A Multicore Implementation of an AUTOSAR-based XCP Module

Mahmoud Osama
Dept. of Computer & Systems Eng.
Faculty of Engineering
Ain Shams University
Cairo, Egypt
2101394@eng.asu.edu.eg

Sherif Hammad
Garraio, CEO
Cairo, Egypt
sherif.hammad@garraio.com

Nancy E. ElHady
Dept. of Mechatronics Eng.
Faculty of Engineering
Ain Shams University
Cairo, Egypt
nancy_emad@eng.asu.edu.eg

M. Watheq El-Kharashi
Dept. of Computer & Systems Eng.
Faculty of Engineering
Ain Shams University
Cairo, Egypt
watheq.elkharashi@eng.asu.edu.eg

2023 12th Mediterranean Conference on Embedded Computing (MECO) | 979-8-3503-2291-0/23/$31.00 ©2023 IEEE | DOI: 10.1109/MECO58584.2023.10154903

***Abstract*—The automotive industry depends on computers to control and monitor vehicles behaviour. The Universal Measurement and Calibration Protocol (XCP) connects calibration systems to electronic control units (ECUs). Nowadays, AUTOSAR-based XCP for single core ECUs is widely used in automotive industry. As the demand for computing power is rapidly increasing in the automotive domain, car manufacturers are gradually introducing multi-core ECUs in their electronic architectures. Multi-core ECUs provide high level of parallelism. This research work aims to deliver a multi-core implementation of an AUTOSAR-based XCP module, as it is important to have an efficient process of measurement and calibration on multi-core targets. The paper results are ensuring data consistency on the application level, avoiding race conditions between writer core and reader core which might result in processing rubbish data, and decreasing data dependency between cores.**

***Index Terms*—AUTOSAR, master core, master tool, multi-core, satellite, XCP**

I. Introduction

The automotive industry relies heavily on electronic control units (ECUs) to monitor and control vehicle's behavior, and as such, the demand for computing power in this industry is increasing [1]. The Universal Measurement and Calibration Protocol (XCP) has become a widely used method for connecting calibration systems to ECUs in the automotive industry [2]. However, traditional XCP modules based on single-core ECUs may not be efficient on multi-core platforms, which are being increasingly introduced in electronic architectures [3]. XCP Protocol is based on CAN Calibration Protocol (CCP), in which 'X' generalizes multiple transport layers not only CAN.

XCP is used to gain read/write access to the memory of a controller, the access to the measurement and calibration variables is address-oriented. The design of the XCP protocol is master-slave, the master is a software tool which is used to give commands to the slaves, the slaves are software modules that run on ECUs. The ordinary XCP module is based on one software running on the main core of the slave, which may lead to data inconsistency, race condition between different cores, and high data dependency between cores, which leads to misbehaviour of other cores during the measurement and calibration session, the multi-core XCP is based on the master satellite concept, in which there are dedicated shared buffers per core, data are initially processed with the slave core then the master core accesses the shared buffers instead of the direct access of memory.

In recent studies, researchers have discussed the challenges of implementing XCP on multi-core ECUs and presented solutions for ensuring data consistency, avoiding race conditions between cores and decreasing data dependency between cores by minimizing the shared memory locations. Baumgartner, Grössing, Hammacher, and Stöckert presented the challenges and solutions in the implementation of XCP module on multi-core ECUs in their paper [4]. Fiebig and Gliwa also discussed the challenges of implementing XCP on multi-core ECUs over CAN in their paper [5]. Jurczyk and Gliwa evaluated the performance of the multi-core implementation of the XCP protocol over Ethernet in their paper [6].

Our study builds upon the work of these previous researchers to present a comprehensive solution for implementing XCP on multi-core ECUs that is compatible with both CAN and Ethernet. By bringing together the insights of previous studies, we have demonstrated the compatibility of multiple standards and protocols offering a more complete solution for automotive ECUs.

