

Development of a Hardware-Software Co-Design for a real-time localization protocol for pedestrian safety

Axel Sikora, Dirk Lill, Manuel Schappacher

Zitiervorschlag im APA Stil:

Sikora, A., Lill, D., & Schappacher, M. (2012). Development of a Hardware-Software Co-Design for a real-time localization protocol for pedestrian safety. *Tagungsband zum Workshop der Multiprojekt-Chip-Gruppe Baden-Württemberg*, 47, 47–53. <https://nbn-resolving.org/urn:nbn:de:bsz:ofb1-opus4-60270>

Abstract

The research project Ko-TAG [2], as part of the research initiative Ko-FAS [1], funded by the German Ministry of Economics and Technologies (BMWi), deals with the development of a wireless cooperative sensor system that shall provide a benefit to current driver assistance systems (DAS) and traffic safety applications (TSA). The system's primary function is the localization of vulnerable road users (VRU) e.g. pedestrians and powered two-wheelers, using communication signals, but can also serve as pre-crash (surround) safety system among vehicles. The main difference of this project, compared to previous ones that dealt with this topic, e.g. the AMULETT project, is an underlying FPGA based Hardware-Software co-design. The platform drives a real-time capable communication protocol that enables highly scalable network topologies fulfilling the hard real-time requirements of the single localization processes. Additionally it allows the exchange of further data (e.g. sensor data) to support the accident pre-diction process and the channel arbitration, and thus supports true cooperative sensing. This paper gives an overview of the project's current system design as well as of the implementations of the key HDL entities supporting the software parts of the communication protocol. Furthermore, an approach for the dynamic reconfiguration of the devices is described, which provides several topology setups using a single PCB design.

Nutzungsbedingungen

Dieses Dokument wird unter diesen Bedingungen zur Verfügung gestellt:
Urheberrechtlich geschützt
Für weitere Informationen siehe:
<https://rightsstatements.org/page/InC/1.0/>

Kontakt

Hochschule Offenburg | Bibliothek
Badstraße 24
77652 Offenburg
Telefon: (0781) 205-240
E-Mail: bibliothek@hs-offenburg.de
www.hs-offenburg.de/bibliothek

Development of a Hardware-Software Co-Design for a real-time localization protocol for pedestrian safety

Axel Sikora, Dirk Lill, Manuel Schappacher

Abstract—The research project Ko-TAG [2], as part of the research initiative Ko-FAS [1], funded by the German Ministry of Economics and Technologies (BMWi), deals with the development of a wireless cooperative sensor system that shall provide a benefit to current driver assistance systems (DAS) and traffic safety applications (TSA). The system’s primary function is the localization of vulnerable road users (VRU) e.g. pedestrians and powered two-wheelers, using communication signals, but can also serve as pre-crash (surround) safety system among vehicles. The main difference of this project, compared to previous ones that dealt with this topic, e.g. the AMULETT project, is an underlying FPGA based Hardware-Software co-design. The platform drives a real-time capable communication protocol that enables highly scalable network topologies fulfilling the hard real-time requirements of the single localization processes. Additionally it allows the exchange of further data (e.g. sensor data) to support the accident prediction process and the channel arbitration, and thus supports true cooperative sensing. This paper gives an overview of the project’s current system design as well as of the implementations of the key HDL entities supporting the software parts of the communication protocol. Furthermore, an approach for the dynamic reconfiguration of the devices is described, which provides several topology setups using a single PCB design.

Index Terms—VRU eSafety, localization, system design, hardware-software co-design, time of flight, distance of arrival, driver assistance system.

I. INTRODUCTION

The development of traffic safety and driver assistance systems is a strong objective of the today’s automotive industry. Several techniques e.g. based on radar or ultrasonic waves or camera systems have been evaluated and enhanced. For these systems a

direct line of sight is a hard requirement for a proper operation.

The Ko-FAS initiative (Kooperative Sensorik und kooperative Perzeption für die Präventive Sicherheit im Straßenverkehr) was launched in 2009, and includes the described subproject Ko-TAG. The Ko-TAG project concentrates on the development of a cooperative sensor network between vehicles and vulnerable road users (VRU). Its objective is to add value to existing passive driver assistance systems such as radar- or camera-based systems. At the one hand, this requires to enable the localization of foreign targets even when losing line of sight or in scenarios with multiple objects requiring prioritization. On the other hand the cooperative sensor network provides valuable additional information about the localized target such as its movement parameters or even entire movement patterns that support the accident prediction algorithms at the higher layers. Since those networks in the Car2X environment are highly dynamic, the actual topology can change within short time periods regarding to the number of network participants and the reaction time of the system the different scenarios are handled by an underlying network communication protocol allowing a prioritization of connected devices and therefore affect the measurement update rate.

