

**Delphi Electronics & Safety**

**Powertrain Gas Product Line**

Chery MT22P3

SOH(State of Health) Functinal

Test Report V1.0

Revision log:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Revision** | **Date** | **Author** | **Software Version** | **Description** |
| V1.0 | 24 June 2015 | Fan Daowei | chery\_mt22.3\_fix@121\_improve\_immo\_20150617.zip | SOH Test Report |
|  |  |  |  |  |

Contents

[1. General Information 4](#_Toc422940429)

[2.1 Diagnostic Need 4](#_Toc422940430)

[2.2 Basic Requirements of the ETC SOH 4](#_Toc422940431)

[2.3 Software Data Structure And Variables 5](#_Toc422940432)

[2.4 Required Equipment 6](#_Toc422940433)

[2.5 Mapping Table, SOH fault mapped to Diagnose Variable Interface 6](#_Toc422940434)

[2. Test Cases 7](#_Toc422940435)

[2.1 Test Case 1: Verify ETC SOH normal operation (1) 7](#_Toc422940436)

[2.2 Test Case 2: Verify ETC SOH normal operation (2) 10](#_Toc422940437)

[2.3 Test Case 3: Verify proper validation of system clock 12](#_Toc422940438)

[2.4 Test Case 4: Verify proper validation of ETC SOH interrupt source 14](#_Toc422940439)

[2.5 Test Case 5: Verify proper validation of Real Time Interrupt (RTI) frequency 16](#_Toc422940440)

[2.6 Test Case 6: Verify proper validation of CPU loop sequence 18](#_Toc422940441)

[2.7 Test Case 7: Verify proper validation of fundamental CPU operation (1) 21](#_Toc422940442)

[2.8 Test Case 8: Verify proper validation of fundamental CPU operation (2) 24](#_Toc422940443)

[2.9 Test Case 9: Verify proper validation of ETC SOH test sequence 27](#_Toc422940444)

[2.10 Test Case 10: Verify proper SPI communication error handling 30](#_Toc422940445)

# General Information

## Diagnostic Need

This diagnosis on MT22.3 uses a somewhat complex strategy to monitor the state of health (SOH) of the main microcontroller, ensures that the electronic throttle control (ETC) is operating properly. If any part is detected to be operating improperly the fuel injection and spark ignition will be disabled. This will prevent the possibility of an uncontrolled engine.

## Basic Requirements of the ETC SOH

| Requirement | Overview of Method |
| --- | --- |
| Detect main processor clock frequency <> 10% limit (intended to capture significant frequency errors) | First validate TCNT timer clock against the external 818 Hz. Then validate the oscillator divider interrupt period against TCNT timer and PIT timer, to validate the entire clock chain including bus clock and PLL module. |
| Check Real Time Interrupt Timing. Time for all major loops in the system to occur, must be the expected time +/- the major loop time. | Use sliding window technique. Every minor loop (i.e. 1 ms); store current time from PIT timer. Subtract from time stored 40 loops before (i.e. 40 ms). |
| Secondary means of detecting proper fundamental CPU operation | Challenge and response algorithm between main microcontroller and external SOH ASIC. |
| CPU order of execution / Loop sequencing | Main software writes loop ID to an array that the ETC SOH software interrogates. |
| Main SW must be able to enable / disable Fuel and ETC separate from SOH monitor. | Main software has control of the enables themselves. ETC SOH monitor can disable the drivers by setting the SPI bit FSE\_DISREQ to the SOH ASIC. |
| Realize above requirements with an independent hardware device | Leverage independent modules within the microcontroller, such as the timer module along with an independent, external clock source. Also leverages the challenge & response and timeout logic in the SOH ASIC. |
| Ensure time between ETC SOH interrupts is within the 10% tolerance. | During ETC SOH interrupts, read TCNT timer and compare to previous reading (relies on interrupt occurring). |
| Ensure time between ETC SOH interrupts is < COP timeout period (ensures interrupt is occurring). | ETC SOH interrupt handler responsible for servicing internal COP along with ensuring SOH ASIC timeout is not activated. Also set up the modulus counter that generates the interrupt itself. |
| Validate external clock frequency = Expected Freq (818 Hz) +/- 7.5%. | Use TCNT timer measure the 818 Hz independent clock input frequency from external hardware. |
| Ensure all ETC SOH tests are being run. | Each ETC SOH interrupt loop checks completion flag for all its tests, then uses internal watchdog hardware to test for each ETC SOH loop executed in order. |
| Validate clock bus generating fuel and spark outputs | ECT module is validated against TCNT timer and PIT timer by input capture of 58x. |

## Software Data Structure And Variables

typedef union Soh\_Fault\_Log\_Tag

{

uint16\_t Word; /\* ETC SOH fault code \*/

struct

{

bitfield8\_t SysClkFail : 1; /\* System clock frequency error \*/

bitfield8\_t SohIrqSrcFail : 1; /\* Interrupt source error \*/

bitfield8\_t SysTmrFail : 1; /\* System timer error \*/

bitfield8\_t SohSeqFail : 1; /\* Test sequence error \*/

bitfield8\_t RtiFreqFail : 1; /\* RTI frequency error \*/

bitfield8\_t LoopSeqFail : 1; /\* CPU loop sequence error \*/

bitfield8\_t CRCounterLow : 1; /\* SOH C&R counter value low \*/

bitfield8\_t ShutOffTimeExpire : 1; /\* Shutoff timer expired \*/

bitfield8\_t SPIFail : 1; /\* SPI error \*/

bitfield8\_t SPICommFail : 1; /\* SPI communication error \*/

bitfield8\_t CRDisarmed : 1; /\* SOH C&R disarmed \*/

bitfield8\_t CRTimeoutFail : 1; /\* SOH C&R timeout \*/

bitfield8\_t CRCounterFail : 1; /\* SOH C&R counter value zero \*/

bitfield8\_t : 3;

} Bits;

} Soh\_Fault\_Log\_T;

## Required Equipment

1. Flash Tool

2. Debugger

3. Vehicle Simulator

4. INCA and CAN card

5. Oscilloscope

6. Multimeter

## Mapping Table, SOH fault mapped to Diagnose Variable Interface

The following table maps the SOH flags to one of three diagnostic trouble variables.

