

---

# CAB500 CAN MODIFICATION

## USER GUIDE

---



## Table of Contents:

1.	Introduction .....	3
2.	ECUReset Service : .....	4
2.1	ECUReset Service Description : .....	4
2.2	ECUReset Service example: .....	5
3.	ReadDataByIdentifier Service: .....	6
3.1	ReadDataByIdentifier Service for “CAN ID” subfunction: .....	6
3.1.1	ReadDataByIdentifier Service request .....	6
3.2	ReadDataByIdentifier Service response: .....	6
3.3	ReadDataByIdentifier Service example for “CAN ID” sub-function:.....	8
3.4	ReadDataByIdentifier Service description for “filter frequency” sub-function: .....	9
3.4.1	ReadDataByIdentifier Service request .....	9
3.4.2	ReadDataByIdentifier Service response : .....	9
3.5	ReadDataByIdentifier Service example for “filter frequency” sub-function :.....	10
3.6	ReadDataByIdentifier Service description for “CAN speed” sub-function:.....	12
3.6.1	ReadDataByIdentifier Service request .....	12
3.6.2	ReadDataByIdentifier Service response .....	12
3.7	ReadDataByIdentifier Service example for “CAN speed” sub-function:.....	13
3.8	ReadDataByIdentifier Service description for “CAN CAB500_Ip frame period” sub-function: 15	
3.8.1	ReadDataByIdentifier Service request .....	14
3.8.2	ReadDataByIdentifier Service response .....	14
3.9	ReadDataByIdentifier Service example for “CAN_Ip frame period” sub-function: .....	15
4.	WriteDataByIdentifier Service : .....	16
4.1	WriteDataByIdentifier Service description for “CAN ID” sub-function: .....	16
4.2	WriteDataByIdentifier Service example for “CAN ID” sub-function:.....	19
4.3	WriteDataByIdentifier Service description for “filter frequency” sub-function:.....	20
4.4	WriteDataByIdentifier Service example for “filter frequency” sub-function:.....	21
4.5	WriteDataByIdentifier Service description for “CAN speed” sub-function:.....	21
4.6	WriteDataByIdentifier Service example for “CAN speed” sub-function :.....	22
4.7	WriteDataByIdentifier Service description for “CAB500_Ip frame period” sub-function: ...	23
4.8	Write Data by Identifier Service example for “CAB500_Ip frame period” sub-function:....	24
 <b>Table 1 : Filter frequencies values versus CAN frame values .....</b>		<b>11</b>
<b>Table 2 : CAN speed parameters versus CAN frame values .....</b>		<b>13</b>
<b>Table 3 : CAB500_Ip frame period versus CAN frame values.....</b>		<b>15</b>

# 1. Introduction

This documentation allows modifying the parameters inside the CAB500-C-SP5 for all versions superior to PR06.

**The customer shall not be used the CAN frame ID 0x68d in his CAN network. The ID 0x3C2 and 0x68d will be booked for CAB500 sensor by default.**

The parameters modifiable are:

- \* Sensor IDs (CAB500\_Ip frame ID, UDS\_CLIENT frame ID and UDS\_SERVER frame ID)
- \* IIR Filters frequency used to measure the current
- \* CAN speed
- \* Period of the CAB500\_Ip frame (current frame)

The communication protocol uses:

- \* DoCAN ISO 15765 (2011)
- \* UDS ISO 14229 (2013)

To execute these modifications you need a CAN 2.0 interface which is able to send a CAN frame. (example: PEAK, CANalyzer, MuxyLigth, Ni XNet...). This interface should be connected to the sensor with CANHigh and CANLow pins (See the datasheet). This interface is named “external application “ in all followed paragraphs.

If you want to read a parameter saved in the sensor, use the ReadDataByIdentifier Service. If you want to modify a parameter in the sensor, use the WriteDataByIdentifier Service then execute an ECUReset service in order to update this new function.

