TECHNICAL UNIVERSITY OF MOLDOVA

FACULTY OF COMPUTERS, INFORMATICS AND MICROELECTRONICS
DEPARTMENT OF SOFTWARE ENGINEERING AND AUTOMATICS

EMBEDDED SYSTEMS

Practical Guide for Hardware-Software Interface Development

EDITURA TEHNICA-UTM

2026

CZU 004.3/.4(075.8)
B 76

This work is a practical guide to the development of hardware-software interface components within the Embedded Systems course and is intended for students of the 0613.3 Software Engineering bachelor's program. The document presents the fundamental principles of designing and integrating hardware components with software modules, includes technical recommendations, describes development techniques, and establishes a step-by-step software development process. Fundamental concepts related to embedded system architecture, interaction between microcontroller and peripherals, as well as the development of software flow, are explained in an accessible manner, with emphasis on the autonomous development of practical skills.

Autori: lect. univ., dr. în informatică, Andrei BRAGARENCO
asist. univ., Valentina ASTAFI

Responsible editor:

Reviewer: lect. univ., dr. în informatică, Dumitru MORARU

DESCRIEREA CIP A CAMEREI NAȚIONALE A CĂRȚII DIN RM

Bragarenco, Andrei.

Embedded systems: Practical Guide for Hardware-Software Interface Development / Andrei Bragarenco, Valentina Astafi; Technical University of Moldova, Faculty of Computers, Informatics and Microelectronics, Department of Software Engineering and Automatics.

– Chișinău: Tehnica-UTM, 2026. – 292 p. : fig. color, tab.

Aut. indicați pe verso p. de tit. – Ref. bibliogr. la sfârșitul compartimentelor. – 50 ex.

ISBN 978-9975-64-672-7.

Text: nemediat.

004.3/.4(075.8)
B 76

ISBN 978-9975-64-672-7

General instructions for carrying out projects and preparing reports for the course "Embedded Systems"

1. Use the current *"Practical Guide for Hardware-Software Interface Development"* as a guideline for implementation of the practical assignments for the practical assignments 1-5. Every presented problem in guideline is designed to be solved in one session of 4 academic hours, that could be split into 2 parts of 2 academic hours each.

2. Projects are carried out in parallel with the theoretical components of the course, so that concepts regarding the hardware-software interface, module configuration, and functional integration are understood prior to practical implementation. Students must be familiar with the theoretical material, the functional schematic, and the project procedure before starting the activity. To this end, the relevant chapters from the guideline and materials indicated by the instructor should be studied in advance.

3. All students present work on the same project, each having an individual variant of the assignment. Students identify the hardware configuration, assemble the experimental setup, establish the necessary connections for software communication, and verify subsystem operation. Commissioning of the system is performed only after prior verification of the schematic by the instructor. Any malfunction, abnormal operation, or deviation observed during the project must be reported immediately.

4. Project results are presented in the individual report. The report must be concise, reflect the requirements of the "Report Content" section, and include hardware-software diagrams, explanations regarding interface operation, experimental results, and their analysis. Schematics, diagrams, and tables must be presented in a standardized format. The report should not include general theoretical elements or descriptions unrelated to the results obtained. The report will be submitted by the beginning of the next project session.

5.2.9. Conclusions

In this project dedicated to actuator control, the stages of analysis, requirements definition, architectural modeling, implementation, and validation were methodically covered for an application controlling binary and analog actuators.

Domain analysis highlighted the importance of command signal conditioning for safe and efficient actuator operation, emphasizing the need for saturation, filtering, ramping, and hysteresis techniques. Functional and non-functional requirements ensured a clear structure, focusing on modularity, multitasking, and reusability.

Architectural modeling enabled separation of command interpretation, conditioning, and actuator control components, facilitating independent development and unit testing. Implementation of the signal conditioning library as an independent module demonstrated the advantages of reusability, allowing the same techniques to be applied to various actuators.

Systematic testing validated application functionality and identified aspects requiring optimization, such as ramping parameter adjustment and task latency reduction. Detailed documentation of the development process and test results ensures traceability and reproducibility of the solution.

This project demonstrates the importance of a methodical approach in embedded application development for actuator control. The presented principles and techniques provide a solid foundation for further development of more complex actuation and control systems, applicable in various embedded electronics domains.

5.3. Control questions

1. What is energy conversion into physical action and what are the main stages of this process in embedded electronic systems?
2. Explain the difference between unipolar and bipolar control of analog actuators. What are the advantages of each?