To this end, the contributions of this paper can be summarized as following;

1) Presenting the combination of XCP protocol with AUTOSAR standard and master-satellite design concept. The supported features in the discussed XCP are measurement, calibration, data acquisition, stimulation,

979-8-3503-2291-0/23/$31.00 ©2023 IEEE

Authorized licensed use limited to: The University of British Columbia Library. Downloaded on March 06,2025 at 01:42:33 UTC from IEEE Xplore. Restrictions apply.

block mode, and security mechanisms using seed and key.
2) Discussing how traditional XCP modules based on single-core ECUs may not be efficient on multi-core platforms.
3) Discussing the design and implementation of the multi-core AUTOSAR XCP module, highlighting its advantages over the traditional XCP module.
4) Discussing the design and implementation of the master tool and the environment used to test the XCP module.
5) Focusing on the compatibility of implementing a multi-core AUTOSAR-based XCP module over both CAN and Ethernet, and presents a comprehensive solution for implementing XCP on multi-core ECUs that is compatible with both protocols.

The remaining contents of this paper is organized as following. Section II highlights the literature review on XCP protocol, AUTOSAR and Multi-core systems. Section III discusses the methodology and design of the multi-core AUTOSAR XCP module, highlighting its advantages over the traditional XCP module. Section IV presents the results of our experiment and demonstrate the effectiveness of the multi-core XCP module. Section V presents the master tool implementation results, highlighting its use in the environment. Section VI presents the paper's conclusion regarding the XCP system development and the future work.

II. LITERATURE REVIEW

The literature review is divided into four main parts, which are the XCP protocol, AUTOSAR standard, combining AUTOSAR with XCP, and master satellite design concept.

A. The XCP Protocol

The universal measurement and calibration protocol family or XCP specification has expanded the limits of the original CAN Calibration Protocol or CCP. CCP has proved the usefulness of having reusable software for measurement and calibration. The XCP specification no longer limits users to using CAN as the transport media. The XCP specification has been expanded to allow for ease of use, not only in Electronic Control Units (ECUs) but also in measuring instruments [7].

XCP allows for the measurement and calibration of parameters such as sensor data and control signals, and supports a variety of communication protocols, including CAN and Ethernet. The XCP standard is maintained by the Association for Standardization of Automation and Measuring Systems (ASAM) [8].

B. AUTOSAR

A major problem of the developed software or hardware modules in the automotive industry is a poor reusability because of the high dependency between the OEMs (Original equipment manufacturer) and suppliers. Also, the fact that embedded systems do not support the full HW abstraction or limited modularity. This means SW must be rewritten from scratch when the HW is changed or a change in the ECU functionality requires at least SW change in other modules. For this, AUTOSAR consortium was introduced in 2004 [9]. The AUTOSAR Classic Platform is a software platform with a layered software architecture defined by AUTOSAR which is used for deeply embedded systems and application software with high requirements for predictability, safety, security and responsiveness [10].

AUTOSAR-based XCP is a version of XCP that is designed to work with AUTOSAR-compliant ECUs. This version of XCP has several advantages, including improved interoperability, flexibility, and scalability. Several studies have been conducted on the implementation of AUTOSAR-based XCP, highlighting its effectiveness in managing ECU parameters and improving performance [11].

C. Combining AUTOSAR with XCP

XCP in combination with an AUTOSAR compliant software basis enables flexible instrumentation and testing of an ECU during the whole process from software development at the beginning to acceptance testing at the end. The provided techniques help to simplify instrumenting an ECU for debugging and testing purposes during the whole lifetime. The on-the-fly configuration of XCP is additionally reducing the overhead of preparing an ECU to special test cases and thus saving valuable time [12]. Several studies have been conducted on the combined use of XCP, AUTOSAR and multi-core master-slave architectures in the automotive industry. These studies have focused on improving the performance and efficiency of XCP on multi-core ECUs. Also, addressing challenges such as data consistency, race conditions and decreasing data dependency between cores by minimizing the shared memory locations. The combined use of these technologies has shown promising results with improving data acquisition, measurement and calibration [13].

III. METHODOLOGY

The XCP was designed to be a combination of multiple features, support multiple communication protocols and based on multiple concepts. The efficiency and flexibility were taken in consideration to give the value of the multi-core master-satellite concept. The proposed XCP design was developed using a combination of top-down and bottom-up design approaches. The top-down approach was used to define the high-level architecture and requirements of the XCP system, while the bottom-up approach was used to implement the individual features and functions of the XCP system. The development of the XCP system was divided into several phases.

