JÖNKÖPING UNIVERSITY
School of Engineering

Registration final thesis work – Computer Engineering 15 credits

Students

Name student 1 Anton Richard Bäck	Name student 2 Boris Bozic
Personal number 1997-10-15-5731	Personal number 1996-09-06-1453
Programme Datateknik: Inbyggda system	Programme Datateknik: Inbyggda system
Mobile number 0760254586	Mobile number 0708361772
E-mail baan20ty@student.ju.se	E-mail bobo20zr@student.ju.se
I am open to writing in English: Yes, the report will be written in English.	I am open to writing in English: Yes, the report will be written in English.
I fulfil the prerequisites indicated in the course syllabus: Yes, I fulfil the prerequisites indicated in the course syllabus	I fulfil the prerequisites indicated in the course syllabus: Yes, I fulfil the prerequisites indicated in the course syllabus

Company/organisation

Name Combitech AB
Contact person Adina Valjakka
Mobile number or switchboard +46 734 18 50 31
E-mail adina.valjakka@combitech.com

JÖNKÖPING UNIVERSITY
School of Engineering

Project description - More information under heading "Explanation project description"

Title of final thesis work (preliminary)
Hardware interfaces in embedded systems: A inductive qualitative study on simulating hardware interfaces using software.

Subject area/main field of study: Computer Engineering
Computer Engineering

Executive Summary
This thesis will conduct research of how simulating an I2C (Inter-Integrated Circuit) with software implementations which could accelera time of development within a project.

The objective of the company/organization
The main goal of this work is to construct a simulated version of I2C hardware interface. To have an efficient I2C simulator, speed and behavior will be examined.

Background

Developing applications for embedded systems becomes constrained due to dependencies with peripherals, operating system and processors. The initiation of the development process requires a physical hardware component in order to start testing the code for potential issues. When an application is to be tested and executed for a microcontroller unit (MCU), the hardware user interface for the the MCU needs to be accounted for as well. To perform a simulation for this, a larger quantity of code needs to develop before executing the code. Issues which could arise is when a unit test results negatively, the developer needs to troubleshoot both the behavior of the application and the usage of the hardware interface (Logge, 2021). To effectively solve this issue an excellent tool would be to use of Real-time operating system (RTOS) which includes emulations of circuit interfaces and driver (Hambarde, Varma & Jha, 2014). Although RTOS providers such as FreeRTOS and Zephyr has a large quantity of developed emulators, the emulators API only sends mock data and does not include all of the communication messages as an actual embedded circuit does (FreeRTOS, 2022; Zephyr, 2022).

An important electrical component within embedded circuits is I2C which has been substantially used in digital applications since the conception of it. I2C is a component which is widely used for communication applications where many Input/Output devices is needed, furthermore, industries such as public security, automation controllers, space technology uses I2C as a communication device. A large quantity of microcontroller is designed to use I2C to execute and control their several peripheral devices.

The current studies on emulating hardware interfaces and embedded circuits mainly revolves around using existing hardware on a microcontroller with different types of software algorithms to resemble the behavior of the specific circuit of which they are trying to emulate (Molina-Robles et al. 2021). There is a large gap of in research of emulating integrated circuits such as an I2C circuit with only raw software solutions. This thesis will conduct research of how emulating an I2C circuit with software implementations which could accelerate the time of development within a project.

Logge, M. (2021) KTH, School of Engineering Sciences in Chemistry, Biotechnology and Health (CBH), Biomedical Engineering and Health Systems, Health Informatics and Logistics Direct link: http://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-296507

Market leading RTOS (real time operating system) for embedded systems with internet of things extensions (2022) FreeRTOS. Available at: https://freertos.org/ (Accessed: September: 29, 2022).

P. Hambarde, R. Varma and S. Jha, "The Survey of Real Time Operating System: RTOS," 2014 International Conference on Electronic Systems, Signal Processing and Computing Technologies, 2014, pp. 34-39, doi: https://doi.org/10.1109/ICESC.2014.15

R. Molina-Robles, R. García-Ramírez, A. Chacón-Rodríguez, R. Rimolo-Donadio and A. Arnaud, "Low-level algorithm for a software-emulated I2C I/O module in general purpose RISC-V based microcontrollers," 2021 IEEE URUCON, 2021, pp. 90-94, doi: https://doi.org/10.1109/URUCON53396.2021.9647309

Zephyr project documentation (2022) Zephyr Project Documentation - Zephyr Project Documentation. Available at: https://docs.zephyrproject.org/latest/ (Accessed: September: 29, 2022).

