|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Title:** | **DFM\_SX5\_ATC\_FC1616\_66 Product Test Specification** | | | | |
| **Project:** | **DFM\_SX5\_Low\_ATC** | | **Phase:** | D | |
| **HW Version:** | 06 | | **SW Version:** | D1.01 | |
| **EE:** | Liu Guoqian | **SW:** | Li Zhirui | **DEQ:** | Lin Kaiming |
| **Document Version:** | A03 | **Document Date:** | 2016/10/26 | **Document Status:** | Release |

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:**

DFM\_SX5\_FC1616/66\_PTS\_A03\_Release.doc

.

|  |  |  |  |
| --- | --- | --- | --- |
| History | | | |
| **Version** | **Date** | **Name, Department** | **Changes** |
| A01 |  |  | The First Version |
| A02 |  |  | SW(基于SX5高配PTS编写):  4.10.2 Digital Part Control   1. 删除命令FF下的FF参数。 2. 修改AC digital OFF电压为>10V(电池电压14V时) 3. 修改Rear Defrost Digital Port off电压为>10V(电池电压14V时) 4. 删除座椅加热IO口控制   4.10.3 Read the digital port  删除座椅加热指示灯信号读取  4.21 Write Product Information  Model名改为FC1616/66  4.22 Read Product Information  Model名改为FC1616/66  4.10.1 Analogue Part  5v feedback AD值改为0xA3.  4.12.1 Read the Key when KEY Pressed   1. 无用KEY已写NA. 2. 删除座椅加热KEY.   4.13 Actuator Output Test  更改逆时针方向电压范围  4.14 Blower Test  鼓风机最大档电压改为电池电压。  4.16 LED Test  删除座椅加热指示灯  4.19.1 Write Memory by Address  例子中的发送第三帧数据的Byte1的值改为0x03  4.21 Write Product Information   1. Model ID的读写，应该明确说明Data0的值为0x02 2. 增加Write Hardware Number   4.22 Read Product Information  增加Read Hardware Number |
| A03 |  | Li Zhirui  Lin Kaiming | SW 2016-09-06:  4.10.3 Read the digital port  备注说明无用的位所设置的特定值。  DEQ(2016-10-26)  表头信息：更新软硬件版本。 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Review List | | | |
| **Role** | **Name** | **Company/Department** | **Location** |
| Manufacturing Planning | Yang Kunxing | PED | HZH |
| Quality Planning (QP) | Liu Shuhan | QM-PQM | HZH |
| EE (senior engineer) | Liu Guoqian | BIC R&D EE | HZH |
| SW (senior engineer) | Li Zhirui | BIC R&D SW | HZH |
| Q&PTV (senior engineer) | Lin Kaiming | BIC DEQ | HZH |

|  |  |  |  |
| --- | --- | --- | --- |
| Distribution List | | | |
| **Role** | **Name** | **Company/Department** | **Location** |
| Manufacturing Planning | Yang Kunxing | PED | HZH |
| Quality Planning (QP) | Liu Shuhan | QM-PQM | HZH |

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

Contents List Page

目录

[1 Introduction 6](#_Toc465321925)

[1.1 Objectives 6](#_Toc465321926)

[1.2 Definitions, Acronyms and Abbreviations 6](#_Toc465321927)

[2 General Concepts 7](#_Toc465321928)

[2.1 Description 7](#_Toc465321929)

[2.2 Main Connector Pin Out 7](#_Toc465321930)

[3 Manufacturing 10](#_Toc465321931)

[3.1 Production Flow and Control Plan 10](#_Toc465321932)

[3.2 Handling of Components, PCBs and Finished DUTs 10](#_Toc465321933)

[3.2.1 ESD Care 10](#_Toc465321934)

[3.2.2 Bending of Components 10](#_Toc465321935)

[3.3 Visual Inspection 10](#_Toc465321936)

[3.4 Documentation/Traceability 10](#_Toc465321937)

[4 PCB Functional Test 11](#_Toc465321938)

[4.1 Purpose 11](#_Toc465321939)

[4.1.1 Operating Values 11](#_Toc465321940)

[4.1.2 Failures in PCB Functional Test 11](#_Toc465321941)

[4.2 General Measurement Conditions 11](#_Toc465321942)

[4.2.1 Test Jig Connection 11](#_Toc465321943)

[4.3 Supply Voltage Test 16](#_Toc465321944)

[4.4 Operation Current Test 16](#_Toc465321945)

[4.5 HW Test 16](#_Toc465321946)

[4.6 SW Test 16](#_Toc465321947)

[4.6.1 Diagnostic Command 16](#_Toc465321948)

[4.6.2 Bus 16](#_Toc465321949)

[4.6.3 Communication mechanism 16](#_Toc465321950)

[4.7 Enter PTS Mode 17](#_Toc465321951)

[4.8 Exit PTS Mode 17](#_Toc465321952)

[4.9 Quiescent Current Test 17](#_Toc465321953)

[4.10 Input & Output Test 18](#_Toc465321954)

[4.10.1 Analogue Part 18](#_Toc465321955)

[4.10.2 Digital Part Control 20](#_Toc465321956)

[4.10.3 Read the digital port 21](#_Toc465321957)

[4.10.4 Control the PWM/Frequency output for periphery components 22](#_Toc465321958)

[4.10.5 Read the PWM/ Frequency input 22](#_Toc465321959)

[4.11 NAT System Test 22](#_Toc465321960)

[4.12 Key and Knob Test 22](#_Toc465321961)

[4.12.1 Read the Key when KEY Pressed 22](#_Toc465321962)

[4.12.2 Knob Test 23](#_Toc465321963)

[4.13 Actuator Output Test 24](#_Toc465321964)

[4.14 Blower Test 25](#_Toc465321965)

[4.15 **UART Test** 25](#_Toc465321966)

[4.16 **LED Test** 25](#_Toc465321967)

[4.17 **LCD Test** 27](#_Toc465321968)

[4.18 **Control PWM of Dimming** 27](#_Toc465321969)

[4.19 EEPROM Access 28](#_Toc465321970)

[4.19.1 Write Memory by Address 28](#_Toc465321971)

[4.19.2 Read Memory by Address 31](#_Toc465321972)

[4.20 Read Software Version 33](#_Toc465321973)

[4.21 Write Product Information 35](#_Toc465321974)

[4.22 Read Product Information 35](#_Toc465321975)

[4.23 Factory Setting 36](#_Toc465321976)

# Introduction

## Objectives

This Product Test Specification (PTS) defines the requirements for the PCB Functional Test and the End of Line Test of the **Project Name** Heating, Ventilation and Air Conditioning (HVAC) Controller unit.

## Definitions, Acronyms and Abbreviations

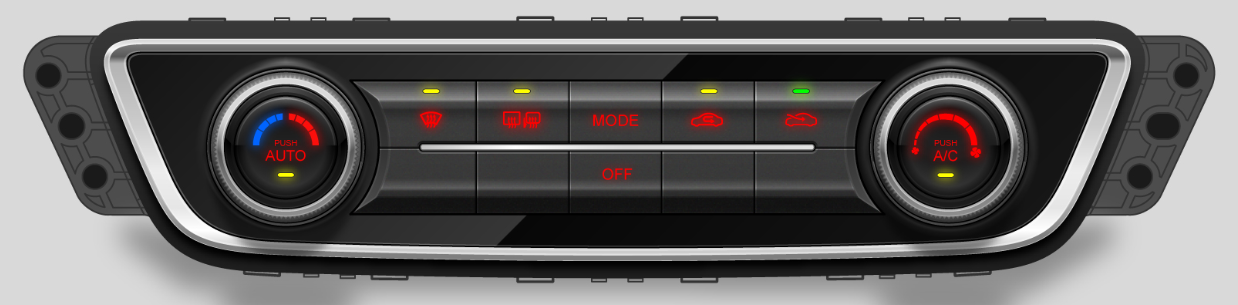
| **Abbreviation** | **Meaning** |
| --- | --- |
| PTS | Product Test Specification |
| BOM | Bill of material |
| DUT | Device under Test |
| EOL | End of Line |
| FA | Final Assembly |
| HVAC | Heating, Ventilation and Air Conditioning |
| HVACC | HVAC Controller |
| OAT | Outside Ambient Temperature |
| ICT | In Car Temperature |
| EVA | Evaporator Temperature |
| AD | Air Distribution |
|  |  |
|  |  |

# General Concepts

## Description

**Two PCBs are** used for HVACC.

**FC1616/66 without Seat heating:**



(10)

(6)

(7)

(8)

(9)

(5)

(4)

(3)

(2)

(1)

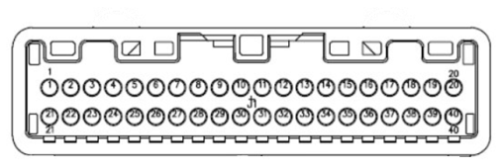
**Control Buttons and Display:**

| **Buttons** | **Function** | **Backlight** | **Telltale** | **Display Area** |
| --- | --- | --- | --- | --- |
| (1) | Temp Knob 温度变化调节编码器 | Red & Blue | NA | NA |
| (2) | AUTO Button 开启自动控制模式按键 | Red | Yellow | NA |
| (3) | FDEF Button 开启前除霜按键 | Red | Yellow | NA |
| (4) | RDEF Button 开启后窗后视镜除霜按键 | Red | Yellow | NA |
| (5) | MODE Button 开启模式选择按键 | Red | NA | NA |
| (6) | OFF Button 关闭空调控制按键 | Red | NA | NA |
| (7) | REC Button 开启内循环控制按键 | Red | Yellow | NA |
| (8) | FRESH Button 开启外循环控制按键 | Red | Green | NA |
| (9) | A/C Button 开启压缩机按键 | Red | Yellow | NA |
| (10) | Blower Knob 风量变化调节编码器 | Red | NA | NA |

