Message end-to-end protection for small monolithic devices

Fred Rennig (STMicroelectronics)

Message end-to-end (E2E) protection is required for safety critical functions used in automotive, industrial and other applications. A cyclic-redundancy check as it is part of the CAN protocol is not sufficient for these message protection needs. Other failure mechanisms like the ones specified in ISO 26262 must be respected as well.

The presentation describes an E2E message protection proposal for small devices with a monolithic CAN FD light responder implementation. These circuits do not embed a computation core capable of running software. Therefore the implementation must be done entirely in hardware. The proposed E2E message protection scheme is based on AUTOSAR E2E profiles adapted to the constraints and limitations of monolithic integrated devices used for actuators and sensors that are only able to contain a limited amount of digital functions.

Monolithic end-point devices in the car

With the evolution of centralized car network architectures such as the zonal car architecture more and more computation power is going to be moved into central controllers like zonal gateways or zonal computers.

This leads to an increase in sensor/actuator devices that do not run any or only very limited software. These devices must be connected to the computational controllers by a safe and reliable network connection. Ideally, these so-called end-node devices are full-monolithic integrated single device chips that work software-less.

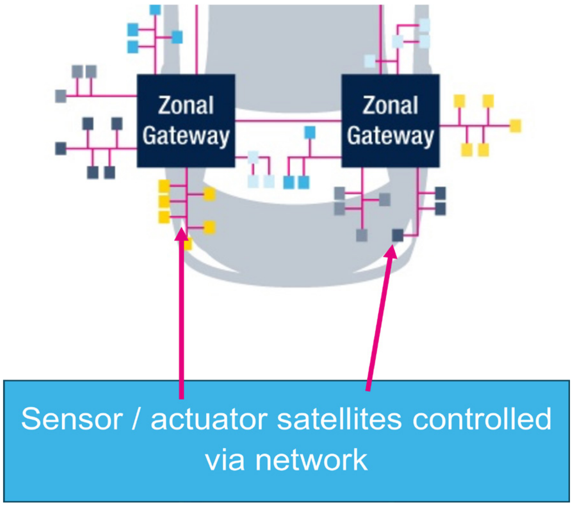

Figure 1: End-nodes in a zonal car architecture

Many of these end-nodes are part of safety critical functions in the body, safety or lighting domain. Therefore, they must fulfill automotive safety integrity level (ASIL) requirements.

Communication with end-point devices

An example of a communication solution is the CAN FD light network protocol, which is an implementation of the CAN FD protocol aimed at the cost-efficient integration into small monolithically integrated semiconductor devices.

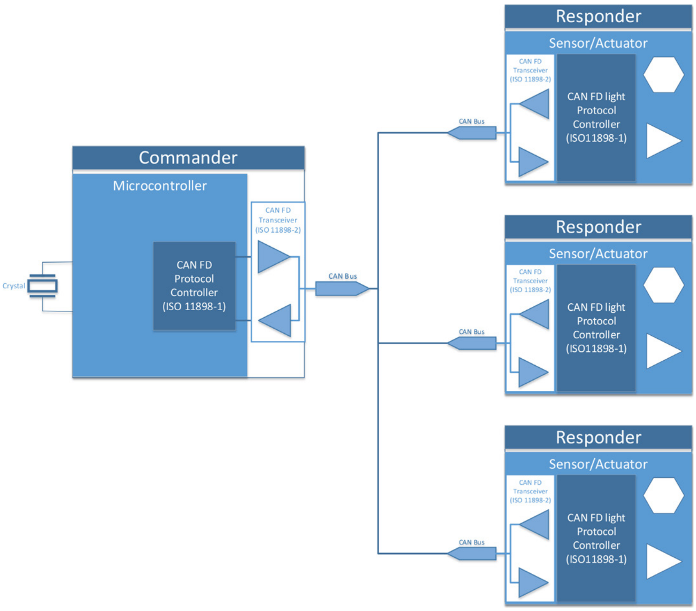
Commander
Microcontroller
CAN FD Protocol Controller (ISO 11898-1)
CAN FD Transceiver (ISO 11898-2)
Crystal
CAN Bus
Responder
Sensor/Actuator
CAN FD Transceiver (ISO 11898-2)
CAN FD light Protocol Controller (ISO11898-1)
CAN Bus
Responder
Sensor/Actuator
CAN FD Transceiver (ISO 11898-2)
CAN FD light Protocol Controller (ISO11898-1)
CAN Bus
Responder
Sensor/Actuator
CAN FD Transceiver (ISO 11898-2)
CAN FD light Protocol Controller (ISO11898-1)
CAN Bus

