



Manual de
funcionamiento del OBD
con n-Core

v.1.0

1	Introducción.....	2
2	Cableado del Sirius OBD	3
3	Programación y configuración del Sirius radion	5
3.1	Conexión del Sirius Radion al Sirius OBD.....	5
3.2	Programación del Sirius Radion.....	5
3.3	Configuración del Sirius Radion.....	5
4	Comunicación con el vehículo	7
4.1	Breve resumen del funcionamiento del ELM327	7
4.2	Herramienta ncConfTool	8
4.2.1	Campo "Send Query"	8
4.2.2	Obd Messages.....	9
4.2.3	OBD parameter	10
4.2.4	Define query.....	11
5	Respuestas desde el OBD	12
5.1	Respuestas del dispositivo ELM327	12
5.2	Respuestas con datos desde el OBD del vehículo	13
5.2.1	Respuesta de datos del OBD en una única trama	14
5.2.2	Respuesta de datos del OBD en varias tramas.....	16

1 Introducción

En este documento se exponen los primeros pasos que se deben realizar para poder comunicarnos con cualquier vehículo, a través del OBD¹, utilizando el Sirius OBD y la plataforma n-Core.

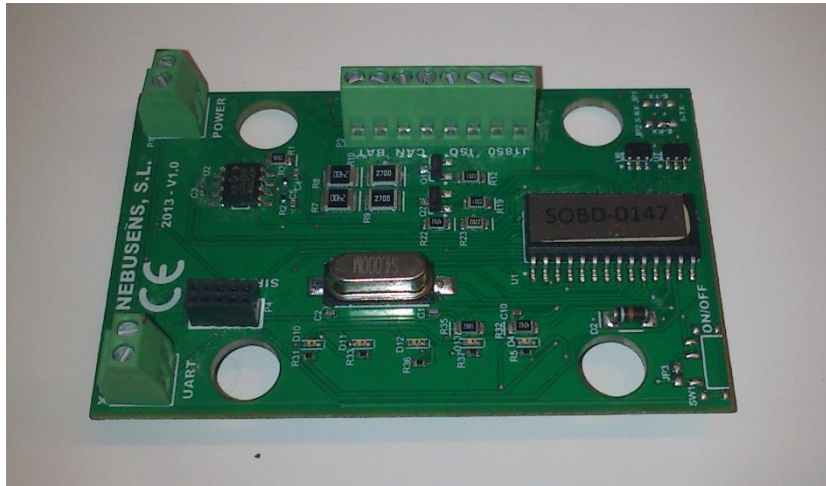


Figura 1. Sirius OBD

El Sirius OBD se caracteriza porque en su interior lleva un chip, denominado **ELM327**, que es el encargado de traducir y enviar al OBD del vehículo todos los datos que se envían a través del Sirius RadIOn, así como de enviar a este mismo dispositivo todos los datos y mensajes recibidos del OBD del vehículo.

¹ On Board Diagnostics

2 Cableado del Sirius OBD

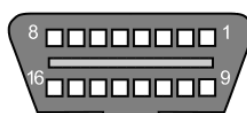
El Sirius OBD dispone de varias conexiones/enganches donde se pueden conectar cables que vayan conectados directamente al OBD del vehículo o que vayan soldados/conectados a un envoltente estándar de OBD para posteriormente conectarlo al OBD del vehículo.

En la siguiente imagen se puede ver una envoltente estándar para conectar a vehículos a través de su OBD.



Figura 2. Envoltente estándar para el OBD

El conector del envoltente estándar tiene el siguiente *pinout*:



- 5 (Signal Ground)
- 4 (Chassis Ground)
- 6 (CAN High (J-2284))
- 7 (ISO 9141-2 K Line)
- 14 (CAN Low J-2284)
- 10 (J1850 Bus-)
- 2 (J1850 Bus+)
- 15 (ISO 9141-2 L Line)
- 16 (Battery Power)

Figura 3. Pines internos conector OBD

Para poder comunicarnos con el vehículo a través del conector estándar de OBD, es necesario conectar/soldar cables en los pines que vayamos a utilizar (battery power y signal/chassis² ground son obligatorios, y luego los del protocolo o protocolos que

² Se recomienda que ambos pines estén conectados a la conexión Bat - del Sirius OBD.

vayamos a utilizar) y conectar esos cables en sus correspondientes conexiones del Sirius OBD.