|  |  |  |
| --- | --- | --- |
| Bit Position | Description  (0 – no error, 1 – error logged) | Diagnostic Report  (if shutoff timer expired) |
| 16 | System clock frequency error | SbCOND\_SOH\_MainCPU\_Flt |
| 15 | Interrupt source error | SbCOND\_SOH\_MainCPU\_Flt |
| 14 | System Timer error | SbCOND\_SOH\_MainCPU\_Flt |
| 13 | Test sequence error | SbCOND\_SOH\_MainCPU\_Flt |
| 12 | RTI frequency error | SbCOND\_SOH\_MainCPU\_Flt |
| 11 | CPU loop sequence error | SbCOND\_SOH\_MainCPU\_Flt |
| 10 | SOH C&R counter value low | SbCOND\_SOH\_CheckingCPU\_Flt |
| 9 | Shutoff timer expired | SbCOND\_SOH\_MainCPU\_Flt |
| 8 | SPI error | SbCOND\_SOH\_CommFlt |
| 7 | SPI communication error | SbCOND\_SOH\_CommFlt |
| 6 | SOH C&R disarmed | SbCOND\_SOH\_CheckingCPU\_Flt |
| 5 | SOH C&R timeout | SbCOND\_SOH\_CheckingCPU\_Flt |
| 4 | SOH C&R counter value zero | SbCOND\_SOH\_CheckingCPU\_Flt |
| 3:1 | Not used |  |

# Test Cases

## Test Case 1: Verify ETC SOH normal operation (1)

Purpose:

Verify the following requirements.

a) Scheduling of ETC SOH Tasks.

b) ETC SOH Initialization Sequence.

c) Odd/Even loop tests execution time measurement.

Procedures:

1. Follow the steps in the common procedures.

2. Watch the following variables by debuggers: Soh\_FaultLogNVM.

3. Reset the ECU and run the program. Monitor the following performance for no less than 15 minutes.

4. Verify the software is able to run without causing any reset.

5. Measure the ETC SOH interrupt period via the GPIO pin in MCU. Set the pin to high level when the SOH interrupt occurs, and toggle the GPIO pin when SOH interrupt is returned. Normally SOH interrupt period is 20ms.

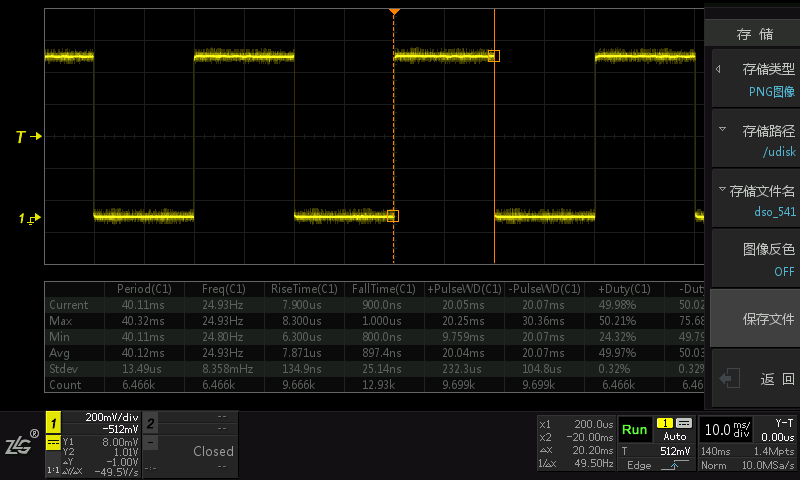
6. Measure the execution time for the odd and even loop tests.

7. Cycle the ignition key OFF-ON. Verify no ETC SOH fault code(s) is/are written to the EEPROM.

Test Results:

1. Is the software able to run without causing any reset? Yes  No

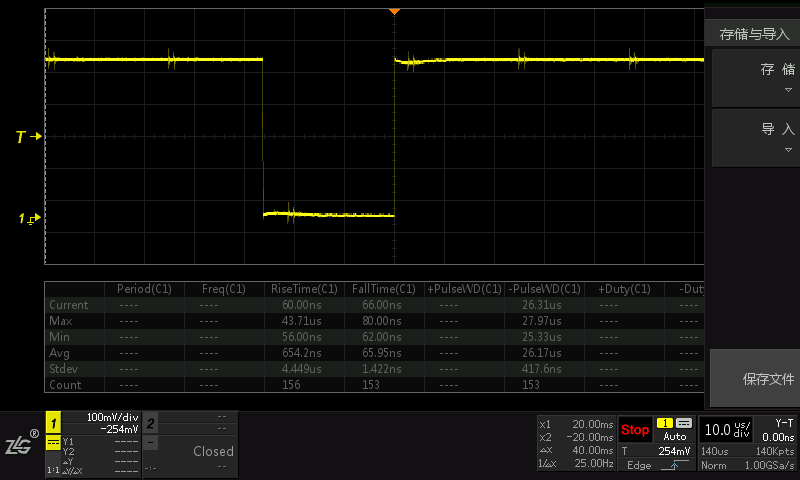
2. Is the measured ETC SOH interrupt period = **20 ms**? Yes  No



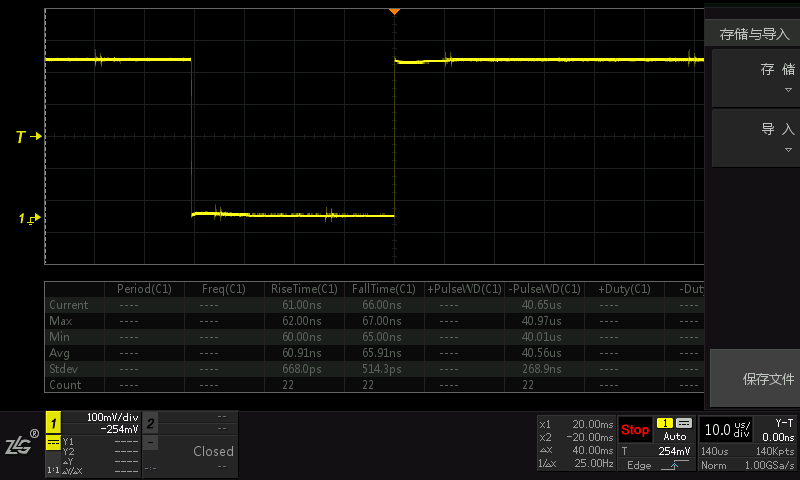
3. Odd loop average test execution time = 26.17 us

Even loop average test execution time = 40.56 us

Are the odd and even loops balanced? Yes  No



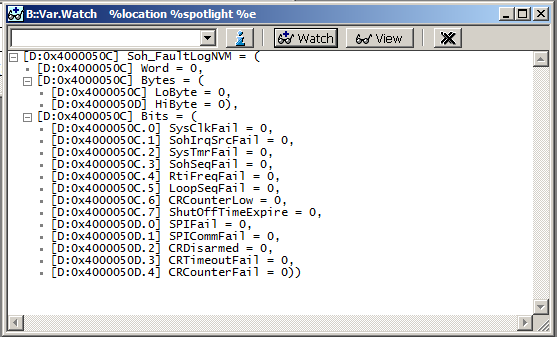
Odd Loop



Even Loop

Note: though the delta between odd and even execution time is about 14us, but the both execution time are not beyond 50us.

