OBDII STREAMER FAMILY

Command and Response Document

Table of Contents

Document Revision History

Version	Date	Changes	Author
Draft	1/19/06	Initial Draft	Jeff M. Patrick
1.00	1/23/06	- Added negative response to Bootloader handshaking procedure. - Added fuel type setting to command 0x15.	Jeff M. Patrick
1.01	2/6/06	- Updated startup procedure to allow responses to the following requests during startup: o READ_MODEL_NUMBER o FIRMWARE_VERSION o READ_SERIAL o UPGRADE_FIRMWARE - Modified the ENABLE_PARAMETERS command to allow enabling or disabling individual parameters. o Added 3 examples to this command. - Added vehicle status update information to the overview section. - Split command 0x30 into commands 0x30, 0x31, 0x32, & 0x33. Also moved command 0x31 to 0x35. - Added Brake Switch Status, Cruise Control Speed, Cruise Control Status, and Turn Signal Status to the supported parameter list. - Added GMLAN Low Speed bus to the supported protocol list in command GET_VEHICLE_INFO. - Updated CHANGE_PARAMETER_THRESHOLD command.	Jeff M. Patrick
1.02	2/21/06	- Changed GET_VEHICLE_INFO command to only allow confirmed DTCs to be retrieved.	Jeff M. Patrick
1.03	3/20/06	- Shifted Parameter ID's by -1 so the first parameter is 0 instead of 1. - Modified the Odometer parameter to be 4 bytes instead of 2. - Removed Fuel Type bit from CMD 0x15	Jeff M. Patrick
1.04	6/7/06	- Fixed Documentation Errors	Jeff M. Patrick
1.05	7/6/06	- Added Store Record(0x60) Command - Added Parameter ranges to any still marked "TBD" - Updated CMDs 0x21 & 0x35 to include 0xFF select all options. - Added CMDs 0x46 & 0x47 to control bypass modes. Added bypass mode description.	Jeff M. Patrick

		- Added CMD 0x16 to dump the startup logfile.	
1.06	7/7/06	- Updated Bootload Command	Jeff M. Patrick
1.07	7/31/06	- Added ignition off status message - Changed DELAY_COMMUNICATION to VEHICLE_NOT_DETECTED - Added naming convention for firmware files	Sean Grady Kevin Nelson
1.08	8/18/06	- Added description to firmware update procedure	Kevin Nelson
1.09	8/23/06	- Added note to command 0x35 about response to timed update request with the ignition off.	Larry Reeves
1.10	9/7/06	- Corrected Update Firmware Version examples - Remove Airbag Deployment Status and Cruise Control Speed from supported parameters - Added note on fuel rate accuracy - Changed GENERAL_CONFIGURATION to SERIAL_BAUD and removed section on enabling/disabling status messages. - Removed line stating that “OBDII” and “HOST” will be in the firmware .hex files - Corrected Ignition Status example - Change Odometer scaling to be 1:1	Larry Reeves Kevin Nelson
1.11	10/25/06	- Removed Command 0x45 – Enter Bypass Mode - Fixed Ignition Status in example under 0x23 request. - Removed references to non-volatile parameters - Added ignition off mode description. - Fix UPDATE_EEPROM reply strings - Added description of how the EEPROM BLOCK_ADDRESS is incremented	Kevin Nelson Jeff M. Patrick
1.12	11/3/06	- Corrected checksum in reply of UPDATE_EEPROM example. - Added programming order requirements - Added timing requirements between update files. - Corrected 0x22 command to send to remove extra 0x01 after DL character.	Kevin Nelson
1.14	1/5/07	- Added SET_SLEEP_MODE_BEHAVIOR and WAKE_FROM_SLEEP_MODE commands - Added Calculated Trip Odometer & Trip Fuel parameters - Added Trip Reset command	Jeff M. Patrick Mohammad Siddiqui
1.15	3/26/07	- Added more description to SET_SLEEP_MODE_BEHAVIOR and WAKE_FROM_SLEEP_MODE commands - Removed references to clock switching.	Kevin Nelson

		- Removed references to the startup log.	
1.16	5/24/07	- Added section on transferring to a new vehicle - Update description of Ignition/OBDII Status Message. - Update description of Ignition Off Status Message	Kevin Nelson
1.17	6/5/07	- Added a note that the turn signal could toggle with the flashers - Added a description of update rate to ENABLE_PARAMETERS command. - Added a list of parameters stored in non-volatile memory. - Added notes on timeouts to the UPDATE_FIRMWARE and UPGRADE_EEPROM commands. - Added note in ENTER_SLEEP_MODE command that the device will not enter sleep when detecting the vehicle. - Added note in ENTER_SLEEP_MODE command that the device will not update calculated parameters when commanded into sleep mode.	Kevin Nelson

Communication Protocol Definition

All commands and responses to or from the OBDII Streamer are formatted in this style.

Byte	Byte	Byte(s)	Byte	Byte(s)	Byte
Start of Frame	Control Length	Control Bytes	Data Length	Data Bytes	Checksum

Start of Frame 0x01 – This byte never changes.

Control Length The length of the control bytes. The control length of a request or response varies by command and will be defined under each commands heading in this document.

Control Byte(s) The first control byte in a request is the command being send. In a response the control byte will be 0x80 greater than the request control byte.

Each command or response may have additional control bytes as defined by the specific command or response. Each message must have at least one control byte.

Data Length The number of data bytes in the current message. Zero data byte messages are valid.

Data Bytes This field contains the data portion of the message.

Checksum The checksum is a one byte sum of all bytes including Start of Frame, Control Length, Control Bytes, Data Length, and Data Bytes.

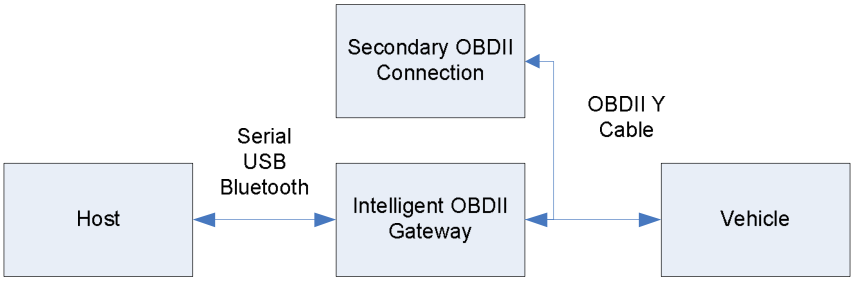
Checksum = SUM AND 0xFF

Overview

The OBDII Streamer is an OBDII to Serial gateway. The raw data, multiple PIDS per parameter, and timing issues of the OBDII bus are abstracted from the host. The OBDII Streamer allows a developer to quickly access data from the vehicle without having knowledge of the OBDII bus.

The OBDII Streamer is intended to be installed in a vehicle permanently. Coupled with B&B Electronics OBDII Y-Cable the OBDII Streamer will immediately disconnect from the OBDII bus when another scan tool is connected to the bus.

The maximum update rate of data from the OBDII bus is 50Hz. Each parameter enabled constitutes one update from the vehicle. The update rate from the vehicle is not linked to any messages from the host. The host will be able to request data from the OBDII Streamer immediately after the previous response, but the OBDII data will be limited to an update rate of 50Hz.



Vehicle status parameters, such as the MIL and DTCs, will be updated from the vehicle once per minute. The host can request these parameters from the OBDII Streamer at any rate desired.

Startup Procedure

On initial power on the OBDII Streamer must detect the vehicle type and supported parameters. This process can take up to 1 minute. During this time the OBDII Streamer will be responsive to host requests.