## Main Connector Pin Out

**(1) Main connectors:**

Main connectors: One connector



| Pin | Symbol | I/O | Description | | Remarks |
| --- | --- | --- | --- | --- | --- |
| A1 | KL30 | 蓄电池电源正 | | Input |  |
| A2 | KL15 | 点火电源正 | | Input |  |
| A3 | ILL+ | 背光电源+（输入） | | Input |  |
| A4 | - | - | | - | NC |
| A5 | MOTOR\_TEMP\_R- | 右温度电机全冷方向CW：A5为+，A6为- | | output | Only used in SX7 High and SX7 Low; |
| A6 | MOTOR\_TEMP\_R+ | 右温度电机全热方向CCW：A5为-，A6为+ | | output | Only used in SX7 High and SX7 Low; |
| A7 | MOTOR\_R/F- | 内外循环电机-（A7为+,A8为-时外循环） | | output |  |
| A8 | MOTOR\_R/F+ | 内外循环电机+（A7为-,A8为+时内循环） | | output |  |
| A9 | MOTOR\_MODE+ | 模式电机吹面方向CW：A9为-，A10为+ | | output |  |
| A10 | MOTOR\_MODE- | 模式电机除霜方向CCW：A9为+，A10为- | | output |  |
| A11 | MOTOR\_TEMP\_L- | 左温度电机全冷方向CW：A11为+，A12为- | | output |  |
| A12 | MOTOR\_TEMP\_L+ | 左温度电机全热方向CCW：A11为-，A12为+ | | output |  |
| A13 | A/C\_REQUEST | AC输出，低有效 | | output |  |
| A14 | REAR\_DEFROST\_OUT | 后除霜请求，低有效 | | output |  |
| A15 | REAR\_AC\_OUT | 后空调控制，低有效 | | output | Only used in F600A and SX6A; |
| A16 | SHEAT\_L\_OUT | 左座椅加热控制，高有效 | | output | Only used in SX5 High & SX7 High & S60 High |
| A17 | SHEAT\_R\_OUT | 右座椅加热控制，高有效 | | output | Only used in SX5 High & SX7 High & S60 High |
| A18 | - | - | | - | NC |
| A19 | CAN\_H | CAN通讯H | | Input/Output |  |
| A20 | CAN\_L | CAN通讯L | | Input/Output |  |
| A21 | GND | 电源地 | | Input |  |
| A22 | SGND | 传感器地 | | Input |  |
| A23 | ILL- | 背光电源-（接地） | | Input |  |
| A24 | MOTOR\_MODE\_FB | 模式电机位置反馈 | | Input |  |
| A25 | MOTOR\_TEMP\_L\_FB | 左温度电机位置反馈 | | Input |  |
| A26 | MOTOR\_TEMP\_R\_FB | 右温度电机位置反馈 | | Input | Only used in SX7 High and SX7 Low; |
| A27 | TEMP\_EVAP | 蒸发器温度传感器 | | Input |  |
| A28 | TEMP\_INCAR | 车内传感器 | | Input |  |
| A29 | TEMP\_AMB | 车外传感器 | | Input |  |
| A30 | SUN\_SENS\_L | 左阳光传感器 | | Input |  |
| A31 | SUN\_SENS\_R | 右阳光传感器 | | Input | Only used in SX7 High and SX7 Low; |
| A32 | REF\_5V | 5V参考电压 | | output |  |
| A33 | BLOWER\_CTRL | 鼓风机控制信号 | | output |  |
| A34 | BLOWER\_FB- | 鼓风机负反馈信号 | | Input |  |
| A35 | BLOWER\_FB+ | 鼓风机正反馈信号 | | Input |  |
| A36 | SHEAT\_L\_LED1 | 左座椅加热指示灯1，低有效 | | Input | Only used in SX5 High & SX7 High & S60 High |
| A37 | SHEAT\_L\_LED2 | 左座椅加热指示灯2，低有效 | | Input | Only used in SX5 High & SX7 High & S60 High |
| A38 | SHEAT\_R\_LED1 | 右座椅加热指示灯1，低有效 | | Input | Only used in SX5 High & SX7 High & S60 High |
| A39 | SHEAT\_R\_LED2 | 右座椅加热指示灯2，低有效 | | Input | Only used in SX5 High & SX7 High & S60 High |
| A40 | REAR\_DEFROST\_FB | 后除霜反馈信号，高有效 | | Input |  |

# 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 a HVACC unit. 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** | **Test Point** | **Values/Connection** | **Remark** |
| --- | --- | --- | --- | --- |
| KL30 | A1 | I50 | 14V ±0.2V | Current limitation to 1.5 A |
| KL15 | A2 | I53 | 14V ±0.2V | Current limitation to 0.5A |
| ILL+ | A3 | I158 | 14V ±0.2V | Current limitation to 0.5A |
| GND | A21 | I116 | GND(0 ±0.2V) | Current limitation to 1.5 A |
| ILL- | A23 | I159 | GND(0 ±0.2V) | Current limitation to 0.5A |

### 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

**FC1616/66 without Seat heating:**

| **Signal Name** | **Signal Abbreviation** | **Connection** | **Test Pin** |
| --- | --- | --- | --- |
| **Battery input** | **KL30** | **Positive Voltage Source** | **A1** |
| **Ignition input** | **KL15** | **Positive Voltage Source** | **A2** |
| **Illumination Input** | **ILL+** | **Positive Voltage Source** | **A3** |
| **Battery GND** | **GND** | **Connect to GND** | **A21** |
| **Signal GND** | **SGND** | **Connect to Signal GND** | **A22** |
| **Illumination output** | **ILL-** | **Connect to GND** | **A23** |
| **Right Mixed air actuator** | **MOTOR\_TEMP\_R-** | **Connect to 50Ω and 100uH across A5, A6** | **A5** |
|  | **MOTOR\_TEMP\_R+** |  | **A6** |
| **Recirculation actuator** | **MOTOR\_R/F-** | **Connect to 50Ω and 100uH across A7, A8** | **A7** |
|  | **MOTOR\_R/F+** |  | **A8** |
| **Mode actuator** | **MOTOR\_MODE+** | **Connect to 50Ω and 100uH across A9, A10** | **A9** |
|  | **MOTOR\_MODE-** |  | **A10** |
| **Left Mixed air actuator** | **MOTOR\_TEMP\_L-** | **Connect to 50Ω and 100uH across A11, A12** | **A11** |
|  | **MOTOR\_TEMP\_L+** |  | **A12** |
| **A/C Request Signal** | **A/C\_REQUEST** | **Connect to R5(1kΩ)** | **A13** |
| **Rear Defrost Request Signal** | **REAR\_DEFROST\_OUT** | **Connect to R6(1kΩ)** | **A14** |
| **Rear A/C Request Signal** | **REAR\_AC\_OUT** | **Connect to R7(1kΩ)** | **A15** |
| **Left Seat heat Request Signal** | **SHEAT\_L\_OUT** | **Connect to R8(1kΩ)** | **A16** |
| **Right Seat heat Request Signal** | **SHEAT\_R\_OUT** | **Connect to R9(1kΩ)** | **A17** |
| **CAN H Signal** | **CAN\_H** |  | **A19** |
| **CAN L Signal** | **CAN\_L** |  | **A20** |
| **EVAP temperature signal** | **TEMP\_EVAP** | **Resistor R10(3.3KΩ) to SGND** | **A27** |
| **Incar temperature signal** | **TEMP\_INCAR** | **Resistor R11(2.2KΩ) to SGND** | **A28** |
| **Outside air temperature signal** | **TEMP\_AMB** | **Resistor R12(3.3KΩ)to SGND** | **A29** |
| **Left SUN sensor signal** | **SUN\_SENS\_L** | **Resistor R13(15KΩ)to SGND** | **A30** |
| **Right SUN sensor signal** | **SUN\_SENS\_R** | **Resistor R14(10KΩ)to SGND** | **A31** |
| **+5V Output to motor** | **REF\_5V** | **Supply power to motor FB** | **A32** |
| **MODE motor actuator feedback** | **MOTOR\_MODE\_FB** | **Connect to R15(1.5kΩ) and R16(3.3kΩ)** | **A24** |
| **Left TEMP actuator feedback** | **MOTOR\_TEMP\_L\_FB** | **Connect to R17(2.7kΩ) and R18(3.3kΩ)** | **A25** |
| **Right TEMP motor actuator feedback** | **MOTOR\_TEMP\_R\_FB** | **Connect to R19(1.5kΩ) and R20(3.3kΩ)** | **A26** |
| **Blower feedback signal+** | **BLOWER\_FB+** | **Connect to Battery by K5** | **A35** |
| **Blower feedback signal-** | **BLOWER\_FB-** | **Connect to BLOWER\_FEEDBACK**  **of blower control module** | **A34** |
| **Blower control signal** | **BLOWER\_CTRL** | **Connect to BLOWER\_CTRL of blower control module** | **A33** |
| **Left Seat heat indicator light1 Signal** | **SHEAT\_L\_LED1** | **Connect to Battery by R22(10K), and connect to GND by K6** | **A36** |
| **Left Seat heat indicator light2 Signal** | **SHEAT\_L\_LED2** | **Connect to Battery by R23(10K), and connect to GND by K7** | **A37** |
| **Right Seat heat indicator light1 Signal** | **SHEAT\_R\_LED1** | **Connect to Battery by R24(10K), and connect to GND by K8** | **A38** |
| **Right Seat heat indicator light2 Signal** | **SHEAT\_R\_LED2** | **Connect to Battery by R25(10K), and connect to GND by K9** | **A39** |
| **Rear defrost feedback** | **REAR\_DEFROST\_FB** | **Connect to the switch K10** | **A40** |