Figure 2: CAN FD light communication network

In the network shown in Figure 2 the commander is a standard microcontroller or microprocessor with a CAN FD protocol controller according to ISO 11898-1. The responders have a CAN FD light responder protocol controller according to the upcoming ISO 11898-1:2024 Annex A implemented. This protocol can be efficiently integrated into monolithic sensor/actuator devices that do not run any software. It is cost-efficient and does not need an external frequency setting component like an automotive crystal. All devices on the bus are connected by an ISO 11898-2 physical layer, either integrated together with the other functions on the same chip or as a standard transceiver product.

Communication Faults

A variety of communication faults may occur in all types of communication networks. For communication safety these faults must be detected so that the system can react and correct them. A list of possible faults can be obtained e.g. from the standard for functional safety ISO 26262.

The potential faults include repetition of information, loss of information, delay of information, insertion of information, masquerading, incorrect addressing, incorrect sequence of information, corruption of information, asymmetric information sent from a sender to multiple receivers, information from a sender received by only a subset of the receivers and blocking access to a communication channel.

Communication protection techniques

From this long list and from the nature of these faults it can be deduced that a CRC is not sufficient to detect all of these errors. Additional measures must be implemented to be able to discover the above-mentioned communication failures.

Table 1 shows the proposed communication protection technique for the communication failures listed in ISO 26262 and which faults can be detected by them. These techniques must be implemented into the communication end-nodes to improve ASIL-grade safety in end-to-end (E2E) communications. Besides the CRC a data identifier and a message counter are needed to protect the message itself plus a watchdog for timeout monitoring of regular transmitted messages.

Fault detection implementation

The end-to-end protection is added on top of the existing communication protocol like CAN FD light. Figure 3 shows an implementation in a microcontroller in software as it is used in many communication systems.

In this example, the communication data is exchanged via the CAN FD protocol over the bus using a CAN FD protocol controller hardware implementation and CAN FD transceivers. The protocol controller checks the CRC code and additional errors such as form and stuff errors.

Table 1: Protection techniques

E2E protection technique	Detectable communication faults
Cyclic redundancy check (CRC)	Corruption, Asymmetric information
Transmission on a regular basis and timeout monitoring using E2E-Supervision (including watchdog)	Loss, delay, blocking
Data ID + CRC, (commander - responder scheme)	Masquerade, incorrect addressing, insertion
Message Counter	Repetition, Loss, insertion, incorrect sequence, blocking

Additional end-to-end protection is implemented in software and protects the data transmitted via the CAN FD protocol. Here, the above-mentioned end-to-end protection techniques are used to find the faults as shown in Table 1.

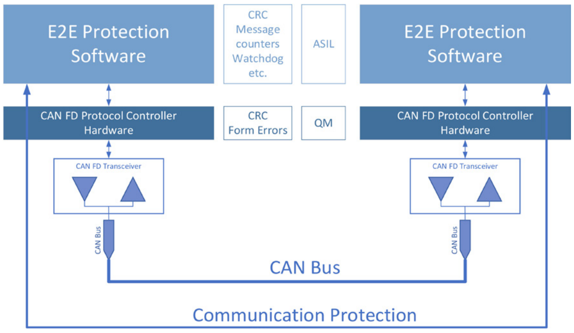

Figure 3: E2E protection as implemented today in software

As can be seen from Figure 3 the protection techniques of the protocol controller and the ones from the additional end-to-end protection are used together.