A. Define the Features

In the first phase, the communication protocols, concepts, and features which were required to be included in the system were defined. Measurement, calibration, data acquisition, stimulation, block mode, and security mechanisms using seed and key were supported in the system. The measurement is reading a certain value from the slave controller's memory.

Authorized licensed use limited to: The University of British Columbia Library. Downloaded on March 06,2025 at 01:42:33 UTC from IEEE Xplore. Restrictions apply.

The calibration is overwriting a certain value in the slave controller's memory. Data acquisition is similar to measurement but a list of elements are read at the same time according to certain event, which is better than measurement in data consistency. Stimulation is similar to calibration but a list of elements are written at the same time according to a certain event, which is better than calibration in data consistency. Block mode is a mode which allows responding on multiple frames with single response. The seed and key method is available for checking whether or not a connection attempt is authorized. The CAN and Ethernet were supported in the system as communication protocols. The CAN protocol was used for low-speed communication, while Ethernet was used for high-speed communication. The AUTOSAR standard was used to ensure compatibility with other systems in the vehicle.

B. Detailed Design

In the second phase, the detailed design of the XCP system was performed. This included the implementation of the XCP protocol, which was used for communication between the master tool and slaves.

To ensure that the XCP system would be compatible with other systems in the vehicle, it was designed to adhere to the AUTOSAR standard. This ensured that the XCP system could communicate with other systems in the vehicle and could be integrated seamlessly into the overall vehicle system.

Finally, the multi-core master-satellite design concept was used to ensure that the XCP system is optimized for multi-core architectures. The master-satellite design ensured that the XCP system can take advantage of the performance and scalability benefits of multi-core systems, while also ensuring that it can handle the complexity of multi-core architectures.

C. Implementation and Testing

In the final phase, the XCP system was implemented and tested. The software components were developed using C programming language, and the hardware components were simulated using C++ tools and Linux OS. The XCP system was tested using various scenarios to ensure that it would meet the specifications, the system was tested to ensure that it would function correctly and would meet the requirements of the vehicle system. Any issues or bugs that were found had been fixed.

IV. XCP MODULE IMPLEMENTATION RESULTS

The developed XCP system successfully implemented the required features and supported the CAN and Ethernet communication protocols. The XCP system was designed to be efficient and flexible, allowing it to take advantage of the performance and scalability benefits of multi-core systems. The new design has overcome the problems which appeared in the multi-core targets with old XCP design.

The problems which have faced the single core implementation are processing rubbish data due to simultaneous access, and inconsistency of the data being processed. Fig. 1 shows the simultaneous accessing by the master and satellite core of the satellite core's dedicated memory.

The XCP protocol was implemented and tested using a software environment. The environment used is based on simulating part of the AUTOSAR stack which runs on vehicles. This includes stubs for all the modules which interact with the XCP module such as the operating system, transport layer modules and also stubbing the memory itself, it was found to be reliable and efficient.

To ensure compatibility with other systems in the vehicle, the XCP system was designed to adhere to the AUTOSAR standard. This ensured that the XCP system could communicate with other systems in the vehicle and that it could be integrated seamlessly into the overall vehicle system.

The multi-core master-satellite design concept was used to optimize the XCP system for multi-core architectures. The master-satellite design ensured that the XCP system could take advantage of the performance and scalability benefits of multi-core systems while also handling the complexity of multi-core architectures.

The master-satellite implementation was done as following, there is a Master core which has the communication stack integrated to it, this core is responsible to handle all XCP protocol requests and responses, which is similar to the old XCP implementation. The new part is that there is event collector runnables per each core, each runnable is used to handle the core's specific memory access operation. Also, there are shared buffers, one buffer per DAQ list which are used as intermediate buffers between the master and satellite cores. The buffers have locking mechanism, which prevents any race conditions and decreases data dependency between cores.

