Dual-modular Redundancy for CAN networks

Uwe Koppe, Johann Tiderko (MicroControl)

The CiA 307 document "Framework for maritime electronics" /1/ was the initial approach to use two CAN cables for mission-critical application in the year 2004. However, this standard was only applicable for classical CANopen and it has been withdrawn meanwhile. Based on the original ideas of the CiA 307 document, a new standard has been developed, called dual-modular redundancy (DMR). The DMR is suitable for CAN as well as CAN FD networks and can be used to implement CAN devices and CAN networks with the need of safety-critical, mission-critical and high-availability communication. The DMR has been designed to be independent from an application layer like CANopen, CANopen FD, J1939 or any other system specific application.

As described in /2/ an integrated system designed for high availability (HA) shall be arranged with sufficient redundancy and/or segregation to prevent loss of essential functions or multiple main functions upon a single failure. In addition, any network integrating control and/or monitoring systems shall be single point of failure-tolerant or alternatively designed so that the effect of a single failure does not exceed this principle. This implies that the network with its necessary components and cables shall be designed with adequate redundancy.

As such, a CAN network used in HA applications (e.g. avionics, maritime, etc.) must have at least two independent CAN interfaces, each driving physically separated CAN lines.

A single failure within this redundant CAN network is defined by one of the following incidents:

- Single interrupt of a CAN line
- Single failure of CAN transceiver
- Single failure of CAN controller (including bus-off condition)

A failure inside the application (e.g. a heartbeat event inside the CANopen application) is not regarded as a failure of the CAN network.

Based on these requirements a CAN device supporting redundant communication shall have two CAN nodes (see figure 1). Each CAN node includes one DLL, one PCS and one PMA. Please note that these two CAN nodes are not fully independent, they share e.g. the same configuration for bit rate setting. The layer 7 application is connected to these CAN nodes by means of the DMR function (i.e. the network layer).

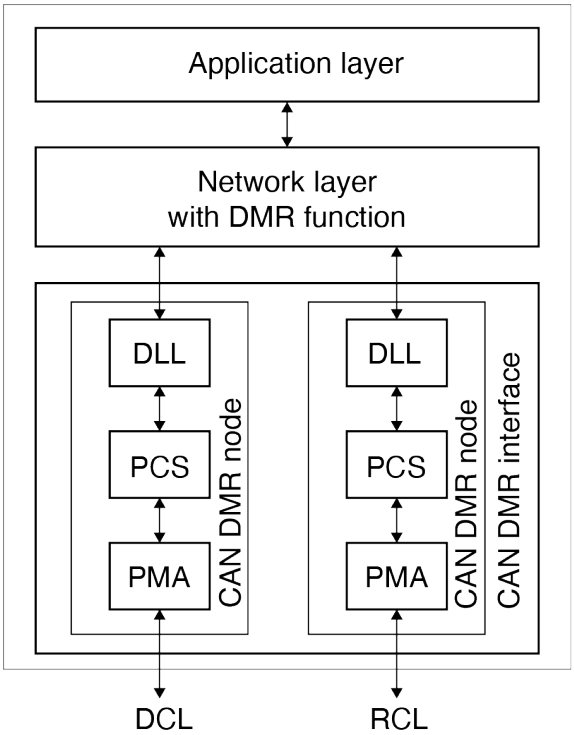

Figure 1: CAN interface with DMR function

The purpose of the DMR function is the duplication of packets between the application and the transmitting CAN nodes, as well as the de-duplication of packets received by the two CAN nodes of one CAN interface. As such, the CAN communication on both CAN lines is intended to be identical.

Naming the CAN lines

The CAN interface is connected to two cables, whereas one cable connects the first CAN node of all CAN devices in the network, it is called the default CAN line (DCL). The second cable connects the second CAN node of all CAN devices in the network and is called the redundant CAN line (RCL). These names have been taken over from the original CiA 307 specification. During system installation it must be guaranteed that cables are not interchanged.