End-to-end protection in monolithic devices

For small end-nodes like the before-mentioned sensor/actuator devices an end-to-end protection in software is either not feasible or not economical because it would require an additional microcontroller core and software on ASIL-level only for this purpose.

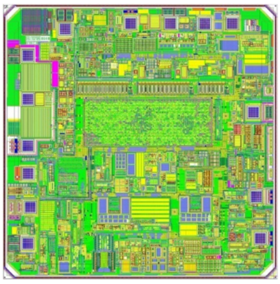

Figure 4: Mixed signal chip

Figure 4 shows a common mixed signal monolithic device. Most of the area is used for analog or driver components and only a little portion in the center is used for digital circuitry. Due to the nature of the manufacturing processes for this kind of mixed-signal-devices the digital circuitry occupies a larger area than it would when using a manufacturing process for pure digital circuits like e.g. microcontrollers. Therefore, an end-to-end implementation in hardware must be of minimal complexity using as few digital components as possible.

End-to-end protection profile

The additional information needed to transmit the end-to-end protected data must fit together with the transmitted data into the 64 bytes of a CAN FD (light) data segment.

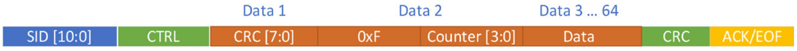

Figure 5: CAN FD (light) frame with E2E protected data

Figure 5 shows a CAN FD (light) frame with its data. In the first two data bytes an eight-bit CRC value and a four-bit message counter are transmitted. The data identifier is not part of the data payload but is used to calculate the eight-bit CRC.

End-to-end protection generation

The portions to calculate the CRC are shown in Figure 6:

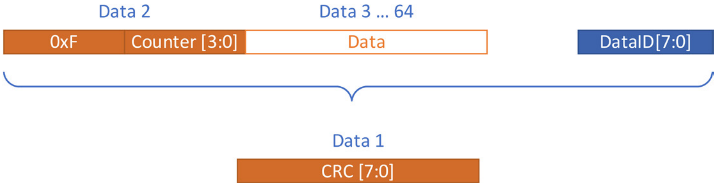

Figure 6: CRC calculation parts

With this profile the flow to generate the message control field becomes straightforward. This is shown in Figure 7:

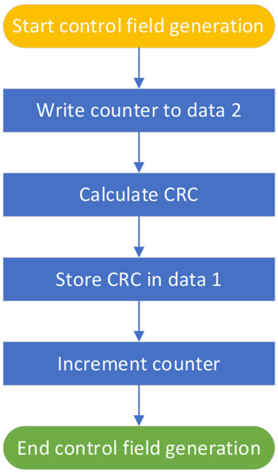

Figure 7: Control field generation

After the message counter has been written to data byte 2 the CRC is calculated and stored in data byte 1. Afterwards the message counter is increased so it is ready for the next message.

End-to-end protection check

The end-to-end protection check flow on the recipient side is similar straightforward as can be seen in Figure 8.