While the host is detecting the vehicle's information most messages will be responded to with the following response.

```
0x01 0x01 0x81 0x00 0x83            //Hold Off Communications message
```

The following commands will respond normally during startup:

- READ_MODEL_NUMBER
- FIRMWARE_VERSION
- READ_SERIAL
- UPGRADE_FIRMWARE

When the OBDII Streamer has finished the initialization sequence it will send a DEVICE_CONFIGURED message.

After this message is received the OBDII Streamer will be ready for communications.

Non-Volatile Storage

Most configuration settings and parameters are stored in volatile memory and must be reconfigured on startup. The following settings and parameters are saved in non-volatile memory.

- Serial Number
- BAUD
- OEM_ID
- Model Number
- VIN
- Application Selection (Normal, Bootload, EEPROM Program, Bypass)
- Firmware/EEPROM Versions
- Vehicle information (OBDII Bus and Available Parameters)

Transferring to a New Vehicle

The OBDII Streamer stores all vehicle information in non-volatile memory. To improve startup times in the event of a power loss all parameters will be reloaded from memory.

If the OBDII Streamer is removed from one vehicle and installed in another, the vehicle must be force detected. On startup the previous vehicle's data will be loaded. To force the OBDII Streamer to redetect vehicle information send the REDETECT_VEHICLE command.

Ignition Detection

The vehicles J1962 diagnostic connector is always powered, even with the vehicle ignition off. The OBDII Streamer detects the ignition status by analyzing data retrieved over the OBDII bus. There are two methods used to determine ignition status.

First, the engine speed is monitored from the vehicle. When engine speed is detected as zero the OBDII Streamer will enter ignition off mode.

Second, most vehicles stop responding to OBDII requests when the ignition is off. After three seconds of no responses from the vehicle the OBDII Streamer will enter ignition off mode.

The OBDII Streamer polls the vehicle for ignition status every 2 seconds. When the ignition status changes from off to on the device will return to normal mode and will send an ignition status message to the host.

Low Power Modes

Near-Zero Power option:
The lowest power option is for the Host device to supply power to the LDV Streamer through Pin 9 of the RS-232 connector.

- The only power draw from the vehicle is the secondary scan tool detect circuit.
- Maximum Power Consumption: 0.036W
- Maximum Time from power-on to valid vehicle data: 30 Seconds when vehicle is running.

Sleep Mode:
If the Host device can not supply power to the LDV Streamer, the next lowest power mode is Sleep Mode. During Sleep Mode there is no polling of the vehicle for its ignition status.

- Sleep Mode can be entered in one of two ways:
  - Command from the Host
  - Entered within 5 seconds of communication loss or 0 RPM when the LDVDS is configured to Sleep Mode as its default low power mode.
- Sleep Mode can only be exited by a command from the Host
- Only the Host interface is required to remain responsive.
- Maximum Power Consumption: 0.38W.
- Maximum Time from Wake from Sleep command to valid vehicle data: 15 Seconds when vehicle is running.

Standby Mode:
During Standby Mode the vehicle is polled periodically to find out its ignition status. Standby Mode will be the default mode used.

- Standby Mode is entered within 5 seconds of communications loss or 0 RPM.
- Standby Mode is exited within 10 seconds after vehicle starts communicating
- LDV Streamer remains responsive to all host communications while in Standby Mode.
- Maximum Power Consumption: 0.42W.
- Maximum Time from power-on to valid vehicle data: 15 Seconds when vehicle is running.

Special Cases:

- First Time Installation: The OBDII Streamer does not enter any low power modes until it has been installed on a vehicle and has detected the supported parameters.
- Firmware updates: After a firmware update, the OBDII Streamer will go into the appropriate power state for normal operation.
- Unintentional Reset (Battery removal, brownout, etc.): After an unintentional reset, the OBDII Streamer will go into the appropriate power state for normal operation. Any power modes that have been commanded should remain after an unintentional reset.
- Move to Different Vehicle:

- The LDV Streamer should remain in the appropriate low power mode until it has been determined that the engine is running.
- The LDV Streamer requires a REDETECT_VEHICLE command to work properly after moving to a different vehicle.

Settings

The following variables make up all of the settings that should be known or changed when using the OBDII Streamer.

Overview

The OBDII Streamer is capable of transmitting messages to the host automatically based on user set criteria. By default these messages are disabled.

Time Based Updates

"Time Based" updates will provide an update of all enabled parameters at every time interval specified by the user.

Time based updating will work simultaneously with "Threshold Based" updating and "As Requested" updating. If "Full Speed" update mode is enabled all other update methods for the selected parameter will be disabled.

Threshold Based Updates

"Threshold Based Updating" will send an update to the host whenever an enabled parameter with threshold checking enabled exceeds the threshold.

"Threshold Based" updating will work simultaneously with "Time Based" updating and "AS Requested" updating. If "Full Speed" update mode is enabled all other update methods for that parameter will be disabled.

Full Speed Updates

"Full Speed" update mode will send a message to the host every time updated data is received for the "Full Speed" parameter.

When "Full Speed" update mode is enabled both "Time Based" and "Threshold Based" update modes will be disabled for the selected parameter.

Example: 2 parameters have been enabled for updates using the "Full Speed" update mode. Two message will be sent to the vehicle every time the parameter list is cycled through. Each message will contain the latest value of one of the "Full Speed" parameters.

As Requested Updates

"As Requested" update mode allows the host to request a single parameter, or all parameters, at any time and receive a snapshot of the parameter values.

"As Requested" update mode is not a exclusive mode. This means that "Time Based", "Threshold Based", and "Full Speed" update modes can be enabled and parameters can still be requested at any time.

BAUD

The BAUD rate setting is only applicable to the RS-232 serial version of the OBDII Streamer. The BAUD rate will default to 115.2kbps.

OEM_ID

The OEM_ID field is available for the customer to program as they choose. This field is 10 bytes long and defaults to all '0x00'.

An example use of this field is to key the software to a particular piece of hardware.

Update Mode

On power up all update modes are disabled. In order to use any of the update modes listed below they must be configured and enabled using the SET_UPDATE_MODE and CHANGE_PARAMETER_THRESHOLD commands.

- Time Based
- Threshold Based
- Full Speed

In order to receive automatic messages based on time or threshold the CHANGE_PARAMETER_THRESHOLD command must be sent for each parameter.

Enable_Parameters

By default, all parameters will be updated from the vehicle. The Enable_Parameters command can be sent in order to enable updates on only the desired parameters.

Setting the Enable_Parameters command is desirable in some cases to increase the parameter update rate. The update rate for each parameter is equal to the total update rate from the vehicle divided by the number of enabled parameters.

The update rate from the vehicle is fixed and varies from vehicle to vehicle, but will not exceed 50Hz.

Bypass Mode

The OBDII Streamer is a combination of a LDV OBDII interface and an application processor. The application processor is responsible for hiding the vehicle specific information from the host.

The host can bypass the application processor and communicate directly with the OBDII interface by setting bypass mode. There are three commands that control bypass mode.

- ENABLE_BYPASS_MODE – This command will temporarily enable bypass mode. A 10 second period of no traffic from the host will cause the application processor to return to normal mode.

- SET_BYPASS_MODE – This command will cause the application processor to stay in bypass mode until commanded back to normal mode. This information is stored in non-volatile memory and will cause the application processor to remain in bypass mode on subsequent power cycles.
- EXIT_BYPASS_MODE – This command will immediately exit bypass mode and return the application processor to normal operation. If bypass mode settings were stored in non-volatile memory then that will be overwritten with normal mode settings.

To determine which mode the OBDII Streamer is operating in send a model number request (CMD 0x02) to the hardware. If in bypass mode the hardware will respond with “LDVOBDII” and in normal mode the response will be “LDVDS-S”.

