Practical Exercise – Microcontroller Techniques

Prof. Dr. L. Reindl
Laboratory for Electrical Instrumentation

# Exercise sheet 8 & 9 - Project

**Important Note:**

These exercise sheets combine the experience from the previous experiments. There are **three** task options given, of which you have to **solve only one within exercise 8 and 9**; still, provide a feedback for both exercises. Instead of a detailed task description, you will only receive a rough sketch of a task which you should implement.

If you have your own idea for a small project, which can be realized with our development board and which seems to have a similar workload as the proposed tasks, please contact your tutor.

**Exercise sheet 8:** For exercise 8, please create a flowchart for your program, i.e. a graphical representation of the algorithm/program you want to implement (also see the lecture slides for details). Keep the flow chart on an abstract level, so don't include every small detail you want to use within your code (e.g. don't show how a counter variable will be increased, etc.).

**Exercise sheet 9:** For exercise 9, implement your ideas and sketches of exercise 8 within a real microcontroller program.

### Option 1 - Wireless data communication between two dev boards

- Realize a wireless data transmission between two boards (either look for a second board in the library or use the board of a collegue). Send data by the serial interface to board 1, which transmits the data packet to board 2 wirelessly. Board 2 will then return the data by the serial interface. The transfer should be bidirectional.
- You can choose any physical medium for the communication (i.e. optical, acoustic, electromagnetic, ...), just don't connect the boards by wires.
- It may be beneficial to implement some form of error detection under some circumstances (e.g. parity bits or CRC).
- You may implement some form of a flow control, so that the buffer of the serial interface does not overflow (e.g. enable an LED indicating that the system is not able to receive any new commands as long as the data is not transmitted successfully).

### Option 2 - Mastermind on two dev boards

- Program a modification of the game Mastermind on two development boards (either look for a second board in the library or use the board of a collegue).

- Physically connect two boards by multiple wires for data communication. It is also necessary to connect the **ground pins** of the boards, otherwise the applied potentials are not the same.

- The abstract concept of the game is the following: Both controllers should agree on a common secret code (randomly chosen) and give a visual or audible feedback that the boards are ready to play. Both players have to guess this code one after the other. After each code entry, the system will display how well the entry and the secret code match together.

- You are completely free in designing and modifying the game principle. For example, the code could be the pressing sequence of four buttons (e.g. `PB1` to `PB4`). So, the sequence `PB3` - `PB1` - `PB2` - `PB4` has to pressed in order to win the game. Once you enter any sequence not being the correct one, the LEDs D$i$ ($i$ going from 1 to 4) might indicate that PB$i$ was pressed at the correct position as a feedback. Alternatively, every player might have to enter a code of several 4-bit values. Remember, it's up to you. You can also use other peripherals (for example `PB5` and `PB6`, or else `U_POT`) as input and output.

- 'The feedback for each incorrect entry can be realized as a short visual or audible signal.

- If entered correctly, both boards are intended to provide a visual and audible feedback (depending on whether the corresponding player has won or lost).

- After a few seconds a new game should start.

### Option 3 - Tetris on the terminal

- Program an executable Tetris game on the microcontroller. For this purpose, realize a graphical output of your game by the serial interface to show it on a terminal program. We recommend to use Putty[1].

- Use the buttons on the circuit board for the control.

**Note:** You should make yourself familiar with the VT100 control codes; e.g. you can use the command `serialWrite(0x0C);` to clear the the display on the terminal and to move the cursor to the top left corner. You can also redraw only certain areas of the display. If desired, it is also possible to use different colors.

**Note for all tasks:**
Besides the basic functionality, you are completely free in the realization of the program. Be creative.

[1] http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html

**Feedback (for both exercises)**

a) Create a file `feedback.txt` with a brief feedback statement, which contains specific problems and issues you experienced while solving the exercise, additional requests, positive remarks, etc. (**1 pt.**).

b) Upload this text file `feedback.txt` together with the rest of your deliverables.