The flow of data in DAQ (data acquisition) is shown in Fig. 2. The satellite core reads the data from the memory, then it packs them into ODTs (Object Description Tables). The satellite core checks the master buffer's lock, if it is not

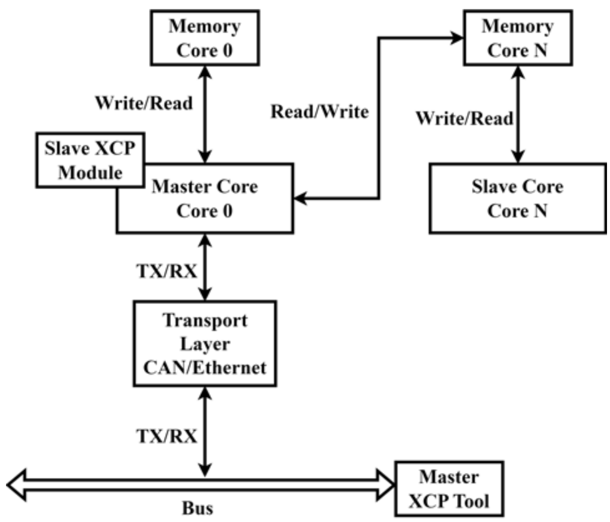

Fig. 1. Single Core XCP diagram.

Authorized licensed use limited to: The University of British Columbia Library. Downloaded on March 06,2025 at 01:42:33 UTC from IEEE Xplore. Restrictions apply.

locked, it locks the slave buffer's lock, then it writes the ODTs to the buffer. Finally, it releases the slave buffer's lock. On the other side, the master core checks the slave buffer's lock, if it is not locked, it locks the master buffer's lock. Then, the master core starts reading the ODTs, converts them to DTOs (Data transfer objects) and sends them on the bus. Finally, it releases the master buffer's lock.

The flow of data in STIM (stimulation) is opposite to the flow in DAQ, it is shown in Fig. 3. The DTOs are received by the master core. The master core decides which DAQ list and event the received DTOs belong to. Then, the master core checks the slave buffer's lock. If the slave is using the buffer or not, if it is not locked, the master locks the master buffer's lock, the buffering of the ODTs takes place. Finally, it releases the master buffer's lock. On the other side, the Satellite core then checks master buffer's lock, if it is not locked, the slave locks the slave buffer's lock. Then, it starts reading from the buffer and writing to the memory. Finally, it releases the slave buffer's lock.

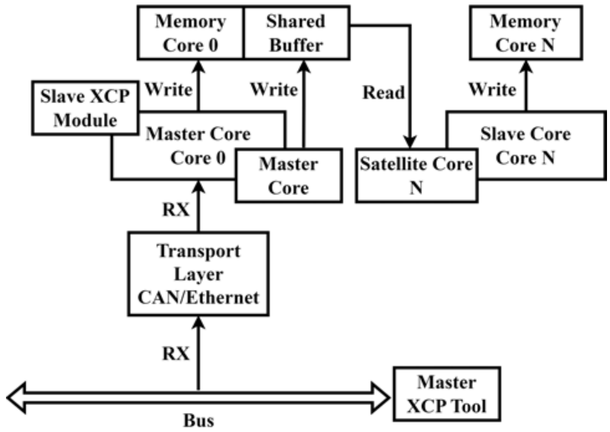

Fig. 3. Multi-core XCP while making stimulation diagram.

V. MASTER TOOL IMPLEMENTATION RESULTS

In addition, a master tool was developed to be used in the environment to communicate with the XCP module and test the master tool support all the required features. The required features include measurement, calibration, data acquisition, and stimulation. The tool is capable of parsing A2L files and extract the measurement variables, calibration variables, lists info and events info. Also, the tool can communicate with the slave using Ethernet. It can lock and unlock features. It can connect and disconnect from the slave. It is capable of drawing diagrams for the measurement variables.

The tool was used to implement complicated test scenarios and test the XCP module to make sure all the features work together. Fig. 4 shows the master tool while handling two simultaneous features at the same time, which are data acquisition and calibration, while the multi-core feature is turned on. This shows the execution after successfully parsing the A2L file, connecting to the slave and unlocking the data acquisition feature. The calibration takes place by manipulating certain variable value. The data acquisition is used to read a group of elements including the variable which is being manipulated. The value of the calibration variable is plotted to show the modifications which took place on it versus the time.

