Write Data by Identifier

Purpose: "Purpose : Write data into server using Identifier"

Introduction

- ✓ The Write Data By Identifier service is used to write some information into the ECU at an **internal location** specified by the provided **data identifier** (DID).
- ✓ The written data record can be identified by a data Identifier that may or may not follow security algorithm for the data record **(Security Access (0x27)** may or may not be included as prior service).
- ✓ Dynamically defined data Identifiers (**service)** will not be used with this service.
- ✓ This is **vehicle manufacturer's** constraints that the server conditions are met when performing this service**. (NRC 22)**

Sub-functions

No sub-functions

What can be written? What are the limitations?

- **Configuration information can be written into the ECU (e.g.Part number, Hardware number, SW number etc.)**
- **Erasing NVM Data**
- **Resetting calibration values or Learned values**
- **Setting option content**

- **Not all the identifiers are re-writable; some may be read only (as defined by the system supplier/vehicle manufacturer for read-only identifiers, etc.).**

Request Frame:

1. **Service Id**
2. **Data Identifier**
3. **Data**

Positive Response Frame:

1. **Service Id**
2. **Data Identifier**

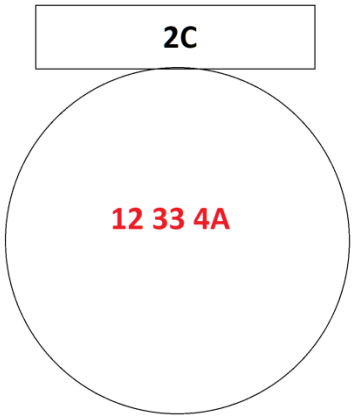
Negative Response Frame:

1. **Negative Response (7F)**
2. **Service Id**
3. **NRC Code**

Under circumstances that The ECU will resume to set the trouble code in to the server's memory

1. **Sub-function (DTC Setting Type : Resume)**
2. **ECU Reset**
3. **Session transition where Service (0x85) is not supported**
4. **Clear DTC Information**

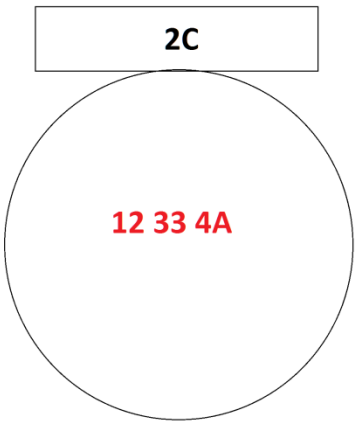
Understanding on Control DTC Settings !!



