SUMMARY

Being a keystone of Industry 4.0, embedded systems represented an intriguing field that I decided to further explore and study through the simulation project at Proekspert. The project aims at building a simulation for the testing machine HW to reduce the average test time and provide developers with more flexibility. The simulation shall be portable and reliable, require no special setup and provide developers with ability to test their software without the real testing machines.

The simulation development started with a precise definition of the problem and the design requirements. Then, necessary background information about the used technologies was collected. The conceptual solution was formulated as the first step of the solution development. Afterwards, different technologies were compared and the project was configured taking the design requirements into consideration. The project was first run using fakes which are meant at eliminating errors during building and runtime. Finally, the debug screen was replaced by the console window and the core successfully sent commands to the simulated UI.

Reviewing the flow of the development process, it is evident that the process of running the simulation program with fakes demanded more time and effort than other stages. It lasted about 40 days while the process of implementing the documented features lasted around 20 days. Analysing the work done during that stage and type of errors that were tackled, poor coding style, lack of documentation and distorted architecture prove to cause a significant portion of the errors. In conclusion, good coding styles and documentation are essential for maintainability and editability of the software.

A variety of features was implemented which reveals about the potential of the project to efficiently replace the HW. Furthermore, the features were implemented in a reasonable period which highlights the effectiveness of the simulation thanks to the well-developed and flexible project setup. Moreover, the simulation empowers modular development as the core can be developed separately form the UI even when the BSP is being changed.

By the end of the project, the project was tested on other members' PCs, and it successfully ran. The project took 18 secs to build, old testing machines take up to 45 mins to flash the program, and all the edits made in the repository were reflected in the project. Necessary features were smoothly implemented within the specified period of development. Accordingly, the project successfully satisfied all the design requirements.

Future developments include the implementation of more features to fully simulate the HW. At first, the hardcoded bytes will be replaced with a simulated UI handler and the interface to the UI is to be

completed so that the UI and the core can communicate in both directions. Later, memories are to be replaced with system files and sensors readings are to be replaced with values that change according to a timer. Afterwards, features are to be added on request depending on the features needed for the core.