The protocol also keeps the security aspects such as the en-/decryption of user sensible data, data integrity and user anonymity. To ease a later integration into existing Car2X technologies or into other currently active projects dealing with this topic such as the sim^{TD} project [3], existing standards are considered during the system design.

The authors’ task in the project is to provide a scalable, stable and secure networking platform between vehicles and VRU’s that manages the different tasks of the localization process. This platform must meet the strong real-time requirements as well as it has to be flexible since the system is still at a development state. Therefore the authors follow an approach of a hardware-software co-design. The general approach of the Ko-TAG project and its communication and measurement protocols are described e.g. in [4] and [5].

Axel Sikora, axel.sikora@hs-offenburg.de; Dirk Lill, dirk.lill@stzedn.de; Manuel Schappacher, manuel.schappacher@stzedn.de are with Steinbeis Innovationszentrum für Embedded Design und Networking, Poststrasse 35, D-79423 Heitersheim, Germany

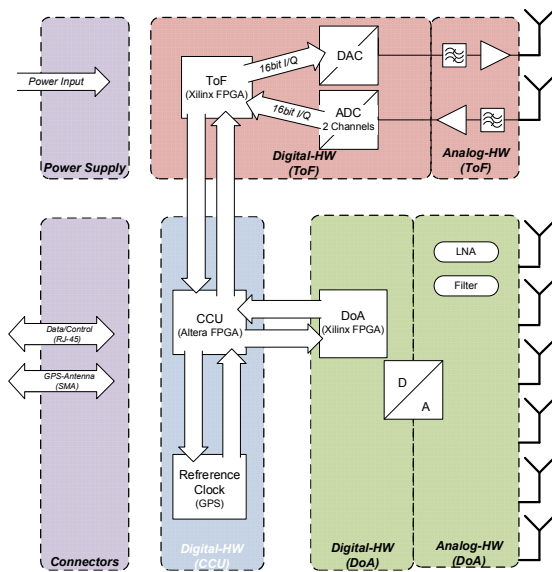


Figure 1: Architecture of the Localization Unit

II. SYSTEM ARCHITECTURE

The system architecture consists of several subcomponents to handle the single tasks of the localization and accident prediction process. The system and the underlying protocol design are built on two basic types of nodes. It is anticipated that vehicles get equipped with a Localization Unit (LU) that is able to communicate with as well as to localize SafeTAGs (ST) as their counterpart. An ST is a multifunctioning device that can change its role and behaviour in the network depending on the environment it shall be used in.

A. Architecture of the Localization Unit

The architecture of the LU is shown in Figure 1. It consists of

- A Time of Flight (ToF) system developed by [6], responsible for measuring the distances to the localized objects. This module is implemented in a Xilinx FPGA.
- A Destination of Arrival (DoA) system, developed by [7], to measure the elevation and azimuth to SafeTAGs implemented in a combination of a Xilinx FPGA and a DSP.
- The Communication & Control Unit (CCU), which handles the communication protocol as well as the coordination among the several sub-components.
- An Ethernet interface connecting the LU to a further component called the fusion unit (FU). The FU is an advanced board computer accepting the sensor and measurement data from the LU. Using this information optionally with data from further connected sensors such as cameras, the FU is able to calculate eventual collision risks. Depending on the determined risks, the LU can reconfigure

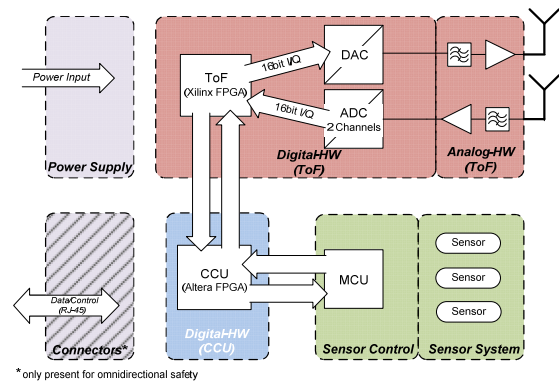


Figure 2: Architecture of the SafeTAG

the LU e.g. to priorities endangered SafeTAGs and therefore increasing their measurement update rate.

