|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Title:** | EL/ET CAR Product Test Specification | | | | |
| **Project:** | ESTONA\_EL-ET-CAR\_ID5027-88 | | **Phase:** | A | |
| **HW Version:** | 01 | | **SW Version:** | A104 | |
| **EE:** | Shi Jianchao | **SW:** | Li Jiongqiu | **DEQ:** | Qiu Chuanhao |
| **Document Version:** | A01 | **Document Date:** | 2016-11-17 | **Document Status:** | Review |

Hints:

**Document Status**

Draft: The document is in work and not yet finished

Ready: All content is available in detail and the document is ready for review.

Reviewed: The doc. is reviewed by all reviewers and updated according to the review remarks.

Released: Before the document will be stored in SAP-PDM.

**Document Version**

The document needs a version number, for each new document version made available to someone different from the author(s), the version number has to be increased.

The documents must be reviewed and released!

**Use file name:**

ModelName\_PTS\_Version\_Status.doc

.

|  |  |  |  |
| --- | --- | --- | --- |
| History | | | |
| **Version** | **Date** | **Name, Department** | **Changes** |
| A01 | 2016-11-22 | Shi Jianchao/Li Jiongqiu | Create |
| A01 | 2016-11-29 | Li Jiongqiu | 1,Update 4.8; 4.12.3 |
|  |  |  |  |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Review List | | | |
| **Role** | **Name** | **Company/Department** | **Location** |
| Manufacturing Planning |  | MED | HZH |
| Quality Planning (QP) |  | QM-PQM | HZH |
| EE (senior engineer) | Shi Jianchao | BIC R&D EE | HZH |
| SW (senior engineer) | Li Jiongqiu | BIC R&D SW | HZH |
| Q&PTV (senior engineer) | Qiu Chuanhao | BIC DEQ | HZH |

|  |  |  |  |
| --- | --- | --- | --- |
| Distribution List | | | |
| **Role** | **Name** | **Company/Department** | **Location** |
| Manufacturing Planning |  | MED | HZH |
| Quality Planning (QP) |  | QM-PQM | HZH |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Applicable documents | | | | |
| **Title / document key** | **Description** | **Author** | **Date** | **Version/ Status** |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

Contents List Page

[1 Introduction 4](#_Toc467654004)

[1.1 Objectives 4](#_Toc467654005)

[1.2 Definitions, Acronyms and Abbreviations 4](#_Toc467654006)

[2 General Concepts 5](#_Toc467654007)

[2.1 Description 5](#_Toc467654008)

[2.2 Main Connector Pin Out 5](#_Toc467654009)

[3 Manufacturing 7](#_Toc467654010)

[3.1 Production Flow and Control Plan 7](#_Toc467654011)

[3.2 Handling of Components, PCBs and Finished DUTs 7](#_Toc467654012)

[3.2.1 ESD Care 7](#_Toc467654013)

[3.2.2 Bending of Components 7](#_Toc467654014)

[3.3 Visual Inspection 7](#_Toc467654015)

[3.4 Documentation/Traceability 7](#_Toc467654016)

[4 PCB Functional Test 8](#_Toc467654017)

[4.1 Purpose 8](#_Toc467654018)

[4.1.1 Operating Values 8](#_Toc467654019)

[4.1.2 Failures in PCB Functional Test 8](#_Toc467654020)

[4.2 General Measurement Conditions 8](#_Toc467654021)

[4.2.1 Test Jig Connection 8](#_Toc467654022)

[4.3 Supply Voltage Test 10](#_Toc467654023)

[4.4 Operation Current Test 10](#_Toc467654024)

[4.5 HW Test 11](#_Toc467654025)

[4.5.1 Under-&over-voltage 11](#_Toc467654026)

[4.5.2 Hardwire Telltales 11](#_Toc467654027)

[4.6 SW Test 11](#_Toc467654028)

[4.6.1 Diagnostic Command 11](#_Toc467654029)

[4.6.2 Bus 11](#_Toc467654030)

[4.6.3 Communication mechanism 12](#_Toc467654031)

[4.7 Enter PTS Mode 12](#_Toc467654032)

[4.8 Exit PTS Mode 13](#_Toc467654033)

[4.9 Quiescent Current Test 13](#_Toc467654034)

[4.10 EEPROM Access 13](#_Toc467654035)

[4.10.1 Write Memory by Address 13](#_Toc467654036)

[4.10.2 Read Memory by Address 13](#_Toc467654037)

[4.10.3 Init EEPROM Data from ROM 14](#_Toc467654038)

[4.11 Read Version Information 14](#_Toc467654039)

[4.12 Input & Output Test 15](#_Toc467654040)

[4.12.1 Analog Part 15](#_Toc467654041)

[4.12.2 Analogue Output 16](#_Toc467654042)

[4.12.3 Read the digital port 16](#_Toc467654043)

[4.12.4 Digital Output 18](#_Toc467654044)

[4.13 Key 18](#_Toc467654045)

[4.13.1 Read the Key when KEY Pressed 18](#_Toc467654046)

[4.14 **Bus LED Test** 18](#_Toc467654047)

[4.14.1 Backlight LED Test 20](#_Toc467654048)

[4.15 **Peripheral Test** 21](#_Toc467654049)

[4.15.1 **Movement Outputs Test** 21](#_Toc467654050)

[4.15.2 **Buzzer Operation** 25](#_Toc467654051)

[4.16 ADC Calibration 25](#_Toc467654052)

[4.17 **Pointer Calibration** 25](#_Toc467654053)

[4.17.1 **Pressure Pointer** 25](#_Toc467654054)

[4.17.2 **Pre-Actions** 25](#_Toc467654055)

[4.17.3 **Calibration** 26](#_Toc467654056)

[4.17.4 **Functionality Check** 31](#_Toc467654057)

[4.18 **Stress Test** 32](#_Toc467654058)

[4.19 **LCD Test** 32](#_Toc467654059)

[4.19.1 Segment Display 32](#_Toc467654060)

[4.19.2 LCD Pin Test 33](#_Toc467654061)

[4.19.3 TFT Display 33](#_Toc467654062)

[4.19.4 Dot Display 35](#_Toc467654063)

[4.20 EOL 35](#_Toc467654064)

[4.20.1 Power Gauge 35](#_Toc467654065)

[4.20.2 SOC Gauge 36](#_Toc467654066)

[4.21 Check Zero Point 37](#_Toc467654067)

[4.22 Factory Setting 37](#_Toc467654068)

[4.23 EEPROM Layout Factory Check 38](#_Toc467654069)

# Introduction

## Objectives

This Product Test Specification (PTS) defines the requirements for the PCB Functional Test and the End of Line Test of the **EL/ET CAR Instrument Cluster Unit**.

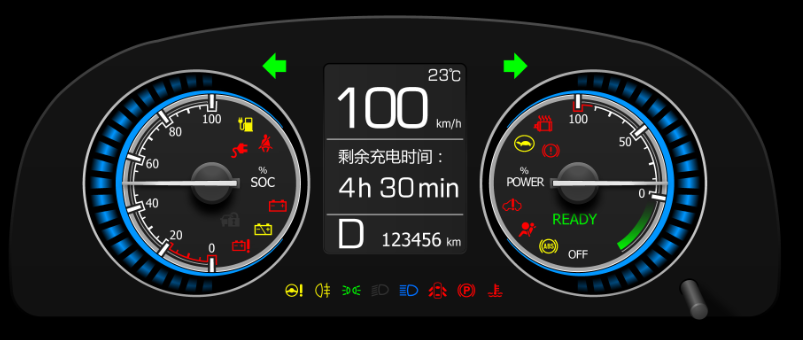
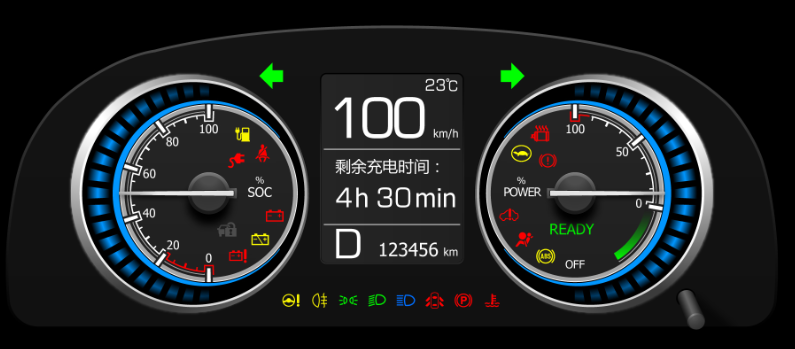
## Definitions, Acronyms and Abbreviations

| **Abbreviation** | **Meaning** |
| --- | --- |
| PTS | Product Test Specification |
| BOM | Bill of material |
| DUT | Device under Test |
| EOL | End of Line |
| FA | Final Assembly |
| KL15 | Ignition Voltage |
| KL30 | Battery Voltage |
| GND | Ground |
| LCD | Liquid Crystal Display |
| SM | Stepper Motor |
| uC | Microcontroller |
| AD | Analog to Digital |
| 0xXX | Means data not care |
|  |  |

# General Concepts

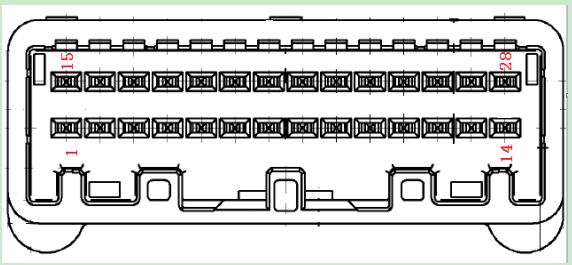
## Description

**1 PCB P/N is** 3140 193 52611



## Main Connector Pin Out

1. **connectors A:**



| Pin | Symbol | I/O | Description | Remarks |
| --- | --- | --- | --- | --- |
| 1 |  |  |  |  |
| 2 | CAN\_H | I |  |  |
| 3 | CAN\_L | I |  |  |
| 4 |  |  |  |  |
| 5 | Driver Seatbelt | I | IDL |  |
| 6 | Park Brake | I | IDL |  |
| 7 | Air Bag | I | IDL |  |
| 8 | EPS | I | IDL |  |
| 9 |  |  |  |  |
| 10 | Vehicle Speed Out | O | 50mA；OC输出，方波信号 |  |
| 11 | Reverse Lamp | O | 100-150mA；  Low Side |  |
| 12 |  |  |  |  |
| 13 | IGN | I | IDH |  |
| 14 | BATT+ | I | Vcc |  |
| 15 | OAT Return | - | 预留 |  |
| 16 | OAT Signal | - | 预留 |  |
| 17 | PFAF Trigger | - | output，预留 |  |
| 18 |  |  |  |  |
| 19 | Door Open | I | IDL |  |
| 20 | Rear Fog | I | IDH |  |
| 21 | BMS wakeup | I | IDH |  |
| 22 | Turn Left | I | IDH |  |
| 23 | Turn Right | I | IDH |  |
| 24 | Lights On | I | IDH |  |
| 25 | High Beam | I | IDH |  |
| 26 | Low Beam | I | IDH |  |
| 27 |  | - |  |  |
| 28 | GND | I |  |  |

# Manufacturing

## Production Flow and Control Plan

The components test has to be done before the product test. The purpose of the test is to verify that the value of the component is within its specified tolerance. The tolerance allowed is increased from the components original because tolerance of the test equipment should be taken into account as well.