4. Does the fault code indicate “no error” ? Soh\_FaultLogNVM Yes  No



## Test Case 2: Verify ETC SOH normal operation (2)

Purpose:

Verify the following ETC SOH operations.

a) Odd loop tests pass.

b) Even loop tests pass.

c) MCU and VSEP IC challenge and response SPI message exchange period.

Procedures:

1. Follow the steps in the common procedures.

2. Watch and data log the following variables: Soh\_FaultLogNVM, Soh\_CnRStatus, Soh\_TestResult and Soh\_TestErr.

3. Reset the ECU and run the program. Monitor the following performance for no less than 15 minutes.

4. Record the value of Soh\_CnRStatus, Soh\_TestResult, Soh\_TestErr and Soh\_FaultLogNVM.

5. Verify fuel, spark and ETC output signals are not disabled.

6. Measure the average MCU and VSEP IC SPI challenge and response (C&R) exchange period, verify if the exchange period is less than 48ms.

Method: Use the toggle pin in MCU to measure the challenge and response (C&R) exchange period. Set the toggle pin to high level when challenge is received, and toggle the toggle pin when response is sent.

Test Results:

1. a) Is the **Soh\_CnRStatus.Bits.Respcount** counter value = 19? Yes  No

b) Is the **Soh\_CnRStatus.Bits.GEN\_Stat** bit set to true? Yes  No

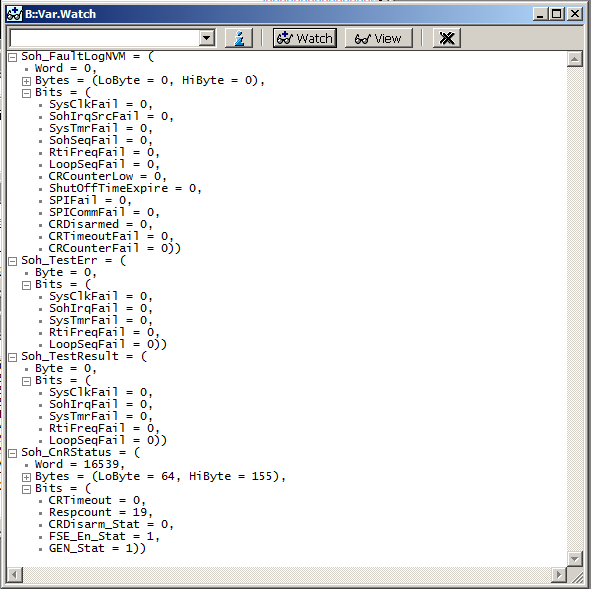
c) Is the **Soh\_CnRStatus.Bits.FSE\_En\_Stat** bit set to true? Yes  No

d) Are all other VSEP IC status bits = 0? Yes  No

e) Is the value of **Soh\_TestResult** = 0? Yes  No

f ) Is the value of **Soh\_TestErr** = 0? Yes  No

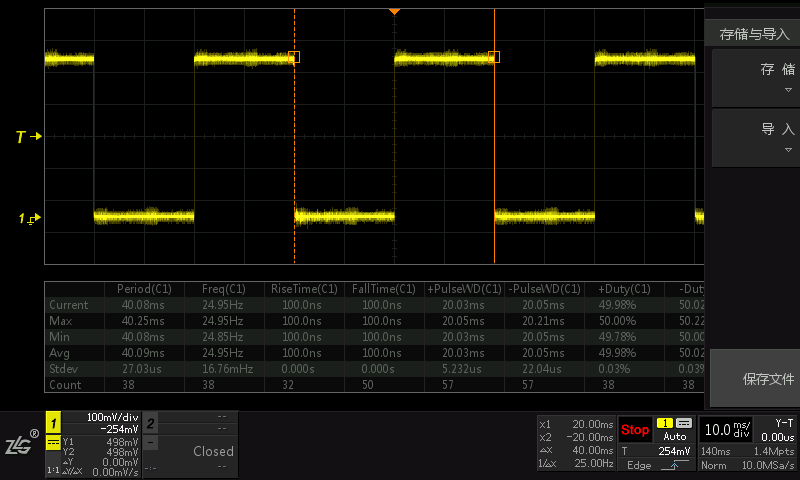
g) Is the value of **Soh\_FaultLogNVM** = 0? Yes  No



2. Are fuel, spark and ETC output signals present? Yes  No

3. a) Measured average MCU and VSEP IC C&R exchange period = 20.03 ms

b) Is the measured C&R exchange period less than the timeout period (48ms)? Yes  No



## Test Case 3: Verify proper validation of system clock

Purpose:

Verify the following requirements:

a) Detect main processor clock frequency <> 10% limit.

b) Validate independent clock frequency = 818Hz +/- 7.5%.

c) ETC SOH Error Handling.

d) Disabling from the ETC SOH Interrupt Handler.

Procedures:

1. Set the independent reference clock to 818 Hz, change the system clock frequency (fSYS) by adjusting the value of PLL to set the SYSCLK to 40MHz.

2. Verify the following:

a) Detected the failure (**Soh\_TestResult.Bits.SysClkFail** == TRUE)

b) Verify **Soh\_EvenErrCnt** and **Soh\_ShutOffUpTimer** vary as expected.

c) Verify that the spark, fuel and ETC outputs are disabled.

d) Verify VSEP IC status bit **Soh\_CnRStatus.Bits.FSE\_En\_Stat** is cleared when the failure condition is met.

e) Verify the fault code **Soh\_FaultLogNVM.Bits.** **SysClkFail** is set when the failure condition is met.

f) Verify the fault code **Soh\_FaultLogNVM.Bits.** **ShutOffTimeExpire** is set when the error timer expires.

g) Verify the diagnose interface variable **SbCOND\_SOH\_MainCPU\_Flt** are set

3. Vary the frequency of the external 818Hz reference clock (fREFCLK) by define the frequency **REFCLK\_FREQ\_HZ** to 1000Hz in the function **ValidateSysClkFreq(**…**)**. Then repeat the above steps.

