FlexRay demonstrator for certification

Felix Bruckmüller, Erwin Krister, Wilfried Kubinger

FlexRay demonstrator for certification

Felix Bruckmüller[1], Erwin Kristen[2], Wilfried Kubinger[3]

AIT Austrian Institute of Technology GmbH, Department Safety & Security – Safe and Autonomous Systems, Donau-City-Straße 1, 1220 Vienna, Austria[1,2]
University of Applied Sciences Technikum Wien, Department for Mechatronics, Hoechstaedtplatz 6, 1200 Vienna, Austria[3]
felix.bruckmueller.fl@ait.ac.at[1], erwin.kristen@ait.ac.at[2], kubinger@technikum-wien.at[3]

Abstract. An always increasing number of electronic control units monitor and operate drive-train and other safety related functions in automotive applications. Transferring this kind of functionalities like steering or breaking to these devices requires high dependability and reliability during every possible operation scenario. Therefore developing and certifying safety related electronic components to appropriate specification of standards is a key task.
The certification process is a complex, labor-intensive, hence costly but absolutely necessary task to certify this kind of products to allow their legal usage. This paper describes a tool framework with a prototype example to demonstrate automated testing targeting certification. The example application to test is a steer-by-wire demonstrator utilizing four electronic control units interconnected via FlexRay to control a mechanical system build using fischertechnik.

Keywords: Automated certification, FlexRay, SafeCer, Steer-by-wire

1 Introduction

Testing and certification are two absolutely essential tasks during development of any safety related product or service. At the same time these two tasks are often also the most time consuming parts of development, especially for safety critical systems. In the near future safety critical systems will be added to many different systems.

Furthermore, testing often requires human interaction to proceed during some steps of the workflow. Reducing the amount of human interaction required is a goal which leads to great potential savings in time and money. There are currently projects running trying to reach this goal by automating these steps within the workflow. One of these is the ARTEMIS JU SafeCer project [1].

This paper describes the dedicated hardware and software developed by AIT during the SafeCer project to demonstrate the procedures and methods used to test and validate safety critical systems. This is done in the SafeCer subproject BOLDI (BusScope On-Line Diagnosis) demonstrator. Furthermore the design is planned to fit inside a suitcase to allow easy transportation to exhibitions and demonstration events.

1.1 AIT Safety and Security

AIT is the largest non-university research institute in Austria mainly focused on infrastructure topics of the future. Acting as a bridge between university research and industrial application the range between developing a proof of concept and a demonstrator of the end product is covered. AIT provides a research environment and service for mid- and long-term goals for European and in particular Austrian industries.

AIT's department "Safety and Security" works on methods and solutions to validate safety critical systems. As participant in the SafeCer project AIT contains work on a definition for an automated testing and certification process for safety critical applications.

1.2 SafeCer

SafeCer is an international research project coordinated by the Swedish Volvo Technology Corporation with up to 29 participating organizations spread across six European countries. National sponsorship in combination with funding supplied by the ARTEMIS JU allows spending a total amount of € 25.7 Mio. during a 48 month period in the two subprojects nSafeCer and pSafeCer. Definition and development of a FlexRay demonstrator and using that to show manual and automated testing is one part of this project [2,3].

2 Safety critical systems

With evermore electronic devices controlling safety related functionalities safety critical systems are getting more attention. A safety critical system is a system where a failure could lead to massive consequences that are in some cases even life-threatening. Therefore, assuring the claimed level of safety is crucial. To do so the device in question is extensively tested to provide information on its behavior under different conditions. This information is bundled to a safety case which is used to verify the device for a certain safety integrity level (SIL). For road vehicles ISO 26262 („Road vehicles – Functional safety") defines four SIL levels from ASIL A (automotive SIL) with a probability of failure of <10-6 per hour to ASIL D with a probability of failure of <10-8 per hour. A steer-by-wire application used as demonstration for the BOLDI demonstrator has to be verified to the ASIL D level. Current processes used for testing of safety critical systems require human interaction in different steps, especially for test case definition and evaluation.

3 BOLDI demonstrator

The intention of the BOLDI demonstrator is to develop a showcase for a tool framework for a highly automated execution of test cases to generate the safety case for the test object (see [1]). The test object within the BOLDI demonstrator is a time triggered FlexRay communication network cluster for a steer-by-wire application.

