MANUFACTURING SMOKE TEST

# Version History

|  |  |  |  |
| --- | --- | --- | --- |
| **Version Number** | **Version Date** | **Author** | **Description of Changes** |
| 0.1 | 30.JUN.2022 | Carlos Sanchez |  |

Table of Contents

[Version History 2](#_Toc107471415)

[1. Objective 4](#_Toc107471416)

[2. How to enter to the Manufacturing Mode 4](#_Toc107471417)

[2.1 Using Common Meet Tool 4](#_Toc107471418)

[2.2 Using VSSU Utility 7](#_Toc107471419)

[2.3 Exit of Manufacturing Mode 12](#_Toc107471420)

[3. Test Coverage 13](#_Toc107471421)

[3.1. Diagnostic Routines 13](#_Toc107471422)

[3.2. Memory Test 13](#_Toc107471423)

[3.3. Manufacturing Data 14](#_Toc107471424)

[3.4. Digital Input Ports 14](#_Toc107471425)

[3.5. Analog Ports 16](#_Toc107471426)

# Objective

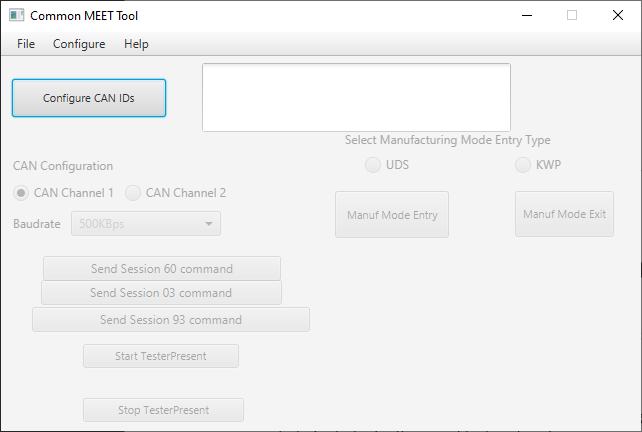
The objective of this document is to orientate to the validation engineers to make a correct manufacturing test using the correct tools and services.

**IMPORTANT:** **In some cases, it’s necessary the changes of some VOPS, if is necessary change them before starts the manufacturing test.**

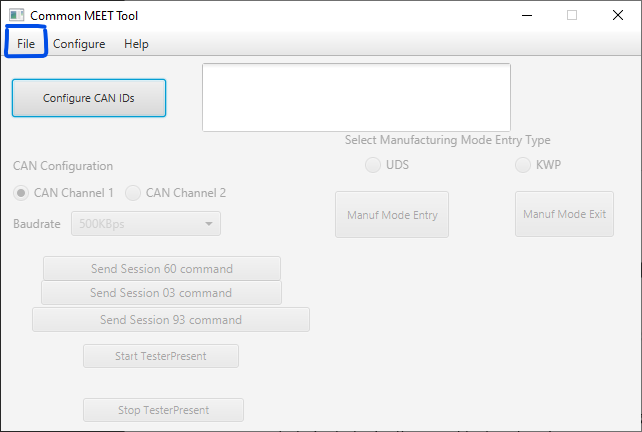
# How to enter to the Manufacturing Mode

## Using Common Meet Tool

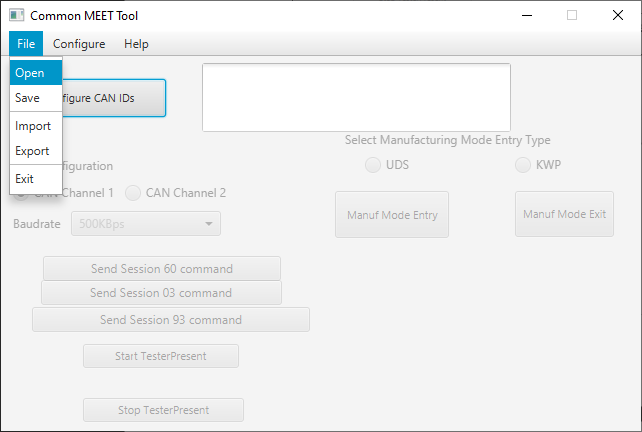
1. The easiest form to put a cluster in Manufacturing Mode is using the program Common Meet Tool, it’s necessary to have the most recently version of Java, this is necessary because if we have an old version Common Meet Tool probably crash or never opens.
2. Making the steps of the point 1. Open Common Meet Tool



