XT6300
Product Manual

Firmware Version: 1133NA1

Revised April 03, 2023

Table of Contents

1. Release Notes

1.1. Documentation Changes

- Added new :utota ftp and :utota http commands.

2. Functional Description

2.1. Overview

The XT6300 is a device that interfaces with a vehicle's on-board diagnostics port to monitor and record vehicle activity. The XT6300 was developed primarily for clientele with large automotive fleets such as vehicle rental companies or courier delivery services. The XT6300 uses an OBD-II interface, a cellular modem, a Bluetooth modem, a GPS modem, a piezo speaker, LED's, and an accelerometer to collect information that may be of interest to a client's application. Alongside these build in services, a SWI and UART interface are available to interact with external devices to gather additional information.

One key benefit of the XT6300 is the ability to easily adjust the device's operation to suit an individual client's desires. The XT6300 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. By configuring custom system parameters and interpreter scripts, not only can device functionality be adjusted, but entirely new functionality can be added to the device. In theory, the XT6300 can emulate any other Xirgo device if properly configured through the system script.

2.2. Mechanical

2.2.1. XT6300

Mechanical	
Dimensions	3.35" x 3.1" x 0.8" (8.5 cm x 7.9 cm x 2.0 cm)
Weight	3 oz. (85 g) 3.2 oz. (92g) with internal battery
Physical Connections	14-pin micro-fit connector 24-pin micro-fit connector
Operating Temperature	-30°C to 75°C (No internal battery) -30°C to 60°C (Internal battery)

2.2.2. XT2400

Specification	Description
Dimensions	2.4” x 1.8” x 1.1” (5.9 x 4.6 x 2.8 cm)
Weight	< 2 oz. (34 g)
Physical Connection	OBD J1962
Operating Temperature	(-30°C to 70°C)
Power Requirement	DC 8 – 24 V

3. Device Setup

3.1. 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.

4. Interfaces

4.1. UART

XT6300 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.

4.2. LED Behavior

4.2.1. Amber LED

Primary Operation

Indicate Cellular Communication Status

Cell Status	LED Status
Off (Low power mode/ sleep)	Off
Searching for network	Fast Blink (5 Hz)
Registered home/roaming	On

Secondary Operation

Indicate Active Inputs (1-4)

When any or all user inputs (1-4) are active, the LED will flash (1 Hz) the amount of times corresponding to the active input index, pause, then repeat.

When multiple inputs are active, the LED will flash (1 Hz) the input index for each one that is active with a pause between to differentiate.

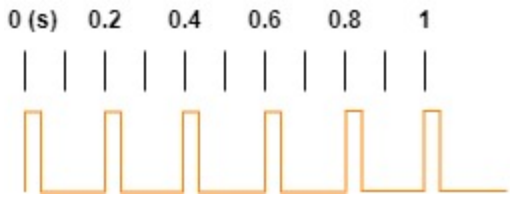
When all active inputs have been indicated, the LED will pause before repeating everything again. If no user inputs are active, the LED will revert to Primary Operation.

Override Operation

When ignition state is off, the LED will flash continuously at 0.5 Hz. When ignition state is on, the LED will revert to Primary Operation.

When ignition state is off and there is no cellular registration, the LED will double-flash at 0.33Hz. When the ignition state is off and there is cellular registration, the LED will flash continuously at 0.5 Hz. When ignition state is on, the LED will revert to Primary Operation.

Example: The diagram below shows the amber LED blink pattern indicating cellular searching for network.



4.2.2. Green LED

Primary Operation

Indicate GPS Lock Status

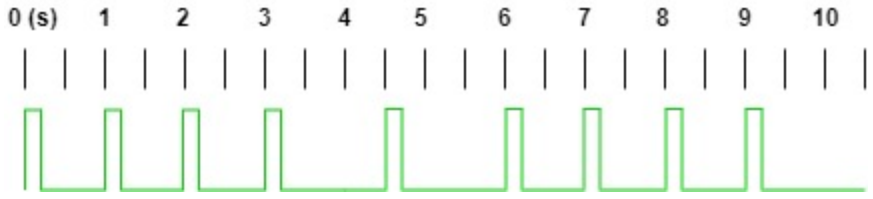
GPS Lock Status	LED Status
Off (sleep) or unlocked	Off
Locked	On for 2 seconds before transitioning to secondary operation

Secondary Operation

Indicate Locked Current Satellite Count

When GPS solution is Locked Good and the Primary Operation has completed, the LED will flash (1Hz) the amount of times corresponding to the number of satellites utilized in the current GPS solution, with a pause before repeating.

Example: LED Blink Pattern (after GPS solution is Locked Good and Primary Operation has completed) indicating number of satellites utilized in the current GPS solution. The example below shows the green LED indicating four satellites utilized in the GPS solution with a single blip at approximately 4.5 seconds before repeating.



4.2.3. Blue LED

Primary Operation

Indicate bluetooth connection status

BT Status	LED Status
Off or unconnected	Off
Connected	Slow blink (0.5Hz)

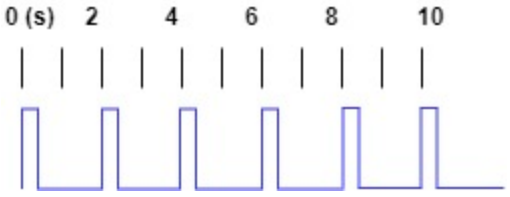
Secondary Operation

Indicate System Module Faults

If a system module fault is detected in firmware, the LED will flash at 1 Hz, the number of times corresponding to the fault index defined below, with a pause before repeating.

LED Blinks	Description
1	TPS module fault
2	Cellular module fault
3	GPS module fault
4	N/A
5	Bluetooth Module fault
6	Accelerometer module fault
7	OBD module fault

Example: The diagram below expresses a blue LED blink pattern indicating Bluetooth connection.



5. BLUETOOTH®/BLE

5.1. OVERVIEW

The XT6300 uses Bluetopia provided by StoneStreet one; the GAP and GATT profiles are of specific interest.

5.1.1. Security

There are several layers

Target	Pairing	Authentication	Encryption	Android/iOS	Advertising Name
AAb1	Just Works	None	None	Both	HarpBT_<serial>
AAb3	Just Works	SHA256 Client/Server	AES-CBC	Both	HarpBT_<serial> or Configurable
AAb4	N/A	N/A	N/A	None	N/A
AAb5	Just Works	None	None	Both	Link2_<serial>
AAb6	Just Works	None	None	Both	HarpBT_<serial> or Configurable

5.1.2. Definitions

Keyword	Description
Pairing	Bluetooth LE standard mechanism to exchange encryption keys, primarily handled by Bluetooth stack on client (phone or tablet). OS may prompt user for permission to pair and/or for a passkey.
Authentication	Custom service providing another layer of security; the XT6300 won't let client read/write without completing the authentication process.
Encryption	Another layer of encryption (AES) on top of Bluetooth. Characteristics are expanded out to a multiple of 16 bytes.

Keyword	Description
Just Works	Mechanism in simple secure pairing (SSP) that allows for devices to pair with no explicit user authentication
Passkey	Authentication used when pairing with a remote device; device displays a number the user must enter using the keypad on the remote device

5.2. AAB1/AAB6 BLE

5.2.1. HOS Service UUID

The HOS service does not have encrypted characteristics

Description	UUID
HOS	a59611ba-78b7-4fd2-96fb-9b0f66d2311e

5.2.2. HOS Service UUID List

Descriptions	UUID
GPS Info	e7737830-1018-11e6-a148-3e1d05defe78
Speed	5eed6ea2-0390-11e5-8418-1697f925ec7b
True Odo	5eed6d4e-0390-11e5-8418-1697f925ec7b
Derived Odo	5eed6719-0390-11e5-8418-1697f925ec7b
Eng/Ign	5eed665c-0390-11e5-8418-1697f925ec7b
RPM	31e6e24b-f2fb-4bb9-a16b-9d17a9c4e4ad
Protocol	5eed6477-0390-11e5-8418-1697f925ec7b
VIN	a9b9f487-5e60-43d5-a249-4d1d3f317d7e
Eng Hours	95dbace5-fcee-467f-bbe9-fe42e195bb04
Moving	5eed659a-0390-11e5-8418-1697f925ec7b
ECM	5eed70e6-0390-11e5-8418-1697f925ec7b
Fuel Level	1c3917a6-7d33-4152-9a33-858b6f1fc99b
Eng Run Time	b71eb3aa-4ed7-4990-ab22-ced09cf58f34
Eng Hours Seconds	87e22dd2-e27b-4094-968a-03e1537e7eb7
External Power	0dd2af9d-7220-474f-8b34-4a94a6a97498
Start Trip GPS info	d7b083dd-bf65-44e5-9fa1-959bddf2af0b
Start Trip True Odo	ba4ac777-52ec-4133-9f18-adad8b98f142
Start Trip Eng Hours Seconds	40418a44-9b8d-49f8-8ef2-afcea1643786
ECU Eng Hours Seconds	0d407e4c-39c9-11e8-b467-0ed5f89f718b
Start Trip ECU Eng Hours Seconds	d4fb1556-39d2-11e8-b467-0ed5f89f718b
Start Trip System Unix Time	0f834728-39d5-11e8-b467-0ed5f89f718b
System Unix Time	48969a7e-39d5-11e8-b467-0ed5f89f718b
Driver ID	66aa658a-0003-4cee-86f0-4c775bbe5a54
Start Trip Derived Odom	5ceb24b2-c273-40b4-b84b-f499e14c3f09

5.2.3. HOS Service Characteristics Table

Descriptions	Type	Size	Encr. Size	Data / Units
GPS Info	Read Notify *Type: Notify on lock or time change*	19	32	See GPS Info Structure below
Speed	Read	2+1	16	0.1 KPH + 1 byte updated indicator
True Odo	Read	4+1	16	Meters + 1 byte updated indicator
Derived Odo	Read	4+1	16	Meters + 1 byte updated indicator
Eng/Ign	Read Notify *Type: Notify on change*	1	16	Index: Description 0: Engine off 1: Engine on
RPM	Read	4+1	16	RPM + 1 byte updated indicator
Protocol	Read	1	16	Index: Description 0: None 1: J-Bus 2: OBDII
VIN	Read	18	32	Null Terminated String
Eng Hours	Read	4	16	Hours (Only available on Heavy Duty Vehicles, won't default to persist, not commonly used by customers)
Moving	Read Notify *Type: Notify on change*	1	16	Index: Description 0: Not Moving 1: Moving (speed > 5KPH)
ECM	Read Notify *Type: Notify on change*	1	16	Index: Description 0: Inactive 1: Pending 2: Connected 3: Active 4: Disabled by cinfig
Fuel Level	Read	2	16	Tenths of a percent
Eng Run Time	Read	4	16	Seconds
Eng Hours Seconds	Read	4	16	Seconds (Defaults to persist when on non-heavy-duty vehicles)

Descriptions	Type	Size	Encr. Size	Data / Units
External Power	Read	1	16	Index: 0 = No external power; 1 = External power
Start Trip GPS info	Read	19	32	See Below (but now historically flavored!)
Start Trip True Odo	Read	4+1	16	Meters + 1 byte updated indicator
Start Trip Eng Hours Seconds	Read	4	16	Seconds (Defaults to persist when on non-heavy-duty vehicles)
ECU Eng Hours Seconds	Read	4	16	ECU only (s)
Start Trip ECU Eng Hours Seconds	Read	4	16	ECU only (s)
Start Trip System Unix Time	Read	4	16	Seconds since epoch
System Unix Time	Read	4	16	Seconds since epoch
Driver ID	Read Write	4(R)/ 8(W)	16	ROM ID, i.e., 0x12345678. Driver ID values must be written in reverse byte order.
Start Trip Derived Odom	Read	4	16	Meters

5.2.4. GPS Info Structure

Name	Size	Type	Description
Unix Time	4	sint32	seconds since epoch
Lat	4	sint32	degrees * 1e6
Lon	4	sint32	degrees * 1e6
Speed	1	uchar	0.1 KPH
Heading	2	sint16	degrees * 10
HDOP	2	uint16	hdop * 10
Sats	1	uchar	Number of Satellites

Name	Size	Type	Description	
Lock	1	uchar	Index	Description
			0	Unknown
			1	Unlocked
			2	Locked
			3	Sleep
			4	LockedGood

5.3. AAb3 BLE

AAb3 BLE implementation supports three services:

- Identity
- Engine
- Authentication

5.3.1. Identity Service Information

Description	UUID
Identity	decc0000-dc4f-4376-9df9-8de1f845b254

5.3.2. Identity Service UUID List

Descriptions	UUID
Device ID	decc0001-dc4f-4376-9df9-8de1f845b254
Product ID	decc0002-dc4f-4376-9df9-8de1f845b254
Interface Version	decc0003-dc4f-4376-9df9-8de1f845b254

5.3.3. Identity Service Characteristic Table

- Types available are Read, Write, and Notify
- Size is measured in bytes

Descriptions	Type	Size	Data / Units
Device ID	Read	10	Esn
Product ID	Read	5	"AAb3"
Interface Version	Read	4	Integer Version Number

5.3.4. Engine Service Information

Description	UUID
Engine Service	1B19B844-038F-11E5-8418-1697F925EC7B

5.3.5. HOS Service UUID List

Descriptions	UUID
GPS Info	e7737830-1018-11e6-a148-3e1d05defe78
Speed	5eed6ea2-0390-11e5-8418-1697f925ec7b
True Odo	5eed6d4e-0390-11e5-8418-1697f925ec7b
Derived Odo	5eed6719-0390-11e5-8418-1697f925ec7b
Eng/Ign	5eed665c-0390-11e5-8418-1697f925ec7b
RPM	31e6e24b-f2fb-4bb9-a16b-9d17a9c4e4ad
Protocol	5eed6477-0390-11e5-8418-1697f925ec7b
VIN	a9b9f487-5e60-43d5-a249-4d1d3f317d7e
Eng Hours	95dbace5-fcee-467f-bbe9-fe42e195bb04
Moving	5eed659a-0390-11e5-8418-1697f925ec7b
ECM	5eed70e6-0390-11e5-8418-1697f925ec7b
Fuel Level	1c3917a6-7d33-4152-9a33-858b6f1fc99b
Eng Run Time	b71eb3aa-4ed7-4990-ab22-ced09cf58f34
Eng Hours Seconds	87e22dd2-e27b-4094-968a-03e1537e7eb7
External Power	0dd2af9d-7220-474f-8b34-4a94a6a97498
Start Trip GPS info	d7b083dd-bf65-44e5-9fa1-959bddf2af0b
Start Trip True Odo	ba4ac777-52ec-4133-9f18-adad8b98f142
Start Trip Eng Hours Seconds	40418a44-9b8d-49f8-8ef2-afcea1643786
ECU Eng Hours Seconds	0d407e4c-39c9-11e8-b467-0ed5f89f718b
Start Trip ECU Eng Hours Seconds	d4fb1556-39d2-11e8-b467-0ed5f89f718b
Start Trip System Unix Time	0f834728-39d5-11e8-b467-0ed5f89f718b
System Unix Time	48969a7e-39d5-11e8-b467-0ed5f89f718b
Driver ID	66aa658a-0003-4cee-86f0-4c775bbe5a54
Start Trip Derived Odom	5ceb24b2-c273-40b4-b84b-f499e14c3f09

5.3.6. HOS Service Characteristics Table

The AAb3 HOS service has encrypted characteristics.

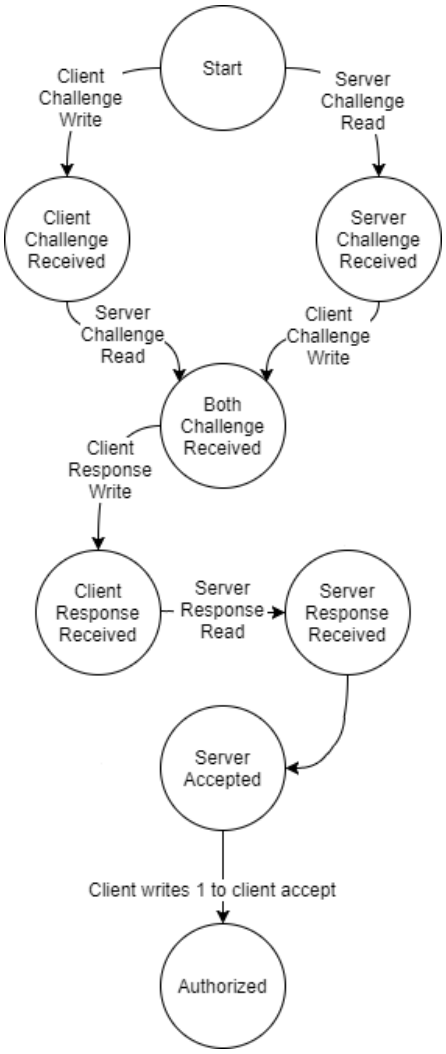
- Types available are Read, Write, and Notify
- Size and Encrypted Size is measured in bytes

Descriptions	Type	Size	Encr. Size	Data / Units	Data Type
GPS Info	Read Notify *Type: Notify on lock or time change*	19	32	See GPS Info Structure below	
Speed	Read	2‡	16	0.1 KPH	uint16 + uint8
True Odo	Read	4‡	16	Meters	uint32 + uint8
Derived Odo	Read	4‡	16	Meters	uint32 + uint8
Eng/Ign	Read Notify *Type: Notify on change*	1	16	Index: 0 = Engine off; 1 = Engine on	uint8
RPM	Read	4‡	16	RPM	uint32 + uint8
Protocol	Read	1	16	Index: 0 = None; 1 = J-Bus; 2 = OBDII	sint8
VIN	Read	18	32	Null Terminated String	Null Terminated String
Eng Hours	Read	4	16	Configurable via ovr[0] to be ECU only, derived only, or derived if ECU unavailable (hours)	uint32
Moving	Read Notify *Type: Notify on change*	1	16	Index: 0 = Not Moving; 1 = Moving (speed > 5 KPH)	sint8 (value will not change if interpeted as uint8)

Descriptions	Type	Size	Encr. Size	Data / Units	Data Type
ECM	Read Notify *Type: Notify on change*	1	16	Index: Description 0: Inactive 1: Pending 2: Connected 3: Active 4: Disabled by config	sint8 (value will not change if interpeted as uint8)
Fuel Level	Read	2	16	Tenths of a percent	uint16
Eng Run Time	Read	4	16	Seconds	uint32
Eng Hours Seconds	Read	4	16	Seconds. Configurable via ovr[0] to be ECU or Derived ONLY, or Derived unless ECU is available	uint32
External Power	Read	1	16	Index: Description 0: No external power 1: External power	sint8 (value will not change if interpeted as uint8)
Start Trip GPS info	Read	19	32	See Below	
Start Trip True Odo	Read	4	16	Meters	uint32
Start Trip Eng Hours Seconds	Read	4	16	Configurable via ovr[0] to be ECU only, derived only, or Derived if ECU is unavailable (s)	uint32
ECU Eng Hours Seconds	Read	4	16	ECU only (seconds)	uint32
Start Trip ECU Eng Hours Seconds	Read	4	16	ECU only (seconds)	uint32
Start Trip System Unix Time	Read	4	16	Seconds since epoch	uint32
System Unix Time	Read	4	16	Seconds since epoch	uint32
Driver ID	Read Write	4(R)/8 (W)	16	ROM ID, i.e., 0x12345678. Driver ID values must be written in reverse byte order.	uint32
Start Trip Derived Odom	Read	4	16	Meters	uint32

NOTE: (‡) Contains a +1 updated byte only when in the encrypted version

5.3.7. Authentication Service Process



5.3.8. Authentication Service UUID

SHA256 is used for the authentication service on the AAb3 target. The input for each response includes the two challenges along with the customer AES key.

Description	UUID
Authentication	1b19bf16-038f-11e5-8418-1697f925ec7b

Descriptions	UUID
Server_challenge	5eedbbdd-0390-11e5-8418-1697f925ec7b
Client_response	5eed91ef-0390-11e5-8418-1697f925ec7b
Client_challenge	5eed8812-0390-11e5-8418-1697f925ec7b:
Server_response	5eed4793-0390-11e5-8418-1697f925ec7b
Client_accepts	5eed58ad-0390-11e5-8418-1697f925ec7b

5.3.9. Authentication Service Characteristics Table

Descriptions	Type	Size	Data / Units
Server_challenge	Read	16	Randomly generated challenge value
Client_response	Write	16	SHA256 Response computed by client sha256(cat(client_challenge, server_ challenge, secret_key))
Client_challenge	Write	16	Randomly generated challenge value
Server_response	Read Notify	16	SHA256 response computed by server sha256(cat(server_challenge, client_ challenge, secret_key))
Client_accepts	Write	1	Index: 0 — Client rejects server response Index: 1 — Client accepts server response

5.3.10. GPS Info Structure

Name	Size	Type	Description
Unix Time	4	sint32	seconds since epoch
Lat	4	sint32	degrees * 1e6
Lon	4	sint32	degrees * 1e6
Speed	1	uchar	0.1 KPH
Heading	2	sint16	degrees * 10
HDOP	2	uint16	hdop * 10
Sats	1	uchar	Number of Satellites
Lock	1	uchar	Index: 0 — Unknown

Name	Size	Type	Description

Index	Description
1	Unlocked
2	Locked
3	Sleep
4	LockedGood

5.4. Legacy AAb2 BLE

Legacy AAb2 BLE implementation supports three services:

- Engine
- Timer
- Pass-Through

5.4.1. Engine Service Information

Description	UUID
Engine	1b19b844-038f-11e5-8418-1697f925ec7b

5.4.2. Engine Service UUID List

Descriptions	UUID
VIN	a9b9f487-5e60-43d5-a249-4d1d3f317d7e
ECM Status	5eed70e6-0390-11e5-8418-1697f925ec7b
Ignition	5eed665c-0390-11e5-8418-1697f925ec7b
RPM	31e6e24b-f2fb-4bb9-a16b-9d17a9c4e4ad
Engine Time	95dbace5-fcee-467f-bbe9-fe42e195bb04
Engine Hours	5eed639c-0390-11e5-8418-1697f925ec7b
Derived Engine Hours	5eed628e-0390-11e5-8418-1697f925ec7b
GPS Status	5eed7230-0390-11e5-8418-1697f925ec7b
Date Time	2a08
Longitude	2aaf
Latitude	2aae
Speed	5eed6ea2-0390-11e5-8418-1697f925ec7b
Odometer (True)	5eed6d4e-0390-11e5-8418-1697f925ec7b
Odometer (Derived)	5eed6719-0390-11e5-8418-1697f925ec7b
Odo Interval	f05252bb-9263-4491-9e06-40ab3e37dcf1
Moving	5eed659a-0390-11e5-8418-1697f925ec7b
Protocol	5eed6477-0390-11e5-8418-1697f925ec7b

5.4.3. Engine Service Characteristic Table

- Types available are Read, Write, and Notify
- Size is measured in bytes

Descriptions	Type	Size	Data / Units
VIN	Read	18	Null terminated VIN string
ECM Status	Read Notify	1	Index: Description 0: Off 1: Pending 2: Connected 3: Active 4: Disabled by config
Ignition	Read Notify	1	Index: Description 0: Ignition off 1: Ignition On
RPM	Read	4	Engine RPM
Engine Time	Read	4	Seconds since last engine on. Once a trip ends, it reports the trip length and then restart to 0 on next ignition on.
Engine Hours	Read	4	Number of hours engine has been on from OBD if available, otherwise the number is derived.
Derived Engine Hours	Read	4	Number of hours engine has been on, derived.
GPS Status	Read Notify	1	Index: Description 0: Unlocked or Unknown 4: Locked and Integrated
Date Time	Read	7	See GATT standard
Longitude	Read	4	Longitude according to GATT standard; defaults to 0 if unavailable
Latitude	Read	4	Latitude according to GATT standard; defaults to 0 if unavailable
Speed	Read	2	Units in 0.1 KPH (or KPH * 10)
Odometer (True)	Read Notify	4	From OBD (m)
Odometer (Derived)	Read Notify	1	Derived (m)
Odo Interval	Read Write	1	Notify frequency (km)

