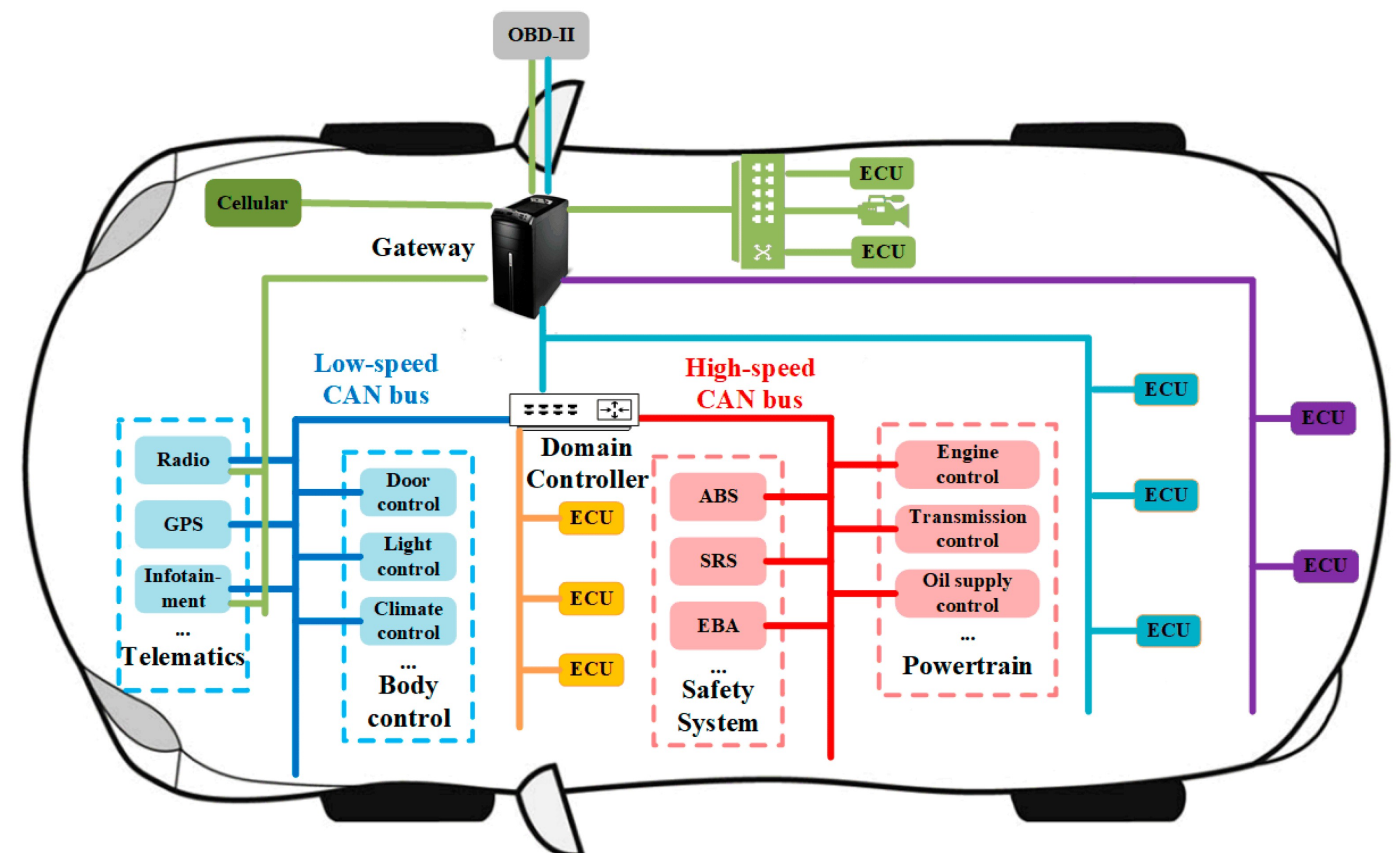


Debugging over CAN

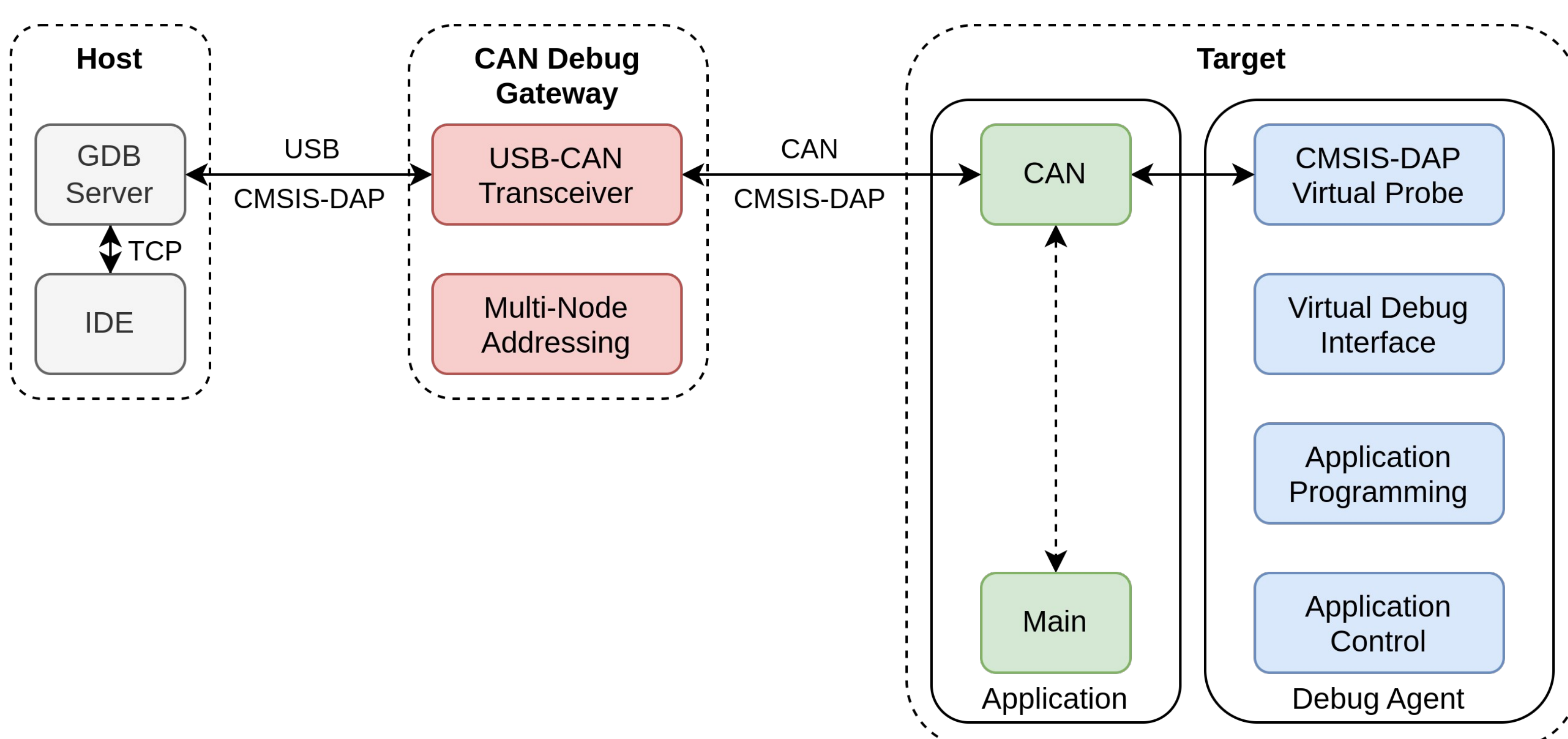
Inaccessible Debug Interfaces

Debugging complex hardware and software is crucial in order to ensure their functionality and detect errors at an early development stage. In space-constrained embedded systems, such as in modern cars, the debug interface of their components is often hardly accessible. Industry partner NXP, as a developer of automotive solutions, has a particular interest in improving this situation.

Instead of expensive additional wiring or modifying the hardware, the objective of this Master's thesis was to enable hardware debugging for such systems via an existing communication interface. In this case, CAN FD, an extension of the classical CAN protocol, was used.