Refer to：DFM\_SX5\_ATC(FC1606\_66)\_PTS\_TEST\_A1\_AA.pdf





**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** | **≤1.5A** | **A1** | **Close switch K1, Read A2 ≤ 1.5A** | **Refer to 4.7** |
| **KL15** | **≤0.5A** | **A2** | **Close switch K2, Read A3 ≤ 0.5A** | **Refer to 4.7** |
| **ILL+** | **≤0.5A** | **A3** | **Close switch K3, Read A1 ≤ 0.5A** | **Refer to 4.11** |
| **MOTOR\_TEMP\_R-** | **/** | **A5** | **Software sends messages to control the motor. Read V1** | **Refer to 4.7** |
| **MOTOR\_TEMP\_R+** |  | **A6** |  |  |
| **MOTOR\_R/F-** | **/** | **A7** | **Software sends messages to control the motor. Read V2** | **Refer to 4.7** |
| **MOTOR\_R/F+** |  | **A8** |  |  |
| **MOTOR\_MODE+** | **/** | **A9** | **Software sends messages to control the motor. Read V3** | **Refer to 4.7** |
| **MOTOR\_MODE-** |  | **A10** |  |  |
| **MOTOR\_TEMP\_L-** | **/** | **A11** | **Software sends messages to control the motor. Read V4** | **Refer to 4.7** |
| **MOTOR\_TEMP\_L+** |  | **A12** |  |  |
| **A/C\_REQUEST** | **OFF：>10V**  **ON： 0±0.2V** | **A13** | **Close switch K4,Software sends messages to control the signa ON/OFF. Read V5** | **Refer to 4.7** |
| **REAR\_DEFROST\_OUT** | **OFF：>10V**  **ON： 0±0.2V** | **A14** | **Close switch K5,Software sends messages to control the signa ON/OFF. Read V6** | **Refer to 4.7** |
| **REAR\_AC\_OUT** | **OFF：>10V**  **ON： 0±0.2V** | **A15** | **Close switch K6,Software sends messages to control the signa ON/OFF. Read V7** | **Refer to 4.7** |
| **SHEAT\_L\_OUT** | **OFF：0±0.2V**  **ON： >10V** | **A16** | **Software sends messages to control the signa ON/OFF. Read V8** | **Refer to 4.7** |
| **SHEAT\_R\_OUT** | **OFF：0±0.2V**  **ON： >10V** | **A17** | **Software sends messages to control the signa ON/OFF. Read V9** | **Refer to 4.7** |
| **CAN\_H** | **/** | **A19** |  |  |
| **CAN\_L** | **/** | **A20** |  |  |
| **TEMP\_EVAP** | **1.85±0.1V** | **A27** | **Read V10** | **Refer to 4.7.1** |
| **TEMP\_INCAR** | **2.5±0.1V** | **A28** | **Read V11** | **Refer to 4.7.1** |
| **TEMP\_AMB** | **2.06±0.1V** | **A29** | **Read V12** | **Refer to 4.7.1** |
| **SUN\_SENS\_L** | **2.5±0.1V** | **A30** | **Read V13** | **Refer to 4.7.1** |
| **SUN\_SENS\_R** | **2.0±0.1V** | **A31** | **Read V14** | **Refer to 4.7.1** |
| **REF\_5V** | **5±0.2V** | **A32** | **Read V15** | **Refer to 4.7** |
| **MOTOR\_MODE\_FB** | **3.44±0.1V** | **A27** | **Read V16** | **Refer to 4.7.1** |
| **MOTOR\_TEMP\_L\_FB** | **2.75±0.1V** | **A28** | **Read V17** | **Refer to 4.7.1** |
| **MOTOR\_TEMP\_R\_FB** | **3.44±0.1V** | **A29** | **Read V18** | **Refer to 4.7.1** |
| **BLOWER\_FB+** | **/** | **A35** | **Close to K7, use the battery voltage for blower control.** | **Refer to 4.7** |
| **BLOWER\_FB-** | **/** | **A34** | **Read V19** | **Refer to 4.7** |
| **BLOWER\_CTRL** | **/** | **A33** | **Read V20** | **Refer to 4.7** |
| **SHEAT\_L\_LED1** | **/** | **A36** | **Close to K8, Software sends messages to detect signal voltage value. Low active.** | **Refer to 4.7** |
| **SHEAT\_L\_LED2** | **/** | **A37** | **Close to K9, Software sends messages to detect signal voltage value. Low active.** | **Refer to 4.7** |
| **SHEAT\_R\_LED1** | **/** | **A38** | **Close to K10, Software sends messages to detect signal voltage value. Low active.** | **Refer to 4.7** |
| **SHEAT\_R\_LED2** | **/** | **A39** | **Close to K11, Software sends messages to detect signal voltage value. Low active.** | **Refer to 4.7** |
| **REAR\_DEFROST\_FB** | **/** | **A40** | **Close to K12, Software sends messages to detect signal voltage value. High active.** | **Refer to 4.7.1** |

## Supply Voltage Test

**(1) Battery Voltage**

(Describe how to operate the DUT.)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Signal Name** | **Signal Abbreviation** | **Operation Voltage** | **Test Pin** | **Test Point** |
| Battery | KL30 | 14V ±0.2V | A1->A21 | I50->I116 |

**(2) Ignition Voltage**

(Describe how to operate the DUT.)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Signal Name** | **Signal Abbreviation** | **Operation Voltage** | **Test Pin** | **Test Point** |
| Ignition | KL15 | 14V ±0.2V | A2->A21 | I53->I116 |

**(3) Illumination Voltage**

(Describe how to operate the DUT.)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Signal Name** | **Signal Abbreviation** | **Operation Voltage** | **Test Pin** | **Test Point** |
| Illumination | ILL+ | 14V ±0.2V | A3->A23 | I158->I59 |

## Operation Current Test

**(1) Battery Current**

(Describe how to operate the DUT.)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Signal Name** | **Signal Abbreviation** | **Operation Current** | **Test Pin** | **Test Point** |
| Battery | KL30 | ≤0.5A | A1 | I50 |

**(2) Ignition Current**

(Describe how to operate the DUT.)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Signal Name** | **Signal Abbreviation** | **Operation Current** | **Test Pin** | **Test Point** |
| Ignition | KL15 | ≤0.5A | A2 | I53 |

**(3) Illumination Current**

(Describe how to operate the DUT.)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Signal Name** | **Signal Abbreviation** | **Operation Current** | **Test Pin** | **Test Point** |
| Illumination | ILL+ | ≤0.5A | A3 | I158 |

## HW Test

NA

## SW Test

### Diagnostic Command

### Bus

The bus used in the HVACC is CAN bus.

#### Bus Interface

This typical ratio for the data transfer of the CAN-Bus is 500k bit/s.

### Communication mechanism

Use the Master-Slave mechanism,

|  |
| --- |
| The tester need to send the request firstly, and then the ECU will response the Data within **20** ms. |
|
| Tester Request Format:   |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | | **ID** | **Byte0** | **Byte1** | **Byte2** | **Byte3** | **Byte4** | **Byte5** | **Byte6** | **Byte7** | | **0x7C0** | **0xXX** | **0xXX** | **0xXX** | **0xXX** | **0xXX** | **0xXX** | **0xXX** | **0xXX** | |
|

ECU Response Format:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Byte0** | **Byte1** | **Byte2** | **Byte3** | **Byte4** | **Byte5** | **Byte6** | **Byte7** |
| **0x7D0** | **0xXX** | **0xXX** | **0xXX** | **0xXX** | **0xXX** | **0xXX** | **0xXX** | **0xXX** |

* 1. Enter PTS Mode

Wait 2 seconds after KL30 &KL15 is ON (Before entering PTS mode).

**Step1：Send Password to ECU**

**Send the below message to ECU for PTS test enable: Password is: DESAYSV**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Byte0** | **Byte1** | **Byte2** | **Byte3** | **Byte4** | **Byte5** | **Byte6** | **Byte7** |
| **0x7C0** | **0x27** | **0x44** | **0x45** | **0x53** | **0x41** | **0x59** | **0x53** | **0x56** |

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

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Byte0** | **Byte1** | **Byte2** | **Byte3** | **Byte4** | **Byte5** | **Byte6** | **Byte7** |
| **0x7D0** | **0x67** | **0x44** | **0x45** | **0x53** | **0x41** | **0x59** | **0x53** | **0x56** |

Delay more than 50ms,

**Step2: Send commend for changing into PTS mode**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Byte0** | **Byte1** | **Byte2** | **Byte3** | **Byte4** | **Byte5** | **Byte6** | **Byte7** |
| **0x7D0** | **0x10** | **0x22** | **0x00** | **0x00** | **0x00** | **0x00** | **0x00** | **0x00** |

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

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Byte0** | **Byte1** | **Byte2** | **Byte3** | **Byte4** | **Byte5** | **Byte6** | **Byte7** |
| **0x7D0** | **0x50** | **0x22** | **0x00** | **0x00** | **0x00** | **0x00** | **0x00** | **0x00** |

## Exit PTS Mode

**Send commend for exiting the PTS mode**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Byte0** | **Byte1** | **Byte2** | **Byte3** | **Byte4** | **Byte5** | **Byte6** | **Byte7** |
| **0x7C0** | **0x10** | **0x20** | **0x00** | **0x00** | **0x00** | **0x00** | **0x00** | **0x00** |

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

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Byte0** | **Byte1** | **Byte2** | **Byte3** | **Byte4** | **Byte5** | **Byte6** | **Byte7** |
| **0x7D0** | **0x50** | **0x20** | **0x00** | **0x00** | **0x00** | **0x00** | **0x00** | **0x00** |

## Quiescent Current Test

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

**FC1616/66 without Seat heating:**

**Firstly, Client sends the command of sleep quickly, and then ECU will respond the request. Secondly, turn KL15 off.**

**Thirdly, disconnect the LED feedback of seat heating.**

**2 seconds later, test the current.**

(Describe how to operate the DUT.)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Signal Name** | **Signal Abbreviation** | **Quiescent Current** | **Test Pin** | **Test Point** |
| Battery supply | BATTERY | <1mA | A1 | I50 |

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

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Byte0** | **Byte1** | **Byte2** | **Byte3** | **Byte4** | **Byte5** | **Byte6** | **Byte7** |
|  | **Service**  **Type** | **Control**  **ID** | **Control**  **Sub Type** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** |
| **0x7C0** | **0x30** | **0x27** | **0x00** | **0x00** | **0x00** | **0x00** | **0x00** | **0x00** |

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

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Byte0** | **Byte1** | **Byte2** | **Byte3** | **Byte4** | **Byte5** | **Byte6** | **Byte7** |
|  | **Service**  **Type** | **Control**  **ID** | **Control**  **Sub Type** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** |
| **0x7D0** | **0x70** | **0x27** | **0x00** | **0x00** | **0x00** | **0x00** | **0x00** | **0x00** |

测完静态电流后，KL15上电后需重新进入PTS模式。

## Input & Output Test

### Analogue Part

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

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Byte0** | **Byte1** | **Byte2** | **Byte3** | **Byte4** | **Byte5** | **Byte6** | **Byte7** |
|  | **Service**  **Type** | **ReadOut**  **ID** | **Control**  **Sub Type** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** |
| **0x7C0** | **0x22** | **0x05** | **0xXX** | **0x00** | **0x00** | **0x00** | **0x00** | **0x00** |

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

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Byte0** | **Byte1** | **Byte2** | **Byte3** | **Byte4** | **Byte5** | **Byte6** | **Byte7** |
|  | **Service**  **Type** | **ReadOut**  **ID** | **Control**  **Sub Type** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** |
| **0x7D0** | **0x62** | **0x05** | **0xXX** | **0xXX** | **0xXX** | **0xXX** | **0xXX** | **0xXX** |

The data field is defined as following table:

**FC1616/66 without Seat heating:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Control**  **Sub Type** | **Data field** | **Description** | **AD Value** | **Voltage** | **Test Pin** | **Test Point** |
| 0x01 | Data0 | Battery | 0xAA±0x0A | 14V±0.196V  (Test Pin，Formula 2) | A1 | I50 |
| Data1 | ( Left ) Mixed Motor Feedback | 0x8C±0x0A | 2.75V±0.196V  (Formula 1) | A25 | I43 |
| Data2 | Right Mixed Motor Feedback |  |  | N.A. | NA |
| Data3 | Air distribution Motor Feedback | 0xAF±0x0A | 3.43V±0.196V  (Formula 1) | A24 | I42 |
| Data4 | Recirculation motor feedback |  |  | N.A. | NA |
| 0x02 | Data0 | Evaporator Sensor | 0x5E±0x0A | 1.85V±0.196V  (Formula 1) | A27 | I9 |
| Data1 | ( Left ) Solar Sensor | 0x7F±0x0A | 2.5V±0.196V  (Formula 1) | A30 | I12 |
| Data2 | Right Solar Sensor |  |  | N.A. | NA |
| Data3 | Ambient Temperature Sensor | 0x69±0x0A | 2.06V±0.196V  (Formula 1) | A29 | I11 |
| Data4 | In Car sensor | 0x7F±0x0A | 2.5V±0.196V  (Formula 1) | A28 | I10 |
| 0x03 | Data0 | Heat core sensor |  |  | N.A. | NA |
| Data1 | AQS Sensor |  |  | N.A. | NA |
| Data2 | Pressure Sensor |  |  | N.A. | NA |
| Data3 | Reserved for IG1 | 0xAA±0x0A | 14V±0.196V  (Test Pin，Formula 2) | A2 | I53 |
| Data4 | Reserved for IG2 |  |  | N.A. | NA |
| 0x04 | Data0 | Blower Feedback |  |  | N.A. | NA |
| Data1 | 5V feedback | 0xA3±0x0A | 5V±0.196V  (Formula 1) | N.A. | I250 |
| Data2 | Rear Defrost Feedback | N.A | N.A | N.A. | N.A. |
| Data3 | Left Seat Heating Feedback | N.A | N.A | N.A. | N.A. |
| Data4 | Right Seat Heating Feedback | N.A | N.A | N.A. | N.A. |
| 0x05 | Data0 | Dimming detect | N.A | N.A | N.A. | N.A. |
| Data1 | Left Face Blowout Temperature sensor | N.A | N.A | N.A. | N.A. |
| Data2 | Right Face Blowout Temperature sensor | N.A | N.A | N.A. | N.A. |
| Data3 | Left Foot Blowout Temperature sensor | N.A | N.A | N.A. | N.A. |
| Data4 | Right Foot Blowout Temperature sensor | N.A | N.A | N.A. | N.A. |
| 0x06 | Data0 | NATS Surface Temperature Sensor | N.A | N.A | N.A. | N.A. |
| Data1 | NATS PCB Temperature Sensor | N.A | N.A | N.A. | N.A. |
| Data2 | NATS IR Sensor | N.A | N.A | N.A. | N.A. |
| Data3 | Reserved |  |  |  |  |
| Data4 | Reserved |  |  |  |  |
| 0x07 | Data0 | PCB NTC | 15~40 degree celsius（Refer to table below） | 15~40 degree celsius（Refer to table below） | N.A. | I45 |
| Data1 | User defined |  |  |  |  |
| Data2 | User defined |  |  |  |  |
| Data3 | User defined |  |  |  |  |
| Data4 | User defined |  |  |  |  |
| …… | …… |  |  |  |  |  |

AD value and voltage relative to Temperature of PCB NTC:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **T(℃)** | **0** | **5** | **10** | **15** | **20** | **25** |
| **AD(hex)** | 0xBA±0x0A | 0xAF±0x0A | 0xA3±0x0A | 0x97±0x0A | 0x8B±0x0A | 0x7F±0x0A |
| **Vol(V)** | 3.66±0.196 | 3.44±0.196 | 3.21±0.196 | 2.98±0.196 | 2.74±0.196 | 2.50±0.196 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **T(℃)** | **30** | **35** | **40** | **45** | **50** | **55** | **60** |
| **AD(hex)** | 73±0x0A | 0x68±0x0A | 0x5D±0x0A | 0x53±0x0A | 0x4B±0x0A | 0x42±0x0A | 0x3A±0x0A |
| **Vol(V)** | 2.27±0.196 | 2.04±0.196 | 1.84±0.196 | 1.64±0.196 | 1.48±0.196 | 1.30±0.196 | 1.15±0.196 |

Note: All the AD conversion is done in 0 to 5V range. The conversion of the data value to the analogue voltage is done according to the following formula.

Formula 1: Vn = [ Data x 5V ] / 255

**Example:** Data = 0x5C;

Analogue voltage, Vn = (92 x 5)/255 = 1.80V.

Formula 2: Vn = [(Data x 4) / 51] +0.6

**Example:** Data = 0x5C;

Analogue voltage, Vn = [(Data x 4) / 51] +0.6 = 7.82V.

### Digital Part Control

**The following command can be used to control the digital port:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Byte0** | **Byte1** | **Byte2** | **Byte3** | **Byte4** | **Byte5** | **Byte6** | **Byte7** |
|  | **Service**  **Type** | **Control**  **ID** | **Control**  **Sub Type** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** |
| **0x7C0** | **0x30** | **0x21** | **0xXX** | **0xXX** | **0x00** | **0x00** | **0x00** | **0x00** |

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

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Byte0** | **Byte1** | **Byte2** | **Byte3** | **Byte4** | **Byte5** | **Byte6** | **Byte7** |
|  | **Service**  **Type** | **Control**  **ID** | **Control**  **Sub Type** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** |
| **0x7D0** | **0x70** | **0x21** | **0xXX** | **0xXX** | **0x00** | **0x00** | **0x00** | **0x00** |

The Control Sub Type and the Data0 which used to control each of the various signal outputs are defined as follow table (test sequence should meet “signal on – signal off- signal on again”):

| **Control**  **Sub Type** | **Data0** | **Description** | **Voltage** | **Test Pin** | **Test Point** |
| --- | --- | --- | --- | --- | --- |
| **0x01** | 0x00 | AC Digital Off | >10V | A13 | I236 |
| 0x01 | AC Digital On | 0V±0.2V | A13 | I236 |
| **0x02** | 0x00 | 5V Digital Off | 0V±0.2V | A32 | I243 |
| 0x01 | 5V Digital On | 5V±0.2V | A32 | I243 |
| **0x03** | 0x00 | Sensor supply Digital Port off | 0V±0.2V | N.A. | I7 |
| 0x01 | Sensor supply Digital on | 5V±0.2V | N.A. | I7 |
| **0x04** | 0x00 | Rear Defrost Digital Port off | >10V | A14 | I237 |
| 0x01 | Rear Defrost Digital Port on | 0V±0.2V | A14 | I237 |
| **0x05** | 0x00 | Blower High Speed Digital Port off | N.A. | N.A. | N.A. |
| 0x01 | Blower High Speed Digital Port on | N.A. | N.A. | N.A. |
| **0xFF** | 0x00 | Control all the Digital Port off |  |  |  |
| 0x01 | Control all the Digital Port on |  |  |  |

### Read the digital port

**The following command can be used to Read the PIN Input:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Byte0** | **Byte1** | **Byte2** | **Byte3** | **Byte4** | **Byte5** | **Byte6** | **Byte7** |
|  | **Service**  **Type** | **Read Out**  **ID** | **Read Out Sub Type** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** |
| **0x7C0** | **0x22** | **0x03** | **0x02** | **0x00** | **0x00** | **0x00** | **0x00** | **0x00** |

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

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Byte0** | **Byte1** | **Byte2** | **Byte3** | **Byte4** | **Byte5** | **Byte6** | **Byte7** |
|  | **Service**  **Type** | **ReadOut**  **ID** | **Control**  **Sub Type** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** |
| **0x7D0** | **0x62** | **0x03** | **0x02** | **0xXX** | **0x00** | **0x00** | **0x00** | **0x00** |

**// 下面的内容需要根据实际项目更新**

| **Data0** | **Description** | **Control** | | **Test Voltage** | **Test Pin** | **Test Point** |
| --- | --- | --- | --- | --- | --- | --- |
| **Value** | **Description** |
| Bit0 | Rear Defog Feedback | 0 | Low Level | 0±0.2V | A40 | I55 |
| 1 | High Level | >10V | A40 | I55 |
| Bit1 | i-Stop feedback | 0 | Low Level | N.A. | N.A. | N.A. |
| 0 | High Level | N.A. | N.A. | N.A. |
| Bit2 | Blower | 0 | Blower OFF | N.A. | N.A. | N.A. |
| 0 | Blower ON | N.A. | N.A. | N.A. |
| Bit3 | Backlight Signal | 0 | Low Level | 0±0.2V | A3 | I158 |
| 1 | High Level | >10V | A3 | I158 |
| Bit4 | Reserved4 | 1 | N.A. | N.A. | N.A. | N.A. |
| 1 | N.A. | N.A. | N.A. | N.A. |
| Bit5 | Reserved5 | 1 | N.A. | N.A. | N.A. | N.A. |
| 1 | N.A. | N.A. | N.A. | N.A. |
| Bit6 | Reserved6 | 1 | N.A. | N.A. | N.A. | N.A. |
| 1 | N.A. | N.A. | N.A. | N.A. |
| Bit7 | Reserved7 | 1 | N.A. | N.A. | N.A. | N.A. |
| 1 | N.A. | N.A. | N.A. | N.A. |

Remark:

Bit1~Bit2 are reserved. They are set to 0.

Bit4~Bit7 are reserved. They could only be set to 1 because of design of circuit.

### Control the PWM/Frequency output for periphery components

**N.A.**

### Read the PWM/ Frequency input

**N.A.**

## NAT System Test

**N.A.**

## Key and Knob Test

### Read the Key when KEY Pressed

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

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Byte0** | **Byte1** | **Byte2** | **Byte3** | **Byte4** | **Byte5** | **Byte6** | **Byte7** |
|  | **Service**  **Type** | **Read Out**  **ID** | **Read Out**  **Sub Type** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** |
| **0x7C0** | **0x22** | **0x03** | **0x01** | **0x00** | **0x00** | **0x00** | **0x00** | **0x00** |

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

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Byte0** | **Byte1** | **Byte2** | **Byte3** | **Byte4** | **Byte5** | **Byte6** | **Byte7** |
|  | **Service**  **Type** | **Read Out**  **ID** | **Read Out**  **Sub Type** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** |
| **0x7D0** | **0x62** | **0x03** | **0x01** | **0xXX** | **0xXX** | **0xXX** | **0x00** | **0x00** |

**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** |
| Data0 | Bit.0 | OFF Key | No / Pressed | 0/1 | I103 |
| Bit.1 | DEF Key | No / Pressed | 0/1 | I98 |
| Bit.2 | AUTO Key | No / Pressed | 0/1 | I96 |
| Bit.3 | N.A. | No / Pressed | 0/1 | N.A. |
| Bit.4 | MODE Key | No / Pressed | 0/1 | I102 |
| Bit.5 | N.A. | No / Pressed | 0/1 | N.A. |
| Bit.6 | AC Key | No / Pressed | 0/1 | I97 |
| Bit.7 | REC Key | No / Pressed | 0/1 | I101 |
| Data1 | Bit.0 | FRESH Key | No / Pressed | 0/1 | I99 |
| Bit.1 | RDEF Key | No / Pressed | 0/1 | I100 |
| Bit.2 | N.A. | No / Pressed | 0/1 | N.A. |
| Bit.3 | N.A. | No / Pressed | 0/1 | N.A. |
| Bit.4 | N.A. | No / Pressed | 0/1 | N.A. |
| Bit.5 | N.A. | No / Pressed | 0/1 | N.A. |
| Bit.6 | N.A. | No / Pressed | 0/1 | N.A. |
| Bit.7 | N.A. | No / Pressed | 0/1 | N.A. |
| Data3 | Bit.0 | N.A. | No / Pressed | 0/1 | N.A. |
| Bit.1 | N.A. | No / Pressed | 0/1 | N.A. |
| Bit.2 | N.A. | No / Pressed | 0/1 | N.A. |
| Bit.3 | N.A. | No / Pressed | 0/1 | N.A. |
| Bit.4 | N.A. | No / Pressed | 0/1 | N.A. |
| Bit.5 | N.A. | No / Pressed | 0/1 | N.A. |
| Bit.6 | N.A. | No / Pressed | 0/1 | N.A. |
| Bit.7 | N.A. | No / Pressed | 0/1 | N.A. |