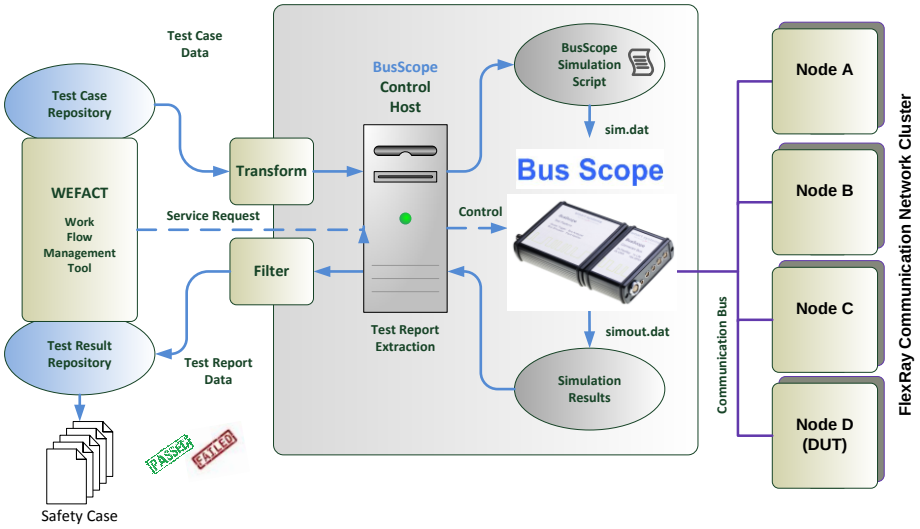

Fig. 1. BOLDI demonstrator overview

As shown in **Fig. 1**, the BOLDI demonstrator uses four FlexRay nodes (node A to node D) to realize a steer-by-wire application with node D being assigned as DUT (Device Unter Test). For the purpose of demonstration only the software of the DUT node shall be verified. To perform the verification process the BusScope test and simulation platform developed by AIT is a central component acting as a test data injector and data tracer. Appropriate test cases are provided by the WEFACT work flow management tool which generates simulation scripts holding startup and simulation matrices that are used by the BusScope via the BusScope control host. All responded data frames recorded by the BusScope are sent to the control host, are collected and then send back to WEFACT for further steps to generate the safety case.

3.1 Steer-by-wire application

A literature research reveals that different compositions of steer-by-wire systems were already developed and described [4-7]. Such a system has two main functionalities as described in [4]. The most important service provided is road wheel control. This is the main, vehicle and driver safety related task that must be fulfilled during every situation the vehicle is put through. The second functionality is providing road feedback to the driver via the steering wheel which is not covered in the current implementation of the demonstrator.

For proper functionality of the steer-by-wire system the steering controller evaluates the steer angle information generated by sensors connected to the steering wheel and transfers this information to the road controller. The road controller calculates the current steer angle by analyzing values produced by sensors within the front steering wheel plant for actuation of the road wheels. This information together with the target steer angle is used to generate a motor torque value to actuate the steering assembly.

For a steer-by-wire application a reliable communication between sensors and actuators is absolutely necessary. Consequently, a bus system with focus on automotive applications is chosen. There are several different bus protocols available on the market divided into event-triggered and time-triggered protocols. Time-triggered protocols like FlexRay have the advantage of being deterministic. They use a predefined, recurring communication scheme where every device is only allowed to send its messages during its time slot.

3.2 FlexRay network communications protocol

FlexRay is chosen instead of CAN, another protocol heavily used in the automotive sector, mainly because of its deterministic characteristic resulting from being a time triggered protocol allowing a totally predictable behavior of communication. In contrast CAN is an event-triggered bus system where a message is sent when an event occurs and the bus is free of other communication. Communication on the FlexRay bus works according to a predefined schedule. More information on FlexRay is provided by [8] and [9] and additional information on time-triggered protocols by [10].

BusScope is able to analyze communication on the bus and to add messages to the communication in order to generate test data needed for a safety case.