Descriptions	Type	Size	Data / Units
Moving	Read Notify	1	Index: 0 — Not Moving Index: 1 — Moving
Protocol	Read	1	Index: 0 — Unknown Index: 1 — J-Bus Index: 2 — OBDII

5.4.4. Timer Service Information

Description	UUID
Timer	1b19bb5a-038f-11e5-8418-1697f925ec7b

5.4.5. Timer Service UUID List

Descriptions	UUID
Timer	1B19BB5A-038F-11E5-8418-1697F925EC7B
Interval	5eed73b6-0390-11e5-8418-1697f925ec7b

5.4.6. Timer Service Characteristic Table

- Types available are Read, Write, and Notify
- Size is measured in bytes

Descriptions	Type	Size	Data / Units
Timer	Read Notify	4	Current Unix Time (s)
Interval	Read Write	1	Notify interval (s)

5.4.7. Pass-Through Service Information

Description	UUID
Pass-through	1b19bce0-038f-11e5-8418-1697f925ec7b

5.4.8. Pass-Through Service UUID List

Descriptions	UUID
TX	5eed63d4-0390-11e5-8418-1697f925ec7b
RX	5eed640e-0390-11e5-8418-1697f925ec7b
Header	5eed627e-0390-11e5-8418-1697f925ec7b

5.4.9. Pass-through Service Characteristics Table

- Types available are Read, Write, and Notify
- Size is measured in bytes

Descriptions	Type	Size	Data / Units
TX	R/W/N	512	See documentation
RX	Read Notify	512	N/A
Header	Read Write	16	N/A
Cell State	Read Notify	1	N/A

5.5. AAB4 BLE

5.5.1. Overview

AAb4 BLE supports two services:

- EDL
- Info

5.5.2. EDL Service UUID

Description	UUID
EDL	4117a79a-6c31-4ecc-a779-f1c9b177177b

5.5.3. EDL UUID List

Descriptions	UUID
VIN	a9b9f487-5e60-43d5-a249-4d1d3f317d7e
Moving	5eed659a-0390-11e5-8418-1697f925ec7b
Mileage/Odo Interval	f05252bb-9263-4491-9e06-40ab3e37dcf1
Mileage (True Odo)	5eed6d4e-0390-11e5-8418-1697f925ec7b
GPS Info	53b13abb-41af-42de-8156-21e7b48143d7
Gear Position	9b342763-249d-46b5-9301-ba4612a5d9f6

Descriptions	UUID
Engine Status	5eed665c-0390-11e5-8418-1697f925ec7b
Engine Hours Sec	42d80e22-fc83-11e7-8450-fea9aa178066
Engine Hours	95dbace5-fcee-467f-bbe9-fe42e195bb04
ECU Status	5eed70e6-0390-11e5-8418-1697f925ec7b
Driver ID	66aa658a-0003-4cee-86f0-4c775bbe5a54
Date/Time ("Unix")	2A08
Buffer Full	7beb654c-98dc-4d8b-8cce-a427c8f3187f

5.5.4. EDL Service Characteristic table

- Types available are Read, Write, and Notify
- Size is measured in bytes

Descriptions	Type	Size	Data / Units
VIN	Read	18	char[18] (Null Terminated)
Moving	Read Notify	1	Bool: 0 = Stopped; 1 = Moving
Mileage/Odo Interval	Read Write	1	Kilometers
Mileage (True Odo)	Read Notify (on interval)	4(R)/1 (W)	Meters (designed to fallback to gps lifetime odo when not available from engine bus)
GPS Info	Read	23	See GPS Info Structure
Gear Position	Read Notify	1	Index: 0 = Neutral; 1 = Forward; 2 = Reverse; 3 = Park
Engine Status	Read Notify	1	Bool: 0 = Off; 1 = On
Engine Hours Sec	Read	4	Seconds
Engine Hours	Read	4	Hours

Descriptions	Type	Size	Data / Units
ECU Status	Read	1	Index: Description 0: Off 1: Pending 2: Connected 3: Active 4: Disabled by config
Driver ID	Read Write	4(R)/8 (W)	char[8]**
Date/Time ("Unix")	Read	7	(See GATT Standard)
Buffer Full	Read Notify	1	Bool: Description 0: Not full 1: Full

5.5.5. Info Service UUID

Description	UUID
Info	ede3469c-f706-11e7-8c3f-9a214cf093ae

5.5.6. Info Service UUID List

Descriptions	UUID
Main FW Version	75ee1b84-f707-11e7-8c3f-9a214cf093ae
Bootloader FW Version	0cb8b4d8-f709-11e7-8c3f-9a214cf093ae
gcv[0] value	60fd30d2-f709-11e7-8c3f-9a214cf093ae
Device ID	d1371d4a-f709-11e7-8c3f-9a214cf093ae

5.5.7. Info Service Characteristic Table

- Types available are Read, Write, and Notify
- Size is measured in bytes

Descriptions	Type	Size	Data / Units
Main FW Version	Read	16	Version string
Bootloader FW Version	Read	16	Version string

Descriptions	Type	Size	Data / Units
gcv[0] value	Read	4	Unsigned long
Device ID	Read	4	ESN

5.5.8. GPS Info Structure

Name	Type	Size	Description
Unix Time	sint32	4	seconds since epoch
Lat	sint32	4	degrees * 1e6
Lon	sint32	4	degrees *1e6
speed	uchar	1	0.1 KPH
Heading	sint16	2	degress*10
HDOP	uint16	2	HDOP*10
Sats	uchar	1	Number of Satellites
Lock	uchar	1	Value / Description: 0 Unknown; 1 Unlocked; 2 Locked; 3 Sleep; 4 LockedGood
Altitude	sint32	4	meters*10

5.6. AAB5 BLE

5.6.1. Overview

AAb5 BLE supports two services:

- EDL
- Info

5.6.2. EDL Service UUID

Description	UUID
EDL	4117a79a-6c31-4ecc-a779-f1c9b177177b

5.6.3. EDL UUID List

Descriptions	UUID
VIN	a9b9f487-5e60-43d5-a249-4d1d3f317d7e
Moving	5eed659a-0390-11e5-8418-1697f925ec7b
Mileage/Odo Interval	f05252bb-9263-4491-9e06-40ab3e37dcf1
Mileage (True Odo)	5eed6d4e-0390-11e5-8418-1697f925ec7b
GPS Info	53b13abb-41af-42de-8156-21e7b48143d7
Gear Position	9b342763-249d-46b5-9301-ba4612a5d9f6
Engine Status	5eed665c-0390-11e5-8418-1697f925ec7b
Engine Hours Sec	42d80e22-fc83-11e7-8450-fea9aa178066
Engine Hours	95dbace5-fcee-467f-bbe9-fe42e195bb04
ECU Status	5eed70e6-0390-11e5-8418-1697f925ec7b
Driver ID	66aa658a-0003-4cee-86f0-4c775bbe5a54
Date/Time ("Unix")	2A08
Buffer Full	7beb654c-98dc-4d8b-8cce-a427c8f3187f

5.6.4. EDL Service Characteristic table

- Types available are Read, Write, and Notify
- Size is measured in bytes

Descriptions	Type	Size	Data / Units
VIN	Read	18	char[18] (Null Terminated)
Moving	Read Notify	1	Bool: 0 = Stopped; 1 = Moving
Mileage/Odo Interval	Read Write	4(R)/1 (W)	Kilometers
Mileage (True Odo)	Read Notify (on interval)	4	Meters (designed to fallback to gps lifetime odo when not available from engine bus)
GPS Info	Read	23	See GPS Info Structure
Gear Position	Read Notify	1	Index: 0 = Neutral; 1 = Forward; 2 = Reverse; 3 = Park
Engine Status	Read Notify	1	Bool: 0 = Off

Descriptions	Type	Size	Data / Units
			Bool: 1 — Description: On
Engine Hours Sec	Read	4	Seconds
Engine Hours	Read	4	Hours
ECU Status	Read	1	Index: 0 — Description: Off Index: 1 — Description: Pending Index: 2 — Description: Connected Index: 3 — Description: Active Index: 4 — Description: Disabled by config
Driver ID	Read Write	4(R)/8 (W)	char[8]**
Date/Time ("Unix")	Read	7	(See GATT Standard)
Buffer Full	Read Notify	1	Bool: 0 — Description: Not full Bool: 1 — Description: Full

5.6.5. Info Service UUID

Description	UUID
Info	ede3469c-f706-11e7-8c3f-9a214cf093ae

5.6.6. Info Service UUID List

Descriptions	UUID
Main FW Version	75ee1b84-f707-11e7-8c3f-9a214cf093ae
Bootloader FW Version	0cb8b4d8-f709-11e7-8c3f-9a214cf093ae
gcv[0] value	60fd30d2-f709-11e7-8c3f-9a214cf093ae
Device ID	d1371d4a-f709-11e7-8c3f-9a214cf093ae

5.6.7. Info Service Characteristic Table

- Types available are Read, Write, and Notify
- Size is measured in bytes

Descriptions	Type	Size	Data / Units
Main FW Version	Read	16	Version string
Bootloader FW Version	Read	16	Version string
gcv[0] value	Read	4	Unsigned long
Device ID	Read	4	ESN

5.6.8. GPS Info Structure

Name	Type	Size	Description
Unix Time	SINT32	4	seconds since epoch
Lat	SINT32	4	degrees * 1e6
Lon	SINT32	4	degrees *1e6
Speed	UCHAR	1	0.1 KPH
Heading	SINT16	2	Degrees*10
HDOP	UINT16	2	HDOP*10
Sats	UCHAR	1	Number of Satellites
LOCK	UCHAR	1	Value: Description 0: Unknown 1: Unlocked 2: Locked 3: Sleep 4: LockedGood
Altitude	SINT32	4	Meters*10