* + 1. The first step when the program launch is going to File *(Image 1)* then go to Open *(Image 2)*



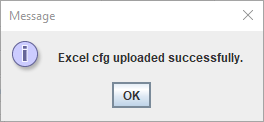
***Image 1. File Indication***

  
 ***Image 2. Open Indication***

* + 1. Once we open, we must select the file **CommonMEET\_Cfg\_p7** *(Image 3)* (*the name of the file is variable this name is only an example*) when the file was uploaded the program opens a window saying the next message *(Image 4)*

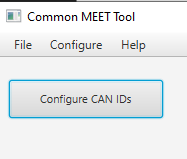


***Image 3. Files***

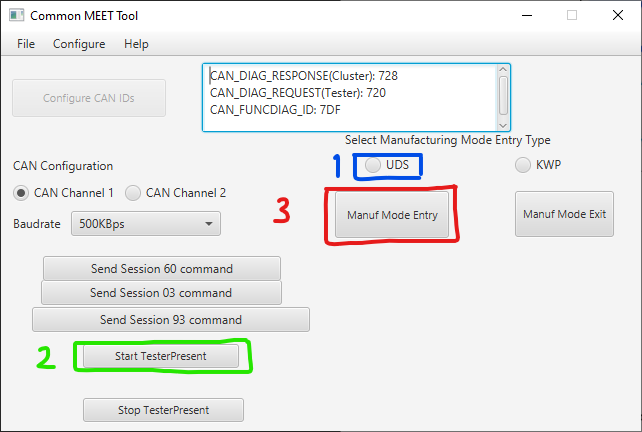


***Image 4. Message Window***

* + 1. After the file was uploaded, push the button Configure CAN IDs *(Image 5)* to enable the other options that the program gives us.



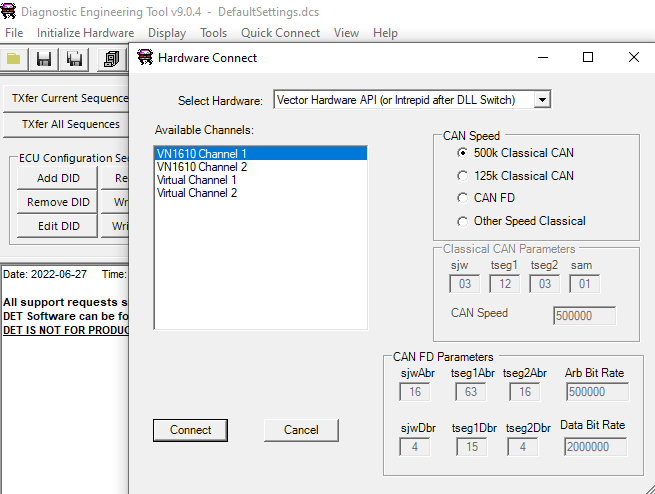
***Image 5. CAN IDs Button***

* + - 1. When the CAN IDs was activated, the program shows an emergency window giving the confirmation that now the CAN IDs is enable. After push ok to the emergency window, click on UDS (1) then Star TesterPresent (2) and finally click Manuf Mode Entry (3).

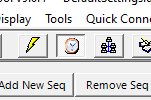
***Image 6. VSSU Utility***

* + - 1. After this sequence the program shows a lot of emergency windows with different messages confirming the entering to the manufacturing mode, in some cases the cluster can’t enter with Common Meet Tool in these cases use **VSSU Utility** (2.2).