B. Architecture of the SafeTAG

The SafeTAG's architecture shows some differences compared to the LU since it needs less complexity.

- The ToF subsystem is less complex compared to the LU counterpart since here no measuring has to be performed. The messages are just reflected.
- Since the DoA measurements are performed using the electromagnetic waves of the data communication, the SafeTAG does not contain a DoA module. Instead it is connected to a sensor array developed by [7], which provides important sensor data, which support the accident prediction process.
- The SafeTAG also includes a CCU for the wireless protocol handling and for controlling the sub components.

Depending on the environment, the SafeTAG used in it provides further interfaces to interact with other Car2X elements. The architecture of the SafeTAG is given in figure 2.

III. COMMUNICATION & CONTROL UNIT

A. General Setup of the CCU

The design and the development of the CCU is one of the main tasks of the authors in the Ko-TAG project. The CCU handles the whole data communication including the establishment and configuration of the cooperative sensor network, as well as the exchange of the sensor and meta-data packets. During the communication the CCU has to take care of the security aspects of the system like the encryption/decryption of user sensible data or the integrity of the transmitted and received data. Additionally, the traceability of users has to be prevented.

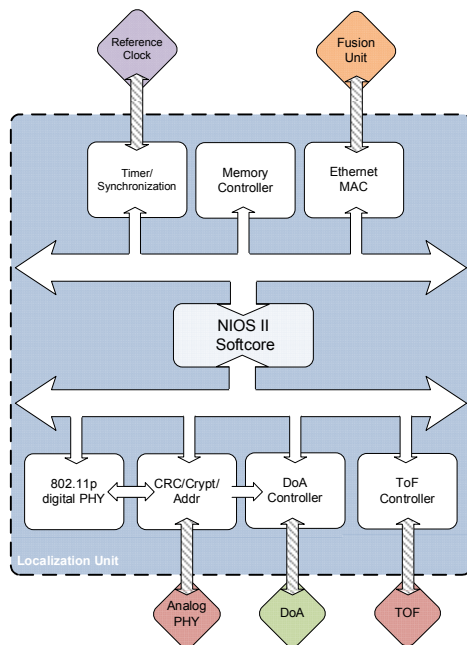


Figure 3: Architecture of the CCU of the localization unit

Besides the exchange of the data packets the single sub modules that take part in the localization process have to be controlled and coordinated by the CCU. To handle the different tasks described above, a hardware-software co-design was implemented in an Altera ARRIA II GX FPGA, including a NIOS II Soft-Core processor to run the appropriate firmware. To support the CPU and to handle the time critical parts such as synchronization, the design includes several application specific IP-Cores (Intellectual Property Core) implemented in VHDL. An overview of the CCU's architecture is shown in Figure .

B. NIOS II Soft-Core

At the centre of the system design a NIOS II Soft-Core is implemented in the FPGA description. The appropriate software primarily manages the Network. Highly scalable networks as in the Car2X environment require a lot of management effort as well as the maintenance of several tables and data structures. Since the system and the protocol design both still are in a development state, a software implementation provides the largest benefit regarding flexibility and efforts to meet future design changes. Furthermore, since all the HDL-modules developed in the project are implemented as Avalon [15] Bus-Slaves, several interfaces are provided to the software implementation allowing a dynamic configuration of the according hardware units.

C. Digital PHY

For the Physical Layer (PHY) of the data communication, an application specific digital PHY (dPHY) IP-

Core has been developed and provided [8] that takes care of the baseband modulation. It closely follows the IEEE 802.11p specification what eases the later integration into existing systems and additionally provides application specific features and interfaces to support the localization process. For example it allows seamless appending of a user defined signal sequence to standard 802.11p communication signals e.g. to ease the DoA estimation.

Besides a memory mapped register interface that allows the configuration and the monitoring of the PHY, an external 8 Bit wide FIFO interface gives fast access to its transmit and receive paths. Additional signal lines enable an interconnection to a self-developed Data Chain IP-Core described below.

On the PCB side, the dPHY is interconnected with an analog PHY (aPHY) designed and developed by [6] with a 16Bit wide IQ data interface and further configuration and control lines e.g. to switch the carrier frequency or to support the automatic gain control (AGC).