En la siguiente imagen se puede ver un Sirius OBD con todos los cables conectados a sus pines correspondientes en el conector del envoltorio estándar del OBD:

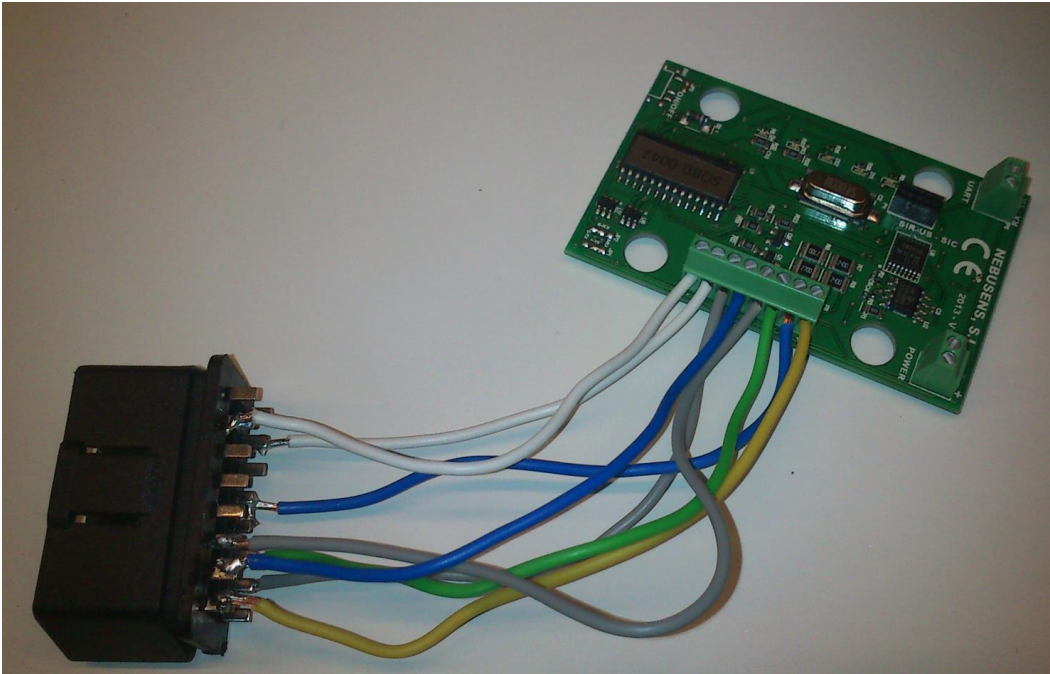


Figura 4. Conexiones entre el envoltorio estándar y el Sirius OBD

Finalmente se muestra una imagen con el Sirius OBD y un RadlOn conectado a él dentro de un envoltorio estándar:



Figura 5. Sirius OBD (con RadlOn) dentro de un envoltorio estándar

3 Programación y configuración del Sirius RadIOn

3.1 Conexión del Sirius RadIOn al Sirius OBD

Para poder hacer uso del dispositivo Sirius OBD es necesario conectar un Sirius RadIOn a dicho dispositivo.

El Sirius RadIOn se conecta a través del SIC al Sirius OBD tal y como se muestra en la siguiente imagen:

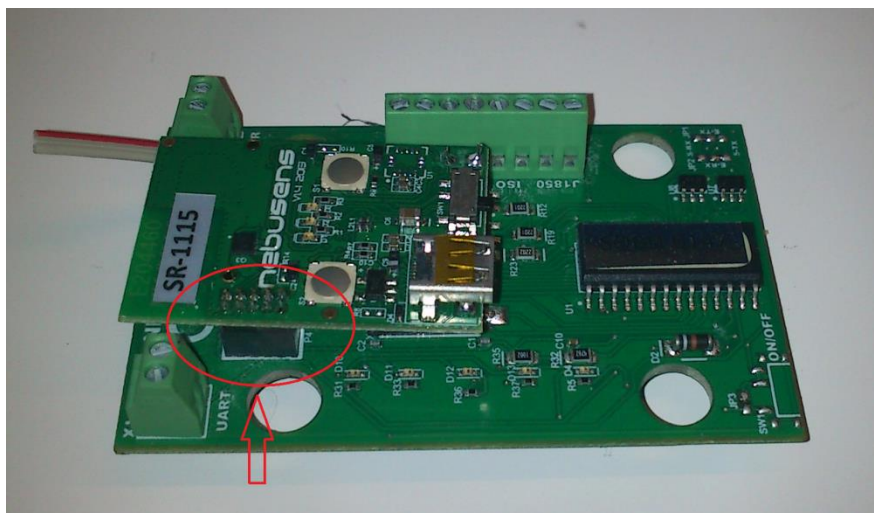


Figura 6. Sirius OBD y Sirius RadIOn

3.2 Programación del Sirius RadIOn

Para poder comunicarnos a través del Sirius OBD mediante n-Core, el Sirius RadIOn que va conectado al Sirius OBD debe tener cargado un determinado firmware:

- ncFirm_2_4_SiriusQ_R_v_1_4_OBD_v_x_x_x_x.srec³
- ncFirm_2_4_SiriusQ_R_OBD_v_x_x_x_x.srec⁴

3.3 Configuración del Sirius RadIOn

El Sirius RadIOn conectado al Sirius OBD debe tener configurado el parámetro "Operation mode" de una forma especial para que podamos comunicarnos con el Sirius OBD.

Este parámetro se encuentra en n-Core en el bloque "Kernel" y puede ser modificado a través de la herramienta **ncConfTool** en la pestaña "**Data**":

³ Los Sirius RadIOn v.1.4 son todos aquellos RadIOn que tienen un "Serial Number" con valor igual o superior a SR13240570 (etiquetados con valor igual o superior a SR-570).

⁴ Los Sirius RadIOn son todos aquellos RadIOn que tienen un "Serial Number" con valor igual o inferior a SR13240569 (etiquetados con un valor igual o inferior a SR-569).

- Parameter “**Operation mode**”:
 - Valor que hay que fijar mediante “set params”:
 - **SENT_OBD_MSG_BY_ESC_CHAR**
 - Escape char (Hex): **0x0D**

IMPORTANTE: en cuanto fijemos el valor de este parámetro (“operation mode”) al valor “Sent_OBD_msg_by_esc_char”, únicamente podremos comunicarnos con dicho dispositivo por el aire. De forma que si realizamos la configuración del Sirius RadlOn conectándolo directamente al ordenador a través de un IOn-D o de un IOn-P (v.1.4), una vez fijado el nuevo valor se perderá la comunicación, por lo que se recomienda revisar (y modificar) todos los demás parámetros del dispositivo (pan id, extended pan id, etc.) antes de de modificar este parámetro.

4 Comunicación con el vehículo

Una vez listos tanto el Sirius OBD como el Sirius RadlOn únicamente necesitamos otro dispositivo (coordinador o colector) que se comuniquen por el aire con el Sirius RadlOn que está conectado al Sirius OBD para poder comenzar a comunicarnos con el vehículo.

En este manual vamos a utilizar la herramienta ncConfTool para realizar los primeros pasos con el Sirius OBD. Para el desarrollo de aplicaciones más específicas es necesario utilizar la API de n-Core.

Como ya se ha comentado anteriormente, el Sirius OBD lleva en su interior el chip **ELM327**, por lo que para poder seguir adelante y poder comunicarnos de forma satisfactoria con el vehículo se recomienda encarecidamente leer el manual del dispositivo ELM327 (ELM327DS.pdf).

4.1 Breve resumen del funcionamiento del ELM327

El chip ELM327 se comunica con el OBD de los vehículos utilizando principalmente comandos at.

Una pequeña muestra de los "comandos at" que más se utilizan son los siguientes:

- **atz** -> resetea el dispositivo
- **atws** -> warm start (resetea el dispositivo). Es un método más rápido que el anterior.
- **atd** -> pone todos los parámetros a sus valores por defecto
- **atspX** -> permite definir el protocolo de comunicaciones que se va a utilizar en la comunicación del ELM327 con el OBD del vehículo. Se pueden fijar los siguientes protocolos:
 - 0 - Automatic
 - 1 - SAE J1850 PWM (41.6 kbaud)
 - 2 - SAE J1850 VPW (10.4 kbaud)
 - 3 - ISO 9141-2 (5 baud init, 10.4 kbaud)
 - 4 - ISO 14230-4 KWP (5 baud init, 10.4 kbaud)
 - 5 - ISO 14230-4 KWP (fast init, 10.4 kbaud)
 - 6 - ISO 15765-4 CAN (11 bit ID, 500 kbaud)
 - 7 - ISO 15765-4 CAN (29 bit ID, 500 kbaud)
 - 8 - ISO 15765-4 CAN (11 bit ID, 250 kbaud)
 - 9 - ISO 15765-4 CAN (29 bit ID, 250 kbaud)
 - A - SAE J1939 CAN (29 bit ID, 250 kbaud)
 - B - USER1 CAN (11 bit ID, 125 kbaud) -> user adjustable
 - C - USER2 CAN (11 bit ID, 50 kbaud) -> user adjustable
- **atsX** -> permite definir si en las respuestas que da el OBD del vehículo aparecen o no espacios entre bytes

- **ateX** -> permite definir si en la respuesta del OBD del vehículo se hace eco de la petición que se ha realizado.
- **atshXXX** -> permite definir la cabecera con la que se enviarán los comandos al OBD.
- **athX** -> permite definir si en las respuestas que da el OBD del vehículo se muestran o no las cabeceras de los comandos.

El método general del funcionamiento con el OBD es el siguiente:

- Configurar el dispositivo mediante, por ejemplo, los siguientes comandos at:
 - atws
 - atd
 - atsX
 - ateX
 - athX
 - atspX
- En función del modo de comunicación que se quiera llevar a cabo:
 - Método general:
 - Fijar la cabecera del comando que se va a enviar:
 - atshXXX
 - Enviar el comando
 - XXXX
 - PID (parameter identification)
 - Este método viene explicado en la página 29 del documento ELM327DS.pdf.

4.2 Herramienta ncConfTool

La herramienta de configuración ncConfTool dispone de una pestaña denominada "OBD" que tiene distintos campos con los que podemos comunicarnos con el OBD:

- Send query
- Define query
- OBD messages
- OBD parameter

4.2.1 Campo "Send Query"

El campo "Send query" se puede utilizar para enviar cualquier tipo de query o petición al Sirius OBD.

En este campo se puede enviar cualquier texto (ascii) que se desee. Es importante señalar que en el campo "Escape char" siempre tiene que estar el carácter "0D" cuando enviemos cualquier petición pulsando el botón "Set params".

En las siguientes figuras se pueden ver algunos ejemplos de peticiones utilizando el campo "Send Query".



Figura 7. Ejemplo de Send Query – atSP6

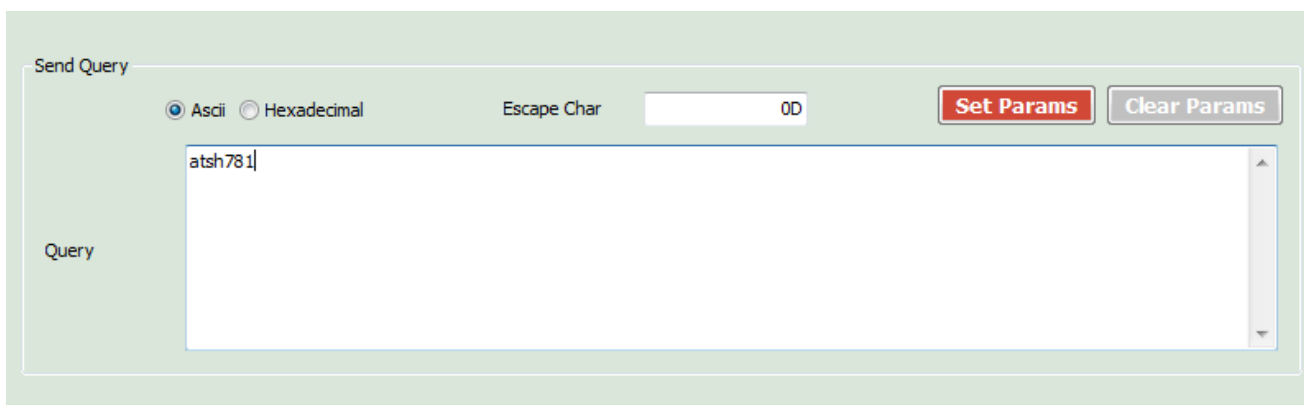


Figura 8. Ejemplo de Send Query – atSH781



Figura 9. Ejemplo de Send Query - 2101

4.2.2 OBD Messages

El campo "OBD Messages" permite ver las respuestas que da el OBD a las peticiones que se le van realizando. Para ello únicamente tenemos que seleccionar la opción "Register messages".