Parameters

This list matches a parameter with its identifier.

ID	Parameter	Return Size	Units	Range
0x00	Vehicle Speed	2 Bytes	MPH	0 to 160
0x01	Engine Speed	2 Bytes	RPM	0 to 16384
0x02	Throttle Position	2 Bytes	%	0 to 100
0x03	Odometer	4 Bytes	Miles	0 to 999,992
0x04	Fuel Level	2 Bytes	%	0 to 100
0x05	Fuel Level Remaining	2 Bytes	Gallons	0 to 500
0x06	Transmission Gear	2 Bytes	PRND	PRND
0x08	Ignition Status	2 Bytes	On/Off	On/Off
0x09	MIL Status	2 Bytes	On/Off	On/Off
0x0A	Airbag dash indicator	2 Bytes	On/Off	On/Off
0x0B	ABS dash indicator	2 Bytes	On/Off	On/Off
0x0C	Fuel Rate	2 Bytes	Gallons per Hour	0 to 29.99
0x0D	Battery Voltage	2 Bytes	Volts	0 to 18
0x0E	PTO Status	2 Bytes	On/Off	On/Off
0x0F	Seatbelt Fastened	2 Bytes	Yes/No	N/A
0x10	Misfire Monitor	2 Bytes	Status	N/A
0x11	Fuel System Monitor	2 Bytes	Status	N/A
0x12	Comprehensive Component Monitor	2 Bytes	Status	N/A
0x13	Catalyst Monitor	2 Bytes	Status	N/A
0x14	Heated Catalyst Monitor	2 Bytes	Status	N/A
0x15	Evaporative System Monitor	2 Bytes	Status	N/A
0x16	Secondary Air System Monitor	2 Bytes	Status	N/A
0x17	A/C System Refrigerant Monitor	2 Bytes	Status	N/A
0x18	Oxygen Sensor Monitor	2 Bytes	Status	N/A
0x19	Oxygen Sensor Heater Monitor	2 Bytes	Status	N/A
0x1A	EGR System Monitor	2 Bytes	Status	N/A
0x1B	Brake Switch Status	2 Bytes	Pressed/Not Pressed	N/A
0x1D	Cruise Control Status	2 Bytes	On/Off	On/Off
0x1E	Turn Signal Status	2 Bytes	Left/Right/Off	N/A
0x1F	Oil Pressure Lamp	2 Bytes	On/Off	On/Off
0x20	Brake Indicator Light	2 Bytes	On/Off	On/Off
0x21	Coolant Hot Light	2 Bytes	On/Off	On/Off
0x22	Trip Odometer	4 Bytes	Miles	0 to 999,992
0x23	Trip Fuel Consumption	4 Bytes	Gallons	0 to 999,992

0x00 Vehicle Speed

Scaling Equation:
Result = Returned Data * 1 / 410
Units: Miles per hour

EX: Return Data = 26650
Result = 26650 * 1 / 410
Result = 65 Miles per hour

0x01 Engine Speed

Scaling Equation:
Result = Returned Data * 1 / 4
Units: Revolutions per Minute

EX: Return Data = 12000
Result = 12000 * 1 / 4
Result = 3000 Revolutions per Minute

0x02 Throttle Position

Scaling Equation:
Result = Returned Data * 1 / 655
Units: %

EX: Return Data = 12000
Result = 12000 * 1 / 655
Result = 18.32 % Throttle Pedal Position

0x03 Odometer

Scaling Equation:
Result = Returned Data * 1 / 1
Units: Miles

EX: Return Data = 58,125
Result = 58,125 * 1 / 1
Result = 58,125 Miles

0x04 Fuel Level

Scaling Equation:
Result = Returned Data * 1 / 655
Units: %

EX: Return Data = 23578
Result = 23578 * 1 / 655
Result = 35.99 %

0x05 Fuel Level Remaining

Scaling Equation:
Result = Returned Data * 1 / 131
Units: Gallons

EX: Return Data = 1577
Result = 1577 * 1 / 131
Result = 12.03 Gallons

0x06 Transmission Gear

Return Value:	Gear:
0	Unknown
1	Park
2	Neutral
3	Drive
4	Reverse

0x08 Ignition Status

Return Value of 1 = Ignition Off
Return Value of 0 = Ignition On

0x09 MIL Status

Return Value of 1 = MIL Off
Return Value of 0 = MIL On

0x0A Airbag Dash Indicator

Return Value of 1 = Airbag Indicator Off
Return Value of 0 = PTO On

0x0B ABS Dash Indicator

0x0C Fuel Rate

Scaling Equation:
Result = Returned Data * 1 / 2185
Units: Gallons per Hour

EX: Return Data = 25650
Result = 25650 * 1 / 2185
Result = 11.74 Gallons per Hour

0x0D Battery Voltage

Scaling Equation:
Result = Returned Data * 1 / 3641
Units: Volts

EX: Return Data = 45650
Result = 45650 * 1 / 3641
Result = 12.538 Volts

0x0E PTO Status

Return Value of 1 = PTO Off
Return Value of 0 = PTO On

0x0F Seatbelt Fastened

Return Value of 1 = Seat Belt not Fastened
Return Value of 0 = Seat Belt Fastened

0x10 to 0x1A Monitors

Return Value of 1 = Monitor Not Complete
Return Value of 0 = Monitor Complete

0x1B Brake Switch Status

Return Value of 1 = Brake Switch Off
Return Value of 0 = Brake Switch On

0x1D Cruise Control Status

Return Value of 1 = Cruise Control Off
Return Value of 0 = Cruise Control On

0x1E Turn Signal Status

Return Value of 0 = Turn Signal Off
Return Value of 1 = Left Turn On
Return Value of 2 = Right Turn On
Return Value of 3 = Both Signals On

NOTE: On some vehicles the turn signal reports on and off periodically as the signal flashes. Polling may have to be done more frequently from the Streamer to catch the changing signal.

0x1F Oil Pressure Lamp

Return Value of 1 = Oil Pressure Lamp Off
Return Value of 0 = Oil Pressure Lamp On

0x20 Brake Indicator Light

Return Value of 1 = Brake Indicator Light Off
Return Value of 0 = Brake Indicator Light On

0x21 Coolant Hot Lamp

Return Value of 1 = Coolant Hot Lamp Off
Return Value of 0 = Coolant Hot Lamp On

0x22 Trip Odometer

Scaling Equation:
Result = Returned Data * 1 / 10
Units: Miles

EX: Return Data = 58,125
Result = 58,125 * 1 / 10
Result = 5,812.5 Miles

0x23 Trip Fuel Consumption

Scaling Equation:
Result = Returned Data * 1 / 128
Units: Gallons
EX: Return Data = 8,128
Result = 8128 * 1 / 128
Result = 63.5 Gallons
NOTE: It is basically returned in Ounces.

Commands

0x02 – READ_MODEL_NUMBER

Description:
This command will return the model number as an alpha-numeric ASCII string.

The following model numbers can be returned depending on the type of Streamer used.

LDVDS-S	RS-232 OBDII Streamer
LDVDS-U	USB OBDII Streamer
LDVDS-B	Bluetooth OBDII Streamer

Command to Send:
0x01 0x01 0x02 0x00 0x04

Response:
0x01 0x01 0x82 DL MODEL CS

Field Description:

MODEL	Variable length model number up to 16 digits long. This field is returned as ASCII characters.
DL	Data Length (Length of model number)
CS	Summation checksum

Example:
TX: 0x01 0x01 0x02 0x00 0x04

RX: 0x01 0x01 0x82 0x07 0x4C 0x44 0x56 0x44 0x53 0x2D 0x53 0x88

The model number returned is LDVDS-S.