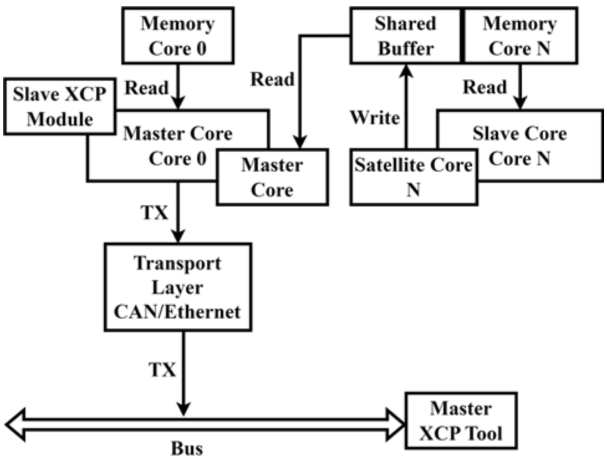

Fig. 2. Multi-core XCP while making data acquisition diagram.

VI. CONCLUSION

The paper introduced the combination of XCP protocol with AUTOSAR standard and master-satellite design concept. We have explored the concepts and features discussed in the previous studies and demonstrated the feasibility of combining them in one product. The XCP system developed in this paper had successfully implemented the required features and supported the CAN and Ethernet communication protocols. The XCP protocol used for communication between the master and satellite cores was optimized for performance, memory usage, and scalability. The XCP system was tested using various scenarios, and it was found to be reliable and efficient. The system was designed to adhere to the AUTOSAR standard and optimized for multi-core architectures using the master-satellite design concept.

The XCP system's ability to support multiple communication protocols (CAN and Ethernet) and the multiple features (measurement, calibration, data acquisition, and stimulation) made it a versatile and adaptable system. The system could handle different use cases and operating conditions and could be customized to fit specific vehicle system requirements.

In conclusion, the XCP system's development presented in this paper provides a comprehensive approach to the development of an efficient, flexible, and optimized communication protocol for modern vehicle systems. The XCP system's success in meeting the requirements of modern vehicle systems ensures its value in improving the reliability and efficiency of communication between electronic components and systems.

Future work can focus on expanding the XCP system's capabilities to include additional features and support for other communication protocols. Further research can also investigate the XCP system's performance in more complex scenarios,

Authorized licensed use limited to: The University of British Columbia Library. Downloaded on March 06,2025 at 01:42:33 UTC from IEEE Xplore. Restrictions apply.

