>	Bytes	Units	Resolution	Range	N/A	N/A	N/A	N/A
Bytes	Units	Resolution	Range							
N/A	N/A	N/A	N/A							
0x68	Accel Metrics	Rounded to nearest whole number								
		<table><tr><th>Bytes</th><th>Units</th><th>Resolution</th><th>Range</th></tr><tr><td>4</td><td>N/A</td><td>N/A</td><td>N/A</td></tr></table>	Bytes	Units	Resolution	Range	4	N/A	N/A	N/A
Bytes	Units	Resolution	Range							
4	N/A	N/A	N/A							
0x69	Boot Status	8-bit unsigned integer; see Boot Status Values Table below								
		<table><tr><th>Bytes</th><th>Units</th><th>Resolution</th><th>Range</th></tr><tr><td>1</td><td>N/A</td><td>1</td><td>0 to 7</td></tr></table>	Bytes	Units	Resolution	Range	1	N/A	1	0 to 7
Bytes	Units	Resolution	Range							
1	N/A	1	0 to 7							
0x6a	Apn Index	8-bit unsigned integer								
		<table><tr><th>Bytes</th><th>Units</th><th>Resolution</th><th>Range</th></tr><tr><td>1</td><td>N/A</td><td>1</td><td>0 to 3</td></tr></table>	Bytes	Units	Resolution	Range	1	N/A	1	0 to 3
Bytes	Units	Resolution	Range							
1	N/A	1	0 to 3							
0x6b	ObdDtcPacketSize	Unsigned integer								

ID	Name	Description								
		<table><tr><th>Bytes</th><th>Units</th><th>Resolution</th><th>Range</th></tr><tr><td>2</td><td>N/A</td><td>N/A</td><td>N/A</td></tr></table>	Bytes	Units	Resolution	Range	2	N/A	N/A	N/A
Bytes	Units	Resolution	Range							
2	N/A	N/A	N/A							
0x6c	ObdDtcPacket	<mil_status><num_ecu_dtc><ecu_id_0><num_dtc_0><dtc_code_ascii_0><fmi_0>...<dtc_code_ascii_n><fmi_n>...<ecu_id_m><num_dtc_0><dtc_code_ascii_0><fmi_0>...<dtc_code_ascii_n><fmi_n> dtc formatted is 1-byte ascii + 4-bytes code + 1-byte fmi <table><tr><th>Bytes</th><th>Units</th><th>Resolution</th><th>Range</th></tr><tr><td>Variable</td><td>N/A</td><td>N/A</td><td>N/A</td></tr></table>	Bytes	Units	Resolution	Range	Variable	N/A	N/A	N/A
Bytes	Units	Resolution	Range							
Variable	N/A	N/A	N/A							
0x6d	ObdBackoff	See OBD Backoff Mode Table below <table><tr><th>Bytes</th><th>Units</th><th>Resolution</th><th>Range</th></tr><tr><td>1</td><td>N/A</td><td>N/A</td><td>0 to 3</td></tr></table>	Bytes	Units	Resolution	Range	1	N/A	N/A	0 to 3
Bytes	Units	Resolution	Range							
1	N/A	N/A	0 to 3							
0x6e	ObdProtocols	See OBD Protocol Table, Secondary protocol in MS word, Primary protocol in LS word. e.g. secondary J1708 + primary J1939 = (0x00400100) <table><tr><th>Bytes</th><th>Units</th><th>Resolution</th><th>Range</th></tr><tr><td>4</td><td>Hex</td><td>N/A</td><td>N/A</td></tr></table>	Bytes	Units	Resolution	Range	4	Hex	N/A	N/A
Bytes	Units	Resolution	Range							
4	Hex	N/A	N/A							
0x6f	CrashPacketSize	Unsigned integer <table><tr><th>Bytes</th><th>Units</th><th>Resolution</th><th>Range</th></tr><tr><td>2</td><td>N/A</td><td>N/A</td><td>N/A</td></tr></table>	Bytes	Units	Resolution	Range	2	N/A	N/A	N/A
Bytes	Units	Resolution	Range							
2	N/A	N/A	N/A							
0x70	CrashPacket	A collection of configurable historical reoriented basic accelerometer vectors structured in little Endian 2 byte values as x, y, z in milli-G's <table><tr><th>Bytes</th><th>Units</th><th>Resolution</th><th>Range</th></tr><tr><td>Variable</td><td>N/A</td><td>N/A</td><td>N/A</td></tr></table>	Bytes	Units	Resolution	Range	Variable	N/A	N/A	N/A
Bytes	Units	Resolution	Range							
Variable	N/A	N/A	N/A							
0x71	ObdTotPTOTime	32-bit unsigned integer <table><tr><th>Bytes</th><th>Units</th><th>Resolution</th><th>Range</th></tr><tr><td>4</td><td>seconds</td><td>1</td><td>0 to</td></tr></table>	Bytes	Units	Resolution	Range	4	seconds	1	0 to
Bytes	Units	Resolution	Range							
4	seconds	1	0 to							

ID	Name	Description			
		Bytes	Units	Resolution	Range
					4294967295
0x72	ObdTotPTOFuel	32-bit unsigned integer			
		Bytes	Units	Resolution	Range
		4	liters	1	0 to 4294967295
0x73	FwRev	16-byte null-terminated ascii string. Version string cuts off the first 9 bytes, so "AAb1-1133BB1-e6" would report as "BB1-e6".			
		Bytes	Units	Resolution	Range
		16	ascii	N/A	N/A
0x74	OBDPidReserved1	Signed integer decimal value of 4-byte hex string divided by 10			
		Bytes	Units	Resolution	Range
		4	N/A	0.1	-2147483648 to 2147483647
0x75	OBDPidReserved2	Signed integer decimal value of 4byte hex string divided by 10			
		Bytes	Units	Resolution	Range
		4	N/A	0.1	-2147483648 to 2147483647
0x76	OBDPidReserved3	Signed integer decimal value of 4byte hex string divided by 10			
		Bytes	Units	Resolution	Range
		4	N/A	0.1	-2147483648 to 2147483647
0x77	OBDPidReserved4	Signed integer decimal value of 4byte hex string divided by 10			
		Bytes	Units	Resolution	Range
		4	N/A	0.1	-2147483648 to 2147483647
0x78	OBDPidReserved5	Signed integer decimal value of 4byte hex string divided by 10			
		Bytes	Units	Resolution	Range
		4	N/A	0.1	-2147483648 to 2147483647

ID	Name	Description			
		Bytes	Units	Resolution	Range
		4	N/A	0.1	-2147483648 to 2147483647
0x79	ObdTransGear	16-bit unsigned integer			
		2	0,1,2,3	1	0=neutral 1=forward 2=reverse 3=park
0x7A	ObdFuelTemp	16-bit signed integer			
		2	Celsius	1	-40 to 215
0x7B	ObdOilTemp	16-bit signed integer			
		2	Celsius	1	-40 to 215
0x7C	ObdThrottlePos	16-bit unsigned integer			
		2	%	1	0 to 100
0x7D	ObdMPG	16-bit unsigned integer			
		2	mpg	0.1	0.0 to 256.0
0x7E	ObdAccelPos	16-bit unsigned integer			
		2	%	1	0 to 100
0x7F	ObdEngLoad	16-bit unsigned integer			

ID	Name	Description			
		Bytes	Units	Resolution	Range
		2	%	1	0 to 250
0x80	ObdEngTorque	16-bit signed integer			
		2	%	1	-125 to 125
0x81	ObdOilLevel	16-bit unsigned integer			
		2	%	1	0 to 100
0x82	ObdOilPressure	16-bit unsigned integer			
		2	kPa	1	0 to 100
0x83	ObdCoolantPressure	16-bit unsigned integer			
		2	kPa	1	0 to 100
0x84	ObdIntakeAirTemp	16-bit signed integer			
		2	Celsius	1	-40 to 215
0x85	ObdManifoldTemp	16-bit signed integer			
		2	Celsius	1	-40 to 215
0x86	ObdCoolantLevel	16-bit unsigned integer			
		2	%	1	0 to 100
0x87	LinkageVer	32-bit unsigned integer; see Linkage Version Table below			