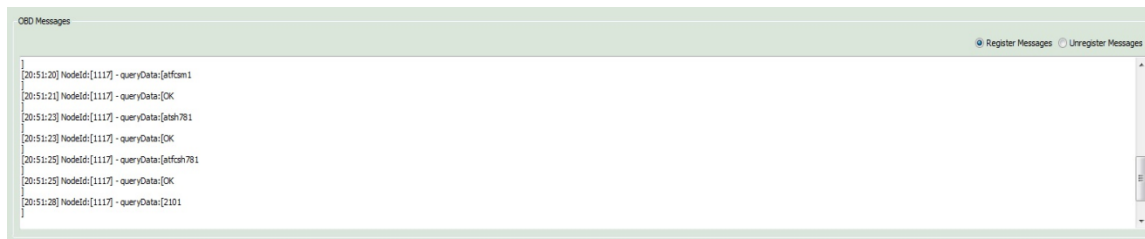


Figura 10. Ejemplo de Mensajes recibidos desde el OBD

4.2.3 OBD parameter

En este campo se puede definir el denominado "OBD parameter".



Figura 11. OBD parameter

El **OBD parameter** es un parámetro que permite que el Sirius RadIOn conectado al Sirius OBD realice de forma periódica una rutina en la que enviará, de forma autónoma, una serie de peticiones al OBD de forma secuencial.

El OBD parameter consta de los siguientes campos:

- Routine enabled? -> permite activar/desactivar la rutina de peticiones en el Sirius RadIOn
- Spaces enabled? -> envía el comando ats0 o ats1 en función de si deshabilitamos o habilitamos los espacios entre bytes en las respuestas dadas por el OBD respectivamente.
- Headers enabled? -> envía el comando ath0 o ath1 en función de si deshabilitamos o habilitamos, respectivamente, que en las respuestas del OBD se incluya también la cabecera.
- Echo enabled? -> envía el comando ate0 o ate1 en función de si deshabilitamos o habilitamos, respectivamente, que en el OBD envíe como respuesta un eco de la petición que se le realiza.
- Set to default? -> envía el comando atd al OBD para que ponga todos sus parámetros a valores por defecto.
- Reset devices? -> envía el comando atws al OBD para que se resetee
- OBD protocol -> envía el comando atspX al OBD para fijar el protocolo de comunicaciones.
- Period -> permite definir el periodo, en milisegundos, con el que se realizará la rutina autónoma. Es recomendable definir tiempos altos, como p. ej. 60000 ms.
- MSB -> Most significant Byte

Si está activada la rutina, cada vez que ésta se realice, el Sirius RadlOn va a enviar al OBD todos los comandos explicados anteriormente, teniendo en cuenta cómo estén definidos en el OBD parameter, y en el siguiente orden:

1. atws -> Si está definido a true el campo "Reset devices?"
2. atd -> Si está definido a true el campo "Set to default?"
3. ateX -> definido en "Echo enabled?"
4. atspX -> definido en "OBD protocol"
5. athX -> definido en "Headers enabled?"
6. atsX -> definido en "Spaces enabled?"

A continuación, una vez enviados todos estos comandos el RadlOn continuará la rutina enviando las "query" que tenga almacenadas en el dispositivo (definidas por el usuario).