D. Data Chain

An important part in the design is a multi-functional IP-Core that is placed between the digital PHY's data interface and the CPU's system bus. It performs several tasks regarding the data communication that have been outsourced from software to a hardware description to increase the system performance.

1) Data Access

On top of the IP-Core the data chain implements an Avalon Bus-Master, which enables an autonomous direct memory access (DMA) to the system's included memory both in Tx and Rx mode. This discharges the CPU activity since the software routines for transmitting and receiving data packets equal to simple memory access functions. This gives an important benefit regarding time consumption and implementation effort compared to SPI-Interfaces used in most common transceiver chips.

Besides the general data, further flags indicating different packet states such as CRC or addressing errors or security information are included in the packets meta-information using a special buffer descriptor implementation on a per packet base. The software implements the specified driver similar to standard POSIX (Portable Operating System Interface) interfaces to access those buffer descriptors and therefore enables the lower layers to transmit and receive communication frames.

2) CRC Generation and Validation

To ensure the data integrity, a configurable CRC block is included into the data chain. Checksums are calculated for transmitted frames and verified for incoming packets automatically if configured. This

filters damaged packets and therefore saves CPU and bus time. Since the data chain block is connected to further IP-Cores managing parts of the localization process, it also prevents broken frames to be analysed for measurements.

The Data Chain driver allows the software to configure the CRC hardware block in different manners. Generally, the CRC generation or verification can be en- or disabled dynamically. Furthermore, broken frames can be discarded immediately or forwarded to the software e.g. for debugging purposes with an additional error flag set in the resulting buffer descriptors content.

3) Security & Privacy

The data chain also includes an autonomous en- and decryption block. Configured using the data chain's buffer descriptor interface e.g. regarding key information, incoming packets get decrypted and transmitted packets get encrypted without a further user interaction from software during the process. It is anticipated to use an AES crypto-unit here for providing security and privacy. Since the security design is still subject of on-going development it will be described in a future contribution.

4) Address Checking

Incoming packets will be checked automatically for valid addressing and configured before the reception from software via register accesses. This unloads the system bus as well as the CPU since on the one hand, the address check has not to take place in software and on the other hand, misaddressed frames do not reach the system bus.

The hardware address checking is also important for the DoA estimation process since this is a highly time critical operation. Therefore the data chain is also interconnected to the DoA-Controller via an additional interface to filter unwanted packets from further analysis.

Since the addressing in the communication protocol is not fixed and depends on the according frame type, the HDL entity has to provide several configuration options. Here the data chain driver allows the software configuring several address fields together with their length and their offsets in the communication frames to support an address validation of the several frame types.

E. DoA Controller

The DoA controller serves as interconnection between the CCU and the DoA estimation module. The Controller handles exchange of measurement and control frames as well as it triggers and configures new measurements.

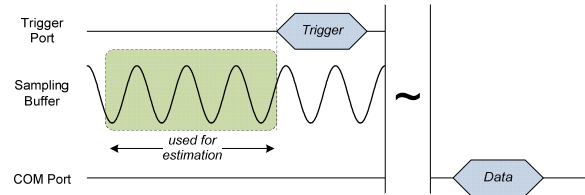


Figure 4: DoA Sampling and Trigger Behaviour

The DoA estimation takes place over regular data packets transmitted during the DoA phase of the network protocol. Through the underlying time slotted channel access mechanism during this phase, the address of the expected data frame is well known. After a successful check of the data integrity and the address information of the received package, which is all done in hardware by the data chain and without software interaction, a new DoA estimation gets triggered by the DoA-Controller. Since the answer to the measurement request is asynchronous, an internal ID, previously defined from software access, is included in the trigger to allow an exact identification of the returned values on reception.

The DoA component itself includes a ring buffer which samples the signals on the communication channel constantly. Once the DoA unit receives a trigger from the DoA-Controller, it analyses the specified window of the data in the sampling buffer for an angle estimation. To get valid measurement results, the offset between the start of the frame and the according trigger needs to be static to guarantee that the correct signal pattern gets used for the following calculation as shown in Figure .

Once the calculation has finished, the DoA module provides several value pairs including their qualities and the device ID of the measurements are associated to via a serial interface to the DoA-Controller. The Software finally collects the data from the DoA Controller and passes it to the LU for further analysis.