The complete product test is split into the following sections:

Functional Test

The boards are tested for shorts, open wires and component faults.

The PCB function test gives the possibility to find hardware failures before the PCB will be mounted into the housing.

EOL (End of Line)

If there are any faults on the device, as many as possible, it should be detected at the PCB Functional Test to ensure lower failure rates at the test station (EOL).

## Handling of Components, PCBs and Finished DUTs

### ESD Care

The ESD regulations for handling must be followed.

### Bending of Components

When handling the PCBs it is very important that no components are damaged by bending.

## Visual Inspection

All components must be subjected to a 100 % visual inspection. These components will be identified by the BOM and the layout data.

Components whose value and tolerance cannot be clearly identified must be subject of a type check at every change of tape or lot.

The PCB must be checked for

* existence of components,
* correct position of the components,
* Soldering quality.

## Documentation/Traceability

All components to be documented and the measured values of the tests must be assigned to the respective DUT and recorded in the data interlocking system.

# PCB Functional Test

## Purpose

The aim of the sequence of tests performed for the PCB Functional Test is to ensure that the PCBA produced will be functional when assembled into the Instrument Cluster. The DUT must be programmed (flashed) to start the functional test.

If a fault at the DUT is detected by any test of the PCB Functional Test, the fault is reported, the test is stopped after the termination of the current test block and the DUT is sent to the repair station.

### Operating Values

The following external connections must be applied to the DUT to put it into operation.

To test the DUT in a realistic way, the following voltages or components have to supplied or applied to the DUT.

Operating values for the PCB Functional Test:

| **Signal** | **Test Pin**  **(Blue Block)** | **Test Point** | **PCB (Top**  **Bottom)** | **Values/Connection** | **Remark** |
| --- | --- | --- | --- | --- | --- |
| KL15 | PIN13 | F108 | Bottom | Power 13.5V | Ignition |
| KL30 | PIN14 | F109 | Top | Power 13.5V | Quiecent Current ≤300uA;  Typical = 200u A |
| GND | PIN28 | F110 | Bottom | Ground 0V |  |

### Failures in PCB Functional Test

If any module fails in the PCB Functional Test, the module should be sent to the repair station. At the repair station the failure is fixed. This might be caused by placing a missing component of replacing of a component that is out of tolerance or does not pass the digital test in the case of transistors.

The module is then tested in PCB Functional Test station once again.

## General Measurement Conditions

### Test Jig Connection

Refer to：将文件最新文件名字写上





**Remarks:**

Refer to the original document <ModelName\_Test Box Schematic\_Version\_Status.pdf> which attached with this PTS.

| **Signal Name** | **Operation Voltage/Current** | **Connection Pin** | **Test Measurement** | **Soft Commend** |
| --- | --- | --- | --- | --- |
| KL30 | 13.5±0.5V，I<300uA | PIN13:KL30 |  |  |
| KL15 | 13.5±0.5V，I<500mA | PIN14:KL15 |  |  |
| Low Beam | 13.5±0.5V,I<1mA | PIN26:Low Beam |  |  |
| High Beam | 13.5±0.5V,I<1mA | PIN25:High Beam |  |  |
| Lights on | 13.5±0.5V,I<1mA | PIN24:Lights On |  |  |
| Turn Right | 13.5±0.5V,I<1mA | PIN23:Turn Right |  |  |
| Turn Left | 13.5±0.5V,I<1mA | PIN22:Turn Left |  |  |
| BMS Wakeup | 13.5±0.5V,I<1mA | PIN21:BMS Wakeup |  |  |
| Rear Fog | 13.5±0.5V,I<1mA | PIN20:Rear Fog |  |  |
| Door open | 0~1V,I<2mA | PIN19:Door Open |  |  |
| EPS | 0~1V,I<2mA | PIN8:EPS |  |  |
| Air bag | 0~1V,I<2mA | PIN7:Air Bag |  |  |
| Park Brake | 0~1V,I<2mA | PIN6:Park Brake |  |  |
| Driver Sealbelt | 0~1V,I<2mA | PIN5:Driver Sealbelt |  |  |
| Reverse Lamp | 0~1V,100mA<I<150mA | PIN11:Reverse Lamp |  |  |
| Can\_L |  | PIN2:CAN\_H |  |  |
| Can\_H |  | PIN3:CAN\_L |  |  |
| PFAF Trigger | NA | PIN17: PFAF Trigger |  |  |
| OTA Signal | NA | PIN16: OTA Signal |  |  |
| OTA Return | NA | PIN15: OTA Return |  |  |
| 车速输出 | 0,13.5±1V V,I<2mA | PIN10:Speed Out |  |  |
| GND | GND | PIN28:GND |  |  |

## Supply Voltage Test

**(1) Battery Voltage**

(Describe how to operate the DUT.)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Signal Name** | **Signal Abbreviation** | **Operation Voltage** | **Test Pin** | **Test Point** |
| KL30 | Battery + | 13.5V | PIN14 | F109 |

**Measure the voltage 3s later after the connectors is inserted.**

**(2) Ignition Voltage**

(Describe how to operate the DUT.)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Signal Name** | **Signal Abbreviation** | **Operation Voltage** | **Test Pin** | **Test Point** |
| KL15 | Ignition | 13.5V | PIN13 | F108 |

**Measure the voltage 3s later after the connectors is inserted.**

## Operation Current Test

**(1) Battery Current**

(Describe how to operate the DUT.)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Signal Name** | **Signal Abbreviation** | **Operation Current** | **Test Pin** | **Test Point** |
| KL30 | Battery + | 300mA~400mA | PIN14 | F109 |

**Measure the voltage 5s later after the connectors is inserted.**

**(2) Ignition Current**

(Describe how to operate the DUT.)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Signal Name** | **Signal Abbreviation** | **Operation Current** | **Test Pin** | **Test Point** |
| KL15 | Ignition | 100mA~200mA | PIN13 | F108 |

**Measure the voltage 5s later after the connectors is inserted.**

## HW Test

### Under-&over-voltage

Testing (with CAN-Communication):

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Action** | **Waiting time** | **Measuring point** | **Reaction** | **Note** |
| VBatt = 13.5V, Ter.30 On, Ter.15 Off, LCD illumination OFF, Dial illumination Off,  almost telltales off | >10ms | **F129** | 5,0V ± 0,1V | 5V supply |
|  | 20ms | **F130** | 0V + 0,5V | Reset (typ. 40ms) |
|  | >40ms | **F130** | 5V ± 0,2V | Release reset |
|  |  |  |  |  |
|  |  |  |  |  |
|  | >200ms | **ITer.30** | I≤4mA | 1\*) |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
| VBatt = 18V, Key switch On, ILL Off, | >4s | **F129** | 0V + 0,5V | IGN input (over-voltage) |
| VBatt = 13.5V, Key switch On, ILL Off, | >10ms | **F129** | 5,0V ± 0,1V | 5V supply |
|  |  |  |  |  |
| VBatt = 9V, Key switch On, ILL Off, | >4s | **F129** | 0V + 0,5V | IGN input (under-voltage) |
| VBatt = 13.5V, Key switch On, ILL Off, | >10ms | **F129** | 5,0V ± 0,1V | 5V supply |
|  |  |  |  |  |
|  |  |  |  |  |

### Hardwire Telltales

N.A

## SW Test

### Diagnostic Command

### Bus

The bus used in the ICU is CAN bus.

#### Bus Interface

The cluster connect to high speed CAN bus.

The communication speed is 500KBps, half-duplex communication line.

CAN

Interface

**EL/ET CAR** Cluster

CAN\_H

CAN\_L

### Communication mechanism

Use the Master-Slave mechanism,

|  |
| --- |
| The tester needs to send the request firstly, and then the ECU wills response the Data. |
|
| Tester Request Format:   |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | | **Can ID(STD)** | **Byte0** | **Byte1** | **Byte2** | **Byte3** | **Byte4** | **Byte5** | **Byte6** | **Byte7** | | 0x772 | 0xXX | 0xXX | 0xXX | 0xXX | 0xXX | 0xXX | 0xXX | 0xXX | |
|

ECU Response Format:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Byte0** | **Byte1** | **Byte2** | **Byte3** | **Byte4** | **Byte5** | **Byte6** | **Byte7** |
| 0x774 | 0xXX | 0xXX | 0xXX | 0xXX | 0xXX | 0xXX | 0xXX | 0xXX |

Send password which is group into two sets described below to the ECU and the ECU will response positive acknowledge, and then enter PTS Mode.

* 1. Enter PTS Mode

Wait 1 seconds after KL30 &KL15 is ON (Before entering PTS mode).

**Step1：Send PasswordSet1**

**Send the below message to ECU for PTS test enable: password set1**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x772 | 0x27 | 0x55 | 0xB6 | 0x54 | 0xA2 | 0x48 | 0xD5 | 0xE5 |

**The device answer with the acknowledgement sent as following:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x774 | 0x67 | 0x11 | 0xAA | 0xAA | 0xAA | 0xAA | 0xAA | 0xAA |

“--” means padding pattern value, the default value is “0xAA”.