### Knob Test

**Step1：The following command can be used to enter the Knob test.**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Byte0** | **Byte1** | **Byte2** | **Byte3** | **Byte4** | **Byte5** | **Byte6** | **Byte7** |
|  | **Service**  **Type** | **Control**  **ID** | **Control**  **Sub Type** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** |
| **0x7C0** | **0x30** | **0x2C** | **0x01** | **0x00** | **0x00** | **0x00** | **0x00** | **0x00** |

**When receive the command, the ECU will clear all the knobs state and counter.**

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

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Byte0** | **Byte1** | **Byte2** | **Byte3** | **Byte4** | **Byte5** | **Byte6** | **Byte7** |
|  | **Service**  **Type** | **Control**  **ID** | **Control**  **Sub Type** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** |
| **0x7D0** | **0x70** | **0x2C** | **0x01** | **0x00** | **0x00** | **0x00** | **0x00** | **0x00** |

**Step 2: The following command can be used to read the knob counter.**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Byte0** | **Byte1** | **Byte2** | **Byte3** | **Byte4** | **Byte5** | **Byte6** | **Byte7** |
|  | **Service**  **Type** | **Read Out**  **ID** | **Read Out**  **Sub Type** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** |
| **0x7C0** | **0x22** | **0x04** | **0xXX** | **0x00** | **0x00** | **0x00** | **0x00** | **0x00** |

**When receive the command, the ECU will response the current knob counter.**

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

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Byte0** | **Byte1** | **Byte2** | **Byte3** | **Byte4** | **Byte5** | **Byte6** | **Byte7** |
|  | **Service**  **Type** | **Read Out**  **ID** | **Read Out**  **Sub Type** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** |
| **0x7D0** | **0x62** | **0x04** | **0xXX** | **0xXX** | **0xXX** | **0x00** | **0x00** | **0x00** |

|  |  |
| --- | --- |
| **Read Out**  **Sub Type** | **Description** |
| 0x01 | Left temperature knob |
| 0x02 | N.A. |
| 0x03 | Blower speed knob |
| 0x04 | N.A. |
| 0xXX | N.A. |

|  |  |
| --- | --- |
| **Data field** | **Description** |
| Data0 | CW counter(顺时针计数)  Remark：There 30 steps for 1 whole round. |
| Data1 | CCW counter（逆时针计数）  Remark：There 30 steps for 1 whole round. |

**Step 3: The following command can be used to exit the knob test.**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Byte0** | **Byte1** | **Byte2** | **Byte3** | **Byte4** | **Byte5** | **Byte6** | **Byte7** |
|  | **Service**  **Type** | **Control**  **ID** | **Control**  **Sub Type** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** |
| **0x7C0** | **0x30** | **0x2C** | **0xFF** | **0x00** | **0x00** | **0x00** | **0x00** | **0x00** |

**When receive the command, the ECU will clear the knob counter.**

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

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Byte0** | **Byte1** | **Byte2** | **Byte3** | **Byte4** | **Byte5** | **Byte6** | **Byte7** |
|  | **Service**  **Type** | **Control**  **ID** | **Control**  **Sub Type** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** |
| **0x7D0** | **0x70** | **0x2C** | **0xFF** | **0x00** | **0x00** | **0x00** | **0x00** | **0x00** |

**注：此测试针对于可360度旋转的无极性旋钮，对于电位器的旋钮需要在Analogue Part 章节体现。**

## Actuator Output Test

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

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Byte0** | **Byte1** | **Byte2** | **Byte3** | **Byte4** | **Byte5** | **Byte6** | **Byte7** |
|  | **Service**  **Type** | **Control**  **ID** | **Control**  **Sub Type** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** |
| **0x7C0** | **0x30** | **0x26** | **0xXX** | **0xXX** | **0xXX** | **0xXX** | **0xXX** | **0xXX** |

**When receive the command, the ECU will control the actuators.**

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

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Byte0** | **Byte1** | **Byte2** | **Byte3** | **Byte4** | **Byte5** | **Byte6** | **Byte7** |
|  | **Service**  **Type** | **Control**  **ID** | **Control**  **Sub Type** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** |
| **0x7D0** | **0x70** | **0x26** | **0xXX** | **0xXX** | **0xXX** | **0xXX** | **0xXX** | **0xXX** |

**Command for DC Motors:**

| **Control**  **Sub Type** | **Data0~Data4** | **Motor**  **Description** | **Direction** | **Data0** | **Test Pin** | **Voltage** |
| --- | --- | --- | --- | --- | --- | --- |
| 0x01 | Data0 | Left Mac Motor | (Stop) | 0x00 | A11->A12 | 0±0.2V |
| (Clockwise) | 0x01 | A11->A12 | >10V |
| (Anti-Clockwise) | 0x02 | A11->A12 | <-10V |
| Data1 | Right Mac Motor | (Stop) | 0x00 | N.A. | N.A. |
| (Clockwise) | 0x01 | N.A. | N.A. |
| (Anti-Clockwise) | 0x02 | N.A. | N.A. |
| Data2 | Mode Motor | (Stop) | 0x00 | A10->A9 | 0±0.2V |
| (Clockwise) | 0x01 | A10->A9 | >10V |
| (Anti-Clockwise) | 0x02 | A10->A9 | <-10V |
| Data3 | RFA Motor | (Stop) | 0x00 | A7->A8 | 0±0.2V |
| (Clockwise) | 0x01 | A7->A8 | >10V |
| (Anti-Clockwise) | 0x02 | A7->A8 | <-10V |

## Blower Test

**The following command can be used to control the blower speed.**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Byte0** | **Byte1** | **Byte2** | **Byte3** | **Byte4** | **Byte5** | **Byte6** | **Byte7** |
|  | **Service**  **Type** | **Control**  **ID** | **Control**  **Sub Type** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** |
| **0x7C0** | **0x30** | **0x25** | **0xXX** | **0x00** | **0x00** | **0x00** | **0x00** | **0x00** |

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

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Byte0** | **Byte1** | **Byte2** | **Byte3** | **Byte4** | **Byte5** | **Byte6** | **Byte7** |
|  | **Service**  **Type** | **Control**  **ID** | **Control**  **Sub Type** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** |
| **0x7D0** | **0x70** | **0x25** | **0xXX** | **0x00** | **0x00** | **0x00** | **0x00** | **0x00** |

The data is defined as follow table (**0Bar, 1Bar,** 4Bar**,** 8Bar speed is must to be test):

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Signal Name** | **Level**  **(Infomation)** | **Control**  **Sub Type** | **Test Pin** | **Test point** | **Blower Voltage** |
| **Blower Control** | **0 Bar** | **0x00** | A35->A34 | I246->I245 | 0V±0.35V |
| **1Bar** | **0x01** | A35->A34 | I246->I245 | 4.03V±0.35V |
| **4 Bar** | **0x04** | A35->A34 | I246->I245 | 6.61V±0.35V |
| **8 Bar** | **0x08** | A35->A34 | I246->I245 | >12.5V |

## **UART Test**

N.A.

## **LED Test**

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

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Byte0** | **Byte1** | **Byte2** | **Byte3** | **Byte4** | **Byte5** | **Byte6** | **Byte7** |
|  | **Service**  **Type** | **Control**  **ID** | **Control**  **Sub Type** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** |
| **0x7C0** | **0x30** | **0x23** | **0x01** | **0xXX** | **0xXX** | **0xXX** | **0xXX** | **0x00** |

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

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Byte0** | **Byte1** | **Byte2** | **Byte3** | **Byte4** | **Byte5** | **Byte6** | **Byte7** |
|  | **Service**  **Type** | **Control**  **ID** | **Control**  **Sub Type** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** |
| **0x7D0** | **0x70** | **0x23** | **0x01** | **0xXX** | **0xXX** | **0xXX** | **0xXX** | **0x00** |

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

The value of each ‘BitX’ is used to control a relative LED.

| **NO** | **Description** | **Color** | **Control** | | | |
| --- | --- | --- | --- | --- | --- | --- |
| **TT\_bit** | **Value** | **Data0** | **Description** |
| 1 | Reversed telltale |  | Bit0 | 0 | 0x00 | Turn off |
| 0 | 0x00 | Turn off |
| 2 | Reversed telltale |  | Bit1 | 0 | 0x00 | Turn off |
| 0 | 0x00 | Turn off |
| 3 | Reversed telltale |  | Bit2 | 0 | 0x00 | Turn off |
| 0 | 0x00 | Turn off |
| 4 | Reversed telltale |  | Bit3 | 0 | 0x00 | Turn off |
| 0 | 0x00 | Turn off |
| 5 | Reversed telltale |  | Bit4 | 0 | 0x00 | Turn off |
| 0 | 0x00 | Turn off |
| 6 | Reversed telltale |  | Bit5 | 0 | 0x00 | Turn off |
| 0 | 0x00 | Turn off |
| 7 | Reversed telltale |  | Bit6 | 0 | 0x00 | Turn off |
| 0 | 0x00 | Turn off |
| 8 | Reversed telltale |  | Bit7 | 0 | 0x00 | Turn off |
| 0 | 0x00 | Turn off |

| **NO** | **Description** | **Color** | **Control** | | | |
| --- | --- | --- | --- | --- | --- | --- |
| **TT\_bit** | **Value** | **Data1** | **Description** |
| 9 | AUTO telltale |  | Bit0 | 0 | 0x00 | Turn off |
| 1 | 0x01 | Turn on |
| 10 | A/C telltale |  | Bit1 | 0 | 0x00 | Turn off |
| 1 | 0x02 | Turn on |
| 11 | Reversed telltale |  | Bit2 | 0 | 0x00 | Turn off |
| 0 | 0x00 | Turn off |
| 12 | FRE telltale |  | Bit3 | 0 | 0x00 | Turn off |
| 1 | 0x08 | Turn on |
| 13 | REC telltale |  | Bit4 | 0 | 0x00 | Turn off |
| 1 | 0x10 | Turn on |
| 14 | R-DEF telltale |  | Bit5 | 0 | 0x00 | Turn off |
| 1 | 0x20 | Turn on |
| 15 | Reversed telltale |  | Bit6 | 0 | 0x00 | Turn off |
| 0 | 0x00 | Turn off |
| 16 | F-DEF telltale |  | Bit7 | 0 | 0x00 | Turn off |
| 1 | 0x80 | Turn on |

