

Managing Failure Detections and Using Required Components to Meet ISO7637 pulse 1 on MC33903/4/5 Common Mode Choke Implementation

1 Overview

This document describes the implementation of a recommended electrical component at the device's CAN terminal as well as the recommended software management of the CAN interface for the MC33903/4/5 family in case of failure detection.

2 Scope

This document is divided into 3 sections:

1. Application schematic in order to increase device performance related to ISO pulse applied on CAN bus.
2. Recommended software practice in order to detect, manage and ensure proper recovery in case of ECU local failures occurring on TxD or RxD terminals, and in case of CAN over temperature.
3. Recommended software practice to properly detect and report flags in the CAN register, to ensure reliable failure report to main software.

Contents

1 Overview	1
2 Scope	1
3 CAN Interface Management 2	2
3.1 CAN Terminal Schematic	2
3.2 TxD, RxD and Over-Temp Failure	3
3.3 Other CAN Faults	6

3 CAN Interface Management

3.1 CAN Terminal Schematic

In order to improve the device performance regarding ISO7637 pulse 1 applied to the ECU CAN connector (CANH-ecu and CANL-ecu in figure below) when the device is implemented with a common mode choke, a dedicated protection element (D1) should be connected between the common mode choke and the device CANH and CANL terminal as shown on the schematic below (See Figure 1).

This component (D1) should be connected per the recommendation below, and only if a common mode choke (L1) is used. In cases where this common mode choke is not connected, protection element (D1) is not necessary. The proposed protection element is a dual zener diode, specifically developed for CAN applications, and offering low parasitic capacitance.

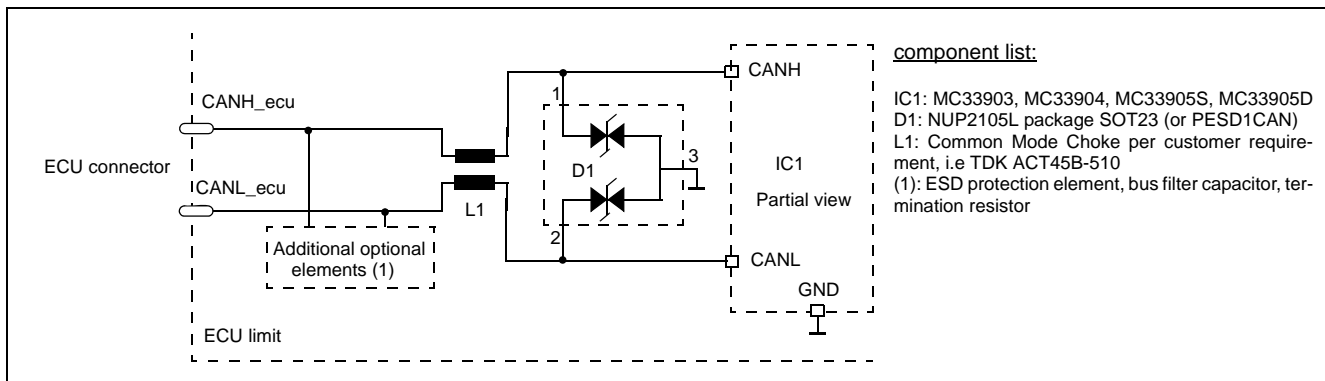


Figure 1. CAN Terminal Schematic

3.2 TxD, RxD and Over-Temp Failure

The device detects and manages several types of failures that occur either on the CAN bus or locally on the ECU. Some failures are reported as information only (flag), and some will in addition to being reported as flag disable the CAN driver in order to protect the device or prevent a general system failure. This is the case for the following:

- Failures at TxD or RxD terminals
- CAN driver over temperature

TxDominant failure means TxD internal ECU line is in permanent dominant state. Once this failure is detected, the CAN driver is disabled in order to avoid the CAN bus to be in permanent dominant state, preventing any further communication on the network.

RxD failure means that RxD internal ECU line is shorted to gnd or 5V. Once this failure is detected, the CAN driver is also disabled, since the MCU could interpret RxD failure as bus idle state.

CAN driver over temperature will also disable the driver in order to prevent device from possible destruction.