Las "query" que se pueden almacenar en dispositivo, únicamente pueden tener el siguiente formato:

- Header (3 números): XXX
- Code (4 números): XXXX

Un ejemplo de "query" sería:

- Header: 781
- Code: 2101

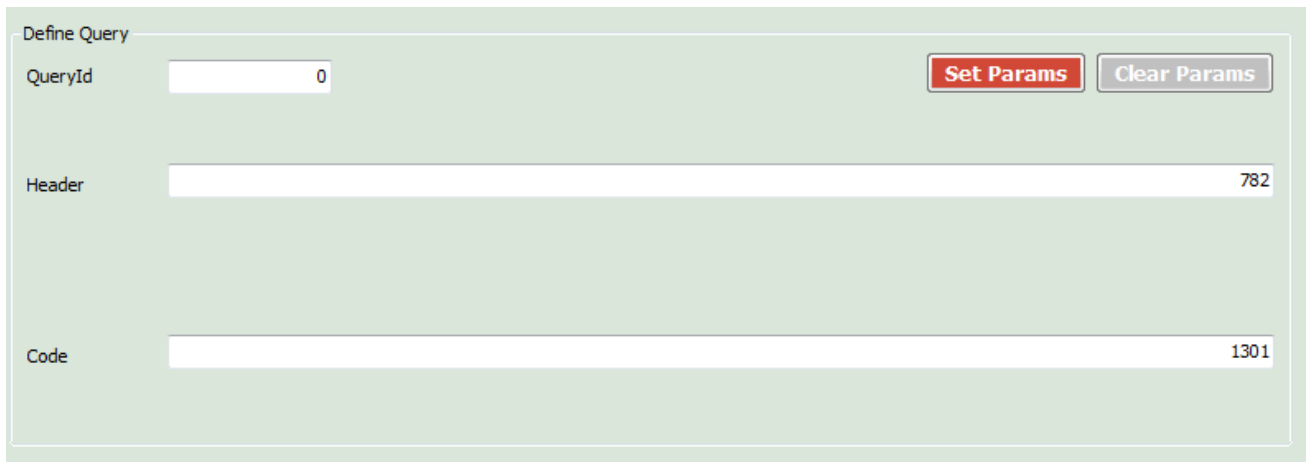
El dispositivo enviará las peticiones almacenadas de la siguiente manera:

- atshXXX -> comando at para fijar una determinada cabecera
- XXXX -> envía el código con la petición

Destacar que si la siguiente petición almacenada tiene la misma cabecera que la anterior entonces el dispositivo no enviará el comando atshXXX puesto que ya está fijada la cabecera.

4.2.4 Define query

En este campo el usuario puede definir las peticiones (query) que desea que el RadlOn realice al OBD de forma periódica.



Define Query

QueryId

Header

Code

Figura 12. Define query

Como ya se ha comentado anteriormente, las peticiones que se pueden almacenar en el dispositivo deben estar compuestas por una cabecera (3 dígitos) y un código (4 dígitos).

Actualmente (firmware versión v.2.1.15.0), el número máximo de peticiones que se puede almacenar en el dispositivo es de 15.

5 Respuestas desde el OBD

Se van a recibir dos tipos de respuestas desde el Sirius OBD:

- Respuestas del dispositivo ELM327
- Respuestas con datos desde el OBD del vehículo

5.1 Respuestas del dispositivo ELM327

El dispositivo ELM327 nos informa mediante ciertos mensajes sobre si la comunicación se ha realizado correctamente o se ha producido algún error.

En la página 72 del documento ELM327DS.pdf se muestran la mayoría de mensajes de error que puede enviar este dispositivo.

A continuación se explican brevemente algunos de los mensajes que puede enviar el ELM327:

- "OK" -> si se ha realizado con éxito la operación del comando at enviado
- "NO DATA" -> si ha expirado el *timer* que tiene el ELM327 para recibir respuesta del OBD a una petición realizada.
- "UNABLE TO CONNECT" -> suele aparecer cuando se envía el comando at "atSPX" para definir el protocolo de comunicaciones con el OBD del vehículo y el ELM327 no consigue comunicarse él utilizando ese protocolo.

5.2 Respuestas con datos desde el OBD del vehículo

Desde el OBD del vehículo nos llegarán respuestas con los datos que hayamos solicitado mediante nuestra petición.

Tanto las peticiones como las respuestas del OBD suelen seguir un determinado patrón, aunque en muchos casos no es así.

Aquí se van a mostrar brevemente los 2 casos más generales.