The default parameters are :

- Sensors ID:

Frame Name	CAB500_Ip frame	UDS_CLIENT	UDS_SERVER
Default CAN ID	0x3C2	0x68D	0x68E

- Filter frequency : the sensor does not have a IIR filter, it computes an average on 10ms
- CAN speed : 500kbauds
- Period of the CAB500\_Ip frame is 10ms

The IDs ranges available are between 0x221 to 0x3FF and between 0x40B to 0x7F0.

# 1. ECUReset Service :

## 1.1 ECUReset Service Description :

ECUReset service with two sub-functions:

- HardReset
- SoftReset

Message Description	CAN ID	Name	Data Length (Nb bytes)	Message launch type	Signal Description	Signal Values	Start Bit	End Bit
ECUReset service request to reset the software	0x68D	UDS_CLIENT	3	no cyclic message, send by the external application	Single frame composed by 2 data	0x02	0	7
					ECUReset service	0x11	8	15
					Sub-functions available : - hardReset - SoftReset -	0x01 0x03 Other : no action	16	31

Message Description	CAN ID	Name	Data Length (Nb bytes)	Message launch type	Signal Description	Signal Values	Start Bit	End Bit
ECUReset service response to reset the software	0x68E	UDS_SERVER	8	no cyclic message, send by the CAB500 sensor	Single frame composed by 2 data	0x02	0	7
					ECUReset service : - positive response - negative response	0x51 0x7F	8	15
					Sub-functions available : - hardReset -SoftReset -	0x01 0x03 Other : no action	16	23
					Unused	0x00	24	56

## 1.2 ECUReset Service example:

The external application is the client and wants to reset the sensor with the HardReset sub-function: 0x01.

545.758568	1	68D	UDS_CLIENT_1	Rx	3	02 11 01
545.758990	1	68E	UDS_SERVER_1	Rx	8	02 51 01 00 00 00 00 00

Figure 1 : ECUReset Service frames to reset the sensor

This external application send the UDS\_CLIENT (ID= 0x68D) frame in which :

- 0x02 defined a single frame composed of 2 usable data
- 0x11 defined the ECUReset service
- 0x01 defined the HardReset sub-function.

The sensor response with the UDS\_SERVER (ID = 0x68E) frame in which :

- 0x02 defined a single frame composed of 2 usable data
- 0x51 defined the positive response of ECUReset service
- 0x01 defined the HardReset sub-function.

## 2. ReadDataByIdentifier Service:

ReadDataByIdentifier Service with three sub-functions:

- CAN ID : 0xF010
- Filter frequency : 0xF011
- CAN Speed : 0xF012
- CAN frame period : 0xF013

### 2.1 ReadDataByIdentifier Service for “CAN ID” subfunction:

#### 2.1.1 ReadDataByIdentifier Service request

For all these sub-functions, the external application sends this request frame:

Message Description	CAN ID	Name	Data Length (Nb bytes)	Message launch type	Signal Description	Signal Value	Start Bit	End Bit
ReadDataBy Identifier service request to read parameter of the software	0x68D	UDS_CLIENT	4	no cyclic message, send by the external application	Single frame composed by 3 data	0x03	0	7
					ReadDataByIdentifier service	0x22	8	15
					Sub-functions available : - CAN ID	0xF010	16	31

### 2.2 ReadDataByIdentifier Service response:

For the CAN ID sub-function response, the process of 2 consecutive frames is used :

- First consecutive frame
- Flow control frame
- Second consecutive frame

The first consecutive frame is composed of:

Message Description	CAN ID	Name	Data Length (Nb bytes)	Message launch type	Signal Description	Signal Values	Start Bit	End Bit
ReadData ByIdentifier service response to read CAN ID sub-function	0x68E	UDS_SERVER	8	no cyclic message, send by the sensor	<b>First consecutive frame</b>	0x10	0	7
					composed of 9 usable data	0x09	8	15
					ReadDataByIdentifier service :			
					- positive response	0x62	16	23
					- negative response	0x7F		
					Sub- function : CAN ID	0xF010	24	39
CAB500_Ip frame ID	0x3C2 (Default values)	40	55					
MSB byte of UDS_CLIENT frame ID	0x06 (Default values)	56	63					