3.3 BusScope test and simulation platform

BusScope is based on a specific analyzer technology which uses a digital signal sampler, sample storage and a processor, which reconstructs data within the received FlexRay frames but also generates frames to inject into the network. The BusScope provides four main functionalities available through a graphical user interface and a programming API. The “Scope Trigger” function generates trigger signals depending on different characteristics of frames like Frame ID or payload length. The “Bus Analyzer” function directly parses the bus signal without introducing delay or signal changes as a communication controller would do. The “Bus Simulator” function simulates one or more complete FlexRay nodes and can act as bus master. The “Fault Injector” function allows using the BusScope as a bridge between two bus segments and for manipulating frames according to a predefined schema.
During this work the BusScope will be used as Bus Analyzer and Bus Simulator as shown in **Fig. 1**.

4 BOLDI demonstrator concept and implementation

The BOLDI demonstrator is a system to demonstrate the possibilities of AITs automated testing and certification system. The first of the two main parts is the mechanical steer-by-wire hardware assembly consisting of four motors (MAA, MAB, MDA, MDB) with embedded rotatory sensors actuated by two dual motor controllers and a steering gear assembled from three differential gearboxes (DiffA, DiffB, DiffOUT) with attached rotatory sensors (RA, RB, ROUT). **Fig. 2** shows the block diagram of

this hardware assembly. The second part is the electronic part consisting of the FlexRay nodes, the BusScope and the BusScope control host PC. In **Fig. 3** the preliminary demonstrator is shown.

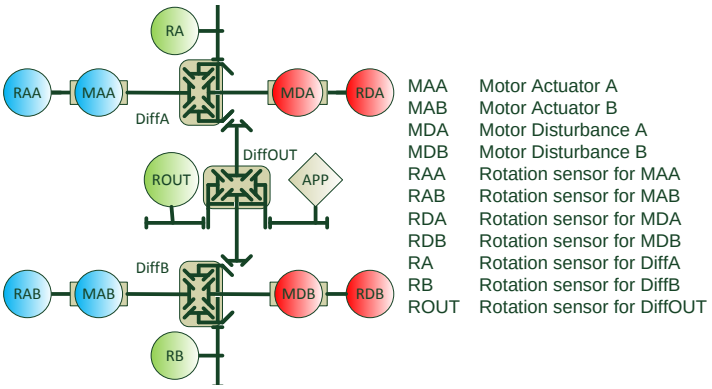

Fig. 2. Mechanical demonstrator concept

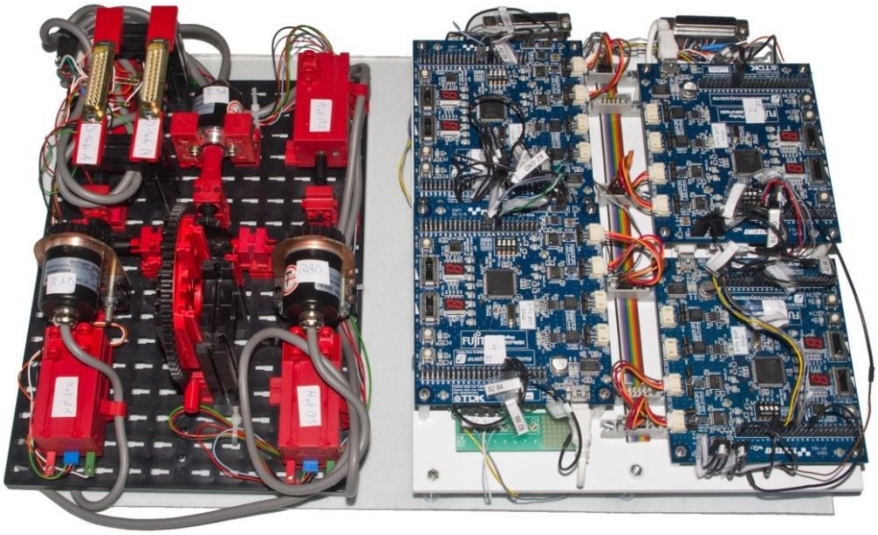

Fig. 3. Demonstrator prototype assembly: Mechanical assembly on the left hand side, board assembly on the right hand side

Derived from the proposed demonstrator functionality and the planned timeline the following targets are defined for the BOLDI demonstrator.