0x04 – OBDII_FIRMWARE_VERSION

Description:
This command will return the firmware version number of the OBDII interface.

Command to Send:
0x01 0x01 0x04 0x00 0x06

Response:
0x01 0x01 0x84 0x03 VER1 VER2 VER3 CS

Field Description:

VER1	Hex version number digit 1
VER2	Hex version number digit 2
VER3	Hex version number digit 3
CS	Summation checksum

Example:
TX: 0x01 0x01 0x04 0x00 0x06

RX: 0x01 0x01 0x84 0x03 0x04 0x00 0x03 0x90

The version number returned is 0x04 0x00 0x03 = 4.03

0x05 – FIRMWARE_VERSION

Description:
This command will return the firmware version number of the host interface processor.

Command to Send:
0x01 0x01 0x05 0x00 0x07

Response:
0x01 0x01 0x85 0x03 VER1 VER2 VER3 CS

Field Description:

VER1	Hex version number digit 1
VER2	Hex version number digit 2
VER3	Hex version number digit 3
CS	Summation checksum

Example:
TX: 0x01 0x01 0x05 0x00 0x07

RX: 0x01 0x01 0x85 0x03 0x01 0x00 0x01 0x8C

The version number returned is 0x01 0x00 0x01 = 1.01

0x06 – DATABASE_FIRMWARE_VERSION

Description:
This command will return the firmware version number of the embedded database..

Command to Send:
0x01 0x01 0x06 0x00 0x08

Response:
0x01 0x01 0x86 0x03 VER1 VER2 VER3 CS

Field Description:

VER1	Hex version number digit 1
VER2	Hex version number digit 2
VER3	Hex version number digit 3
CS	Summation checksum

Example:
TX: 0x01 0x01 0x06 0x00 0x08

RX: 0x01 0x01 0x86 0x03 0x01 0x00 0x01 0x8D

The version number returned is 0x01 0x00 0x01 = 1.01

0x07 – READ_SERIAL_NUMBER

Description:

This command will return the ten digit hardware serial number. The serial number is programmed at manufacturing time and should match the serial number sticker on the side of the hardware.

Command to Send:

0x01 0x01 0x07 0x00 0x09

Response:

0x01 0x01 0x87 0x0A SERIAL CS

Field Description:

SERIAL 10 digit serial number returned as numeric ASCII characters.

CS Summation checksum

Example:

TX: 0x01 0x01 0x07 0x00 0x09

RX: 0x01 0x01 0x87 0x0A 0x31 0x35 0x33 0x31 0x38 0x32 0x36 0x34 0x33 0x37 0x9B

The serial number returned is 1531826437.

0x09 – CONFIG_OEM_ID

Description:
This command will set or read the 10 digit OEM ID. Each digit of the OEM_ID is 1 byte and can be any possible value.

The OEM_ID will be saved in non-volatile memory.

The first byte of the data field indicates if the OEM_ID will be read or written.

0x00 = read
0x01 = write

Command to send to read the OEM_ID:
0x01 0x01 0x09 0x01 0x00 0x0C

Response:
0x01 0x01 0x89 0x0B 0x00 OEM_ID CS

Command to send to write the OEM_ID:
0x01 0x01 0x09 DL 0x01 OEM_ID CS

Response:
0x01 0x01 0x89 0x0B 0x01 OEM_ID CS

Field Description:

OEM_ID	10 digit OEM_ID
DL	Data field length
CS	Summation checksum

Example: Set the OEM_ID
TX:
0x01 0x01 0x09 0x0B 0x01 0x01 0x35 0x73 0x99 0x24 0x72 0xF3 0x17 0xAC 0xBB 0x60

RX:
0x01 0x01 0x89 0x0B 0x01 0x01 0x35 0x73 0x99 0x24 0x72 0xF3 0x17 0xAC 0xBB 0xE0

Example: Read the OEM_ID
TX: 0x01 0x01 0x09 0x01 0x00 0x0C

RX:
0x01 0x01 0x89 0x0B 0x00 0x01 0x35 0x73 0x99 0x24 0x72 0xF3 0x17 0xAC 0xBB 0xDF

0x15 – SERIAL_BAUD

Description:
This command will read or write RS-232 Baud Rate settings.

The configuration data is stored in non-volatile memory and will be loaded on system power-on.

Command to Send:
0x01 0x01 0x15 0x03 RD/WR BAUD 0x01 CS

Response:
0x01 0x01 0x95 0x03 RD/WR BAUD 0x01 CS

Field Description:

RD/WR — This indicates if the message will be a read or a write.
0x00 = read   0x01 = write

Note: If the request message is a configuration read then the BAUD and STATUSMSG fields should not be included in the request message, and the data length will be 1 byte.

BAUD — BAUD setting
0x00 = 9600kbps   0x01 = 19200kbps
0x02 = 38400kbps   0x03 = 56000kbps
0x04 = 115200kbps (Default)

CS — Summation checksum

Example: Baud = 19200kbps
TX: 0x01 0x01 0x15 0x03 0x01 0x01 0x01 0x1D

RX: 0x01 0x01 0x95 0x03 0x01 0x01 0x01 0x9D

0x20 – GET_SUPPORTED_PARAMETERS

Description:
This command will return an identifier byte for each supported parameter.

Only supported parameters will be requested from the vehicle. Parameter values reported for an unsupported parameter are invalid. Supported parameters are available after the vehicle has been detected.

Command to Send:
0x01 0x01 0x20 0x00 0x22

Response:
0x01 0x01 0xA0 DL SUPPORTED_PARAMETERS CS

Field Description:

DL	Data Length Byte
SUPPORTED_PARAMETERS	Variable length data field containing one byte for each supported parameter.
CS	Summation checksum

Example:
TX:
0x01 0x01 0x20 0x00 0x22

RX:
0x01 0x01 0xA0 0x07 0x01 0x03 0x04 0x07 0x09 0x0A 0x13 0xDE

The supported parameters in this example are:
0x00 – Vehicle Speed
0x02 – Throttle Position
0x03 – Odometer
0x06 – Transmission Gear
0x08 – Ignition Status
0x09 – MIL Status
0x11 – Fuel System Monitor

0x21 – ENABLE_PARAMETERS

Description:
This command will enable or disable only the specified parameters to be updated from the vehicle. If parameters are enabled that are not supported by the vehicle the OBDII Streamer will send a response indicating which parameters are not supported.

By default all available parameters are enabled and queried from the vehicle. Because the update rate of the vehicle is limited to a maximum of 50 Hz, it may be desirable to disable some parameters to get better update rates on parameters of interest.

Command to Send:
0x01 0x01 0x21 DL RD/WR ENABLE/DISABLE PARAMETER_LIST CS

Response to write:
0x01 0x01 0xA1 DL 0x01 ENABLE/DISABLE NOT_SUPP CS

If sending a command to disable parameters, the NOT_SUPP field will not be included in the response.

Response to read:
0x01 0x01 0xA1 DL 0x00 PARAMETER_LIST CS

Field Description:

DL
Data Length Byte

RD/WR
This indicates if the message will be a read or a write.
0x00 = read
0x01 = write

Note: If the request message is a configuration read then the ENABLE/DISABLE and PARAMETER_LIST fields should not be included, and the data length will be 1 byte.

ENABLE/DISABLE
This field in 1 byte that indicates if the parameters specified in PARAMETER_LIST should be enabled or disabled. The state of the parameters not specified in the PARAMETER_LIST field will not change.
0x00 = Enable specified parameters
0x01 = Disable specified parameters

This field will not be present in a response to a read enabled parameters command.