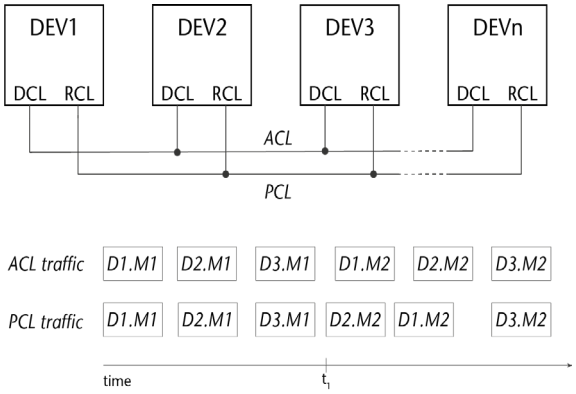

Figure 2: Traffic in redundant CAN network

Upon reception, CAN packets of only one selected line are forwarded to the application by the DMR function. This line is called active CAN line (ACL). As default, the DCL is the ACL upon system initialization. The DMR function drops packets of the other CAN line, this line is called passive CAN line (PCL). As default, the RCL is the PCL upon system initialization.

An example for redundant CAN network with traffic generated by CAN devices 1 to 3 is depicted in figure 2. Device DEV1 generates the messages D1.M1 and D1.M2, device DEV2 generates the messages D2.M1 and D2.M2 and finally device DEV3 is the transmitter of D3.M1 and D3.M2. Due to run-time issues inside the DMR or frame re-transmissions on the network it is not guaranteed that all CAN messages are sent on both CAN lines in the same order. As shown in figure 2 message D2.M2 is sent in a different order on RCL than on DCL at time t1. It is the task of the DMR function to handle this situation.

DMR function principle

As already mentioned, the purpose of the DMR function (see figure 3) is the duplication of packets to be send by the two CAN nodes of the transmitting CAN interface as well as the de-duplication of packets by the receiving two CAN nodes of the CAN interface. The CAN node HAL uses a mailbox model for hardware abstraction. A mailbox typically has a unique direction, either receive or transmit. The DMR function provides one counter and one timer for each mailbox.

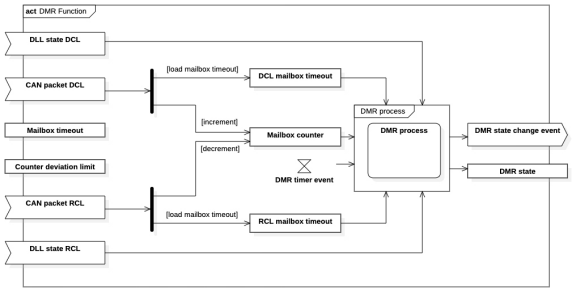

Figure 3: DMR function

A successful transmission or reception of a packet on DCL increments the associated mailbox counter by one. A successful transmission or reception of a packet on RCL decrements the associated mailbox counter by one. The DLL transmission request loads a configured timeout value for the effected mailbox. After expiration of the mailbox timer the associated counter shall be in balance, i.e. have the value 0.

The DMR process also checks the actual DLL state of each CAN node, e.g. a bus-off condition on one CAN node will prevent message transmission and as such cause an event generated for the internal DMR state machine. In addition, a DMR event is generated in case the mailbox counter exceeds a configurable deviation limit or the timeout value expires. The limits for counter deviation and mailbox timeout can be configured for the DMR function.

DMR fault detection

The fault detection is executed periodically within the DMR process action. Inside the fault detection action, the deviation from a given counter limit is calculated (see figure 4). The purpose of the deviation limit is to make

sure that the same number of messages are either transmitted or received by both CAN nodes. Due to CAN error frames or run-time issues counter values are not expected to be balanced (i.e. hold the value 0) at a specific time.