The flow control frame sends by external application to validate the reception of the first consecutive frame receiving:

Message Description	CAN ID	Name	Data Length (Nb bytes)	Message launch type	Signal Description	Signal Value	Start Bit	End Bit
ReadData ByIdentifier service flow control frame to acknowledge the first consecutive frame receiving (CAN ID sub-function)	0x68D	UDS_CLIENT	3	no cyclic message, send by the external application	<b>Flow control frame and FS</b>	0x30	0	7
					BS (no managed)	0	8	15
					STmin (no managed)	0	16	23

The second consecutive frame is composed of:

Message Description	CAN ID	Name	Data Length (Nb bytes)	Message launch type	Signal Description	Signal Values	Start Bit	End Bit
ReadDataByIdentifier service response to read CAN ID	0x68E	UDS_SERVER	8	no cyclic message, send by the sensor	Second consecutive frame	0x21	0	7
					LSB byte of UDS_CLIENT frame ID	0x8D (Default values)	8	15
					UDS_SERVER frame ID	0x068E (Default values)	16	31
					Unused	0x00	32	63

**Note :** Please send the first and second consecutive frames without waiting the flow control frame (bug known).

### 2.3 ReadDataByIdentifier Service example for “CAN ID” sub-function:

The external application is the client and wants to read the sensor identifiers (CAN ID).

2385.811947	1	68D	UDS_CLIENT_1	Rx	4	03 22 F0 10
2385.812429	1	68E	UDS_SERVER_1	Rx	8	10 09 62 F0 10 03 C2 06
2385.813221	1	68D	UDS_CLIENT_1	Rx	3	30 00 00
2385.813662	1	68E	UDS_SERVER_1	Rx	8	21 8D 06 8E 00 00 00 00

Figure 2 : ReadDataByIdentifier service frames to read CAN frames identifiers

This external application send the UDS\_CLIENT (ID= 0x68D) frame in which :

- 0x03 defined a single frame composed of 3 usable data
- 0x22 defined the ReadDataByIdentifier service
- 0xF010 defined the Sensor frames identifiers

The sensor response with the UDS\_SERVER (ID = 0x68E) frame in which :

- 0x10 followed by 0x09 defined a first consecutive frame composed of 9 usable data
- 0x62 defined the positive response of ReadDataByIdentifier service
- 0xF010 defined the Sensor identifiers sub-function
- 0x03 followed by 0xC2 defined the CAB500\_Ip frame ID : 0x3C2
- 0x06 is the first highest byte of the UDS\_CLIENT ID.

This external application send the UDS\_CLIENT (ID= 0x68D) frame in which :

- 0x30 defined the flow control frame in response to the first consecutive frame (BS and STmin are not managed)

The sensor response with the UDS\_SERVER (ID = 0x68E) frame in which :

- 0x21 the second consecutive frame

- 0x8D is the lowest byte of the Client request identifier. So the UDS\_CLIENT identifier is equal to 0x68D
- 0x06 followed by 0x8E defined the Server response ID ( UDS\_SERVER= 0x68E).

## 2.4 ReadDataByIdentifier Service description for “filter frequency” sub-function:

### 2.4.1 ReadDataByIdentifier Service request

For all these sub-functions, the external application sends this request frame:

Message Description	CAN ID	Name	Data Length (Nb bytes)	Message launch type	Signal Description	Signal Value	Start Bit	End Bit
ReadDataBy Identifier service request to read parameter of the software	0x68D	UDS_CLIENT	4	no cyclic message, send by the external application	Single frame composed by 3 data	0x03	0	7
					ReadDataByIdentifier service	0x22	8	15
					Sub-functions available : - Filter frequency	0xF011	16	31

### 2.4.2 ReadDataByIdentifier Service response :

For the filter frequency sub-function response, the process is just composed of 1 single response frame:

Message Description	CAN ID	Name	Data Length (Nb bytes)	Message launch type	Signal Description	Signal Values	Start Bit	End Bit	
ReadDataBy Identifier service response to read the filter frequency	0x68E	UDS_SERVER	8	no cyclic message, send by the sensor	Single frame composed of 4 data	0x04	0	7	
					ReadDataBy Identifier service:				
					- positive response	0x62	8	15	
					- negative response	0x7F			
					Sub-function : filter frequency	0xF011	16	31	
filter frequency value	See the Table1	32	39						
Unused	0x00	39	63						

## 2.5 ReadDataByIdentifier Service example for “filter frequency” sub-function :

An external application is the client and wants to read the filter frequency.

```

237.917953  1  68D  UDS_CLIENT_1  Rx  4  03 22 F0 11
237.918393  1  68E  UDS_SERVER_1  Rx  8  04 62 F0 11 01 00 00 00
  
```

Figure 3 : ReadDataByIdentifier service frames to read the filter frequency

This external application send the UDS\_CLIENT (ID= 0x68D) frame in which :

- 0x03 defined a single frame composed of 3 usable data
- 0x22 defined the ReadDataByIdentifier service
- 0xF011 defined the filter frequency sub-function

The sensor response with the UDS\_SERVER (ID = 0x68E) frame in which :

- 0x04 defined a single frame composed of 4 usable data
- 0x62 defined the positive response of ReadDataByIdentifier service
- 0xF011 defined the filter frequency sub function
- 0x01 defined that the filter frequency is 10Hz (function = ValueReceived \*10 )

**Table 1 : Filter frequencies values versus CAN frame values**

Frequency filter	Filter frequency value on CAN frame
10Hz	0x01
20Hz	0x02
30Hz	0x03
40Hz	0x04
50Hz	0x05
60Hz	0x06
70Hz	0x07
80Hz	0x08
90Hz	0x09
100Hz	0x0A
110Hz	0x0B
120Hz	0x0C
130Hz	0x0D
140Hz	0x0E
150Hz	0x0F
160Hz	0x10
Average every 10ms whitout filter	0xFF

## 2.6 ReadDataByIdentifier Service description for “CAN speed” sub-function:

### 2.6.1 ReadDataByIdentifier Service request

For all these sub-functions, the external application sends this request frame:

Message Description	CAN ID	Name	Data Length (Nb bytes)	Message launch type	Signal Description	Signal Value	Start Bit	End Bit
ReadDataBy Identifier service request to read parameter of the software	0x68D	UDS_CLIENT	4	no cyclic message, send by the external application	Single frame composed by 3 data	0x03	0	7
					ReadDataByIdentifier service	0x22	8	15
					Sub-functions available :  - CAN Speed	0xF012	16	31

### 2.6.2 ReadDataByIdentifier Service response

For the CAN speed sub-function response, the process is just composed of 1 single response frame:

Message Description	CAN ID	Name	Data Length (Nb bytes)	Message launch type	Signal Description	Signal Values	Start Bit	End Bit
ReadDataBy Identifier service response to read the CAN speed	0x68E	UDS_SERVER	8	no cyclic message, send by the sensor	Single frame composed by 5 data	0x05	0	7
					ReadDataBy Identifier service :			
					- positive response	0x62	8	15
					- negative response	0x7F		
					Sub-function : CAN speed	0xF012	16	31
CAN speed value	See Table 2	32	47					
Unused	0x00	48	63					

**Table 2 : CAN speed parameters versus CAN frame values**

CAN speed	CAN speed value on CAN frame
125k	0x007D
250k	0x00FA
500k	0x01F4

## 2.7 ReadDataByIdentifer Service example for “CAN speed” sub-function:

The external application is the client and wants to read the CAN speed.

```

.....✉ 3130.012078 1 68D UDS_CLIENT_1 Rx 4 03 22 F0 12
.....✉ 3130.012483 1 68E UDS_SERVER_1 Rx 8 05 62 F0 12 00 7D 00 00
    
```

*Figure 4 : ReadDataByIdentifer service frames to read CAN speed*

This service uses the single frame to send the data.