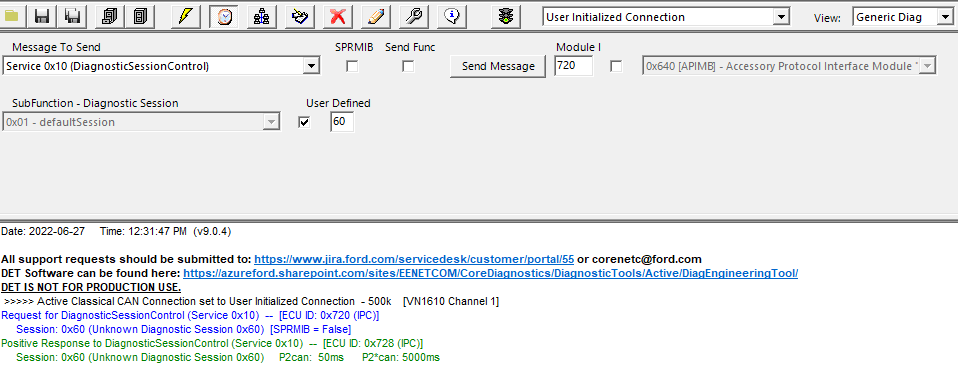
### Using VSSU Utility

1. If we can’t put the cluster in manufacturing mode with other tools, we use VSSU Utility, in this case it’s necessary the use of DET.
2. The first step is turn on the cluster with CAN Working *(image 7).* First in Initialize Hardware (1), then in the Hardware Connect it’s necessary to have selected the correct channel of the VN (2) once we have it it’s only necessary to make click in connect (3) and now the cluster is on.

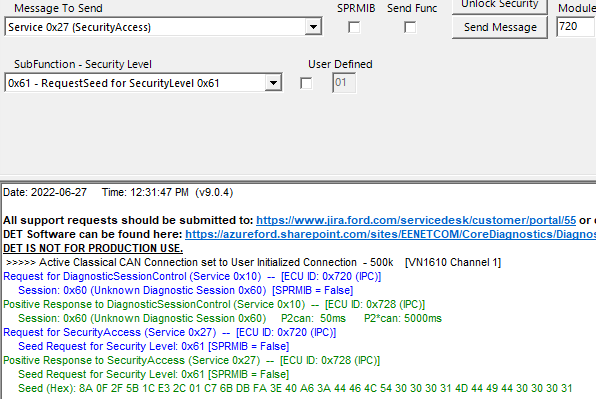
***Image 7. Cluster Awake with DET***

1. We need to activate the tester present ***(image 8)***.

***Image 8. Tester Present Button***

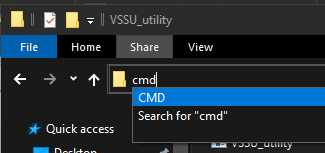
1. Once we did the last steps, we need to go to the Generic Diag view and go to the Service 0x10 and SubFunction 60.

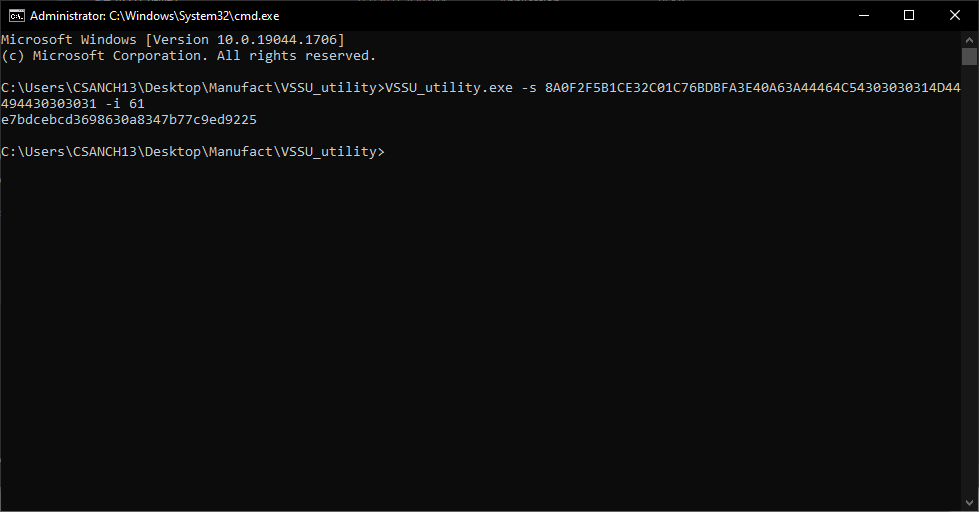
***Image 8. Session 60***

1. After we have a positive response, we need to request the seed for security level 0x61 and copy the seed **(mark in red)**.

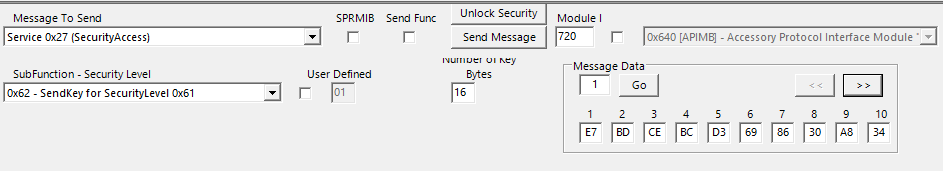
***Image 9. Seed***

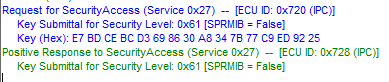
1. After the pass 5 we need to remove the spaces of the seed and use the key generator utility. For this case we only need to go to the executable of VSSU Utility and use the CMD to execute it **(image 10)**. Once we execute the file, we need to put the next message in the CMD: **>> VSSU\_utility.exe -s <SeedSentByClusterWithNoSpaces> -i 61**

As we can see in the image 11.