5.2.1 Respuesta de datos del OBD en una única trama

A continuación se muestran dos ejemplos:

Ejemplo 1

- Código pedido (Request):
 - **atsh781** (Cabecera 781)
 - **2184** (Comando de petición – Request)
- Respuesta (Response):

- **7910461840D00**
- Se desglosa en:
- 791 04

61 84 0D 00

Datos obtenidos del OBD para ese comando (**2 bytes**)

Comando de respuesta-Response (0x21 + 0x40 = 0x61) (**2 bytes**)

Información enviada como respuesta por el OBD (**4 bytes**) -> **Comando respuesta + datos**



Tamaño de bytes que hay en la trama a partir de este byte (4)

Cabecera de respuesta (791)

- Traducción de los datos obtenidos:
 - Se han recibido ->"0D 00"
 - Ordenamos los datos 00 0D (hex)
 - Estos datos están en hexadecimal
 - Se pasan a decimal -> 13⁵

⁵ Esta petición se corresponde con la temperatura del motor en una máquina -> 13 grados

Ejemplo 2

- Código pedido (Request):
 - **atsh784** (Cabecera 784)
 - **2114** (Comando de petición – Request)
- Respuesta (Response):

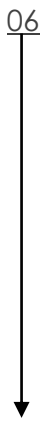
- **79406611420555A00**
- Se desglosa en:
- 794 06

61 14 20 55 5A 00

Datos obtenidos del OBD para ese comando (**4 bytes**)

Comando de respuesta-Response (0x21 + 0x40 = 0x61) (**2 bytes**)

Información enviada como respuesta por el OBD (**6 bytes**) -> **Comando respuesta + datos**



Tamaño de bytes que hay en la trama a partir de este byte (6)

Cabecera de respuesta (791)

- Traducción de los datos obtenidos:
 - Se han recibido ->"20 55 5A 00"
 - Ordenamos los datos 00 5A 55 20 (hex)
 - Estos datos están en hexadecimal
 - Los pasamos a decimal -> 5920032 ⁶

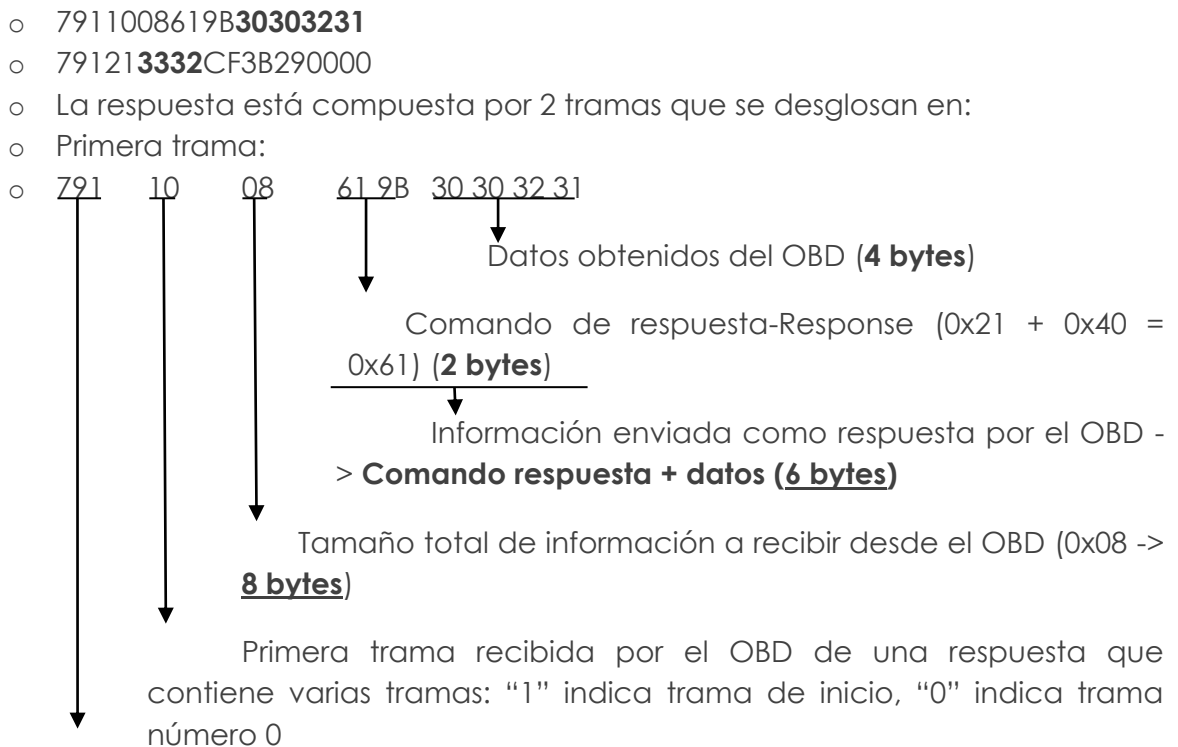
⁶ Este valor se corresponde con el tiempo de trabajo de una máquina -> 5920032 segundos