ID	Name	Description			
		Bytes	Units	Resolution	Range
		4	N/A	1	0x0 to 0xffffffff
0x88	GSPTripOdom2	See Odometer Details Table below			
		4	m	1	0 to 4294967295
0x89	GenCfgVal0	Generic Configuration Value set via paramset			
		4	N/A	1	0 to 4294967295
0x8a	GenCfgVal1	Generic Configuration Value set via paramset			
		4	N/A	1	0 to 4294967295
0x8b	GenCfgVal2	Generic Configuration Value set via paramset			
		4	N/A	1	0 to 4294967295
0x8c	GenCfgVal3	Generic Configuration Value set via paramset			
		4	N/A	1	0 to 4294967295
0x8d	GenCfgVal4	Generic Configuration Value set via paramset			
		4	N/A	1	0 to 4294967295
0x8e	OBDPidReserved6	Signed integer decimal value of 4-byte hex string divided by 10			
		4	N/A	0.1	-2147483648 to 2147483647
0x8f	OBDPidReserved7	Signed integer decimal value of 4-byte hex string divided by 10			

ID	Name	Description			
		Bytes	Units	Resolution	Range
		4	N/A	0.1	-2147483648 to 2147483647
0x90	OBDPidReserved8	Signed integer decimal value of 4-byte hex string divided by 10			
		4	N/A	0.1	-2147483648 to 2147483647
0x91	OBDPidReserved9	Signed integer decimal value of 4-byte hex string divided by 10			
		4	N/A	0.1	-2147483648 to 2147483647
0x92	OBDPidReserved10	Signed integer decimal value of 4-byte hex string divided by 10			
		4	N/A	0.1	-2147483648 to 2147483647
0x93	OBDDHarshAccelValue	Unsigned integer decimal value of acceleration in mph/second			
		1	Mph/s	1	0 to 255
0x94	OBDDHarshBrakeValue	Unsigned integer decimal value of deceleration in mph/second			
		1	Mph/s	1	0 to 255
0x95	CellSerialId	24-byte null-terminated ascii string			
		24	ascii	N/A	N/A
0x96	PktChkSum	Check sum for the packet, calculated so all bytes sum to 0x00			

ID	Name	Description			
		Bytes	Units	Resolution	Range
		1	N/A	N/A	N/A
0x97	TPSFields0	Inclusion of a configured length variable content section defined by the corresponding pct config entry			
		variable	N/A	N/A	N/A
0x98	TPSFields1	Inclusion of a configured length variable content section defined by the corresponding pct config entry			
		variable	N/A	N/A	N/A
0x99	TPSFields2	Inclusion of a configured length variable content section defined by the corresponding pct config entry			
		variable	N/A	N/A	N/A
0x9a	TPSFields3	Inclusion of a configured length variable content section defined by the corresponding pct config entry			
		variable	N/A	N/A	N/A
0x9b	TPSFields4	Inclusion of a configured length variable content section defined by the corresponding pct config entry			
		variable	N/A	N/A	N/A
0x9c	OBDPidReserved11	4-byte ascii normalization of the vehicle response			
		4	ascii	N/A	N/A
0x9d	OBDPidReserved12	4-byte ascii normalization of the vehicle response			

ID	Name	Description			
		Bytes	Units	Resolution	Range
		4	ascii	N/A	N/A
0x9e	OBDPidReserved13	4-byte ascii normalization of the vehicle response			
		4	ascii	N/A	N/A
0x9f	OBDPidReserved14	4-byte ascii normalization of the vehicle response			
		4	ascii	N/A	N/A
0xa0	OBDPidReserved15	4-byte ascii normalization of the vehicle response			
		4	ascii	N/A	N/A
0xa1	Debug-PacketSize	Unsigned integer			
		2	N/A	N/A	N/A
0xa2	DebugPacket	Payload			
		variable	N/A	N/A	N/A
0xa3	UserVar8[16]	8-bit signed integer			
		1	N/A	1	-128 to 127
0xa4	UserVar8[17]	8-bit signed integer			
		1	N/A	1	-128 to 127
0xa5	UserVar8[18]	8-bit signed integer			

ID	Name	Description			
		Bytes	Units	Resolution	Range
		1	N/A	1	-128 to 127
0xa6	UserVar8[19]	8-bit signed integer			
		1	N/A	1	-128 to 127
0xa7	UserVar8[20]	8-bit signed integer			
		1	N/A	1	-128 to 127
0xa8	UserVar8[21]	8-bit signed integer			
		1	N/A	1	-128 to 127
0xa9	UserVar8[22]	8-bit signed integer			
		1	N/A	1	-128 to 127
0xaa	UserVar8[23]	8-bit signed integer			
		1	N/A	1	-128 to 127
0xab	UserVar8[24]	8-bit signed integer			
		1	N/A	1	-128 to 127
0xac	UserVar8[25]	8-bit signed integer			
		1	N/A	1	-128 to 127
0xad	UserVar8[26]	8-bit signed integer			
		1	N/A	1	-128 to 127

ID	Name	Description			
		Bytes	Units	Resolution	Range
		1	N/A	1	-128 to 127
0xae	UserVar8[27]	8-bit signed integer			
		1	N/A	1	-128 to 127
0xaf	UserVar8[28]	8-bit signed integer			
		1	N/A	1	-128 to 127
0xb0	UserVar8[29]	8-bit signed integer			
		1	N/A	1	-128 to 127
0xb1	UserVar8[30]	8-bit signed integer			
		1	N/A	1	-128 to 127
0xb2	UserVar8[31]	8-bit signed integer			
		1	N/A	1	-128 to 127
0xb3	UserVar16[8]	16-bit signed integer			
		2	N/A	1	-32768 to 32767
0xb4	UserVar16[9]	16-bit signed integer			
		2	N/A	1	-32768 to 32767
0xb5	UserVar16[10]	16-bit signed integer			

ID	Name	Description			
		Bytes	Units	Resolution	Range
		2	N/A	1	-32768 to 32767
0xb6	UserVar16[11]	16-bit signed integer			
		2	N/A	1	-32768 to 32767
0xb7	UserVar16[12]	16-bit signed integer			
		2	N/A	1	-32768 to 32767
0xb8	UserVar16[13]	16-bit signed integer			
		2	N/A	1	-32768 to 32767
0xb9	UserVar16[14]	16-bit signed integer			
		2	N/A	1	-32768 to 32767
0xba	UserVar16[15]	16-bit signed integer			
		2	N/A	1	-32768 to 32767
0xbb	UserVar32[8]	32-bit signed integer			
		4	N/A	1	-2147483648 to 2147483647
0xbc	UserVar32[9]	32-bit signed integer			
		4	N/A	1	-2147483648 to 2147483647