Every command must be fix to 8 bytes.

Delay more about 50ms,

**Step2: Send Password Set2**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x772 | 0x27 | 0x64 | 0x52 | 0xC5 | 0x7D | 0x66 | 0x88 | 0x75 |

**The ECU answer with the acknowledgement sent as following:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x774 | 0x67 | 0x11 | -- | -- | -- | -- | -- | -- |

## Exit PTS Mode

**Send commend for exiting the PTS mode**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID** | **Byte0** | **Byte1** | **Byte2** | **Byte3** | **Byte4** | **Byte5** | **Byte6** | **Byte7** |
| 0x772 | 0x11 | 0xAA | 0xAA | 0xAA | 0xAA | 0xAA | 0xAA | 0xAA |

**The device answer with the acknowledgement sent as following:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID** | **Byte0** | **Byte1** | **Byte2** | **Byte3** | **Byte4** | **Byte5** | **Byte6** | **Byte7** |
| 0x774 | 0x51 | 0x01 | -- | -- | -- | -- | -- | -- |

## Quiescent Current Test

**Test method of quiescent current as following:**

**Firstly, KL30 on , KL15 on , enter production check.**

**Second, KL30 on , KL15 off, and send TT1 to enter sleep and then ECU will respond the quest.**

**Thirdly, wait for 2s, then test the current.**

(Describe how to operate the DUT.)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Signal Name** | **Signal Abbreviation** | **Quiescent Current** | **Test Pin** | **Test Point** |
| GND/KL30 |  | ≤300uA (TYP=200uA) | PIN14/ PIN28 |  |

**The following command can be used to enter the sleep mode quickly.**

**TT1**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x772 | 0x12 | 0x33 | 0x5A | 0xAA | 0xAA | 0xAA | 0xAA | 0xAA |

**The ECU answer with the acknowledgement sent as following:**

**TT1:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| **0x774** | **0x52** | **0x33** | **--** | **--** | **--** | **--** | **--** | **--** |

## EEPROM Access

### Write Memory by Address

**Access EEPROM request:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x772 | 0x3D | AddH | AddL | Length | 0xXX | 0xXX | 0xXX | 0xXX |

**Receive data:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x774 | 0x7D | Length | -- | -- | -- | -- | -- | -- |

Remark: Address range should be 0-2048 for 24C16(16kbit EEPROM)， data should be 0-3bytes

### Read Memory by Address

**Send data**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x772 | 0x23 | AddrH | AddrL | DataLength | 0xXX | 0xXX | 0xXX | 0xXX |

**ECU will response with the following data.**

**Receive Data**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x774 | 0x63 | 0xXX | 0xXX | 0xXX | 0xXX | 0xXX | 0xXX | 0xXX |

**Remarks: For 24C16 EEPROM， valid address range should be 0-2048, DataLength to read should be 0-7 bytes.**

### Init EEPROM Data from ROM

Command 0x04 Send:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x772 | 0x3C | 0x04 | 0xAA | 0xAA | 0xAA | 0xAA | 0xAA | 0xAA |

Ecu Response:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x774 | 0x7C | 0x04 | 0x11 | 0x22 | 0x33 | 0x44 | --- | --- |

Command 0x05 Send:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x772 | 0x3C | 0x05 | 0x11 | 0x22 | 0x33 | 0x44 | 0xAA | 0xAA |

Ecu Response:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x774 | 0x7C | 0x05 | --- | --- | --- | --- | --- | --- |

In example above, the rand series generated by ECU is 0x11 0x22 0x33 0x44.

## Read Version Information

**1. The following command can be used for reading the** **Read hardware and software version //读取硬件和软件版本号:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | Data0 | Data1 | Data2 | Data3 | Data4 | Data5 | Data6 | Data7 |
| 0x772 | 0x1A | 0x80 | 0xAA | 0xAA | 0xAA | 0xAA | 0xAA | 0xAA |

**The ECU answer with the acknowledgement sent as following:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | Data0 | Data1 | Data2 | Data3 | Data4 | Data5 | Data6 | Data7 |
| 0x774 | 0x5A | 0x80 | 0xXX | 0xXX | 0xXX | 0xXX | --- | --- |

**The data field is defined as follow table:**

|  |  |
| --- | --- |
| **Data field** | **Description** |
| Data2 | Hardware version number high byte |
| Data3 | Hardware version number low byte |
| Data4 | Software version number high byte |
| Data5 | Software version number low byte |

**Example:**

Data2 = 0x00,Data3=0x01 means hardware version number is 0001

Data4= 0xB1,Data5=0x01 means software version number is B101:

**2. The following command can be used for reading the Read** **software release date//读取软件发布日期:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | Data0 | Data1 | Data2 | Data3 | Data4 | Data5 | Data6 | Data7 |
| 0x772 | 0x1A | 0x81 | 0xAA | 0xAA | 0xAA | 0xAA | 0xAA | 0xAA |

**The ECU answer with the acknowledgement sent as following:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | Data0 | Data1 | Data2 | Data3 | Data4 | Data5 | Data6 | Data7 |
| 0x774 | 0x5A | 0x81 | 0xXX | 0xXX | 0xXX | 0xXX | --- | --- |

**The data field is defined as follow table:**

|  |  |
| --- | --- |
| **Data field** | **Description** |
| Data2 | Software release year high byte |
| Data3 | Software release year low byte |
| Data4 | Software release month byte |
| Data5 | Software release day byte |

**Example:**

Data2=0x20 Data3=0x15 Data4=0x12 Data5=0x22, means Software release data is 2015-12-22

**3. The following command can be used for reading the Read** **program data checksum //读取程序数据校验和:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | Data0 | Data1 | Data2 | Data3 | Data4 | Data5 | Data6 | Data7 |
| 0x772 | 0x1A | 0x82 | 0xAA | 0xAA | 0xAA | 0xAA | 0xAA | 0xAA |

**The ECU answer with the acknowledgement sent as following:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | Data0 | Data1 | Data2 | Data3 | Data4 | Data5 | Data6 | Data7 |
| 0x774 | 0x5A | 0x82 | 0xXX | 0xXX | --- | --- | --- | --- |

**The data field is defined as follow table:**

|  |  |
| --- | --- |
| **Data field** | **Description** |
| Data2 | Program data checksum high byte |
| Data3 | Program data checksum low byte |

**Example:**

Data2 = 0x12,Data3=0x34 means program data checksum is 0x1234.

**4. The following command can be used for reading the Read EEPROM Version //读取EEPROM 版本:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | Data0 | Data1 | Data2 | Data3 | Data4 | Data5 | Data6 | Data7 |
| 0x772 | 0x1A | 0x83 | 0xAA | 0xAA | 0xAA | 0xAA | 0xAA | 0xAA |

**The ECU answer with the acknowledgement sent as following:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | Data0 | Data1 | Data2 | Data3 | Data4 | Data5 | Data6 | Data7 |
| 0x774 | 0x5A | 0x83 | 0xXX | 0xXX | --- | --- | --- | --- |

**The data field is defined as follow table:**

|  |  |
| --- | --- |
| **Data field** | **Description** |
| Data2 | EEPROM version high byte |
| Data3 | EEPROM version low byte |

**5. The following command can be used for reading the Read Manufacturing date//制造日期:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | Data0 | Data1 | Data2 | Data3 | Data4 | Data5 | Data6 | Data7 |
| 0x772 | 0x1A | 0x84 | 0xAA | 0xAA | 0xAA | 0xAA | 0xAA | 0xAA |

**The ECU answer with the acknowledgement sent as following:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | Data0 | Data1 | Data2 | Data3 | Data4 | Data5 | Data6 | Data7 |
| 0x774 | 0x5A | 0x84 | 0xXX | 0xXX | 0xXX | --- | --- | --- |

**The data field is defined as follow table:**

|  |  |
| --- | --- |
| **Data field** | **Description** |
| Data2 | year byte |
| Data3 | month byte |
| Data4 | day byte |

## Input & Output Test

### Analog Part

**Test under KL30 = 13.5V**

**The following command can be used to get analogue port data of Power Voltage Value**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x772 | 0x21 | 0x11 | 0xAA | 0xAA | 0xAA | 0xAA | 0xAA | 0xAA |

**The ECU answer with the acknowledgement sent as following:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x774 | 0x61 | 0x11 | 0xXX | 0xXX | 0xAA | 0xAA | 0xAA | 0xAA |

The data field is defined as following table:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Data1** | **Data field** | **Description** | **AD Value** | **Voltage** | **Test Pin** | **Test Point** | **Top**  **Bottom** |
| 0x11 | Data2 | Power Voltage high byte | ±0.5V | ±0.1V  2.6V | PIN14 | F124 | Bottom |
| Data3 | Power Voltage low byte |

**Example:** Data2 = 0x34, Data3 = 0x7B;

Data = 0x343B=13435, V = 13.435V

### Analogue Output

**N.A**

### Read the digital port

**The following command can be used to get analogue port data**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x772 | 0x21 | 0x41 | 0xAA | 0xAA | 0xAA | 0xAA | 0xAA | 0xAA |