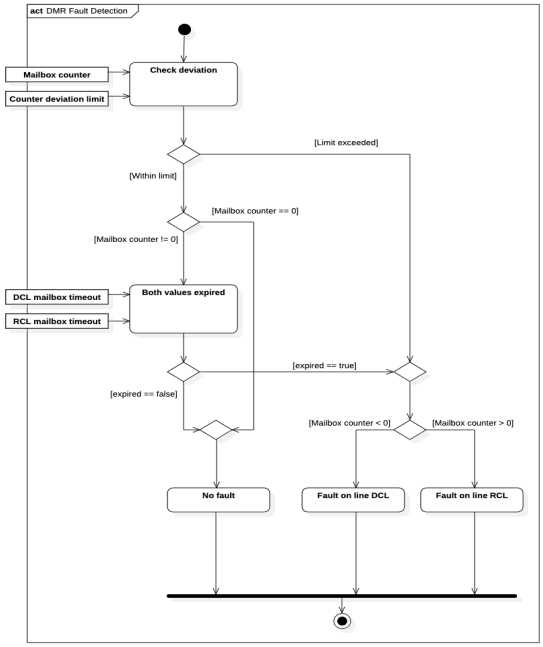

Figure 4: Fault detection

In case a mailbox counter is not greater than the given deviation limit and not balanced (i.e. not equal to 0), the mailbox timeout value of both, DCL and RCL, are tested for expiration.

The faulty CAN line is identified by evaluation of the mailbox counter value. A positive value denotes that more messages have been transferred over DCL, hence the RCL is faulty. In the same way, a negative value denotes that the DCL is faulty.

As example, a single interrupt of a CAN line at one position is depicted in figure 5: the CAN line DCL is interrupted at device DEV3.

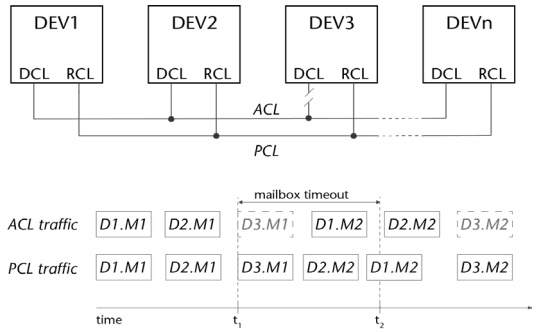

Figure 5: Example for fault condition

The failure is detected by device DEV3 since CAN message D3.M1 has not been sent on DCL (which is the ACL at this time) after expiration of the configured mailbox timeout value. A mailbox timeout is started inside the CAN device upon transmission request at time t1. The timeout is detected at time t2, causing device DEV3 to switch the ACL to line RCL. CAN nodes which are receiving messages from DEV3 will detect an error condition by detecting a receive counter difference after expiration of the configured mailbox timeout value.

In case of a line switch the DMR function must ensure that any message reception FIFOs existing inside the layer 2 driver are synchronized. In addition, any pending messages to be transmitted on the faulty line – i.e. the new PCL – must be deleted for the upcoming DMR recovery process.

The DMR function does not transmit a specific CAN message to the network to inform about a local alarm event. Generation of alarm events is in the scope of the application layer.

DMR state machine

The DMR function has a built-in state machine which is controlled by means of the DMR processing algorithm, depicted in figure 6.

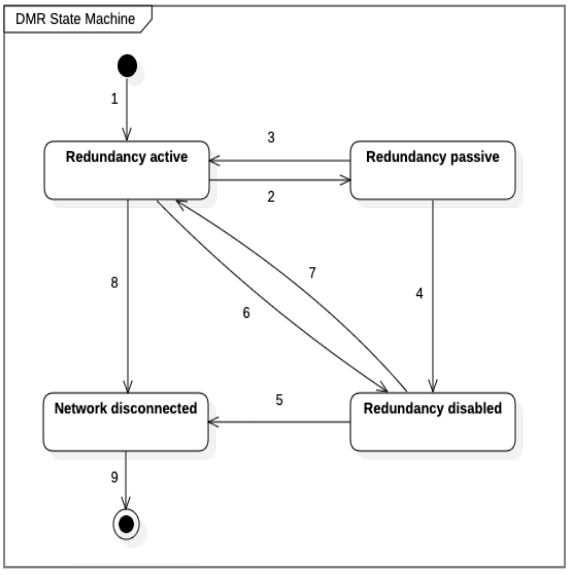

Figure 6: DMR state machine

After initialization, the DMR switches to Redundancy active state. In Redundancy active state both CAN lines, DCL and RCL, are fully operational. By default, the DCL is used as ACL.

