Exercise Sheet 1 - Display Interface

Important:
To get started with this course, please consider the slides of the introduction course being available on ILIAS. Please also consider the course material of the lecture „Mikrocomputertechnik / Microcontroller Techniques".

Background

Driving an external liquid crystal display (LCD) is an essential task in many embedded system projects. In this exercise, we're going to interface a *Hitachi HD44780* display controller (on a GE-1602B-TMI display module), which can be considered as a standard display driver used in various industrial implementations.
The interface from microcontroller to LCD controller consists of a 4-bit or 8-bit wide parallel data bus, a clock line (E), a read/write control line (R/W) and a reset line (RS). Most of the pins are accessible through `CON5` of the *advanced extension board* (see page 2 of the schematic). For minimal I/O usage, we're going to implement the 4-bit control interface.

Moreover, recall the contents from the introduction course: Libraries typically consist of a header file (file extension .h) and a source file (file extension .c). The header file now contains the function declarations, i.e. it lists all the existing function names together with their arguments and return values. The source file contains the source code itself with all function definitions, i.e. the actual implementation. By dividing the code into two parts, the exact implementation of the library is no longer of interest for the final user, only the interface defined by the header file has to be considered when working with the library.

Getting started

1) Familiarize yourself with the schematic of the *advanced extension board* and the data sheet of the *HD44780* controller.

2) Connect the *advanced extension board* to the regular microcontroller board using the ribbon cable. Make sure that the jumpers `JP1`, `JP2` and `JP3` are set to the default configuration connecting pin 1 and 2 (pin 1 carries an arrow on the board's silkscreen). Adjust the potentiometer `POT1` until you clearly see the character fields of the first line (see figure 1, rotate counter-clockwise).

3) Connect the data lines (`CON5:D4` - `CON5:D7`) to (`CON3:P2.0` - `CON3:P2.3`).

4) Connect the control lines in the following way:
`CON5:RS` to `CON4:P3.0`, `CON5:R/W` to `CON4:P3.1` and `CON5:E` to `CON4:P3.2`.

5) In Code Composer Studio, import the project archive `Exercise_1.zip`, which you can find on the ILIAS web platform.

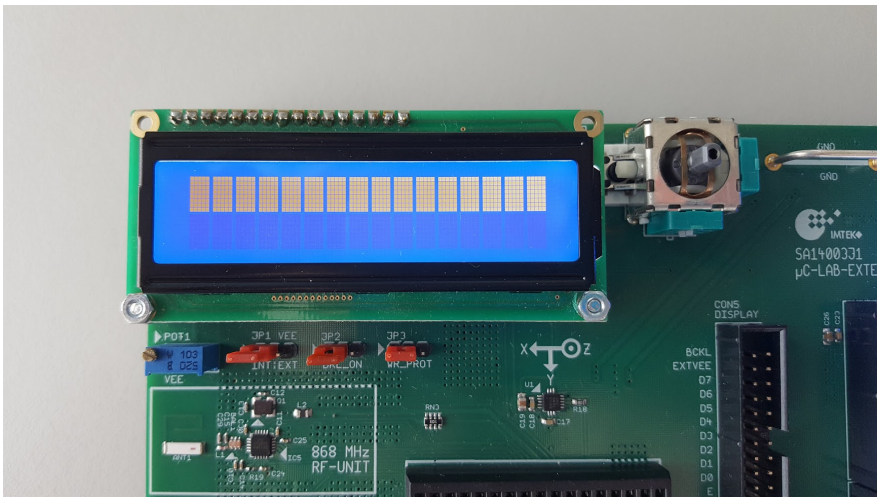

Abbildung 1: Basic configuration of the LCD.

Task 1

Implement a library for controlling the *HD44780*-based LCD display on the *advanced board*. Therefore, consider the following steps:

a) Check the 'libs' folder in the given project for the (already existing) header file *LCD.h*, which contains function prototypes together with a description of what these functionals shall do. Your task is to write the appropriate implementation of the functions in the `LCD.c` source file. You may add more functions to perform subtasks, but you must not change the names of the existing ones.

b) Properly comment your code. It's not required to write several lines of comments for each line of your code, but it should be enough to roughly understand what you did.

For the individual point distribution, consider the header file *LCD.h* **(7 pts. in total)**.

Be careful:

You must take care to prevent data collisions on the bus, i.e. check for the BUSY-flag or use delay times according to the data sheet.

Task 2

a) Write a basic demonstration of your library which instantly starts when powering the microcontroller and which runs autonomously without any user input. This demonstration should contain all of the functions that originally have been defined in the header file. **(2 pts.)**

b) *Bonus*: Implement at least one custom character (using CGRAM) and integrate it into your main routine. **(2 bonus pts.)**

Task 3

Prepare your exercise sheet for submission using the following subtasks **(1 pt.)**:

a) Fill the file `feedback.txt` with a brief feedback statement, which contains specific problems and issues you experienced while solving the exercise, additional requests, positive remarks and alike.

b) Please assure that every file contains a header comment including...
   - your personal information (i.e. name, matriculation number, e-mail contact).
   - the exercise sheet number.
   - a brief description of the given code and its purpose; the description of the file *main.c* should also list the required pin connections.

c) Rename your project to *Exercise_[ExerciseNo]_[YourLastName]* within Code Composer Studio.

d) Export your project to an archive file (ZIP) using Code Composer Studio (File $>$ Export... $>$ General $>$ Archive File).

e) Upload your file to ILIAS considering the individual deadline.