This external application send the UDS\_CLIENT (ID= 0x68D) frame in which :

- 0x03 defined a single frame composed of 3 usable data
- 0x22 defined the ReadDataByIdentifer service
- 0xF012 defined the CAN speed sub-functions

The sensor response with the UDS\_SERVER (ID = 0x68E) frame in which :

- 0x05 defined a single frame composed of 5 usable data
- 0x62 defined the positive response of ReadDataByIdentifer service
- 0xF012 defined the CAN Speed sub function
- 0x00 followed by 0x7D defined that the CAN speed is 125kBds

## 2.8 ReadDataByIdentifier Service description for “CAN CAB500\_Ip frame period” sub-function:

### 2.8.1 ReadDataByIdentifier Service request

For all these sub-functions, the external application sends this request frame:

Message Description	CAN ID	Name	Data Length (Nb bytes)	Message launch type	Signal Description	Signal Value	Start Bit	End Bit
ReadDataBy Identifier service request to read parameter of the software	0x68D	UDS_CLIENT	4	no cyclic message, send by the external application	Single frame composed by 3 data	0x03	0	7
					ReadDataByIdentifier service	0x22	8	15
					Sub-functions available : - CAN frame period	0xF013	16	31

### 2.8.2 ReadDataByIdentifier Service response

For the CAN CAB500\_Ip frame period sub-function response, the process is just composed of 1 single response frame:

Message Description	CAN ID	Name	Data Length (Nb bytes)	Message launch type	Signal Description	Signal Values	Start Bit	End Bit
ReadDataBy Identifier service response to read the CAN CAB500_Ip frame period	0x68E	UDS_SERVER	8	no cyclic message, send by the sensor	Single frame composed by 5 data	0x05	0	7
					ReadDataBy Identifier service :			
					- positive response	0x62	8	15
					- negative response	0x7F		
					Sub-function : CAB500_Ip frame period	0xF013	16	31
CAB500_Ip frame period	See Table 3	32	47					
Unused	0x00	48	63					

**Table 3 : CAB500\_Ip frame period versus CAN frame values**

CAB500_Ip frame period	CAB500_Ip frame period on CAN frame
10ms	0x000A and all other values
100ms	0x0064

## 2.9 ReadDataByIdentifer Service example for “CAN\_Ip frame period” sub-function:

The external application is the client and wants to read the CAN CAB500\_Ip frame period.

```

... [envelope] 4479.115878 1 68D UDS_CLIENT_1 Rx 4 03 22 F0 13
... [envelope] 4479.116284 1 68E UDS_SERVER_1 Rx 8 05 62 F0 13 00 0A 00 00
  
```

Figure 5 : ReadDataByIdentifier service frames to read CAB500\_Ip frame period

This service uses the single frame to send the data.

This external application send the UDS\_CLIENT (ID= 0x68D) frame in which :

- 0x03 defined a single frame composed of 3 usable data
- 0x22 defined the ReadDataByIdentifier service
- 0xF013 defined the CAN CAB500\_ip period sub-function

The sensor response with the UDS\_SERVER (ID = 0x68E) frame in which :

- 0x05 defined a single frame composed of 5 usable data
- 0x62 defined the positive response of ReadDataByIdentifier service
- 0xF013 defined the CAN Speed sub function
- 0x00 followed by 0x7A defined that the CAN CAB500\_Ip period is the corresponding decimal value: 10ms

### 3. WriteDataByIdentifier Service :

WriteDataByIdentifier with three sub-functions:

- CAN ID : 0xF010
- Filter frequency : 0xF011
- CAN Speed : 0xF012
- CAN frame period : 0xF013

The process is specific for each sub function.