F. ToF Controller

The ToF-Controller manages the ToF hardware developed by [6]. Before the start of a new ToF phase it configures the ToF hardware before it triggers a new measurement. The configuration primarily specifies addressing information, which gets included into the following ToF beacon. This additional information is needed for the ST's to identify the vehicle the burst was sent from, as well as for the vehicle for a proper correlation of the ToF replies sent by STs.

During the ToF localization process the ToF-Controller serves as a clock source to indicate the single time slots since their offsets have to be regarded for the appropriate distance calculation. Furthermore, the Controller receives continuous measure data containing the calculated distance and measurement quality via a 16 bit wide parallel interface that have to be associated to the correct device before they get for-

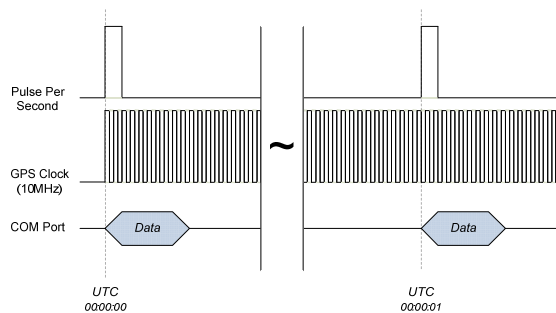


Figure 5: Timing diagram of the GPS synchronization

warded to the FU. To allow a later collection of the measured values at the software layers, an additional FIFO holds the data, together with the according slot indices until the end of the measurement procedure.

G. Synchronization

Since the network protocol uses a time slotted channel access mechanism to provide a deterministic behaviour for time critical processes, the LU's must be synchronized among each other to provide a common grid of the slots to the ST's. A separate synchronization/timer IP-Core takes care about the global synchronization using an external reference clock as time input source to keep given guard times.

Therefore the Ko-TAG project uses a high precision timing GPS module of the u-blox LEA-6T family. The LEA-6T module provides the needed performance regarding the accuracy with an error below 60 ns [9]. Using valid almanac data stored from previously established satellite connections, software can help to accelerate the device start-up procedure and a start-up time of around 1 s can be reached. The module also provides several additional interfaces that finally feed the synchronization IP-Core.

The GPS timing module provides two separate time pulse outputs. One output gives a GPS locked pulse per second (PPS). The second output is a configurable clock with a frequency up to 10 MHz. A combination of these pulse signals and an additional serial protocol implemented in the GPS module enables the IP-Core to synchronize with a maximum error less than 100 ns among other LU's. This synchronization permits the ST's to derive their time slot information from any LU. Since ST's are not equipped with an additional reference clock source they synchronize on the network using the ToF beacons sent by vehicles.

As shown in Figure , the GPS timing module used in the project triggers the start of a new second using the PPS strobe followed by a serial NMEA (National Marine Electronics Association) packet indicating the current time. The time information is interpreted by software and is used to configure the IP-Core via register accesses. In its actual configured state the hardware block uses the following PPS trigger to

synchronize according to the previously set configuration. From now on, the GPS locked 10 MHz clock signal is used for the CCUs internal time reference. Once synchronized the IP-Core can be monitored and configured using further register accesses. It provides several hardware timers/counters with a common time reference among all synchronized LU's.

H. Ethernet Interface

The CCU includes a GBIT-Ethernet MAC IP-Core developed by the authors' team and presented in [13]. This interface is used in different manners. According to Figure the CCU is connected with a fusion unit (FU) via an Ethernet interface. This interface is used to provide measurement results and data from the sensor network to the FU as well as to receive control information. The interface uses an extended LocON protocol [15] at application layer.

A further benefit of the Ethernet connection is the possibility to give the user simple access to the system. The user is able to set static configurations such as the wireless network parameters and to observe the current device status. Furthermore, an interface is given to update the device software as well as the FPGA image. An embedded software TCP/IP stack [12] provides the TCP/IP interconnection on the higher layers including an embedded web server.

IV. SOFTWARE DESIGN

Besides the HDL entities described in chapter III, a firmware was designed and implemented primarily to manage the network and to control and configure the several IP-Cores. Generally, it consists of a wireless communication stack, the application itself and further interface drivers and protocol implementations depending on the actual device type. The main tasks of the software are handled in a wireless communication stack that currently covers the physical (PHY), the data link (DLL) and the application (APL) layers.

