Zurich University
of Applied Sciences

School of Engineering

InES Institute of Embedded Systems

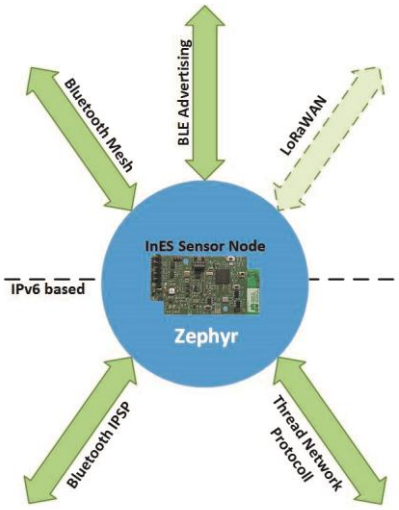
Consolidating IoT Firmware Architectures with Zephyr

Diplomand/in
Benjamin Häring

Dozent/in
Andreas Rüst

Currently, the IoT team at the Institute of Embedded Systems (InES) mainly implements firmware related to projects or used microcontrollers. As a result, source code can often only be used once, because porting it to another microcontroller would be too time-consuming. In addition, the repeated implementation of basic drivers such as I2C, SPI or UART is time-consuming and valuable time is lost. This problem should be solved by using a hardware independent Real Time Operating System (RTOS). One of the most promising RTOS is 'Zephyr Project' of the Linux Foundation. Zephyr is organized as an open source project and gets a lot of support from well-known microcontroller manufacturers such as Intel, NXP, STMicroelectronics and Nordic Semiconductor. Especially products of the latter three are often used in Team IoT.

At the beginning of this work, the Zephyr RTOS was completely unknown at InES. The necessary knowledge was acquired independently and in specific work packages during this master's thesis. The porting of Zephyr to the InES sensor node, which was developed by Team IoT, served as a starting point. First, the necessary board files were implemented and added to the RTOS. As a result, many of the sample projects available in Zephyr can be used with the sensor node. Impressive here is the strong hardware abstraction provided by Zephyr. The missing sensor drivers could be implemented based on the generic APIs of Zephyr. With this step, these drivers can be used in future projects independent of the selected microcontroller without the adaptation of a single line of code.



System overlayed with all tested communication protocols for the InES sensor node in combination with Zephyr

Additional energy measurements show that Zephyr has a compact boot process and an effective power management system. When this system is activated, the kernel automatically manages the energy consumption of the RTOS. The idle thread checks when the next task is due. If the time interval is long enough, the kernel automatically switches to one of the sleep states of the microcontroller. The periphery is also managed by this system. Unused peripherals are automatically deactivated. In addition, the developer can suspend peripherals via this system if required. The system works perfectly and measurements show that it can compete with other low-power applications of the institute.

For further research, projects with LoRaWAN, OpenThread and Bluetooth Low Energy have been successfully ported, tested and measured. All results were mainly positive, and no major disadvantages were found. Thus, Zephyr covers a large part of the application areas in Team IoT, since the communication protocols selected are the most frequently used in the team. The useful and extensive function set of the kernel is also noticeable. It contains work queues, timers and delays. This set allows quick and easy porting.

The Zephyr knowledge gained in the master thesis can be used in many ways. Already, it has been used as a basis of decision-making to realize other projects like Bluetooth Mesh or a Security Workshop with Zephyr. The advantages and disadvantages of Zephyr could be clearly demonstrated. The newly set up central Github Repository of the team combines own implementations like boards or drivers, code examples and a wiki. Any Zephyr developer in the team, no matter which of the supported microcontrollers is used, can use the collected knowledge. In addition, Zephyr can be used to develop on Linux, Windows and MacOS. Developers can support each other and discuss without major barriers or dependencies. Overall, the work shows that existing applications can be ported to Zephyr quickly and with little effort. The applications ported to Zephyr are very performant and energy efficient.