ID	Name	Description						
0xbd	UserVar32[10]	32-bit signed integer						
		<table><tr><th>Bytes</th><th>Units</th><th>Resolution</th><th>Range</th></tr><tr><td>4</td><td>N/A</td><td>1</td><td>-2147483648 to 2147483647</td></tr></table>	Bytes	Units	Resolution	Range	4	N/A
Bytes	Units	Resolution	Range					
4	N/A	1	-2147483648 to 2147483647					
0xbe	UserVar32[11]	32-bit signed integer						
		<table><tr><th>Bytes</th><th>Units</th><th>Resolution</th><th>Range</th></tr><tr><td>4</td><td>N/A</td><td>1</td><td>-2147483648 to 2147483647</td></tr></table>	Bytes	Units	Resolution	Range	4	N/A
Bytes	Units	Resolution	Range					
4	N/A	1	-2147483648 to 2147483647					
0xbf	UserVar32[12]	32-bit signed integer						
		<table><tr><th>Bytes</th><th>Units</th><th>Resolution</th><th>Range</th></tr><tr><td>4</td><td>N/A</td><td>1</td><td>-2147483648 to 2147483647</td></tr></table>	Bytes	Units	Resolution	Range	4	N/A
Bytes	Units	Resolution	Range					
4	N/A	1	-2147483648 to 2147483647					
0xc0	UserVar32[13]	32-bit signed integer						
		<table><tr><th>Bytes</th><th>Units</th><th>Resolution</th><th>Range</th></tr><tr><td>4</td><td>N/A</td><td>1</td><td>-2147483648 to 2147483647</td></tr></table>	Bytes	Units	Resolution	Range	4	N/A
Bytes	Units	Resolution	Range					
4	N/A	1	-2147483648 to 2147483647					
0xc1	UserVar32[14]	32-bit signed integer						
		<table><tr><th>Bytes</th><th>Units</th><th>Resolution</th><th>Range</th></tr><tr><td>4</td><td>N/A</td><td>1</td><td>-2147483648 to 2147483647</td></tr></table>	Bytes	Units	Resolution	Range	4	N/A
Bytes	Units	Resolution	Range					
4	N/A	1	-2147483648 to 2147483647					
0xc2	UserVar32[15]	32-bit signed integer						
		<table><tr><th>Bytes</th><th>Units</th><th>Resolution</th><th>Range</th></tr><tr><td>4</td><td>N/A</td><td>1</td><td>-2147483648 to 2147483647</td></tr></table>	Bytes	Units	Resolution	Range	4	N/A
Bytes	Units	Resolution	Range					
4	N/A	1	-2147483648 to 2147483647					
0xc3	OBD Start Reason	16-bit signed integer; see OBD Trip Start Reason (bit mask) table below						
		<table><tr><th>Bytes</th><th>Units</th><th>Resolution</th><th>Range</th></tr><tr><td>4</td><td>N/A</td><td>1</td><td>-2147483648 to</td></tr></table>	Bytes	Units	Resolution	Range	4	N/A
Bytes	Units	Resolution	Range					
4	N/A	1	-2147483648 to					

ID	Name	Description			
		Bytes	Units	Resolution	Range
					2147483647
0xc4	OBD End Reason	32-bit signed integer; see OBD Trip End Reason (bit mask) table below			
		Bytes	Units	Resolution	Range
		4	N/A	1	-2147483648 to 2147483647
0xc5	TpsTagPayloadSize	Unsigned integer			
		Bytes	Units	Resolution	Range
		2	N/A	1	N/A
0xc6	TpsTagPayload	Payload			
		Bytes	Units	Resolution	Range
		variable	N/A	N/A	N/A
0xc7	OBD Seat Belt Status	Bytes	Units	Resolution	Range
		1	N/A	N/A	N/A
0xc8	VehicleBattery_2byte	2-byte representation of vehicle battery voltage			
		Bytes	Units	Resolution	Range
		2	V	0.1	0.0 to 6553.5
0xc9	ExtAdc[0]_2byte	2-byte representation of external ADC[0] voltage			
		Bytes	Units	Resolution	Range
		2	V	0.1	0.0 to 6553.5
0xca	AuxPassthruPacketSize	Size of entire aux passthru packet			
		Bytes	Units	Resolution	Range
		2	N/A	1	1 to 256

ID	Name	Description
0xcb	AuxPassthruPacket	Bytes Units Resolution Range
		variable N/A N/A N/A
0xcc	Geofence States Bitfield 1	Bitfield representing current state of active geofences 0 to 31, where 1 is inside, and 0 is outside or inactive.
		Bytes Units Resolution Range
		4 N/A 1 0 to 4294967295
0xcd	Geofence States Bitfield 2	Bitfield representing current state of active geofences 32 to 49, where 1 is inside, and 0 is outside or inactive.
		Bytes Units Resolution Range
		4 N/A 1 0 to 4294967295
0xce	Script version	Version number of currently loaded script
		Bytes Units Resolution Range
		4 N/A 1 0 to 4294967295
0xcf	Paramset version	Version number of currently loaded paramset
		Bytes Units Resolution Range
		4 N/A 1 0 to 4294967295
0xd0	Cell ICCID	Cellular ICCID string
		Bytes Units Resolution Range
		21 ascii N/A N/A
0xd1	Cell MDN	Cellular MDN string
		Bytes Units Resolution Range
		20 ascii N/A N/A
0xd2	OBD debug packet size	Unsigned integer
		Bytes Units Resolution Range
		2 N/A 1 N/A

ID	Name	Description
0xd3	OBD debug packet	Bytes
		Units
		Resolution
		Range
		variable
		N/A
		N/A
		N/A
0xd4	Cell Access Technology	Cell Access Technology values
		Bytes
		Units
		Resolution
		Range
		1
		N/A
		N/A
		0 to 7,255
0xd5	Derived Engine Seconds	Derived engine seconds
		Bytes
		Units
		Resolution
		Range
		4
		seconds
		1
		0 to
		4294967295
0xd6	ECU Engine Seconds	ECU engine seconds
		Bytes
		Units
		Resolution
		Range
		4
		seconds
		1
		0 to
		4294967295
0xd7	InputActiveStates	Same as the Input States bitfield, only the reported states are determined by the IDP config settings (see Input State Bits table below)
		Bytes
		Units
		Resolution
		Range
		1
		N/A
		1
		0x0 to 0xff
0xd8	CellularEnvironment	Each byte represents a different value
		MSB
		LSB
		[CSQ]
		[RSRP]
		[RSRQ]
		[CINR]
		CINR is not currently implemented (always 0xFF).
		RSRP & RSRQ are always 0xFF for modules that do not support AT+CESQ.
		Bytes
		Units
		Resolution
		Range
		4
		Varies
		1
		0 to 0xff (within each byte)
0xdd	GPS Speed (high-resolution)	Unsigned integer. Greater range equals higher accuracy.

ID	Name	Description			
		Bytes	Units	Resolution	Range
		2	kph	0.01	0.00 to 655.35
0xde	OBD MPG 5 second average	Averaging synchronized with OBD derived lifetime odometer timing.			
		Bytes	Units	Resolution	Range
		2	mpg	0.1	0.0 to 256.0
0xdf	GPS Harsh Accel Violations Cnt	Bytes	Units	Resolution	Range
		4		1	0 to 4294967295
0xe0	GPS Harsh Accel Value	Bytes	Units	Resolution	Range
		2	kph/s	0.01	0.00 to 655.35
0xe1	GPS Harsh Decel Violations Cnt	Bytes	Units	Resolution	Range
		4		1	0 to 4294967295
0xe2	GPS Harsh Decel Value	Bytes	Units	Resolution	Range
		2	kph/s	0.01	0.00 to 655.35
0xe3	OBD to Fuel Used HR	High resolution total fuel used.			
		Bytes	Units	Resolution	Range
		4	liters/bit	0.5	0 to 4294967295

5.3.2. Input/Output State Bits

Input

Bit	Input State
0	Ignition
1	Input 1
2	Input 2
3	Input 3
4	Input 4
5	Device In Motion (Accel)
6	Main supply voltage available
7	Accelerometer Reorientation Valid

output

Bit	Output State
0	Output 0
1	Output 1
2	Output 2
3	Onboard Buzzer
4	Garmin Power

5.3.3. Odometer Details

Odometer	Reset	Description
GPS Trip Odom 1	Reset by script	Great-circle distance between subsequent sets of GPS latitude and longitude
GPS Trip Odom 2	Reset by script	Great-circle distance between subsequent sets of GPS latitude and longitude
OBD Derived Trip Odom	Reset by script	Distance calculated based on averaged subsequent OBD speed readings
GPS Lifetime Odom	Never resets	Great-circle distance between subsequent sets of GPS latitude and longitude
OBD Lifetime Odom	Never resets	Accumulated odometer since install, based either on ECU odometer or averaged OBD speed readings
OBD True Odom	Cannot modify vehicle odometer	Distance acquired directly from ECU, 1m per bit

5.3.4. Linkage Version

Developer Revision	Number of Defined Fields	Sum of Defined Field sizes
4	208	> 645bytes
5	151	> 588bytes