| **NO** | **Description** | **Color** | **Control** | | | |
| --- | --- | --- | --- | --- | --- | --- |
| **TT\_bit** | **Value** | **Data2** | **Description** |
| 17 | Reversed telltale |  | Bit0 | 0 | 0x00 | Turn off |
| 0 | 0x00 | Turn off |
| 18 | Reversed telltale |  | Bit1 | 0 | 0x00 | Turn off |
| 0 | 0x00 | Turn off |
| 19 | Reversed telltale |  | Bit2 | 0 | 0x00 | Turn off |
| 0 | 0x00 | Turn off |
| 20 | Reversed telltale |  | Bit3 | 0 | 0x00 | Turn off |
| 0 | 0x00 | Turn off |
| 21 | Reversed telltale |  | Bit4 | 0 | 0x00 | Turn off |
| 0 | 0x00 | Turn off |
| 22 | Reversed telltale |  | Bit5 | 0 | 0x00 | Turn off |
| 0 | 0x00 | Turn off |
| 23 | Reversed telltale |  | Bit6 | 0 | 0x00 | Turn off |
| 0 | 0x00 | Turn off |
| 24 | Reversed telltale |  | Bit7 | 0 | 0x00 | Turn off |
| 0 | 0x00 | Turn off |

| **NO** | **Description** | **Color** | **Control** | | | |
| --- | --- | --- | --- | --- | --- | --- |
| **TT\_bit** | **Value** | **Data3** | **Description** |
| 25 | Reversed telltale |  | Bit0 | 0 | 0x00 | Turn off |
| 0 | 0x00 | Turn on |
| 26 | Reversed telltale |  | Bit1 | 0 | 0x00 | Turn off |
| 0 | 0x00 | Turn on |
| 27 | Reversed telltale |  | Bit2 | 0 | 0x00 | Turn off |
| 0 | 0x00 | Turn on |
| 28 | Reversed telltale |  | Bit3 | 0 | 0x00 | Turn off |
| 0 | 0x00 | Turn on |
| 29 | Reversed telltale |  | Bit4 | 0 | 0x00 | Turn off |
| 0 | 0x00 | Turn on |
| 30 | Reversed telltale |  | Bit5 | 0 | 0x00 | Turn off |
| 0 | 0x00 | Turn off |
| 31 | Reversed telltale |  | Bit6 | 0 | 0x00 | Turn off |
| 0 | 0x00 | Turn off |
| 32 | Reversed telltale |  | Bit7 | 0 | 0x00 | Turn off |
| 0 | 0x00 | Turn off |

**The following data need to be used for controlling the Led**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Data Field** | | | | **Description** |
| **Data0** | **Data1** | **Data2** | **Data3** |
| **0xFF** | **0xFF** | 0xFF | 0xFF | Turn on all the LED |
| **0x00** | **0x00** | 0x00 | 0x00 | Turn off all the LED |
| **0x00** | **0x23** | 0x00 | 0x12 | Turn on the odd LED, and turn off the even LED, including the 9, 10, 14, 26, 29 |
| **0x00** | **0x98** | 0x00 | 0x09 | Turn on the even LED, and turn off the odd LED, including12, 13, 16, 25, 28 |

## **LCD Test**

**N.A.**

## **Control PWM of Dimming**

**The PWM of dimming can be controlled with following command format:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Byte0** | **Byte1** | **Byte2** | **Byte3** | **Byte4** | **Byte5** | **Byte6** | **Byte7** |
|  | **Service**  **Type** | **Control**  **ID** | **Control**  **Sub Type** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** |
| **0x7C0** | **0x30** | **0x24** | **0xXX** | **0xXX** | **0x00** | **0x00** | **0x00** | **0x00** |

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

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Byte0** | **Byte1** | **Byte2** | **Byte3** | **Byte4** | **Byte5** | **Byte6** | **Byte7** |
|  | **Service**  **Type** | **Control**  **ID** | **Control**  **Sub Type** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** |
| **0x7D0** | **0x70** | **0x24** | **0xXX** | **0xXX** | **0x00** | **0x00** | **0x00** | **0x00** |

|  |  |
| --- | --- |
| **Control**  **Sub Type** | **Description** |
| 0x01 | Set Led PWM |
| 0x02 | N.A. |
| 0x03 | N.A. |

For Led, the PWM value of Day and Night as following:

|  |  |
| --- | --- |
| Data0 | Description |
| 0x64 | Set telltales PWM to 100% (Day mode) |
| 0x32 | Set telltales PWM to 50%(Night Mode) |

For Lcd, the PWM value of Day and Night as following:

N.A.

For Character backlight, the PWM value of Day and Night as following, for more detail information please refer to the Chapter **Illumination Module Measurement**

N.A.

## EEPROM Access

### Write Memory by Address

**Access EEPROM request:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Byte0** | **Byte1** | **Byte2** | **Byte3** | **Byte4** | **Byte5** | **Byte6** | **Byte7** |
|  | **Service**  **Type** | **Control**  **ID** | **AddH** | **AddL** | **LengthH** | **LengthL** |  |  |
| **0x7C0** | **0x3D** | **0x20** | **0xXX** | **0xXX** | **0xXX** | **0xXX** | **0x00** | **0x00** |

**Receive data:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Byte0** | **Byte1** | **Byte2** | **Byte3** | **Byte4** | **Byte5** | **Byte6** | **Byte7** |
|  | **Service**  **Type** | **Control**  **ID** | **AddrH** | **AddrL** | **LengthH** | **LengthL** |  |  |
| **0x7D0** | **0x7D** | **0x20** | **0xXX** | **0xXX** | **0xXX** | **0xXX** | **0x00** | **0x00** |

写的起始地址必须为偶地址且>=0x3C0, <0x3FF，写的长度必须是2的整数倍并且不能够超过64个字节。

**Access EEPROM successfully.**

16 bytes will be written into EEPRPM at most in one time operation.：一次最多只能够写16个字节到EEPROM中。

在接收到正响应后发送第一帧数据：

**Send data:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Byte0** | **Byte1** | **Byte2** | **Byte3** | **Byte4** | **Byte5** | **Byte6** | **Byte7** |
|  | **Packet ID** | **Data**  **Length** | **AddrH** | **AddrL** | **Data**  **Format** | **EEP0** | **EEP1** | **EEP2** |
| **0x7C0** | **0x01** | **0xXX** | **0xXX** | **0xXX** | **0x00** | **0xXX** | **0xXX** | **0xXX** |

**Receive data:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Byte0** | **Byte1** | **Byte2** | **Byte3** | **Byte4** | **Byte5** | **Byte6** | **Byte7** |
|  | **Packet ID** |  |  |  |  |  |  |  |
| **0x7D0** | **0x41** | **0x01** | **0x00** | **0x00** | **0x00** | **0x00** | **0x00** | **0x00** |

在接收到正响应后发送第二帧数据

**Send data:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Byte0** | **Byte1** | **Byte2** | **Byte3** | **Byte4** | **Byte5** | **Byte6** | **Byte7** |
|  | **PacketID** | **EEP3** | **EEP4** | **EEP5** | **EEP6** | **EEP7** | **EEP8** | **EEP9** |
| **0x7C0** | **0x02** | **0xXX** | **0xXX** | **0xXX** | **0xXX** | **0xXX** | **0xXX** | **0xXX** |

**Receive data:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Byte0** | **Byte1** | **Byte2** | **Byte3** | **Byte4** | **Byte5** | **Byte6** | **Byte7** |
|  | **PacketID** |  |  |  |  |  |  |  |
| **0x7D0** | **0x42** | **0x01** | **0xXX** | **0xXX** | **0xXX** | **0xXX** | **0xXX** | **0xXX** |

在接收到正响应后发送第三帧数据

**Send data:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Byte0** | **Byte1** | **Byte2** | **Byte3** | **Byte4** | **Byte5** | **Byte6** | **Byte7** |
|  | **PacketID** | **EEP10** | **EEP11** | **EEP12** | **EEP13** | **EEP14** | **EEP15** | **EEPEOM**  **Checksum** |
| **0x7C0** | **0x03** | **0xXX** | **0xXX** | **0xXX** | **0xXX** | **0xXX** | **0xXX** | **0xXX** |

**Receive data:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Byte0** | **Byte1** | **Byte2** | **Byte3** | **Byte4** | **Byte5** | **Byte6** | **Byte7** |
|  | **PacketID** |  |  |  |  |  |  |  |
| **0x7D0** | **0x43** | **0x01** | **0xXX** | **0xXX** | **0xXX** | **0xXX** | **0xXX** | **0xXX** |

在接收到正响应后发送结束帧

**Send data:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Byte0** | **Byte1** | **Byte2** | **Byte3** | **Byte4** | **Byte5** | **Byte6** | **Byte7** |
| **0x7C0** | **0x01** | **0x00** | **0x00** | **0x00** | **0x01** | **0xFF** | **0x00** | **0x00** |

**Receive data:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Byte0** | **Byte1** | **Byte2** | **Byte3** | **Byte4** | **Byte5** | **Byte6** | **Byte7** |
| **0x7D0** | **0x41** | **0x01** | **0x00** | **0x00** | **0x00** | **0x00** | **0x00** | **0x00** |

在接收到正响应后表明数据已经成功写入EERPOM

备注：EEPEOM Checksum计算

EEPEOM Checksum =(~((DataLength + AddrH + AddrL + DataType + Data[0] + **……** Data[n]) % 0x100))+1

**For example**:

16 bytes of information which is 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 10 will be written into EEPROM with start address: 0x03C0（0x03C0 = 960, 960/2 = 480,可以被2整除）.

