Proceedings on Automation in Medical Engineering
Proc AUTOMED, Vol 2, No 1 (2023): Article Ref 724
www.journals.infinite-science.de/automed/article/view/724

ASMO: a decentralized and verifiable interoperability platform in intensive care

F. Berg[1*], M. Wiartalla[1], M. Hüllmann[1], A. Derks[1], S. Kowalewski[1] and A. Stollenwerk[1]

[1] Informatik 11 - Embedded Software, RWTH Aachen University, Aachen, Germany
[*] Corresponding author, email: berg@embedded.rwth-aachen.de

Abstract: Interconnected medical devices enable new therapies and automate existing ones. Due to various manufacturers and interfaces, interoperability needs to be enabled with the help of auxiliary hardware. Since functional safety is indisputably critical, verifiability is essential, which is often neglected by state-of-the-art medical hardware platforms. We propose the ASMO hardware platform, which provides various interfaces to enable interoperability and where the workload is distributed such that the complexity of each unit can be reduced, while still providing enough capabilities for embedded machine learning. By using microcontrollers running an embedded real-time operating system, the verifiability can be further increased. The intrinsically created distributed architecture additionally allows for flexible rearrangement and efficient extension if needed.

I. Introduction

Interconnected medical devices will be essential to enable new therapies and improve already existing ones. Following this approach, we identified two challenges: Interoperability and Verifiability. Due to various manufacturers and interfaces, direct interoperability of medical devices is not always guaranteed. Addressing the safety of the interconnected medical application, verifiability is essential. It depends, among other aspects, on the complexity of the system, which is increasing with the interconnection of medical devices. In intensive medical care, patients are in a critical state but not always under direct monitoring of the staff, such that malfunctions must be prevented under any circumstances. Using e.g. a Raspberry Pi to enable interoperability is limited in terms of available interfaces and verifying a Linux operating system (OS) is difficult due to e.g. pre-built libraries or the high complexity and memory model of the Linux kernel [1].

In this work, we propose a decentralized and modular platform, which we termed ASMO hardware platform. Each medical device is attached to an individual microcontroller board. To enable the interoperability, the boards offer interfaces for various communication principles. The hardware platform design also considers verifiability. Using a microcontroller supported by embedded real-time operating systems (RTOS), which only contain a small code size and are solely based on open-source libraries, allows the verification of the entire used software system. Furthermore, reducing the complexity of the system will be beneficial for the verification of the used algorithms. We distribute the tasks and workload onto multiple smaller units. This enables the plain alteration of the created cyber medical system by design. Each unit will have a comparable lower complexity without reducing the overall processing capabilities of the whole system.

II. Related Work

There are already commercially available hardware platforms to enable interoperability, like Capsule from Philips [2] or SCALEXIO by dSPACE [3]. Using Capsule, microcontroller boards are attached to medical devices to read measurements. However, the system cannot close a feedback loop to control devices. SCALEXIO provides the LabBox, which also fulfills various input/output (IO) requirements and enables a feedback loop. The LabBox offers a modular design by providing board slots for different functionalities, like analog-digital (AD) / digital-analog (DA) converters, Controller Area Network (CAN), Ethernet, Serial Peripheral Interface (SPI), etc. Nevertheless, due to the closed design, it is not possible to easily adjust the Box to fulfill custom interface needs.

However, there are also open-source solutions like OpenICE [4], where so-called dongles (Raspberry Pi's) are attached to the medical devices. The dongles itself are interconnected via Ethernet. One restriction is the limited flexibility regarding the interfaces due to the prebuild design of the Raspberry Pi's. Further, they are operated with a Linux OS, which complicates the verification compared to an embedded RTOS as stated earlier. This also applies to the platform developed by the Technical University of Munich [5] and the SyncBox [6], where one main unit, based on a Linux OS, is used to connect to the medical devices. In addition, those centralized setups further decrease the verifiability due to the increased complexity of the code of the main unit, handling all tasks. Moreover, adding new devices is more difficult since the main unit will eventually run out of resources.

III. ASMO Hardware Platform

With the sketched challenges and the limitations in mind, we developed the ASMO hardware platform (derived from Greek for safe translation), which can be used during the development stage, but also forms a basis for possible subsequent clinical use. A resulting ASMO board is a microcontroller board with various additional peripherals, which is directly attached to the medical device. The layout of such a board is presented in Fig. 1 (the schematics are available in [11]). To enable the use in various environments, it can be flexibly powered with 5-12V. For fast development, there is a Joint Test Action Group (JTAG) interface for debugging, a rotary button for ingoing and a display for outgoing information, as well as an SD card slot to store data persistently. To be able to run embedded RTOSs like FreeRTOS [7] or ChibiOS [8], we chose the STM32 F767ZI [9] microcontroller unit (MCU), which is supported with its ARM Cortex-M7 32-bit RISC core. This MCU offers enough computational power for embedded machine learning with a focus on energy efficiency, data security and low latency [10].

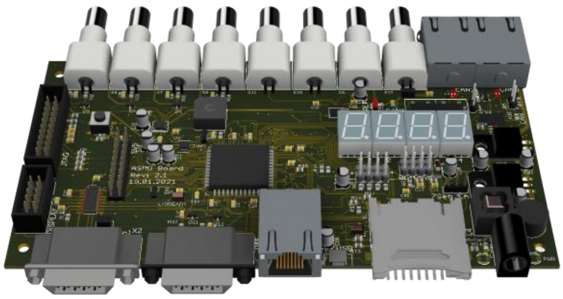

Figure 1: An ASMO board with various interfaces to enable interoperability and CAN / Ethernet for interconnectivity.