**The ECU answer with the acknowledgement sent as following:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x774 | 0x61 | 0x41 | 0xXX | --- | --- | --- | --- | --- |

| **Data2~4**  **Bit** | **Description** | **Control** | | **Test Voltage** | **Test Pin** | **Test Point**  **（BLOCK）** | **Test Points**  **（MCU Port）** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Value** | **Description** |
| 7 | **Driver Seatbelt** | 0 | Low Level | <1V | PIN5 | F102 (Bottom) | F188 (Bottom) |
| 1 | High Level | >9V |
| 6 | **Park Brake** | 0 | Low Level | <1V | PIN6 | F103 (Bottom) | F189 (Bottom) |
| 1 | High Level | >9V |
| 5 | **Air Bag** | 0 | Low Level | <1V | PIN7 | F104 (Bottom) | F190 (Bottom) |
| 1 | High Level | >9V |
| 4 | **EPS** | 0 | Low Level | <1V | PIN8 | F105 (Bottom) | F187 (Bottom) |
| 1 | High Level | >9V |
| 3 | **Door Open** | 0 | Low Level | <1V | PIN19 | F117 (TOP) | F133 (Bottom) |
| 1 | High Level | >9V |
| 2 | **Rear Fog** | 0 | Low Level | <1V | PIN20 | F116 (TOP) | F161 (Bottom) |
| 1 | High Level | >9V |
| 1 | **BMS wakeup** | 0 | Low Level | <1V | PIN21 | F115 (TOP) | F165 (Bottom) |
| 1 | High Level | >9V |
| 0 | **Turn Left** | 0 | Low Level | <1V | PIN22 | F114 (TOP) | F159 (Bottom) |
| 1 | High Level | >9V |
| 7 | **Turn Right** | 0 | Low Level | <1V | PIN23 | F113 (TOP) | F160 (Bottom) |
| 1 | High Level | >9V |
| 6 | **Lights On** | 0 | Low Level | <1V | PIN24 | F112 (TOP) | F163 (Bottom) |
| 1 | High Level | >9V |
| 5 | **High Beam** | 0 | Low Level | <1V | PIN25 | F111 (TOP) | F162 (Bottom) |
| 1 | High Level | >9V |
| 4 | **Low Beam** | 0 | Low Level | <1V | PIN26 | F140 (TOP) | F164 (Bottom) |
| 1 | High Level | >9V |

| **No** | **Description** | **State1** | **State2** | **State3** | **State4** |
| --- | --- | --- | --- | --- | --- |
| 1 | **Driver Seatbelt** | H | L | H | L |
| 2 | **Park Brake** | H | L | L | H |
| 3 | **Air Bag** | H | L | H | L |
| 4 | **EPS** | H | L | L | H |
| 5 | **Door Open** | H | L | H | L |
| 6 | **Rear Fog** | H | L | L | H |
| 7 | **BMS wakeup** | H | L | H | L |
| 8 | **Turn Left** | H | L | L | H |
| 9 | **Turn Right** | H | L | H | L |
| 10 | **Lights On** | H | L | L | H |
| 11 | **High Beam** | H | L | H | L |
| 12 | **Low Beam** | H | L | L | H |

**The ECU answer with the acknowledgement sent as following:**

**State1**

**The following command can be used to get analogue port data**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x772 | 0x21 | 0x41 | 0xAA | 0xAA | 0xAA | 0xAA | 0xAA | 0xAA |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x774 | 0x61 | 0x41 | 0xFF | 0xF0 | 0x00 | --- | --- | --- |

**State2**

**The following command can be used to get analogue port data**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x772 | 0x21 | 0x41 | 0xAA | 0xAA | 0xAA | 0xAA | 0xAA | 0xAA |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x774 | 0x61 | 0x41 | 0x00 | 0x00 | 0x00 | --- | --- | --- |

**State3**

**The following command can be used to get analogue port data**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x772 | 0x21 | 0x41 | 0xAA | 0xAA | 0xAA | 0xAA | 0xAA | 0xAA |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x774 | 0x61 | 0x41 | 0xAA | 0xA0 | 0x00 | --- | --- | --- |

**State4**

**The following command can be used to get analogue port data**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x772 | 0x21 | 0x41 | 0xAA | 0xAA | 0xAA | 0xAA | 0xAA | 0xAA |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x774 | 0x61 | 0x41 | 0x55 | 0x50 | 0x00 | --- | --- | --- |

### Digital Output

**N.A**

## Key

### Read the Key when KEY Pressed

1. **The following command can be used to get Key when Key is pressed.**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x772 | 0x30 | 0x05 | 0xAA | 0xAA | 0xAA | 0xAA | 0xAA | 0xAA |

**The ECU answer with the acknowledgement sent as following:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x774 | 0x70 | 0x05 | 0xXX | --- | --- | --- | --- | --- |

**The request command of getting key test should be sent circularly every 50ms during the test.**

When the Key is pressed, every key has its bit, and then the value will be as following:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Data field** | | **Read Key Description** | **Operation** | **Value** | **Test Point**  **(MCU Port)** |
| Data2 | Bit.1 / PS0 | 2up\_KeySW2 | No / Pressed | 4.8~5.2V / 0~0.3V | F220 (Bottom) |

## **LED Test**

### Bus LED Test

**The following command can be used to control the Led.**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x772 | 0x30 | 0x00 | 0xXX | 0xXX | 0xAA | 0xAA | 0xAA | 0xAA |

The Control Sub Type means index of LEDs, when it is Data2 = 0x80, if Data3 is 1, it means all LED on, if Data3 is 0, it means all LED off.

**The ECU answer with the acknowledgement sent as following:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x774 | 0x70 | 0x00 | 0x01 | -- | -- | -- | -- | -- |

**Note: When send this command， the LED PWM will be set as 100%.**

Table below shows Control Sub Type relations with LED telltale.

| **Data2** | **Sorting** | **Description** | **Color** | **ICON** | **Test Point（ends）** | **Top / Bottom** | **Test Voltage** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | D300 | ChargeCordConnected | Red |  | F307 | TOP | 1.8V~2.3V  TYP@2.0V |
|  |  |  |  |  | F319 | TOP |  |
| 1 | D301 | TractionBatteryCharging | Yellow |  | F308 | TOP | 1.9V~2.4V  TYP@2.0V |
|  |  |  |  |  | F320 | TOP |  |
| 2 | D302 | DriverBelt | Red |  | F309 | TOP | 1.8V~2.3V  TYP@2.0V |
|  |  |  |  |  | F321 | TOP |  |
| 3 | D303 | TractionBatteryFailed | Red |  | F310 | TOP | 1.8V~2.3V  TYP@2.0V |
|  |  |  |  |  | F322 | BOTTOM |  |
| 4 | D304 | TractionBatterycutoff | Yellow |  | F311 | TOP | 1.9V~2.4V  TYP@2.0V |
|  |  |  |  |  | F323 | BOTTOM |  |
| 5 | D305 | Battery | Red |  | F312 | TOP | 1.8V~2.3V  TYP@2.0V |
|  |  |  |  |  | F324 | TOP |  |
| 6 | D306 | DoorOpen | Red |  | F313 | BOTTOM | 1.8V~2.3V  TYP@2.0V |
|  |  |  |  |  | F325 | BOTTOM |  |
| 7 | D307 | HighBeam | Blue |  | F314 | BOTTOM | 3.0V~4.1V  TYP@3.2V |
|  |  |  |  |  | F326 | BOTTOM |  |
| 8 | D308 | LowBeam | Green |  | F315 | BOTTOM | 2.7V~3.7V  TYP@3.0V |
|  |  |  |  |  | F327 | BOTTOM |  |
| ~~9~~ | D309 | LightOn | Green |  | F316 | BOTTOM | 2.7V~3.7V  TYP@3.0V |
|  |  |  |  |  | F328 | BOTTOM |  |
| 10 | D310 | RearFog | Yellow |  | F317 | BOTTOM | 1.9V~2.4V  TYP@2.0V |
|  |  |  |  |  | F329 | BOTTOM |  |
| 11 | D323 | EPS | Yellow |  | **F351** | BOTTOM | 1.9V~2.4V  TYP@2.0V |
|  |  |  |  |  | F363 | BOTTOM |  |
| 12 | D312 | TurnLeft | Green |  | F340 | TOP | 2.7V~3.7V  TYP@3.0V |
|  |  |  |  |  | F352 | BOTTOM |  |
| 13 | D374 | Ready | Green |  | F3144 | BOTTOM | 2.7V~3.7V  TYP@3.0V |
|  |  |  |  |  | F341 | TOP |  |
|  | D313 |  |  |  | F353 | BOTTOM |  |
| 14 | D314 | TurnRight | Green |  | F342 | TOP | 2.7V~3.7V  TYP@3.0V |
|  |  |  |  |  | F354 | BOTTOM |  |
| 15 | D315 | CripplingModeOn | Yellow |  | F343 | TOP | 1.9V~2.4V  TYP@2.0V |
|  |  |  |  |  | F355 | BOTTOM |  |
| 16 | D316 | ElectricalEngineOverheated | Red |  | F344 | TOP | 1.8V~2.3V  TYP@2.0V |
|  |  |  |  |  | F356 | BOTTOM |  |
| 17 | D317 | BrakeFailure | Red |  | F345 | TOP | 1.8V~2.3V  TYP@2.0V |
|  |  |  |  |  | F357 | TOP |  |
| 18 | D318 | SystemFailure | Red |  | F346 | TOP | 1.8V~2.3V  TYP@2.0V |
|  |  |  |  |  | F358 | TOP |  |
| 19 | D319 | AirBag | Red |  | F347 | TOP | 1.8V~2.3V  TYP@2.0V |
|  |  |  |  |  | F359 | BOTTOM |  |
| 20 | D320 | ABS | Yellow |  | F348 | TOP | 1.9V~2.4V  TYP@2.0V |
|  |  |  |  |  | F360 | BOTTOM |  |
| 21 | D321 | CoolantTempHigh | Red |  | F349 | BOTTOM | 1.8V~2.3V  TYP@2.0V |
|  |  |  |  |  | F361 | BOTTOM |  |
| 22 | D322 | ParkBrake | Red |  | F350 | BOTTOM | 1.8V~2.3V  TYP@2.0V |
|  |  |  |  |  | F362 | TOP |  |
| ~~23~~ | ~~D311~~ | ~~Body\_Immobi~~ | ~~Red~~ |  | ~~F318~~ | ~~BOTTOM~~ | ~~1.8V~2.3V~~  ~~TYP@2.0V~~ |
|  |  |  |  |  | ~~F330~~ | ~~TOP~~ |  |
| 128  (0x80) |  | All led status control 1 |  |  |  |  |  |

The data field is defined as following table:

|  |  |  |
| --- | --- | --- |
| **Data3** | **Description** | **LED Status** |
| 0x01 | ON | LED On |
| 0x00 | OFF | LED Off |

Remark:

1. In case of Control Sub Type equal to 0x80, Data0 contrtols all Led ONOFF, if Data3 is 0, then all Leds OFF, if other than 0, all Leds On.