**After writing (after using this command), wait at least 1 second before turning off to avoid a memory corruption (error code =0x40).**

#### 3.1 WriteDataByIdentifier Service description for “CAN ID” sub-function:

For the CAN ID sub-function, the request is composed of :

- First consecutive frame sent by the external application
- Flow control frame sent by sensor to confirm the receiving of the previous frame
- Second consecutive frame sent by the external application

Message Description	CAN ID	Name	Data Length (Nb bytes)	Message launch type	Signal Description	Signal Value	Start Bit	End Bit
WriteData By Identifier service request to write CAN ID	0x68D	UDS_CLIENT	8	no cyclic message, send by the external application	<b>First consecutive frame</b>	0x10	0	7
					composed of 9 data	0x09	8	15
					WriteDataByIdentifier service	0x2E	16	23
					Sub function : CAN ID	0xF010	24	39
					CAB500_Ip frame ID value	0x03C2 (default value)	40	55
					MSB byte of the UDS_CLIENT frame ID value	0x06 (default value)	56	63

Message Description	CAN ID	Name	Data Length (Nb bytes)	Message launch type	Signal Description	Signal Value	Start Bit	End Bit
WriteDataBy Identifier service response to write CAN ID parameters	0x68E	UDS_SERVER	8	no cyclic message, send by the external application	Flow control frame and FS	0x30	0	7
					BS (no managed)	0x00	8	15
					STmin (no managed)	0x00	16	23
					Unused	0x00	24	63

Message Description	CAN ID	Name	Data Length (Nb bytes)	Message launch type	Signal Description	Signal Value	Start Bit	End Bit
WriteDataBy Identifier service request to write CAN ID parameters	0x68D	UDS_CLIENT	4	no cyclic message, send by the external application	Second consecutive frame	0x21	0	7
					LSB of the UDS_CLIENT frame ID value	0x8D (default value)	8	15
					UDS_SERVER frame ID value	0x068E (default value)	16	23

Message Description	CAN ID	Name	Data Length (Nb bytes)	Message launch type	Signal Description	Signal Values	Start Bit	End Bit
WriteDataBy Identifier service response to write CAN ID parameters	0x68E	UDS_SERVER	8	no cyclic message, send by the sensor	Single frame composed by 3 data	0x03	0	7
					WriteDataByIdentifier service: - positive response - negative response	0x6E 0x7F	8	15
					Sub-function : CAN ID	0xF010	16	31

					Unused	0x00	32	63
--	--	--	--	--	--------	------	----	----

### 3.2 WriteDataByIdentifer Service example for “CAN ID” sub-function:

In this first example, the external application is the client and wants to modify the sensor IDs parameters with CAB500\_IP = 0x3C0 ; UDS\_CLIENT = 0x600 and UDS\_SERVER= 0x601.

89.256766	1	68D	UDS_CLIENT_1	Rx	8	10 09 2E F0 10 03 C0 06
89.257159	1	68E	UDS_SERVER_1	Rx	8	30 00 00 00 00 00 00 00
89.257803	1	68D	UDS_CLIENT_1	Rx	4	21 00 06 01
89.289798	1	68E	UDS_SERVER_1	Rx	8	03 6E F0 10 00 00 00 00

Figure 6 : WriteDataByIdentifier service frames to modify CAN identifiers

This service uses the consecutive frame to send the data.

This external application send the UDS\_CLIENT (ID= 0x68D) frame in which :

- 0x10 followed by 0x09 defined a first frame composed of 9 usable data
- 0x2E defined the WriteDataByIdentifier service
- 0xF010 defined the Sensor ID identifiers sub-functions
- 0x03 followed by 0xC0 defined that the new CAB500\_ip frame ID is 0x3c0
- 0x06 defined the first high byte of the client request frame ID

The sensor response with the UDS\_SERVER (ID = 0x68E) frame in which :

- 0x30 defined the flow control frame which means that the sensor has read the first frame and wait the next data. (BS and STmin are not managed)

This external application send the next data with the UDS\_CLIENT (ID= 0x68D) frame in which :

- 0x21 defined that this frame is the first consecutive frame
- 0x00 is the low byte of the client request ID. So the UDS\_CLIENT frame ID is 0x600.
- 0x06 followed by 0x01 defined that the new server response frame ID is 0x601 (UDS\_SERVER)

The sensor response with the UDS\_SERVER (ID = 0x68E) frame in which :

- 0x03 defined a single frame composed of 3 usable data
- 0x6E defined the positive response of WriteDataByIdentifier service
- 0xF010 defined the Sensor ID identifier

This service modifies the EEPROM memory and this modification will be taken account after a reset.