Purpose and research questions

Zephyr supports a simple emulator framework to support testing of drivers without requiring real hardware. The emulators are used to emulate hardware devices like I2C interface. When using Zephyr I2C emulator the data that is received is static data which is will always be the same. The reoccurring problem is that the there is a minimal quantity of protocol messages for example, error messages which will be received. This requires you to connect the hardware physically to mirror the real behavior of the I2C interface and its communication protocol.

Current emulation techniques revolve around using existing hardware to simulate a virtual one, this will still create a dependency with having a physical hardware present (Molina-Robles, 2021; Schaarschmidt, Uelschen, & Pulvermüller, 2022). To remove the hardware dependency, you could create a software simulation of a hardware interface (Johnson et al. 2021). To create a successful simulation of a hardware interface the simulation must have the same functionality which leads to our first research question:
[RQ1] How can we simulate the behavior of an I2C circuit with software implementations?

When simulating a hardware interface, one thing that is important besides the behavior is the data rate. Different types of interfaces have different data rates, and you want the simulation to mimic that data rate so that it reflects a physical connection. This leads to our second research question:
[RQ2] How can we simulate I2C interface data rate with software solutions?

Johnson, E., Bland, M., Zhu, Y., Mason, J., Checkoway, S., Savage, S., & Levchenko, K. (2021). Jetset: Targeted firmware rehosting for embedded systems. In 30th USENIX Security Symposium (USENIX Security 21) (pp. 321-338). Jetset: Targeted Firmware Rehosting for Embedded Systems | USENIX

Schaarschmidt, M., Uelschen, M., & Pulvermüller, E. (2022). Hunting Energy Bugs in Embedded Systems: A Software-Model-In-The-Loop Approach. Electronics, 11(13), 1937. https://doi.org/10.3390/electronics11131937

R. Molina-Robles, R. García-Ramírez, A. Chacón-Rodríguez, R. Rimolo-Donadio and A. Arnaud, "Low-level algorithm for a software-emulated I2C I/O module in general purpose RISC-V based microcontrollers," 2021 IEEE URUCON, 2021, pp. 90-94, doi: https://doi.org/10.1109/URUCON53396.2021.9647309

Method

To begin answering our research question, theoretical research will be conducted within the framework of embedded systems, I2C circuits, and real-time operating system emulators. A literature review will be collected as theoretical data as to why this research questions requires to be investigated, as well as quantitative data which will be collected from numerous different test cases both with real-time operating system emulated hardware API, emulated hardware with generated protocol buffer and actual hardware to analyze implementation efficiency.

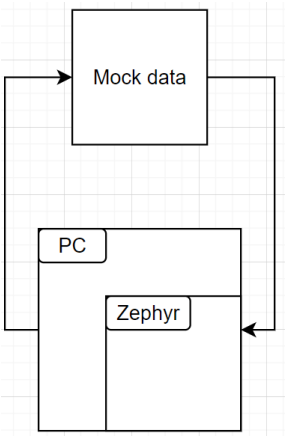

Figure 1: Illustrates how Zephyrs I2C emulator transfer data between a personal computer and their Application Programmable Interface (API)

Figure 1 illustrates the current state of how Zephyrs I2C emulator works with their I2C Application Programmable Interface (API). The API only reflects a minimal amount of the protocol messages that an actual I2C circuit sends out.

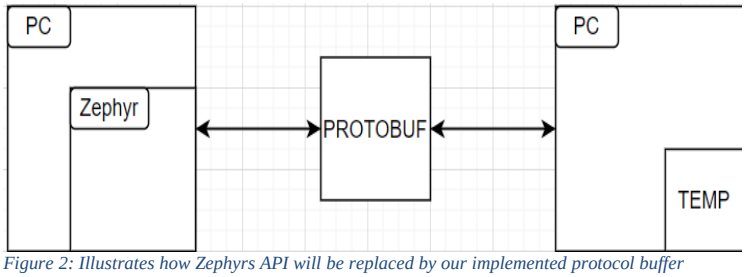

Figure 2: Illustrates how Zephyrs API will be replaced by our implemented protocol buffer

Our method to include all of the protocol messages of which an I2C circuit sends out, we will develop a serialized structured data format known as *protocol buffers* using programming language Python which will copy the behaviour of given I2C circuits.
The communication between the two PC's given on Figure 2 will be through WebSocket's implementation.

Relevance to the main field of study

The topic to emulate I2C hardware interface communication protocol with Real-time operating systems and software implementations will draw inspiration and expertise from the content of the following courses previously held withing our education programme:

- Microcontroller
- Operating system
- Electronic interfaces
- Network programming

The listed courses above have provided us with expertise for embedded and digital circuits, software development in embedded systems for microcontroller units, and the development of Real-time operating.