In Redundancy passive state transmission of CAN frames is possible on both CAN lines, DCL and RCL. However, the DMR processing has detected a fault upon frame reception.

In Redundancy disabled state one of the CAN nodes, either DCL or RCL, is not able to transmit. As such, CAN communication is possible only via one single CAN node. The DMR processing has detected a fault upon frame transmission.

In Network disconnected state both physical CAN interfaces are not operational.

The reason for introducing a DMR state machine which allows to distinguish between fault conditions on reception or transmission is error diagnostics. Referring to the example from figure 6 the individual state of devices DEV1 to DEV3 is shown in table 1.

Table 1: DMR states for example from figure 5

Device	DMR state
DEV1	Redundancy passive
DEV2	Redundancy passive
DEV3	Redundancy disabled

This information may be forwarded to the network control application by each device via the used higher layer protocol for fault finding.

DMR Connection Service

The intention of the DMR connection service is to check for faulty wiring (DCL connected to RCL) and to support the fault recovery. For this purpose, each CAN line uses a unique CAN identifier (see figure 7). These two CAN identifiers are reserved solely for usage by the network layer (DMR function).

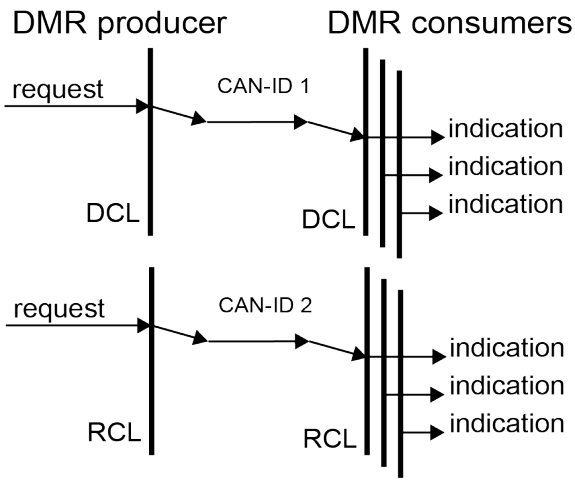

Figure 7: DMR connection service

The configuration of CAN-IDs for the DMR connection service is done by the local application and should of course not interfere with the used application protocol.

Using different CAN-IDs for each CAN line allows to monitor if all DCL lines are connected as well as all RCL lines. In fact, it is difficult to detect the source of a faulty connection with a high degree of probability, since there is no source address within the message (a DLC value of 0 is used). As such, the number of „good“ and „bad“ messages is counted and made available for application access inside the DMR state information. Reporting a faulty connection is not within the scope of the DMR, instead it is in the scope of the application.

Fault Recovery

An important feature of the DMR is the automatic fault recovery, depicted in figure 8. The fault recovery is executed periodically within the DMR process action. Inside the fault recovery action, a delay of 2 seconds is inserted to recover from a potential bus-off situation. Afterwards the CAN line is switched into ’Operational’ state, followed by transmission of the DMR Connection Service.