### Backlight LED Test

• 1 – Set Illumination On

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x772 | 0x30 | 0x00 | 0x81 | 0x01 | 0xAA | 0xAA | 0xAA | 0xAA |

The ECU answer with the acknowledgement sent as following:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x774 | 0x70 | 0x00 | 0x01 | --- | --- | --- | --- | --- |

All backlight LEDs are on after the command is send.

LED backlight

| **Sorting** | **Description** | **Color** | **Test Point** | **Top /Bottom** | **Test Voltage** |
| --- | --- | --- | --- | --- | --- |
| D370 | Pointer Backlight | White | F128 | BOTTOM | 2.8V~3.5V  TYP @3.0V |
| F3134 | TOP |
| D372 |
| F3136 | TOP |
| D401 | Pointer Backlight | White | F395 | TOP | 2.8V~3.5V  TYP @3.0V |
| F398 | BOTTOM |
| D371 | Pointer Backlight | White | F128 | BOTTOM | 2.8V~3.5V  TYP @3.0V |
| F3135 | TOP |
| D373 | F3137 | TOP |
| D402 | Pointer Backlight | White | F128 | BOTTOM | 2.8V~3.5V  TYP @3.0V |
| F3140 | TOP |
| F3141 | TOP |
| D342 | Dail Backlight | White | F128 | BOTTOM | 2.7V~3.5V  TYP @3.0V |
| F3102 | TOP |
| D351 | F3111 | BOTTOM |
| D343 | Dail Backlight | White | F128 | BOTTOM | 2.7V~3.5V  TYP @3.0V |
| F3103 | TOP |
| D352 | F3112 | TOP |
| D344 | Dail Backlight | White | F128 | TOP | 2.7V~3.5V  TYP @3.0V |
| F3104 | TOP |
| D353 | F3113 | TOP |
| D3108 | Dail Backlight | White | F3110 | TOP | 2.7V~3.5V  TYP @3.0V |
| F3109 | TOP |
| D345 | Dail Backlight | White | F128 | BOTTOM | 2.7V~3.5V  TYP @3.0V |
| D354 | F3105 | BOTTOM |
| F3114 | BOTTOM |
| D346 | Dail Backlight | White | F128 | BOTTOM | 2.7V~3.5V  TYP @3.0V |
| F3106 | TOP |
| D355 | F3115 | BOTTOM |
| D347 | Dail Backlight | White | F128 | BOTTOM | 2.7V~3.5V  TYP @3.0V |
| F3107 | BOTTOM |
| D356 | F3116 | BOTTOM |
| D357 | Dail Backlight | White | F3108 | BOTTOM | 2.7V~3.5V  TYP @3.0V |
| F3117 | BOTTOM |
| D324 | LCD Backlight | White | F128 | BOTTOM | 2.7V~3.5V  TYP @3.0V |
| F369 | TOP |
| D333 |
| F378 | TOP |
| D325 | LCD Backlight | White | F128 | BOTTOM | 2.7V~3.5V  TYP @3.0V |
| D334 | F370 | BOTTOM |
| F379 | TOP |
| D326 | LCD Backlight | White | F128 | BOTTOM | 2.7V~3.5V  TYP @3.0V |
| D335 | F371 | TOP |
| F380 | TOP |
| D327 | LCD Backlight | White | F128 | BOTTOM | 2.7V~3.5V  TYP @3.0V |
| F372 | BOTTOM |
| D336 |
| F381 | TOP |
| D328 | LCD Backlight | White | F128 | BOTTOM | 2.7V~3.5V  TYP @3.0V |
| D337 | F373 | BOTTOM |
| F382 | TOP |
| D329 | LCD Backlight | White | F128 | BOTTOM | 2.7V~3.5V  TYP @3.0V |
| F374 | TOP |
| D338 | F383 | TOP |
| D330 | LCD Backlight | White | F128 | BOTTOM | 2.7V~3.5V  TYP @3.0V |
| F375 | BOTTOM |
| D339 | F384 | TOP |
| D340 | LCD Backlight | White | F376 | TOP | 2.7V~3.5V  TYP @3.0V |
| F385 | TOP |

• 2 – Set Illumination Off

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x772 | 0x30 | 0x00 | 0x81 | 0x00 | 0xAA | 0xAA | 0xAA | 0xAA |

The ECU answer with the acknowledgement sent as following:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x774 | 0x70 | 0x00 | 0x01 | --- | --- | --- | --- | --- |

All backlight LEDs are off after the command is send.

## **Peripheral Test**

### **Movement Outputs Test**

OBJECT:

Test gauges driver outputs.

METHOD:

Measure voltages of the step motor ports.

PROCEDURE:

All loads/values applied as per specification in the following table.

NOTE:

All voltages and resistances are relative to ground unless otherwise stated. The tolerance of wave form measurement can be ±(TBD)% of indicated values.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Point  (Apply) | Apply | Description | Test Point  (Measure) | Value | Measure |
| F109 | 13.5VDC | Battery | F109(Top) | 13.5V±0.2V |  |
| F110 | GND | GND | F110(Bottom) | 0V |  |
| F108 | 13.5VDC | Ignition | F108(Bottom) | 13.5V±0.2V |  |
| F100 | CANH | CANH | F100(Bottom) |  |  |
| F101 | CANL | CANL | F101(Bottom) |  |  |

Diagnostic Telegrams:

**The following command can be used to control the motor Pin.**

• **1** – Disable Normal Gauges

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x772 | 0x30 | 0x01 | 0x30 | 0x01 | 0xAA | 0xAA | 0xAA | 0xAA |

**The ECU answer with the acknowledgement sent as following:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x774 | 0x70 | 0x01 | 0x01 | -- | -- | -- | -- | -- |

• **2**– Gauges test

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x772 | 0x30 | 0x01 | 0x31 | X1 | 0xAA | 0xAA | 0xAA | 0xAA |

**The ECU answer with the acknowledgement sent as following:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x774 | 0x70 | 0x01 | 0x01 | -- | -- | -- | -- | -- |

There are 2 Execution bytes in request command from tester, the format as follow:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Byte | Bit | Description | Port | Test points | Top / Bottom | Value |
| Execution Byte1  (X1) | 7 | Power SM Port 4 | Pin9 | F207 | Top | 1: ON / 0: OFF |
| 6 | Power SM Port 3 | Pin10 | F208 | Top | 1: ON / 0: OFF |
| 5 | Power SM Port 2 | Pin11 | F209 | Top | 1: ON / 0: OFF |
| 4 | Power SM Port 1 | Pin12 | F210 | Top | 1: ON / 0: OFF |
| 3 | SOC SM Port 4 | Pin15 | F211 | Top | 1: ON / 0: OFF |
| 2 | SOC SM Port 3 | Pin16 | F212 | Top | 1: ON / 0: OFF |
| 4 | SOC SM Port 2 | Pin17 | F213 | Top | 1: ON / 0: OFF |
| 0 | SOC SM Port 1 | Pin18 | F214 | Top | 1: ON / 0: OFF |

Note: Port in the table means MCU Pins.

1.Testing Power SM Ports:

Step 1

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x772 | 0x30 | 0x01 | 0x31 | 0xF0 | 0xAA | 0xAA | 0xAA | 0xAA |

**The ECU answer with the acknowledgement sent as following:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x774 | 0x70 | 0x01 | 0x01 | -- | -- | -- | -- | -- |

Test the test point

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Action | Waiting time | Measuring point | Reaction | Note |
| Power Port 4 to "1"  Power Port 3 to "1"  Power Port 2 to "1"  Power Port 1 to "1" | >30ms | F207  F208  F209  F210 | **>4.5V DC**  **>4.5V DC**  **>4.5V DC**  **>4.5V DC** |  |

Step 2

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x772 | 0x30 | 0x01 | 0x31 | 0x70 | 0xAA | 0xAA | 0xAA | 0xAA |

**The ECU answer with the acknowledgement sent as following:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x774 | 0x70 | 0x01 | 0x01 | -- | -- | -- | -- | -- |

Test the test point

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Action | Waiting time | Measuring point | Reaction | Note |
| Power Port 4 to "0"  Power Port 3 to "1"  Power Port 2 to "1"  Power Port 1 to "1" | >30ms | F207  F208  F209  F210 | **<0.5V DC**  **>4.5V DC**  **>4.5V DC**  **>4.5V DC** |  |

Step 3

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x772 | 0x30 | 0x01 | 0x31 | 0xB0 | 0xAA | 0xAA | 0xAA | 0xAA |

**The ECU answer with the acknowledgement sent as following:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x774 | 0x70 | 0x01 | 0x01 | -- | -- | -- | -- | -- |

Test the test point

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Action | Waiting time | Measuring point | Reaction | Note |
| Power Port 4 to "1"  Power Port 3 to "0"  Power Port 2 to "1"  Power Port 1 to "1" | >30ms | F207  F208  F209  F210 | **>4.5V DC**  **<0.5V DC**  **>4.5V DC**  **>4.5V DC** |  |

Step 4

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x772 | 0x30 | 0x01 | 0x31 | 0xD0 | 0xAA | 0xAA | 0xAA | 0xAA |

**The ECU answer with the acknowledgement sent as following:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x774 | 0x70 | 0x01 | 0x01 | -- | -- | -- | -- | -- |

Test the test point

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Action | Waiting time | Measuring point | Reaction | Note |
| Power Port 4 to "1"  Power Port 3 to "1"  Power Port 2 to "0"  Power Port 1 to "1" | >30ms | F207  F208  F209  F210 | **>4.5V DC**  **>4.5V DC**  **<0.5V DC**  **>4.5V DC** |  |

Step 5

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x772 | 0x30 | 0x01 | 0x31 | 0xE0 | 0xAA | 0xAA | 0xAA | 0xAA |

**The ECU answer with the acknowledgement sent as following:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x774 | 0x70 | 0x01 | 0x01 | -- | -- | -- | -- | -- |

Test the test point

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Action | Waiting time | Measuring point | Reaction | Note |
| Power Port 4 to "1"  Power Port 3 to "1"  Power Port 2 to "1"  Power Port 1 to "0" | >30ms | F207  F208  F209  F210 | **>4.5V DC**  **>4.5V DC**  **>4.5V DC**  **<0.5V DC** |  |

2. Testing SOC SM Ports:

Step 1

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x772 | 0x30 | 0x01 | 0x31 | 0xF0 | 0xAA | 0xAA | 0xAA | 0xAA |

