

## The second lab assignment

1. Prepare an application in a compiled language (C or C++ preferred, make sure that the chosen language is supported in BR), which uses the buttons and LED diodes. When reading the buttons, please take into consideration the „contact bounce“ effect.
2. It is required that the application implements thorough error checking (you can't assume that e.g., all hardware communication functions always succeed).
3. The application must respond to changes in the state of the buttons without active waiting (should sleep waiting for the required change)
4. The application functionality should be defined by the student (below are just examples):
  1. **The timer controlled by the buttons (with lap-time function).** LEDs should be used to confirm the operation. The measured time should be displayed on the console.
  2. **Coded lock with tampering detection.** LEDs should signal the opening of the lock, an erroneous code, alarm state after the certain number of unsuccessful attempts.
  3. **„Memory game“:** application displays a sequence using the LEDs. The user must repeat it with the buttons.
  4. **Application for simple tests.** The questions (read together with the right answers from an external file) are displayed in the console. The user answers with the buttons. After each response, the LEDs signal: good answer, wrong answer, skipped question. At the end, the result is displayed in the console.
  5. **Morse code transmitter and receiver.** The LED transmits the text sent by the console. The text entered by the switch is decoded and output to the console).
5. The application should be converted into the Buildroot package
6. The student should demonstrate debugging of his application with the gdb debugger
7. *Additional task:* Prepare the equivalent application in a selected script language (Python? Lua?) and compare both implementations (resource usage - the application itself and the application + required libraries and/or interpreters, event handling time - measured by a system call and saved to a file).