### 3.3 WriteDataByIdentifer Service description for “filter frequency” sub-function:

For the filter frequency sub-function response, the process is just composed of one request and one response frames:

Message Description	CAN ID	Name	Data Length (Nb bytes)	Message launch type	Signal Description	Signal Values	Start Bit	End Bit
WriteDataBy Identifier service request to write filter frequency parameters	0x68D	UDS_CLIENT	5	no cyclic message, send by the external application	Single frame composed by 4 data	0x04	0	7
					WriteDataByIdentifier service	0x2E	8	15
					Sub-function : Filter frequency	0xF011	16	31
					Filter frequency value	See the Table 1	32	39

Message Description	CAN ID	Name	Data Length (Nb bytes)	Message launch type	Signal Description	Signal Values	Start Bit	End Bit
WriteDataBy Identifier service response to write filter frequency parameters	0x68E	UDS_SERVER	0	no cyclic message, send by the sensor	Single frame composed by 3 data	0x03	0	7
					WriteDataBy Identifier service:			
					- positive response	0x6E	8	15
					- negative response	0x7F		
					Sub-function : filter frequency	0xF011	16	31
					Unused	0x00	32	63

### 3.4 WriteDataByIdentifier Service example for “filter frequency” sub-function:

In this second example, the external application is the client and wants to modify the filter frequency to set 50Hz. You can see the figure 3 to have the link between frequencies and CAN frame data.

```

... ☒ 641.651253      1    68D  UDS_CLIENT_1      Rx    5    04 2E F0 11 05
... ☒ 641.662211      1    68E  UDS_SERVER_1     Rx    8    03 6E F0 11 00 00 00 00
    
```

Figure 7 : WriteDataByIdentifier service frames to modify the filter frequency

This service uses the single frame to send the data.

This external application send the UDS\_CLIENT (ID= 0x68D) frame in which :

- 0x04 defined a single frame composed of 4 usable data
- 0x2E defined the WriteDataByIdentifier service
- 0xF011 defined the filter frequency sub-functions
- 0x05 defined that the filter frequency is the corresponding hexadecimal number : 0x05 (ValueSent = frequency /10 )

The sensor response with the UDS\_SERVER (ID = 0x68E) frame in which :

- 0x03 defined a single frame composed of 3 usable data
- 0x6E defined the positive response of WriteDataByIdentifier service
- 0xF011 defined the filter frequency sub function

This service modifies the EEPROM memory and this modification will be taken account after a reset.

### 3.5 WriteDataByIdentifier Service description for “CAN speed” sub-function:

For the CAN speed sub-function response, the process is just composed of one request and one response frames:

Message Description	CAN ID	Name	Data Length (Nb bytes)	Message launch type	Signal Description	Signal Values	Start Bit	End Bit
WriteDataBy Identifier service request to write CAN speed parameters	0x68D	UDS_CLIENT	6	no cyclic message, send by the external application	Single frame composed by 4 data	0x04	0	7
					WriteDataBy Identifier service:	0x2E	8	15
					Sub-function: CAN speed	0xF012	16	31
					CAN speed value	See Table 2	32	47