- Provision of redundant actuation motors and motors introducing disrupting forces
- Rotational speed control for single motors and control algorithm for steering angle
- Usage of four ECU for distributed control with all variables exposed over FlexRay
- Complete system shall fit in a suitcase for easy transportation

- Sophisticated closed-loop control algorithms are not needed for verification demo
- Only commercial off-the-shelf components used for the steer-by-wire assembly

The Fujitsu “bits pot blue” FlexRay evaluation boards based on the MB91F465XA microcontroller running at 100 MHz are used as FlexRay nodes. Featuring one dual-channel FlexRay controller (BOSCH E-Ray on-chip), one RS-232 interface and more than 40 input and output pins it connects with austriamicrosystems AS8221D transceivers to the FlexRay network. Software examples for all board functionalities are available from Fujitsu together with the IDE Fujitsu SOFTUNE Workbench. Furthermore DuxSolutions provides an open source FlexRay driver and the Network Designer::FlexRay used to design the FlexRay communication schedule.

The FlexRay node D acts as the master steering angle controller. It receives all data from the other nodes and computes the target torque of the actuation motors to reach the target steering position. Another node features a serial RS232 link allowing sending commands to the cluster via a PC. All values measured or generated by the nodes are made available on the FlexRay bus. The one node defined as the device under test reacts to externally defined FlexRay messages provided by the BusScope for testing.

The FlexRay schedule designed for the demonstrator incorporates six nodes, each using five 32 word long payload frames. Four of these six nodes are the nodes used by the demonstrator while the other nodes are simulated by BusScope during testing operations. With a cycle time of 5ms about 50% of the communication cycle occupied by the 30 static slots and the remaining time is used by 354 dynamic slots which are not in use.

Within an Excel sheet the assignment of messages and data to FlexRay frames is defined. In addition this Excel sheet automatically generates a header file defining the frame setup to use with the C-code of the microcontroller implementation for easy assembly and disassembly of FlexRay messages to send and receive.

The software architecture uses interrupt assisted tasks for the main operations of the devices. Time critical tasks and tasks without regular occurrence like rotational speed measurement are executed directly by the corresponding interrupt routines. Regular tasks like communication over FlexRay, calling control algorithms or update board output values (PWM for motor control, LEDs for user interaction) are executed within the main loop aligned to the start of cycle interrupt generated by the FlexRay controller. Doing so allows starting the task chain every 5ms and therefore generating a stable time-base for the control algorithms running with 200Hz. All run time calculations use fixed point operations to avoid slow emulation of floating point operations.

5 Test process

Testing is a crucial task to provide information on safety critical system behavior under different conditions in order to produce a safety case. For this process the BusScope, the BusScope control host and the demonstrator as application to test are used. For the purpose of demonstration only the application software of node D is tested. Valid and invalid target angle values are sent to node D to verify the correct operation of the application software.

5.1 Test execution

A test run starts with an overall reset of the FlexRay nodes to reach a defined start of test conditions. Then the BusScope starts up the FlexRay communication network cluster with the "Startup Communication Matrix" as seen in **Fig. 4**. This matrix is executed until all nodes are synchronized to the network and ready to start the actual test. This communication matrix is free of faults to ease the integration phase.

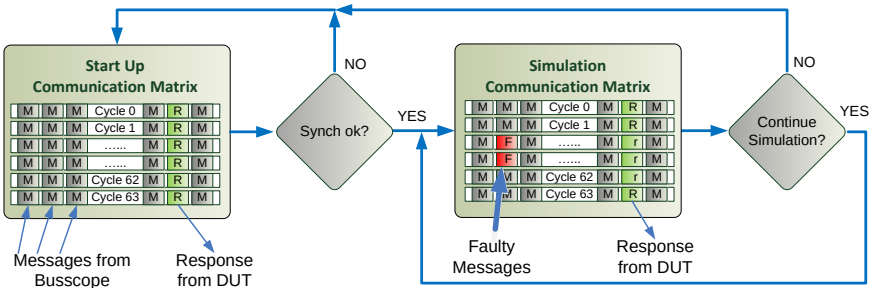

Fig. 4. BusScope simulation run

Before the BusScope device switches over to the "Simulation Communication Matrix" data tracing is started to record all FlexRay frames and timestamps on the BusScope control host. The "Simulation communication matrix" holds prepared data frames to test the reaction of the device under test. The "Simulation communication matrix" is executed one time or a given number of times before the BusScope switches back to the "Startup communication matrix". The trace file is closed and a test case data report is generated.