PARAMETER_LIST
Variable length data field containing one byte for each parameter to be enabled. Each byte corresponds to a

parameter ID. This should not be included in the request message if reading the ENABLE_PARAMETERS data.

If a 0xFF is included in the parameter list all parameters will be either enabled or disabled according to the ENABLE/DISABLE byte.

NOT_SUPP Variable length field containing one byte for each parameter that was requested to be enabled, but is not supported by the vehicle. This may be a zero length field.

CS Summation checksum

Example 1: Enable Parameters
TX: 0x01 0x01 0x21 0x05 0x01 0x00 0x00 0x01 0x02 0x2C

RX: 0x01 0x01 0xA1 0x02 0x01 0x00 0xA6

The command sent requests Vehicle Speed, Engine Speed, and Throttle Position to be enabled. All parameters that were requested to be enabled are supported by the vehicle.

Example 2: Enable Parameters
TX: 0x01 0x01 0x21 0x05 0x01 0x00 0x00 0x01 0x02 0x2C

RX: 0x01 0x01 0xA1 0x03 0x01 0x00 0x02 0xA9

The command sent requests Vehicle Speed, Engine Speed, and Throttle Position to be enabled. All parameters that were requested to be enabled are supported by the vehicle except for Throttle Position.

Example 3: Read Enabled Parameters
TX: 0x01 0x01 0x21 0x01 0x00 0x24

RX: 0x01 0x01 0xA1 0x03 0x00 0x00 0x01 0xA7

The command sent requests a list of all enabled parameters. Vehicle Speed and Engine Speed are enabled and listed in the response.

Example 4: Disable Parameters
TX: 0x01 0x01 0x21 0x04 0x01 0x01 0x00 0x01 0x2A

RX: 0x01 0x01 0xA1 0x04 0x01 0x01 0x00 0x01 0xAA

The command sent requests that Vehicle Speed and Engine Speed be disabled. The response indicates that both Vehicle Speed and Engine Speed are disabled.

0x22 – GET_PARAMETER

Description:
This command will return the current value for the requested parameter. Since the most current parameter's results are stored in the OBDII Streamer the response delay will be very short.

Command to Send:
0x01 0x01 0x22 DL PARAMETER_ID CS

Response:
0x01 0x01 0xA2 DL PARAMETER_DATA CS

Field Description:

DL Data Length Byte

PARAMETER_ID Variable length data field containing one byte for each parameter requested. Each byte corresponds to a parameter ID.

Multiple parameters can be specified in the PARAMETER_ID field, or 0xFF can be specified to request all enabled parameters.

A maximum of 11 parameters can be specified in one message. If 0xFF is sent to request all enabled parameters then the response may be broken into multiple responses depending on the number of enabled parameters.

PARAMETER_DATA The PARAMETER_DATA field is a variable length data field that contains both the requested parameter(s) identifier and the parameter data. All parameter identifiers are one byte long, and parameter data is variable length. The format of the data field is parameter identifier, Data, parameter identifier, Data, Repeat.

CS Summation checksum

Example:
TX:
0x01 0x01 0x22 0x03 0x01 0x02 0x03 0x2D

RX:
0x01 0x01 0xA2 0x09 0x01 0x01 0x23 0x02 0x98 0x76 0x03 0x55 0x66 0xA0

The breakdown of the response is:

Parameter 0x01:	Data = 0x0123
Parameter 0x02:	Data = 0x9876
Parameter 0x03:	Data = 0x5566

0x23 – GET_VEHICLE_STATUS

Description:
This command will return the status of the vehicle ignition and the secondary OBDII tool status.

Command to Send:
0x01 0x01 0x23 0x00 0x25

Response:
0x01 0x01 0xA3 0x02 IGNITION OBDII_STAT CS

Field Description:

IGNITION — This is a 1 byte field that indicates the status of the vehicle ignition.

0x00 = Ignition OFF
0x01 = Ignition ON

Note: The methods used to determine the ignition status actually detect when the OBDII bus is responding to requests, and not the ignition status. It is possible that some vehicles will continue to respond while the ignition is off. All vehicles are required to communicate when the key is in the on position and the engine is not running.

OBDII_STAT — This is a 1 byte field that indicates if a second scan tool is connected to the OBDII port.

0x00 = No Tool
0x01 = Tool Connected

CS — Summation checksum

Example:
TX:
0x01 0x01 0x23 0x00 0x25

RX:
0x01 0x01 0xA3 0x02 0x01 0x00 0xA8

The response indicates that the ignition is on and no secondary tool is connected.

0x24 – REDETECT_VEHICLE

Description:
This command will force all detected vehicle information to be redetected. The REDETECT_VEHICLE command should only be required if the Streamer is removed from one vehicle and moved to another.

Command to Send:
0x01 0x01 0x24 0x00 0x26

Response:
0x01 0x01 0xA4 0x00 0xA6

This response will be transmitted immediately. When the detection process is finished the DEVICE_CONFIGURED message will be transmitted.

Example:
TX: 0x01 0x01 0x24 0x00 0x26

RX: 0x01 0x01 0xA4 0x00 0xA6 -Received immediately

The configuration time may take up to 1 minute to complete. Any requests during this time will be responded to with a DELAY_COMMUNICATIONS message.

RX: 0x01 0x01 0x80 0x00 0x82 -DEVICE_CONFIGURED message

0x25 – GET_VEHICLE_INFO

Description:
The GET_VEHICLE_INFO command will return the VIN, OBDII protocol being used, or any DTCs set.

Command to Send:
0x01 0x01 0x25 0x01 INFO_TYPE CS

Response:
0x01 0x01 0xA5 DL INFO_TYPE INFO_RESPONSE CS

Field Description:

DL — Data Length Byte

INFO_TYPE — The INFO_TYPE field specifies the return data.

0x00 = VIN
0x01 = OBDII Protocol
0x02 = Diagnostic Trouble Codes (DTCs)

INFO_RESPONSE — This is a variable length response. This field is specified by INFO_TYPE and will be either the VIN or the OBDII Protocol.

INFO_TYPE = 0 = VIN
The INFO_RESPONSE is a 17 character VIN.

INFO_TYPE = 1 = OBDII Protocol
INFO_RESPONSE is a 1 digit number that defines the OBDII protocol detected.
0 = None
1 = J1850VPW
2 = J1850PWM
3 = ISO9141-2
4 = SCI
5 = KWP2000
6 = CAN 11 bit
7 = CAN 29 bit
8 = GMLAN Low Speed

INFO_TYPE = 2 = DTCs
The INFO_RESPONSE is a variable length field containing DTCs. The DTCs will be ASCII values and each code will be 5 digits long.

Only confirmed DTCs are reported.

CS Summation Checksum

Example
This command requests all DTCs from the vehicle.
TX: 0x01 0x01 0x25 0x01 0x02 0x2A

RX:
0x01 0x01 0xA5 0x0B 0x02 0x50 0x30 0x34 0x33 0x30 0x50 0x30 0x32 0x35 0x30 0xE2

The above response indicates that two DTCs are present on the vehicle.

0x26 – SET_SLEEP_MODE_BEHAVIOR

Description:
This command will setup the behavior used when the LDVDS detects that the vehicle ignition is off.

Possible settings:

- Standby Mode: Poll the vehicle bus every two seconds in order to determine the ignition status.(DEFAULT)
- Sleep Mode: Remain in low power mode until commanded to wake. No vehicle polling.

If the LDVDS is set to remain in Sleep Mode until commanded and the device is reset by a power glitch the LDVDS will enter Sleep Mode on startup. The host must command the LDVDS out of sleep mode in this scenario.