Reference

1. Arduino, *Arduino Serial Reference*. [Online]. Available: https://www.arduino.cc/reference/en/language/functions/communication/serial/. Accessed: Oct. 01, 2025.
2. AVR Libc, *AVR STDIO Library*. [Online]. Available: https://www.nongnu.org/avr-libc/user-manual/group__avr__stdio.html. Accessed: Oct. 01, 2025.
3. Bristol Watch, *L298N Motor Controller Theory Projects*. [Online]. Available: https://www.bristolwatch.com/L298N/index.htm. Accessed: Nov. 12, 2025.
4. Electronics Hub, *LED Interfacing with 8051 Microcontroller*. [Online]. Available: https://www.electronicshub.org/led-interfacing-8051/. Accessed: Oct. 01, 2025.
5. Electronics Tutorials, *H-Bridge Motor Control Circuit*. [Online]. Available: https://www.electronics-tutorials.ws/io/h-bridge-circuit.html.Accessed: Nov. 12, 2025.
6. EnergyPurse, *Types of Room Heaters*. [Online]. Available: https://www.energypurse.com/types-of-room-heaters/. Accessed: Jun. 07, 2025.
7. FreeRTOS, *FreeRTOS – Binary Semaphore*. [Online]. Available: https://www.freertos.org/Documentation/02-Kernel/02-Kernel- features/02-Queues-mutexes-and-semaphores/02-Binary- semaphores. Accessed: Oct. 01, 2025.
8. FreeRTOS, *FreeRTOS – Message Queue*. [Online]. Available: https://www.freertos.org/Documentation/02-Kernel/02-Kernel-features/02-Queues-mutexes-and-semaphores/01-Queues. Accessed: Oct. 01, 2025.
9. FreeRTOS, *FreeRTOS – Mutex*. [Online]. Available: https://www.freertos.org/Documentation/02-Kernel/02-Kernel-features/02-Queues-mutexes-and-semaphores/04-Mutexes. Accessed: Mar. 24, 2024.
10. G Mostofa BD, *8051 Push Button Interfacing*. [Online]. Available: https://gmostofabd.github.io/8051-Push-Button/. Accessed: Oct. 01, 2025.

11. How To Mechatronics, *Arduino DC Motor Control Tutorial – L298N | PWM | H-Bridge*. [Online]. Available: https://howtomechatronics.com/tutorials/arduino/arduino-dc-motor-control-tutorial-l298n-pwm-h-bridge/.Accessed: Nov. 12, 2025.
12. How To Mechatronics, *Control High Voltage Devices – Arduino Relay Tutorial*. [Online]. Available: https://howtomechatronics.com/tutorials/arduino/control-high-voltage-devices-arduino-relay-tutorial/. Accessed: Jun. 07, 2025.
13. Ikalogic, *8-bit Digital to Analog Converter (DAC)*. [Online]. Available: http://www.ikalogic.com/8-bit-digital-to-analog-converter-dac/. Accessed: Apr. 10, 2025.
14. IQS Directory, *Electric Motor*. [Online]. Available: https://www.iqsdirectory.com/articles/electric-motor.html. Accessed: Jun. 07, 2025.
15. Microsoft, *Visual Studio Code*. [Online]. Available: https://code.visualstudio.com/download. Accessed: Oct. 01, 2025.
16. OSEK-VDX, *OSEK-VDX*. [Online]. Available: http://www.openosek.org/. Accessed: Oct. 01, 2025.
17. SparkFun, *Serial Communication*. [Online]. Available: https://learn.sparkfun.com/tutorials/serial-communication. Accessed: Oct. 01, 2025.
18. Tech Explorations, *DC Motor with Transistor*. [Online]. Available: https://techexplorations.com/guides/arduino/motors/dc-motor-with- transistor/. Accessed: Jun. 07, 2025.
19. TechTarget, *Embedded System Definition*. [Online]. Available: https://internetofthingsagenda.techtarget.com/definition/embedded - system. Accessed: Oct. 01, 2025.
20. Texas Instruments, *LM20 Precision Analog Temperature Sensor Datasheet*. [Online]. Available: https://www.ti.com/lit/ds/symlink/lm20.pdf. Accessed: Jun. 08, 2025.
21. The Engineering Projects, *Proteus Library*. [Online]. Available: https://www.theengineeringprojects.com/2018/04/how- to-add-new-library-in-proteus-8.html. Accessed: Oct. 01, 2023.

22. Wikipedia, *Electric light*. [Online]. Available: https://en.wikipedia.org/wiki/Electric_light. Accessed: Jun. 07, 2025.
23. Wokwi, *Wokwi Documentation*. [Online]. Available: https://docs.wokwi.com/. Accessed: Oct. 01, 2025.
24. Wokwi, *Wokwi Simulator*. [Online]. Available: https://wokwi.com/. Accessed: Oct. 01, 2025.