6. 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
```

6.1. Acceleration Event Threshold (AET)

The *Acceleration Event Threshold* parameter determines which conditions must be met for an event to register.

Configuration instances: 8

Command Type	Syntax
SET	**Command** :wycfg aet[x] <direction> <start_duration> <end_duration> <mg_thresh> **Response** :OK
READ	**Command** :rycfg aet[x] **Response** | aet[x]: dir:<direction> (hex_val), start:<start_duration> (hex_val), end:<end_duration> (hex_val), mg:<mg_thresh> (hex_val) :OK

Parameter	Range	Description
direction	N/A	Value: 0 = Acceleration; 1 = Deceleration; 2 = Left Turn; 3 = Right Turn
start_duration	0 to 65535	Time the acceleration in the target direction must stay above the mg_thresh before an event is considered to have started (ms).
end_duration	0 to 65535	Time that an acceleration in the target direction must stay below the mg_thresh before an event is considered to have concluded. If another acceleration threshold is exceeded before this time, it is negated (ms).
mg_thresh	0 to 2000	The threshold that determines when start_duration and end_duration trigger (mg).

Target	Default Value(s)
AAb1, AAa4, AAb5, AAb6	aet[0] 0 1500 1000 100 aet[1] 1 1500 1000 100 aet[2] 2 1500 1000 100 aet[3] 3 1500 1000 100 aet[4] 0 60000 1000 100 aet[5] 1 60000 1000 100 aet[6] 2 60000 1000 100

Target	Default Value(s)
	aet[7] 3 60000 1000 100
AAb3	aet[0] 0 1000 1000 240 aet[1] 1 1000 1000 280 aet[2] 2 2000 1000 900 aet[3] 3 2000 1000 900 aet[4] 0 60000 1000 100 aet[5] 1 60000 1000 100 aet[6] 2 60000 1000 100 aet[7] 3 60000 1000 100

6.2. Acceleration Filter Coefficients (AFC)

Configuration instances: 1

Command Type	Syntax
SET	**Command** :wycfg afc <filter_len> <coef_1> <coef_2> ..... <coef_31> **Response** :OK
READ	**Command** :rycfg afc **Response** | afc: stages:<filter_len> (hex_val), coef1:<coef_val_1> (hex_val), coef2:<coef_val_2> ..... (hex_val), coef32:<coef_val_31> (hex_val) :OK

Parameter	Range	Description
filter_len	0 to 31	The number of stages to be used in the FIR filter
coef_val_XX	0 to 4294967295	The coefficient for the filter stage denoted by 'XX'. It can be input as a decimal number (1.234567), as the internal system representation of the decimal value multiplied by one million (1234567), or as the raw hex of the system representation of the decimal value (0x12D687). NOTE: Ideally, all filter coefficients should sum to one unless attenuation or gain is desired.

NOTE: This parameter does not have a factory default setting

6.3. Acceleration Percent Correction (APC)

The *Acceleration Percent Correction* parameter is a scalar value multiplied by the filter output to produce gain or attenuation. APC is input as the explicit percent change interested in.

Configuration instances: 1

> **NOTE:** apc (and arc) corrections are only applied to the max mg value at the end of the event. They do NOT affect the determination of accelerometer events. The equation is (roughly) as follows:

$$reported\_max\_mg = \frac{(actual)(apc)}{100} + arc$$

E.g., if an APC of 50% is desired, set *apc* to the value of 50. There is a separate percent correction factor for the x axis and the y axis. There is no percent correction factor for the z axis.

Command Type	Syntax
SET	**Command** :wycfg apc <percent_correction_factor_x_axis> <percent_correction_factor_y_axis> **Response** :OK
READ	**Command** :rycfg apc **Response** | apc: corr_x:<percent_correction_factor_x_axis> (hex_val), corr_y:<percent_correction_factor_y_axis> (hex_val) :OK

Parameter	Range	Description
percent_correction_factor_x_axis	1 to 200	Percentage value
percent_correction_factor_y_axis	1 to 200	Percentage value

Target	Default Value(s)
AAb1, AAb3, AAa4, AAb5, AAb6	apc[0] 100 100

6.4. APNs (APN)

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

The device will hunt through the APNs in sequence (0, 1, 2, 3), attempting to activate context once on each, until it is successful.

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. **NOTE:** APN characters are case sensitive.
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.

Target	Default Value(s)
AAb1, AAb3, AAb4, AAb5, AAb6	apn[0] "" "user" "passwd" apn[1] "" "user" "passwd"

Target	Default Value(s)
	apn[2] "" "user" "passwd" apn[3] "" "user" "passwd"

NOTE: Network carriers are inconsistent with handling special characters (~`!@$%^&*()-_+={};/?,.) over SMS. It is recommended to only use UDP commands to set passwords that contain special characters.

6.5. Acceleration Report Correction (arc)

The *acceleration report correction* factor is a scalar value that is added to the output of the filter to add a DC offset to the signal. It can be input as a decimal number (1.234567), the internal system representation of the decimal value multiplied by one million (1234567), or as the raw hex of the system representation of the decimal value (0x12D687).

There is a separate correction factor for the x axis and the y axis; there is no correction factor for the z axis.

Configuration instances: 1

> **NOTE:** arc (and apc) corrections are only applied to the max mg value at the end of the event. They do NOT affect the determination of accelerometer events. The equation is (roughly) as follows:

$$reported\_max\_mg = \frac{(actual\_max\_mg)(apc)}{100} + arc$$

Command Type	Syntax
SET	**Command** :wycfg arc <correction_factor_x_axis> < correction_factor_y_axis> **Response** :OK
READ	**Command** :rycfg arc **Response** | arc: corr_x:<correction_factor_x_axis> (hex_val), corr_y:<correction_factor_y_axis> (hex_val) :OK

Parameter	Range	Description
correction_ factor_x_axis	-2000 to 2000	Scalar value
correction_ factor_y_axis	-2000 to 2000	Scalar value

Target	Default Value(s)
AAb1, AAb3, AAa4, AAb5, AAb6	arc[0] 0 0

6.6. Alternator State Threshold (AST)

The *Alternator State Threshold* configuration determines the voltage and time thresholds that cause logic events. There are four alternator state thresholds, each with their own parameter ranges:

- ast[0] -> Alternator On Threshold
- ast[1] -> Alternator Off Threshold
- ast[2] -> 24V Check Threshold
- ast[3] -> Dynamic Alternator Delta

Configuration instances: 4

Command Type	Syntax
SET	**Command** :wycfg ast[x] <threshold> <time_val> **Response** :OK
READ	**Command** :rycfg ast[x] **Response** | ast[x]: thresh:<threshold> (hex_val), time:<time_val> (hex_val) :OK

Instance	Range	Description
ast[0]	120 to 400	*Threshold (0.1 V per bit)*
	1 to 60	*Time_val (seconds)* Number of seconds voltage is above threshold to begin OBD Ignition On logic
ast[1]	100 to 400	*Threshold (0.1 V per bit)* If Alternator Off Threshold is greater than or equal to Alternator On Threshold, the internal Alternator Off Threshold will be limited to Alternator On Threshold - 0.1V.
	1 to 60	*Time_val (seconds)* Number of seconds voltage is below threshold to detect OBD Ignition Off logic
ast[2]	0 (disable), 140 to 200	*Threshold (0.1 V per bit)*

Instance	Range	Description
	1 to 10	*Time_val (seconds)* Number of seconds voltage is above threshold to detect 24V system
ast[3]	0 (disable), 3 to 10	*Threshold (0.1 V per bit) -- See Dynamic Alternator on p. 191*
	1 to 10	*Time_val (seconds)* Not used

Target	Default Value(s)
AAb1, AAb3, AAa4, AAb5, AAb6	ast[0] 132 3 ast[1] 131 3 ast[2] 160 1 ast[3] 0 1

6.7. BLE Authentication Disable (bad)

Disables or enables BLE authentication and encryption of characteristics for the AAb3 target only

Configuration instances: 1

Command Type	Syntax
SET	**Command** :wycfg bad <disable> **Response** :OK
READ	**Command** :rycfg bad **Response** | bad: disable:<disable> (hex_val) :OK

Parameter	Description
disable	Disables Bluetooth:

Value	Description
0	Enables BLE authentication and encryption of characteristics
1	Disables BLE authentication and encryption of characteristics

Target	Default Value(s)
AAb1, AAb3, AAa4, AAb5, AAb6	bad 1

6.8. OBD Backoff (bko)

Configuration instances: 6

Command Type	Syntax
SET	**Command** :wycfg bko[x] <enable_value> **Response** :OK
READ	**Command** :rycfg bko[x] **Response** | bko[x]: enable:<enable_value> (hex_val) :OK

Parameter	Description
enable	OBD stops vehicle bus communication when feature is enabled.

Index	Description
0	Disable (default)
1	Enable

Instance	Description
bko[0]	Scantool Detect Enable (retries communication in 60 seconds)
bko[1]	New MIL Status Detect Enable (end communication until next ignition cycle on a new MIL status detection)
bko[2]	Persistent MIL Status Detect Enable (read MIL status for each ignition on, if still set, end communication)
bko[3]	Protocol Detect Failure (OBD protocol failed start protocol three consecutive ignition cycles - disable OBD communication until this feature is disabled)
bko[4]	Scantool Detect Enable (backs off until next ignition cycle)
bko[5]	Scantool Detect Mask (for use with Scantool Detect) OBD stops vehicle bus communication on specific messages when mask is set. *Mask range:*

Hex	Description
0x01	OnStar request
0x02	OBDII request
0x04	Diagnostic requests
0x08	Security access requests
0x10	User defined (currently not implemented)
0x20	Ford EOL request (Target AAb3 only)

Target	Default Value(s)
AAb1, AAb3, AAa4, AAb5, AAb6	bko[0] 0 bko[1] 0 bko[2] 0 bko[3] 0 bko[4] 0 bko[5] 0x03

6.9. 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:

Value	Description
0	Disables bluetooth communications
1	Enables bluetooth communications (not available on AAa4 target)

Target	Default Value(s)
AAb1, AAb3, AAa4, AAb5, AAb6	blt[0] 0

6.10. BLUETOOTH TRANSMIT POWER (BTP)

Configuration instances: 1

Command Type	Syntax
SET	**Command** :wycfg btp <power_val> **Response** :OK
READ	**Command** :rycfg btp **Response** | btp: power:<power_val> (hex_val) :OK

Parameter	Range	Description
power_val	0 to 4	The Bluetooth Transmit Power setting; requires a reset to take effect.

Target	Default Value(s)
AAb1, AAb3, AAa4, AAb5, AAb6	btp 4

6.11. 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 bzr[x] <frequency1> <duration1> ... <frequency8> <duration8> **Response** :OK
READ	**Command** :rycfg bzr[x] **Response** | bzr[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)

Target	Default Value(s)
AAb1, AAb3, AAb4, AAb5, AAb6	bzr[0] 3000 500 0 200 0 0 0 0 0 0 0 0 0 0 0 0 bzr[1] 3000 500 0 50 0 0 0 0 0 0 0 0 0 0 0 0 bzr[2] 3000 200 0 800 0 0 0 0 0 0 0 0 0 0 0 0 bzr[3] 3000 200 0 300 0 0 0 0 0 0 0 0 0 0 0 0 bzr[4] 3000 3000 0 2 0 0 0 0 0 0 0 0 0 0 0 0 bzr[5] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 bzr[6] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 bzr[7] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 bzr[8] 262 125 294 125 349 125 294 125 440 125 0 125 440 300 392 300 bzr[9] 262 250 294 125 349 125 294 125 392 125 0 125 392 300 349 300

6.12. CAN Controller Settings (CAN)

Configuration instances: 6

Command Type	Syntax
SET	**Command** :wycfg can[x] <param_val> **Response** :OK
READ	**Command** :rycfg can[x] **Response** can[x]: value:<param_val> :OK

Parameter	Description
param_val	Voltage threshold in 0.1 V per bit

Instance	Range	Description
can[0]	N/A	CAN transmit Retry

Index	Description
0	Disable retry
1	Enable retry

Instance	Range	Description
can[1]	N/A	CAN Baud rate

Index	Description
0	Autodetect (default)
1	500k
2	250k
3	125k
4	33k

Instance	Range	Description
can[2]	2 to 16	CAN TSEG1 Time Segment Before Sample Point
can[3]	1 to 8	CAN TSEG2 Time Segment After Sample Point
can[4]	1 to 4	CAN SJW Synchronization Jump Width
can[5]	1 to 1024	BRP Baud Rate Prescaler

Target	Default Value(s)
AAb1, AAb3, AAa4, AAb5, AAb6	can[0] 1

Target	Default Value(s)
	can[1] 0 can[2] 8 can[3] 6 can[4] 4 can[5] 8

6.13. Crash Event Parameters (cep)

Configuration instances: 1

Command Type	Syntax
SET	**Command** :wycfg cep <delta_mg_thresh> <poll_freq> <hist_depth> **Response** :OK
READ	**Command** :rycfg cep **Response** | cep: mg:<delta_mg_thresh> (hex_val), freq:<poll_freq> (hex_val), depth:<hist_depth> (hex_val) :OK

Parameter	Range	Description
delta_mg_thresh	1 to 2000	If the change in force in any direction exceeds this mg value, the event will trigger.
poll_freq	1 to 5	How often the accelerometer is polled for history data (Hz).
hist_depth	1 to 75	Number of accelerometer plots saved; each plot is a reoriented simple accelerometer vector (x, y, z).

Target	Default Value(s)
AAb1, AAb3, AAa4, AAb5, AAb6	cep 1000 5 75

6.14. Accelerometer Confidence Slew (csl)

Configuration only available for target AAb3

Command Type	Syntax
SET	**Command** :wycfg csl <rate_val> **Response** :OK
READ	**Command** :rycfg csl **Response** | csl: rate:<rate_val> (hex_val) :OK

Parameter	Range	Description
rate_val	1 to 4294967295	Scalar for accelerometer orientation falling slew.

Target	Default Value(s)
AAb3	csl 10000

6.15. Accelerometer Confidence Threshold (cth)

Configuration instances: 1

Command Type	Syntax
SET	**Command** :wycfg cth <confidence_threshold> <hysteresis> **Response** :OK
READ	**Command** :rycfg cth **Response** | cth: conf:<confidence_threshold> (hex_val), hyst:<hysteresis> (hex_val) :OK

Parameter	Range	Description
confidence_ threshold	1 to 255	A value, above which accelerometer events will be reported to the interpreter.
hysteresis	0 to 255	A hysteresis value applied to a falling confidence level.

Target	Default Value(s)
AAb1, AAb3, AAa4, AAb5, AAb6	cth 120 60

6.16. DNS Cache Time to Live (DNS)

Configuration instances: 1

Command Type	Syntax
SET	**Command** :wycfg dns <dns_time_to_live> **Response** :OK
READ	**Command** :rycfg dns **Response** | dns: seconds:<dns_time_to_live>(hex_val) :OK

Parameter	Range	Description
dns_time_to_live	0 to 604800	The time in seconds that a domain name to IP address mapping will be stored in cache.

Target	Default Value(s)
AAb1, AAa4, AAb5, AAb6	dns 86400

6.17. 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.

Target	Default Value(s)
AAb1, AAa4, AAb5, AAb6	dst[0] "0.0.0.0" 1024 dst[1] "0.0.0.0" 1024 dst[2] "0.0.0.0" 1024 dst[3] "0.0.0.0" 1024 dst[4] "0.0.0.0" 1024 dst[5] "0.0.0.0" 1024 dst[6] "0.0.0.0" 1024 dst[7] "0.0.0.0" 1024 dst[8] "0.0.0.0" 1024 dst[9] "216.70.51.106" 3110
AAb3	dst[0] "0.0.0.0" 1024 dst[1] "0.0.0.0" 1024 dst[2] "0.0.0.0" 1024 dst[3] "0.0.0.0" 1024 dst[4] "0.0.0.0" 1024 dst[5] "0.0.0.0" 1024 dst[6] "0.0.0.0" 1024 dst[7] "0.0.0.0" 1024 dst[8] "0.0.0.0" 1024 dst[9] null

6.18. OBD DTC Threshold (dtc)

Configuration instances: 1

Command Type	Syntax
SET	**Command** :wycfg dtc <poll period> <speed threshold 1mph/bit> **Response** :OK
READ	**Command** :rycfg dtc **Response** | dtc: polfreq:<poll period> (hex_val), thresh:<speed threshold 1mph/bit> (hex_val) :OK

Parameter	Range	Description
poll period	0 to 30	When vehicle speed is at or below <speed threshold>, periodically poll for DTC change (in minutes) NOTE: Setting this to 5, for example, means the device will poll once every 5 min
speed threshold	0 to 100	Begin DTC reading when vehicle is at or below <speed threshold> in mph

Target	Default Value(s)
AAb1, AAb3, AAa4, AAb5, AAb6	dtc[0] 0 10

6.19. Flag Save Mask (fsm)

Configuration instances: 1

Command Type	Syntax
SET	**Command** :wycfg fsm <32bit_mask> **Response** :OK
READ	**Command** :rycfg fsm **Response** fsm: mask:<32bit_mask> :OK

Parameter	Range	Description
32bit_mask:	0x0 to 0xFFFFFFFF	A bitmask value corresponding to which scriptable user flags will be saved in persist data.

Target	Default Value(s)
AAb1, AAb3, AAa4, AAb5, AAb6	fsm 0

6.20. FTP CONFIGURATION (FTP)

FTP is only supported on SARA-R410M, LARA-4202 and ELS61-US cellular modules at this time.

Configuration instances: 1

Command Type	Syntax
SET	**Command** :wycfg ftp "<ip>|<hostname>" <port_number> "<directory>" "<username>" "<password>" <mode_val> <retry_max> **Response** :OK
READ	**Command** :rycfg ftp **Response** | ftp: addr:"<ip>|<hostname>", port:<port_number> (hex_val), dir:"<directory>" , user:"<username>", passwd:"<password>", mode:<mode_val> (hex_val), rtrymax:<retry_max_val> (hex_val) :OK

Parameter	Range	Description
<ip>|<hostname>	63 char	Takes the form of an IP address (192.0.0.0), or a hostname .
port_number	1 to 65533	Specifies the port on which to connect to when using an IP address or hostname.
directory	31 char	A sub-directory on the FTP server where the requested files can be found.
username	24 char	Username used to access the FTP server given by the first argument.
password	12 char	Password used to access the FTP server given by the first argument. NOTE: Network carriers are inconsistent with handling special characters (~`!@$%^&*()-_+={};/?,.) over SMS. It is recommended to only use UDP commands to set passwords that contain special characters.
mode_val	N/A	Index: 0 — Description: (Discontinued) Index: 1 — Description: Passive
retry_max_val	0 to 255	Number of times the device tries to fetch the requested file before giving up and flushing the request queue.

Target	Default Value(s)
AAb1, AAb3, AAa4, AAb5, AAb6	ftp "0.0.0.0" 21 "dir" "usn" "pwd" 0 3

6.21. GARMIN™ BLACKLIST (GBL)

Configuration instances: 16

Command Type	Syntax
SET	**Command** :wycfg gbl[x] <flags_hex> <pid_hex> <size_val> <byte0_hex> <byte1_hex> **Response** :OK
READ	**Command** :rycfg gbl **Response** | gbl[x]: flags:<flags_hex>, pid:<pid_hex>, size:<size_val> (hex_val), byte0:<byte0_hex>, byte1:<byte1_hex> :OK

Parameter	Range	Description
flags_hex	N/A	Mask that configures the match method: Hex: Description 0x1: Match byte 1 0x2: Match byte 0 0x4: Match size 0x8: Match PID
pid	0x0 to 0xFF	Single byte PID id match
size	0 to 255	Single byte Garmin packet size match
byte0_hex	0x0 to 0xFF	Single byte Garmin packet first byte match
byte1_hex	0x0 to 0xFF	Single byte Garmin packet second byte match

Target	Default Value(s)
AAb1, AAb3, AAa4, AAb5, AAb6	gbl[x] 0

6.22. 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.

Target	Default Value(s)
AAb1, AAb3, AAa4, AAb5, AAb6	gcv[x] 0

6.23. Geofence (gfn)

This config allows for both circular and polygonal geofences.

Configuration instances: 50

Circular Fence

Command Type	Syntax
SET	**Command** :wycfg gfn[x] <latitude> <longitude> <radius_value> **Response** :OK
READ	**Command** :rycfg gfn[x] **Response** | gfn[x]: lat:<latitude> (hex_val), lon:<longitude> (hex_val), radius:<radius_value> (hex_val) :OK

Polygonal Fence

At least 3, but no more than 11 coordinate pairs needed

Command Type	Syntax
SET	**Command** :wycfg gfn[x] <latitude0> <longitude0>...<latitude10> <longitude10> **Response** :OK
READ	**Command** :rycfg gfn[x] **Response** | gfn[x]: lat0:<latitude0> (hex_val), lon0<longitude0> (hex_val), ... lat10:<latitude10> (hex_val), lon10:<longitude10> (hex_val) :OK

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

Parameter	Range	Description
latitude	-90.000000 to 90.000000	Latitude point to 6 decimal places (e.g. 44.863511)
longitude	-180.000000 to 180.000000	Longitude point to 6 decimal places (e.g. -93.340138)
radius_value	1 to 1000000	Only for circular fences, distance in meters from center point of circle to outer perimeter

Target	Default Value(s)
AAb1, AAb3, AAa4, AAb5, AAb6	all gfn[x] args default to '0'

6.24. GPS Motion Thresholds (GPS)

Configuration instances: 1

Command Type	Syntax
SET	**Command** :wycfg gps <motion_start> <motion_stop> **Response** :OK
READ	**Command** :rycfg gps **Response** | gps: start:<motion_start> (hex_val), stop:<motion_stop> (hex_val) :OK

Parameter	Range	Description
motion_start	0 to 200	Configure GPS motion detection start speed threshold in tenths of MPH.
motion_stop	0 to 200	Configure GPS motion detection stop speed threshold in tenths of MPH

Target	Default Value(s)
AAb1, AAb3, AAa4, AAb5, AAb6	gps 100 10

6.25. GARMIN™ CONFIGURATION (GRM)

Configuration instances: 1

Command Type	Syntax
SET	**Command** :wycfg grm<filter_enable> **Response** :OK
READ	**Command** :rycfg grm **Response** | grm: enable:<filter_enable> (hex_val) :OK

Parameter	Range	Description
filter_enable	N/A	Activates Garmin message filter that drops the first delete all stops request and the first stop 1 status request in a session.

Target	Default Value(s)
AAb1, AAb3, AAa4, AAb5, AAb6	grm 0

6.26. IMPEDE DEVICE MANAGER (IDM)

Configuration instances: 1

Command Type	Syntax
SET	**Command** :wycfg idm <select> **Response** :OK
READ	**Command** :rycfg idm **Response** | idm: sel:<select> (hex_val) :OK

Parameter	Range	Description	
select	N/A	**Index**	**Description**
		0	DM service operates normally
		1	DM checkin messages and interactions are disabled

Target	Default Value(s)
AAb1, AAb3, AAa4, AAb5, AAb6	idm 0

6.27. 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.

Target	Default Value(s)
AAb1, AAb3, AAa4, AAb5, AAb6	idp[0] 0 idp[1] 0 idp[2] 0 idp[3] 0

6.28. IGNITION SENSE (IGN)

Configuration instances: 1

Command Type	Syntax
SET	**Command** :wycfg ign <type> <onsec> <offsec> OR :wycfg ign <type> <andmask> <ormask> **Response** :OK
READ	**Command** :rycfg ign **Response** | ign: type:<type> (hex_val), param1:<param1> (hex_val), param2:<param2> (hex_val) :OK

Parameter	Range	Description
type	1 to 3	Ignition detection type. Type 1 and 2 use ignition onsec and offsec parameters. Type 3 is used for and/or masks. Index: 1 — Wired 2 — Vehicle Bus 3 — (Composite)
onsec	1 to 65535	Configured ignition type source must be true for x seconds before reported to script.
offsec	1 to 65535	Configured ignition type source must be false for x seconds before reported to script.
andmask/ormask	N/A	Ignition detection masks Index: 0x0001 — Wired Ignition (not allowed on AAa4 XT2400 target) 0x0002 — IN1 0x0004 — IN2 0x0008 — IN3 0x0010 — IN4 0x0020 — Reserved

Parameter	Range	Description

Index	Description
0x0040	Reserved
0x0080	Reserved

NOTE: Ignition sources above will attempt to have a 1:1 with Input/Output State Bits on p. 160

Index	Description
0x0100	Vehicle Bus
0x0200	Virtual Battery (Configured by Alternator State Threshold (ast) on p. 40)
0x0400	Virtual Vibration (Non-configurable)
0x0800	GPS-based movement (Configured by GPS Motion Thresholds (gps) on p. 59)
0x1000	Reserved
0x2000	Reserved
0x4000	Reserved
0x8000	ANDMODE bit • Default (when cleared): ANDMODE=ALL_OFF • when bit is set the ANDMODE=ANY_OFF • ANDMODE only affects the judgement of ignition-off. All AND-sources must be on to judge ignition-on.

Target	Default Value(s)
AAb1, AAb3, AAb5, AAb6	ign 1 1 1
AAa4	Ign 2 1 1

6.29. IoT Destination Address/Port (iip)

Configuration instances: 1

Command Type	Syntax
SET	**Command** :wycfg iip "<ip>|<hostname>" <port_number> <QoS Level> <Keep Alive> **Response** :OK
READ	**Command** :rycfg iip **Response** | iip: addr:"<ip>|<hostname>", port:<port_number> (hex_val), qoslev:<QoS Level> (hex_val), keepa:<Keep Alive> (hex_val) :OK

Parameter	Range	Description
<ip>|<hostname>:	128 Characters	Can take the form of an IP address (192.0.0.0), or a hostname, e.g., www.xirgotech.com
port_number	1 to 65533	When using an IP address or hostname, port can be used to specify the port on which to connect.
QoS Level	N/A	Quality of service level Index: 0 — Fire and forget; device will send messages once Index: 1 — ACK required; device will send messages at least once
Keep Alive	30 to 1200	MQTT keepalive interval. This tells the server how often to expect MQTT PING packets before disconnecting the device. An interval of 60 seconds will tell the server to expect a PING or some other traffic from the device every 60 seconds. If the device does not reply for 90 seconds (keepalive x 1.5) the server will assume the device has gone offline and will close the TLS socket.

Target	Default Value(s)
AAb1, AAb3, AAa4, AAb5, AAb6	iip "0.0.0.0" 1024 1 60

6.30. IoT Shadow Params (ish)

Configuration instances: 1

Command Type	Syntax
SET	**Command** :wycfg ish "<topic_string>" <shadow trim val> **Response** :OK
READ	**Command** :rycfg ish **Response** | ish: topic:“<topic_string>”, shadtrm:<shadow trimval> (hex_val) :OK

Parameter	Range	Description
topic_string	128 Characters	AWS Shadow topic string
Shadow trim val	N/A	Shadow trim enable

Index	Description
0	Disable AWS shadow
1	Enable AWS shadow

Target	Default Value(s)
AAb1, AAb3, AAa4, AAb5, AAb6	Ish "" 0

6.31. Input State Source (iss)

- iss[0] -> Source for user input 1
- iss[1] -> Source for user input 2
- iss[2] -> Source for user input 3
- iss[3] -> Source for user input 4

Configuration instances: 4

Command Type	Syntax
SET	**Command** :wycfg iss[x] <source_val> **Response** :OK
READ	**Command** :rycfg iss[x] **Response** | idp[x]: source:<source_val> (hex_val) :OK

Parameter	Description	
source_val	**Index**	**Description**
	0	Physical hardware pins determine reported state
	1	Script determines reported state

Target	Default Value(s)
AAb1, AAb3, AAa4, AAb5, AAb6	iss[0] 0 iss[1] 0 iss[2] 0 iss[3] 0

6.32. IoT Thing String (its)

Configuration instances: 1

Command Type	Syntax
SET	**Command** :wycfg its "<topic_string>" "<client id string>" **Response** :OK
READ	**Command** :rycfg its **Response** | its: topic:"<topic_string>", clntid:"<client id string>" :OK

Parameter	Range	Description
topic_string	128 Characters	AWS topic string
client id string	64 Characters	AWS client id string. If left blank, defaults to “XT6379_<IMEI>”

Target	Default Value(s)
AAb1, AAb3, AAa4, AAb5, AAn6	its "" ""

6.33. MIP/SIP CONTROL (MIP)

Configuration instances: 1

Command Type	Syntax
SET	**Command** :wycfg mip <mode_value> **Response** :OK
READ	**Command** :rycfg mip **Response** | mip: mode:<mode_value> (hex_val) :OK

Parameter	Description
mode value	Select cellular mobile IP mode: Index: 0 — Simple IP only Index: 1 — Mobile IP preferred Index: 2 — Mobile IP only

Target	Default Value(s)
AAb1, AAb3, AAa4, AAb5, AAb6	mip 1

6.34. MPG CONTROL (MPG)

This is a user configuration for MPG calculations.

Configuration instances: 1

Command Type	Syntax
SET	**Command** :wycfg mpg <fuel type value> <use type value> <EFR coefficient> <MAF coefficient> **Response** :OK
READ	**Command** :rycfg mpg **Response** | mpg: fueltyp:<fuel type value> (hex_val), usetype:<use type value> (hex_val), efrcoef:<EFR coefficient> (hex_val), mafcoef:<MAF coefficient> (hex_val) :OK

Parameter	Range	Description
fuel type value	N/A	Index: 0 — Discover Fuel Type from vehicle (default) 1 — Gasoline 2 — Diesel
use type value	N/A	Index: 0 — Discover support type from vehicle (default) 1 — Use Engine Fuel Rate 2 — Use Mass Air Flow
EFR coefficient	0 to 10	Coefficient used in EFR calculation
MAF coefficient	0 to 10	Coefficient used in MAF calculation

Target	Default Value(s)
AAb1, AAb3, AAa4, AAb5, AAb6	mpg 0 0 1 1

6.35. 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.

Target	Default Value(s)
AAb1, AAb3, AAa4, AAb5, AAb6	mst 100

6.36. NMEA Stream Output (nso)

- nso[0] -> USB
- nso[1] -> Aux0
- nso[2] -> Aux1

Configuration instances: 3

Command Type	Syntax
SET	**Command** :wycfg nso[x] <8bit_mask> **Response** :OK
READ	**Command** :rycfg nso[x] **Response** | nso[x]: mask:<8bit_mask> :OK

Parameter	Description
8bit_mask	One bit for each of the available NMEA sentences; values can be added to stream more than one sentence type. Set mask to zero to stop streaming.

Hex	Description
0x01	GPGGA
0x02	GPGLL
0x04	GPGSA
0x08	GPGSV
0x010	GPRMC

Target	Default Value(s)
AAb1, AAb3, AAa4, AAb5, AAb6	nso[0] 0 nso[1] 0 nso[2] 0

6.37. 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

Target	Default Value(s)
AAb1, AAb3, AAa4, AAb5, AAb6	oad[0] 100 oad[1] 100

6.38. OBD Feature Enable/Disable (obd)

Configuration instances: 8

Command Type	Syntax
SET	**Command** :wycfg obd[x] <flags_index> <mode_index> **Response** :OK
READ	**Command** :rycfg obd[x] **Response** | obd[x]: flags:<flags_index> (hex_val), mode:<mode_index> (hex_val) :OK

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

Instance	Description
obd[0]	Obd Comms Enable. *flags_index* **Index** / **Description** 0 – Disable 1 – Enable *mode_index* **Index** / **Description** 0 – Immediate 1 – Next Ignition off detected
obd[1]	Obd VIN Read Enable. The OBD[1] VIN Read Enable feature can be configured using masks to perform a Ford VIN Read and a VIN Read. Using this configuration will look for Ford VIN first. If VIN not obtained via Ford VIN read the device will attempt alternate VIN read. *flags_index* **Index** / **Description** 0 – Disable 1 – VIN enable

Instance	Description
	Index: 2 — Ford VIN Read enable Index: 3 — Ford VIN Read enable and VIN Read enable
obd[2]	Obd Debug Enable (OBD Datalogger). *flags_index* Index: 1 — Enable *mode_index* Index: 0 — Overwrite old debug data continuously Index: 1 — Cease logging when memory is full
obd[3]	Obd Monitor Only (no bus requests) Enable. *flags_index* Index: 0 — Disable Index: 1 — Enable Index: 2 — Listen Only CAN/J1708 during protocol detect
obd[4]	Obd Electric (JBUS only) Enable. *flags_index* Index: 0 — Disable Index: 1 — Enable
obd[5]	Obd ECO Enable. *flags_index* Index: 0 — Disable Index: 1 — Enable *mode_index* Index: 0 to 10 — Eco mode end trip minutes

Instance	Description
obd[6]	Obd Wake Listen Enable. *flags_index* **Index** / **Description** 0 Disable 1 Enable
obd[7]	Obd Wake GPS Movement Enable. *flags_index* **Index** / **Description** 0 Disable 1 Enable

Target	Default Value(s)
AAb1, AAb3, AAa4, AAb5, AAb6	obd[0] 1 1 obd[1] 1 1 obd[2] 1 0 obd[3] 0 1 obd[4] 0 0 obd[5] 1 3 obd[6] 1 0 obd[7] 1 0

6.39. OBD Ign Off Accum (oio)

- oio[0] -> Gps Lifetime Odometer ignition off accumulation
- oio[1] -> Gps Trip Odometer 1 ignition off accumulation
- oio[2] -> Gps Trip Odometer 2 ignition off accumulation

Configuration instances: 3

Command Type	Syntax
SET	**Command** :wycfg oio[x] <accum> **Response** :OK
READ	**Command** :rycfg oio[x] **Response** | oio[x]: offacc:<accum> (hex_val) :OK

Parameter	Description	
accum	**Index**	**Description**
	0	Do not accumulate distance when ignition is off
	1	Accumulate distance when ignition is off

Target	Default Value(s)
AAb1, AAb3, AAa4, AAb5, AAb6	oio[0] 0 oio[1] 0 oio[2] 0

6.40. 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

Instance	Description
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

Target	Default Value(s)
AAb1, AAb3, AAa4, AAb5, AAb6	ors[0] 1500 15 ors[1] 75 15

6.41. 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)

Target	Default Value(s)
AAb1, AAb3, AAa4, AAb5, AAb6	ota[0] 0

6.42. OBD Override (ovr)

Configuration instances: 3

Command Type	Syntax
SET	**Command** :wycfg ovr[x] <enable_value> **Response** :OK
READ	**Command** :rycfg ovr[x] **Response** | ovr[x]: enable:<enable_value> (hex_val) :OK

<table>
<tr><th>Parameter</th><th>Description</th></tr>
<tr><td>enable_value</td><td>
<table>
<tr><th>Instance</th><th>Description</th></tr>
<tr><td>ovr[0]</td><td>Override Engine Hours
<table>
<tr><th>Index</th><th>Description</th></tr>
<tr><td>0</td><td>use derived engine hours if ECU engine hours not available</td></tr>
<tr><td>1</td><td>use Derived Engine Hours only – ignore ECU engine hours</td></tr>
<tr><td>2</td><td>use ECU Engine Hours only (keep in persist) – do not calculate derived engine hours</td></tr>
</table>
</td></tr>
<tr><td>ovr[1]</td><td>Override ECU Vehicle Speed
<table>
<tr><th>Index</th><th>Description</th></tr>
<tr><td>0</td><td>use ECU Vehicle Speed</td></tr>
<tr><td>1</td><td>use GPS Vehicle Speed</td></tr>
</table>
</td></tr>
</table>
</td></tr>
<tr><td>enable_value (protocol)</td><td>
<table>
<tr><th>Instance</th><th>Description</th></tr>
<tr><td>ovr[2]</td><td>Override OBD Protocol Discovery
<table>
<tr><th>Index</th><th>Description</th></tr>
<tr><td>0</td><td>No override</td></tr>
<tr><td>0x001 to 0x200</td><td>OBD Protocol Table in OBD Run States Bits on p. 161</td></tr>
</table>
NOTE: Any specified Protocol ID will be the ONLY protocol attempted
</td></tr>
</table>
</td></tr>
</table>

Target	Default Value(s)
AAb1, AAb3, AAa4, AAb5, AAb6	ovr[0] 0 ovr[1] 0 ovr[2] 0

6.43. Packet Creation Append (pca)

Configuration instances: 16

Command Type	Syntax
SET	**Command** :wycfg pca[x] "<append_hex_string>" **Response** :OK
READ	**Command** :rycfg pca[x] **Response** | pca[x]: append_hex_string:"<append_hex_string>" :OK

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

Parameter	Description
append_hex_string	Refer to Packet Creation Append for configuration details

Target	Default Value(s)
AAb1, AAb3, AAa4, AAb5, AAb6	pca[x] ""

6.44. 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	See Packet Recipe on p. 127 for configuration details

Target	Default Value(s)
AAb1, AAb3, AAa4, AAb5, AAb6	pcr[x] ""

6.45. Packet Configure TPS (pct)

Configuration instances: 5

Command Type	Syntax
SET	**Command** :wycfg pct[x] <reason_lo> <reason_hi> <Tps_Tag0> <Sz_size0> ... <Tps_Tag9> <Sz_size9> **Response** :OK
READ	**Command** :rycfg pct[x] **Response** | pct[x]: rsnlo:<reason_lo> (hex_val), rsnhi:<reason_hi> (hex_val), tag0:<Tps_Tag0> (hex_val), size0:<Sz_size0> (hex_val), ... tag9:<Tps_Tag9> (hex_val), size9:<Sz_size9> (hex_val) :OK

Parameter	Range	Description
reason_lo	0 to 255	Reason code range low end for TPS event data fetch
reason_hi	0 to 255	Reason code range high end for TPS event data fetch
Tps_Tagn	0 to 65535	Tag identifier to be fetched from TPS engine and placed into an outgoing packet
Sz_Sizen	0 to 255	Data size of the TPS tag data being fetched from the TPS engine

Target	Default Value(s)
AAb1, AAb3, AAa4, AAb5, AAb6	all pct[x] args default to '0'

6.46. PDP Reset (pdp)

Configuration instances: 1

Command Type	Syntax
SET	**Command** :wycfg pdp <minutes> <sessions> **Response** :OK
READ	**Command** :rycfg pdp **Response** | pdp: rsttime:<minutes> (hex_val), rsstintv:<sessions> (hex_val) :OK

Parameter	Range	Description
minutes	0 to 65535	Time threshold to issue a PDP connection reset
sessions	0 to 255	Sessions threshold to issue a PDP connection reset

Target	Default Value(s)
AAb1, AAb3, AAa4, AAb5, AAb6	pdp 0 0

6.47. Packet Retry (prt)

- prt[0] -> Ignition off
- prt[1] -> Ignition on

Configuration instances: 2

Command Type	Syntax
SET	**Command** :wycfg prt[x] <seconds> **Response** :OK
READ	**Command** :rycfg prt[x] **Response** | prt[x]: sec:<seconds> (hex_val) :OK

Parameter	Range	Description
seconds	0 to 65535	Number of seconds to wait between message send attempts

Target	Default Value(s)
AAb1, AAb3, AAa4, AAb5, AAb6	prt[0] 300 prt[1] 15

6.48. SMS Receive Mask (sms)

Configuration instances: 1

Command Type	Syntax
SET	**Command** :wycfg sms "<sms source match>" **Response** :OK
READ	**Command** :rycfg sms <sms source match> **Response** | sms: numb:"<sms source match>" :OK

Parameter	Range	Description
sms source match	32 Char	ASCII string limiting SMS received messages to those whose source matches; 0 disables the match.

Target	Default Value(s)
AAb1, AAb3, AAa4, AAb5, AAb6	sms "0"

6.49. 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

<table>
<tr><th>Parameter</th><th>Range</th><th>Description</th></tr>
<tr><td>baudrate</td><td>N/A</td><td>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</td></tr>
<tr><td>dps_value</td><td>3 Char</td><td>Acceptable dps values: 8N1</td></tr>
<tr><td>Mode_value</td><td>N/A</td><td>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></td></tr>
<tr><td>Console_print</td><td>N/A</td><td>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></td></tr>
</table>

Target	Default Value(s)
AAb3, AAa4, AAb5, AAb6	sps[0] 115200 "8N1" 0 0 sps[1] 115200 "8N1" 0 0
AAb1	sps[0] 9600 "8N1" 3 0 sps[1] 115200 "8N1" 0 0

6.50. Serial Pass Thru Settings (sts)

Configuration instances: 1

Command Type	Syntax
SET	**Command** :wycfg sts <PktTimeoutMs> <PktByteCount><PktDataFormat> **Response** :OK
READ	**Command** :rycfg sts **Response** | sts: timeout:<PktTimeoutMs> (hex_val), numbyte:<PktByteCount> (hex_val), datafmt:<PktDataFormat> (hex_val) :OK

Parameter	Range	Description
PktTimeoutMs	0 to 60000	A non-zero value defines a passthru packet with an inter-byte timeout. Setting 0 disables this packet delimiter.
PktByteCount	0 to 256	A non-zero value defines a passthru packet as a specific number of bytes. Setting 0 disables this packet delimiter.
PktDataFormat	0 to 1	Represents the packet data format. A setting of 0 means passthru data sent out over cellular will be in ASCII format; 1 will be in binary hexadecimal.

Target	Default Value(s)
AAb1, AAb3, AAa4, AAb5, AAb6	sts 1000 0 0

6.51. SYNCH TIME THRESHOLD (STT)

Configuration instances: 1

Command Type	Syntax
SET	**Command** :wycfg stt <threshold> **Response** :OK
READ	**Command** :rycfg stt <threshold> **Response** | stt: thresh:<threshold> (hex_val) :OK

Parameter	Range	Description
threshold	3 to 255	If the current system time is more than +/- thresh seconds different than the latest GPS sourced time, an automatic system time-sync will occur.

Target	Default Value(s)
AAb1, AAb3, AAa4, AAb5, AAb6	stt 3

6.52. TPS Config Values (tcv)

Configuration instances: 1

Command Type	Syntax
SET	**Command** :wycfg tcv <destination_index> <timeout> <retries_value> <discovery_time> **Response** :OK
READ	**Command** :rycfg tcv **Response** | tcv: index:<destination_index> (hex_val), seconds:<timeout> (hex_val), retries:<retries_value> (hex_val), distime:<discovery_time> (hex_val) :OK

Parameter	Range	Description
Destination_ Index	0 to 9	The index of the DST that is used to point to the server
Timeout	1 to 4294967295	The time (sec) that the device will wait for a reply from the server
Retries_value	-1 to 32767	The number of times the device will try to reach the server again after a timeout before cancelling communications
Discovery_time	0 to 4294967295	The minimum amount of time (sec) that the device accepts a successful “discovery session”.

Target	Default Value(s)
AAb1, AAb3, AAa4, AAb5, AAb6	tcv 7 30 3 60

6.53. Tester Present Message (tpm)

Command Type	Syntax
SET	**Command** :wycfg tpm <enable> **Response** :OK
READ	**Command** :rycfg tpm **Response** | tpm: en:0 (0x0) :OK

Parameter	Range	Description
enable	0 to 1	Enables tester present message. Value: 0 — Disables OBDII tester present message Value: 1 — Enables OBDII tester present message

Target	Default Value(s)
AAb1, AAb3, AAa4, AAb5	tpm 0

6.54. 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. Index: 0 — Disables autostart Index: 1 — Enables autostart

Target	Default Value(s)
AAb1, AAb3, AAa4, AAb5, AAb6	tmr[x] 0 0

6.55. 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
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

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)

Parameter	Range	Description
Pid_list_2	0x0 to 0xFFFFFFFF	Bit list for supported PIDs (32 to 32) <reserved>
Flag	0 to 255	Reserved

Target	Default Value(s)
AAb1, AAb3, AAa4, AAb5, AAb6	tpv 8 24 0x00 0x00 0

6.56. OBD Transmit Delay (txd)

Configuration instances: 2

Command Type	Syntax
SET	**Command** :wycfg txd[x] <value ms> **Response** :OK
READ	**Command** :rycfg txd[x] **Response** | txd[x]: value:<value ms> (hex_val) :OK

Parameter	Description		
value	**Instance**	**Range**	**Description**
	txd[0]	0 to 800	Message transmit delay. Value (ms)
	txd[1]	0 to 10800000	Start OBD communication after ignition on detect (for immobilizer issues). Value (ms)

Target	Default Value(s)
AAb1, AAb3, AAa4, AAb5, AAb6	txd[0] 0 txd[1] 0

6.57. VERSION INFO (VER)

- ver[0] -> Script
- ver[1] -> Paramset
- ver[2] -> Overlay0
- ver[3] -> Overlay1
- ver[4] -> Overlay2
- ver[5] -> Overlay3
- ver[6] -> Overlay4
- ver[7] -> Tps

Configuration instances: 8

Command Type	Syntax
SET	**Command** :wycfg ver[x] "<string>" <revision no> <flags_hex> **Response** :OK
READ	**Command** :rycfg ver[x] **Response** | ver[x]: str:"<string>" rev:<revision no> (hex_val), flags:<flags_hex> :OK

Parameter	Range	Description
string	N/A	ASCII name for a loaded item (31 characters)
revision no.	0 to 4294967295	Assigned revision number for a loaded item
flags_hex	N/A	Bitmask of error status values pertaining to loaded item

Hex	Description
0x01	Default
0x02	Locally Modified
0x04	Error
0x40	Unused
0x80	Unknown

Target	Default Value(s)
AAb1, AAb3, AAa4, AAb5, AAb6	ver[0] "~unknown" 0 128

Target	Default Value(s)
	ver[1] "default" 0 1 ver[2] "~unused" 0 64 ver[3] "~unused" 0 64 ver[4] "~unused" 0 64 ver[5] "~unused" 0 64 ver[6] "~unused" 0 64 ver[7] "~unknown" 0 128

6.58. VIN (VIN)

Configuration instances: 1

Command Type	Syntax
SET	**Command** :wycfg vin "<string>" **Response** :OK
READ	**Command** :rycfg vin **Response** | vin: str:"<string>" :OK

Parameter	Rates	Description
String	17 char	Set VIN for vehicles that do not support VIN or have an invalid VIN

Target	Default Value(s)
AAb1, AAb3, AAa4, AAb5, AAb6	VIN ""

6.59. Variable Save Mask (vsm)

- vsm[0] > covers uservar8 0-15, uservar16 0-7, and uservar32 0-7
- vsm[1] > covers uservar8 16-32, uservar16 8-15, and uservar32 8-15

Configuration instances: 2

Command Type	Syntax
SET	**Command** :wycfg vsm[x] <32bit_mask> **Response** :OK
READ	**Command** :rycfg vsm[x] **Response** | vsm[x]: mask:<32bit_mask> :OK

Parameter	Range	Description
32bit_mask	0 to 0xFFFFFFFF	Bitmask value corresponding to which scriptable user variables are saved in persist data.

Target	Default Value(s)
AAb1, AAb3, AAa4, AAb5, AAb6	vsm[0] 0 vsm[1] 0

6.60. Wake Battery Voltage (wbv)

Configuration instances: 1

Command Type	Syntax
SET	**Command** :wycfg wbv <millivolts> **Response** :OK
READ	**Command** :rycfg wbv <millivolts> **Response** | wbv: wakemv:<millivolts> (hex_val) :OK

Parameter	Range	Description
millivolts	10000 to 16000	Configures the voltage threshold at which the device will wake from sleep if exceeded.

Target	Default Value(s)
AAb1, AAb3, AAa4, AAb5, AAb6	wbv 13200

7. Device Manager Protocol

7.1. Definitions

7.1.1. ACK/NAK

An ACK is used to signal that a command has been accepted. The ACK packet is an echo of the CMD packet's header with the Packet Type changed to ACK. A NAK is used any time a command cannot be accepted (due to improper command formatting, device busy, etc.). The NAK packet is an echo of the CMD packet's header with the Packet Type changed to NAK.

7.1.2. BROADCAST

A broadcast packet type does not expect a RESP/ACK/NAK, it is used to notify or send data that has been requested.

7.1.3. SESSION REJECT

A session reject message can be sent by the device at any time during a transaction to cancel the session.

7.2. Device Manager Packet

Byte(s)	Description
0 - 7	DM Header
8 - N	Packet-Specific Data

7.3. Device Manager Header

Byte(s)	Description
0 - 1	Packet Info
2 - 3	Session ID
4 - 7	ESN

7.3.1. Device Manager Header: Packet Info

Bit(s)	Description
0 - 3 (LSB)	Action/Info Code
4 - 7	Content Type
8 - 12	Content Index
13 - 15 (MSB)	Packet Type

Packet Info Register View

<table>
<tr><th>Bit Range/Type</th><th>Description</th></tr>
<tr><td>Action/Info Code: 0 - 15</td><td>
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>SessionAvailable</td></tr>
<tr><td>1</td><td>SessionOpen</td></tr>
<tr><td>2</td><td>SessionClose</td></tr>
<tr><td>3</td><td>QueryContent</td></tr>
<tr><td>4</td><td>Query</td></tr>
<tr><td>5</td><td>MultiBlockSetup</td></tr>
<tr><td>6</td><td>MultiBlockSend</td></tr>
<tr><td>7</td><td>MultiBlockResend</td></tr>
<tr><td>8</td><td>BlockSend</td></tr>
<tr><td>9</td><td>RunCMDLang (TBD)</td></tr>
<tr><td>10</td><td>CheckInNow (TBD)</td></tr>
<tr><td>11</td><td>SessionReject</td></tr>
</table>
</td></tr>
<tr><td>Content Type: 0 - 15</td><td>
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Bootloader</td></tr>
<tr><td>1</td><td>Application</td></tr>
<tr><td>2</td><td>Script</td></tr>
<tr><td>3</td><td>ParamSet</td></tr>
<tr><td>4</td><td>CheckIn</td></tr>
<tr><td>5</td><td>Control</td></tr>
<tr><td>6</td><td>TPS</td></tr>
<tr><td>7</td><td>-------</td></tr>
<tr><td>8</td><td>Overlay</td></tr>
<tr><td>9</td><td>CMDlang</td></tr>
<tr><td>10</td><td>ThirdParty Embedded Application</td></tr>
<tr><td>11</td><td>ThirdParty Data</td></tr>
</table>
</td></tr>
<tr><td>Content Index: 0 - 31</td><td>Describes the index of the active Content Type.
Overlays are the only content Type to have multiple instances so the content Index for an overlay could be 0, 1, 2, 3, or 4.
All other content types have Content Index = 0.</td></tr>
<tr><td>Packet Type: 0 - 7</td><td>
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>ACK</td></tr>
<tr><td>1</td><td>NAK</td></tr>
<tr><td>2</td><td>CMD</td></tr>
<tr><td>3</td><td>RESP</td></tr>
</table>
</td></tr>
</table>

Bit Range/Type	Description	
	Bit	**Description**
	4	BROADCAST
	5	N/A
	6	N/A
	7	N/A

7.4. Packet-Specific Data

7.4.1. BROADCAST: Session Available

Byte(s)	Description
8 to 11	Content Mask: one but flipped for each content type in session
12 to 15	Session IP(v4)
16 to 17	Session port

7.4.2. CMD: Multi-block Setup

Byte(s)	Description
8 to 9	Multi-block size code: the size of the block sent is (Multi-block size code * 16).

7.4.3. CMD: Multi-block Send

Byte(s)	Description
8 to 9	Block-start index: the index of the binary block to start with in the next multi-block sub-session. Where 0 = binary bytes 0 through ((Multi-block size code *16)-1).
10	Number of blocks: number of consecutive binary blocks to send starting with Block-start index (Valid values 0-31). The actual number of blocks desired/sent is (Number of blocks +1).

7.4.4. CMD: Multi-block Resend

Byte(s)	Description
8 to 11	Resend Mask: Each bit from LSB to MSB represents on of the binary blocks missed in the most recent Multi-block sub-session.

7.4.5. CMD: Block Send

Byte(s)	Description
8 to 11	Offset: binary offset of active content.
10	Size: binary size to send (Starting at offset).

7.4.6. RESP: Query Content

Byte(s)	Description
8 to 11	Content Mask: Each bit from LSB to MSB represents one instance of available content (32 possible).

Example: if three overlays are available, content mask = 0x7 (0b111).

7.4.7. RESP: Query

> **NOTE:** Bytes 12-N are only used for Script/ParamSet/Overlays/TPS, NOT for BootLoader/Application/ThirdParty.

Byte(s)	Description
8 to 11	Size: size of available content
12 to 15	Revision: unique revision number of available content.
16 to N	Version: zero-terminated ASCII string.

7.4.8. BROADCAST: Multi-block Send

Byte(s)	Description
8 to 9	Block-start index.
10	Multi-block sub-session index (Value 0-31).
11 to N	Block: binary block of size (Multi-block size code * 16).

7.4.9. RESP: Block Send

Byte(s)	Description
8 to 11	Offset: binary offset of active content.
12 to 15	Size: binary size to send (starting at offset).
16 to N	Block: binary block of size.

7.4.10. Example Device Manager Session

This example device manager session shows updating an interpreter script. The script used is an "empty script" which is quite useless on the device but works well for this example.

The binary contents of "empty_script.bin" are as follows:

```
00 00 00 00 00 00 00 00 00
```

Found below are the messages exchanged in the device manager session:

```
Broadcast:SessionAvail
50 80 ef be 15 cd 5b 07 04 00 00 00 7e 3c f1 32 57 30
-------------------------------------------------------------------------------
>
```

```
SessionOpen
51 40 ef be 15 cd 5b 07
<-------------------------------------------------------------------------------
--
```

```
Ack:SessionOpen
51 00 ef be 15 cd 5b 07
-------------------------------------------------------------------------------
---------------------------------------->
```

```
Query
24 40 ef be 15 cd 5b 07
<-------------------------------------------------------------------------------
----------------------------------------
```

```
Resp:Query
24 60 ef be 15 cd 5b 07 09 00 00 00 01 00 00 00 65 6d 70 74 79 5f 73 63 72 69 70 74
2e 62 69 6e 00
-------------------------------------------------------------------------------
>
```

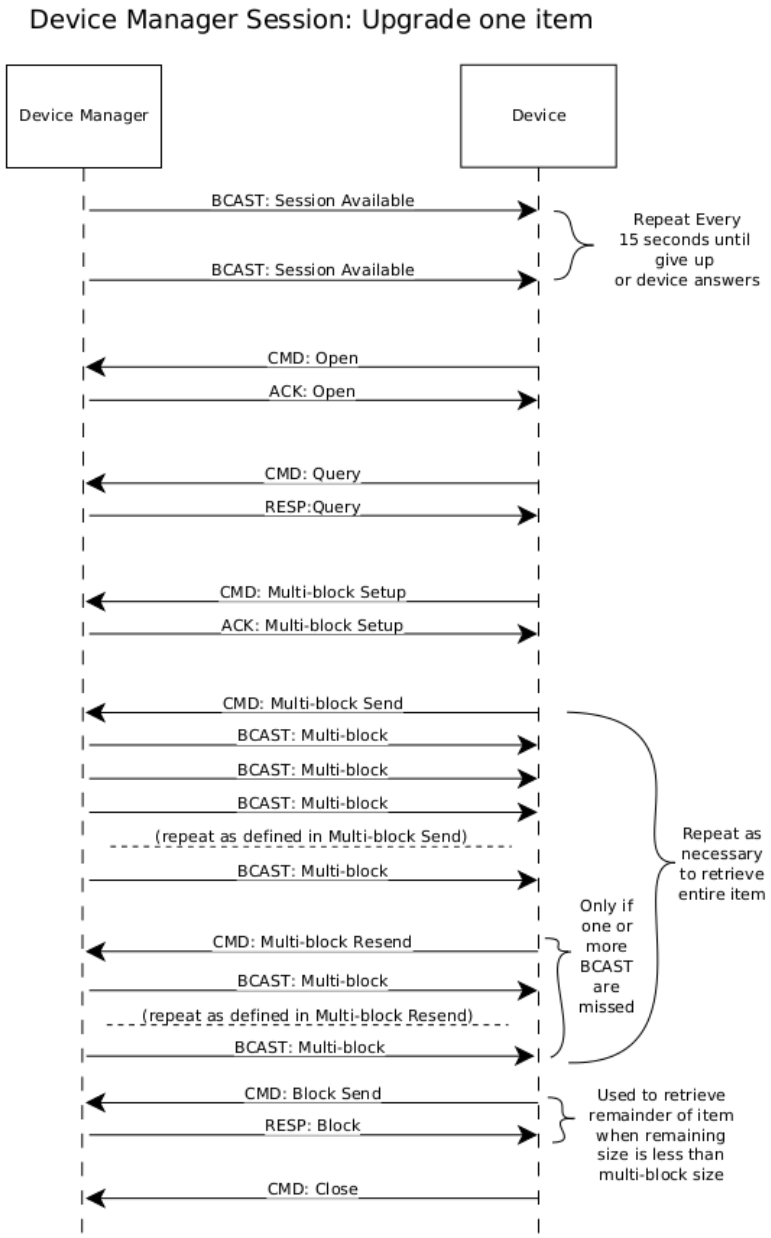
```
MultiBlockSetup
25 40 ef be 15 cd 5b 07 3f 00
<-------------------------------------------------------------------------------
----------------------------------------
```

```
Ack:MultiBlockSetup
25 00 ef be 15 cd 5b 07
 -------------------------------------------------------------------------------
---------------------------------------->
```

```
BlockSend
28 40 ef be 15 cd 5b 07 00 00 00 00 09 00 00 00
<-------------------------------------------------------------------------------
-
```

```
Resp:BlockSend
28 60 ef be 15 cd 5b 07 00 00 00 00 09 00 00 00 00 00 00 00 00 00 00 00 00 00 00
-------------------------------------------------------------------------------
>
```

```
SessionClose
52 80 ef be 15 cd 5b 07
<-------------------------------------------------------------------------------
-
```



7.5. Device Manager Check-in

NOTE: all 2 to 4 byte items are little endian byte-order.

Bit(s)	Description
0 – 7	DM Header (see description above)
8	Check-in Format Version
9 – N	Check-in Data

7.5.1. Check-in Data

Byte(s)	Size	Format Version	Description
9	1	0	Bootloader Version string size
10 - x	variable	0	Bootloader Version string
x	1	0	FW Version string size
x - x	variable	0	FW Version string
x	4	0	Script Revision Number
x	1	0	Script Flags
x	4	0	ParamSet Revision Number
x	1	0	ParamSet Flags
x	1	0	Number of Used Overlays
		0	one of the section for each Used Overlay
x	4	0	Overlay Revision Number
x	1	0	Overlay Flags
x	4	0	IP address (unsigned integer)
x	1	0	cell type: 0=CDMA, 1=GSM
		0	if cell type == 0 (CDMA)
x	1	0	MEID string size
x - x	variable	0	MEID string
x	1	0	MDN string size
x - x	variable	0	MDN string
		0	if cell type == 1 (GSM)
x	1	0	IMEI string size
x - x	variable	0	IMEI string
x	1	0	MSISDN string size
x - x	variable	0	MSISDN string
x	1	0	ICCID string size
x - x	variable	0	ICCID string
x	1	0	Third Party Version string size
x - x	variable	0	Third Party Version string
x	1	2	Check-In message info bitfield
x	4	3	TPS File Revision Number
x	1	3	TPS File Flags
x	1	4	Cell Module Model string size
x - x	variable	4	Cell Module Model string

Byte(s)	Size	Format Version	Description
x	1	4	Cell Module FW Version string size
x - x	variable	4	Cell Module FW Version string
x	1	5	I/O Box FW version string size (XT2500 Only)
x - x	variable	5	I/O Box FW version string (XT2500 Only)
x	4	5	I/O Box ESN (XT2500 Only)

7.5.2. Script/Parameter Set/Overlay Flags

Hex	Description
0x01	Default (Parameter Set ONLY)
0x02	Locally Modified (Parameter Set ONLY)
0x04	Error
0x40	Unused (Overlays ONLY)
0x80	Unknown (Script ONLY)

7.5.3. Check-in Message Info Bitfield Definitions

Hex	Description
0x01	Vehicle Comms active
0x02	TPS transfer active
0x04	N/A
0x08	N/A
0x10	N/A
0x20	N/A
0x40	N/A
0x80	N/A

7.5.4. Check-in Message Changelog

Rev	Description
0	Initial Offering
1	Stop truncating bootloader and main firmware revision strings
2	Added check-in message info bitfield
3	Added TPS file revision information
4	Added cell module model and FW version strings (JA1.1)
5	Added I/O Box FW version string and ESN (XT2500 Only)

8. INTERPRETER

8.1. LANGUAGE

8.1.1. 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 km/h 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 km/h AND the vehicle heading is within 10 degrees of North:

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

8.1.2. 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.

8.1.3. 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
```

8.1.4. 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
condact	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

Keyword	Description
	Action Block will ALWAYS run when the Trigger Block's test(s) are true.
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>, <b>)	Bitwise AND of arguments a and b
BitOr(<a>, <b>)	Bitwise OR of arguments a and b
BitXor(<a>, <b>)	Bitwise XOR of arguments a and b
BitShift(<a>, <b>)	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>, <b>)	Arithmetic addition of arguments a and b
Subtract(<a>, <b>)	Arithmetic subtraction of argument b from a
Multiply(<a>, <b>)	Arithmetic multiplication of arguments a and b

Tests

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

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

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

System Values

Script Language	Description
SystemVal(<index>)	Pass the enumeration index of the desired System Value to be accessed (see Packet Recipe Fields for acceptable range)
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)
DriverId1	Maps directly to DriverIdCode1 packet field (ID: 0x0e)
DriverId2	Maps directly to DriverIdCode2 packet field (ID: 0x0f)
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	
MotionState	Maps directly to OBDRunStates packet field (ID: 0x2c)
EngineState	Maps directly to OBDCommsState packet field (ID: 0x2d)
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)
ExtADC0	Maps directly to ExternADC0_1byte packet field (ID: 0x66)

Events

Script Language	Description
TimerExpired(<index>)	Evaluates to 1 when timer is expired; otherwise 0 *Range: 0 to 31*
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. *Range: 0 to 255*
AccelEventIsActive(<index>)	Evaluates to 1 when an accelerometer event is detected; otherwise 0