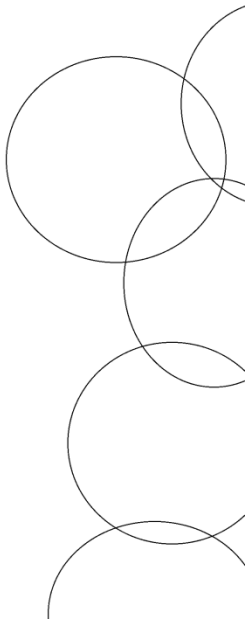
2 C
0010 1100

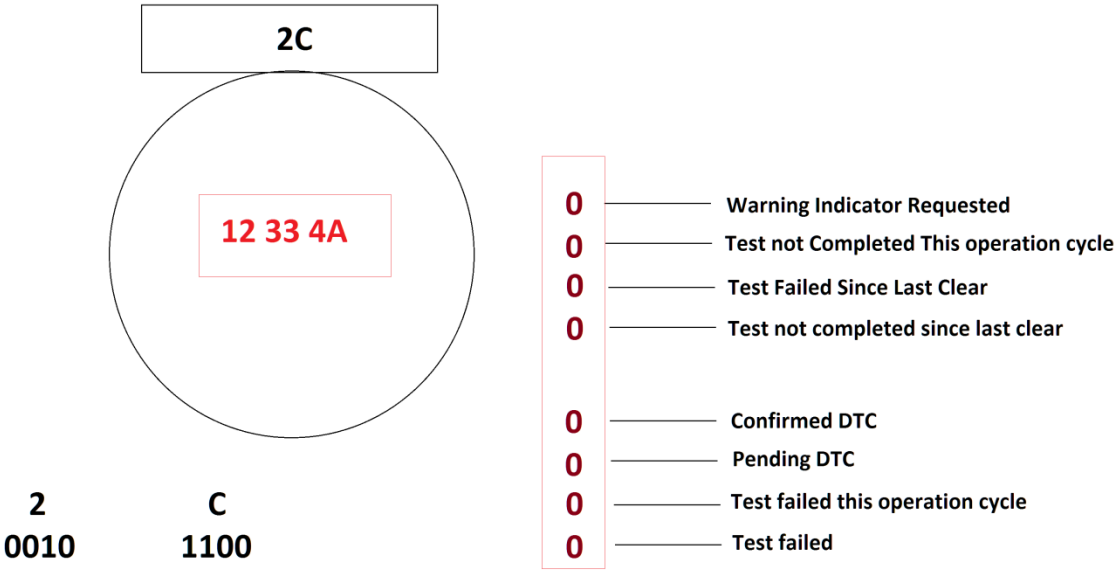
0 — Warning Indicator Requested
0 — Test not Completed This operation cycle
1 — Test Failed Since Last Clear
0 — Test not completed since last clear

1 — Confirmed DTC
1 — Pending DTC
0 — Test failed this operation cycle
0 — Test failed



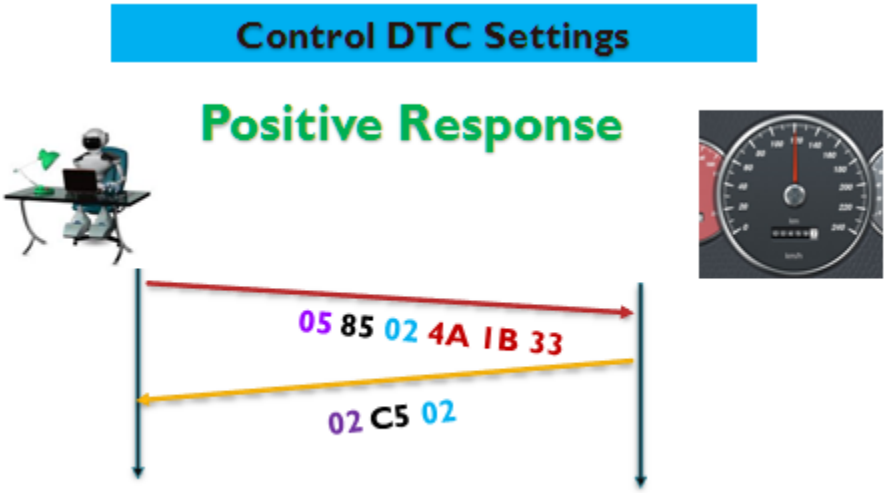
2 C
0010 1100





Assumption scenario:

- **Tester wants to diagnose/read faults other than Airbag failure fault (4A 1B 33), so the mentioned DTC is switched OFF (sub-function – 0x02)**



In the above transmission the DTC status bit update is supspended so this DTC will not log again !!

DTC's Suppressed because of 4A 1B 33 :