Test Results in the above two settings 1&3:

1) Is the bit **Soh\_TestResult.Bits.SysClkFail** ? Yes  No

2) Do **Soh\_EvenErrCnt** and **Soh\_ShutOffUpTimer** vary as expected? Yes  No

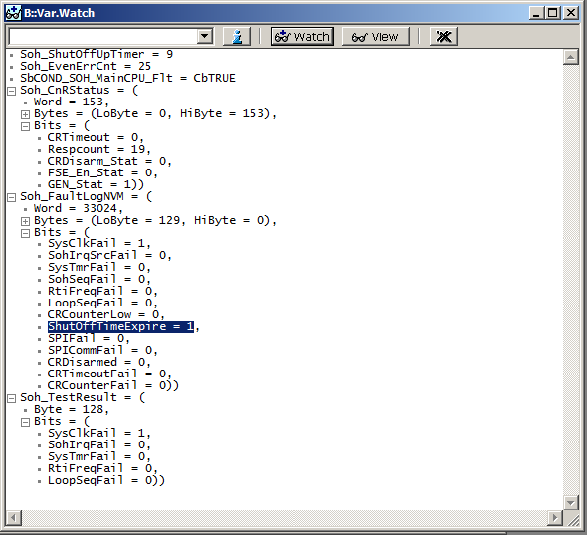
3) Are spark, fuel and ETC disabled? Yes  No

4) Is the VSEP IC status bit **Soh\_CnRStatus.FSE\_En\_Stat** cleared to false? Yes  No

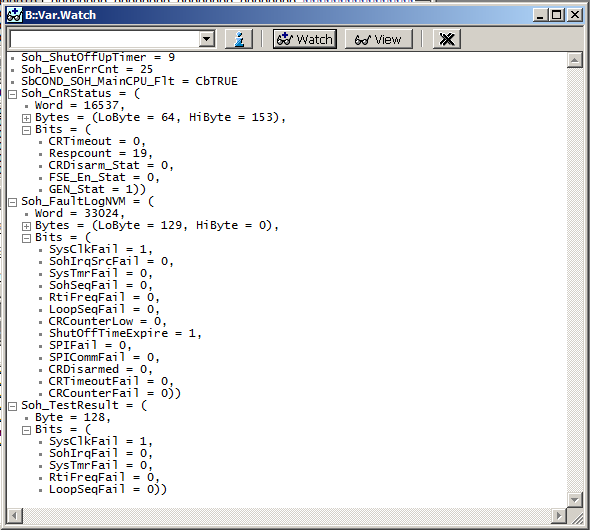
5) Is the bit **Soh\_FaultLogNVM.Bits.** **SysClkFail** set? Yes  No

6) Is the bit **Soh\_FaultLogNVM.Bits.** **ShutOffTimeExpire** set? Yes  No

7) Is the variable **SbCOND\_SOH\_MainCPU\_Flt** set? Yes  No



Test 1, change the PLL value to set SYSCLK to 40MHz



Test 3, change the **REFCLK\_FREQ\_HZ** value to 1000Hz

## Test Case 4: Verify proper validation of ETC SOH interrupt source

Purpose:

Verify the following requirements.

a) Ensure time between ETC SOH interrupts is within the 10% tolerance.

c) ETC SOH Error Handling.

d) Disabling from the ETC SOH Interrupt Handler.

Procedures:

1. Set real SOH IRQ period to 25ms(over than +10%) by writing test code.

a) Verify **Soh\_EvenErrCnt** and **Soh\_ShutOffUpTimer** vary as expected.

b) Verify that the spark, fuel and ETC outputs are disabled.

c) Verify VSEP IC status bit **Soh\_CnRStatus.FSE\_En\_Stat** is cleared when the failure condition is met.

d) Verify the fault code **Soh\_FaultLogNVM.Bits.SohIrqSrcFail** is set when the failure condition is met.

e) Verify the fault code **Soh\_FaultLogNVM.Bits.** **ShutOffTimeExpire** is set when the error timer expires.

f) Verify the diagnose interface variable **SbCOND\_SOH\_MainCPU\_Flt** are set

Test Results:

1) Do **Soh\_EvenErrCnt** and **Soh\_ShutOffUpTimer** vary as expected? Yes  No

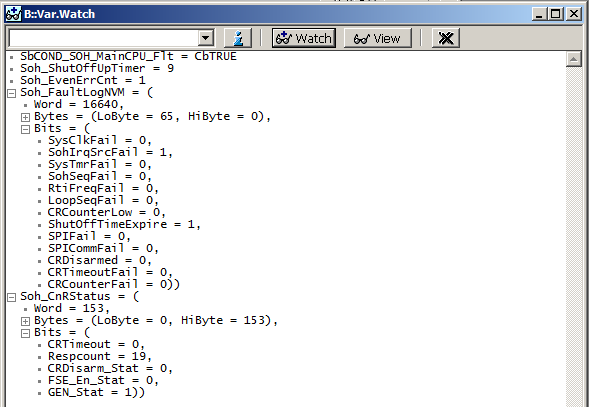
2) Are spark, fuel and ETC disabled? Yes  No

3) Is the VSEP IC status bit **Soh\_CnRStatus.FSE\_En\_Stat** cleared to false? Yes  No

4) Is the bit **Soh\_FaultLogNVM.Bits.** **SohIrqSrcFail** set? Yes  No

5) Is the bit **Soh\_FaultLogNVM.Bits.** **ShutOffTimeExpire** set? Yes  No

6) Is the variable **SbCOND\_SOH\_MainCPU\_Flt** set? Yes  No



## Test Case 5: Verify proper validation of Real Time Interrupt (RTI) frequency

Purpose:

Verify the following requirements.

a) Time for all major loops to occur must be the expected time +/- one major loop time.

b) ETC SOH Error Handling.

c) Disabling from the ETC SOH Interrupt Handler.

Procedures:

1. Watch and data log the following variables: **Soh\_RtiLoopTime**, **Soh\_LstRtiCirBufIdx**, **Soh\_RtiCirBufIdx**.

2. Set real RTI IRQ period to 1.15ms(over than +10%) by writing test code.

3. Verify the following:

a) Verify that the spark, fuel and ETC outputs are disabled.

b) Verify VSEP IC status bit **Soh\_CnRStatus.FSE\_En\_Stat** is cleared when the failure condition is met.

c) Verify the fault code **Soh\_FaultLogNVM.Bits.RtiFreqFail** is set when the failure condition is met.

d) Verify the fault code **Soh\_FaultLogNVM.Bits.ShutOffTimeExpire** is set when the error timer expires.

e) Verify the diagnose interface variable **SbCOND\_SOH\_MainCPU\_Flt** is set.