5.2 Test evaluation

The results of test execution in the form of a test case data report must be evaluated to get a test result. **Fig. 5** shows the test report extraction process flow. The test case data report is filtered for relevant data according to the requirements of the test case. The evaluator module uses this data to determine the test result (PASSED or FAILED). If an error occurs during test case execution the result NOT ACCOMPLISHABLE is reported in the test report of the corresponding test case.

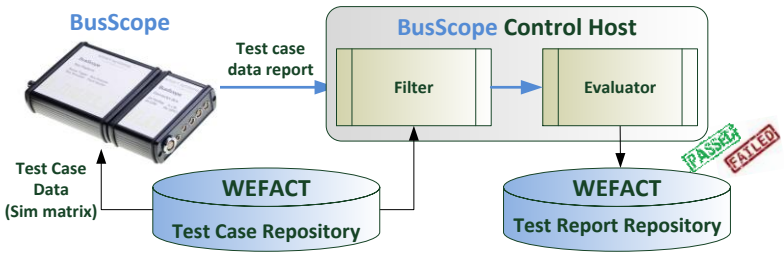

Fig. 5. Test result extraction process flow

6 Conclusion and future work

In this paper the design, implementation and future usage of the BOLDI demonstrator as part of the SafeCer project is described. It uses a combination of the BusScope test and simulation platform, FlexRay evaluation boards and fischertechnik mechanics to demonstrate testing of a safety critical application. A steer-by-wire assembly is used as an example for a safety critical system using FlexRay for communication. Finally the procedure for testing a FlexRay node using the BusScope and evaluating the results is described.

During development the Fujitsu FlexRay evaluation boards proved to be a very reasonably priced way to prototype FlexRay applications. The provided tools and examples facilitate straightforward development. The BusScope aids the development process by providing an intuitive way to analyze communication on the FlexRay bus.

In the future further developments will be made to reach the target of fully automated testing. One step on this way is the implementation of an automatic test case generator. Furthermore the demonstrator hardware will be rearranged and fitted into a proper suitcase for easy transportation to exhibitions and demonstration events.

7 References

1. SafeCer project homepage. http://www.safecer.eu/
2. SafeCer project partners, 2012. nSafeCer summary. http://vif.tugraz.at/fileadmin/user_upload/area_e/nSafeCer_leaflet_201210.pdf
3. SafeCer project partners, 2012. pSafeCer summary. http://vif.tugraz.at/fileadmin/user_upload/area_e/pSafeCer_leaflet_201210.pdf
4. Chaaban, K., Rizoug, N., Barbedette, B. & Saudrais, S., 2012. Model-based development of an embedded steering-by-wire system. In: IEEE, 2012 8th International Symposium on Mechatronics and its Applications (ISMA). American University of Sharjah, April 10-12, 2012, Sharjah, United Arab Emirates.
5. Hayama, R., Higashi, M., Kawahara, S., Nakano, S. & Kumamoto, H., 2010. Fault-tolerant automobile steering based on diversity of steer-by-wire, braking and acceleration. Reliability Engineering & System Safety, 95(1), pp.10–17.
6. Frede, D., Khodabakhshian, M. & Malmquist, D., 2010. A state-of-the-art survey on vehicular mechatronics focusing on by-wire systems: KTH Royal Institute of Technology.
7. Song, C.-C., Chen, W.-C., Feng, C.-F. & Liaw, D.-C., 2010. Study of a vehicular drive-by-wire system based on flexray protocol. In: IEEE, Proceedings of SICE Annual Conference 2010. The Grand Hotel, August 18-21, 2010, Taipei, Taiwan.
8. FlexRay Consortium, 2005. FlexRay Communications System Protocol Specification Version 2.1. Stuttgart, Germany.
9. Lorenz, S., 2010. The FlexRay Electrical Physical Layer Evolution. SPECIAL EDITION HANSER automotive FLEXRAY 2010, December 2010 pp.14–16.
10. Kopetz, H. & Bauer, G., 2003. The time-triggered architecture. In: Proceedings of the IEEE, 91(1), pp.112–126.