5.3.5. Signal Quality Measure (SQM) Table

SQM Value	dBm
0	-113.0
1	-111.8
2	-110.5
3	-109.3
4	-108.1
5	-106.8
6	-105.6
7	-104.4
8	-103.1
9	-101.9
10	-100.7
11	-99.4
12	-98.2
13	-97.0
14	-95.7
15	-94.5
16	-93.3
17	-92.0
18	-90.8
19	-89.6
20	-88.3
21	-87.1
22	-85.9
23	-84.6
24	-83.4
25	-82.2
26	-80.9
27	-79.7
28	-78.5
29	-77.2
30	-76.0
31	-75 or better
99	not known/ not detectable

5.3.6. System State Bits

Bit	State
0	Valid Registration
1	Valid IP
2	Valid GPS
3	Valid Script
4	DM Session Active
5	Accel Oriented
6	Bluetooth Discoverable (TBD)
7	Bluetooth Paired (TBD)
8	Bluetooth Authenticated (TBD)
9	Bluetooth Connected (TBD)
10	Bluetooth Powered (TBD)
11	Cellular Powered
12	GPS Powered
13	Garmin Powered
14	Motion Via Accel
15	Motion Via GPS
16	External Power

5.3.7. OBD Run States Bits

Bit	State
0	Engine Running
1	PTO On
2	Moving
3	Ignition On

5.3.8. OBD Comms State

Value	Description
0	OBD comms off
1	OBD comms pending (ECU detected, initiating connection)
2	OBD comms connected (waiting for RPM change or vehicle motion)
3	OBD comms active

5.3.9. OBD Backoff Mode

Value	Description
0	OBD not in Backoff
1	OBD Backoff Scantool Detect
2	OBD Backoff MIL Detect
3	OBD Backoff MIL Persisted
4	OBD Backoff Protocol Fail Detect
5	OBD Backoff Scantool Detect until Ignition Off

5.3.10. OBD Protocol Table

Value (Hex)	Description
0x0001	ISO 9141-2 (Keyword Byte 08 08)
0x0002	ISO 9141-2 (Keyword Byte 94 94)
0x0004	ISO14230 (KWP2000) (5-Baud init)
0x0008	ISO14230 (KWP2000) (Fast init)
0x0010	J1850 PWM (typically older Ford vehicles)
0x0020	J1850 VPW (typically older GM vehicles and some older Chrysler and Toyota)
0x0040	J1708/J1587
0x0080	CAN protocol 11-bit CAN ids
0x0100	J1939
0x0200	CAN protocol 29-bit CAN ids

5.3.11. OBD Trip Start Reason (Bit Mask)

Value (Hex)	Description
0x0001	Detected Alternator On or Switched Ignition Input
0x0002	Detected GPS Movement
0x0004	Detected Vehicle Bus Traffic
0x0008	Detected Changing Vehicle Speed
0x0010	Detected Changing RPM

5.3.12. OBD Trip End Reason (Bit Mask)

Value	Description
0x00000001	Start Trip Detect Failed
0x00000002	Protocol detected, but no start trip conditions met
0x00000004	Failed to enable and configure OBD communications to a previous known protocol
0x00000008	Failed all attempts at vehicle protocol communications
0x00000010	Trip was active; failed to monitor any valid messages for x seconds or failed all requests for mode 1, pid 0
0x00000020	Master Vehicle Speed ECU has stopped responding
0x00000040	Master RPM ECU has stopped responding
0x00000080	Three-Minute Timer expired with no changing RPM.
0x00000100	ECO mode not enabled and RPM of 0 detected
0x00000200	TPS Filed upload request
0x00000400	CAN Transport is not configured
0x00000800	J1939 No Valid Claim Address
0x00001000	CAN Transport - transmit not ready
0x00002000	CAN Send - CAN Controller not Active
0x00004000	CAN Bus Off
0x00008000	CAN Transmit Timeout
0x00010000	CAN hardware Off
0x00020000	CAN Invalid Transmit Length

5.3.13. Boot Status Values

Value	Boot Status
0	Device can't boot (FW won't run; all 3 LEDs illuminated)
1	FW Invalid/Faulty (FW reverted by loading Backup FW)
2	Backup Image Invalid/Missing (current FW stored to Backup FW)
3	Normal Boot
4	FW Upgrade with Invalid/Faulty current FW and Invalid/Missing Backup FW(New FW loaded, New FW stored to BackupFW)
5	FW Upgrade with Invalid/Faulty current FW (New FW loaded)
6	FW Upgrade with Invalid/Missing Backup (current FW stored to Backup FW, New FW loaded)
7	Normal FW Upgrade (current FW stored to Backup FW, New FW loaded)

5.3.14. Proprietary PID Recipe Slots

Field	Assigned Data
OBDPidReserved1	Odometer
OBDPidReserved2	Oil Life percent
OBDPidReserved3	LF Tire pressure
OBDPidReserved4	RF Tire pressure
OBDPidReserved5	LR Tire pressure
OBDPidReserved6	RR Tire pressure
OBDPidReserved7	SPARE Tire pressure
OBDPidReserved8	LRI Tire Pressure
OBDPidReserved9	RRI Tire Pressure
OBDPidReserved10	W5 Tire Pressure
OBDPidReserved11	N/A
OBDPidReserved12	N/A
OBDPidReserved13	Driver Seatbelt Stat
OBDPidReserved14	N/A
OBDPidReserved15	N/A

5.3.15. BuildAndSend Message ACK

Byte(s)	Value	Type	Description
0-1	0x8888	N/A	The first 2 bytes of the ACK must be set to 0x8888.
2-3	PacketSerialNum	16-bit unsigned int, big-endian byte order	This value is the unique packet serial/sequence number (Packet recipe ID: 0x05).

5.3.16. Packet Creation Characteristics

- A message is created from a single packet recipe and may be appended with fields from a separate append recipe.
- The device can store 128 packet recipes (each containing up to 40 fields) and 16 append recipes (each containing up to 16 fields).
- Each packet/append recipe is assigned a packet ID (0-254; 0x0 to 0xfe).
- Each append recipe is assigned a range of reason codes.

Example: It's possible to create an append recipe adding fields [x, y, z] to the end of the message. The designer can assign a range, e.g. 4 to 7, to this append recipe. Any time the script calls BuildAndSendMsg() with reason code 4, 5, 6, or 7, [x, y, z] fields are appended to the message. It is up to the customer to design and group reason codes and use them in the script to append desired data in certain situations.

5.3.17. Packet Creation Recipe Configuration

Command Structure:

```
:wycfg pcr[<slot_index>] "<recipe_hex_string>"
```

<slot_index> is in range of 0 to 127

"<recipe_hex_string>" is a quoted string of hexadecimal bytes (represented by two ASCII characters)

String Structure:

```
<packet_id><num_fields><field_0><field_1>...<field_N>
```

5.3.18. Packet Creation Append

Command Structure:

```
:wycfg pca[<slot_index>] "<append_hex_string>"
```

<slot_index> is in range of 0 to 15

<recipe_hex_string> is a quoted string of hex bytes (represented by two ASCII characters)

String Structure:

<reason_lo><reason_hi><flags><num_fields><field_0><field_1>...<field_N>

<reason_lo> to <reason_hi> is the reason code range



NOTE: A single reason code assignment is possible when <reason_lo> = <reason_hi>

<flags> is in range of 0 to 1. See <flags> range table below:

<flags> range table:

Range	Description
0	Store the appended fields with the message in non-volatile memory but do NOT send the appended fields to the backend server.
1	Store the appended fields with the message in non-volatile memory AND send the appended fields to the backend server.

5.3.19. Examples

Packet Recipe Creation

The table below represents three example packet recipe configurations (one on each row). The recipes are stored in the first three recipe slots (out of 128).

Packet ID	Fields	Command
0x0	PacketID, DeviceID, UnixTime	:wycfg pcr[0] "0003010306"
0x14	PacketID, DeviceID, Latitude, Longitude, Altitude	:wycfg pcr[1] "14050103070809"
0xfe	PacketID, DeviceID, Hdop, and NumSats	:wycfg pcr[2] "fe0401031213"

Append Recipe Creation

Reason Code Range	Fields and description	Command
0 to 13	Do NOT Send appended fields to backend server Field: GarminPktResp	:wycfg pca[0] "000d000127"
30 to 38	Send appended fields to backend server; Fields: MotionStatus, EngineStatus, DerivedOdometer	:wycfg pca[1] "1e2601032c2d2e"

6. COMMAND LANGUAGE

6.1. OVERVIEW

The XT6300 device utilizes commands that can be sent over various methods such as USB, RS232, and SMS. This command language can be utilized to configure nearly all the device's parameters and morph its functionality to fit the requirements of a client. The implemented command language has been designed to be easy to learn/remember, human readable, and flexible. Each command is sent to the device in ascii format, and the device responds through the same medium that the command was sent with.