4. Corrupt the value of **Soh\_RtiCirBufIdx** bydebugger. Repeat steps 3a-3e.

Test Results:

1) Are spark, fuel and ETC disabled? Yes  No

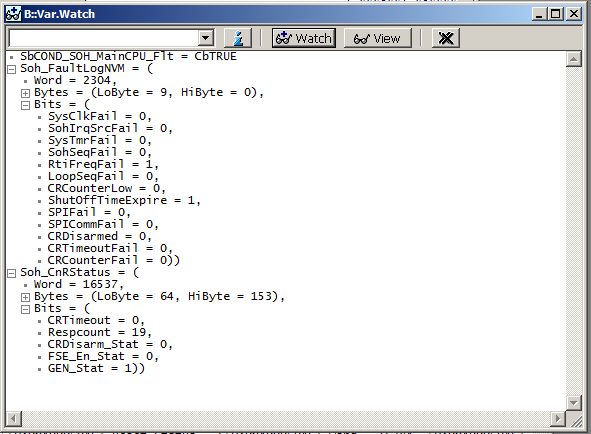
2) Is the VSEP IC status bit **Soh\_CnRStatus.FSE\_En\_Stat** cleared to false? Yes  No

3) Is the bit **Soh\_FaultLogNVM.Bits.RtiFreqFail** set? Yes  No

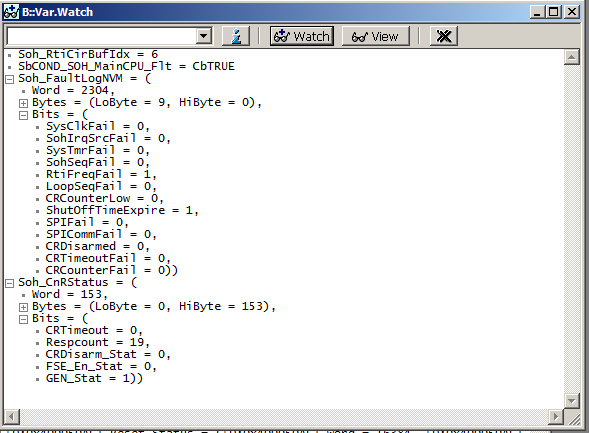
4) Is the bit **Soh\_FaultLogNVM.Bits.** **ShutOffTimeExpire** set? Yes  No

5) Is the variable **SbCOND\_SOH\_MainCPU\_Flt** set? Yes  No

Figures of 2 way test:



Test 2 Set real RTI IRQ period to 1.15ms



Test4 Corrupt the value of **Soh\_RtiCirBufIdx**

## Test Case 6: Verify proper validation of CPU loop sequence

Purpose:

Verify the following requirements.

a) CPU order of execution / Loop sequencing.

b) ETC SOH Error Handling.

c) Disabling from the ETC SOH Interrupt Handler.

Procedures:

1. Watch and data log the following variables: **Soh\_IdTagExpect, Soh\_LoopSeq, Soh\_IrqLoopSeqIdx, Soh\_SchdLoopSeqIdx.**

2. Reset the ECU and run the program. Ensure normal operation without any error.

3. Corrupt the values in the array **Soh\_LoopSeq** while the program is running.

4. Verify the following:

a) Check the array entries in **Soh\_LoopSeq**, verify the fault is detected immediately when an entry is invalid.

b) Verify **Soh\_OddErrCnt** and **Soh\_ShutOffUpTimer** vary as expected.

c) Verify that the spark, fuel and ETC outputs are disabled.

d) Verify VSEP IC status bit **Soh\_CnRStatus.FSE\_En\_Stat** is cleared when the failure condition is met.

e) Verify the fault code **Soh\_FaultLogNVM.Bits.LoopSeqFail** is set when the failure condition is met,

f ) Verify the fault code **Soh\_FaultLogNVM.Bits.ShutOffTimeExpire** is set when the error timer expires,

g) Verify the diagnose interface variable **SbCOND\_SOH\_MainCPU\_Flt** is set.

5. Corrupt the variable **Soh\_IdTagExpect**. Repeat step 4.

6. Corrupt the value of **Soh\_IrqLoopSeqIdx**. Repeat step 4.

7. Corrupt the value of **Soh\_SchdLoopSeqIdx**. Repeat step 4.

Test Results:

a) Is the fault detected immediately when an entry is invalid? Yes  No

b) Do **Soh\_OddErrCnt** and **Soh\_ShutOffUpTimer** vary as expected? Yes  No

c) Are spark, fuel and ETC disabled? Yes  No

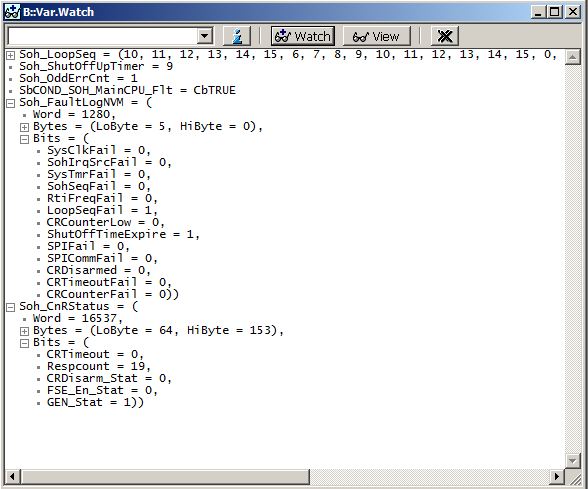
d) Is the VSEP IC status bit **Soh\_CnRStatus.FSE\_En\_Stat** cleared to false? Yes  No