Various interfaces can be placed on the ASMO board to enable interoperability. For instance, there is an RS-232 and a USB interface. Moreover, there is an additional external AD/DA converter supporting 5V. Furthermore, enabling custom communication protocols is possible by simply mapping general-purpose IOs, SPI pins or AD/DA channels to a pin header. The general ASMO board is useful in the development stage, while specialized ASMO boards, reduced to essential interfaces can be designed for productive operation. The boards can be interconnected via CAN or Ethernet. For Ethernet, we use message prioritization via the IEEE 802.1p standard. Modularity is guaranteed, thanks to the abstraction layers provided by the embedded RTOSs, which e.g. enable an easy upgrade to a more powerful microcontroller, as we already proved in the past, when upgrading from an AT91SAM7 MCU. Modularity is also important regarding the hardware setup. In our decentralized system, the devices are connected to individual ASMO boards, which are interconnected via the dedicated communication interface. This setup reduces the complexity of code that needs to be executed on a single board, since tasks are distributed to multiple ASMO boards without reducing the overall processing power. Thus, code modularity is already enforced by this decentralized hardware setup. Thanks to the reduced complexity of each unit and only having a small code size due to the use of an embedded RTOS, we were able to apply various formal methods using Polyspace by the MathWorks.

We connected the ASMO platform to different devices and peripherals. RS-232 was useful to connect, among other devices, to a Datex-Ohmeda AS/3 patient monitor, a Siemens Servo 300 ventilator, a TERUMO CDI 500 blood parameter monitoring system, a Reglo ICC roller pump from Ismatec and a SonoTT flow sensor from em-tec. Via Ethernet, it was possible to communicate with a MX500 patient monitor from Philips. Ethernet and RS232 are standardized interfaces, which are widely available. However, more custom solutions were necessary, for instance to connect to a Transonic HT100 flowmeter or a ESCON 50/5 engine control unit, which in turn was used to control a rotary blood pump. Both only offer AD/DA channels and custom pins for the communication. Thus, we designed pin headers, which can be directly placed on the ASMO board and then wired to the associated device.

IV. Conclusion

In this work, we proposed the ASMO hardware platform, which was developed considering the limitations of the related work, like customizability and verifiability. It enables interoperability by offering various interfaces and a standardized communication channel to interconnect the ASMO boards. The board layout is publicly available and can easily be adapted. To enable verifiability, which is essential for the functional safety of medical devices in intensive care, we proposed a decentralized setup with the workload distributed to multiple boards and thus reducing the complexity of each unit and allowing for efficient extension of the whole setup. Further, we emphasized the importance of running an embedded RTOS with a small code size and open-source libraries to ensure the verifiability of subsequently developed medical devices.

Author's statement

Conflict of interest: Authors state no conflict of interest.

References

[1] D. Bristot de Oliveira, *Automata-based Formal Analysis and Verification of the Real-Time Linux Kernel*, 2020. doi: 10.13140/RG.2.2.30777.39523.

[2] Capsule, *Capsule Connectivity Management.* Capsule. https://capsuletech.com/connectivity-management/ (accessed Nov. 29, 2022).

[3] dSPACE GmbH, *dSPACE MAGAZINE*, vol. 2, 2017.

[4] R. Ivanov, H. Nguyen, J. Weimer, O. Sokolsky, and I. Lee, *OpenICE-lite: Towards a Connectivity Platform for the Internet of Medical Things,* in 2018 IEEE 21st International Symposium on Real-Time Distributed Computing (ISORC), May 2018, pp. 103–106. doi: 10.1109/ISORC.2018.00022.

[5] A. M. Garcia, M. R. Huizar, B. Baumgartner, U. Schreiber, and A. Knoll, *Embedded platform for automation of medical devices,* in 2011 Computing in Cardiology, Sep. 2011, pp. 829–832.

[6] F. Aytac Durmaz, Altay Brusan, and Cengizhan Ozturk, *Unified Open Hardware Platform for Digital X-Ray Devices; its Conceptual Model and First Implementation,* IEEE J Transl Eng Health Med, vol. 8, p. 1800311, Jun. 2020, doi: 10.1109/JTEHM.2020.3000011.

[7] FreeRTOS, *FreeRTOS - Market leading RTOS for embedded systems*. https://www.freertos.org (accessed Jan. 30, 2023).

[8] G. Di Sirio, *ChibiOS free embedded RTOS - ChibiOS Homepage*, 2022. https://chibios.org (accessed Jan. 23, 2022).

[9] STMicroelectronics, *STM32F767ZI - High-performance and DSP with FPU, Arm Cortex-M7 MCU,* https://www.st.com/en/microcontrollers-microprocessors/stm32f767zi.html (accessed Nov. 29, 2022).

[10] T. Ajani, A. Imoize, and Prof. A. Atayero, *An Overview of Machine Learning within Embedded and Mobile Devices-Optimizations and Applications,* Sensors, vol. 21, pp. 1–44, Jun. 2021, doi: 10.3390/s21134412.

[11] A. Stollenwerk, A. Derks, *ASMO: a decentralized and verifiable interoperability platform in intensive care,* 2023. https://doi.org/10.18154/RWTH-2023-00139