Command to Send:
0x01 0x01 0x26 0x01 SLEEP CS

Response:
0x01 0x01 0xA6 0x01 SLEEP CS

Field Description:

SLEEP	This field is a 1 byte in length and determines if the LDVDS will poll the vehicle (Standby Mode) or wait for the host to send a wake command (Sleep Mode). 0 = Go to Standby Mode when ignition off is detected. (DEFAULT) 1 = Go to Sleep Mode when ignition off is detected.
CS	Summation checksum

Example:
This command sets LDVDS to remain in sleep mode until commanded out by the host.
TX:
0x01 0x01 0x26 0x01 0x01 0x2A

RX:
0x01 0x01 0xA6 0x01 0x01 0xAA

0x27 – ENTER_SLEEP_MODE

Description:

This command will force the LDVDS to enter Sleep Mode. The firmware will not exit Sleep Mode until the WAKE_FROM_SLEEP_MODE command has been received.

NOTE:

The LDVDS does not enter sleep mode when it is detecting the vehicle. If the ENTER_SLEEP_MODE command is issued when the Streamer is first installed or after a REDETECT_VEHICLE command, the LDVDS will remain in full power mode until the vehicle is detected.

NOTE:

The LDVDS does not poll the vehicle during sleep mode. Parameters that are cumulative such as Calculated Trip Odometer and Trip Fuel will have errors introduced if the Streamer is commanded to enter sleep when the vehicle is running.

Command to Send:

0x01 0x01 0x27 0x00 0x29

Response:

0x01 0x01 0xA7 0x00 0xA9

0x28 – WAKE_FROM_SLEEP_MODE

Description:

This command will enable the LDVDS to wake from Sleep Mode. When this command is sent, the LDVDS will enter Standby Mode until the engine RPM is greater than zero.

Command to Send:

0x01 0x01 0x28 0x00 0x2A

Response:

0x01 0x01 0xA8 0x00 0xAA

0x30 – SET_TIME_UPDATES

Description:
This command will setup the periodic update settings for a single parameter.

Note that when a timed update message is sent to the host the message will be packaged in the same format as command 0x22 but with a new control byte. Time based update messages will use control byte 0xC0.

The SET_UPDATE_MODE command must be sent to enable all parameters configured for time based updates to be sent.

Command to Send:
0x01 0x01 0x30 0x04 PID SETTINGS TVALUE CS

Response:
0x01 0x01 0xB0 0x04 PID SETTINGS TVALUE CS

Field Description:
PID This field is a 1 byte PID. Use this field to select the PID that will be configured.

SETTINGS Settings

Bit 0 - This bit will enable or disable periodic updates for the selected parameter.

0x00 = Disabled
0x01 = Enabled

If disabling a parameter the TVALUE byte must be sent, but can be set to any value since it will be ignored.

TVALUE Time Value

This field is two bytes long and will configure the period between updates. The resolution of the timer is 50mS.

Timeout Range = 50mS to 54.6 minutes

Period = 50mS * TVALUE

CS Summation checksum

Example:
This command sets up the vehicle speed PID to be transmitted every 250mS.

TX:
0x01 0x01 0x30 0x04 0x00 0x01 0x00 0x05 0x3C

RX:
0x01 0x01 0xB0 0x04 0x00 0x01 0x00 0x05 0xBC

250mS delay

RX:
0x01 0x01 0xC0 0x03 0x00 0x01 0x23 0xE9

The above response is received 250 mS after the Time Updating was setup and indicates that the vehicle speed has an unscaled value of 0x0123.

0x31 – SET_THRESHOLD_UPDATES

Description:
This command will setup the threshold triggered messages for a single parameter.

Note that when a threshold is tripped and a message is sent to the host the message will be packaged in the same format as command 0x22 but with a new control byte. Threshold based update messages will use control byte 0xC1.

The SET_UPDATE_MODE command must be sent to enable all parameters configured for threshold based updates to be sent.

Command to Send:
0x01 0x01 0x31 0x04 PID SETTINGS TVALUE CS

Response:
0x01 0x01 0xB1 0x04 PID SETTINGS TVALUE CS

Field Description:

PID — This field is a 1 byte PID. Use this field to select the PID that will be configured.

SETTINGS — Threshold Settings

Bit 0 - This bit will enable or disable threshold updates for the selected parameter.

0x00 = Disabled
0x01 = Enabled

If disabling a parameter the TVALUE byte must be sent, but can be set to any value since it will be ignored.

Bit 1 – This bit specifies if trigger level is greater than or less than the TVALUE.
0 = Send a message when the parameter > TVALUE
1 = Send a message when the parameter < TVALUE

If the parameter being configured only has two states then Bit 1 can be configured as follows:
0 = Yes/True
1 = No/False

In this case the TVALUE will be ignored, but should still be sent to the OBDII Streamer in the request message.

TVALUE Threshold Value

This field is two bytes long and will configure the threshold value used to determine when to send a message. The value in this field should be passed to the Intelligent OBDII Streamer using reverse scaling.

CS Summation checksum

Example:

This command sets up the vehicle speed PID to be transmitted when the vehicle speed is greater than 40 MPH.

TX:

0x01 0x01 0x31 0x04 0x00 0x01 0x40 0x10 0x88

RX:

0x01 0x01 0xB1 0x04 0x00 0x01 0x40 0x10 0x08

Threshold Value = 0x4010 = 16400

16400 / 410 = 40MPH

Unknown Delay. The following message will be received every time the parameter is updated AND the vehicle speed parameter is over 40 MPH.

RX:

0x01 0x01 0xC1 0x03 0x00 VEHICLE_SPEED CS

0x32 – SET_FULLSPEED_UPDATES

Description:
This command will setup the full speed updating parameters for a single parameter.

Note that a message will be sent to the host every time the configured parameter is updated from the vehicle. The update frequency may vary, and will not be faster than 50Hz. All full speed update messages will use a control byte of 0xC2.

The SET_UPDATE_MODE command must be sent to full speed update messages in addition to using this command.

Command to Send:
0x01 0x01 0x32 0x02 PID SETTINGS CS

Response:
0x01 0x01 0xB2 0x02 PID SETTINGS CS

Field Description:

PID	This field is a 1 byte PID. Use this field to select the PID that will be configured.
SETTINGS	Settings Bit 0 - This bit will enable or disable full speed updates for the selected parameter. 0x00 = Disabled 0x01 = Enabled
CS	Summation checksum

Example:
This command sets up the vehicle speed PID to be transmitted every time a response is received from the vehicle..
TX:
0x01 0x01 0x32 0x02 0x00 0x01 0x37

RX:
0x01 0x01 0xB2 0x02 0x00 0x01 0xB7

RX:
0x01 0x01 0xC2 0x03 0x00 0x01 0x23 0xEB

0x33 – READ_PARAMETER_UPDATE_MODES

Description:
This command will return the update mode settings for the specified parameter.

Command to Send:
0x01 0x01 0x33 0x01 PID CS

Response:
0x01 0x01 0xB3 0x06 PID TIME THRESHOLD TSETTINGS CS

Field Description:

PID — This field is a 1 byte PID.

TIME — This field is two bytes and is reports the period set for the selected parameter.

Period = TIME * .05 seconds.

THRESHOLD — This field is two bytes and reports the threshold set for the selected parameter.

TSETTINGS — Threshold Settings

Bit 0 – Time updates Enabled

0x00 = Disabled
0x01 = Enabled

Bit 1 – Threshold updates Enabled

0x00 = Disabled
0x01 = Enabled

Bit 2 – Full Speed updates Enabled

0x00 = Disabled
0x01 = Enabled

Bit 3 – This bit specifies if a message will be sent when the parameter data is above the threshold value or below.