e) Is the fault code **Soh\_FaultLogNVM.Bits.LoopSeqFail** is set? Yes  No

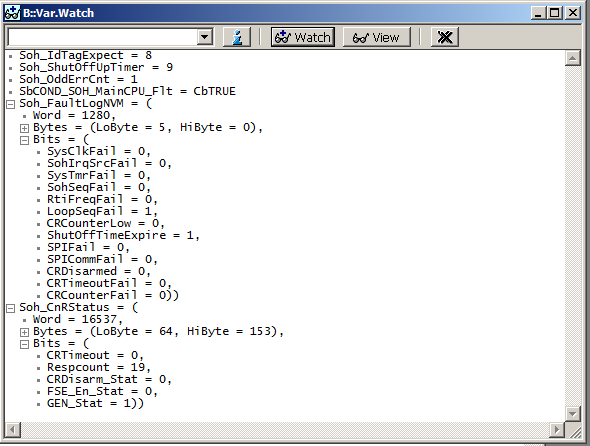
f ) Is the fault code **Soh\_FaultLogNVM.Bits.ShutOffTimeExpire** is set? Yes  No

g) Is the variable **SbCOND\_SOH\_MainCPU\_Flt** set? Yes  No

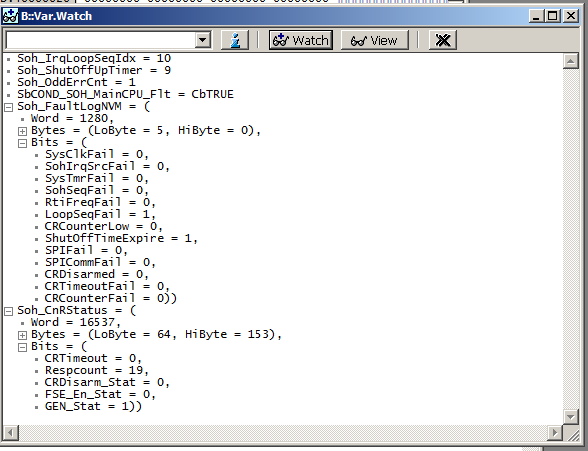
Figures of 4 different tests:



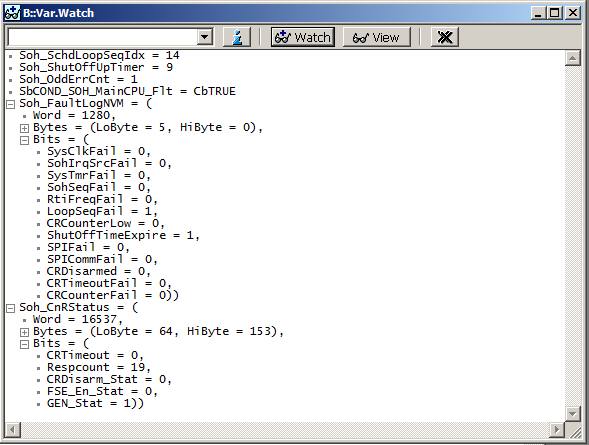
Corrupt the values in the array **Soh\_LoopSeq**



Corrupt the variable **Soh\_IdTagExpect**.



Corrupt the value of **Soh\_IrqLoopSeqIdx**.



Corrupt the value of **Soh\_SchdLoopSeqIdx**

## Test Case 7: Verify proper validation of fundamental CPU operation (1)

**Notice: This test should write test code to control the response error times.**

Purpose:

Verify the following requirements.

a) Secondary means of detecting proper fundamental CPU operation.

b) ETC SOH Error Handling.

c) Disabling from the ETC SOH Interrupt Handler.

Procedures:

1. Follow the steps in the common procedures.

2. Watch and data log the following variables: **Soh\_CnRValue**, **Soh\_CnRStatus**.

3. Reset the ECU and run the program. Ensure normal operation without any error.

4. Corrupt the computed response values while the program is running. Verify the following:

a) The VSEP IC **RESPCOUNT** counter value is decremented by four for each C&R test with error and incremented by one for each C&R test without error. The maximum **RESPCOUNT** counter value is 19,

b) The same challenge value is received again when the previous C&R test is in error,

c) If RESPCOUNT counter value falls below **KSOHCRTH**, the fault code **Soh\_FaultLogNVM.Bits.CRCounterLow** is set.

d) Verify that the spark, fuel and ETC outputs are disabled.

e) Verify VSEP IC status bit **Soh\_CnRStatus.FSE\_En\_Stat** is cleared when the failure condition is met.

f ) Verify the diagnose interface variable **SbCOND\_SOH\_CheckingCPU\_Flt** is set.

5. Change value of calibration parameter **KSOHCRTH** to 0. Reset the ECU and run the program. Ensure normal operation without any error. Corrupt the computed response values while the program is running.

a) Verify if **RESPCOUNT** counter value equals zero, and the fault code **Soh\_FaultLogNVM.Bits.CRCounterFail** is set,

b) Verify the outputs is cut-off when **RESPCOUNT** counter value equals zero,

c) Verify VSEP IC status bit **Soh\_CnRStatus.FSE\_En\_Stat** is cleared when the failure condition is met.

d) Verify the diagnose interface variable **SbCOND\_SOH\_CheckingCPU\_Flt** is set.

Test Results:

4.a) Is RESPCOUNT decremented by four for each C&R test with error? Yes  No

Is RESPCOUNT incremented by one for each C&R test without error? Yes  No

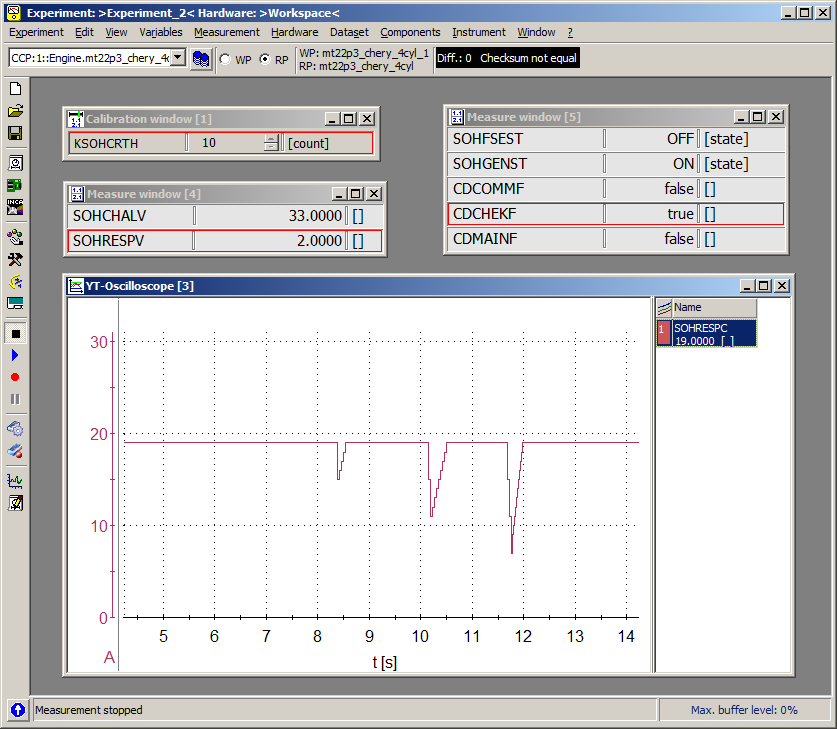
b) Is the same challenge value received when the previous C&R test is in error? Yes  No