### 3.6 WriteDataByIdentifier Service example for “CAN speed” sub-function :

Message Description	CAN ID	Name	Data Length (Nb bytes)	Message launch type	Signal Description	Signal Values	Start Bit	End Bit	
WriteDataBy Identifier service response to write CAN Speed parameters	0x68E	UDS_SERVER	8	no cyclic message, send by the sensor	Single frame composed by 3 data	0x03	0	7	
					WriteDataBy Identifier service:				
					- positive response	0x6E	8	15	
					- negative response	0x7F			
					Sub-function : CAN speed	0xF012	16	31	
					Unused	0x00	32	63	

In this third example, the external application is the client and wants to modify the CAN speed to set 125k/Bds .

```

1728.191803  1  68D  UDS_CLIENT_1  Rx  6  05 2E F0 12 00 7D
1728.202751  1  68E  UDS_SERVER_1  Rx  8  03 6E F0 12 00 00 00
  
```

Figure 8 : WriteDataByIdentifier service frames to modify CAN speed

This service uses the single frame to send the data.

This external application send the UDS\_CLIENT (ID= 0x68D) frame in which :

- 0x05 defined a single frame composed of 5 usable data
- 0x2E defined the WriteDataByIdentifier service
- 0xF012 defined the CAN speed sub-functions
- 0x00 followed by 0x7D defined that the speed is the corresponding decimal number : 125

The sensor response with the UDS\_SERVER (ID = 0x68E) frame in which :

- 0x03 defined a single frame composed of 3 usable data
- 0x6E defined the positive response of WriteDataByIdentifier service
- 0xF012 defined the CAN Speed sub function

This service modifies the EEPROM memory and this modification will be taken account after a reset.

### 3.7 WriteDataByIdentifier Service description for “CAB500\_Ip frame period” sub-function:

For the CAB500\_Ip frame period sub-function, the process is just composed of one request and one response frames:

Message Description	CAN ID	Name	Data Length (Nb bytes)	Message launch type	Signal Description	Signal Values	Start Bit	End Bit
WriteDataBy Identifier service request to write CAB500_Ip frame period parameters	0x68D	UDS_CLIENT	6	no cyclic message, send by the external application	Single frame composed by 5 data	0x05	0	7
					WriteDataBy Identifier service:	0x2E	8	15
					Sub-function: CAB500_Ip frame period	0xF013	16	31
					CAB500_Ip frame period value	See Table 3	32	47

Message Description	CAN ID	Name	Data Length (Nb bytes)	Message launch type	Signal Description	Signal Values	Start Bit	End Bit
WriteDataBy Identifier service response to write CAB500_Ip frame period parameters	0x68E	UDS_SERVER	8	no cyclic message, send by the sensor	Single frame composed by 3 data	0x03	0	7
					WriteDataBy Identifier service:		8	15
					- positive response	0x6E		
					- negative response	0x7F		
		Sub-function : CAB500_Ip frame period	0xF013	16	31			
		Unused	0x00	32	63			

### 3.8 Write Data by Identifier Service example for “CAN CAB500\_Ip frame period” sub-function:

In this third example, the external application is the client and wants to modify the CAN CAB500\_Ip frame period to set 100ms.

...	✉	2084.709104	1	68D	UDS_CLIENT_1	Rx	6	05 2E F0 13 00 64
...	✉	2084.720011	1	68E	UDS_SERVER_1	Rx	8	03 6E F0 13 00 00 00 00

Figure 9 : WriteDataByIdentifierservice frames to modify CAB500\_Ip frame period

This service uses the single frame to send the data.

This external application send the UDS\_CLIENT (ID= 0x68D) frame in which :

- 0x05 defined a single frame composed of 5 usable data
- 0x2E defined the WriteDataByIdentifier service
- 0xF013 defined the “CAN CAB500\_Ip frame period” sub-function
- 0x00 followed by 0x64 defined that the period is the corresponding decimal number : 100

The sensor response with the UDS\_SERVER (ID = 0x68E) frame in which :

- 0x03 defined a single frame composed of 3 usable data
- 0x6E defined the positive response of WriteDataByIdentifier service
- 0xF013 defined the “CAN CAB500\_Ip frame period” sub-function

This service modifies the EEPROM memory and this modification will be taken account after a reset.