 ***Image 10. Execution***

***Image 11. VSSU Execution***

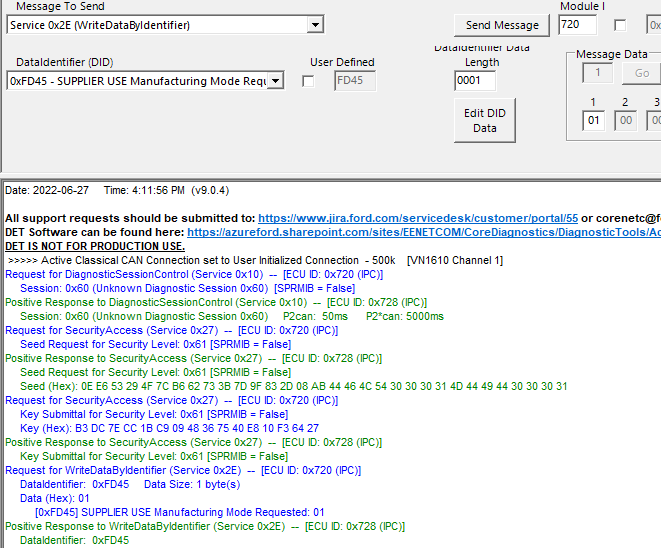
1. Once we have the generated key **(mark in red on image 11).** We need to send it using the service 0x27 and subservice 0x62. Please remember to set the length to 16 bytes and fill the key into the message data (use the buttons to navigate) **(marked in red)**

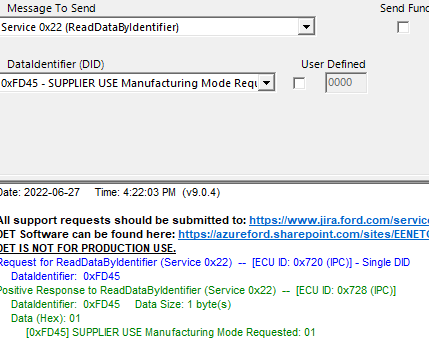
***Image 12. Generated Key***

***Image 13. Positive Response for the generated key***

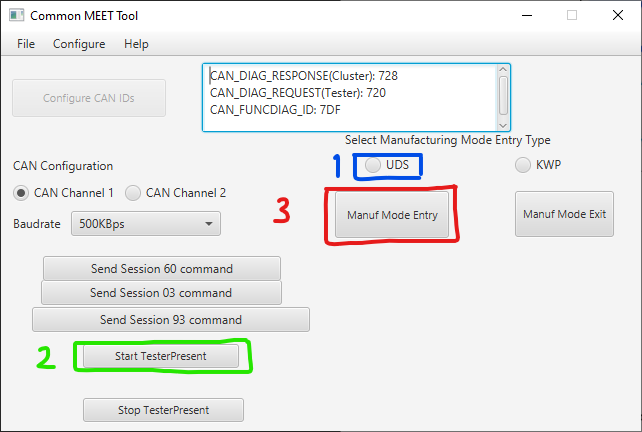
**Important: The generated key is different each time you request a seed. So be careful and modify all the VOPS**

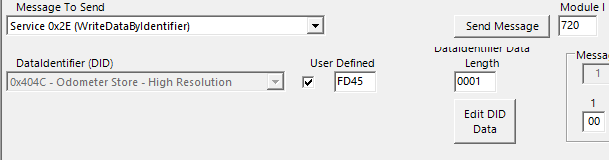
1. After we wrote the generated key, we need to go to the Service 0x2E and DataIdetifier (DID) in FD45, in message data at the first byte que need to put 01 ***(mark in red)*** and click on send message ***(image 14)***, after we have a positive response, we only need to reed the message in Service 0x22 and DID FD45 ***(image 15)***, if the response is 01 we enter to the manufacturing mode if not, we need to restart the process.

