michael.engel [☺] hs-coburg.de

Embedded Project
Summer 2018

Assignment 2

Assignment issued on Thursday, April 19, 2018

Honor code: Work on the assignments is to be carried out in the assigned groups of two or three students. You're welcome to talk to classmates about assignments conceptually, but the implementation must be your own unless otherwise noted. For example, speaking with a classmate about why some bug might be present in your program is okay as long as your classmate doesn't share with you a program that fixes your bug. You should not use any existing code that we don't supply, whether it is online or otherwise. Further, copying code from places like Stack Exchange is prohibited: you will learn much more if you write your own programs.

2.1 LC Display

In order to visualize the actions of the sensor control code to be written in the upcoming assignments, enabling output to the robot's LC display is useful.
MC mobile uses a monochrome graphical LCD with a resolution of 240 $\times$ 128 Pixel. The display can show text (as ASCII characters) as well as pixel graphics.
The LCD controller is a T6963C by Toshiba (data sheet in moodle). It is connected to the Cortex-M§ as a *memory-mapped* device, i.e. it can be accessed using normal read and write accesses to given memory addresses.

- To enable the use of memory-mapped devices, the external memory controller of the LPC1778 has to be initialized first. The LCD controller is connected to the LPC1778 via the following pins:

pins	function
P4.0–P4.15	adress bus A0–A15
P3.0–P3.7	data bus D0–D7
P4.24	output enable OE
P4.25	write enable WE
P4.30	chip select CS0

  Your code should configure all these pins to enable the external memory functionality and activate the pull-up resistor and the hysteresis. The corresponding information can be found in the LPC1778 user manual (UM10470) in chapters 7, 8, and 9 – see tables 70, 76, and 77.

- Enable the power supply for the external memory interface (register PCONP, cf. user manual ch. 4, tab. 37).

- Now, enable the external memory controller itself (ch. 10, EMCControl register) and configure the following parameters in the EMCStaticMemoryConfiguration register:

  - EMC8Bit
  - Page mode disabled
  - ChipSelectPolarityActiveLow
  - ByteLaneState
  - ExtendedWaitDisabled

  The remaining bits of the register must be set to 0!

- The LCD controller data register is located at memory address 0x80004000, the command register at address 0x80004004. After configuring and activating the external memory cotroller of the LPC1778, you now have to configure the LCD controller T6963 itself.

  To develop this code, read the section "Flowchart of communications with the MCU" (p. 9), in the LCD controller data sheet. Take special care of the bits STA0 und STA1, these are important for ensuring correct communication with the LCD controller!

  Initialize the controller (table p. 11) as follows:

  - Set Text Home Address to 0x0000
  - Set Text Area to 30 (decimal) - one character is 8 pixels wide, thus 30 characters fit into 240 horizontal pixels
  - Set Graphic Home Address to 0x0200
  - Set Graphic Area to 30 (decimal): 30 bytes per line
  - Activate the EXOR modus (overlapping pixels of text and graphics memory are combined via XOR)
  - Set Display Mode for text and Graphics to ON

- Write a function to output strings on the display. Use the controller commands "Set Address Pointer" and "Data Write and Increment ADP". The text memory starts, as configured before (Text Home Address) in the *controller memory* at address 0.

- Write a function to load bitmaps into the LCD's graphics memory. You can define these in C code as follows:

```
unsigned char bild_0 [3840] = {.....................};
```

  You can load multiple bitmaps into the controller graphics memory (e.g. at addresse 512, at address 512 + 3840, at 512+3840*2 etc.) and switch between these by setting the Graphics Home Address. This way you can implement animations.

  With the program `LCDProject.exe` you can convert bmp files into the byte values required for your source code. To get this to work, you will have to replace every "db" in the output of LCDProject.exe by a comma (e.g. in notepad)

- Write graphic functions for basic operations:

  - Set and erase single pixels
  - Draw rectangles (outline and filled)
  - Optional: draw circles/ellipses

- Especially the transfer of complete bitmaps is rather slow, since the status has to be checked after each written byte. The LCD controller also supports a faster "Data Auto Write" mode. Use this mode to load bitmaps. You can disable the auto write mode (e.g., after copying a bitmap) with the command "Auto Reset".