<table>
<tr><th>Script Language</th><th>Description</th></tr>
<tr><td></td><td>Range: 0 to 7</td></tr>
<tr><td>GarminResponseReceived (<index>)</td><td>Evaluates to 1 when a non-blacklisted Garmin response is received (and Garmin is active), otherwise 0

Range: 0 to 5

<table><tr><th>Value</th><th>Description</th></tr><tr><td>0</td><td>Ack (or Nak) packet (only triggered when specified in SendGarminMsg())</td></tr><tr><td>1</td><td>Unit ID (a.k.a. Garmin ESN) response</td></tr><tr><td>2</td><td>Product ID response</td></tr><tr><td>3</td><td>Throttle Message response</td></tr><tr><td>4</td><td>Ping response</td></tr><tr><td>5</td><td>Generic response; any response that isn't listed above and isn't blacklisted</td></tr></table></td></tr>
<tr><td>GarminUserPayloadReceived(0)</td><td>Evaluates to 1 when a payload (destined for garmin device) has been received; evaluates to 0 otherwise. This signals that a payload has been sent to the device by a user/server. Typically used to trigger a SendGarminMsg() action.</td></tr>
<tr><td>DriverIdRead(<index>)</td><td>Evaluates to 1 when driver id is detected on 1-wire channel <index>, otherwise 0

Range: 0 to 1</td></tr>
<tr><td>SystemEventIsActive(<index>)</td><td>Evaluates to 1 when system event <index> is active, otherwise 0

Range: 0 to 20

<table><tr><th>Value</th><th>Description</th></tr><tr><td>0</td><td>Reset</td></tr><tr><td>1</td><td>Wakeup</td></tr><tr><td>2</td><td>BootloaderUpgrade (not implemented)</td></tr><tr><td>3</td><td>FWUpgrade</td></tr><tr><td>4</td><td>ScriptUpgrade</td></tr><tr><td>5</td><td>ParamSetUpgrade</td></tr><tr><td>6</td><td>OverlayUpgrade (not implemented)</td></tr><tr><td>7</td><td>ManualConfigChange</td></tr><tr><td>8</td><td>CellRegistrationChange</td></tr><tr><td>9</td><td>IPChange</td></tr><tr><td>10</td><td>SMSReceived</td></tr><tr><td>11</td><td>SMSSendOK</td></tr><tr><td>12</td><td>SMSSendFailure</td></tr></table></td></tr>
</table>

Script Language	Description
	Value: 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.
BluetoothResponseReceived(0)	Evaluates to 1 when a bluetooth message is received, otherwise 0. (Only applicable to target AAb3)
TpsBlockCompleteEvent (<index>)	Evaluates to 1 when TPS block request on the vehicle bus has completed, otherwise 0. Value: 1 — Primary Bus 2 — Secondary Bus
AuxPassthruPacketReceived(0)	Evaluates to 1 when an aux passthru packet has been received, 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
UserVar8(<index>)	Evaluates to value stored in 8bit user variable Index range: 0 to 31
UserVar16(<index>)	Evaluates to value stored in 16bit user variable *Index range: 0 to 15*
UserVar32(<index>)	Evaluates to value stored in 32bit user variable *Index Range: 0 to 15*
FlagIsSet(<index>)	Evaluates to 1 when flag is set, otherwise 0 *Index Range: 0 to 31*

<table>
<tr><th>Script Language</th><th>Description</th></tr>
<tr><td>GeofenceState(<index>)</td><td>
<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>
Index range: 0 to 49
</td></tr>
<tr><td>InputState(<index>)</td><td>Evaluates to 1 when input is high/when condition is true, otherwise 0

Index range 0 to 7:

<table>
<tr><th>Index</th><th>Description</th></tr>
<tr><td>0</td><td>Ignition</td></tr>
<tr><td>1</td><td>Input 1</td></tr>
<tr><td>2</td><td>Input 2</td></tr>
<tr><td>3</td><td>Input 3</td></tr>
<tr><td>4</td><td>Input 4</td></tr>
<tr><td>5</td><td>Vibration Detected (via Accel)</td></tr>
<tr><td>6</td><td>Main supply power available</td></tr>
<tr><td>7</td><td>Accelerometer reorientation validity</td></tr>
</table>
</td></tr>
<tr><td>SystemState(<index>)</td><td>Evaluates to 1 when true, otherwise 0

Index range: 0 to 16

<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>
</td></tr>
</table>

<table>
<tr><th>Script Language</th><th>Description</th></tr>
<tr><td>GenericConfig(<index>)</td><td>Evaluates to value stored in gcv[<index>] config parameters.

Index range: 0 to 63</td></tr>
<tr><td>PacketsPending(<index>)</td><td>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></td></tr>
<tr><td>NoAckRxRetryCount(0)</td><td>Evaluates to number of retried failed ACK packets</td></tr>
<tr><td>DeviceResetReason(0)</td><td>Evaluates to most recent device reset reason as defined by the microprocessor

Resolution range: 0 to 9

<table><tr><th>Resolution</th><th>Description</th></tr><tr><td>0</td><td>ResetCause_PowerApplied
Main supply voltage connected to device</td></tr><tr><td>1</td><td>ResetCause_LowPowerMode
Reset caused by the micro Hibernation module (not utilized)</td></tr><tr><td>2</td><td>ResetCause_Watchdog
Reset caused by the micro Watchdog module</td></tr><tr><td>3</td><td>ResetCause_Software
Reset caused by our device firmware</td></tr><tr><td>4</td><td>ResetCause_User
Main micro reset pin toggled (not utilized)</td></tr><tr><td>5</td><td>ResetCause_BrownOut
Reset caused by the micro BOR module</td></tr><tr><td>6</td><td>ResetCause_SupplyMonitor
Not implemented</td></tr><tr><td>7</td><td>ResetCause_FactoryReset
Reset caused by the micro Hardware System Service Request</td></tr><tr><td>8</td><td>ResetCause_SystemError
Not implemented</td></tr><tr><td>9</td><td>ResetCause_Unknown
Catch for any other micro reset source we have not defined</td></tr></table></td></tr>
<tr><td>TpsEcusRemaining(<default value>)</td><td>Evaluates to TPS number of ECUs yet to be queried for the current Block ID</td></tr>
</table>

Script Language	Description
	Default value: 0
GetVarPktSize(<default value>)	Evaluates to generated variable packet byte count, otherwise 0. **Index** — **Description** 0x26 — Garmin Packet 0xca — Aux Passthru Packet 0x6f — Crash Packet 0x6b — Vehicle Bus DTC Packet 0x28 — Bluetooth Payload Packet 0x2a — Command Payload Packet 0xc5 — TPS Payload
AuxPassthruScratchpadSize(0)	Evaluates to aux passthru scratchpad size (in bytes).
AuxPassthruScratchpadByte (<index>)	Evaluates to the byte value at index of the aux passthru scratchpad.
IPAddress(0)	Device IP Address

Actions

Script Language	Description
ResetDevice()	N/A
ResetModem()	N/A
ResetGPS()	N/A
TurnOffGPS()	N/A
TurnOnGPS()	N/A
SetUserVar(<type>, <index>, <value>)	Type range: 0 to 2 **Type** — **Description** 0 — 8-bit variable 1 — 16-bit variable 2 — 32-bit variable Index range: **Index** — **Description** 0 to 31 — 8-bit variables 0 to 15 — 16-bit variables 0 to 15 — 32-bit variables Value range:

<table>
<tr><th>Script Language</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>-128 to 127</td><td>8-bit variables</td></tr>
<tr><td>-32768 to 32767</td><td>16-bit variables</td></tr>
<tr><td>-2147483648 to 2147483647</td><td>32-bit variables</td></tr>
</table>
</td></tr>
<tr><td>AdjustUserVar (<type>, <index>, <adjust_amount>)</td><td>Type range: 0 to 2
<table>
<tr><th>Type</th><th>Description</th></tr>
<tr><td>0</td><td>8bit variable</td></tr>
<tr><td>1</td><td>16bit variable</td></tr>
<tr><td>2</td><td>32bit variable</td></tr>
</table>
Index range:
<table>
<tr><th>Index</th><th>Description</th></tr>
<tr><td>0 to 31</td><td>8bit variables</td></tr>
<tr><td>0 to 15</td><td>16bit variables</td></tr>
<tr><td>0 to 15</td><td>32bit variables</td></tr>
</table>
Adjust_amount range:
<table>
<tr><th>Adjust Amount</th><th>Description</th></tr>
<tr><td>-128 to 127</td><td>8bit variables</td></tr>
<tr><td>-32768 to 32767</td><td>16bit variables</td></tr>
<tr><td>-2147483648 to 2147483647</td><td>32bit variables</td></tr>
</table>
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).</td></tr>
<tr><td>SetFlag(<flag_ index>)</td><td>Index range: 0 to 31</td></tr>
<tr><td>ClearFlag(<flag_ index>)</td><td>Index range: 0 to 31</td></tr>
<tr><td>StartTimer(<timer_ index>)</td><td>Index range: 0 to 31</td></tr>
<tr><td>StopTimer(<timer_ index>)</td><td>Index range: 0 to 31</td></tr>
<tr><td>ResetTimer(<timer_ index>)</td><td>Index range: 0 to 31</td></tr>
<tr><td>BuildAndSendMsg (<packet_id>, <reason_code>, <destination_id>,</td><td>
<table>
<tr><th>Parameter</th><th>Range</th><th>Description</th></tr>
<tr><td>Packet_id</td><td>0 to 254</td><td>Should be an ID configured with :wycfg pcr[x]</td></tr>
<tr><td>reason_code range</td><td>0 to 255</td><td>Assigns the reason_code to the index in the script. Any reason_code less than 0 or greater</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Script Language</th><th>Description</th></tr>
<tr><td><ack>)</td><td>
<table>
<tr><th>Parameter</th><th>Range</th><th>Description</th></tr>
<tr><td></td><td></td><td>than 255 will wrap.</td></tr>
<tr><td>destination_id range</td><td>0 to 9</td><td>Should be one of the destinations configured with :wycfg dst[x] ...</td></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.
</td></tr>
<tr><td>ClearLog()</td><td>Erases any messages/packets stored in flash</td></tr>
<tr><td>EnterDeepSleep (<wake_mask>, <wake_minutes>)</td><td>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>
</td></tr>
<tr><td>EnterSleep(<wake_ mask>, <wake_ minutes></td><td>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>
</td></tr>
<tr><td>SetGarminPower</td><td>power_state range: 0 – 1</td></tr>
</table>

<table>
<tr><th>Script Language</th><th>Description</th></tr>
<tr><td>(<power_state>)</td><td><table><tr><th>Range</th><th>Description</th></tr><tr><td>0</td><td>Off</td></tr><tr><td>1</td><td>On</td></tr></table></td></tr>
<tr><td>SendGarminMsg (<msg_id>, <flags>)</td><td><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></td></tr>
<tr><td>ClearTripOdom (<odom_index>)</td><td>Index range: 0 to 2
<table><tr><th>Range</th><th>Description</th></tr><tr><td>0</td><td>GPSTripOdom1</td></tr><tr><td>1</td><td>GPSTripOdom2</td></tr><tr><td>2</td><td>ObdDerivedTripOdom</td></tr></table></td></tr>
<tr><td>SetOutput(<output_ index>)</td><td>Index range: 0 to 2</td></tr>
<tr><td>ClearOutput (<output_index>)</td><td>Index range: 0 to 2</td></tr>
<tr><td>PulseOutput (<output_index>, <seconds_on>)</td><td>Index range: 0 to 2
seconds_on range: 1 to 65535</td></tr>
<tr><td>FlashOutput (<output_index>, <blink_rate>)</td><td>Index range: 0 to 2
Blink_rate range: 1 to 100 Hz</td></tr>
</table>

Script Language	Description
ClearDriverIds()	N/A
ClearDriverID (<index>)	<index> of 0 or 1
CheckInNow()	N/A
SetInput(<input_ index>)	*Input_index range: 0 to 3* If input state source is not configured to “script”, it is ignored
ClearInput(<input_ index>)	*Input_index range: 0 to 3* If input state source is not configured to “script”, it is ignored
BuzzerPlay(<song_ index> <volume> <loops>)	*Song_index range: 0 to 9* *Volume range: 0 to 100* *Loops clamped to range: 0 to 255; 0 is continuous play*
BuzzerStop()	N/A
ResetObdHarshAccel Cnt()	N/A
ResetObdHarshBrak eCnt()	N/A
BluetoothSendOk()	N/A
TempSenseScan()	N/A
TempSenseRead()	N/A
ClearObdData (<mask>)	Mask range: 0x00000000 to 0x0001FFFF **Range** / **Description** 0x000001 — Clear engine run data (e.g. rpm, speed) 0x000002 — Clear time data (e.g. engine hours, idle time) 0x000004 — Clear level data (e.g. fuel level, fuel used) 0x000008 — Clear distance data (e.g. ecu odometer) 0x000010 — Clear temperature data (e.g. coolant temp, fuel temp) 0x000020 — Clear pressure & load data (e.g. coolant pressure, eng torque) 0x000040 — Clear fuel economy data (e.g. mpg) 0x000080 — Clear violations data (e.g. harsh accel count) 0x00FFFF — Clear all the above 0x010000 — Clear persisted VIN 0x020000 — Clear DTCs
SaveDataToFlash()	Force a save of persistent data to NVM. **See footnote 4 below table.**
TpsLoadScratchpad (<tag>)	Load TPS scratchpad with the requested Tag data

Script Language	Description
VehicleBusListen (<timeout_s>)	Listen for traffic on vehicle bus for <timeout_s>
TpsRequestBlock (<block_id>)	Request the given TPS block on the vehicle bus
AuxPassthruScratchpadLoad()	Load the Aux Passthru scratchpad with the most recently-received packet
GsmSleep()	Put the cellular module to sleep
GsmListen()	Put the cellular module into listen mode
GsmWake()	Wake the cellular module up
BluetoothSleep()	Put the bluetooth module to sleep
BluetoothWake()	Wake the bluetooth module **NOTE:** Cannot be called if Bluetooth is already awake.
TpsSync()	Trigger a TPS server sync
ResetGpsHarshAccelCnt()	Resets the harsh accel count on the GPS.
ResetGpsHarshBrakeCnt()	Resets the harsh brake count on the GPS.

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.

8.1.5. Wake Reason Masks

<table>
<tr><th>Script Language</th><th>Description</th></tr>
<tr><td>EnterDeepSleep (<wake_mask>, <wake_minutes>)</td><td>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></td></tr>
<tr><td>EnterSleep(<wake_ mask>, <wake_ minutes></td><td>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></td></tr>
</table>

8.2. Interpreter Scripting Trigger Block Examples

8.2.1. 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)
```

```
    condact when Eq(FlagIsSet(0), 1) [Debounce(0, 0)]
        actions
            run BuildAndSendMsg(1, 0, 1, 0)
            run ClearFlag(0)
    condact 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)
    condact always
        actions
            run ResetTimer(0)
```

8.2.2. 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)
    condact always
        actions
            run BuildAndSendMsg(4, 32, 1, 0)
```

