DESN2000 (Computer Engineering) 2024 T2

Lab sheet 2 (weeks 3 and 4)

Last edited: 12/06/2024

Make reasonable assumptions if not explicitly stated and state such assumptions to your demonstrator when getting marked.

Task 1 (10%)

Follow the debugging tutorial available under lab 2 in WebCMS. Demonstrate to your tutor, how you will debug an example bug using debugging tools available in cubeIDE and answer any question. Yes, this is an easy mark. But we want to ensure you all learn debugging methods so that life will become easier in the long run.

Task 2 (20%)

Write a programme that displays the following on the LCD.

Hello
World

You may want to implement the LCD code from scratch if you are courageous. But if you need some guidance, refer to the following for initialising the LCD.

```
  // 1. wait for enough time to stabilise
  HAL_Delay(50);

  // 2. send command 0011 (function set) and wait for >=4.1 ms (enough wait inside the pulse)
  LCD_PutNibble(0b0011); LCD_Pulse();

  // 3. send command 0011 (function set) again and wait for >=100 us
  // fill this

  // 4. send command 0011 (function set) again
  // fill this

  // 5. send command 0010 to set to 4-bit bus mode
  // fill this

  // 6. send command 0010 1100 (function set: 4-bit mode, 2-lines, 5x8 font)
  LCD_SendCmd(0b00101100);

  // 7. Send command 0000 1000 to display ON/OFF
  LCD_SendCmd(0b00001000);

  // 8. Send command to clear the display
  // fill this

  // 9. Send command set entry mode (increment cursor, no display shift)
  // fill this
  // 10. send command 0000 1111 to display on, cursor on, blink on
  // fill this
```

Check the week 2 Monday lecture on implementing functions such as `LCD_PutNibble` and `LCD_Pulse`.

Hints:

- Use #define statements to define LCD controller instructions with sensible names
- Make sure you don't forget to add `__HAL_RCC_GPIOD_CLK_ENABLE()`

Task 3 (20%)

Write a programme that displays the relevant character on the LCD when a key on the keypad is pressed. One key press should only produce one character – debounce! When the first row of the LCD is full, you should go to the second row. When the second row is full, clear the display and start from the top left of the LCD.

Task 4 (30%)

Implement a simple calculator using the keypad and the LCD. The keys A, B, C, D and # should be used for +, -, x, / and =, respectively. To clear the display use * key. See the following example:

1+2-3/2+2=
3.5

Assume that the expression on row 1 is not longer than 16 characters. Handle errors such as invalid syntax and division by 0, while giving an appropriate error message.

Task 5 (20%)

Without using HAL functions, write a programme that increments a counter when SW1 is pressed and decrements when SW2 is pressed. The counter is only 4-bit and should be displayed on LEDs D4-D1. For this task, you are NOT allowed to use HAL functions. Instead, you should be directly accessing relevant memory addresses in C. What you have to write in C directly would be the following:

- Everything inside the *MX_GPIO_Init(void)*
- Infinite while loop inside the main function

Hints: You can first write a programme using HAL, and then replace each HAL function with direct C code, one by one. This way, it would be easier to debug. Check the week 2 Friday lecture if you need help.