At the lower layers e.g. the software part of the PHY, the stack is responsible for the transmission and reception of the communication frames. As described in chapter III, access to frames in Tx- as well as in Rx-mode is implemented through memory access functions wrapped into buffer descriptor interfaces via the data chain IP-Core. Therefore the physical layer software provides the device driver according to access the data chain. Furthermore, the physical layer implements the required driver elements to configure the dPHY, e.g. in its channel and operation mode.

The data link layer (DLL) represents the most complex part of the software since it has to manage and synchronize the complex processes of the network and interacts with several of the IP-Cores described above. E.g. it configures the data chain for address and CRC validation and feeds the security block with different en/decryption keys depending on the communication

partner. The most significant part of the DLL represents the medium access. Since parts of the protocol use a time slotted channel access it also has to take care of a valid distribution and coordination of the single slots. Here it uses interrupts generated by the synchronization IP-Core's timer to meet the timing requirements and to hit the specific time slots within a given guard time.

The APL has its key task at the OBU. Here it is primarily responsible to accept user prioritization requests from the LU. Internal prioritization queues are fed with those requests and enable the APL to configure specific channel access phases of the DLL. This allows a good scalability of the measurement update rates of the single users. I.e. it allows to reserve channel resources for the mostly endangered devices.

V. DEVICE CONFIGURATION

During the development process, the single components are very expensive. To provide the possibility of evaluating different network topologies despite of the limited resources, an approach of a single PCB design for the LU and the SafeTAG was chosen. This makes it possible to run and test different network setups without changing the underlying PCB.

The role of the device only depends on the running hardware/firmware combination and can be changed at start-up or even during runtime. To have multiple roles available at a single device, the included flash memory contains several combinations of an FPGA-image and its according software, located in specific areas of the target memory that can be loaded at the device start-up or during runtime. For this task several approaches exist:

- The device can be configured using an EPCS device, an active serial EEPROM chip that configures the connected FPGA device at start-up. Usually these devices are limited in their memory size.
- A common method is the configuration via an additional smaller CPLD and an external memory chip. This approach allows the usage of a custom loader firmware, which allows modifying the boot process.
- An almost identical approach uses a microcontroller instead of the CPLD, which helps to reduce costs.

The CCU PCB is designed to support a CPLD configuration as well as the microcontroller approach using the fast passive parallel configuration (FPP) method supported by devices of the Arria II family. The design of the according configuration mechanism is shown in Figure . To support the configuration as different device roles the firmware of the configuration chip accesses a special boot loader section in the external memory device before initiating the configuration process. This section includes information about the base addresses of the different images in memory and provides the possibility to choose the according

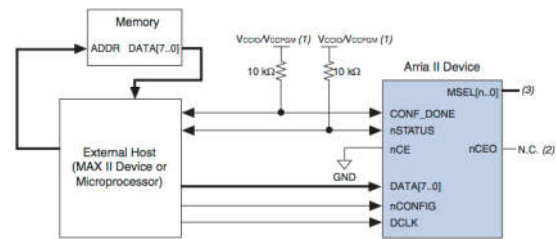


Figure 6: Single Device FPP Configuration Using an External Host [10]

firmware image at start-up time using a DIP panel or a dynamic reconfiguration during runtime.

Since the target devices will be integrated into vehicles at a later development state and thus won't be accessible, a web interface provides easy access to system update and supervision.

VI. SUMMARY AND OUTLOOK

The approach of the hardware-software co-design allows the swapping of time-critical components and processes into fast and parallel FPGA implementations. Nevertheless the system remains flexible and configurable by letting software modules handle the complex task such as the network management.

The elements of the system architecture already have been separately tested and proved for their functionality. Currently the integration of the sub components is in progress and will be followed by a complete system test.

ACKNOWLEDGEMENT

This work results from the joint project Ko-TAG, which is part of the project initiative Ko-FAS and is partially funded by the German Ministry of Economics and Technologies (BMWi) under contract number 19S9011. The authors are grateful for this support and for the excellent teamwork in the consortium, which consists of BMW Group Forschung und Technik, Continental Safety Engineering International GmbH, Daimler AG, Fraunhofer IIS, TU Munich, and Steinbeis Innovation Centre Embedded Design and Networking.