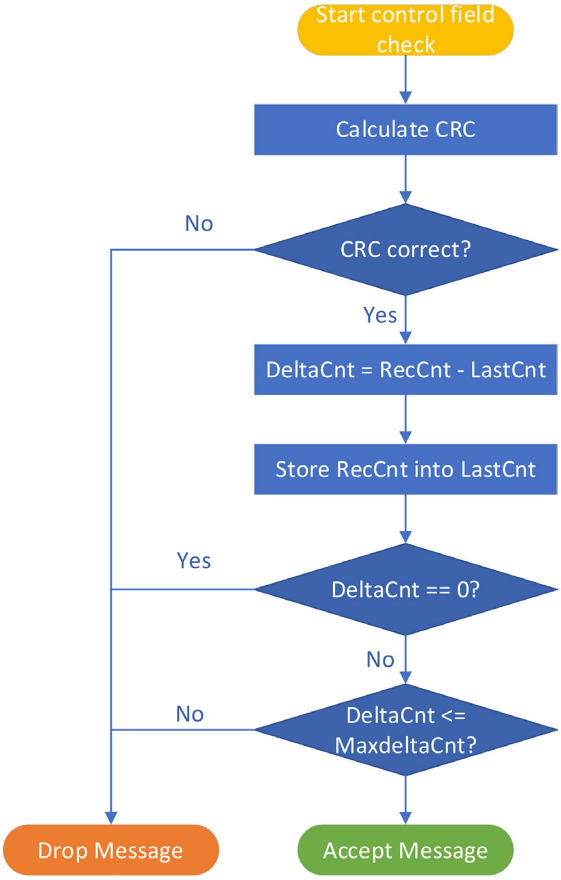

Figure 8: E2E check flow

After the verification of the CRC the difference between the previously received message counter value and the current received message counter value is calculated. In case it is zero the message is seen as repeated and therefore dropped. If the difference is larger than a predefined value it is assumed that too many messages have not been received between the last two received messages and that received message cannot be used anymore. Therefore, this message is dropped as well. The threshold value is usually larger than one to allow for a few not received messages if they do not impact the safety of the system.

For the CRC calculation the stored data identifier belonging to the message which has not been transmitted is used. This requires both sides of the communication to use the same data identifier.

End-to-end protection error counter

Whenever a message has been dropped, an error counter is increased, whenever a message has been correctly received this counter is decreased. The value by which it is increased or decreased is chosen by the implementer based on the system. This determines how many successful receptions are needed to account for one dropped message. In case too many errors have been accumulated the device enters a fail-safe-mode. This flow is shown in Figure 9.

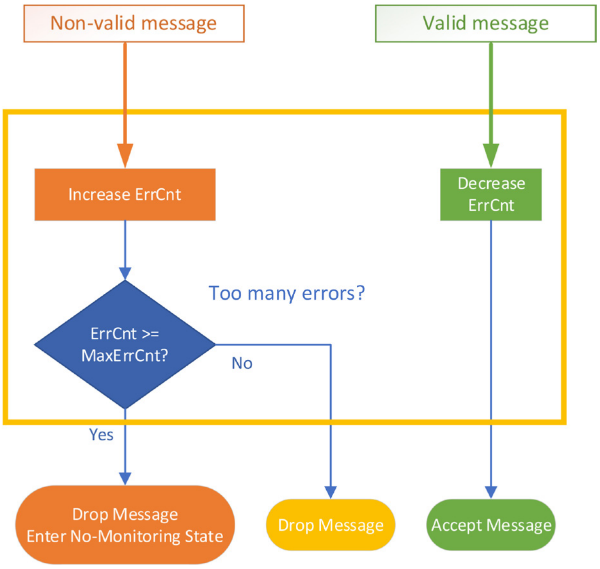

Figure 9: End-to-end error counter

End-to-end protection state diagram

Before the end-to-end protection can be used for safe communication it must be initialized by the sender of the messages. Until the end-to-end protection is not properly started no safety critical messages can be received. The state-diagram is shown in Figure 10.

After power-up reset the device enters the no-monitoring state. In this state the end-to-end protection data is generated and checked after reception, but no message is dropped, and the error counter is not increased. Safety critical communication data is ignored.