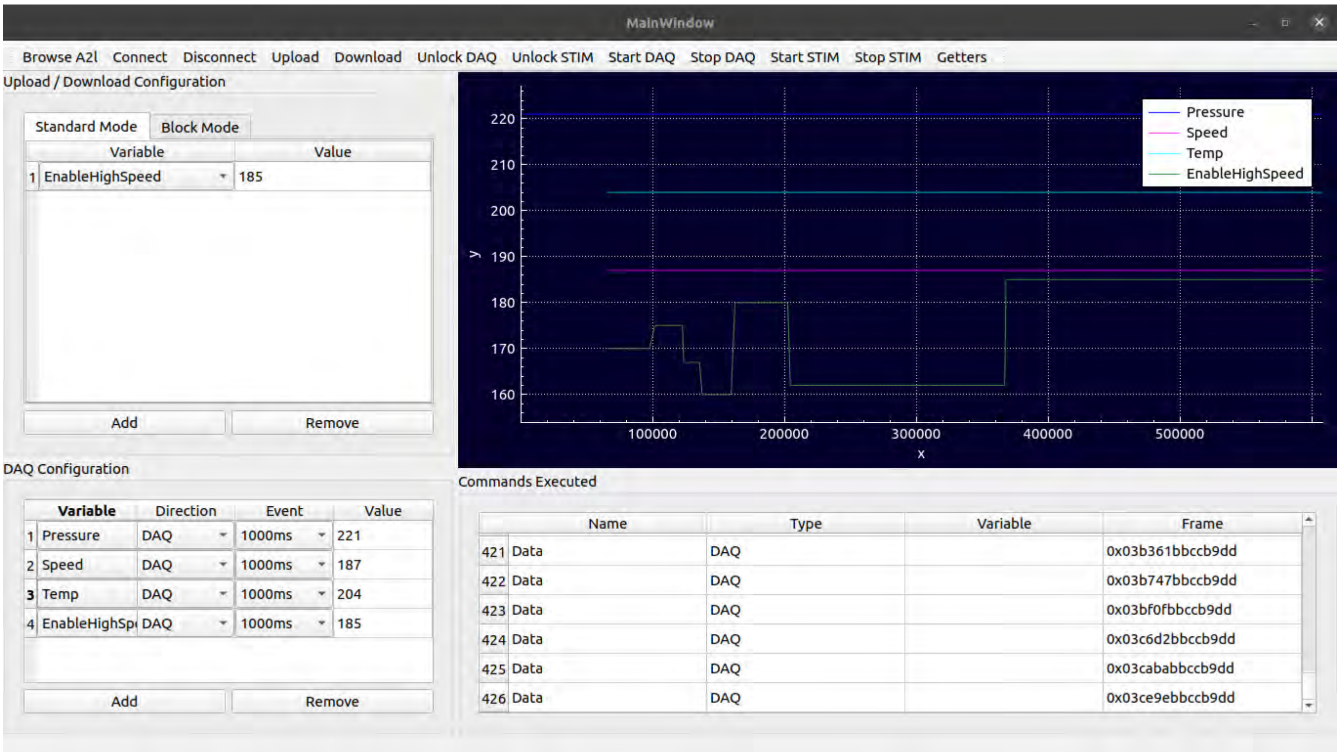

Fig. 4. Data acquisition while making calibration to one of the measured variables at the same time using Multi-core XCP.

such as in autonomous vehicles or in electric vehicles, which have different operating conditions and requirements.

Acknowledgment

We wish to thank Garraio for their financial support.

References

[1] M. Al-Hasan, S. Al-Majed, and A. Al-Awadhi, "A comprehensive survey of automotive software architectures," Journal of Systems Architecture, January 2016.

[2] Vector Informatik GmbH, "Universal Measurement and Calibration Protocol for ECUs (XCP)."

[3] L. Li, Y. Li, X. Huang, and Q. Li, "A survey of multi-core processing in automotive systems," Microprocessors and Microsystems," January 2019.

[4] M. Baumgartner, T. Grössing, C. Hammacher, and G. Stöckert, "XCP on multicore ECUs - challenges and solutions," IEEE Transactions on Vehicular Technology, March 2016.

[5] T. Fiebig and P. Gliwa, "XCP on multicore ECUs," IEEE International Conference on Industrial Technology (ICIT), March 2015.

[6] M. Jurczyk and P. Gliwa, "A multicore implementation of the XCP protocol for measurement and calibration in automotive ECUs," IEEE International Conference on Industrial Technology (ICIT), Taipei, Taiwan, March 2016.

[7] K. Lemon, "Introduction to the universal measurement and calibration protocol XCP," SAE 2003 World Congress & Exhibition, March 2003.

[8] ASAM e.V., "The Universal Measurement and Calibration Protocol, ASAM Standard, Version 1.8.3," October 2021.

[9] C. Briciu, I. Filip and F.Heininger, "A new trend in automotive software: AUTOSAR concept," Applied Computational Intelligence and Informatics (SACI), May 2013.

[10] AUTOSAR, "AUTOSAR - AUTomotive Open System ARchitecture."

[11] S. Thapa and V. P. Namboodiri, "Design and implementation of AUTOSAR-compliant XCP module," International Conference on Circuit, Power and Computing Technologies (ICCPCT), March 2017.

[12] P. Luchscheider, C. Lauer and R. German, "Flexible integration testing of automotive ECUs by combining AUTOSAR and XCP," Magnetism, December 2011.

[13] J. Zhang, B. Wei, J. Chen, S. Wang, and Y. Tian, "A multicore-based calibration and measurement module for electric vehicle controller based on XCP protocol," IEEE 3rd International Conference on Intelligent Transportation Engineering (ICITE), August 2018.

Authorized licensed use limited to: The University of British Columbia Library. Downloaded on March 06,2025 at 01:42:33 UTC from IEEE Xplore. Restrictions apply.