3.2.1 Information Related to Flag Reporting and Consequences

3.2.1.1 Flag reporting

When a failure occurs (i.e TxD), the corresponding flag is set in the CAN register. The flag is latched, until the CAN register is read (i.e SPI 0xE100). After read operation, the flag becomes a “real time” flag, and is then reset as soon as the failure recovers.

3.2.1.2 Failure Consequence

When the CAN is TxRx state (Transmit: Receive state) if TxD, RxD or CAN over temperature failures are detected, the bus driver is automatically disabled and the CAN is set to Rx only (Receive only) state. The CAN driver can be re-enable only via the SPI command with 3 conditions:

1. Error has recovered
2. Flag is cleared by SPI command
3. CAN interface is set to TxRx state by SPI command

The paragraph below describes the recommended practice for the software that controls the device to detect such events, report them to the MAIN software and how to reconfigure the device once failure has recover.

Two cases can be distinguished, depending upon usage or not of interruption (INT).

3.2.1.3 Case 1: INT not used

The main steps consists of the following:

1. Initialization routine
It should contain at least the two configuration commands listed below to ensure CAN interrupt is disabled and CAN interface is set to Transmit/Receive state.
 - INT configuration (disable)
 - CAN interface configuration (Transmit/Receive state)
2. MAIN software routine
It is assumed that the main software runs a periodic task that consists of the following:
 - Check CAN interface state
 - Check if any CAN flag is set
 - If flag is set or CAN interface is not in expected state, reset flags and set CAN in appropriate state.
 - Then return to MAIN S/W and report CAN state information.