0 = Send a message when the parameter > TVALUE
1 = Send a message when the parameter < TVALUE

If the parameter being configured only has two states then Bit 3 is specified as follows:
0 = Yes/True
1 = No/False

CS	Summation checksum

Example:
This command requests the settings for the vehicle speed parameter.

TX:
0x01 0x01 0x33 0x01 0x01 0x37

RX:
0x01 0x01 0xB3 0x06 0x01 0x00 0x05 0x01 0x23 0x01 0xE6

The above response indicates that time based updates are the only enabled updates and a periodic message will be sent every 250 mS.

0x35 – SET_UPDATE_MODE

Description:
This command enables or disables the different types of update modes. All update modes can be enabled simultaneously.

This command is a general enable for all PIDs configured for a particular update mode. For example, if a SET_UPDATE_MODE command is sent to enable time based updating, all parameters configured for time based updating will be enabled and data will start to be transmitted at the user selected intervals.

This configuration data will NOT be saved to non-volatile memory and must be set in order to receive update messages. “As Requested” update mode does not require the SET_UPDATE_MODE message to be sent before parameters can be requested.

Command to Send:
0x01 0x01 0x35 0x02 MODE ENABLE CS

Response:
0x01 0x01 0xB5 0x02 MODE ENABLE CS

Note: If the vehicle is turned off when this command is sent, the response will be an Ignition Off Status Message instead of the above response, and the mode will NOT be enabled.

Field Description:

MODE — The MODE byte selects the update mode.
0x00 = Time Based
0x01 = Threshold Based
0x02 = Full Speed
0xFF = All Modes

ENABLE — The ENABLE byte enables or disables the specified MODE.

0x00 = Disable
0x01 = Enable

CS — Summation checksum

Example:
TX:
0x01 0x01 0x35 0x02 0x01 0x01 0x3B

RX:
0x01 0x01 0xB5 0x02 0x01 0x01 0xBB

This command enables threshold based updating for all parameters configured with a threshold and enabled.

0x46 – SET_BYPASS_MODE

Description:
This command enables a bypass mode that allows the host to communicate directly with the OBDII bus. This command enables the bypass mode permanently. The hardware will start in bypass mode on power up or reset. Normal mode can be restored by sending an EXIT_BYPASS_MODE command.

Command to Send:
0x01 0x01 0x46 0x00 0x48

Response:
0x01 0x01 0xC6 0x00 0xC8

Once bypass mode is enabled the host can communicate directly with the OBDII interface. The OBDII interface commands are defined in the OBDII Interface C&R Document.

0x47 – EXIT_BYPASS_MODE

Description:
This command will force the hardware to exit bypass mode. If bypass mode is not currently active then a response will be sent to this command and no action will be taken.

This command will force both temporary bypass mode (Command 0x45) and permanent bypass mode (Command 0x46) to exit and will return the hardware to normal operation.

Command to Send:
0x01 0x01 0x47 0x00 0x49

Response:
0x01 0x01 0xC7 0x00 0xC9

0x59 – UPGRADE_FIRMWARE

Description:
This command will allow the firmware in the OBDII Streamer to be upgraded in the field. The hardware contains two microcontrollers that handle different functions. These are the Host processor and the OBDII processor. Each microcontroller is programmed separately with it's own firmware file.

If the OBDII Streamer needs to have more than one firmware file or EEPROM database file updated, the updates must always be done in the following order:

1) HOST Processor
2) OBDII Processor
3) EEPROM File

Once successfully programmed, the bootloader will exit and run the new firmware.

NOTE:
A minimum of Six seconds is required between each file update to allow the firmware to reset and run. The Streamer will not be responsive to host commands during this time.

In the event of an incomplete program for any reason, the hardware will remain in the bootloader. If power is lost and no valid firmware is loaded the hardware will start in bootload mode.

The programming must always be successfully completed before moving on to the next update file.

Command to Send:
0x01 0x01 0x59 0x00 0x5B

Response:
0x01 0x01 0xD9 0x00 0xDB

Once this command is received there is a One second window to send each of the following commands. If a bootload error occurs then the bootloader will be locked into this step looking for the first message.

TX: “OBDII” for the OBDII processor firmware file
TX: “HOST” for the HOST processor firmware file.

The OBDII Streamer will respond with “OBDII” or “HOST” to confirm receipt.

TX: “!!!!”
RX: “????”
TX: “####”
RX: “#”

At this point the firmware file can be sent to the OBDII Streamer one line at a time.

Each line of the hex file starts with a Colon and ends with a Carriage Return and Line Feed. Between is a list of numbers in ASCII Hex format.

Example: “:020000040000FA”*CRLF*

All characters in the line should be sent to the Streamer including the Colon (0x3A). The Carriage Return(0x0D) and Line Feed(0x0A) characters mark the end of a line but should not be sent to the device.
The numbers in the message are ASCII characters that denote a hex number. Each two characters denote a hex byte that must be sent to the device

Example:
To program file line “:020000040000FA”
Send 0x3A, 0x02, 0x00, 0x00, 0x04, 0x00, 0x00, 0xFA

In Visual basic this looks like:

```
FUNCTION AssembleSend$

    Temp$ = ""
    READ line$

    Temp$ = ":"
    FOR i = 2 TO (LEN(line$) - 1) STEP 2
        Temp$ = Temp$ + CHR$(VAL("&H" + MID$(line$, i, 2)))
    NEXT i

    AssembleSend$ = Temp$
END FUNCTION
```

After each line has been successfully programmed the OBDII Streamer will respond with a “#”. If a programming error occurs the bootloader will respond with a “E” and the programming process must restart.

After programming is completed the bootloader will exit and the new firmware will execute.

Firmware File Naming Conventions

Firmware files will all have a .hex extension and be named as follows:
Host Processor Firmware: *MODEL*_HOST_*Mmm*.hex
Where *MODEL* is the model number of the device and *Mmm* is the Major and two Minor version numbers.
ex. LDVDS-S_HOST_112.hex

OBDII Processor Firmware: *MODEL*_OBDII_*Mmm*.hex
Where *MODEL* is the model number of the device and *Mmm* is the Major and two Minor version numbers.
ex. LDVDS-S_OBDII_401.hex

0x60 – UPDATE_EEPROM

Description:
This command initiates EEPROM update mode to update vehicle specific parameters in the LDVDS EEPROM. This command is only intended for use in manufacturing or field upgrades.

After this command is received the LDVDS will reset into EEPROM_PROGRAM mode. The LDVDS will remain in this mode as long as a valid EEPROM program message is received within a 10 second timeout window. The timeout will be reset every time a message is received.

Each line in the EEPROM data file corresponds to a data block in the EEPROM. Once in EEPROM_PROGRAM mode these blocks should be sent one at a time to the LDVDS. The block number is not included in the file. The first line corresponds to BLOCK_ADDRESS 0x0000. Each consecutive line should increment one block address until the end of the file.

NOTE:
The LDVDS recognizes the end of file by having no communications from the host for 10 Seconds. The Host device should not resume normal communications with the Streamer until after this time has elapsed.

Command to Send to enter EEPROM_PROGRAM Mode:
0x01 0x01 0x60 0x00 0x62

Response:
0x01 0x01 0xE0 0x00 0xE2

Command to send to program data:
0x01 0x01 0x60 0x22 BLOCK_ADDRESS DATA_BLOCK CS

Response from program data message:
0x01 0x01 0xE0 0x22 BLOCK_ADDRESS DATA_BLOCK CS

EEPROM Program Failure Response:
0x01 0x02 0xFF 0x06 0x00 0x08

Field Description:

BLOCK_ADDRESS — This is a 2-byte field (MSB first) that selects one of 65535 32 byte blocks in the EEPROM to write. Each line of the EEPROM data file represents one BLOCK_ADDRESS. BLOCK_ADDRESSes should start at 0x0000.