**The ECU answer with the acknowledgement sent as following:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x774 | 0x70 | 0x01 | 0x01 | -- | -- | -- | -- | -- |

Test the test point

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Action | Waiting time | Measuring point | Reaction | Note |
| SOC Port 4 to "1"  SOC Port 3 to "1"  SOC Port 2 to "1"  SOC Port 1 to "1" | >30ms | F211  F212  F213  F214 | **>4.5V DC**  **>4.5V DC**  **>4.5V DC**  **>4.5V DC** |  |

Step 2

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x772 | 0x30 | 0x01 | 0x31 | 0x70 | 0xAA | 0xAA | 0xAA | 0xAA |

**The ECU answer with the acknowledgement sent as following:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x774 | 0x70 | 0x01 | 0x01 | -- | -- | -- | -- | -- |

Test the test point

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Action | Waiting time | Measuring point | Reaction | Note |
| SOC Port 4 to "0"  SOC Port 3 to "1"  SOC Port 2 to "1"  SOC Port 1 to "1" | >30ms | F211  F212  F213  F214 | **<0.5V DC**  **>4.5V DC**  **>4.5V DC**  **>4.5V DC** |  |

Step 3

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x772 | 0x30 | 0x01 | 0x31 | 0xB0 | 0xAA | 0xAA | 0xAA | 0xAA |

**The ECU answer with the acknowledgement sent as following:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x774 | 0x70 | 0x01 | 0x01 | -- | -- | -- | -- | -- |

Test the test point

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Action | Waiting time | Measuring point | Reaction | Note |
| SOC Port 4 to "1"  SOC Port 3 to "0"  SOC Port 2 to "1"  SOC Port 1 to "1" | >30ms | F211  F212  F213  F214 | **>4.5V DC**  **<0.5V DC**  **>4.5V DC**  **>4.5V DC** |  |

Step 4

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x772 | 0x30 | 0x01 | 0x31 | 0xD0 | 0xAA | 0xAA | 0xAA | 0xAA |

**The ECU answer with the acknowledgement sent as following:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x774 | 0x70 | 0x01 | 0x01 | -- | -- | -- | -- | -- |

Test the test point

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Action | Waiting time | Measuring point | Reaction | Note |
| SOC Port 4 to "1"  SOC Port 3 to "1"  SOC Port 2 to "0"  SOC Port 1 to "1" | >30ms | F211  F212  F213  F214 | **>4.5V DC**  **>4.5V DC**  **<0.5V DC**  **>4.5V DC** |  |

Step 5

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x772 | 0x30 | 0x01 | 0x31 | 0xE0 | 0xAA | 0xAA | 0xAA | 0xAA |

**The ECU answer with the acknowledgement sent as following:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x774 | 0x70 | 0x01 | 0x01 | -- | -- | -- | -- | -- |

Test the test point

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Action | Waiting time | Measuring point | Reaction | Note |
| SOC Port 4 to "1"  SOC Port 3 to "1"  SOC Port 2 to "1"  SOC Port 1 to "0" | >30ms | F211  F212  F213  F214 | **>4.5V DC**  **>4.5V DC**  **>4.5V DC**  **<0.5V DC** |  |

•3– Enable Normal Gauges

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x772 | 0x30 | 0x01 | 0x30 | 0x00 | 0xAA | 0xAA | 0xAA | 0xAA |

**The ECU answer with the acknowledgement sent as following:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x774 | 0x70 | 0x01 | 0x01 | -- | -- | -- | -- | -- |

### **Buzzer Operation**

**The following command can be used to control the Buzzer.**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x772 | 0x30 | 0x04 | 0x00 | 0xXX | 0xAA | 0xAA | 0xAA | 0xAA |

The ECU answer with the acknowledgement sent as following:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x774 | 0x70 | 0x04 | 0xXX | --- | --- | --- | --- | --- |

Control Sub Type is not used in this case.

When Data3=1 Beep ON, Data3=0 Beeper OFF.

When Beep ON,

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Point  (Apply) | Apply | Description | Test Point  (Measure) | Value | Measure |
| F109 | 13.5VDC | Battery | F109 | 13.5V |  |
| F110 | GND | GND | F110 | 0V |  |
| F108 | 13.5VDC | Ignition | F108 | 13.5V |  |
| F100 | CANH | CANH |  |  |  |
| F101 | CANL | CANL |  |  |  |
|  |  | Buzzer + | F437(TOP) | 7~13V |  |
|  |  | Buzzer - | F438(TOP) | 0~2V |  |

## ADC Calibration

N.A

## **Pointer Calibration**

### **Pressure Pointer**

指针压入力范围：30N-110N；

指针压入高度：距离表盘上表面高度1.25+/-0.25mm

指针压入速度：1.0mm/s

### **Pre-Actions**

Set Illumination on, check all the LEDs on the PCB.

• TT1 – Set Illumination On

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x772 | 0x30 | 0x00 | 0x81 | 0x01 | 0xAA | 0xAA | 0xAA | 0xAA |

The ECU answer with the acknowledgement sent as following:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x774 | 0x70 | 0x00 | 0x01 | --- | --- | --- | --- | --- |

If all the Illumination LEDs can be set on, turn off them.

• TT2 – Set Illumination Off

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x772 | 0x30 | 0x00 | 0x81 | 0x00 | 0xAA | 0xAA | 0xAA | 0xAA |

The ECU answer with the acknowledgement sent as following:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x774 | 0x70 | 0x00 | 0x01 | --- | --- | --- | --- | --- |

Then do the pointer calibration.

### **Calibration**

OBJECT:

Enable operators to check and correct the pointer zero point positions manually (by press UP/DOWN button).

Adjust zero point offsets in EEPROM, if necessary

METHOD:

Send messages via CAN interface to set gauge to zero point, and adjust EEPROM contents according to UP/DOWN steps & "OK" applied by the operator.

16step/1o

PROCEDURE:

1. Supply the Power Kl.30 & Kl.15 and Ground to the Cluster.
2. Wait until pointer goto mechanical zore point finish(about 3s).
3. Entry the Diagnosis session.
4. Disconnect Kl.15, send the check “Quiescent Current test” command, let the Cluster goto sleep.
5. Supply the Power Kl.15, enter diagnose session.
6. Start the Calibration.

#### **Power Gauge**

Diagnostic Telegrams:

• **1** – Get the number of calibrate points

**The following command can be used to get the Number of Calibrate points.**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x772 | 0x1B | 0x00 | 0x01 | 0x44 | 0xAA | 0xAA | 0xAA | 0xAA |

**The ECU answer with the acknowledgement sent as following:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x774 | 0x5B | 0x00 | 0x01 | 0x44 | 0x04 | -- | -- | -- |

**Note: Data4 is the Number of calibrate points.**

• 2 – Began to calibrate

1. The first calibrate point (“OFF”)

–Step 1 Get the calibrate point step

**The following command can be used to Get the first calibrate point step.**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x772 | 0x1B | 0x00 | 0x01 | 0x04 | 0x00 | 0xAA | 0xAA | 0xAA |

**The ECU answer with the acknowledgement sent as following:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x774 | 0x5B | 0x00 | 0x01 | 0x04 | 0x00 | 0xXX | 0xXX | -- |

**Note: Data5 and Data6 are the steps.**

**Data5 = 0 and Data6 = 0, step=0.**

–Step 2 Gauge Movement

**The following command can be used to move the step motor.**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x772 | 0x30 | 0x01 | 0x04 | 0xXX | 0xXX | 0xAA | 0xAA | 0xAA |

**Note: Data3 and Data4 are the steps.**

**Data3 = 0 and Data4 = 0, step=0.**

**The ECU answer with the acknowledgement sent as following:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x774 | 0x70 | 0x01 | 0x01 | -- | -- | -- | -- | -- |

–Step 3 Write the Motor steps

**The following command can be used to write the motor setp.**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x772 | 0x1B | 0x01 | 0x01 | 0x04 | 0x00 | 0xXX | 0xXX | 0xAA |

**Note: Data5 and Data6 are the steps.**

**Data5 = 0 and Data6 = 0, step=0.**

**The ECU answer with the acknowledgement sent as following:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x774 | 0x5B | 0x01 | 0x01 | 0x04 | 0x00 | 0x01 | -- | -- |

1. The second calibrate point (“Green Region”)

–Step 1 Get the calibrate point step

**The following command can be used to Get the first calibrate point step.**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x772 | 0x1B | 0x00 | 0x01 | 0x04 | 0x01 | 0xAA | 0xAA | 0xAA |

**The ECU answer with the acknowledgement sent as following:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x774 | 0x5B | 0x00 | 0x01 | 0x04 | 0x01 | 0xXX | 0xXX | -- |

**Note: Data5 and Data6 are the steps.**

**Data5 = 0 and Data6 = 0, step=0.**

–Step 2 Gauge Movement

**The following command can be used to move the step motor.**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x772 | 0x30 | 0x01 | 0x04 | 0xXX | 0xXX | 0xAA | 0xAA | 0xAA |

**Note: Data3 and Data4 are the steps.**

**Data3 = 0 and Data4 = 0, step=0.**

**The ECU answer with the acknowledgement sent as following:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x774 | 0x70 | 0x01 | 0x01 | -- | -- | -- | -- | -- |

–Step 3 Write the Motor steps

**The following command can be used to write the motor setp.**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x772 | 0x1B | 0x01 | 0x01 | 0x04 | 0x01 | 0xXX | 0xXX | 0xAA |

**Note: Data5 and Data6 are the steps.**

**Data5 = 0 and Data6 = 0, step=0.**

**The ECU answer with the acknowledgement sent as following:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x774 | 0x5B | 0x01 | 0x01 | 0x04 | 0x01 | 0x01 | -- | -- |

1. The thirdly calibrate point (“0%”)

–Step 1 Get the calibrate point step

**The following command can be used to Get the first calibrate point step.**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x772 | 0x1B | 0x00 | 0x01 | 0x04 | 0x02 | 0xAA | 0xAA | 0xAA |

**The ECU answer with the acknowledgement sent as following:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x774 | 0x5B | 0x00 | 0x01 | 0x04 | 0x02 | 0xXX | 0xXX | -- |