***Image 13. Service 0x2E***

 ***Image 14. Service 0x22***

## Exit of Manufacturing Mode

1. In case of Common Meet Tool, we only click on Manuf Mode Exit **(image 15)** and click **ok** in the emergency windows.
2. For VSSU Utility in the service 0x2E and DID FD45 we only write 00 on byte 1 and click on send message, to verify we only need to read on service 0x22 DID FD45 if the response is 00 all is ok if not, we need to reflash the cluster to bring back to developer mode.

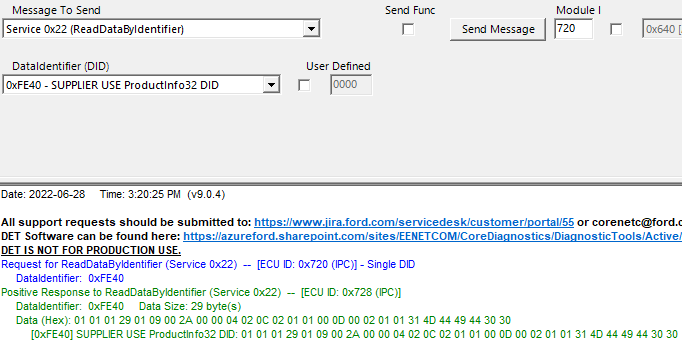
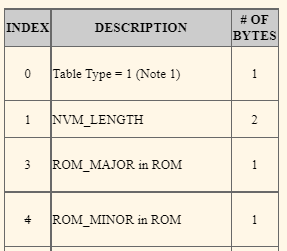
 ***Image 15. Mode Exit***

***Image 16. Service 0x2E in mode exit***

# Test Coverage

## Diagnostic Routines

## Memory Test

1. To read the information of the memory test need, we need the Manufacturing SRD and DET, in the table of ProductInfo2 we can find the information to search in DET the value. In this case for example we can find the ROM Major in ROM and ROM Minor in ROM, we can see the index (image 17) that give us the position of the value and using DET in the Service 0x22 and the DID 0xFE40 (image18) we can find it (mark in red), after we founded the value, we only need to save it in the document of the coverage.