CAN Interface Management

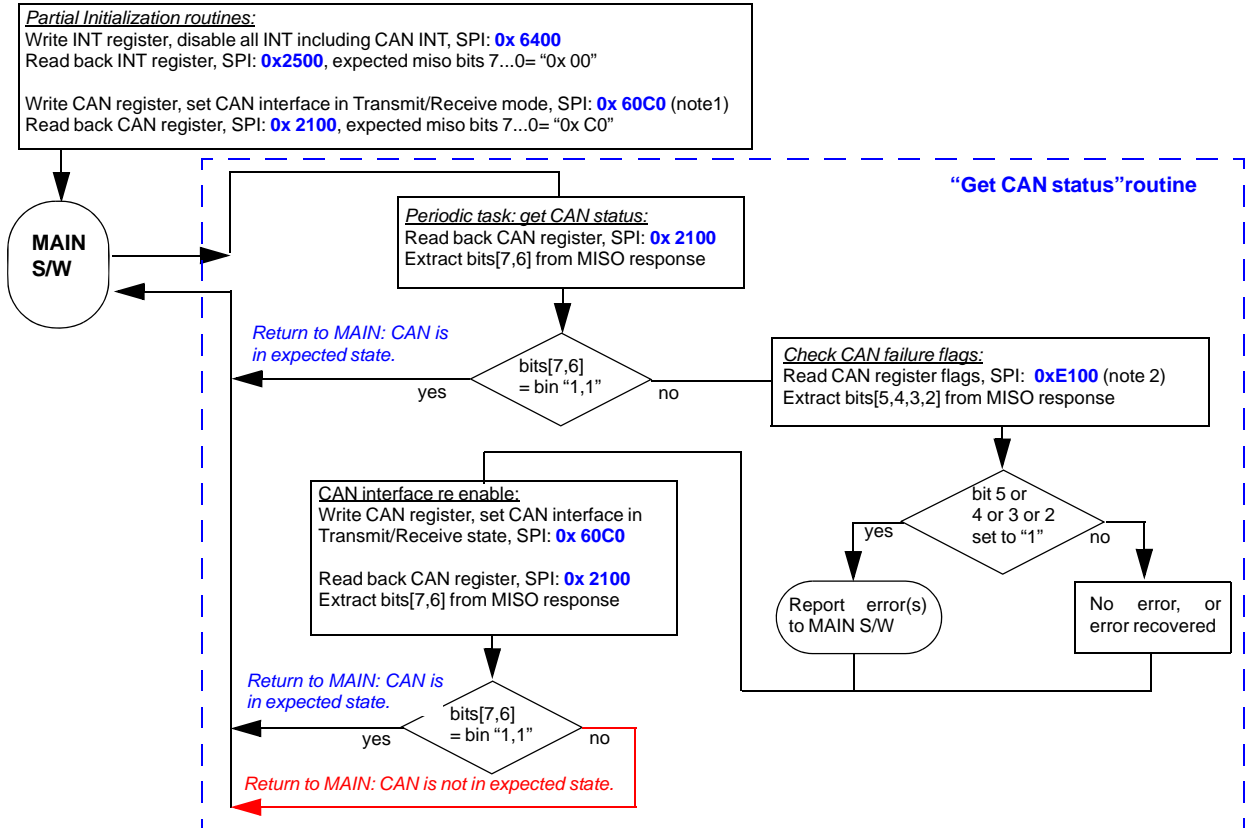


Figure 2. Flow Chart without use of INT

Notes:

- Bit 0 to 5 are set low in example. They could be set to different value, depending upon user choice (bus slew rate, CAN wake up option etc.).
- If flag is set, the SPI 0xE100 command will result in one of the following:
 - If failure has recovered: clears flag
 - If failure has not recovered: switch the flag operation from "latch" to "real time reporting" (flag will be reset as soon as failure will recover, without the need for SPI command).

3.2.1.4 Case 2: INT is used

The main steps consists of the following:

- Initialization routine
It should contain at least the two configurations listed below to ensure CAN interrupt is enabled, and CAN interface is set in Transmit/Receive mode.
 - INT configuration (enable)
 - CAN interface configuration
- MAIN software routine
It is assumed that when INT pulse occurs, the main software runs a task that consists of:
 - Check INT source
 - Check CAN interface state
 - Check if CAN flag is set
 If flag is set or CAN interface is not in expected state, reset flags and set CAN in appropriate state.
 - Then return to MAIN S/W and report CAN state information.
 - The MAIN S/W must periodically attempt to re enable the CAN in Transmit/Receive mode.

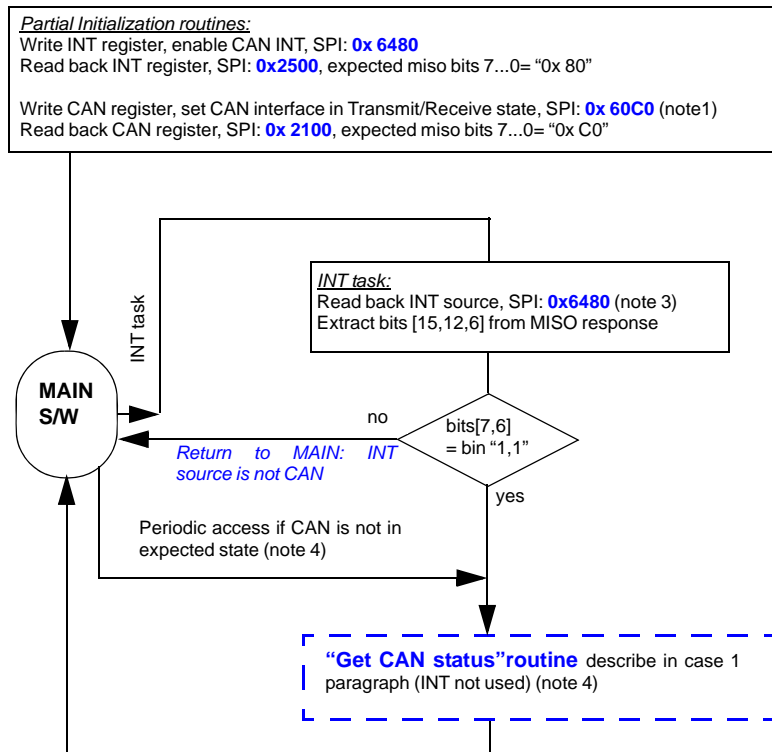


Figure 3. Flow Chart using INT

Notes:

1. Bit 0 to 5 are set low in example. They could be set to different value, depending upon user choice (bus slew rate, CAN wake up option etc.).
2. If flag is set, the SPI 0xE100 command will result in one of the following:
 - If failure has recovered: clears flag
 - If failure has not recovered: switch the flag operation from "latch" to "real time reporting" (flag will be reset as soon as failure recovers, without the need for SPI command).
3. Write command to get 16 bits standard and extended device status. Bit 15, 12 and 6 are INT status, CAN-general flag and CAN local flag.
 - Usage of any write command (i.e W/D refresh 0x5A00) is also possible.
4. The "Get CAN status" routine should be either:
 - One time execution if failure has recovered before INT task.
 - Periodic execution if CAN is not in expected state.