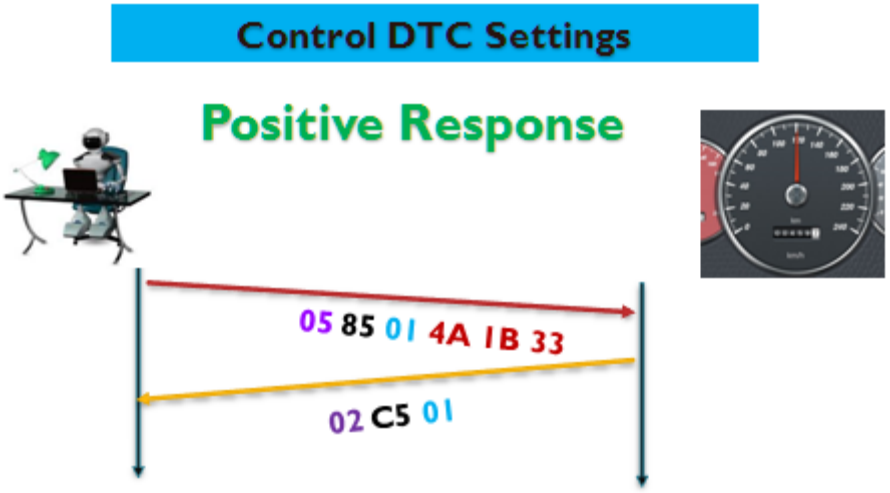
2F 01 00 - Brake failure -

11 3D 11 - Camera Lens Adjustment Needed -

16 21 1F - Spark Ignition gets damaged

10 1B 1A - Wheel pressure is lesser than threshold level

- **Whenever tester wants to switch ON the DTC (4A 1B 33), then mentioned DTC can be switched ON (sub-function – 0x01)**

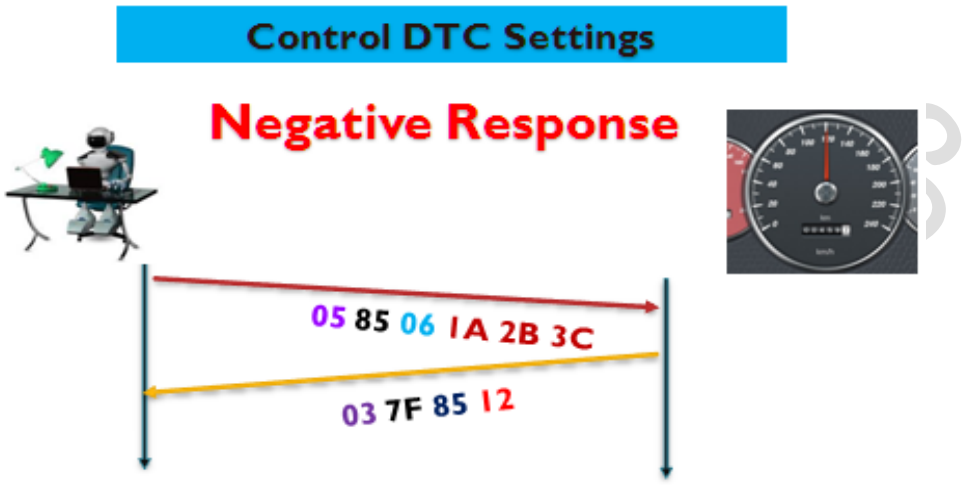


List of NRCs Supported

1. 0x12 Sub-function Not Supported
2. 0x13 Incorrect Message Length
3. 0x22 Conditions Not Correct
4. 0x31 Request Out of Range

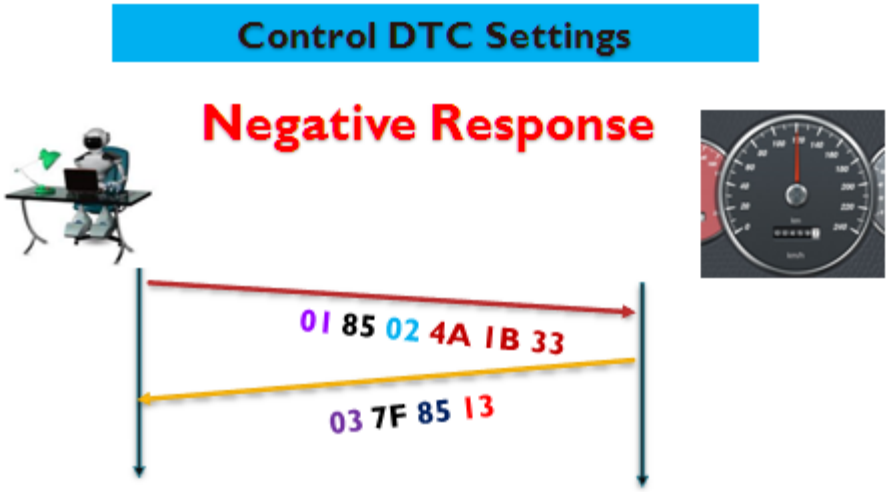
Sub-function Not Supported (0x12)