**Note: Data5 and Data6 are the steps.**

**Data5 = 0 and Data6 = 0, step=0.**

–Step 2 Gauge Movement

**The following command can be used to move the step motor.**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x772 | 0x30 | 0x01 | 0x04 | 0xXX | 0xXX | 0xAA | 0xAA | 0xAA |

**Note: Data3 and Data4 are the steps.**

**Data3 = 0 and Data4 = 0, step=0.**

**The ECU answer with the acknowledgement sent as following:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x774 | 0x70 | 0x01 | 0x01 | -- | -- | -- | -- | -- |

–Step 3 Write the Motor steps

**The following command can be used to write the motor setp.**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x772 | 0x1B | 0x01 | 0x01 | 0x04 | 0x02 | 0xXX | 0xXX | 0xAA |

**Note: Data5 and Data6 are the steps.**

**Data5 = 0 and Data6 = 0, step=0.**

**The ECU answer with the acknowledgement sent as following:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x774 | 0x5B | 0x01 | 0x01 | 0x04 | 0x02 | 0x01 | -- | -- |

1. The fourth calibrate point (“100%”)

–Step 1 Get the calibrate point step

**The following command can be used to Get the first calibrate point step.**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x772 | 0x1B | 0x00 | 0x01 | 0x04 | 0x03 | 0xAA | 0xAA | 0xAA |

**The ECU answer with the acknowledgement sent as following:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x774 | 0x5B | 0x00 | 0x01 | 0x04 | 0x03 | 0xXX | 0xXX | -- |

**Note: Data5 and Data6 are the steps.**

**Data5 = 0 and Data6 = 0, step=0.**

–Step 2 Gauge Movement

**The following command can be used to move the step motor.**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x772 | 0x30 | 0x01 | 0x04 | 0xXX | 0xXX | 0xAA | 0xAA | 0xAA |

**Note: Data3 and Data4 are the steps.**

**Data3 = 0 and Data4 = 0, step=0.**

**The ECU answer with the acknowledgement sent as following:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x774 | 0x70 | 0x01 | 0x01 | -- | -- | -- | -- | -- |

–Step 3 Write the Motor steps

**The following command can be used to write the motor setp.**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x772 | 0x1B | 0x01 | 0x01 | 0x04 | 0x03 | 0xXX | 0xXX | 0xAA |

**Note: Data5 and Data6 are the steps.**

**Data5 = 0 and Data6 = 0, step=0.**

**The ECU answer with the acknowledgement sent as following:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x774 | 0x5B | 0x01 | 0x01 | 0x04 | 0x03 | 0x01 | -- | -- |

#### **SOC Gauge**

Diagnostic Telegrams:

• **1** – Get the number of calibrate points

**The following command can be used to get the Number of Calibrate points.**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x772 | 0x1B | 0x00 | 0x01 | 0x45 | 0xAA | 0xAA | 0xAA | 0xAA |

**The ECU answer with the acknowledgement sent as following:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x774 | 0x5B | 0x00 | 0x01 | 0x45 | 0x03 | -- | -- | -- |

**Note: Data4 is the Number of calibrate points**

• 2 – Began to calibrate

1. The first calibrate point (“0%”)

–Step 1 Get the calibrate point step

**The following command can be used to Get the first calibrate point step.**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x772 | 0x1B | 0x00 | 0x01 | 0x05 | 0x00 | 0xAA | 0xAA | 0xAA |

**The ECU answer with the acknowledgement sent as following:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x774 | 0x5B | 0x00 | 0x01 | 0x05 | 0x00 | 0xXX | 0xXX | -- |

**Note: Data5 and Data6 are the steps.**

**Data5 = 0 and Data6 = 0, step=0.**

–Step 2 Gauge Movement

**The following command can be used to move the step motor.**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x772 | 0x30 | 0x01 | 0x05 | 0xXX | 0xXX | 0xAA | 0xAA | 0xAA |

**Note: Data3 and Data4 are the steps.**

**Data3 = 0 and Data4 = 0, step=0.**

**The ECU answer with the acknowledgement sent as following:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x774 | 0x70 | 0x01 | 0x01 | -- | -- | -- | -- | -- |

–Step 3 Write the Motor steps

**The following command can be used to write the motor setp.**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x772 | 0x1B | 0x01 | 0x01 | 0x05 | 0x00 | 0xXX | 0xXX | 0xAA |

**Note: Data5 and Data6 are the steps.**

**Data5 = 0 and Data6 = 0, step=0.**

**The ECU answer with the acknowledgement sent as following:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x774 | 0x5B | 0x01 | 0x01 | 0x05 | 0x00 | 0x01 | -- | -- |

1. The second calibrate point (“60%”)

–Step 1 Get the calibrate point step

**The following command can be used to Get the first calibrate point step.**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x772 | 0x1B | 0x00 | 0x01 | 0x05 | 0x01 | 0xAA | 0xAA | 0xAA |

**The ECU answer with the acknowledgement sent as following:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x774 | 0x5B | 0x00 | 0x01 | 0x05 | 0x01 | 0xXX | 0xXX | -- |

**Note: Data5 and Data6 are the steps.**

**Data5 = 0 and Data6 = 0, step=0.**

–Step 2 Gauge Movement

**The following command can be used to move the step motor.**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x772 | 0x30 | 0x01 | 0x05 | 0xXX | 0xXX | 0xAA | 0xAA | 0xAA |

**Note: Data3 and Data4 are the steps.**

**Data3 = 0 and Data4 = 0, step=0.**

**The ECU answer with the acknowledgement sent as following:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x774 | 0x70 | 0x01 | 0x01 | -- | -- | -- | -- | -- |

–Step 3 Write the Motor steps

**The following command can be used to write the motor setp.**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x772 | 0x1B | 0x01 | 0x01 | 0x05 | 0x01 | 0xXX | 0xXX | 0xAA |

**Note: Data5 and Data6 are the steps.**

**Data5 = 0 and Data6 = 0, step=0.**

**The ECU answer with the acknowledgement sent as following:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x774 | 0x5B | 0x01 | 0x01 | 0x05 | 0x01 | 0x01 | -- | -- |

1. The thirdly calibrate point (“100%”)

–Step 1 Get the calibrate point step

**The following command can be used to Get the first calibrate point step.**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x772 | 0x1B | 0x00 | 0x01 | 0x05 | 0x02 | 0xAA | 0xAA | 0xAA |

**The ECU answer with the acknowledgement sent as following:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x774 | 0x5B | 0x00 | 0x01 | 0x05 | 0x02 | 0xXX | 0xXX | -- |

**Note: Data5 and Data6 are the steps.**

**Data5 = 0 and Data6 = 0, step=0.**

–Step 2 Gauge Movement

**The following command can be used to move the step motor.**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x772 | 0x30 | 0x01 | 0x05 | 0xXX | 0xXX | 0xAA | 0xAA | 0xAA |

**Note: Data3 and Data4 are the steps.**

**Data3 = 0 and Data4 = 0, step=0.**

**The ECU answer with the acknowledgement sent as following:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x774 | 0x70 | 0x01 | 0x01 | -- | -- | -- | -- | -- |

–Step 3 Write the Motor steps

**The following command can be used to write the motor setp.**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x772 | 0x1B | 0x01 | 0x01 | 0x05 | 0x02 | 0xXX | 0xXX | 0xAA |

**Note: Data5 and Data6 are the steps.**

**Data5 = 0 and Data6 = 0, step=0.**

**The ECU answer with the acknowledgement sent as following:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x774 | 0x5B | 0x01 | 0x01 | 0x05 | 0x02 | 0x01 | -- | -- |

### **Functionality Check**

#### **Power Gauge**

Diagnostic Telegrams:

• 1 –The first calibrate point (“0%”)

**The following command can be used to move the step motor.**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x772 | 0x30 | 0x01 | 0x14 | 0x00 | 0xFE | 0xAA | 0xAA | 0xAA |

**Note: Data3 and Data4 are the value.**

**Data3 = 0 and Data4 = FE, value=254.**

**The ECU answer with the acknowledgement sent as following:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x774 | 0x70 | 0x01 | 0x01 | -- | -- | -- | -- | -- |

• 2 –The second calibrate point (“50%”)

**The following command can be used to move the step motor.**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x772 | 0x30 | 0x01 | 0x14 | 0x01 | 0x7E | 0xAA | 0xAA | 0xAA |

**The ECU answer with the acknowledgement sent as following:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x774 | 0x70 | 0x01 | 0x01 | -- | -- | -- | -- | -- |

• 3 –The thirdly calibrate point (“100%”)

**The following command can be used to move the step motor.**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x772 | 0x30 | 0x01 | 0x14 | 0x01 | 0xFD | 0xAA | 0xAA | 0xAA |

**The ECU answer with the acknowledgement sent as following:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x774 | 0x70 | 0x01 | 0x01 | -- | -- | -- | -- | -- |

#### **SOC Gauge**

Diagnostic Telegrams:

• 1 –The first calibrate point (“20%”)

**The following command can be used to move the step motor.**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x772 | 0x30 | 0x01 | 0x15 | 0x00 | 0x32 | 0xAA | 0xAA | 0xAA |

**Note: Data3 and Data4 are the value.**

**Data3 = 0 and Data4 = FE, value=254.**

**The ECU answer with the acknowledgement sent as following:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x774 | 0x70 | 0x01 | 0x01 | -- | -- | -- | -- | -- |

• 2 –The second calibrate point (“80%”)

**The following command can be used to move the step motor.**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x772 | 0x30 | 0x01 | 0x15 | 0x00 | 0xCB | 0xAA | 0xAA | 0xAA |

**The ECU answer with the acknowledgement sent as following:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x774 | 0x70 | 0x01 | 0x01 | -- | -- | -- | -- | -- |

• 3 –The thirdly calibrate point (“100%”)

**The following command can be used to move the step motor.**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x772 | 0x30 | 0x01 | 0x15 | 0x00 | 0xFE | 0xAA | 0xAA | 0xAA |

**The ECU answer with the acknowledgement sent as following:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x774 | 0x70 | 0x01 | 0x01 | -- | -- | -- | -- | -- |

## **Stress Test**

As described in EEPROM Layout, Group "SPECIAL" Module "StressTest" defines the flag and time of stress test.

wStressTestFlag – Flag of stress test mode, a fixed flag 0xA55A

wStressTestTime – Time setting of stress test, in second unit

Write these contents after pointer calibration, Stress test mode will be entered after power up, and keep running in the defined time, the flag and time in EEPROM will be cleared if timeout, and normal mode will be entered after reset.