c) Is fault code **Soh\_FaultLogNVM.Bits. CRCounterLow** set to true? Yes  No

d) Are spark, fuel and ETC disabled? Yes  No

e) Is the VSEP IC status bit Soh\_CnRStatus.FSE\_En\_Stat cleared to false? Yes  No

g) Is the variable **SbCOND\_SOH\_CheckingCPU\_Flt** set? Yes  No



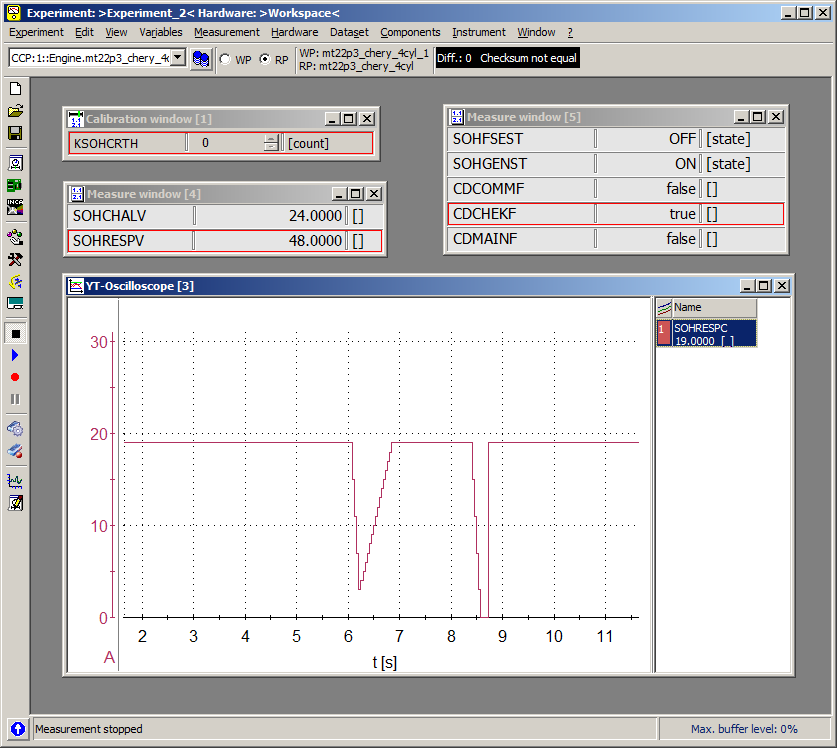
Data logger File: 

5. a) Is the **Soh\_FaultLogNVM.Bits.CRCounterFail** bit set to true? Yes  No

b) Is the outputs cutoff? Yes  No

c) Is the VSEP IC status bit Soh\_CnRStatus.FSE\_En\_Stat cleared to false? Yes  No

g) Is the variable **SbCOND\_SOH\_CheckingCPU\_Flt** set? Yes  No



Data logger File: 

## Test Case 8: Verify proper validation of fundamental CPU operation (2)

Notice: This test should write test code to control the sending of the response value to the VSEP IC, and setting the value of VSEP IC status.

Purpose:

Verify the following requirements.

a) Secondary means of detecting proper fundamental CPU operation.

b) ETC SOH Error Handling.

c) Disabling from the ETC SOH Interrupt Handler.

Procedures:

1. Watch and data log the following variables: **Soh\_CnRValue**, **Soh\_CnRStatus**,

2. Reset the ECU and run the program. Ensure normal operation without any error.

3. Stop the C&R response from sending by test code.

a) Verify the fault code Soh\_FaultLogNVM.Bits. CRTimeoutFail is set.

b) Verify that the spark, fuel and ETC outputs are disabled due to C&R timeout.

c) Verify VSEP IC status bit **Soh\_CnRStatus.FSE\_En\_Stat** is cleared when the failure condition is met.

d) Verify the diagnose interface variable **SbCOND\_SOH\_CheckingCPU\_Flt** is set.

4. Recovery the C&R response sending. Set the VSEP IC status bit **Soh\_CnRStatus.Bits.CRDisarm\_Stat** to 1 by test code. Verify the fault code **Soh\_FaultLogNVM.Bits.** **CRDisarmed** is set.

a) Verify that the spark, fuel and ETC outputs are disabled.

b) Verify VSEP IC status bit **Soh\_CnRStatus.FSE\_En\_Stat** is cleared when the failure condition is met.

c) Verify the diagnose interface variable **SbCOND\_SOH\_CheckingCPU\_Flt** is set.

Test Results:

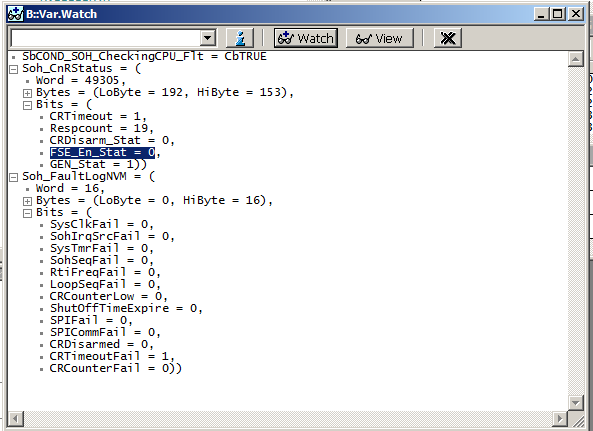
3

a) Is the **Soh\_FaultLogNVM.Bits.** **CRTimeoutFail** bit set to true? Yes  No

b) Are spark, fuel and ETC disabled due to C&R timeout? Yes  No

c) Is the VSEP IC status bit **Soh\_CnRStatus.FSE\_En\_Stat** cleared to false? Yes  No

