

Task-3: Interfacing LCD for debugging

Experiment 1: Getting Familiar with LCD Interfacing– Displaying alphanumeric characters

The aim of this experiment is to get you familiar with LCD interfacing on the Atmega 2560 based firebird V robotic kit. In this experiment, you will write program code to interface a 16x2 LCD with Atmega 2560 microcontroller. The 16x2 LCD is connected to Port C of Atmega 2560. Refer to the LCD pin connection table below:

LCD pin connection:

LCD pins	Microcontroller Port C pins
Register Select (RS)	Pin no.1 (PC0)
Read/Write (R/W)	Pin no.2 (PC1)
Enable (EN)	Pin no.3 (PC2)
Data bus (DB7)	Pin no.8 (PC7)
Data bus (DB6)	Pin no.7 (PC6)
Data bus (DB5)	Pin no.6 (PC5)
Data bus (DB4)	Pin no.5 (PC4)

Your task is to display “**TBT**” on the LCD, starting from the 1st line 4th column and “**e-Yantra**” starting from the 2nd line 2nd column of the LCD.

Procedure:

Step-1: Open Experiment-1 folder. Open the given project named “**Experiment-1.aps**” in AVR Studio.

Note: If the file does not open, start AVR Studio 4, click on Project then click on “Open Project” and browse the “**Experiment1.aps**” file to open the given project.

Step-2: You will notice some pre-written function declarations and also a C file named “**lcd.h**” included for your assistance. Write program code to complete the functions with the help of function description and inline comments.

The `lcd_interfacing.c` file contains three important function definitions needed to configure LCD and send command/data to the LCD registers.

- **lcd_set_4bit ()**

This function is used to configure LCD to work in 4 bit mode.

Example call: `lcd_set_4bit();`

- **lcd_wr_command (unsigned char cmd)**

This function is used to select the command register and transfer desired commands to the LCD for its operation.

Example call: `lcd_wr_command(command in hex);`

- **lcd_wr_data (char letter)**

This function is used to select the data register and transfer the desired data to be displayed on the LCD.

Example call: `lcd_wr_data(ASCII value of data);`

You will notice certain function calls without appropriate arguments, complete them with the help of comments written besides each of them.

Example:

`lcd_wr_command (--)` // Write command to move cursor to 2nd line 2nd column.

Replace “--” with hex value of the command to move cursor to 2nd line 2nd column.

Step-3: Check and debug your code on the robot. Ensure that code performs as expected.

Step-4: Save the project and create a .zip file.

Step-5: Upload the .zip file of your saved project on the portal for evaluation

Click on this link: <https://goo.gl/il834t> to see the expected output for this experiment.

Note: We request you to strictly adhere to the code layout as given in the project C file and avoid adding any new functions. However, you can declare and define new variables, if required.

Grading:

Maximum marks for this experiment are **10**, divided among the following tasks.

1. Initialization of the LCD as mentioned. (2 marks)
2. Displaying the text “**TBT**” at the specified position on the LCD. (4 marks)
3. Displaying the text “**e-Yantra**” at the specified position on the LCD. (4 marks)

Experiment 2: Displaying counter values on the LCD

The aim of this experiment is to get you familiar with LCD interfacing on the Atmega 2560 based firebird V robotic kit. In this experiment you will write a program code to display an up counter on the LCD.

Your task is to display the text “**Counter**” starting from the 1st line 1st column of the LCD and an up counter going from **00** to **25** on the 2nd line 1st column of the LCD. The counter should increment by 1 after every **second** and should stay at **25** once that count is reached.

Procedure:

Step-1: Open Experiment-2 folder. Open the given project named “**Experiment-2.aps**” in AVR Studio.

Step-2: You will notice some familiar pre-written function declarations similar to Experiment-1 except one new function named “**display**” has been included.

- **display (unsigned char val)**

This function is used to split a 2 digit value in to tens digit and unit digit, convert them into ASCII by adding 30H and display them on the LCD.

We have written a part of the code to implement display function, complete this function by writing the remaining part. Refer to the inline comments mentioned and function description for assistance. It might be helpful to go through the written tutorial on “**Programming in C**” given on the Task page.

Write program code to complete all the functions with the help of function description and inline comments.

Step-3: Write program code in main () function to increment a variable from **00** to **25**. The variable value should be incremented by 1 after every second. You can use the function “_delay_ms()” to generate the delay required.

Step-4: Check and debug your code on the robot. Ensure that code performs as expected.

Step-5: Save the project and create a .zip file

Step-6: Upload the .zip file of your saved project on the portal for evaluation

Click on this link: <https://goo.gl/ULB0VC> to see the expected output for this experiment.

Grading:

Maximum marks for this experiment are **10**, divided among the following tasks:

1. Displaying the text “**Counter**” at the specified position on the LCD. (2 marks)
2. Counter values being displayed at the specified position and being incremented by 1. (4 marks)
3. Counter values starting from **00** and ending at **25**. (2 marks)
4. The displayed value is incrementing after the specified delay (2 marks)