**Access EEPROM request:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Byte0** | **Byte1** | **Byte2** | **Byte3** | **Byte4** | **Byte5** | **Byte6** | **Byte7** |
|  | **Service**  **Type** | **Control**  **ID** | **AddH** | **AddL** | **LengthH** | **LengthL** |  |  |
| **0x7C0** | **0x3D** | **0x20** | **0x03** | **0xC0** | **0x00** | **0x10** | **0x00** | **0x00** |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Byte0** | **Byte1** | **Byte2** | **Byte3** | **Byte4** | **Byte5** | **Byte6** | **Byte7** |
|  | **Service**  **Type** | **Control**  **ID** | **AddH** | **AddL** | **LengthH** | **LengthL** |  |  |
| **0x7D0** | **0x7D** | **0x20** | **0x03** | **0xC0** | **0x00** | **0x10** | **0x00** | **0x00** |

在接收到正响应后开始发送第一帧数据：

**Send data:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Byte0** | **Byte1** | **Byte2** | **Byte3** | **Byte4** | **Byte5** | **Byte6** | **Byte7** |
|  | **Packet ID** | **Data**  **Length** | **AddH** | **AddL** | **Data**  **Format** | **EEP0** | **EEP1** | **EEP2** |
| **0x7C0** | **0x01** | **0x10** | **0x03** | **0xC0** | **0x00** | **0x01** | **0x02** | **0x03** |

**Receive data:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Byte0** | **Byte1** | **Byte2** | **Byte3** | **Byte4** | **Byte5** | **Byte6** | **Byte7** |
|  | **Packet ID** |  |  |  |  |  |  |  |
| **0x7D0** | **0x41** | **0x01** | **0x00** | **0x00** | **0x00** | **0x00** | **0x00** | **0x00** |

在接收到正响应后发送第二帧数据

**Send data:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Byte0** | **Byte1** | **Byte2** | **Byte3** | **Byte4** | **Byte5** | **Byte6** | **Byte7** |
|  | **Packet ID** | **EEP3** | **EEP4** | **EEP5** | **EEP6** | **EEP7** | **EEP8** | **EEP9** |
| **0x7C0** | **0x02** | **0x04** | **0x05** | **0x06** | **0x07** | **0x08** | **0x09** | **0x0A** |

**Receive data:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Byte0** | **Byte1** | **Byte2** | **Byte3** | **Byte4** | **Byte5** | **Byte6** | **Byte7** |
|  | **Packet ID** |  |  |  |  |  |  |  |
| **0x7D0** | **0x42** | **0x01** | **0x00** | **0x00** | **0x00** | **0x00** | **0x00** | **0x00** |

在接收到正响应后发送第三帧数据

**Send data:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Byte0** | **Byte1** | **Byte2** | **Byte3** | **Byte4** | **Byte5** | **Byte6** | **Byte7** |
|  | **Packet ID** | **EEP10** | **EEP11** | **EEP12** | **EEP13** | **EEP14** | **EEP15** | **EEP CheckSum** |
| **0x7C0** | **0x03** | **0x0B** | **0x0C** | **0x0D** | **0x0E** | **0x0F** | **0x10** | **0xA5** |

**Receive data:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Byte0** | **Byte1** | **Byte2** | **Byte3** | **Byte4** | **Byte5** | **Byte6** | **Byte7** |
|  | **Packet ID** |  |  |  |  |  |  |  |
| **0x7D0** | **0x43** | **0x01** | **0x00** | **0x00** | **0x00** | **0x00** | **0x00** | **0x00** |

在接收到正响应后发送结束帧

**Send data:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Byte0** | **Byte1** | **Byte2** | **Byte3** | **Byte4** | **Byte5** | **Byte6** | **Byte7** |
| **0x7C0** | **0x01** | **0x00** | **0x00** | **0x00** | **0x01** | **0xFF** | **0x00** | **0x00** |

**Receive data:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Byte0** | **Byte1** | **Byte2** | **Byte3** | **Byte4** | **Byte5** | **Byte6** | **Byte7** |
| **0x7D0** | **0x41** | **0x01** | **0x00** | **0x00** | **0x00** | **0x00** | **0x00** | **0x00** |

已经成功把16字节FF写入到了起始地址在0x03C0 EEPROM中。

上例中把16个字节FF写入到0x3C0为起始地址的EEPROM中chencksum计算结果为

EEPEOM Checksum = (~((0x00 +0x10 + 0x03 + 0xC0 + 0x01+ 0x02+ 0x03+ 0x04+ 0x05+ 0x06+ 0x07+ 0x08+ 0x09+ 0x0A+ 0x0B+ 0x0C+ 0x0D+ 0x0E+ 0x0F+ 0x10)%0x100))+1 = 0xA5

* + 1. Read Memory by Address

**Send data**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Byte0** | **Byte1** | **Byte2** | **Byte3** | **Byte4** | **Byte5** | **Byte6** | **Byte7** |
|  | **Service**  **Type** | **Control**  **ID** | **AddrH** | **AddrL** | **DataLenH** | **DataLenL** |  |  |
| **0x7C0** | **0x23** | **0x10** | **0xXX** | **0xXX** | **0xXX** | **0xXX** | **0x00** | **0x00** |

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

**Receive Data**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Byte0** | **Byte1** | **Byte2** | **Byte3** | **Byte4** | **Byte5** | **Byte6** | **Byte7** |
|  | **Service**  **Type** | **Control**  **ID** | **AddrH** | **AddrL** | **DataLenH** | **DataLenL** |  |  |
| **0x7D0** | **0x63** | **0x10** | **0xXX** | **0xXX** | **0xXX** | **0xXX** | **0x00** | **0x00** |

读取的起始地址以及长度都必须为2的整数倍且起始地址必须>=0x3C0，<0x3FF。读的起始地址必须为偶地址且>=0x3C0, <0x3FF，写的长度必须是2的整数倍并且不能够超过64个字节。

Receive eeprom data

**Frame1:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Byte0** | **Byte1** | **Byte2** | **Byte3** | **Byte4** | **Byte5** | **Byte6** | **Byte7** |
|  | **Packet ID** |  |  |  |  |  |  |  |
| **0x7C0** | **0x01** | **0x00** | **0x00** | **0x00** | **0x00** | **0x00** | **0x00** | **0x00** |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Byte0** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** | **Data5** | **Data6** |
|  | **Packet ID** | **Data**  **Length** | **AddrH** | **AddrL** | **Data**  **Format** | **EEP0** | **EEP1** | **EEP2** |
| **0x7D0** | **0x01** | **0xXX** | **0xXX** | **0xXX** | **0x00** | **0xXX** | **0xXX** | **0xXX** |

**备注**：如果请求读取EEPROM的数据长度大于0x10则DataLength=0x10，否则DataLength=读取的长度。

**Frame2:**

**Send Data:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Byte0** | **Byte1** | **Byte2** | **Byte3** | **Byte4** | **Byte5** | **Byte6** | **Byte7** |
|  | **Packet ID** |  |  |  |  |  |  |  |
| **0x7C0** | **0x02** | **0x00** | **0x00** | **0x00** | **0x00** | **0x00** | **0x00** | **0x00** |

**Receive Data:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Byte0** | **Byte1** | **Byte2** | **Byte3** | **Byte4** | **Byte5** | **Byte6** | **Byte7** |
|  | **Packet ID** | **EEP3** | **EEP4** | **EEP5** | **EEP6** | **EEP7** | **EEP8** | **EEP9** |
| **0x7D0** | **0x02** | **0xXX** | **0xXX** | **0xXX** | **0xXX** | **0xXX** | **0xXX** | **0xXX** |

**Frame3:**

**Send Data:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Byte0** | **Byte1** | **Byte2** | **Byte3** | **Byte4** | **Byte5** | **Byte6** | **Byte7** |
|  | **Packet ID** |  |  |  |  |  |  |  |
| **0x7C0** | **0x03** | **0x00** | **0x00** | **0x00** | **0x00** | **0x00** | **0x00** | **0x00** |

**Receive Data:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Byte0** | **Byte1** | **Byte2** | **Byte3** | **Byte4** | **Byte5** | **Byte6** | **Byte7** |
|  | **Packet ID** | **EEP10** | **EEP11** | **EEP12** | **EEP13** | **EEP14** | **EEP15** | **EEP**  **CheckSum** |
| **0x7D0** | **0x03** | **0xXX** | **0xXX** | **0xXX** | **0xXX** | **0xXX** | **0xXX** | **0xXX** |

**End Frame:**

**Send Data:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Byte0** | **Byte1** | **Byte2** | **Byte3** | **Byte4** | **Byte5** | **Byte6** | **Byte7** |
| **0x7C0** | **0x01** | **0x00** | **0x00** | **0x00** | **0x01** | **0xFF** | **0x00** | **0x00** |

**Receive Data:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Byte0** | **Byte1** | **Byte2** | **Byte3** | **Byte4** | **Byte5** | **Byte6** | **Byte7** |
| **0x7D0** | **0x01** | **0x00** | **0x00** | **0x00** | **0x01** | **0xFF** | **0x00** | **0x00** |

**For example**:

16 bytes of information which is 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 10 is successfully written into EEPROM with start address with start address:0x03C0, and now it will be read from EEPROM.

**Send data**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Byte0** | **Byte1** | **Byte2** | **Byte3** | **Byte4** | **Byte5** | **Byte6** | **Byte7** |
|  | **Service**  **Type** | **Control**  **ID** | **AddrH** | **AddrL** | **LengthH** | **LengthL** |  |  |
| **0x7C0** | **0x23** | **0x10** | **0x03** | **0xC0** | **0x00** | **0x0C** | **0x00** | **0x00** |

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

**Receive Data**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Byte0** | **Byte1** | **Byte2** | **Byte3** | **Byte4** | **Byte5** | **Byte6** | **Byte7** |
|  | **Service**  **Type** | **Control**  **ID** | **AddrH** | **AddrL** | **LengthH** | **LengthL** |  |  |
| **0x7D0** | **0x63** | **0x10** | **0x03** | **0xC0** | **0x00** | **0x0C** | **0x00** | **0x00** |

Receive eeprom data

**Frame1:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Byte0** | **Byte1** | **Byte2** | **Byte3** | **Byte4** | **Byte5** | **Byte6** | **Byte7** |
|  | **Packet ID** |  |  |  |  |  |  |  |
| **0x7C0** | **0x01** | **0x00** | **0x00** | **0x00** | **0x00** | **0x00** | **0x00** | **0x00** |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Byte0** | **Byte1** | **Byte2** | **Byte3** | **Byte4** | **Byte5** | **Byte6** | **Byte7** |
|  | **Packet ID** | **Data Length** | **AddrH** | **AddrL** | **Data**  **Format** | **EEP0** | **EEP1** | **EEP2** |
| **0x7D0** | **0x01** | **0x0C** | **0x03** | **0xC0** | **0x00** | **0x01** | **0x02** | **0x03** |

**Frame2:**

**Send Data:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Byte0** | **Byte1** | **Byte2** | **Byte3** | **Byte4** | **Byte5** | **Byte6** | **Byte7** |
|  | **Packet ID** |  |  |  |  |  |  |  |
| **0x7C0** | **0x02** | **0x00** | **0x00** | **0x00** | **0x00** | **0x00** | **0x00** | **0x00** |