8.2.3. 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)]
    condact always
        actions
            run SetFlag(3)
```

8.2.4. 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)

8.2.5. Third Party 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

Field	Assigned Data
OBDPidReserved6	RR Tire Pressure
OBDPidReserved7	LR Tire Pressure
OBDPidReserved8	RRI Tire Pressure
OBDPidReserved9	SPARE Tire Pressure
OBDPidReserved10	Driver Seatbelt Status
OBDPidReserved11	Passenger Seatbelt Status
OBDPidReserved12	Airbag Lamp
OBDPidReserved13	PRNDL
OBDPidReserved14	Parking Brake State
OBDPidReserved15	Parking Brake Warning Lamp

8.3. Packet Recipe

8.3.1. Fields

<table>
<tr><th>ID</th><th>Name</th><th>Description</th></tr>
<tr><td>0x01</td><td>PacketID</td><td>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></td></tr>
<tr><td>0x02</td><td>FmCustomHeader</td><td>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></td></tr>
<tr><td>0x03</td><td>DeviceId</td><td>Unsigned integer representing numeric ESN
<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>100000000 to 999999999</td></tr>
</table></td></tr>
<tr><td>0x04</td><td>ReasonCode</td><td>Unsigned integer (any reason code < 0 or > 255 will wrap)
<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></td></tr>
<tr><td>0x05</td><td>PacketSerialNum</td><td>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>1</td><td>0 to 65535</td></tr>
</table></td></tr>
<tr><td>0x06</td><td>UnixTime</td><td>Unsigned integer representing number of seconds since Unix Epoch
<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>0x0 to 0xffffffff</td></tr>
</table></td></tr>
<tr><td>0x07</td><td>Latitude</td><td>Signed integer decimal value of 4byte hex string divided by 1000000 (useful range -90.0 to 90.0)</td></tr>
</table>

<table>
<tr><th>ID</th><th>Name</th><th>Description</th></tr>
<tr><td></td><td></td><td>
<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>
</td></tr>
<tr><td>0x08</td><td>Longitude</td><td>Signed integer decimal value of 4-byte hex string divided by 1000000 (useful range -180.0 to 180.0)
<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>
</td></tr>
<tr><td>0x09</td><td>Altitude</td><td>Signed integer
<table>
<tr><th>Bytes</th><th>Units</th><th>Resolution</th><th>Range</th></tr>
<tr><td>2</td><td>m</td><td>0.1</td><td>-3276.8 to 3276.7</td></tr>
</table>
</td></tr>
<tr><td>0x0a</td><td>Heading</td><td>Signed integer (useful range 0.0 to 360.0)
<table>
<tr><th>Bytes</th><th>Units</th><th>Resolution</th><th>Range</th></tr>
<tr><td>2</td><td>Degrees</td><td>1.00E-01</td><td>-3276.8 to 3276.7</td></tr>
</table>
</td></tr>
<tr><td>0x0b</td><td>GpsSpeed</td><td>Unsigned integer
<table>
<tr><th>Bytes</th><th>Units</th><th>Resolution</th><th>Range</th></tr>
<tr><td>1</td><td>km/h</td><td>1</td><td>0 to 255</td></tr>
</table>
</td></tr>
<tr><td>0x0c</td><td>InputStates</td><td>See Input State Bits 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>0x00 to 0xff</td></tr>
</table>
</td></tr>
<tr><td>0x0d</td><td>OutputStates</td><td>Bitfield: see Output State bits 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>0x00 to 0x1f</td></tr>
</table>
</td></tr>
<tr><td>0x0e</td><td>DriverIdCode1</td><td>Unsigned integer representing unique iButton ID</td></tr>
</table>

ID	Name	Description
		Bytes: 4 | Units: N/A | Resolution: 1 | Range: 0 to 4294967295
0x0f	DriverIdCode2	Unsigned integer representing unique iButton ID Bytes: 4 | Units: N/A | Resolution: 1 | Range: 0 to 4294967295
0x10	GSPTripOdom1	See Odometer Details Table. Bytes: 4 | Units: m | Resolution: 1 | Range: 0 to 4294967295
0x11	Flags	Bitfield Bytes: 4 | Units: N/A | Resolution: 1 | Range: 0 to 4294967295
0x12	Hdop	Unsigned integer Bytes: 1 | Units: DOP | Resolution: 0.1 | Range: 0.0 to 25.5
0x13	NumSats	Bytes: 1 | Units: N/A | Resolution: 1 | Range: 0 to 255
0x14	ReceiverSigStr	See Signal Quality Measure Table. Bytes: 2 | Units: dBm | Resolution: 1.23 | Range: -113 to -75
0x15	CellularCarrierId	Bytes: 2 | Units: N/A | Resolution: 1 | Range: 0x0 to 0xffff
0x16	InternalBattVolts	Possibly using a 2-byte unsigned integer representing mV (0 to 65535) Bytes: 1 | Units: V | Resolution: 0.1 | Range: 0.0 to 25.5

<table>
<tr><th>ID</th><th>Name</th><th>Description</th></tr>
<tr><td>0x17</td><td>VehicleBattVolts_1byte</td><td>If value is greater than 25.5v, it will cap 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></td></tr>
<tr><td>0x18</td><td>GpsLifetimeOdom</td><td>Unsigned integer; see Odometer Details Table.<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 4294967295</td></tr></table></td></tr>
<tr><td>0x19</td><td>AccelStartDateTime</td><td>Unsigned integer representing time from Unix Epoch (s)<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>0x0 to 0xffffffff</td></tr></table></td></tr>
<tr><td>0x1a</td><td>AccelStartLat</td><td>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></td></tr>
<tr><td>0x1b</td><td>AccelStartLong</td><td>Signed integer decimal value of 4-byte hex string divided by 1000000 (useful range -180.0 to 180.0)<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></td></tr>
<tr><td>0x1c</td><td>AccelStartSpeed</td><td><table><tr><th>Bytes</th><th>Units</th><th>Resolution</th><th>Range</th></tr><tr><td>1</td><td>km/h</td><td>1</td><td>0 to 255</td></tr></table></td></tr>
<tr><td>0x1d</td><td>AccelStartHeading</td><td><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></td></tr>
<tr><td>0x1e</td><td>MaxAccel</td><td><table><tr><th>Bytes</th><th>Units</th><th>Resolution</th><th>Range</th></tr><tr><td>2</td><td>mG</td><td>1</td><td>0 to 32767</td></tr></table></td></tr>
</table>

<table>
<tr><th>ID</th><th>Name</th><th>Description</th></tr>
<tr><td>0x1f</td><td>AccelEventDuration</td><td><table><tr><th>Bytes</th><th>Units</th><th>Resolution</th><th>Range</th></tr><tr><td>1</td><td>Seconds</td><td>0.1</td><td>0 to 255</td></tr></table></td></tr>
<tr><td>0x20</td><td>AccelEndDateTime</td><td>Unsigned integer representing time from Unix Epoch (s)<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>0x0 to 0xffffffff</td></tr></table></td></tr>
<tr><td>0x21</td><td>AccelEndLat</td><td>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></td></tr>
<tr><td>0x22</td><td>AccelEndLong</td><td>Signed integer decimal value of 4-byte hex string divided by 1000000 (useful range -180.0 to 180.0)<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></td></tr>
<tr><td>0x23</td><td>AccelEndSpeed</td><td><table><tr><th>Bytes</th><th>Units</th><th>Resolution</th><th>Range</th></tr><tr><td>1</td><td>km/h</td><td>1</td><td>0 to 255</td></tr></table></td></tr>
<tr><td>0x24</td><td>AccelEndHeading</td><td><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></td></tr>
<tr><td>0x25</td><td>Not implemented</td><td>Available for future use</td></tr>
<tr><td>0x26</td><td>GarminPacketSize</td><td>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></td></tr>
<tr><td>0x27</td><td>GarminPacket</td><td>Garmin Packet bytes: <DLE><pid><size><payload><sum><DLE><ETX> and DLE stuffing</td></tr>
</table>

ID	Name	Description
		Bytes: Variable; Units: N/A; Resolution: N/A; Range: N/A
0x28	BluetoothPayloadSize *	Unsigned integer Bytes: 2; Units: N/A; Resolution: N/A; Range: N/A
0x29	BluetoothPayload *	Bytes Bytes: Variable; Units: N/A; Resolution: N/A; Range: N/A
0x2a	CommandPayloadSize *	Bytes: 2; Units: N/A; Resolution: N/A; Range: N/A
0x2b	CommandPayload *	Bytes: Variable; Units: N/A; Resolution: N/A; Range: N/A
0x2c	OBDRunStates	Bitfield; see OBD Run States Bits Table Bytes: 1; Units: N/A; Resolution: N/A; Range: 0 to 7
0x2d	OBDCommsState	See OBD Comms State Table Bytes: 1; Units: N/A; Resolution: N/A; Range: 0 to 3
0x2e	ObdDerivedTripOdom	Unsigned integer; see Odometer Details Table Bytes: 4; Units: m; Resolution: 1; Range: 0 to 4294967295
0x2f	ObdLifetimeOdom	Unsigned integer; see Odometer Details Table Bytes: 4; Units: m; Resolution: 1; Range: 0 to 4294967295

ID	Name	Description
0x30	TempSensor0	Bytes: 1 | Units: N/A | Resolution: N/A | Range: N/A
0x31	TempSensor1	Bytes: 1 | Units: N/A | Resolution: N/A | Range: N/A
0x32	UserVar8[0]	8-bit signed integer Bytes: 1 | Units: N/A | Resolution: 1 | Range: -128 to 127
0x33	UserVar8[1]	8-bit signed integer Bytes: 1 | Units: N/A | Resolution: 1 | Range: -128 to 127
0x34	UserVar8[2]	8-bit signed integer Bytes: 1 | Units: N/A | Resolution: 1 | Range: -128 to 127
0x35	UserVar8[3]	8-bit signed integer Bytes: 1 | Units: N/A | Resolution: 1 | Range: -128 to 127
0x36	UserVar8[4]	8-bit signed integer Bytes: 1 | Units: N/A | Resolution: 1 | Range: -128 to 127
0x37	UserVar8[5]	8-bit signed integer Bytes: 1 | Units: N/A | Resolution: 1 | Range: -128 to 127
0x38	UserVar8[6]	8-bit signed integer

ID	Name	Description			
		Bytes	Units	Resolution	Range
		1	N/A	1	-128 to 127
0x39	UserVar8[7]	8-bit signed integer			
		Bytes	Units	Resolution	Range
		1	N/A	1	-128 to 127
0x3a	UserVar8[8]	8-bit signed integer			
		Bytes	Units	Resolution	Range
		1	N/A	1	-128 to 127
0x3b	UserVar8[9]	8-bit signed integer			
		Bytes	Units	Resolution	Range
		1	N/A	1	-128 to 127
0x3c	UserVar8[10]	8-bit signed integer			
		Bytes	Units	Resolution	Range
		1	N/A	1	-128 to 127
0x3d	UserVar8[11]	8-bit signed integer			
		Bytes	Units	Resolution	Range
		1	N/A	1	-128 to 127
0x3e	UserVar8[12]	8-bit signed integer			
		Bytes	Units	Resolution	Range
		1	N/A	1	-128 to 127
0x3f	UserVar8[13]	8-bit signed integer			
		Bytes	Units	Resolution	Range
		1	N/A	1	-128 to 127
0x40	UserVar8[14]	8-bit signed integer			

ID	Name	Description
		Bytes: 1 | Units: N/A | Resolution: 1 | Range: -128 to 127
0x41	UserVar8[15]	8-bit signed integer Bytes: 1 | Units: N/A | Resolution: 1 | Range: -128 to 127
0x42	UserVar16[0]	16-bit signed integer Bytes: 2 | Units: N/A | Resolution: 1 | Range: -32768 to 32767
0x43	UserVar16[1]	16-bit signed integer Bytes: 2 | Units: N/A | Resolution: 1 | Range: -32768 to 32767
0x44	UserVar16[2]	16-bit signed integer Bytes: 2 | Units: N/A | Resolution: 1 | Range: -32768 to 32767
0x45	UserVar16[3]	16-bit signed integer Bytes: 2 | Units: N/A | Resolution: 1 | Range: -32768 to 32767
0x46	UserVar16[4]	16-bit signed integer Bytes: 2 | Units: N/A | Resolution: 1 | Range: -32768 to 32767
0x47	UserVar16[5]	16-bit signed integer Bytes: 2 | Units: N/A | Resolution: 1 | Range: -32768 to 32767
0x48	UserVar16[6]	16-bit signed integer

ID	Name	Description
		Bytes: 2 | Units: N/A | Resolution: 1 | Range: -32768 to 32767
0x49	UserVar16[7]	16-bit signed integer Bytes: 2 | Units: N/A | Resolution: 1 | Range: -32768 to 32767
0x4a	UserVar32[0]	32-bit signed integer Bytes: 4 | Units: N/A | Resolution: 1 | Range: -2147483648 to 2147483647
0x4b	UserVar32[1]	32-bit signed integer Bytes: 4 | Units: N/A | Resolution: 1 | Range: -2147483648 to 2147483647
0x4c	UserVar32[2]	32-bit signed integer Bytes: 4 | Units: N/A | Resolution: 1 | Range: -2147483648 to 2147483647
0x4d	UserVar32[3]	32-bit signed integer Bytes: 4 | Units: N/A | Resolution: 1 | Range: -2147483648 to 2147483647
0x4e	UserVar32[4]	32-bit signed integer Bytes: 4 | Units: N/A | Resolution: 1 | Range: -2147483648 to 2147483647
0x4f	UserVar32[5]	32-bit signed integer

ID	Name	Description
		Bytes: 4 | Units: N/A | Resolution: 1 | Range: -2147483648 to 2147483647
0x50	UserVar32[6]	32-bit signed integer Bytes: 4 | Units: N/A | Resolution: 1 | Range: -2147483648 to 2147483647
0x51	UserVar32[7]	32-bit signed integer Bytes: 4 | Units: N/A | Resolution: 1 | Range: -2147483648 to 2147483647
0x52	SystemStates	Bitfield; see System State Bits Table Bytes: 4 | Units: N/A | Resolution: 1 | Range: 0 to 4294967295
0x53	WakeReason	Hex byte value; see Wake Reason Values Table Bytes: 1 | Units: N/A | Resolution: 1 | Range: 0 to 255
0x54	ObdTrueOdometer	32-bit unsigned integer; see Odometer Details Table Bytes: 4 | Units: m | Resolution: 1 | Range: 0 to 858993459
0x55	ObdTotFuelUsed	32-bit unsigned integer Bytes: 4 | Units: liters | Resolution: 1 | Range: 0 to 4294967295
0x56	ObdTotEngHours	32-bit unsigned integer Bytes: 4 | Units: hours | Resolution: 1 | Range: 0 to 4294967295

ID	Name	Description
0x57	ObdVehicleSpeed	16-bit unsigned integer Bytes: 2 | Units: km/h | Resolution: 0.1 | Range: 0 to 2550
0x58	ObdEngRpm	16-bit unsigned integer Bytes: 2 | Units: rpm | Resolution: 1 | Range: 0 to 65535
0x59	ObdEngCoolantTemp	16-bit signed integer Bytes: 2 | Units: Celsius | Resolution: 1 | Range: -40 to 215
0x5a	ObdFuelLevelPct	16-bit unsigned integer Bytes: 2 | Units: % | Resolution: 0.1 | Range: 0 to 1000
0x5b	ObdTotDrivingSec	32-bit unsigned integer Bytes: 4 | Units: s | Resolution: 1 | Range: 0 to 4294967295
0x5c	ObdTotCruiseSec	32-bit unsigned integer Bytes: 4 | Units: s | Resolution: 1 | Range: 0 to 4294967295
0x5d	ObdTotIdleSec	32-bit unsigned integer Bytes: 4 | Units: s | Resolution: 1 | Range: 0 to 4294967295
0x5e	ObdTotIdleFuel	32-bit unsigned integer Bytes: 4 | Units: liters | Resolution: 1 | Range: 0 to 4294967295

<table>
<tr><th>ID</th><th>Name</th><th>Description</th></tr>
<tr><td>0x5f</td><td>ObdHarshBreakTotCnt</td><td>32-bit unsigned integer
<table><tr><th>Bytes</th><th>Units</th><th>Resolution</th><th>Range</th></tr><tr><td>4</td><td></td><td>1</td><td>0 to 4294967295</td></tr></table></td></tr>
<tr><td>0x60</td><td>ObdSpeedExceedTotCnt</td><td>32-bit unsigned integer
<table><tr><th>Bytes</th><th>Units</th><th>Resolution</th><th>Range</th></tr><tr><td>4</td><td></td><td>1</td><td>0 to 4294967295</td></tr></table></td></tr>
<tr><td>0x61</td><td>ObdRPMExceedTotCnt</td><td>32-bit unsigned integer
<table><tr><th>Bytes</th><th>Units</th><th>Resolution</th><th>Range</th></tr><tr><td>4</td><td></td><td>1</td><td>0 to 4294967295</td></tr></table></td></tr>
<tr><td>0x62</td><td>ObdHarshAccelTotCnt</td><td>32-bit unsigned integer
<table><tr><th>Bytes</th><th>Units</th><th>Resolution</th><th>Range</th></tr><tr><td>4</td><td></td><td>1</td><td>0 to 4294967295</td></tr></table></td></tr>
<tr><td>0x63</td><td>IgnSource</td><td>See Ignition Sense (ign) on p. 63
NOTE: Does not include 0x8000 (ANDMODE bit)
<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>0x0000 to 0xffff</td></tr></table></td></tr>
<tr><td>0x64</td><td>BluetoothName *</td><td><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></td></tr>
<tr><td>0x65</td><td>ObdVIN</td><td><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></td></tr>
<tr><td>0x66</td><td>ExternADC0_1byte</td><td>Value caps at 25.5v</td></tr>
</table>

<table>
<tr><th>ID</th><th>Name</th><th>Description</th></tr>
<tr><td></td><td></td><td><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></td></tr>
<tr><td>0x67</td><td>Reserved *</td><td><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></td></tr>
<tr><td>0x68</td><td>Accel Metrics</td><td>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></td></tr>
<tr><td>0x69</td><td>Boot Status</td><td>8-bit unsigned integer; see Boot Status 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 7</td></tr></table></td></tr>
<tr><td>0x6a</td><td>Apn Index</td><td>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></td></tr>
<tr><td>0x6b</td><td>ObdDtcPacketSize</td><td>See DTC Packet Parsing on p. 158 for more information.<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></td></tr>
<tr><td>0x6c</td><td>ObdDtcPacket</td><td>See DTC Packet Parsing on p. 158 for more information.<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></td></tr>
<tr><td>0x6d</td><td>ObdBackoff</td><td>See OBD Backoff Mode 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>N/A</td><td>0 to 3</td></tr></table></td></tr>
<tr><td>0x6e</td><td>ObdProtocols</td><td>See OBD Protocol Table , Secondary protocol in MS word, Primary protocol in LS word. e.g. secondary J1708 + primary J1939 = (0x00400100)</td></tr>
</table>

ID	Name	Description
		Bytes: 4 | Units: Hex | Resolution: N/A | Range: N/A
0x6f	CrashPacketSize	Unsigned integer Bytes: 2 | Units: N/A | Resolution: N/A | Range: 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 Bytes: Variable | Units: N/A | Resolution: N/A | Range: N/A
0x71	ObdTotPTOTime	32-bit unsigned integer Bytes: 4 | Units: seconds | Resolution: 1 | Range: 0 to 4294967295
0x72	ObdTotPTOFuel	32-bit unsigned integer Bytes: 4 | Units: liters | Resolution: 1 | Range: 0 to 4294967295
0x73	FwRev	16-byte null-terminated ascii string. Version string cuts off the first 9 bytes, so "AAb1-1133KB1.1-3c74751" would report as "KB1.1-3c74751". Bytes: 16 | Units: ascii | Resolution: N/A | Range: N/A
0x74	OBDPidReserved1	Signed integer decimal value of 4-byte hex string divided by 10 Bytes: 4 | Units: N/A | Resolution: 0.1 | Range: -2147483648 to 2147483647
0x75	OBDPidReserved2	Signed integer decimal value of 4byte hex string divided by 10

ID	Name	Description
		Bytes: 4 | Units: N/A | Resolution: 0.1 | Range: -2147483648 to 2147483647
0x76	OBDPidReserved3	Signed integer decimal value of 4byte hex string divided by 10 Bytes: 4 | Units: N/A | Resolution: 0.1 | Range: -2147483648 to 2147483647
0x77	OBDPidReserved4	Signed integer decimal value of 4byte hex string divided by 10 Bytes: 4 | Units: N/A | Resolution: 0.1 | Range: -2147483648 to 2147483647
0x78	OBDPidReserved5	Signed integer decimal value of 4byte hex string divided by 10 Bytes: 4 | Units: N/A | Resolution: 0.1 | Range: -2147483648 to 2147483647
0x79	ObdTransGear	16-bit unsigned integer Bytes: 2 | Units: 0,1,2,3 | Resolution: 1 | Range: 0=neutral 1=forward 2=reverse 3=park
0x7A	ObdFuelTemp	16-bit signed integer Bytes: 2 | Units: Celsius | Resolution: 1 | Range: -40 to 215
0x7B	ObdOilTemp	16-bit signed integer Bytes: 2 | Units: Celsius | Resolution: 1 | Range: -40 to 215

ID	Name	Description
0x7C	ObdThottlePos	16-bit unsigned integer Bytes: 2 | Units: % | Resolution: 1 | Range: 0 to 100
0x7D	ObdMPG	16-bit unsigned integer Bytes: 2 | Units: mpg | Resolution: 0.1 | Range: 0.0 to 256.0
0x7E	ObdAccelPos	16-bit unsigned integer Bytes: 2 | Units: % | Resolution: 1 | Range: 0 to 100
0x7F	ObdEngLoad	16-bit unsigned integer Bytes: 2 | Units: % | Resolution: 1 | Range: 0 to 250
0x80	ObdEngTorque	16-bit signed integer Bytes: 2 | Units: % | Resolution: 1 | Range: -125 to 125
0x81	ObdOilLevel	16-bit unsigned integer Bytes: 2 | Units: % | Resolution: 1 | Range: 0 to 100
0x82	ObdOilPressure	16-bit unsigned integer Bytes: 2 | Units: kPa | Resolution: 1 | Range: 0 to 100
0x83	ObdCoolantPressure	16-bit unsigned integer Bytes: 2 | Units: kPa | Resolution: 1 | Range: 0 to 100

ID	Name	Description
0x84	ObdIntakeAirTemp	16-bit signed integer Bytes: 2 | Units: Celsius | Resolution: 1 | Range: -40 to 215
0x85	ObdManifoldTemp	16-bit signed integer Bytes: 2 | Units: Celsius | Resolution: 1 | Range: -40 to 215
0x86	ObdCoolantLevel	16-bit unsigned integer Bytes: 2 | Units: % | Resolution: 1 | Range: 0 to 100
0x87	LinkageVer	32-bit unsigned integer; see Linkage Version Table Bytes: 4 | Units: N/A | Resolution: 1 | Range: 0x0 to 0xffffffff
0x88	GSPTripOdom2	See Odometer Details Table Bytes: 4 | Units: m | Resolution: 1 | Range: 0 to 4294967295
0x89	GenCfgVal0	Generic Configuration Value set via paramset Bytes: 4 | Units: N/A | Resolution: 1 | Range: 0 to 4294967295
0x8a	GenCfgVal1	Generic Configuration Value set via paramset Bytes: 4 | Units: N/A | Resolution: 1 | Range: 0 to 4294967295
0x8b	GenCfgVal2	Generic Configuration Value set via paramset Bytes: 4 | Units: N/A | Resolution: 1 | Range: 0 to 4294967295

<table>
<tr><th>ID</th><th>Name</th><th>Description</th></tr>
<tr><td>0x8c</td><td>GenCfgVal3</td><td>Generic Configuration Value set via paramset
<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></td></tr>
<tr><td>0x8d</td><td>GenCfgVal4</td><td>Generic Configuration Value set via paramset
<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></td></tr>
<tr><td>0x8e</td><td>OBDPidReserved6</td><td>Signed integer decimal value of 4-byte hex string divided by 10
<table><tr><th>Bytes</th><th>Units</th><th>Resolution</th><th>Range</th></tr><tr><td>4</td><td>N/A</td><td>0.1</td><td>-2147483648 to 2147483647</td></tr></table></td></tr>
<tr><td>0x8f</td><td>OBDPidReserved7</td><td>Signed integer decimal value of 4-byte hex string divided by 10
<table><tr><th>Bytes</th><th>Units</th><th>Resolution</th><th>Range</th></tr><tr><td>4</td><td>N/A</td><td>0.1</td><td>-2147483648 to 2147483647</td></tr></table></td></tr>
<tr><td>0x90</td><td>OBDPidReserved8</td><td>Signed integer decimal value of 4-byte hex string divided by 10
<table><tr><th>Bytes</th><th>Units</th><th>Resolution</th><th>Range</th></tr><tr><td>4</td><td>N/A</td><td>0.1</td><td>-2147483648 to 2147483647</td></tr></table></td></tr>
<tr><td>0x91</td><td>OBDPidReserved9</td><td>Signed integer decimal value of 4-byte hex string divided by 10
<table><tr><th>Bytes</th><th>Units</th><th>Resolution</th><th>Range</th></tr><tr><td>4</td><td>N/A</td><td>0.1</td><td>-2147483648 to 2147483647</td></tr></table></td></tr>
<tr><td>0x92</td><td>OBDPidReserved10</td><td>Signed integer decimal value of 4-byte hex string divided by 10
<table><tr><th>Bytes</th><th>Units</th><th>Resolution</th><th>Range</th></tr><tr><td>4</td><td>N/A</td><td>0.1</td><td>-2147483648 to 2147483647</td></tr></table></td></tr>
</table>

ID	Name	Description
0x93	OBDHarshAccelValue	Unsigned integer decimal value of acceleration in mph/second Bytes: 1 | Units: Mph/s | Resolution: 1 | Range: 0 to 255
0x94	OBDHarshBrakeValue	Unsigned integer decimal value of deceleration in mph/second Bytes: 1 | Units: Mph/s | Resolution: 1 | Range: 0 to 255
0x95	CellSerialId	24-byte null-terminated ascii string Bytes: 24 | Units: ascii | Resolution: N/A | Range: N/A
0x96	PktChkSum	Check sum for the packet, calculated so all bytes sum to 0x00 Bytes: 1 | Units: N/A | Resolution: N/A | Range: N/A
0x97	TPSFields0	Inclusion of a configured length variable content section defined by the corresponding pct config entry Bytes: variable | Units: N/A | Resolution: N/A | Range: N/A
0x98	TPSFields1	Inclusion of a configured length variable content section defined by the corresponding pct config entry Bytes: variable | Units: N/A | Resolution: N/A | Range: N/A
0x99	TPSFields2	Inclusion of a configured length variable content section defined by the corresponding pct config entry Bytes: variable | Units: N/A | Resolution: N/A | Range: N/A
0x9a	TPSFields3	Inclusion of a configured length variable content section defined by the corresponding pct config entry

ID	Name	Description
		Bytes: variable; Units: N/A; Resolution: N/A; Range: N/A
0x9b	TPSFields4	Inclusion of a configured length variable content section defined by the corresponding pct config entry Bytes: variable; Units: N/A; Resolution: N/A; Range: N/A
0x9c	OBDPidReserved11	4-byte ascii normalization of the vehicle response Bytes: 4; Units: ascii; Resolution: N/A; Range: N/A
0x9d	OBDPidReserved12	4-byte ascii normalization of the vehicle response Bytes: 4; Units: ascii; Resolution: N/A; Range: N/A
0x9e	OBDPidReserved13	4-byte ascii normalization of the vehicle response Bytes: 4; Units: ascii; Resolution: N/A; Range: N/A
0x9f	OBDPidReserved14	4-byte ascii normalization of the vehicle response Bytes: 4; Units: ascii; Resolution: N/A; Range: N/A
0xa0	OBDPidReserved15	4-byte ascii normalization of the vehicle response Bytes: 4; Units: ascii; Resolution: N/A; Range: N/A
0xa1	Debug-PacketSize	Unsigned integer Bytes: 2; Units: N/A; Resolution: N/A; Range: N/A

ID	Name	Description
0xa2	DebugPacket	Payload Bytes: variable; Units: N/A; Resolution: N/A; Range: N/A
0xa3	UserVar8[16]	8-bit signed integer Bytes: 1; Units: N/A; Resolution: 1; Range: -128 to 127
0xa4	UserVar8[17]	8-bit signed integer Bytes: 1; Units: N/A; Resolution: 1; Range: -128 to 127
0xa5	UserVar8[18]	8-bit signed integer Bytes: 1; Units: N/A; Resolution: 1; Range: -128 to 127
0xa6	UserVar8[19]	8-bit signed integer Bytes: 1; Units: N/A; Resolution: 1; Range: -128 to 127
0xa7	UserVar8[20]	8-bit signed integer Bytes: 1; Units: N/A; Resolution: 1; Range: -128 to 127
0xa8	UserVar8[21]	8-bit signed integer Bytes: 1; Units: N/A; Resolution: 1; Range: -128 to 127
0xa9	UserVar8[22]	8-bit signed integer Bytes: 1; Units: N/A; Resolution: 1; Range: -128 to 127

ID	Name	Description
0xaa	UserVar8[23]	8-bit signed integer Bytes: 1 | Units: N/A | Resolution: 1 | Range: -128 to 127
0xab	UserVar8[24]	8-bit signed integer Bytes: 1 | Units: N/A | Resolution: 1 | Range: -128 to 127
0xac	UserVar8[25]	8-bit signed integer Bytes: 1 | Units: N/A | Resolution: 1 | Range: -128 to 127
0xad	UserVar8[26]	8-bit signed integer Bytes: 1 | Units: N/A | Resolution: 1 | Range: -128 to 127
0xae	UserVar8[27]	8-bit signed integer Bytes: 1 | Units: N/A | Resolution: 1 | Range: -128 to 127
0xaf	UserVar8[28]	8-bit signed integer Bytes: 1 | Units: N/A | Resolution: 1 | Range: -128 to 127
0xb0	UserVar8[29]	8-bit signed integer Bytes: 1 | Units: N/A | Resolution: 1 | Range: -128 to 127
0xb1	UserVar8[30]	8-bit signed integer Bytes: 1 | Units: N/A | Resolution: 1 | Range: -128 to 127

ID	Name	Description
0xb2	UserVar8[31]	8-bit signed integer Bytes: 1 | Units: N/A | Resolution: 1 | Range: -128 to 127
0xb3	UserVar16[8]	16-bit signed integer Bytes: 2 | Units: N/A | Resolution: 1 | Range: -32768 to 32767
0xb4	UserVar16[9]	16-bit signed integer Bytes: 2 | Units: N/A | Resolution: 1 | Range: -32768 to 32767
0xb5	UserVar16[10]	16-bit signed integer Bytes: 2 | Units: N/A | Resolution: 1 | Range: -32768 to 32767
0xb6	UserVar16[11]	16-bit signed integer Bytes: 2 | Units: N/A | Resolution: 1 | Range: -32768 to 32767
0xb7	UserVar16[12]	16-bit signed integer Bytes: 2 | Units: N/A | Resolution: 1 | Range: -32768 to 32767
0xb8	UserVar16[13]	16-bit signed integer Bytes: 2 | Units: N/A | Resolution: 1 | Range: -32768 to 32767
0xb9	UserVar16[14]	16-bit signed integer Bytes: 2 | Units: N/A | Resolution: 1 | Range: -32768 to 32767

ID	Name	Description
0xba	UserVar16[15]	16-bit signed integer Bytes: 2 | Units: N/A | Resolution: 1 | Range: -32768 to 32767
0xbb	UserVar32[8]	32-bit signed integer Bytes: 4 | Units: N/A | Resolution: 1 | Range: -2147483648 to 2147483647
0xbc	UserVar32[9]	32-bit signed integer Bytes: 4 | Units: N/A | Resolution: 1 | Range: -2147483648 to 2147483647
0xbd	UserVar32[10]	32-bit signed integer Bytes: 4 | Units: N/A | Resolution: 1 | Range: -2147483648 to 2147483647
0xbe	UserVar32[11]	32-bit signed integer Bytes: 4 | Units: N/A | Resolution: 1 | Range: -2147483648 to 2147483647
0xbf	UserVar32[12]	32-bit signed integer Bytes: 4 | Units: N/A | Resolution: 1 | Range: -2147483648 to 2147483647
0xc0	UserVar32[13]	32-bit signed integer Bytes: 4 | Units: N/A | Resolution: 1 | Range: -2147483648 to 2147483647

ID	Name	Description
0xc1	UserVar32[14]	32-bit signed integer Bytes: 4 | Units: N/A | Resolution: 1 | Range: -2147483648 to 2147483647
0xc2	UserVar32[15]	32-bit signed integer Bytes: 4 | Units: N/A | Resolution: 1 | Range: -2147483648 to 2147483647
0xc3	OBD Start Reason	16-bit signed integer; see OBD Trip Start Reason (bit mask) table below Bytes: 4 | Units: N/A | Resolution: 1 | Range: -2147483648 to 2147483647
0xc4	OBD End Reason	32-bit signed integer; see OBD Trip End Reason (bit mask) table below Bytes: 4 | Units: N/A | Resolution: 1 | Range: -2147483648 to 2147483647
0xc5	TpsTagPayloadSize	Unsigned integer Bytes: 2 | Units: N/A | Resolution: 1 | Range: N/A
0xc6	TpsTagPayload	Payload Bytes: variable | Units: N/A | Resolution: N/A | Range: N/A
0xc7	OBD Seat Belt Status	Bytes: 1 | Units: N/A | Resolution: N/A | Range: N/A

<table>
<tr><th>ID</th><th>Name</th><th>Description</th></tr>
<tr><td>0xc8</td><td>VehicleBattery_2byte</td><td>2-byte representation of vehicle battery voltage<table><tr><th>Bytes</th><th>Units</th><th>Resolution</th><th>Range</th></tr><tr><td>2</td><td>V</td><td>0.1</td><td>0.0 to 6553.5</td></tr></table></td></tr>
<tr><td>0xc9</td><td>ExtAdc[0]_2byte</td><td>2-byte representation of external ADC[0] voltage<table><tr><th>Bytes</th><th>Units</th><th>Resolution</th><th>Range</th></tr><tr><td>2</td><td>V</td><td>0.1</td><td>0.0 to 6553.5</td></tr></table></td></tr>
<tr><td>0xca</td><td>AuxPassthruPacketSize</td><td>Size of entire aux passthru packet<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>1 to 256</td></tr></table></td></tr>
<tr><td>0xcb</td><td>AuxPassthruPacket</td><td><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></td></tr>
<tr><td>0xcc</td><td>Geofence States Bitfield 1</td><td>Bitfield representing current state of active geofences 0 to 31, where 1 is inside, and 0 is outside or inactive.<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></td></tr>
<tr><td>0xcd</td><td>Geofence States Bitfield 2</td><td>Bitfield representing current state of active geofences 32 to 49, where 1 is inside, and 0 is outside or inactive.<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></td></tr>
<tr><td>0xce</td><td>Script version</td><td>Version number of currently loaded script<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></td></tr>
<tr><td>0xcf</td><td>Paramset version</td><td>Version number of currently loaded paramset<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></td></tr>
</table>

ID	Name	Description
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
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 **Value** 0 GSM 1 GSM Compact 2 UTRAN 3 GSM w/EGPRS 4 UTRAN w/HSDPA 5 UTRAN w/HSUPA 6 UTRAN w/HSDPA and HSUPA 7 E-UTRAN 255 Invalid
0xd5	Derived Engine Seconds	Derived engine seconds **Bytes** | **Units** | **Resolution** | **Range** 4 | seconds | 1 | 0 to 4294967295

<table>
<tr><th>ID</th><th>Name</th><th>Description</th></tr>
<tr><td>0xd6</td><td>ECU Engine Seconds</td><td>ECU engine seconds
<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 4294967295</td></tr></table></td></tr>
<tr><td>0xd7</td><td>InputActiveStates</td><td>Same as the Input States bitfield, only the reported states are determined by the IDP config settings (see Input State Bits 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>0x0 to 0xff</td></tr></table></td></tr>
<tr><td>0xd8</td><td>CellularEnvironment</td><td>Each byte represents a different value
<table><tr><th>MSB</th><th></th><th></th><th>LSB</th></tr><tr><td>[CSQ]</td><td>[RSRP]</td><td>[RSRQ]</td><td>[CINR]</td></tr></table>CINR is not currently implemented (always 0xFF).
RSRP & RSRQ are always 0xFF for modules that do not support AT+CESQ.
<table><tr><th>Bytes</th><th>Units</th><th>Resolution</th><th>Range</th></tr><tr><td>4</td><td>Varies</td><td>1</td><td>0 to 0xff (within each byte)</td></tr></table></td></tr>
<tr><td>0xd9</td><td>OBDHarshAccelValue</td><td>Unsigned integer decimal value of acceleration in km/second
<table><tr><th>Bytes</th><th>Units</th><th>Resolution</th><th>Range</th></tr><tr><td>2</td><td>km/s</td><td>0.1</td><td>0.0 to 6553.5</td></tr></table></td></tr>
<tr><td>0xda</td><td>OBDHarshBrakeValue</td><td>Unsigned integer decimal value of deceleration in km/second
<table><tr><th>Bytes</th><th>Units</th><th>Resolution</th><th>Range</th></tr><tr><td>2</td><td>km/s</td><td>0.1</td><td>0.0 to 6553.5</td></tr></table></td></tr>
<tr><td>0xdb</td><td>Cell Mobile Country Code</td><td>Cell module country-dependent code
<table><tr><th>Bytes</th><th>Units</th><th>Resolution</th><th>Range</th></tr><tr><td>2</td><td></td><td>1</td><td>0.0 to 65535</td></tr></table></td></tr>
<tr><td>0xdc</td><td>Cell model/version strings</td><td>Cell model string concatenated with cell module fw version string</td></tr>
</table>

<table>
<tr><th>ID</th><th>Name</th><th>Description</th></tr>
<tr><td></td><td></td><td><table><tr><th>Bytes</th><th>Units</th><th>Resolution</th><th>Range</th></tr><tr><td>58</td><td>ascii</td><td></td><td></td></tr></table></td></tr>
<tr><td>0xdd</td><td>GPS Speed (high-resolution)</td><td>Unsigned integer. Greater range equals higher accuracy.<table><tr><th>Bytes</th><th>Units</th><th>Resolution</th><th>Range</th></tr><tr><td>2</td><td>kph</td><td>0.01</td><td>0.00 to 655.35</td></tr></table></td></tr>
<tr><td>0xde</td><td>OBD MPG 5 second average</td><td>Averaging synchronized with OBD derived lifetime odometer timing.<table><tr><th>Bytes</th><th>Units</th><th>Resolution</th><th>Range</th></tr><tr><td>2</td><td>mpg</td><td>0.1</td><td>0.0 to 256.0</td></tr></table></td></tr>
<tr><td>0xdf</td><td>GPS Harsh Accel Violations Cnt</td><td><table><tr><th>Bytes</th><th>Units</th><th>Resolution</th><th>Range</th></tr><tr><td>4</td><td></td><td>1</td><td>0 to 4294967295</td></tr></table></td></tr>
<tr><td>0xe0</td><td>GPS Harsh Accel Value</td><td><table><tr><th>Bytes</th><th>Units</th><th>Resolution</th><th>Range</th></tr><tr><td>2</td><td>kph/s</td><td>0.01</td><td>0.00 to 655.35</td></tr></table></td></tr>
<tr><td>0xe1</td><td>GPS Harsh Decel Violations Cnt</td><td><table><tr><th>Bytes</th><th>Units</th><th>Resolution</th><th>Range</th></tr><tr><td>4</td><td></td><td>1</td><td>0 to 4294967295</td></tr></table></td></tr>
<tr><td>0xe2</td><td>GPS Harsh Decel Value</td><td><table><tr><th>Bytes</th><th>Units</th><th>Resolution</th><th>Range</th></tr><tr><td>2</td><td>kph/s</td><td>0.01</td><td>0.00 to 655.35</td></tr></table></td></tr>
<tr><td>0xe3</td><td>OBD to Fuel Used HR</td><td>High resolution total fuel used.<table><tr><th>Bytes</th><th>Units</th><th>Resolution</th><th>Range</th></tr><tr><td>4</td><td>liters/bit</td><td>0.5</td><td>0 to 4294967295</td></tr></table></td></tr>
<tr><td>0xe8</td><td>ObdDieselExhaustVolume</td><td>Ratio of volume of diesel exhaust fluid to the total volume of diesel exhaust fluid storage container for aftertreatment system 1 (exhaust bank 1). 0% is empty and 100% is full.<table><tr><th>Bytes</th><th>Units</th><th>Resolution</th><th>Range</th></tr><tr><td>1</td><td>%</td><td>0.4</td><td>0 to 100</td></tr></table></td></tr>
</table>

<table>
<tr><th>ID</th><th>Name</th><th>Description</th></tr>
<tr><td>0xe9</td><td>ObdDieselExhaustLow</td><td>Least Significant Bit is not used. See "Diesel Exhaust Fluid Tank Low Level States" 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></td></tr>
<tr><td>0xea</td><td>Upper bytes of DriverId2</td><td>Unsigned integer represent of the upper two bytes of unique six byte iButton ID.<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>0 to 65535</td></tr></table></td></tr>
<tr><td>0xeb</td><td>Upper bytes of DriverId2</td><td>Unigned integer represent of the upper two bytes of unique six byte iButton ID.<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>0 to 65535</td></tr></table></td></tr>
</table>

8.4. DTC Packet Parsing

DTC data is reported via two PCR fields: ObdDtcPacketSize (0x6b) and ObdDtcPacket (0x6c). DTC data can be included in any of the messages sent to the backend and it must include both fields.

ObdDtcPacketSize is a two byte field containing the size of the DTC data packet. The size is the total number of bytes of DTC data in ObdDtcPacket.

ObdDtcPacket is a variable size packet containing the DTC data.

The DTC data format is as follows:

```
<mil_status><num_ecu_dtc><ecu_id_0><num_dtc_0><ascii_0><dtc_code_0><fmi_0>...<ascii_n><dtc_code_n><fmi_n>... <ecu_id_m><num_dtc_0><ascii_0><dtc_code_0><fmi_0>...<ascii_n><dtc_code_n><fmi_n>
```

Field	Size	Description
<mil_status>	1	MIL/Lamp status data
<num_ecu_dtc>	1	Number of ECUs reporting DTCs
<ecu_id_x>	1	ECU ID
<num_dtc_x>	1	Number of DTCs reported by ECU
<ascii_x>	1	DTC ASCII code. One character code indicating the type of system that experienced the failure (e.g. 'P', 'U', etc.)
<dtc_code_x>	4	DTC code
<fmi_0>	1	Failure mode information

8.4.1. OBDII

Field	Description
<ascii_x>	‘P', ‘C’, ‘B’, 'U’ or ' '
<dtc_code_x>	DTC
<fmi_0>	0

8.4.2. J1939

Field	Description
<ascii_x>	'0'
<dtc_code_x>	DTC
<fmi_0>	FMI reported by ECU

8.4.3. J1708

Field	Description
<ascii_x>	'S' or '0'
<dtc_code_x>	DTC
<fmi_0>	FMI reported by ECU

8.4.4. Examples

Vehicle rporting no DTCs, MIL off:

```
ObdDtcPacketSize: 0002

ObdDtcPacket: 0000
    <mil_status> - MIL off
    <num_ecu_dtc> - 0 ECUs reporting DTCs
```

Vehicle reporting two DTCs on ECU 7E8 and one DTC on ECU 7E9, MIL on:

```
ObdDtcPacketSize: 0018 (24 bytes)

ObdDtcPacket: 8002E80250000001010050000010200E901550000100000
    <mil_status> - 80 (MIL on)
    <num_ecu_dtc> - 2 (2 ECUs reporting DTCs)
    <ecu_id_0> - E8 (ECU 7E8)
    <num_dtc_0> - 2 (2 DTCs)
        <ascii_0> - 50 ('P')
        <dtc_code_0> - 00000101 (code P0101)
        <fmi_0> - 00
        <ascii_1> - 50 ('P')
        <dtc_code_1> - 00000102 (code P0102)
        <fmi_1> - 00
    <ecu_id_1> - E9 (ECU 7E9)
    <num_dtc_1> - 1 (1 DTCs)
        <ascii_0> - 55 ('U')
        <dtc_code_0> - 00001000 (code U1000)
        <fmi_0> - 00
```

Heavy-duty Vehicle reporting one DTC (J1939), Amber Warning Lamp on

```
ObdDtcPacketSize: 000A (10 bytes)

ObdDtcPacket: 2001330130000003A10C
    <mil_status> - 20 (Amber Warning Lamp on)
    <num_ecu_dtc> - 1 (1 ECU reporting DTC)
    <ecu_id_0> - 33 (ECU 33)
    <num_dtc_0> - 1 (1 DTC)
        <ascii_0> - 30 ('0')
        <dtc_code_0> - 000003A1 (SPN 929 - Tire location)
        <fmi_0> - 0C (Bad Intelligent Device Or Component)
```

8.4.5. Input/Output State Bits

Input

Bit	Input State
0x01	Ignition
0x02	Input 1
0x04	Input 2
0x08	Input 3
0x10	Input 4
0x20	Device In Motion (Accel)
0x40	Main supply voltage available
0x80	Accelerometer Reorientation Valid

output

Bit	Output State
0x01	Output 0
0x02	Output 1
0x04	Output 2
0x08	Onboard Buzzer
0x10	Garmin Power

8.4.6. OBD Run States Bits

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

8.4.7. 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

8.4.8. 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

8.4.9. 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

8.4.10. 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

8.4.11. 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

8.4.12. 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

8.4.13. Linkage Version

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

8.4.14. 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).

8.4.15. 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) and16 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.

8.4.16. 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>
```

8.4.17. 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.

8.4.18. 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,andNumSats	`: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"`

8.4.19. 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

8.4.20. 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

9. Command Language

9.1. 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.

9.2. 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]"

9.3. 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

9.4. 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'

9.4.1. 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

9.4.2. 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

9.4.3. Object Codes

Object	Code
Bootloader	bl
Chipflash	chip
Config	cfg
Command	cmd
Debug	dbg
Debug Mask	dm
DM Msg	dmmsg
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

9.5. Device Commands

9.5.1. ADC

Command	Function
:raval	Read ADC values

9.5.2. Bluetooth®

Command	Function
:qbi	Query Bluetooth information
:ubcmd sleep	
:ubcmd wake	
:ubcmd rest	
:ubcmd ble_start	RF calibration procedure start
:ubcmd ble_end	RF calibration procedure end
:ubcmd <mod> <pat> <freq>	Puts the device into single frequency test mode

<table>
<tr><th>Command</th><th>Function</th></tr>
<tr><td><power></td><td>
<table>
<tr><th><mod> Val</th><th>Definition</th></tr>
<tr><td>0x00</td><td>CW (Pattern must be 0x03 or 0x04)</td></tr>
<tr><td>0x01</td><td>GFSK (BR)</td></tr>
<tr><td>0x02</td><td>(pi)/4-DQPSK (2-EDR)</td></tr>
<tr><td>0x03</td><td>8DPSK (3-EDR)</td></tr>
<tr><td>0x04</td><td>BLE</td></tr>
</table>
<table>
<tr><th><pat> Val</th><th>Definition</th></tr>
<tr><td>0x00</td><td>PN9</td></tr>
<tr><td>0x01</td><td>PN15</td></tr>
<tr><td>0x02</td><td>0x5555 (0101 0101 0101 0101)</td></tr>
<tr><td>0x03</td><td>All 1's</td></tr>
<tr><td>0x04</td><td>All 0's</td></tr>
<tr><td>0x05</td><td>0xF0F0 (1111 0000 1111 0000)</td></tr>
<tr><td>0x06</td><td>0xFF00 (1111 1111 0000 0000)</td></tr>
</table>
<table>
<tr><th><freq> Val</th><th>Definition</th></tr>
<tr><td>0-39</td><td>Freq = 2402 + 2*val</td></tr>
<tr><td>40-78</td><td>Freq = 2403 + 2*(val - 40)</td></tr>
</table>
<table>
<tr><th><power> Val</th><th>Definition</th></tr>
<tr><td>0x01</td><td>BLE Output Power</td></tr>
<tr><td>0x08</td><td>Min Output Power (non-BLE)</td></tr>
<tr><td>0x0F</td><td>Max Output Power (BLE)</td></tr>
</table>
</td></tr>
<tr><td>:ubcmd adv</td><td></td></tr>
<tr><td>:ubcmd unadv</td><td></td></tr>
<tr><td>:ubcmd authbypass</td><td>NOTE: Command cannot be sent via SMS or UDP</td></tr>
</table>

9.5.3. Accelerometer

Command	Function
:rxval	Read current X, Y, and Z accelerations
:rxval coefs	Read current filter coefficients
:rxval rot	Read current reorientation angles (radians)

Command	Function
:wxval coefs <num_stages> <coef1> <coef2> ... <coefX>	Write the number of filter stages and all filter coefficients
:wxval coef <coef_index> <coef>	Write filter coefficient at index X
: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

9.5.4. 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

9.5.5. 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)

9.5.6. Command Language

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

9.5.7. 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 **NOTE:** Command cannot be sent via SMS or UDP
:gjscr x <serial_port_id> <flag -optional>	Upload VIS file via xmodem (flag = 1; erases all VIS's first) **NOTE:** Command cannot be sent via SMS or UDP
: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)

9.5.8. Device

Command	Function
:rdval reset	Read device reset reason Reason codes:

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.

<table>
<tr><th>Command</th><th>Function</th></tr>
<tr><td>:rdval serial</td><td>Read device serial number</td></tr>
<tr><td>:rdval uptime</td><td>Read device uptime (seconds since reset)</td></tr>
<tr><td>:rdval wake</td><td>Read device wake reason (only accurate if device has slept)</td></tr>
<tr><td>:wdval serial "<serial_no>"</td><td>Write the device serial number</td></tr>
<tr><td>:qdi</td><td>Query device info ("vtu query")</td></tr>
<tr><td>:qdi cell</td><td>Query cellular device info ("vtu query modem")</td></tr>
<tr><td>:qdi gps</td><td>Query GPS device info ("vtu query gps")</td></tr>
<tr><td>:udcmd checkin</td><td>Send a Device Manager Check-In message to dst[9]</td></tr>
<tr><td>:udcmd factory</td><td>Reset device to factory state</td></tr>
<tr><td>:udcmd factory keys</td><td>Reset device to factory state plus revert AES keys to defaults. This is the only command available outside the AES console login.

NOTE: Cannot be sent via SMS or UDP</td></tr>
<tr><td>:udcmd dslp <mask> <minutes></td><td>Enter deep sleep with <mask> wake setting

Mask Values:

<table><tr><th>Hex</th><th>Description</th></tr><tr><td>0x0000</td><td>Ring (Not applicable)</td></tr><tr><td>0x0001</td><td>Wired Ignition/IN0</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>External Battery (configured by wbv)</td></tr><tr><td>0x0100</td><td>Unplug</td></tr></table></td></tr>
<tr><td>:udcmd slp <mask> <minutes></td><td>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>Wired Ignition/IN0</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>External Battery (configured by wbv)</td></tr><tr><td>0x0100</td><td>Unplug</td></tr></table></td></tr>
</table>

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

9.5.9. 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 layout NOTE: 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

9.5.10. 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)

9.5.11. 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

9.5.12. 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)

9.5.13. 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

9.5.14. GPS Simulator

Command	Function
:sgsim config <lat> <long> <heading> <speed>	GPS simulator initialize/enable:

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: **Value** / **Description** 0 None 1 Unlocked 2 Locked 3 Sleep 4 LockedGood
:sgsim hdop <hdop>	GPS simulator force hdop
:sgsim clear	Disable GPS simulator

9.5.15. Cellular

Command	Function
:rtdm	Read transport debug mask
:wtdm <mask>	Write the transport debug mask mask Values: **Hex** / **Description** 0x000 GSMDebug_None 0x001 GSMDebug_State 0x002 GSMDebug_Command 0x004 GSMDebug_Response 0x008 GSMDebug_Parse 0x010 GSMDebug_ResponseHex 0x020 GSMDebug_Mux 0x040 GSMDebug_Backend 0x080 GSMDebug_Mno 0x100 GSMDebug_URC 0x200 GSMDebug_PDU

<table>
<tr><th>Command</th><th>Function</th></tr>
<tr><td>:wtovrd apn ["<apn_ overrride>"]</td><td><apn_overrride>"] Override class 3 APN Only applicable to:
XT6372, XT6374
XT6369
XT6383</td></tr>
<tr><td>:etm</td><td>Enable transport debug channel monitoring</td></tr>
<tr><td>:dtm</td><td>Disable transport debug channel monitoring</td></tr>
<tr><td>:ett</td><td>Enable transport test mode (‘ct’ mode)</td></tr>
<tr><td>:ett direct</td><td>Connect serial console directly to cellular port</td></tr>
<tr><td>:dtt</td><td>Disable transport test mode (‘ct’)</td></tr>
<tr><td>:qti</td><td>Query transport info</td></tr>
<tr><td>:qti ip</td><td>Query cellular mobile IP</td></tr>
<tr><td>:qti certmask</td><td>Query cellular certification mask

(0x0: no security certificates loaded; 0x7: all 3 security certificates loaded successfully)</td></tr>
<tr><td>:utcmd factory</td><td>Set cellular module to factory default state</td></tr>
<tr><td>:utcmd reset</td><td>Queue a reset for the cellular module</td></tr>
<tr><td>:utcmd sleep</td><td>Stop cellular module</td></tr>
<tr><td>:utcmd wake</td><td>Start cellular module</td></tr>
<tr><td>:utota ftp "<filename with extension>" <type></td><td>Request FTP file transfer (only XT6369, XT6379, XT6376)<table><tr><th>Type</th><th>Description</th></tr><tr><td>12</td><td>Cell Module Firmware</td></tr></table></td></tr>
<tr><td>:utota ftp <ftp url> <content_ type></td><td>Request FTP file transfer (XT2494 only).

The <ftp url> must start with <code>ftp://</code> and include the username, pw, and host. The total character must be less than 238 characters total.