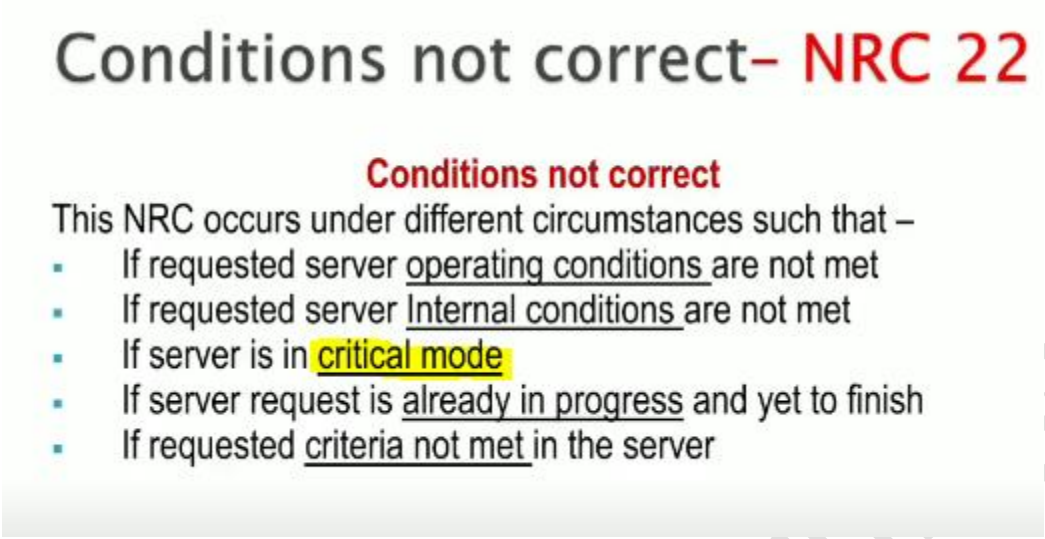
ECU responds with **NRC 12** if tester tries to request with unsupported sub-function and the **sub function is not supported** as per requirement



Incorrect Message Length (0x13)

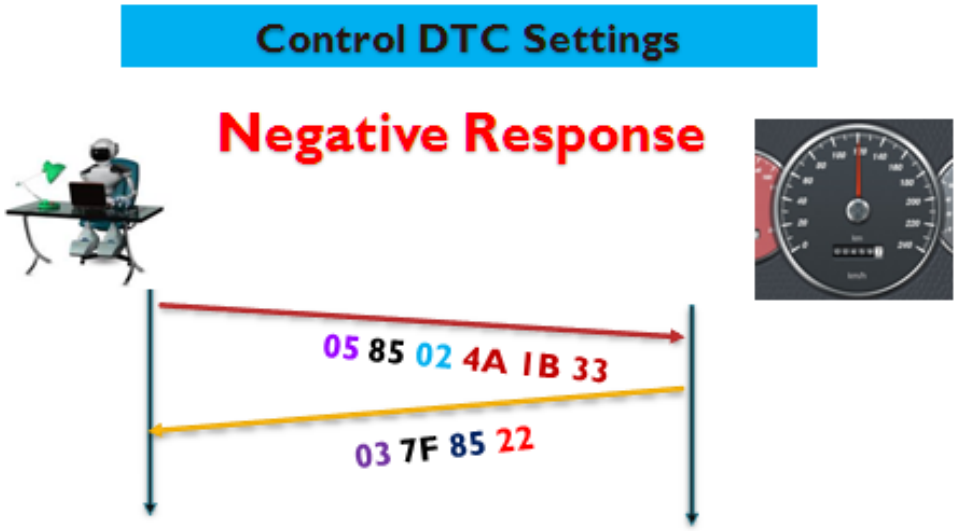
ECU responds with **NRC 13** if tester tries to request with **incorrect message length**





Sub-function Not Supported (0x22)

ECU responds with **NRC 22** if tester tries to request this service when the **conditions are not met**.



Sub-function Not Supported (0x31)

ECU responds with **NRC 31** if tester tries to request this service with DTC that is **out of range**.

Assumption Requirement says, DTC **1A 1B 3C** are not supported for this project. But tester requests with the **unsupported DTC**, Let's see the response for the request