REFERENCES

- [1] Forschungsinitiative Ko-FAS - Kooperative Sensorik und kooperative Perzeption für die Präventive Sicherheit im Straßenverkehr, <http://www.kofas.de/>, 19.02.2012.
- [2] Forschungsprojekt Ko-TAG – Kooperative Transponder, <http://ko-fas.de/deutsch/ko-tag---kooperative-transponder.html>, 19.02.2012.
- [3] Projekt sim^{TD} (Sichere Intelligente Mobilität Testfeld Deutschland), <http://www.simtd.de>, 19.02.2012.
- [4] D. Lill, M. Schappacher, A. Gutjahr, A. Sikora, "Protocol and System Design of a Cooperative Pedestrian Safety System", 11th Int'l Conference on ITS Communication (ITST2011)", 23.-25.8.2011, St. Petersburg (Russia).

- [5] D. Lill, M. Schappacher, S. Islam, A. Sikora, "Wireless Protocol Design for a Cooperative Pedestrian Protection System", 3rd International Workshop on Communication Technologies for Vehicles (Nets4Cars 2011), 23.-24.3.2011, Oberpfaffenhofen.
- [6] Technische Universität München, Fachgebiet Höchstfrequenztechnik, <http://www.hot.ei.tum.de/>
- [7] Fraunhofer-Institut für Integrierte Schaltungen IIS, <http://www.iis.fraunhofer.de/>, 19.02.2012.
- [8] Fraunhofer Heinrich-Herz Institut, <http://www.hhi.fraunhofer.de/>, 19.02.2012.
- [9] u-blox, LEA-6, u-blox 6 GPS modules Data Sheet, http://www.u-blox.com/images/downloads/Product_Docs/LEA-6_DataSheet_%28GPS.G6-HW-09004%29.pdf, 19.02.2012.
- [10] Configuration, Design Security and Remote System Upgrades in Arria II Devices, A11GX51009-4.2, Altera Cooperation December 2011.
- [11] Experteninformationen und Publikationen aus Ko-TAG, <http://ko-fas.de/deutsch/ko-tag--kooperative-transponder/experteninformationen.html>, 19.02.2012.
- [12] N. Braun, A. Sikora, M. Colling, "High Performance Embedded Ethernet and Internetworking", embedded world 2005 Conference, Nürnberg, S.990-997.
- [13] A. Rohleder, S. Jaeckel, A. Sikora, "Design and Test of a Gigabit Ethernet MAC for High-Speed HIL-Support", XLIV. Workshop MPC-Gruppe, Furtwangen, 9.7.2011.
- [14] <http://www.ict-locon.eu/>, 19.02.2012.
- [15] Avalon Interface Specifications, Altera Cooperation, May 2011.



Manuel Schappacher studied Computer Engineering at the University of Applied Sciences, Furtwangen and received his Dipl.-Inform. degree in April 2009. After that, he continued to work as project engineer at Steinbeis Innovation Center Embedded Design and Networking (sizedn) mainly in the field of embedded wireless and wired communication, including simulation of networking protocols.



Dipl.-Ing. (FH) Dirk Lill holds a Diploma of University of Cooperative Education, Loerrach, where he studied mechatronics at the Universities in Loerrach (Germany), Muttenz (Switzerland) and Mulhouse (France). He joined Steinbeis Innovation Centre for Embedded Design and Networking (sizedn) in 2002. Since then, he has worked in various projects in the fields of wired and wireless embedded networking, including standard protocols and proprietary wireless protocol design. He is now deputy head of stzedn. He combines excellent experience in PCB hardware design for HF-circuitry with a deep knowledge of communication protocols and their implementation in hard- and software



Axel Sikora holds a diploma of Electrical Engineering and a diploma of Business Administration, both from Aachen Technical University. He has done a Ph.D. in Electrical Engineering at the Fraunhofer Institute of Microelectronics Circuits and Systems, Duisburg, with a thesis on SOI-technologies. After various positions in the telecommunications and semiconductor industry, he became a professor at the Baden-Wuerttemberg Cooperative State University Loerrach in 1999. In 2011, he joined Offenburg University of Applied Sciences, where he holds the professorship of Embedded Systems and Communication Electronics. His major interest is in the field of efficient, energy-aware, autonomous, and value-added algorithms and protocols for wired and wireless embedded communication.

He is founder and head of Steinbeis Innovation Center Embedded Design and Networking (sizedn). Dr. Sikora is author, co-author, editor and co-editor of several textbooks and numerous papers in the field of embedded design and wireless and wired networking, and head and member of numerous steering and program committees of international scientific conferences.