5.2.2 Respuesta de datos del OBD en varias tramas

A continuación se muestran un par de ejemplos:

Ejemplo 1

- Código pedido (Request):
 - **atsh781** (Cabecera 781)
 - **219B** (Comando de petición – Request)
- Respuesta (Response)(2 tramas):



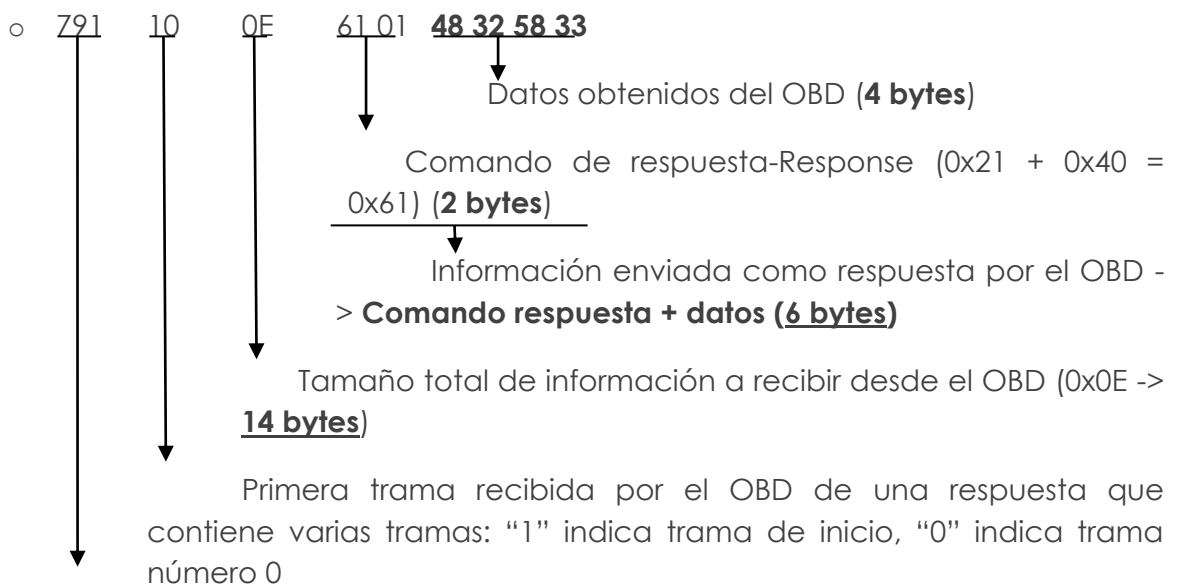
Cabecera de respuesta (791)

- Segunda trama:
 - 791 21 3332 CF3B290000
-
- ↓
 Datos obtenidos del OBD (**2 bytes**)
 ↓
 Trama de un comando multitrama: "2" indica trama de
 continuación, "1" indica trama número 1

Cabecera de respuesta (791)

Ejemplo 2

- Código pedido (Request):
 - **atsh781** (Cabecera 781)
 - **2101** (Comando de petición – Request)
- Respuesta (Response)(3 tramas):
 - 791100E6101**48325833**
 - 79121**38365A30303931**
 - 79122**36**3D010000A10E
 - La respuesta está compuesta por 3 tramas:
 - Primera trama:



Cabecera de respuesta (791)

- Segunda trama:
 - **791 21 38 36 5A 30 30 39 31**
-
- Datos obtenidos del OBD (7 bytes)
- Trama de un comando multitrama: "2" indica trama de continuación, "1" indica trama número 1

Cabecera de respuesta (791)

- Tercera trama:
 - **791 22 36 3D010000A10E**
-
- Datos obtenidos del OBD (1 byte)
- Trama de un comando multitrama: "2" indica trama de continuación, "2" indica trama número 2

Cabecera de respuesta (791)



www.nebusens.com