The syntax for a command is shown below:

```
:<verb><module><object> [<arg1>, <arg2>, ....<argN>]
```

Valid characters for verb/module and objects are as follows:

```
"_abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789"
```

The subsequent sections illustrate the various components that comprise a command. Each of these components has a set of rules that dictate how it should be formatted and what it represents.

6.2. COMMANDS VIA UDP

6.3. VERB AND MODULE

The following are rules regarding the syntax of verb and module codes:

- Always single character codes
- Characters MAY be separated by one valid separator: "[space],()[tab]"

6.4. OBJECT

The following are rules regarding the syntax of object codes:

- Must be followed by a valid separator: "[space],()[tab]"
- Unquoted strings of undefined length
- First character must NOT be a numerical digit

6.5. ARGUMENTS

The following are rules regarding the syntax of arguments:

- Arguments have multiple valid formats:
 - Strings
 - Strings representing enumerated values of valid chars
 - Hexadecimal or decimal values
- Arguments must be separated by a single valid separator: "[space],()[tab]"
- Arguments must not contain the characters: ":[]#<>"
- Anything following the symbol '#' is considered a comment and ignored by the interpreter
- Decimal values have an optional prefix of '+' or '-' to designate their sign
- Hex values can be denoted by prefixing '0x' or '0X'
- Octal values can be denoted by prefixing '0'

6.5.1. Verb Codes

Verb	Code
Clear	c
Create	a
Enable	e
Disable	d
Query	q
Run	u
Read	r
Restore	o
Save	v
Send	x
Set	s
Toggle	t
Upgrade	g
Write	w

6.5.2. Module Codes

Module	Code
ADC	a

Module	Code
Bluetooth	b
TPS	c
Device	d
Geofence	e
Flash	f
GPS	g
Watchdog	h
GPIO	i
Third Party	j
Backend	k
CMDLang	l
Motion	m
Garmin	n
OBD	o
Processor Flash	p
Interpreter	r
Buzzer	s
Telematics/Cell	t
USB	u
Aux Passthru	v
One-Wire-Bus	w
Accelerometer	x
System	y
Zigbee	z
Main Processor	1
2nd Processor	2
available	q3-0

6.5.3. Object Codes

Object	Code
Bootloader	bl
Chipflash	chip
Config	cfg
Command	cmd
Debug	dbg
Debug Mask	dm
DM Msg	dmmmsg
Eeprom	ee
Feature	f

Object	Code
Firmware	fw
Heap	hp (OR) heap
Info	i (OR) info
Monitor	m
Memory	mem
Mode	md (OR) mode
Message	msg
Option	opt
OTA	ota
Override	ovrd
Properties	p (OR) prop
RAM Console	rc
Record	rec
Script	scr (OR) script
Simulation	sim
Stack	stk (OR) stack
State	st (OR) state
Table	tab
Task	tsk (OR) task
Test	t (OR) test
Truststore	trst
Value	val

6.6. DEVICE COMMAND MODULES

6.6.1. ADC

Command	Function
:raval	Read ADC values

6.6.2. Accelerometer

Command	Function
:rxval	Read current X, Y, and Z accelerations
:rxval coefs	Read current filter coefficients
:rxval rot	Read current reorientation angles (radians)
:wxval coefs <num_stages> <coef1> <coef2> ... <coefX>	Write the number of filter stages and all filter coefficients
:wxval coef <coef_index>	Write filter coefficient at index X

Command	Function
<coef>	
:qxi	Query accelerometer info
:exm	Enable accelerometer debug channel monitoring
:dxm	Disable accelerometer debug channel monitoring
:ext	Enable accelerometer motion test
:dxt	Disable accelerometer motion test
:uxcmd findz	Force reorientation to z-axis
:uxcmd findxa	Force reorientation to x-axis using acceleration
:uxcmd findxd	Force reorientation to x-axis using deceleration
:uxcmd voidrot	Force void of current reorientation

6.6.3. Buzzer

Command	Function
:uscmd play <song_index> <volume> <loops>	Play a buzzer song out of the config
:uscmd stop	Stop a buzzer song that is currently playing

6.6.4. Backend

Command	Function
:ekm	Enable Backend debug channel monitoring
:dkm	Disable Backend debug channel monitoring
:rkval ack_timeout	Read what the current ACK timeout value is set to (ms)
:wkval ack_timeout <timeout_val>	Set the ACK timeout value (ms)
:ukcmd tunnel <"cmd" or "dm"> <dst_index>	Send "Hello" out the command or DM port to the address configured at that dst_index (see network tunneling)

6.6.5. Command Language

Command	Function
:wlopt echo <0 1>	Write cmdlang option "echo" to on or off

6.6.6. Third Party

Command	Function
:ejm	Enable Third Party debug channel monitoring
:djm	Disable Third Party debug channel monitoring
:djdbg	Disable Third Party XVBMG debug trace
:ejdbg	Enable Third Party XVBMG debug trace
:qji	Print out parameter tag information
:qji diag	Print out tag transaction diagnostic info
:gjfw x <serial_port_id>	Upgrade Third Party via xmodem
:gjscr x <serial_port_id> <flag - optional>	Upload VIS file via xmodem (flag = 1; erases all VIS's first)
:ujcmd tag <tag id> <ip_dst>	Run XVBMG on tag id; if no VIS file found, downloads VIS first (:edm & :ekm)
:ujcmd init	Initialize Third Party functions (:edm)
:ujcmd exec	Displays Third Party FW version (:edm)

6.6.7. Device

Command	Function																																	
:rdval reset	Read device reset reason																																	
	Reason codes:																																	
	<table><tr><th>Code</th><th>Reason</th><th>Description</th></tr><tr><td>0</td><td>Power applied</td><td>Main supply voltage connected to device</td></tr><tr><td>1</td><td>Low Power</td><td>Reset caused by the micro Hibernation module (not utilized)</td></tr><tr><td>2</td><td>Watchdog</td><td>Reset caused by the micro Watchdog module.</td></tr><tr><td>3</td><td>Software</td><td>Reset caused by our device firmware.</td></tr><tr><td>4</td><td>User</td><td>Main micro reset pin toggled (not utilized)</td></tr><tr><td>5</td><td>Brownout</td><td>Reset caused by the micro BOR module</td></tr><tr><td>6</td><td>Supply Monitor</td><td>Not implemented.</td></tr><tr><td>7</td><td>Factory Reset</td><td>Reset caused by the micro Hardware System Service Request.</td></tr><tr><td>8</td><td>System Error</td><td>Not implemented.</td></tr><tr><td>9</td><td>Unknown</td><td>Catch for any other micro reset source we have not defined.</td></tr></table>	Code	Reason	Description	0	Power applied	Main supply voltage connected to device	1	Low Power	Reset caused by the micro Hibernation module (not utilized)	2	Watchdog	Reset caused by the micro Watchdog module.	3	Software	Reset caused by our device firmware.	4	User	Main micro reset pin toggled (not utilized)	5	Brownout	Reset caused by the micro BOR module	6	Supply Monitor	Not implemented.	7	Factory Reset	Reset caused by the micro Hardware System Service Request.	8	System Error	Not implemented.	9	Unknown	Catch for any other micro reset source we have not defined.
Code	Reason	Description																																
0	Power applied	Main supply voltage connected to device																																
1	Low Power	Reset caused by the micro Hibernation module (not utilized)																																
2	Watchdog	Reset caused by the micro Watchdog module.																																
3	Software	Reset caused by our device firmware.																																
4	User	Main micro reset pin toggled (not utilized)																																
5	Brownout	Reset caused by the micro BOR module																																
6	Supply Monitor	Not implemented.																																
7	Factory Reset	Reset caused by the micro Hardware System Service Request.																																
8	System Error	Not implemented.																																
9	Unknown	Catch for any other micro reset source we have not defined.																																