3.3 Other CAN Faults

The device detects and manages several types of failures occurring either on the CAN bus or locally on the ECU. Some failures are only reported as information (flag), and some will disable the CAN driver in order to protect the device or prevent a general system failure.

This section describes recommended practice to manage the CAN error flag that does not disable or modify the CAN driver state. These flags are only for information and for system diagnostic purpose.

The flags are listed in the table below (extract from device data sheet, CAN register). 0xE100 and 0xE180 SPI commands are used, and device provides 16 bits status on MISO. Table below describes bits 7-0:

SPI command MOSI 16 bits	MISO response, bits 7-0							
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
0xE100	CAN wake-up	-	CAN Over-temp	RxD low	RxD high	TxD dom	Bus Dom clamp	CAN Over-curre nt
0xE180	CAN_UF	CAN_F	CANL to V _{BAT}	CANL to V _{DD}	CANL to GND	CANH to V _{BAT}	CANH to V _{DD}	CANH to GND

Figure 4. CAN Register Flags

Bit 7 indicates that the error is not yet fully identified by the logic circuitry yet or that it is unstable.

Bit 6 indicates that the error is identified by the logic circuitry. The same error has been detected over a series of 5 consecutive TxD transitions.

Bits 5 to 0 report the error detected by the device. These flags do not change the CAN status. CAN will remain in TxRx state independently of the flag being set or not, and the communication is not affected.

However, some of these flags could be set during exposure to EMC in the frequency range below 10MHz. It is recommended to perform multiple sampling of these flags over a period of time before reporting them to the MAIN software.

As an example, the flag sampling can be performed as follows:

- Upon execution of the 'Get CAN status routine' described in above paragraph above.
- Upon entering in Low Power mode
- After device wake up.
- At W/D refresh of the device
- At a predetermined time period independent of above
- Any combination of the above combinations.

If a flag is then consistently set over a number of samplings, it can then be reported to the MAIN software as a consistent failure.

How to Reach Us:

Home Page:

www.freescale.com

Web Support:

<http://www.freescale.com/support>

USA/Europe or Locations Not Listed:

Freescale Semiconductor, Inc.
Technical Information Center, EL516
2100 East Elliot Road
Tempe, Arizona 85284
1-800-521-6274 or +1-480-768-2130
www.freescale.com/support

Europe, Middle East, and Africa:

Freescale Halbleiter Deutschland GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
www.freescale.com/support

Japan:

Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku,
Tokyo 153-0064
Japan
0120 191014 or +81 3 5437 9125
support.japan@freescale.com

Asia/Pacific:

Freescale Semiconductor China Ltd.
Exchange Building 23F
No. 118 Jianguo Road
Chaoyang District
Beijing 100022
China
+86 10 5879 8000
support.asia@freescale.com

For Literature Requests Only:

Freescale Semiconductor Literature Distribution Center
P.O. Box 5405
Denver, Colorado 80217
1-800-441-2447 or +1-303-675-2140
Fax: +1-303-675-2150
LDCForFreescaleSemiconductor@hibbertgroup.com

Information in this document is provided solely to enable system and software implementers to use Freescale Semiconductor products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits or integrated circuits based on the information in this document.

Freescale Semiconductor reserves the right to make changes without further notice to any products herein. Freescale Semiconductor makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale Semiconductor assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. "Typical" parameters that may be provided in Freescale Semiconductor data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including "Typicals", must be validated for each customer application by customer's technical experts. Freescale Semiconductor does not convey any license under its patent rights nor the rights of others. Freescale Semiconductor products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Freescale Semiconductor product could create a situation where personal injury or death may occur. Should Buyer purchase or use Freescale Semiconductor products for any such unintended or unauthorized application, Buyer shall indemnify and hold Freescale Semiconductor and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Freescale Semiconductor was negligent regarding the design or manufacture of the part.



Freescale™ and the Freescale logo are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners.

© Freescale Semiconductor, Inc. 2010. All rights reserved.