EXAMPLE: (e.g. ftp://user:pass@12.345.67.890/filename.bin)<table><tr><th>Type</th><th>Description</th></tr><tr><td>12</td><td>Cell Module Firmware</td></tr></table></td></tr>
<tr><td>:utota http <url> <content_ type></td><td>Request HTTP or HTTPS file transfer (XT2494 only). The <url> must start with <code>http://</code> or <code>https://</code> and be less than 236 characters long.<table><tr><th>Type</th><th>Description</th></tr><tr><td>12</td><td>Cell Module Firmware</td></tr></table></td></tr>
</table>

<table>
<tr><th>Command</th><th>Function</th></tr>
<tr><td>:utota https "<host>" "<path>" <port> <type>[[[<index>] "<filename>"] <rev>]</td><td>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>
<index> only applies to overlays

"<filename>" is arbitrary and optional name reported in :q1i response

<rev> is the user determined revision value</td></tr>
<tr><td>:utota s3 "<full S3 presigned link>" <port> <type>[[[<index>] "<filename>"] <rev>]</td><td>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><tr><td>12</td><td>Cell Module Firmware (KB1.3 or later)</td></tr></table>
<index> only applies to overlays

"<filename>" is arbitrary and optional name reported in :q1i response</td></tr>
</table>

Command	Function
	<rev> is the user determined revision value
:rtval mno	See Hardware ID and MNO select
:wtval mno <profile>	

9.5.16. 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 **NOTE:** Command cannot be sent via SMS or UDP
:rrscr	Read interpreter script
:rrval odom <index>	Read an odometer Index: 0 GPS Trip 1; 1 GPS Trip 2; 2 OBD Trip; 3 GPS Life; 4 OBD Life
: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: Index: 0 GPS Trip 1; 1 GPS Trip 2; 2 OBD Trip; 3 GPS Life; 4 OBD Life

Command	Function
:crval tripodom <index>	Clear a trip odometer: Index: 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)
:rrval sysval	Read system value (in decimal if signed value; in hex if unsigned value) NOTE: Only values 4 bytes or less are functional; greater than 4 bytes returns a 0 value

9.5.17. 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 **NOTE:** Command cannot be sent via SMS or UDP
:g1fw x <serial_port_id>	Upgrade main firmware via xmodem **NOTE:** Command cannot be sent via SMS or UDP
:q1fw/:q1i	Query main firmware
:u1cmd reset	Reset main processor

9.5.18. 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

9.5.19. 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

NOTE: For each printed value:

- The first number is the raw value from the ECU (units vary by protocol)
- The second number is the converted value our device calulated
- The third number is the "lowest ECU" (address) that reported the value

vss	Trip current speed 1 KPH/bit
rpm	0.125 RPM/bit
ect	• OBDII and J1939: -40 + 1°C x bits • J1708: 1°F/bit
fl	• OBDII: (100/225%)/bit • J1708 and J1939: 0.1%/bit
crztime	1 second/bit

Command	Function
	<table><tr><th></th><th></th></tr><tr><td>vehtime</td><td>1 second/bit</td></tr><tr><td>enghr</td><td>1 hour/bit</td></tr><tr><td>engsec</td><td>1 second/bit</td></tr><tr><td>idle</td><td>• J1939: 0.05L/bit • J1708: 0.125Gal/bit (0.473L/bit)</td></tr><tr><td>pto</td><td>0.05L/bit</td></tr><tr><td>used</td><td>• J1939: 0.05L/bit • J1708: 0.125Gal/bit (0.473L/bit)</td></tr><tr><td>used_hr</td><td>• J1939: 0.05L/bit • J1708: 0.125Gal/bit (0.473L/bit)</td></tr><tr><td>true_odo</td><td>• J1939: Supporting pid 917: 5 meters/bit Not supporting pid 917: 0.125km/bit • J1708: Not supporting pid 917: 0.1 mile/bit (0.161km/bit)</td></tr><tr><td>idletime</td><td>1 second/bit</td></tr><tr><td>ptotime</td><td>1 second/bit</td></tr></table>
:qoi ss2	Query OBD snapshot additional parameter information **NOTE:** For each printed value: • The first number is the raw value from the ECU (units vary by protocol) • The second number is the converted value our device calulated • The third number is the "lowest ECU" (address) that reported the value <table><tr><th></th><th></th></tr><tr><td>gear</td><td>0 = neutral 1 = forward 2 = reverse 3 = park</td></tr><tr><td>fueltemp</td><td>• J1939: -40 + 1°C x bits • J1708: 0.25°F/bit</td></tr></table>

Command	Function	
	oiltemp	• OBDII: -40 + 1°C x bits • J1939: -273 + 0.03125°C x bits • J1708: 0.25°F/bit
	throttle	• OBII: (100/255%)/bit • J1939 and J1708: 0.4%/bit
	fueleco	• J1939: 1/512km/l/bit • J1708: 1/256mpg/bit
	accelpos	• OBII: (100/255%)/bit • J1939 and J1708: 0.4%/bit
	load	• OBII: (100/255%)/bit • J1939: 1%/bit • J1708: 0.5%/bit
	torque	-125 + 1%/bit
	oillvl	• J1939: 0.4%/bit • J1708: 0.5%/bit
	oilpress	• J1939: 4 kpa/bit • J1708: 2.45 kpa/bit
	coollvl	• J1939: 0.4%/bit • J1708: 0.5%/bit
	coolpress	• J1939: 2 kpa/bit • J1708: 0.862 kpa/bit
	iat	• OBII and J1939: -40 + 1°C x bits • J1708: 0.25°F/bit
	mantemp	• OBDII and J1939: -40 + 1°C x bits • J1708: 1°F/bit
	brake	0 = parking brake off 3 = parking brake set **NOTE:** if low ECU = 0xFE, converted value is 0 if not moving 1 if moving
	seat	0 = invalid/not supported 1 = NOT buckled 2 = OK Seat belt buckled

Command	Function
:qoval	Query OBD calculated information
:qoi viol	Query OBD Violations
:qost	Query OBD state
:roval dtc	Read OBD DTC List
:woval p <protocol> <cll>	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

9.5.20. 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 **NOTE:** Command cannot be sent via SMS or UDP
: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:

Hex	Description
0x01	Hwid_2400_A
0x02	Hwid_6370_A
0x03	Hwid_6374_A
0xff	Hwid_6300_B

Command	Function
:wyval hwid <value>	Set the hardware ID value
:wyval key <key_index> <okok> <nkok> <oknk>	Change AES key, where key_index: Index: 0 – Update; 1 – Customer; 2 – Transport
:wyval ptk <key_index> <plaintext_key>	Change AES key, where key_index: Index: 0 – Update; 1 – Customer; 2 – Transport The desired key value is not encrypted. This command is only available via physical connection to the device.

9.5.21. 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 <bock_id>	Read TPS block
:qccmd sync	Sync TPS file with AAb3 target servers
:gcfw x <serial_port_id> "<name>"	Upgrade TPS file via xmodem **NOTE:** Command cannot be sent via SMS or UDP

9.5.22. Aux Passthru

<table>
<tr><th>Command</th><th>Function</th></tr>
<tr><td>:qvi</td><td>Print info</td></tr>
<tr><td>:evm</td><td>Enable aux passthru debug channel monitoring</td></tr>
<tr><td>:dvm</td><td>Disable aux passthru debug channel monitoring</td></tr>
<tr><td>:wvdm <mask_base10></td><td>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></td></tr>
<tr><td>:rvdm</td><td>Read (print out) aux passthru debug mask</td></tr>
<tr><td>:xvmsg <bytes></td><td>Send a packet to be transmitted out over the aux port

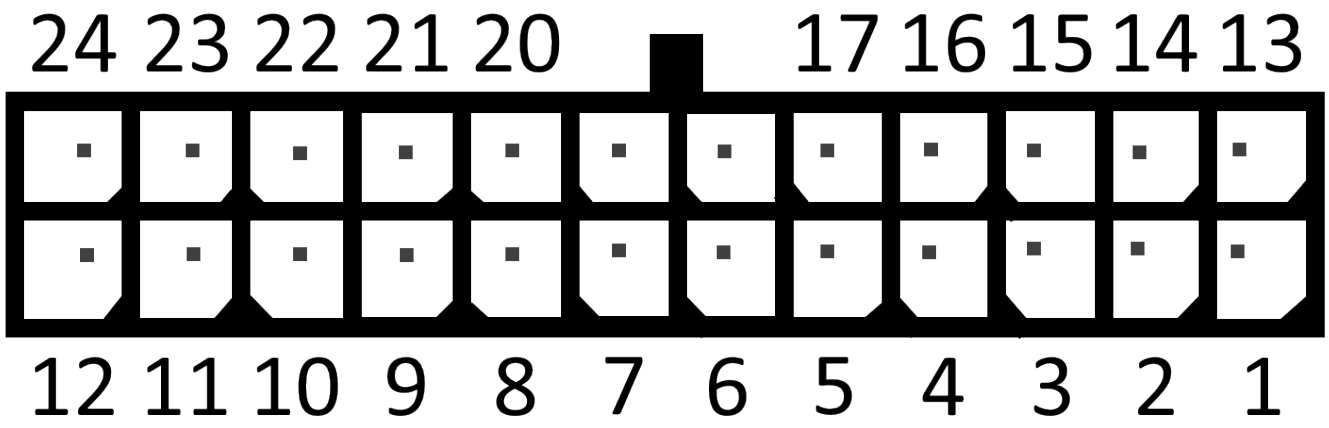
If the <bytes> field begins with "0x", the message is sent in hex;

if the <bytes> field does not begin with "0x", the message is sent in ASCII</td></tr>
</table>

10. Electrical Characteristics

The XT6300 has two electrical connectors on the body of the device which contain all necessary power and data communication lines. One of these connectors is a Molex 24-pin female header (43045-2421) and the other is a Molex 14-pin female header (43045-1421).

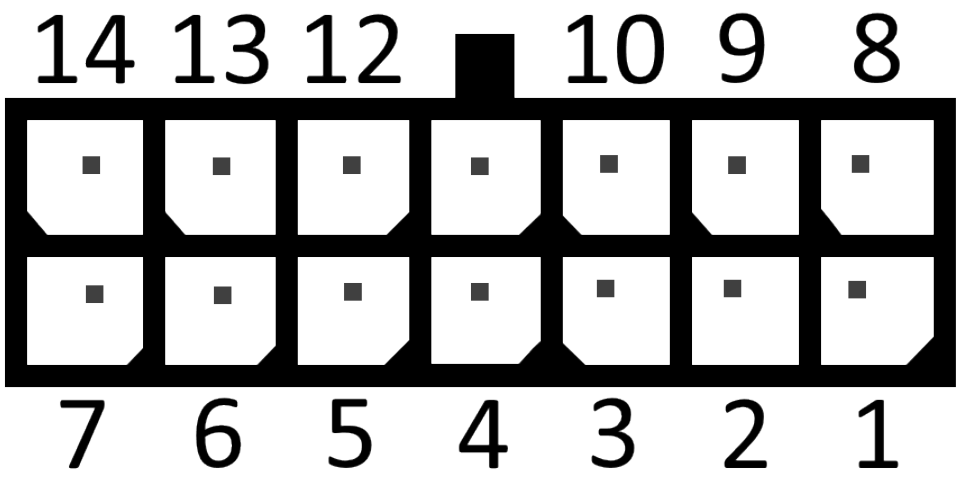
10.1. 24-Pin Connector



Pin	Description
1	Vin 12/24V
2	LED V+
3	GND
4	Ignition Out
5	IN 0
6	IN 1
7	IN 2
8	IN 3
9	IN 4
10	OUT 0
11	OUT 1
12	OUT 2
13	SW controlled Vin output
14	RS232 interface 2 RX
15	RS232 interface 2 TX
16	GND
17	RS232 interface 1 RX
18	RS232 interface 1 TX
19	GND
20	1-Wire interface 1

Pin	Description
21	1-Wire interface 2
22	ADC IN 1
23	ADC IN 2 (THIS IS SHUNTED TO GROUND ON ALL KNOWN XT6300 HARDWARE)
24	GND

10.2. 14-Pin Connector



Pin	Description
1	CAN (J-2234) High
2	J-1708 RX
3	J-1708 TX
4	ISO 9141-2 Low
5	J-1850 Bus -
6	J-1850 Bus +
7	N/C
8	CAN (J-2234) Low
9	Vendor Option Pin 3
10	Vendor Option Pin 11
11	Vendor Option Pin 1
12	ISP 9141-2 K-Line
13	GND
14	Battery +

11. Appendix

11.1. UDPC

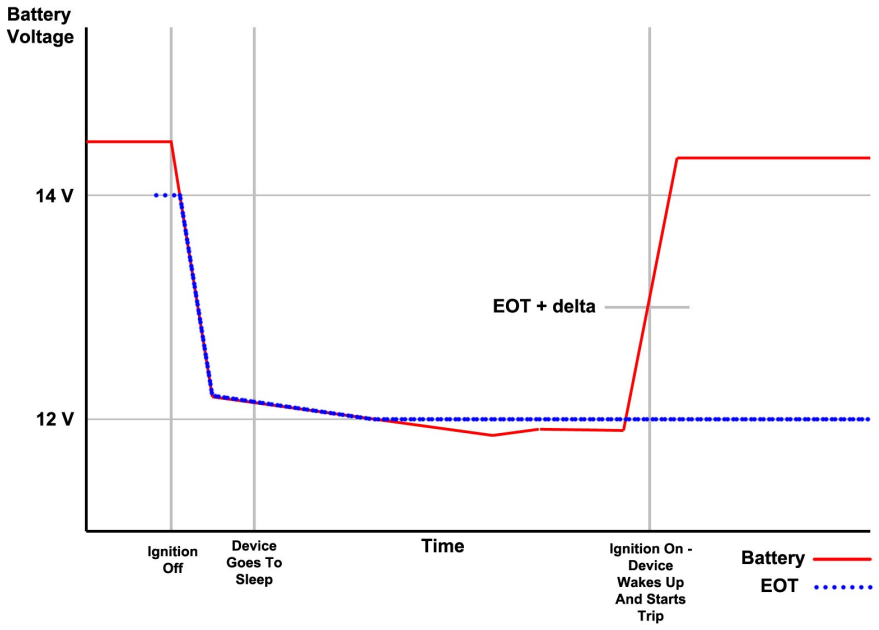
1. Open a standard *udp server* to the configuration of dst[<dst_index>].
2. Issue the command :ukcmd tunnel cmd <dst_index> via console to the device.

> **NOTE:** This will send out a hello message to dst[<dst_index>] and open a temporary socket for an “ack” response to come back on a non-blocked port.

3. Reply with the intended udpc message.

11.2. Dynamic Alternator

This feature is only available on ignition type OBD.



11.2.1. Overview

Due to differences in batteries, battery age and charging systems, setting the alternator on/off thresholds to arbitrary values does not always work to capture the start of a trip. The dynamic alternator feature was developed to try to capture the missing trips by monitoring the battery voltage for jumps instead of just crossing a threshold.

When the dynamic alternator is enabled, the device overrides the alternator on/off settings (ast[0]/ast[1]) with dynamically calculated values as the battery voltage changes. When a voltage change larger than a user

configured voltage delta (ast[3]) is detected, the device attempts to start vehicle communications. If successful, a trip is started.

When vehicle communications stop, the trip is ended and the battery voltage is captured (End of Trip Voltage - EOT). If the battery voltage continues to go down, EOT is adjusted accordingly. EOT is not adjusted if the voltage goes up. EOT is used to calculate the dynamic alternator on/off thresholds as follows:

```
Alternator on threshold = EOT + delta
```

```
Alternator off threshold = EOT + ½ delta
```

The battery voltage continues to be monitored and EOT adjusted as long as the alternator state is off, including during sleep. The device will wake up if the alternator on threshold is crossed and vehicle communications will be attempted. The dynamic alternator overrides the Wake Battery Voltage (wbv) setting.

11.2.2. Configuring

The dynamic alternator feature is enabled by setting a non-zero value in ast[3]. For ast[3] details Alternator State Threshold (ast) on p. 40. Ast[3] = 0 disables dynamic alternator.

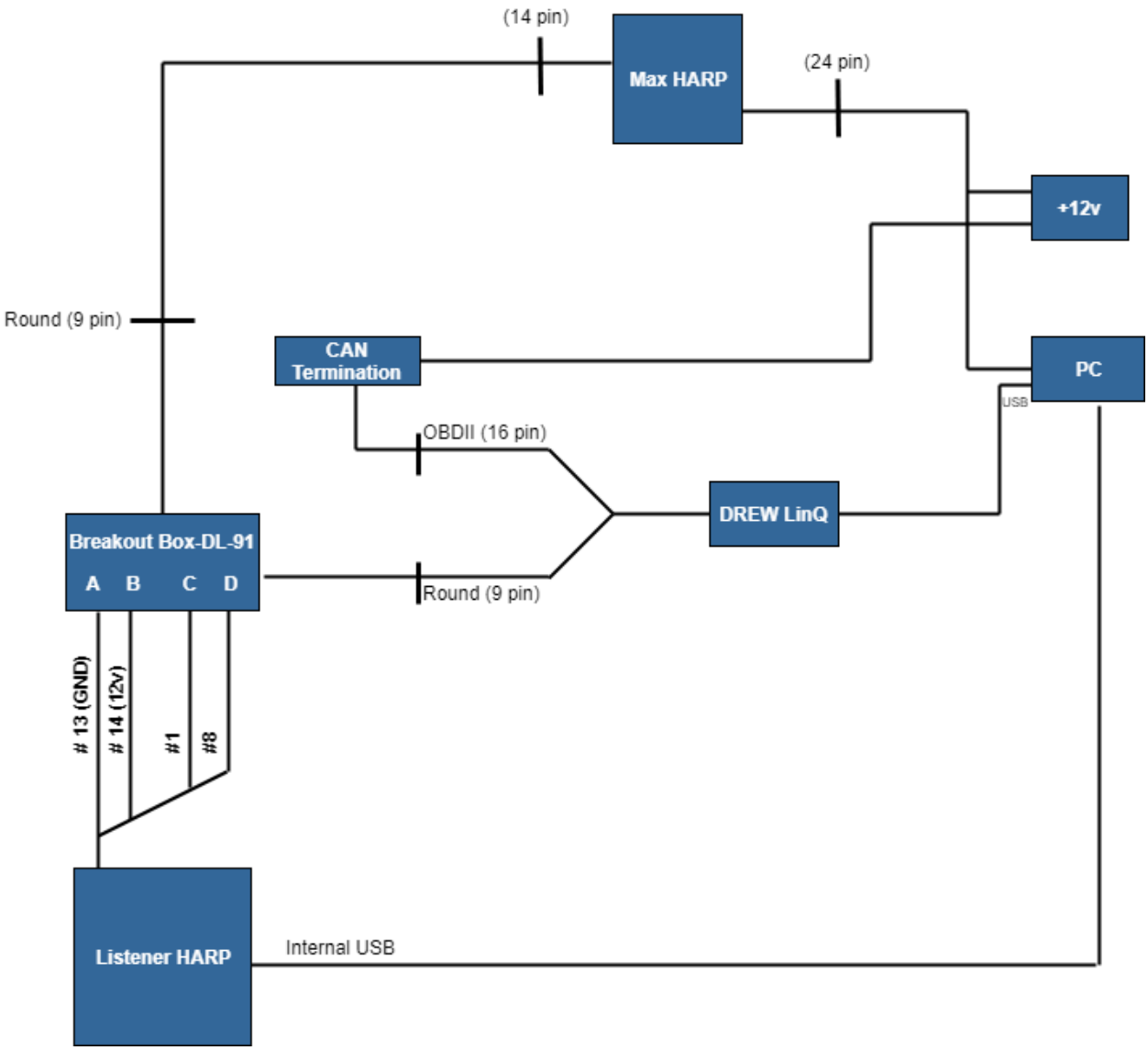
CAUTION: Setting ast[3] too low will cause the device to have too many failed communication attempts, which could affect sleep current consumption. Setting it too high could cause missed trips. Recommended settings of 0.3 - 0.5V deltas seem to work for most vehicles.

EOT is captured at end the of active trip, at end of failed communication attempts, and on device re-connect events. EOT voltage range is 12-14V.

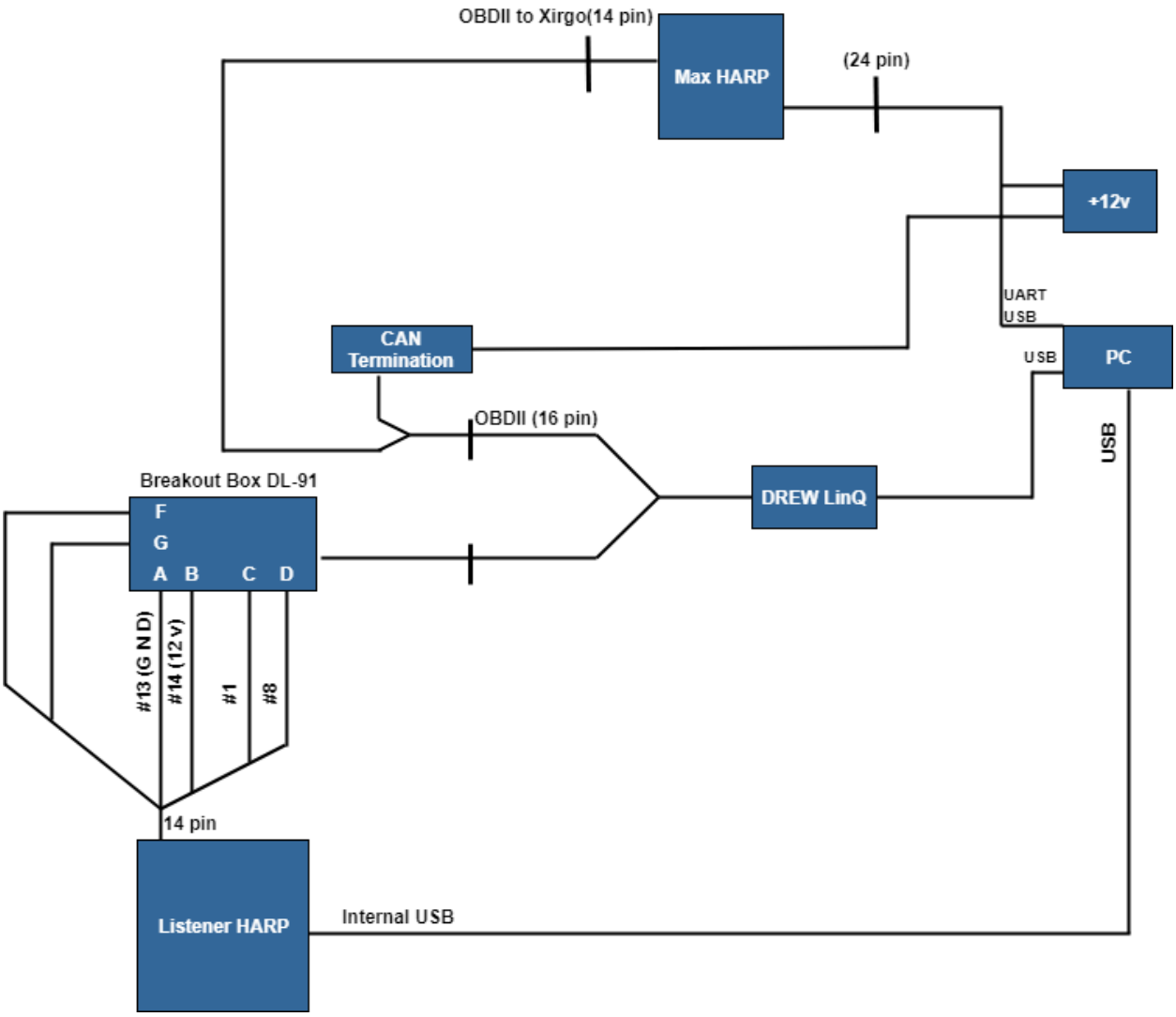
While we are able to capture a larger percentage of trips using dynamic alternator, it does not work well on some hybrids, vehicles with dual batteries, or smart charging systems.

For a more complete solution, dynamic alternator is used in conjunction with GPS motion detection and by listening for bus traffic under certain voltage conditions.

11.3. J1939 AND J1708 TEST SETUP BLOCK DIAGRAM



11.4. J1939 Dual Can Test Setup Block Diagram



12. Regulatory Statements

12.1. FCC

This equipment with FCC-ID: and IC-ID: , Model: XT6300 is subject to the Federal Communications Commission (FCC) and Industry Canada (IC) rules.

NOTICE:

Changes or modifications not expressly approved by the party responsible for compliance could void the user's authority to operate the equipment.

This device complies with Part 15 of the FCC Rules.

Operation is subject to the following two conditions:

1. this device may not cause harmful interference, and
2. this device must accept any interference received, including interference that may cause undesired operation.

Changes or modifications made to this equipment not expressly approved by Xirgo Technologies, LLC may void the FCC authorization to operate this equipment.

NOTE: This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates, uses and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures:

- Reorient or relocate the receiving antenna.
- Increase the separation between the equipment and receiver.
- Connect the equipment into an outlet on a circuit different from that to which the receiver is connected.
- Consult the dealer or an experienced radio/TV technician for help.

Radio frequency radiation exposure Information:

This equipment, XT6300, complies with FCC and ISED radiation exposure limits set forth for an uncontrolled environment. This equipment must be installed and operated with minimum distance of 20cm between the XT6300 and your body.

12.2. IC

Antenna Statement

Under Industry Canada regulations, this radio transmitter may only operate using an antenna of a type and maximum (or lesser) gain approved for the transmitter by Industry Canada. To reduce potential radio interference to other users, the antenna type and its gain should be so chosen that the equivalent isotropically radiated power (e.i.r.p.) is not more than that necessary for successful communication.

Conformément à la réglementation d'Industrie Canada, le présent émetteur radio peut fonctionner avec une antenne d'un type et d'un gain maximal (ou inférieur) approuvé pour l'émetteur par Industrie Canada. Dans le but de réduire les risques de brouillage radioélectrique à l'intention des autres utilisateurs, il faut choisir le type d'antenne et son gain de sorte que la puissance isotrope rayonnée équivalente (p.i.r.e.) ne dépasse pas l'intensité nécessaire à l'établissement d'une communication satisfaisante.

Licence Exempt

This device complies with Industry Canada licence-exempt RSS standard(s). Operation is subject to the following two conditions:

1. this device may not cause interference, and
2. this device must accept any interference, including interference that may cause undesired operation of the device.

Le présent appareil est conforme aux CNR d'Industrie Canada applicables aux appareils radio exempts de licence. L'exploitation est autorisée aux deux conditions suivantes:

1. l'appareil ne doit pas produire de brouillage, et
2. l'appareil doit accepter tout brouillage radioélectrique subi, même si le brouillage est susceptible d'en compromettre le fonctionnement.

12.3. 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