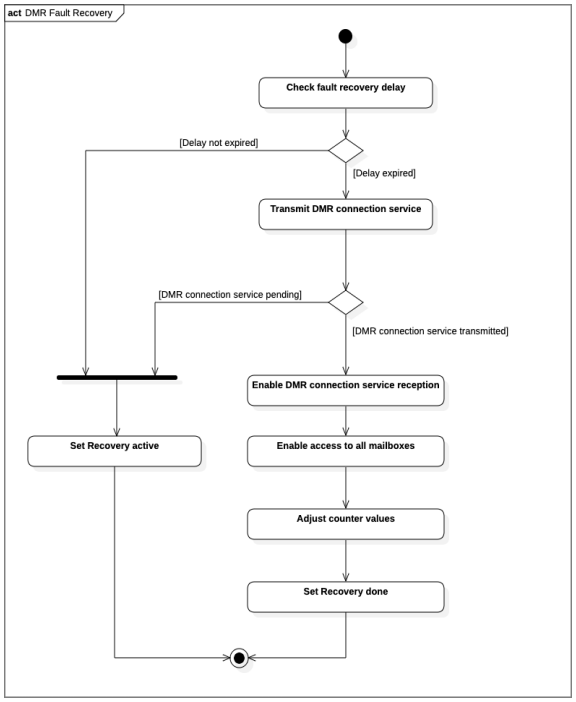

Figure 8: DMR fault recovery

Since the CAN-IDs used by the DMR connection service are filtered by the network layer any interference to a higher layer protocol is prevented. Upon successful transmission of the DMR Connection Service the DMR state is updated.

DMR Testing

For verification a DMR test plan has been created which uses a test setup depicted in figure 9.

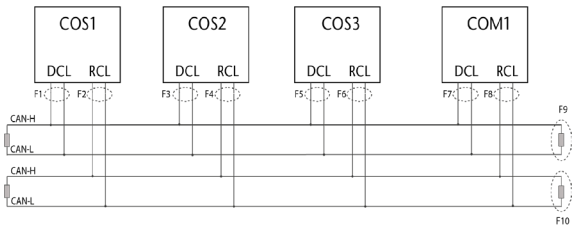

Figure 9: Schematic for DMR test setup

A fault condition can be applied to the network during testing at the places marked from F1 to F10.

Table 2: Fault conditions for test setup

Marker	Operating	Fault
F1 .. F8	CAN bus connected	CAN bus disconnected
F9 .. F10	Termination available	CAN lines short-circuited

The CAN network is operated with a bit rate of 125 kBit/s and an average bus-load of 85% during testing.

As an example test case the “RCL startup failure” is depicted here as pseudo code together with the acceptance criteria.

```
POWER ON DUT
@SubTest01
    VerifyNmtState($dut)
        IF NMT STATE $dut != OPERATIONAL
            ERROR
```

Example 1: RCL startup failure

Table 3: Acceptance criteria upon RCL startup failure

Fault	COS1	COS2	COS3	COM1
F2	DMR: disabled ACL: DCL	DMR: passive ACL: DCL	DMR: passive ACL: DCL	DMR: passive ACL: DCL
F4	DMR: passive ACL: DCL	DMR: disabled ACL: DCL	DMR: passive ACL: DCL	DMR: passive ACL: DCL
F6	DMR: passive ACL: DCL	DMR: passive ACL: DCL	DMR: disabled ACL: DCL	DMR: passive ACL: DCL
F8	DMR: passive ACL: DCL	DMR: passive ACL: DCL	DMR: passive ACL: DCL	DMR: disabled ACL: DCL
F10	DMR: disabled ACL: DCL	DMR: disabled ACL: DCL	DMR: disabled ACL: DCL	DMR: disabled ACL: DCL

Summary

Based on the new CiA 701 standard implementations of the DMR function have been tested and validated on different platforms, including bare-metal silicon like STM32, i.MX RT1170 or operating systems like VxWorks and Linux. First products are already in field test.

The next step on the specification roadmap is the specific adaoption to CANopen CC and CANopen FD, i.e. access to DMR parameters via standardized data elements in the object dictionary.

References

[1] CiA 307, CANopen Framework for maritime electronics, Version 1.01.01, February 2004
[2] DNV GL, Rules for Classification, Ships, part 4, chapter9, Control and monitoring systems
[3] CiA 701, Dual modular CAN networks, Part 1 – Generic network layer, Work Draft

Uwe Koppe
MicroControl
Junkersring 23
DE-53844 Troisdorf
koppe@microcontrol.net
www.microcontrol.net

Johann Tiderko
MicroControl
Junkersring 23
DE-53844 Troisdorf
johann.tiderko@microcontrol.net
www.microcontrol.net