System architecture of a modern vehicle (MDPI.com)



Overview of the debug system with the Cortex-M Debug Framework

ARM Cortex-M Debug Framework

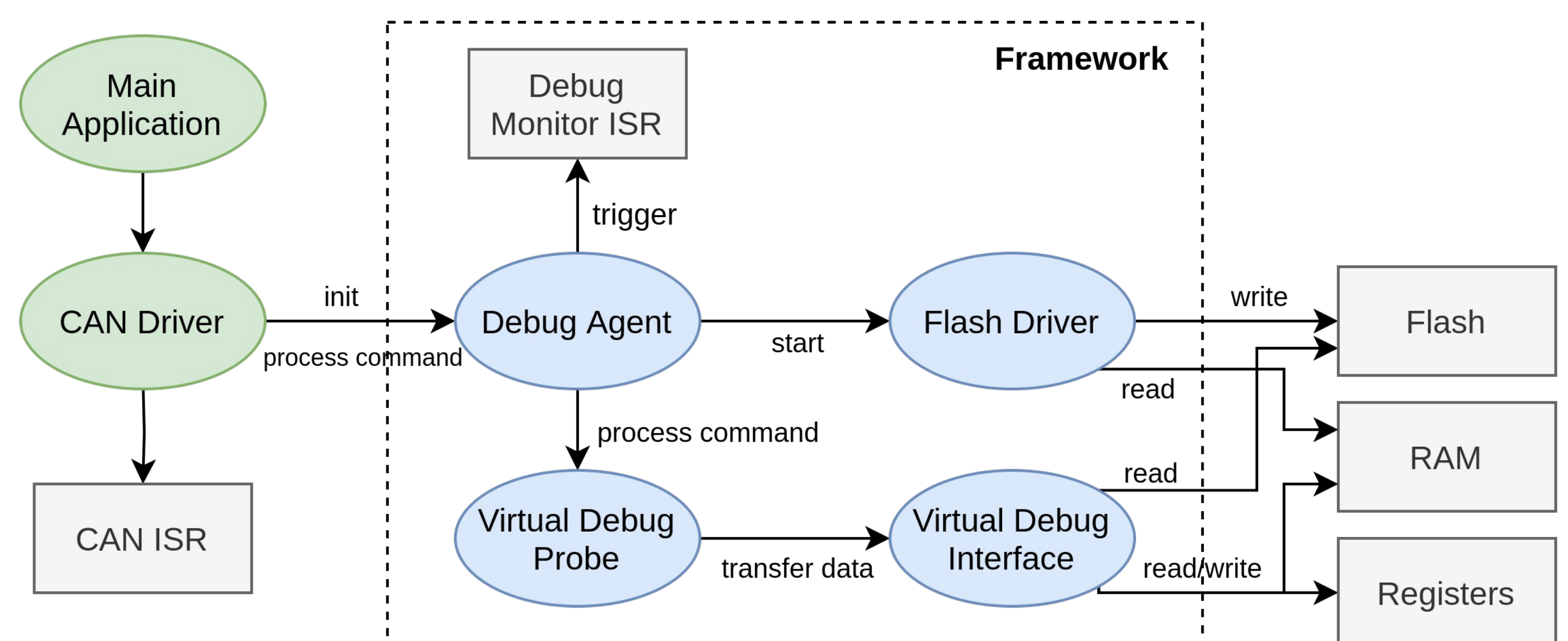
The solution to this challenge comprises a comprehensive framework that can be integrated into existing projects for ARM Cortex-M microcontrollers. The so-called *Debug Agent* processes CMSIS-DAP packets received via CAN and provides access to the debug resources of the MCU via a virtual debug interface. This enables hardware debugging with control over the execution of the application, access to variables and memory and the reprogramming of the flash memory.

With the additional *CAN Debug Gateway* a firmware is provided that simplifies the connection of the host computer to a CAN interface via USB and allows multiple targets to be addressed on the bus.

Results & Outlook

Example projects for bare-metal and FreeRTOS applications on NXP's MCXN947 controller were used to successfully demonstrate the effective use of the *Debug Agent* as debug target. The *CAN Debug Gateway* was implemented for the LPC55S16 from NXP.

Although the use of the framework generates additional overhead on memory, CPU and the communication interface, this remains reasonable in all cases and can be optimized if necessary. The hardware-independent implementation of the framework also allows further possible areas of application beyond the CAN bus (UART, Ethernet, etc.).



Internal and external dependencies of the framework

Key Facts

Cortex-M Debug Agent (NXP MCXN947)

CAN Debug Gateway (NXP LPC55S16)

- extend debug functionality on inaccessible systems
- use of existing communication interface
- allow debugging of multiple targets on same bus

Architecture: ARM Cortex-M (except M0/M0+)

Target OS: bare-metal, FreeRTOS

Host OS: Windows, Linux

GDB Servers: NXP LinkServer, pyOCD, OpenOCD

Author: Simon Kathriner

Advisor: Prof. Erich Styger

Industry Partner: NXP Semiconductors