Command	Function																						
:rdval serial	Read device serial number																						
:rdval uptime	Read device uptime (seconds since reset)																						
:rdval wake	Read device wake reason (only accurate if device has slept)																						
:wdval serial "<serial_no>"	Write the device serial number																						
:qdi	Query device info ("vtu query")																						
:qdi cell	Query cellular device info ("vtu query modem")																						
:qdi gps	Query GPS device info ("vtu query gps")																						
:udcmd checkin	Send a Device Manager Check-In message to dst[9]																						
:udcmd factory	Reset device to factory state																						
:udcmd factory keys	Reset device to factory state plus revert AES keys to defaults. This is the only command available outside the AES console login.																						
:udcmd dslp <mask> <minutes>	Enter deep sleep with <mask> wake setting Mask Values: <table> <tr> <th>Hex</th><th>Description</th></tr> <tr> <td>0x0000</td><td>Ring</td></tr> <tr> <td>0x0001</td><td>Ignition</td></tr> <tr> <td>0x0002</td><td>Input 1</td></tr> <tr> <td>0x0004</td><td>Input 2</td></tr> <tr> <td>0x0008</td><td>Input 3</td></tr> <tr> <td>0x0010</td><td>Input 4</td></tr> <tr> <td>0x0020</td><td>Time</td></tr> <tr> <td>0x0040</td><td>Motion</td></tr> <tr> <td>0x0080</td><td>Battery</td></tr> <tr> <td>0x0100</td><td>Unplug</td></tr> </table>	Hex	Description	0x0000	Ring	0x0001	Ignition	0x0002	Input 1	0x0004	Input 2	0x0008	Input 3	0x0010	Input 4	0x0020	Time	0x0040	Motion	0x0080	Battery	0x0100	Unplug
Hex	Description																						
0x0000	Ring																						
0x0001	Ignition																						
0x0002	Input 1																						
0x0004	Input 2																						
0x0008	Input 3																						
0x0010	Input 4																						
0x0020	Time																						
0x0040	Motion																						
0x0080	Battery																						
0x0100	Unplug																						
:udcmd slp <mask> <minutes>	Enter sleep with <mask> wake setting Mask Values: <table> <tr> <th>Hex</th><th>Description</th></tr> <tr> <td>0x0000</td><td>Ring</td></tr> <tr> <td>0x0001</td><td>Ignition</td></tr> <tr> <td>0x0002</td><td>Input 1</td></tr> <tr> <td>0x0004</td><td>Input 2</td></tr> <tr> <td>0x0008</td><td>Input 3</td></tr> <tr> <td>0x0010</td><td>Input 4</td></tr> <tr> <td>0x0020</td><td>Time</td></tr> <tr> <td>0x0040</td><td>Motion</td></tr> <tr> <td>0x0080</td><td>Battery</td></tr> <tr> <td>0x0100</td><td>Unplug</td></tr> </table>	Hex	Description	0x0000	Ring	0x0001	Ignition	0x0002	Input 1	0x0004	Input 2	0x0008	Input 3	0x0010	Input 4	0x0020	Time	0x0040	Motion	0x0080	Battery	0x0100	Unplug
Hex	Description																						
0x0000	Ring																						
0x0001	Ignition																						
0x0002	Input 1																						
0x0004	Input 2																						
0x0008	Input 3																						
0x0010	Input 4																						
0x0020	Time																						
0x0040	Motion																						
0x0080	Battery																						
0x0100	Unplug																						

Command	Function
:edm	Enable device debug channel monitoring
:ddm	Disable device debug channel monitoring

6.6.8. Flash

Command	Function
:efm	Enable Flash debug channel monitoring
:dfm	Disable Flash debug channel monitoring
:cfmem [all <addr> <size>]	all = erase entire flash chip, <addr>,<size> NOTE: Must be multiples of 0x1000
:rfmem <addr> <size>	Read <size> bytes of flash memory at address <addr> NOTE: <addr> is in decimal
:wfmem <addr> "<hex_string>"	Write "<hex_string>" to address <addr>
:qfmem	Query flash memory layoutNOTE: Only available on debug devices
:qfi	Query flash info (device type)
:qfrec	Query number of pending messages (records)
:rfrec	Read pending messages (records)
:wfrec "<hex_string>"	Write flash message (record)
:qftab <x>	Query flash table (x = 2 for packet log)
:sftab <x> <sz>	Fill flash table x leaving only sz bytes available NOTE: Only available on engineering devices
:ufcmd reset	Erase pending messages
:ufcmd search <match> <size> <offset>	Search flash for a match

6.6.9. Garmin

Command	Function
:enm	Enable Garmin debug channel monitoring
:dnm	Disable Garmin debug channel monitoring
:xnmsg <pid> "<ascii_hex>" <flags>	Send a payload destined for Garmin device (needs script trigger/action)

6.6.10. Geofence

Command	Function
:eem	Enable Geofence debug channel monitoring
:dem	Disable Geofence debug channel monitoring
:redm	Read GPS debug mask
:wedm <mask>	Write the GPS debug mask

6.6.11. GPIO

Command	Function
:qii	Query GPIO info (GPIO pin IDs and default states)
:rival <pin_id>	Read state of GPIO pin
:wival <pin_id> <0 1 x>	Control the state of GPIO pin (x = give control back to micro)

6.6.12. GPS

Command	Function
:rgdm	Read GPS debug mask
:wgdm <mask>	Write the GPS debug mask
:egm	Enable GPS debug channel monitoring
:dgm	Disable GPS debug channel monitoring
:egt	Enable GPS production test
:dgt	Disable GPS production test
:egcmd	Start GPS
:dgcmd	Stop GPS
:qgi	Query GPS info
:ugcmd reset	GPS module reset

6.6.13. GPS Simulator

Command	Function												
:sgsim config <lat> <long> <heading> <speed>	GPS simulator initialize/enable:												
	<table> <tr> <th>Value</th><th>Description</th></tr> <tr> <td>0</td><td>None</td></tr> <tr> <td>1</td><td>Unlocked</td></tr> <tr> <td>2</td><td>Locked</td></tr> <tr> <td>3</td><td>Sleep</td></tr> <tr> <td>4</td><td>LockedGood</td></tr> </table>	Value	Description	0	None	1	Unlocked	2	Locked	3	Sleep	4	LockedGood
Value	Description												
0	None												
1	Unlocked												
2	Locked												
3	Sleep												
4	LockedGood												

Command	Function												
:sgsim coord <lat> <long>	GPS simulator force coordinates												
:sgsim speed <speed>	GPS simulator force speed												
	Values are in tenths (e.g. 300 = 30 mph)												
:sgsim head <heading>	GPS simulator force heading												
	Values are in tenths (e.g. 900 = 90 degrees)												
:sgsim lock <lock>	GPS simulator force lock state:												
	<table> <tr> <th>Value</th><th>Description</th></tr> <tr> <td>0</td><td>None</td></tr> <tr> <td>1</td><td>Unlocked</td></tr> <tr> <td>2</td><td>Locked</td></tr> <tr> <td>3</td><td>Sleep</td></tr> <tr> <td>4</td><td>LockedGood</td></tr> </table>	Value	Description	0	None	1	Unlocked	2	Locked	3	Sleep	4	LockedGood
Value	Description												
0	None												
1	Unlocked												
2	Locked												
3	Sleep												
4	LockedGood												
:sgsim hdop <hdop>	GPS simulator force hdop												
:sgsim clear	Disable GPS simulator												