• TT1 – Write Memory for Stress Test:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x772 | 0x3C | 0x01 | X1 | X2 | 0xAA | 0xAA | 0xAA | 0xAA |

The ECU answer with the acknowledgement sent as following:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x774 | 0x7C | 0x01 | --- | --- | --- | --- | --- | --- |

X1—Stress Test time Data (High Byte) unit Seconds

X2—Stress Test time Data (Low Byte) unit Seconds

## **LCD Test**

### Segment Display

N.A

### LCD Pin Test

N.A

### TFT Display

**• 1 – Illumination On**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x772 | 0x30 | 0x00 | 0x81 | 0x01 | 0xAA | 0xAA | 0xAA | 0xAA |

**The ECU answer with the acknowledgement sent as following:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x774 | 0x70 | 0x00 | 0x81 | --- | --- | --- | --- | --- |

**• 2 – Display Test**

**Step 1**

**The following command can be used to control the LCD display.**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x772 | 0x30 | 0x03 | 0x01 | 0x01 | 0xAA | 0xAA | 0xAA | 0xAA |

**The ECU answer with the acknowledgement sent as following:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x774 | 0x70 | 0x03 | --- | --- | --- | --- | --- | --- |

Display Bright Gray:



**Step 2**

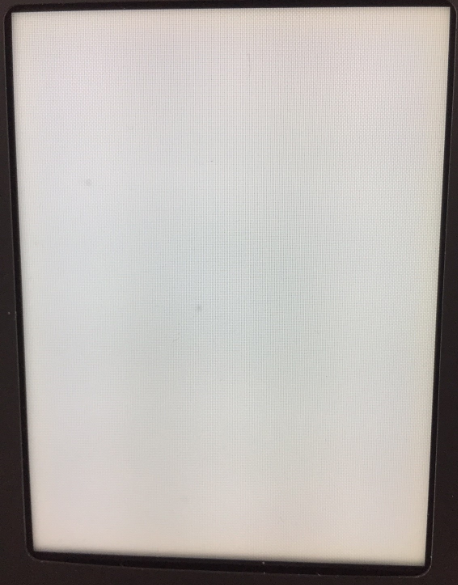
**The following command can be used to control the LCD display.**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x772 | 0x30 | 0x03 | 0x02 | 0x01 | 0xAA | 0xAA | 0xAA | 0xAA |

**The ECU answer with the acknowledgement sent as following:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x774 | 0x70 | 0x03 | --- | --- | --- | --- | --- | --- |

Display Dark Gray:



**Step 3**

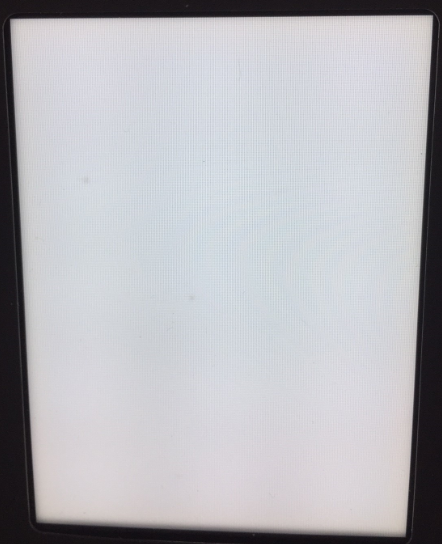
**The following command can be used to control the LCD display.**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x772 | 0x30 | 0x03 | 0x03 | 0x01 | 0xAA | 0xAA | 0xAA | 0xAA |

**The ECU answer with the acknowledgement sent as following:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x774 | 0x70 | 0x03 | --- | --- | --- | --- | --- | --- |

Display White



**Step 4**

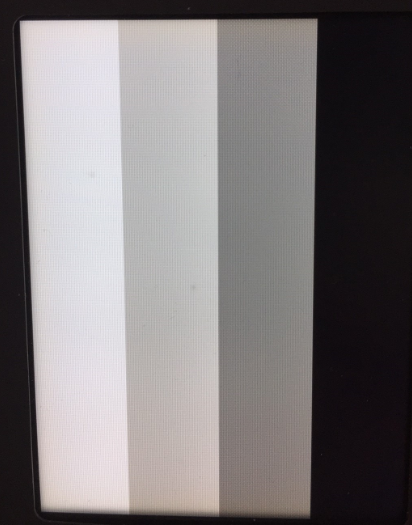
**The following command can be used to control the LCD display.**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x772 | 0x30 | 0x03 | 0x04 | 0x01 | 0xAA | 0xAA | 0xAA | 0xAA |

**The ECU answer with the acknowledgement sent as following:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x774 | 0x70 | 0x03 | --- | --- | --- | --- | --- | --- |

Display All Gray Scale:



**Step 5**

**The following command can be used to control the LCD display.**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x772 | 0x30 | 0x03 | 0x05 | 0x01 | 0xAA | 0xAA | 0xAA | 0xAA |

**The ECU answer with the acknowledgement sent as following:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x774 | 0x70 | 0x03 | --- | --- | --- | --- | --- | --- |

Display black:



### Dot Display

N.A

## EOL

### Power Gauge

OBJECT:

Check Velocity gauge

METHOD:

Send simulation message drive Velocity gauge and visible check.

PROCEDURE:

All loads/values applied as per specification in the following table.

There are 2 ways can be selected for the convenience of operation.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Point  (Apply) | Apply | Description | Test Point  (Measure) | Value | Measure |
| F109 | 13.5VDC | Battery | F109 | 13.5V |  |
| F110 | GND | GND | F110 | 0V |  |
| F108 | 13.5VDC | Ignition | F108 | 13.5V |  |
| F100 | CANH |  |  |  |  |
| F101 | CANL |  |  |  |  |

**1.** Input **-100%**:

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| CAN ID | DLC | TX Cycle | Message | | | | | | | | Function Description |
| 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 |
| 0x446 | 7 | 100 | 00 | 00 | 00 | 01 | 00 | 00 | 00 | -- |  |

**2.** Input **0%**:

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| CAN ID | DLC | TX Cycle | Message | | | | | | | | Function Description |
| 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 |
| 0x446 | 7 | 100 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | -- |  |

**3.** Input **50%**:

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| CAN ID | DLC | TX Cycle | Message | | | | | | | | Function Description |
| 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 |
| 0x446 | 7 | 100 | 00 | 00 | 00 | 00 | 7F | 00 | 00 | -- |  |

**4.** Input **100%**:

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| CAN ID | DLC | TX Cycle | Message | | | | | | | | Function Description |
| 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 |
| 0x446 | 7 | 100 | 00 | 00 | 00 | 00 | FF | 00 | 00 | -- |  |

### SOC Gauge

OBJECT:

Check Revolution gauge

METHOD:

Send simulation message drive Revolution gauge and visible check.

PROCEDURE:

All loads/values applied as per specification in the following table.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Point  (Apply) | Apply | Description | Test Point  (Measure) | Value | Measure |
| F109 | 13.5VDC | Battery | F109 | 13.5V |  |
| F110 | GND | GND | F110 | 0V |  |
| F108 | 13.5VDC | Ignition | F108 | 13.5V |  |
| F100 | CANH |  |  |  |  |
| F101 | CANL |  |  |  |  |

**1.** Input **20%**:

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| CAN ID | DLC | TX Cycle | Message | | | | | | | | Function Description |
| 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 |
| 0x446 | 7 | 100 | 00 | 00 | 00 | 00 | 00 | 33 | 00 | -- |  |

**2.** Input **80%**:

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| CAN ID | DLC | TX Cycle | Message | | | | | | | | Function Description |
| 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 |
| 0x446 | 7 | 100 | 00 | 00 | 00 | 00 | 00 | CC | 00 | -- |  |

**3.** Input **100%**:

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| CAN ID | DLC | TX Cycle | Message | | | | | | | | Function Description |
| 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 |
| 0x446 | 7 | 100 | 00 | 00 | 00 | 00 | 00 | FF | 00 | -- |  |

## Check Zero Point

Check if Cluster’s pointers point to zero or not.

OBJECT:

Check zero point

METHOD:

PROCEDURE:

Send the sleep command.

Wait 500 ms

Go to Sleep

KL15 OFF

KL30 ON& KL15 ON

KL15 ON

Pass

Normal

Check zero

Point

Manual calibration

Out of range

## Factory Setting

**Factory Setting is subset of** [**4.19.3 EEPROM Special Settings**](#_EEPROM_Special_Settings)**.**

**The following command will be used to setting the ECU to factory settings.**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x772 | 0x3C | 0x02 | 0xAA | 0xAA | 0xAA | 0xAA | 0xAA | 0xAA |

**Ecu Response:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Can ID(STD)** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** | **Data7** |
| 0x774 | 0x7C | 0x02 | --- | --- | --- | --- | --- | --- |

## EEPROM Layout Factory Check

Do the EEPROM Layout factory check after Factory Reset, check the date in the eeprom , make sure the date equal to the except the pointer calibration date.

OBJECT:

                        Check the initial value.

METHOD:

                        Send simulation message of diagnostic service.

PROCEDURE:

                        All loads/values applied as per specification in the following table.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Point  (Apply) | Apply | Description | Test Point  (Measure) | Value | Measure |
| Block Pin13 | 13.5VDC | Battery | F108 | 13.5V |  |
| Block Pin 28 | GND | GND | F110 | 0V |  |
| Block Pin 14 | 13.5VDC | Ignition | F109 | 13.5V |  |
| Block Pin 2 | CANH | PCANH | F100 |  |  |
| Block Pin 3 | CANL | PCANL | F101 |  |  |

• TT1 – Read out the data

### Refer to “Read Memory by Address”

• TT2 – check the data

Refer to < EASTONE\_EL\_ET\_CAR\_ID5027-88Y\_XXXX\_0xXXXX\_XX\_EEPROMDataVXX.Hex >