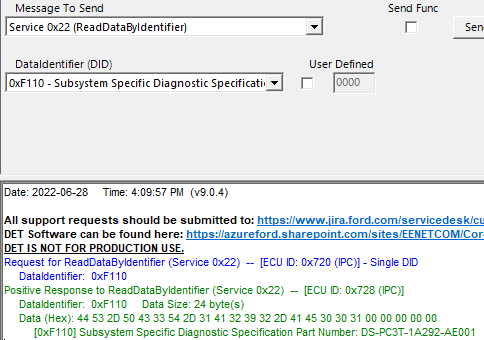
3 4

***Image 17. ProductInfo32***

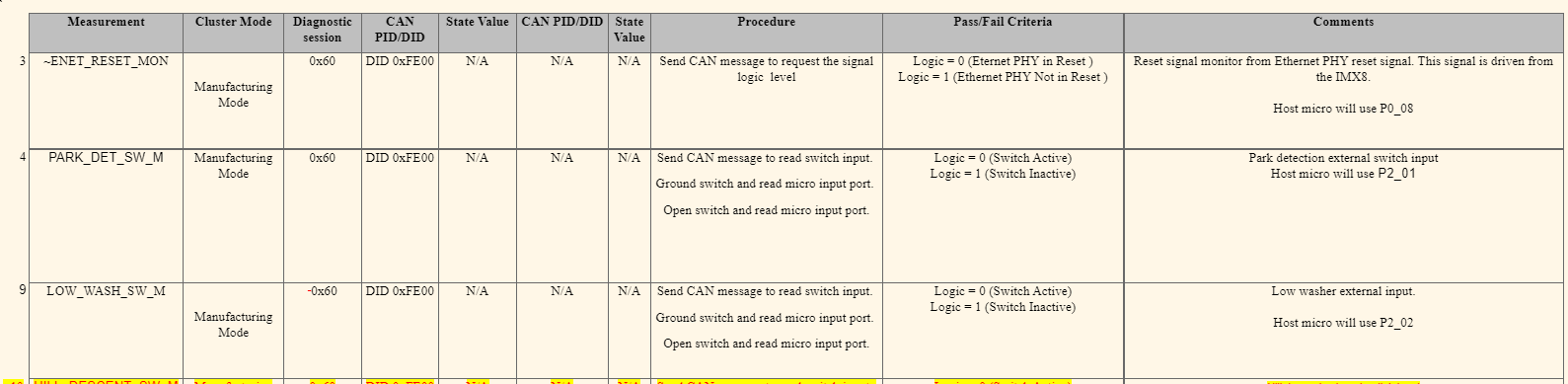
***Image 18. DID Read***

## Manufacturing Data

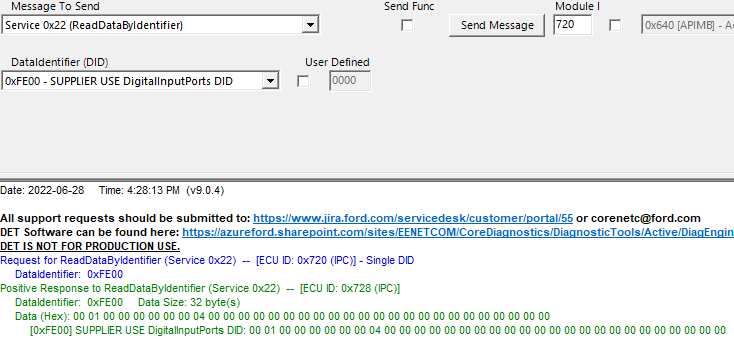
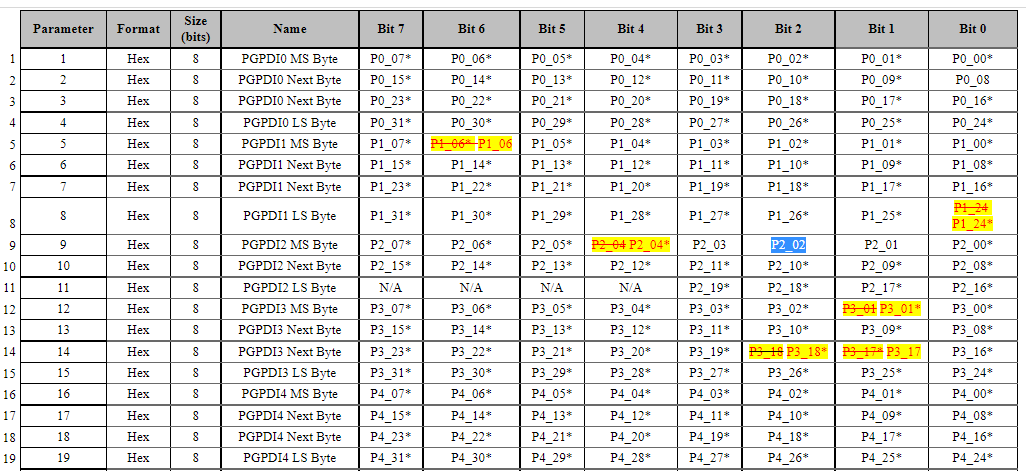
1. Using the service 0x22 and selecting the correct DID as we can see in the image 19.



## Digital Input Ports

1. In the service 0x22 and the DID 0xFE00 we can read the Digital Inputs. With help of the Manufacturing SRD using the Digital Table Ports **(image 21)** and the Digital Ports **(image 20)** we can find the information of the ports. With the parameters we can know in what position start the data of the digitalports **(image 22)** and the bit is to know in what binary bit is the information **(image 23)**.