6.6.14. Cellular

Command	Function																								
:rtdm	Read transport debug mask																								
:wtdm <mask>	Write the transport debug mask																								
	mask Values:																								
	<table> <tr> <th>Hex</th><th>Description</th></tr> <tr> <td>0x000</td><td>GSMDDebug_None</td></tr> <tr> <td>0x001</td><td>GSMDDebug_State</td></tr> <tr> <td>0x002</td><td>GSMDDebug_Command</td></tr> <tr> <td>0x004</td><td>GSMDDebug_Response</td></tr> <tr> <td>0x008</td><td>GSMDDebug_Parse</td></tr> <tr> <td>0x010</td><td>GSMDDebug_ResponseHex</td></tr> <tr> <td>0x020</td><td>GSMDDebug_Mux</td></tr> <tr> <td>0x040</td><td>GSMDDebug_Backend</td></tr> <tr> <td>0x080</td><td>GSMDDebug_Mno</td></tr> <tr> <td>0x100</td><td>GSMDDebug_URC</td></tr> <tr> <td>0x200</td><td>GSMDDebug_PDU</td></tr> </table>	Hex	Description	0x000	GSMDDebug_None	0x001	GSMDDebug_State	0x002	GSMDDebug_Command	0x004	GSMDDebug_Response	0x008	GSMDDebug_Parse	0x010	GSMDDebug_ResponseHex	0x020	GSMDDebug_Mux	0x040	GSMDDebug_Backend	0x080	GSMDDebug_Mno	0x100	GSMDDebug_URC	0x200	GSMDDebug_PDU
Hex	Description																								
0x000	GSMDDebug_None																								
0x001	GSMDDebug_State																								
0x002	GSMDDebug_Command																								
0x004	GSMDDebug_Response																								
0x008	GSMDDebug_Parse																								
0x010	GSMDDebug_ResponseHex																								
0x020	GSMDDebug_Mux																								
0x040	GSMDDebug_Backend																								
0x080	GSMDDebug_Mno																								
0x100	GSMDDebug_URC																								
0x200	GSMDDebug_PDU																								
:etm	Enable transport debug channel monitoring																								

Command	Function																				
:dtm	Disable transport debug channel monitoring																				
:ett	Enable transport test mode ('ct' mode)																				
:ett direct	Connect serial console directly to cellular port																				
:dtt	Disable transport test mode ('ct')																				
:qti	Query transport info																				
:qti ip	Query cellular mobile IP																				
:utcmd factory	Set cellular module to factory default state																				
:utcmd reset	Queue a reset for the cellular module																				
:utcmd sleep	Stop cellular module																				
:utcmd wake	Start cellular module																				
:utota ftp "<filename with extension>" <type> <index> <rev>	Request FTP file transfer (only on SARA-R410M and LARA-R202 (HB1) cell modules currently)																				
	Arg Details:																				
	<table> <tr> <th>Type</th><th>Description</th></tr> <tr> <td>0</td><td>Bootloader firmware</td></tr> <tr> <td>1</td><td>Application firmware</td></tr> <tr> <td>2</td><td>Script binary firmware</td></tr> <tr> <td>3</td><td>Parameter text file</td></tr> <tr> <td>6</td><td>TPS</td></tr> <tr> <td>8</td><td>Overlay text file</td></tr> <tr> <td>10</td><td>Third party code</td></tr> <tr> <td>11</td><td>Third party data</td></tr> <tr> <td>12</td><td>FTP update</td></tr> </table>	Type	Description	0	Bootloader firmware	1	Application firmware	2	Script binary firmware	3	Parameter text file	6	TPS	8	Overlay text file	10	Third party code	11	Third party data	12	FTP update
Type	Description																				
0	Bootloader firmware																				
1	Application firmware																				
2	Script binary firmware																				
3	Parameter text file																				
6	TPS																				
8	Overlay text file																				
10	Third party code																				
11	Third party data																				
12	FTP update																				
	<index> only applies to overlays																				
	<rev> is the user determined revision value																				
:utota https "<host>" "<path>" <port> <type>[[<index>] "<filename>"] <rev>	Request HTTPS file transfer (only on AAb6 currently)																				
	"<host>" is an https server string																				
	"<path>" is an https path to file including file extension																				
	<port> is an https server port																				
	<type> is an enumerated value:																				
	<table> <tr> <th>Type</th><th>Description</th></tr> <tr> <td>0</td><td>Bootloader firmware</td></tr> <tr> <td>1</td><td>Application firmware</td></tr> <tr> <td>2</td><td>Script binary firmware</td></tr> <tr> <td>3</td><td>Parameter text file</td></tr> <tr> <td>6</td><td>TPS</td></tr> <tr> <td>8</td><td>Overlay text file</td></tr> <tr> <td>10</td><td>Third party code</td></tr> <tr> <td>11</td><td>Third party data</td></tr> </table>	Type	Description	0	Bootloader firmware	1	Application firmware	2	Script binary firmware	3	Parameter text file	6	TPS	8	Overlay text file	10	Third party code	11	Third party data		
Type	Description																				
0	Bootloader firmware																				
1	Application firmware																				
2	Script binary firmware																				
3	Parameter text file																				
6	TPS																				
8	Overlay text file																				
10	Third party code																				
11	Third party data																				

Command	Function																		
:utota s3 "<full S3 presigned link>" <port> <type>[[<index>] "<filename>" <rev>]	Request HTTPS file transfer (only on AAb6 currently). "<full S3 presigned link>" https server string and path including file extension <port> is an https server port <type> is an enumerated value: <table> <tr> <th>Type</th><th>Description</th></tr> <tr> <td>0</td><td>Bootloader firmware</td></tr> <tr> <td>1</td><td>Application firmware</td></tr> <tr> <td>2</td><td>Script binary firmware</td></tr> <tr> <td>3</td><td>Parameter text file</td></tr> <tr> <td>6</td><td>TPS</td></tr> <tr> <td>8</td><td>Overlay text file</td></tr> <tr> <td>10</td><td>Third party code</td></tr> <tr> <td>11</td><td>Third party data</td></tr> </table> <index> only applies to overlays "<filename>" is arbitrary and optional name reported in :q1i response <rev> is the user determined revision value	Type	Description	0	Bootloader firmware	1	Application firmware	2	Script binary firmware	3	Parameter text file	6	TPS	8	Overlay text file	10	Third party code	11	Third party data
Type	Description																		
0	Bootloader firmware																		
1	Application firmware																		
2	Script binary firmware																		
3	Parameter text file																		
6	TPS																		
8	Overlay text file																		
10	Third party code																		
11	Third party data																		
:rtval mno	See Hardware ID and MNO select																		
:wtval mno <profile>																			

6.6.15. Interpreter

Command	Function												
:erm [<trigger_index>]	Enable interpreter debug channel monitoring [monitor only <trigger_id>]												
:drm	Disable interpreter debug channel monitoring												
:xrmsg <user_event_id>	Send a user_event to the interpreter												
:grscr x <serial_port_id> "<name>"	Upgrade interpreter script via xmodem (See TBD; formerly Load a Compiled Script to a Device)												
:rrscr	Read interpreter script												
:rrval odom <index>	Read an odometer <table> <tr> <th>Index</th><th>Description</th></tr> <tr> <td>0</td><td>GPS Trip 1</td></tr> <tr> <td>1</td><td>GPS Trip 2</td></tr> <tr> <td>2</td><td>OBD Trip</td></tr> <tr> <td>3</td><td>GPS Life</td></tr> <tr> <td>4</td><td>OBD Life</td></tr> </table>	Index	Description	0	GPS Trip 1	1	GPS Trip 2	2	OBD Trip	3	GPS Life	4	OBD Life
Index	Description												
0	GPS Trip 1												
1	GPS Trip 2												
2	OBD Trip												
3	GPS Life												
4	OBD Life												