**Receive Data:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Byte0** | **Byte1** | **Byte2** | **Byte3** | **Byte4** | **Byte5** | **Byte6** | **Byte7** |
|  | **Packet ID** | **EEP3** | **EEP4** | **EEP5** | **EEP6** | **EEP7** | **EEP8** | **EEP9** |
| **0x7D0** | **0x02** | **0x04** | **0x05** | **0x06** | **0x07** | **0x08** | **0x09** | **0x0A** |

**Frame3:**

**Send Data:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Byte0** | **Byte1** | **Byte2** | **Byte3** | **Byte4** | **Byte5** | **Byte6** | **Byte7** |
|  | **Packet ID** |  |  |  |  |  |  |  |
| **0x7C0** | **0x03** | **0x00** | **0x00** | **0x00** | **0x00** | **0x00** | **0x00** | **0x00** |

**Receive Data:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Byte0** | **Byte1** | **Byte2** | **Byte3** | **Byte4** | **Byte5** | **Byte6** | **Byte7** |
|  | **Packet ID** | **EEP10** | **EEP11** | **EEP12** | **EEP13** | **EEP14** | **EEP15** | **EEP checksum** |
| **0x7D0** | **0x03** | **0x0B** | **0x0C** | **0x0D** | **0x0E** | **0x0F** | **0x10** | **0xA5** |

**End Frame:**

**Send Data:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Byte0** | **Byte1** | **Byte2** | **Byte3** | **Byte4** | **Byte5** | **Byte6** | **Byte7** |
| **0x7C0** | **0x01** | **0x00** | **0x00** | **0x00** | **0x01** | **0xFF** | **0x00** | **0x00** |

**Receive Data:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Byte0** | **Byte1** | **Byte2** | **Byte3** | **Byte4** | **Byte5** | **Byte6** | **Byte7** |
| **0x7D0** | **0x01** | **0x00** | **0x00** | **0x00** | **0x01** | **0xFF** | **0x00** | **0x00** |

上例中把12个字节从0x3C0为起始地址的EEPROM中读取出来，读取出来的值都为0xFF，

chencksum计算结果为

EEPEOM Checksum = (~((0x00 +0x10 + 0x03 + 0xC0 + 0x01+ 0x02+ 0x03+ 0x04+ 0x05+ 0x06+ 0x07+ 0x08+ 0x09+ 0x0A+ 0x0B+ 0x0C+ 0x0D+ 0x0E+ 0x0F+ 0x10)%0x100))+1 = 0xA5

## Read Software Version

**The following command can be used for reading the software version:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Byte0** | **Byte1** | **Byte2** | **Byte3** | **Byte4** | **Byte5** | **Byte6** | **Byte7** |
|  | **Service**  **Type** | **Control**  **ID** | **Control**  **Sub Type** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** |
| **0x7C0** | **0x22** | **0x01** | **0xXX** | **0x00** | **0x00** | **0x00** | **0x00** | **0x00** |

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

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Byte0** | **Byte1** | **Byte2** | **Byte3** | **Byte4** | **Byte5** | **Byte6** | **Byte7** |
|  | **Service**  **Type** | **Control**  **ID** | **Control**  **Sub Type** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** |
| **0x7D0** | **0x62** | **0x01** | **0xXX** | **0xXX** | **0xXX** | **0xXX** | **0xXX** | **0xXX** |

|  |  |
| --- | --- |
| **Control**  **Sub Type** | **Description** |
| 0x01 | Internal software version |
| 0x02 | N.A. |

1. **Read internal Software Version //读取内部软件版本号**

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

|  |  |
| --- | --- |
| **Data field** | **Description** |
| Data[0] | Milestone information |
| Data[1] | SubState infromation |
| Data[2] | Year |
| Data[3] | Month |
| Data[4] | Day |

| **MileStone** | **Data Analysis** | **Data** | **Description** |
| --- | --- | --- | --- |
| (Data0>>4)&0x0F | 0 | A |
| 1 | B |
| 2 | C |
| 3 | D |
| Data0&0x0F | 0 | 0 |
| 1 | 1 |
| 2 | 2 |
| 3 | 3 |
| SubState | Data1 | 0~256 | 0~256 |
| Year | Data2 | 0x00~0x63 | Year:2000~2099 |
| Month | Data3 | 0x01~0x0C | Month:1~12 |
| Day | Data4 | 0x01~0x1F | Day: 1~31 |

**Example:**

If Data[0] = 0x12,Data[1]=0x03

MileStone:

(0x12>>4)&0x0F=1, it means B;

0x12&0x0F=2, so milestone is B2.

SubState:

Data[1]=0x03.

So this software version is B2.3.

Data2~Data4: 0x15, 0x0B, 0x12, so this is 2015-12-18

1. **Read external Software Version //读取外部软件版本号**

N.A.

## Write Product Information

**For some project, if related product information need to be wrote, please use the following command.**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Byte0** | **Byte1** | **Byte2** | **Byte3** | **Byte4** | **Byte5** | **Byte6** | **Byte7** |
|  | **Service**  **Type** | **Control**  **ID** | **Control**  **Sub Type** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** |
| **0x7C0** | **0x30** | **0x2A** | **0xXX** | **0xXX** | **0xXX** | **0xXX** | **0xXX** | **0xXX** |

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

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Byte0** | **Byte1** | **Byte2** | **Byte3** | **Byte4** | **Byte5** | **Byte6** | **Byte7** |
|  | **Service**  **Type** | **Control**  **ID** | **Control**  **Sub Type** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** |
| **0x7D0** | **0x70** | **0x2A** | **0xXX** | **0xXX** | **0xXX** | **0xXX** | **0xXX** | **0xXX** |

|  |  |  |  |
| --- | --- | --- | --- |
| Control  Sub Type | Data | Value | Description |
| 0x01 | Data0 | 0xXX | **Write the Model ID**  Set Data0 to 0x02: FC1616/66  …… |
| Data1 | 0x00(No used) |
| Data2 | 0x00(No used) |
| Data3 | 0x00(No used) |
| Data4 | 0x00(No used) |
| 0x02 | Data0 | 0xXX | **Write the Serial Number**  For example, if the serial number is  0123456789  The CtrlData0 ~CtrlData4 should be:  0x01,0x23,0x45,0x67,0x89 |
| Data1 | 0xXX |
| Data2 | 0xXX |
| Data3 | 0xXX |
| Data4 | 0xXX |
| 0x07 | Data0 | 0xXX | Write Hardware Number  For example, hardware number is 01 or 11  Set Data0 to 0x01: 01  Set Data0 to 0x11: 11 |
| Data1 | 0x00(No used) |
| Data2 | 0x00(No used) |
| Data3 | 0x00(No used) |
| Data4 | 0x00(No used) |

**Note: For most of the projects, may be Control Sub Type (0x01~0x06) is no used, so just delete for you project.**

## Read Product Information

**The following command can be used for reading the product information:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Byte0** | **Byte1** | **Byte2** | **Byte3** | **Byte4** | **Byte5** | **Byte6** | **Byte7** |
|  | **Service**  **Type** | **Control**  **ID** | **Control**  **Sub Type** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** |
| **0x7C0** | **0x22** | **0x08** | **0x01** | **0x00** | **0x00** | **0x00** | **0x00** | **0x00** |

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

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Byte0** | **Byte1** | **Byte2** | **Byte3** | **Byte4** | **Byte5** | **Byte6** | **Byte7** |
|  | **Service**  **Type** | **Control**  **ID** | **Control**  **Sub Type** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** |
| **0x7D0** | **0x62** | **0x08** | **0x01** | **0xXX** | **0xXX** | **0xXX** | **0xXX** | **0xXX** |

|  |  |  |  |
| --- | --- | --- | --- |
| Control  Sub Type | Data | Value | Description |
| 0x01 | Data0 | 0xXX | **Read the Model Name**  Data0~Data1 is 0x46,0x43,0x16,0x16,0x66,  the model name is FC1616/66. |
| Data1 | 0xXX |
| Data2 | 0xXX |
| Data3 | 0xXX |
| Data4 | 0xXX |
| 0x02 | Data0 | 0xXX | **Read the Serial Number**  For example,  The Data0 ~Data4 should be:  0x01,0x23,0x45,0x67,0x89  the serial number is  0123456789 |
| Data1 | 0xXX |
| Data2 | 0xXX |
| Data3 | 0xXX |
| Data4 | 0xXX |
| 0x07 | Data0 | 0xXX | Read Hardware Number  For example,  1. The Data0 is 0x01  the hardware number is 01  2. The Data0 is 0x11  the hardware number is 11 |
| Data1 | 0xXX(No used) |
| Data2 | 0xXX(No used) |
| Data3 | 0xXX(No used) |
| Data4 | 0xXX(No used) |
| …… | …… | …… | …… |

**Note: For all the projects, the Control Sub Type = 0x01(Read the model name) is needed. If the Control Sub Type (0x02~0x06) is not used for your project, just delete them.**

## Factory Setting

**The following command will be used to setting the ECU before factory in the last step.**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Byte0** | **Byte1** | **Byte2** | **Byte3** | **Byte4** | **Byte5** | **Byte6** | **Byte7** |
|  | **Service**  **Type** | **Control**  **ID** | **Control**  **Sub Type** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** |
| **0x7C0** | **0x30** | **0x2D** | **0xXX** | **0x00** | **0x00** | **0x00** | **0x00** | **0x00** |

**When the ECU setting successfully, then will answer with the acknowledgement sent as following:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Byte0** | **Byte1** | **Byte2** | **Byte3** | **Byte4** | **Byte5** | **Byte6** | **Byte7** |
|  | **Service**  **Type** | **Control**  **ID** | **Control**  **Sub Type** | **Data0** | **Data1** | **Data2** | **Data3** | **Data4** |
| **0x7D0** | **0x70** | **0x2D** | **0xXX** | **0x00** | **0x00** | **0x00** | **0x00** | **0x00** |

**When the ECU setting not successfully, then will answer with the acknowledgement sent as following:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Byte0** | **Byte1** | **Byte2** | **Byte3** | **Byte4** | **Byte5** | **Byte6** | **Byte7** |
|  | **Neg Resp** | **Service**  **Type** | **Control**  **ID** | **Control**  **Sub Type** | **Data1** | **Data2** | **Data3** | **Data4** |
| **0x7D0** | **0x7F** | **0x30** | **0x2D** | **0xXX** | **0x00** | **0x00** | **0x00** | **0x00** |

|  |  |
| --- | --- |
| **Control**  **Sub Type** | **Description** |
| 0x01 | The HMI will be restored to the default setting before the product sale |
| 0x02 | The DTC will be cleared before the product sale. |

When received the ECU successful answer, do not exit the PTS mode, just wait for **500**ms, and then turn the KL15 & KL30 off directly.

**Note:**

1. The command use for HMI restored the default setting must be executed for the project that HMI will be stored in the EEPROM when power off.

2. The command use for clearing the DTC is necessary except the projects that nothing to be saved in the EEPROM.