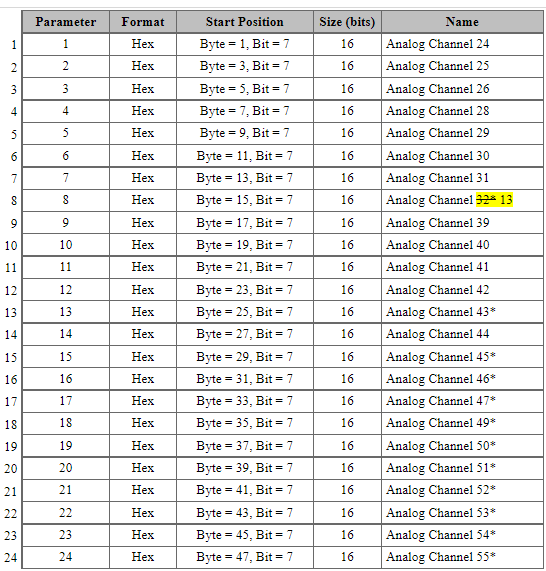
***Image 20. Digital Ports***

***Image 21. Digital Ports Table***

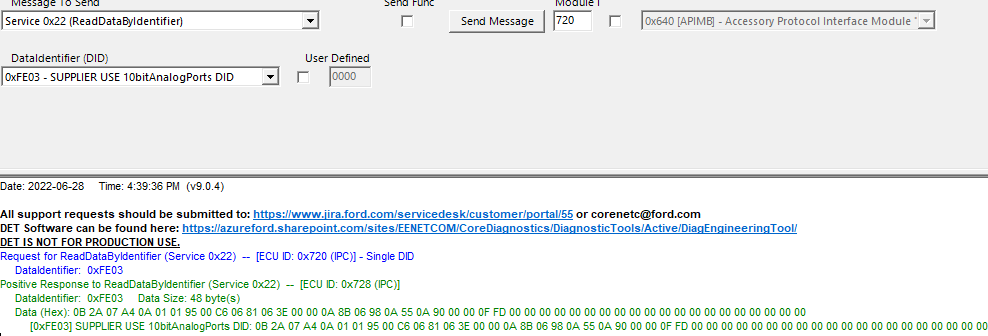
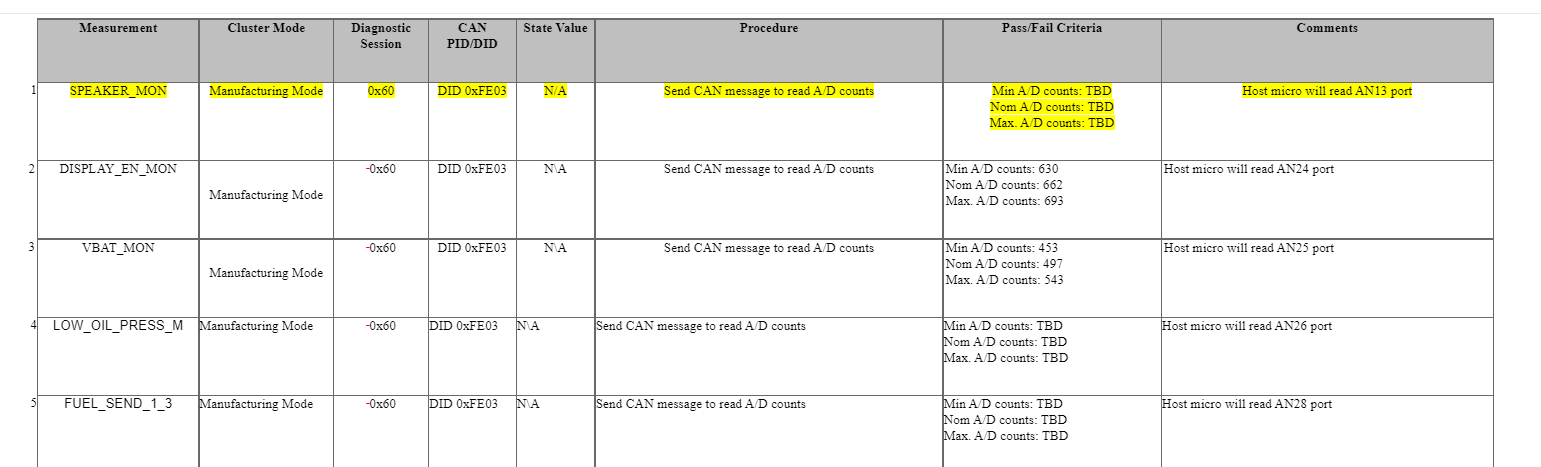
***Image 22. DigitalPorts Data***

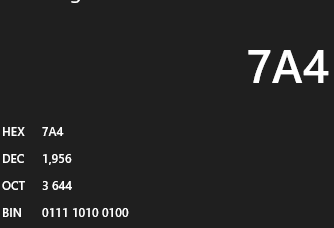
***Image 23. Hex to Bin***

## Analog Ports

For analog ports is the same process as the digital ports but in the data, que take 4 digits to read the value as we can see in the image 26. When we have this value, we need to transform the hexadecimal value to decimal **(image 27)** to know if the value is correct or similar as the value that we can see on the SRD.

***Image 24. Analog Ports Table***

****** ***Image 25. Analog Ports***

***Image 26. Analog Data***

***Image 27. Hex to Dec***