Command	Function												
:rrval flags	Read user script flags												
:rrval uservar <type> <index>	Read a uservar												
:rrval timer <index>	Read remaining time												
:rrval gfnstat	Read geofence active & inside status bitfields												
:wrval odom <index> <value>	<value> must be larger than current value:												
	<table> <tr> <th>Index</th><th>Description</th></tr> <tr> <td>0</td><td>GPS Trip 1</td></tr> <tr> <td>1</td><td>GPS Trip 2</td></tr> <tr> <td>2</td><td>OBD Trip</td></tr> <tr> <td>3</td><td>GPS Life</td></tr> <tr> <td>4</td><td>OBD Life</td></tr> </table>	Index	Description	0	GPS Trip 1	1	GPS Trip 2	2	OBD Trip	3	GPS Life	4	OBD Life
Index	Description												
0	GPS Trip 1												
1	GPS Trip 2												
2	OBD Trip												
3	GPS Life												
4	OBD Life												
:crval tripodom <index>	Clear a trip odometer:												
	<table> <tr> <th>Index</th><th>Description</th></tr> <tr> <td>0</td><td>GPS Trip 1</td></tr> <tr> <td>1</td><td>GPS Trip 2</td></tr> <tr> <td>2</td><td>OBD Trip</td></tr> </table>	Index	Description	0	GPS Trip 1	1	GPS Trip 2	2	OBD Trip				
Index	Description												
0	GPS Trip 1												
1	GPS Trip 2												
2	OBD Trip												
:wrval flags <value>	Write user script flags												
:wrval uservar <type> <index> <value>	Write a uservar												
:drtask	Suspend interpreter task												
:ertask	Resume interpreter task												
:urcmd bs <packet id> <reason code> <dst index> <ack> <qty>	Force a build and send of a packet (any reason code less than 0 or greater than 255 will wrap)												
	NOTE: <qty> only available on engineering devices												
:rrval linkver	Read linkage version information (See TBD; formerly Linkage Version)												

6.6.16. Main Processor

Command	Function
:d1dbg [<lock_code>]	Lock-out JTAG (disable debug); two stage command: 1. Issue :d1dbg (returns <lock_code>) 2. Issue :d1dbg <lock_code>
:g1bl x <serial_port_id>	Upgrade main bootloader via xmodem
:g1fw x <serial_port_id>	Upgrade main firmware via xmodem
:q1fw/:q1i	Query main firmware
:u1cmd reset	Reset main processor

6.6.17. One-Wire Bus

Command	Function
:wwdm <mask>	Sets one-wire bus debug mask
:wwdm ts <mask>	Sets one-wire bus temp sensor debug mask
:wwdm drv <mask>	Sets one-wire bus driver id debug mask
:ewm	Enable one-wire bus debug channel monitoring
:dwm	Disable one-wire bus debug channel monitoring
:rwi <channel 0/1>	Read one-wire bus info
:rwval <channel 0/1>	Read one-wire bus temp sensor value

6.6.18. OBD

Command	Function
:qoi info	Query OBD vehicle information
:qoi status	Query OBD status information
:qoi persist	Query OBD persisted information
:qoi trip	Query OBD trip information
:qoi ss	Query OBD snapshot parameter information
:qoi ss2	Query OBD snapshot additional parameter information
:qoval	Query OBD calculated information
:qoi viol	Query OBD Violations
:qost	Query OBD state
:roval dtc	Read OBD DTC List
:woval p <protocol> <cli>	Set OBD protocol to a com logical link
:woval dtc <mil value> <total dtc> <total ecu count>	Manually write DTC data (defaults to CAN protocol); if all fields are 0, DTCs are cleared
:wocmd debug <0/1>	Disable/enable obd debug log
:wocmd debug erase	Erase OBD debug log
:wocmd debug lock	Lock OBD debug off

6.6.19. System

Command	Function
:cycfg	Clear system configs (set all parameters to default/RAM only and use :vycfg to make permanent)
:cycfg recipes	Clear system configs (set all parameters to default/RAM only and use :vycfg to make permanent); clear all recipes/appends from flash
:ehm	Enable watchdog timer debug monitor
:gycfg x <serial_port_id> "<name>"	Upgrade paramset via xmodem; "<name>" is the version of the paramset

Command	Function										
:rycfg ...	Read system config										
:wycfg ...	Write system config										
:qycfg ...	Query parameter information										
:uycfg revert	Revert all system configs in RAM (reload from FLASH; must be used before :vycfg)										
:vycfg	Save system configuration										
:uyscr x <serial_port_id>	Run system script (after upload via xmodem)										
:ryscr	Read system script (not the interpreter script)										
:wyscr	Write a new system script (script created to recreate current parameters)										
:wyval persist	Force a save of persist data										
:ryval stack	Print out stack high water marks										
:ryval time	Print out current network, gps, and system times										
:qyi	Print out system information										
:uycmd resync <new_system_time_s>	Force system time to resync with available source. If time argument is passed, use as new system time with lower priority than internal sources										
:ryval hwid	Print out the hardware ID value Defined HWIDs: <table> <tr> <th>Hex</th><th>Description</th></tr> <tr> <td>0x01</td><td>Hwid_2400_A</td></tr> <tr> <td>0x02</td><td>Hwid_6370_A</td></tr> <tr> <td>0x03</td><td>Hwid_6374_A</td></tr> <tr> <td>0xff</td><td>Hwid_6300_B</td></tr> </table>	Hex	Description	0x01	Hwid_2400_A	0x02	Hwid_6370_A	0x03	Hwid_6374_A	0xff	Hwid_6300_B
Hex	Description										
0x01	Hwid_2400_A										
0x02	Hwid_6370_A										
0x03	Hwid_6374_A										
0xff	Hwid_6300_B										
:wyval hwid <value>	Set the hardware ID value										
:wyval key <key_index> <okok> <nkok> <oknk>	Change AES key, where key_index: <table> <tr> <th>Index</th><th>Description</th></tr> <tr> <td>0</td><td>Update</td></tr> <tr> <td>1</td><td>Customer</td></tr> <tr> <td>2</td><td>Transport</td></tr> </table>	Index	Description	0	Update	1	Customer	2	Transport		
Index	Description										
0	Update										
1	Customer										
2	Transport										
:wyval ptk <key_index> <plaintext_key>	Change AES key, where key_index: <table> <tr> <th>Index</th><th>Description</th></tr> <tr> <td>0</td><td>Update</td></tr> <tr> <td>1</td><td>Customer</td></tr> <tr> <td>2</td><td>Transport</td></tr> </table> The desired key value is not encrypted. This command is only available via physical connection to the device.	Index	Description	0	Update	1	Customer	2	Transport		
Index	Description										
0	Update										
1	Customer										
2	Transport										

6.6.20. TPS

Command	Function
:ecm	Enable TPS debug prints
:dcm	Disable TPS debug prints
:wcdm <mask>	Set debug mask
:wcsim hdr	Write into the TPS simulator header
:wcsim tags <protocol> <link>	Write a TPS simulator tag
:rctest vin	Check if the current tag VIN matches the vehicle VIN
:rctest file	Checks if there is a valid tps file
:rctest hdr	Read TPS header
:rctest tag <tag_id>	Read TPS tag info
:qci tag	
:rctest data	Read TPS tag data
:qci data	
:qci block <block_id>	Read TPS block
:qccmd sync	Sync TPS file with AAb3 target servers
:gcfw x <serial_port_id> "<name>"	Upgrade TPS file via xmodem

6.6.21. Aux Passthru

Command	Function								
:qvi	Print info								
:evm	Enable aux passthru debug channel monitoring								
:dvm	Disable aux passthru debug channel monitoring								
:wvdm <mask_base10>	Write (configure) aux passthru debug mask								
	Mask Values:								
	<table> <tr> <th>Value</th><th>Description</th></tr> <tr> <td>0</td><td>None</td></tr> <tr> <td>1</td><td>Payload</td></tr> <tr> <td>2</td><td>Scratchpad</td></tr> </table>	Value	Description	0	None	1	Payload	2	Scratchpad
Value	Description								
0	None								
1	Payload								
2	Scratchpad								
:rvdm	Read (print out) aux passthru debug mask								
:xvmsg <bytes>	Send a packet to be transmitted out over the aux port								
	If the <bytes> field begins with "0x", the message is sent in hex;								
	if the <bytes> field does not begin with "0x", the message is sent in ASCII								

6.7. CALIFORNIA PROPOSITION 65 WARNING



WARNING: This product can expose you to chemicals including Nickel (Metallic), which is known to the State of California to cause cancer and Bisphenol A (BPA), which is known to the State of California to cause birth defects or other reproductive harm.

For more information go to www.P65Warnings.ca.gov