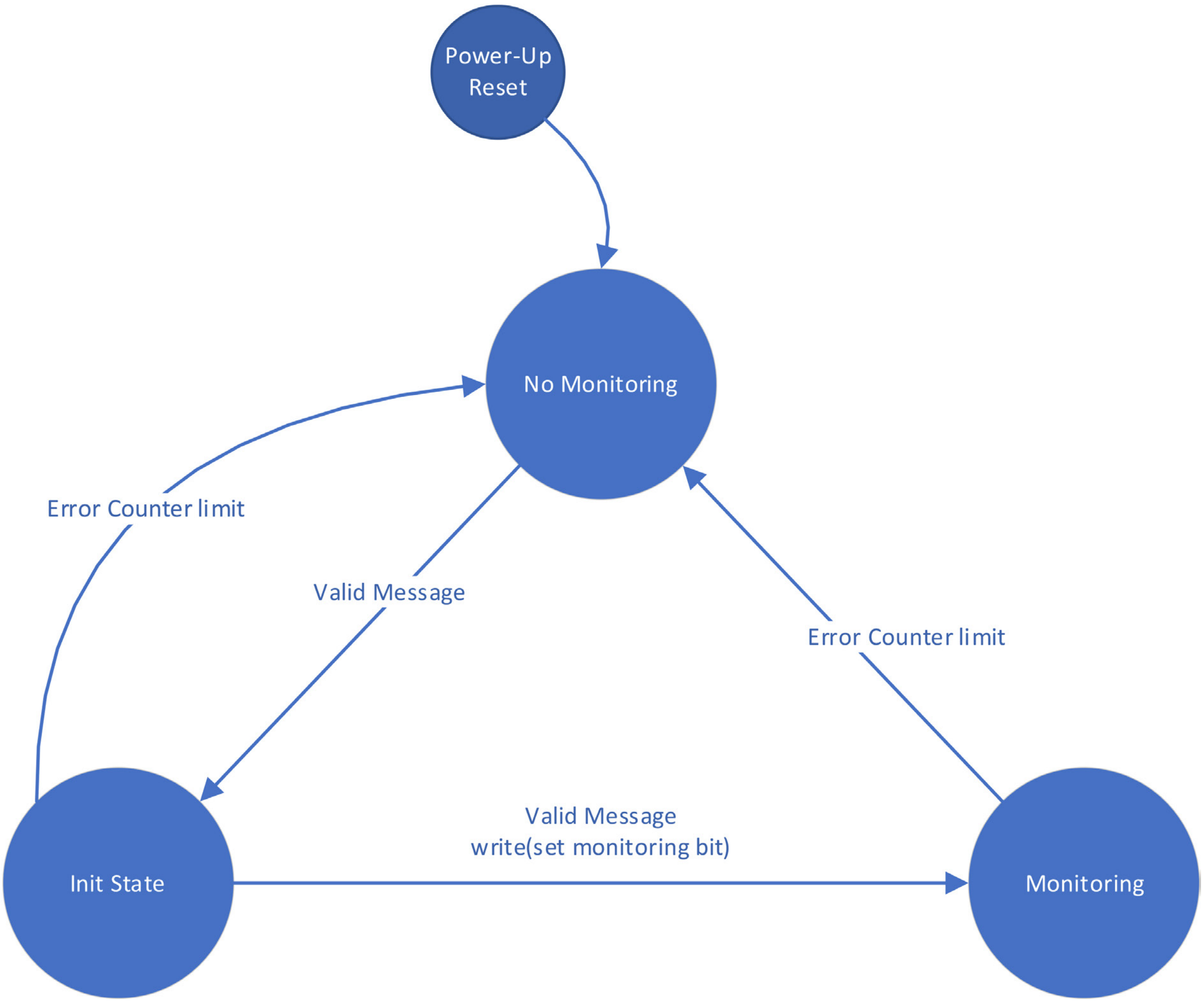

Figure 10: End-to-end protection monitoring state diagram

As soon as a valid message is received the device enters the initialization state (“Init-State”), in which the incoming messages’ end-to-end protection is checked and the error counter is increased and decreased as described. But safety critical communication data is still ignored. In case the error counter reaches its maximum value, the device goes back to the no-monitoring state.

Only in Init-State it is possible to enter the monitoring state in which safety critical communication is allowed. Entering the monitoring state can be achieved by e.g. writing a dedicated monitoring bit in a register.

In case the error counter limit is reached in monitoring state the device enters no-monitoring state and disables safety critical communication until the monitoring state is entered again by using the described procedure.

Summary

End-to-end protection is required for safety critical communication. In monolithic integrated devices resources are very limited and the end-to-end protection must be implemented in hardware only. The CRC of the communication protocol controller is not sufficient, additional measures must be used.

A simplified flow has been shown, which can be integrated into such small devices easily.

Fred Rennig
STMicroelectronics
Bahnofstr. 18
DE-85609 Aschheim-Dornach
www.st.com