d) Is the variable **SbCOND\_SOH\_CheckingCPU\_Flt** set? Yes  No



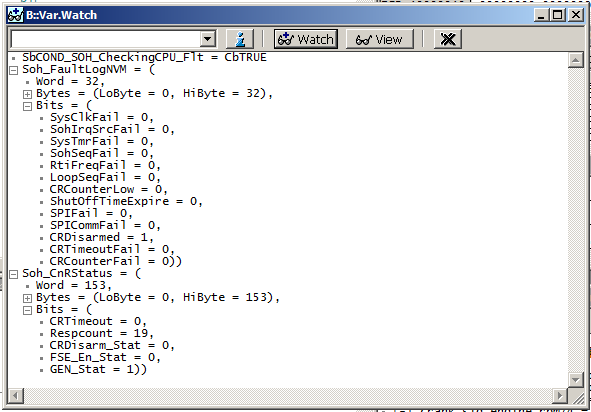
4

a) Is the **Soh\_FaultLogNVM.Bits.CRDisarmed** bit set to true? Yes  No

b) Are spark, fuel and ETC disabled? Yes  No

c) Is the VSEP IC status bit **Soh\_CnRStatus.FSE\_En\_Stat** cleared to false? Yes  No

e) Is the variable **SbCOND\_SOH\_CheckingCPU\_Flt** set? Yes  No



## Test Case 9: Verify proper validation of ETC SOH test sequence

Notice: This test need to write test code to control the execution of all SOH test sequence. Use a global parameter with individual bits to control which odd/even tests to run. Set break before watchdog reset to confirm the reset event.

Purpose:

Verify the following requirements.

a) Ensure all ETC SOH tests are being run.

b) Logging of Fault Codes to EEPROM.

c) Disabling from the ETC SOH Interrupt Handler.

Procedures:

1. Watch the following variables: **Soh\_LoopCnt**, **Soh\_TestComp**

2. Reset the ECU and run the program. Ensure normal operation without any error.

3. Disable the execution of validating system clock frequency in the even loop. Verify the following:

a) SOH test complete flag **Soh\_TestComp** records a missing test.

b) The program will reset due to watchdog access error.

4. Repeat steps 3a – 3b for every test in the odd/even loop one at a time.

a) Disable the execution of validating SOH interrupt frequency in the even loop.

b) Disable the execution of validating CPU operation in the odd loop.

c) Disable the execution of validating RTI frequency in the even loop.

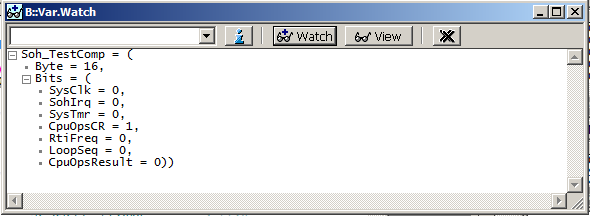
d) Disable the execution of validating CPU loop sequence in the odd loop.

e) Disable the execution of validating CPU operation error handling in the even loop.

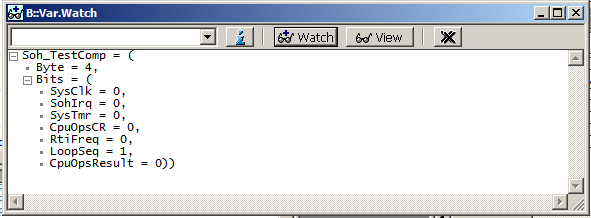
Test Results:

a) Does **Soh\_TestComp** record a missing test in all test cases? Yes  No

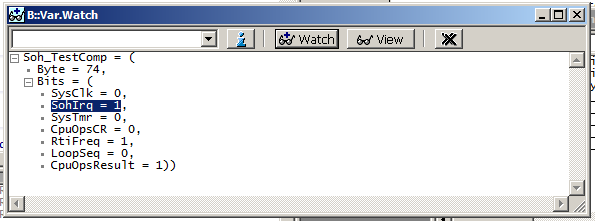
b) Is the program reset due to watchdog access error in all test cases? Yes  No



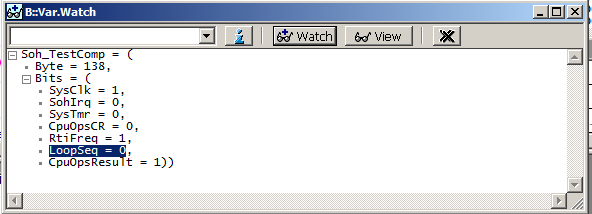
Odd Loop, Disable ValidateCpuLoopSeq()



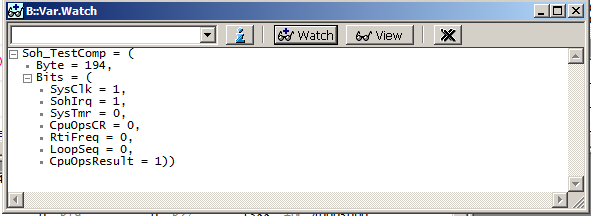
Odd Loop, Disable ValidateCpuOperation ()



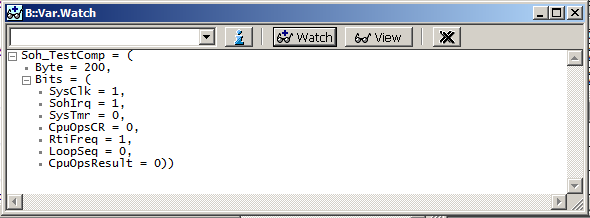
Even Loop, Disable ValidateSysClkFreq()



Even Loop, Disable ValidateEtcSohIrqFreq()



Even Loop, Disable ValidateRtiFreq()



Even Loop, Disable CpuOperatingErrHandler()

## Test Case 10: Verify proper SPI communication error handling

Notice: SOH Logic Not Support **Soh\_FaultLog.Bits.SPIFail** In current software; A software breakpoint should be before system goes to reset.

Purpose:

Verify the following requirements and operations.

a) SPI error handling.

b) Logging of SPI related fault code.

c) Logging of Fault Codes to EEPROM.

Procedures:

1. Watch and data log the following variables: Soh\_FaultLogNVM, Soh\_TestResult,

2. Reset the ECU and run the program. Ensure normal operation without any error.

3. Enable the sending of SPI message to VSEP IC. Corrupt SPI data line to short circuit.

4. a) Verify the fault code **Soh\_FaultLogNVM.Bits.SPICommFail** is set.

b) The program will reset due to watchdog access error.

Test Results:

4. a) Is the **Soh\_FaultLogNVM.Bits.SPICommFail** bit set to true? Yes  No

b) Is the program reset due to watchdog access error? Yes  No