DATA_BLOCK — This field must be 32 bytes long. When sending, this is one line of the EEPROM data file. In the response, this is the data that was read from the EEPROM after programming.

CS — Summation checksum

<u>Example:</u>

TX: 0x01 0x01 0x60 0x00 0x62 - Enter EEPROM_PROGRAM Mode

RX: 0x01 0x01 0xE0 0x00 0xE2 - Response to Enter EEPROM_PROGRAM Mode

TX: - Program Data Block 0
0x01 0x01 0x60 0x22 0x00 0x00 0x00 0x01 0x02 0x03 0x04 0x05 0x06 0x07 0x08 0x09 0x0A 0x0B 0x0C 0x0D 0x0E 0x0F 0x10 0x11 0x12 0x13 0x14 0x15 0x16 0x17 0x18 0x19 0x1A 0x1B 0x1C 0x1D 0x1E 0x1F 0x74

RX: - Program Data Block 0 Response
0x01 0x01 0xE0 0x22 0x00 0x00 0x00 0x01 0x02 0x03 0x04 0x05 0x06 0x07 0x08 0x09 0x0A 0x0B 0x0C 0x0D 0x0E 0x0F 0x10 0x11 0x12 0x13 0x14 0x15 0x16 0x17 0x18 0x19 0x1A 0x1B 0x1C 0x1D 0x1E 0x1F 0xF4

<u>EEPROM Database File Naming Conventions</u>
Database files will all have a .eep extension and be named as follows:
*MODEL*_DB_*Mmm*.eep
Where *MODEL* is the model number of the device and *Mmm* is the Major and two Minor version numbers.
ex. LDVDS-S_DB_012.eep

0x61 – RESET_TRIP

<u>Description:</u>
This command resets the Trip Parameters (Trip Odomater and Trip Fuel Consumption). It will reset the trip parameters stored in EEPROM and RAM to Zero which signifies the start of the trip.

<u>Command to Send:</u>
0x01 0x01 0x61 0x00 0x63

<u>Response:</u>
0x01 0x01 0xE1 0x00 0xE3

Status Messages

Status messages are messages sent from the OBDII Streamer that weren't expected by the host. The status messages include error messages and vehicle status updates.

0x80 – DEVICE_CONFIGURED

Description:
A DEVICE_CONFIGURED message will be sent to the host when the OBDII Streamer has finished connecting to the vehicle. When this message is received the host may initiate communications.

Response:
0x01 0x01 0x80 0x00 0x82

0x81 – VEHICLE_NOT_DETECTED

Description:
The OBDII Streamer will respond to the host with a VEHICLE_NOT_DETECTED message during startup. This command indicates that the OBDII connection is not fully configured yet. Since configuration is automatic, no action needs to be taken to configure the bus.

Startup configuration can take up to 1 minute after turning the key on for some vehicles. When configuration is complete a DEVICE_CONFIGURED message will be sent to the host.

Response:
0x01 0x01 0x81 0x00 0x83

0xA3 – Ignition/OBDII Status

Command Byte: 0xA3

Description:
This message will be returned when the status of the vehicle ignition and the secondary OBDII tool status changes. The format of this message is the same as found in the Ignition Status command.

This message can be disabled using the SET_CONFIGURATION command.

Response:
0x01 0x01 0xA3 0x02 IGNITION OBDII_STAT CS

Field Description:

IGNITION This is a 1 byte field that indicates the status of the vehicle ignition.

0x00 = Ignition OFF
0x01 = Ignition ON

The method used to determine the ignition status detects when the OBDII bus is responding to requests and when the RPM is not zero.

The ignition will transition from off to on when engine speed is not zero. The ignition

It is possible that some vehicles will continue to respond while the ignition is off. All vehicles are required to communicate when the key is in the on position and the engine is not running.

OBDII_STAT This is a 1 byte field that indicates if a second scan tool is connected to the OBDII port.

0x00 = No Tool
0x01 = Tool Connected

CS Summation checksum

Example:
RX:
0x01 0x01 0xA3 0x02 0x01 0x00 0xA8

This message indicates that the ignition is on and no secondary tool is connected.

0xC0 – Time Based Update Message

Description:
This message will be sent for each enabled parameter after the specified interval. If multiple parameters have time based updates enabled and timeout at the same interval the responses will be combined.

Response:
0x01 0x01 0xC0 DL PARAMETER_DATA CS

Field Description:

DL	Data Length Byte
PARAMETER_DATA	The PARAMETER_DATA field is a variable length data field that contains both the requested parameter identifiers and the parameter data. All parameter data is two bytes long, and each parameter identifier is one byte long. The format of the data field is parameter identifier, Data 1, Data 2, Repeat.
CS	Summation checksum

Example:
RX:
0x01 0x01 0xC0 0x09 0x01 0x01 0x23 0x02 0x98 0x76 0x03 0x55 0x66 0xBE

The breakdown of the response is:

Parameter 0x01:	Data = 0x0123
Parameter 0x02:	Data = 0x9876
Parameter 0x03:	Data = 0x5566

0xC1 – Threshold Based Update Message

Description:
This message will be sent when threshold based updating is enabled for a parameter and the parameter data breaches the threshold value.

Response:
0x01 0x01 0xC1 DL PARAMETER_DATA CS

Field Description:

DL	Data Length Byte
PARAMETER_DATA	The PARAMETER_DATA field is a 3 byte data field that contains both the requested parameter identifier and the parameter data. The format of the data field is parameter identifer, Data 1, Data 2.
CS	Summation checksum

Example:
RX:
0x01 0x01 0xC1 0x03 0x01 0x01 0x23 0xEB

The breakdown of the response is:

Parameter 0x01: Data = 0x0123

0xC2 – Full Speed Update Message

Description:
This message will be sent when an enabled parameter that is setup for full speed updates has been updated from the vehicle.

Response:
0x01 0x01 0xC2 DL PARAMETER_DATA CS

Field Description:

DL — Data Length Byte

PARAMETER_DATA — The PARAMETER_DATA field is a 3 byte data field that contains both the requested parameter identifier and the parameter data. The format of the data field is parameter identifer, Data 1, Data 2.

CS — Summation checksum

Example:
RX:
0x01 0x01 0xC2 0x03 0x01 0x01 0x23 0xEB

The breakdown of the response is:

Parameter 0x01: Data = 0x0123

0xD0 – Ignition Off Status Message

Description:
This message will be sent when parameter information is requested from the OBDII Streamer that cannot be updated because the ignition is off.

Response:
0x01 0x01 0xD0 0x00 CS

Field Description:

CS	Summation checksum

Example:
RX:
0x01 0x01 0xD0 0x00 0xD2

0xFF – Error Message

Command Byte: 0xFF

Description:
An error message will be sent to the host any time an improperly formatted message has been received.

Response:
0x01 0x02 0xFF ERRORCODE 0x00 CS

Field Description:

ERRORCODE This byte indicates why the error message is being sent.
0x00 = Incorrect Checksum
0x01 = Invalid Command
0x02 = Invalid Start of Frame
0x03 = Command Parameters out of Range
0x04 = Incorrect Number of bytes in the Message
0x05 = No Communications with OBDII Interface.

CS Summation checksum

Glossary

DTC Diagnostic Trouble Code – This is a standardized code that will give an indication of detected malfunctions with a vehicle. See SAE J2012 for more information.

MIL Malfunction Indicator Light – Same as a check engine light.

OBDII On Board Diagnostics Generation 2

VIN Vehicle Identification Number