# LAB 4-2

# COMMUNICATE WITH SERIAL PORT, EEPROM, RTC

## OBJECTIVE:

- Understand and use UART, I2C, SPI peripherals.
- Understand how to communicate with RTC and EEPROM.

## REFERENCES:

- Lab manual chapter 7, 9, 11
- Atmel-2505-Setup-and-Use-of-AVR-Timers_ApplicationNote_AVR130.pdf

## EXPERIMENT 1:

a) Connect the additional necessary signals to control the character LCD.

b) Connect the UART signals to the RS232 module and establish a connection using a USB-Serial cable.

c) Write a program to receive characters from UART using interrupts, display the received character on the LCD at the leftmost position, and simultaneously count the number of received characters and display it on a 4-digit 7-segment LED display. When the count exceeds 1000, reset it to 0.

Instructions:

- Use UART interrupts to receive characters. Inside the UART interrupt, increment a 16-bit counter variable, convert it to BCD format, and store it in four bytes of the LED7segValue array. For example, if the count is 500, store 0-5-0-0 in these four bytes.
- The LED scanning part remains the same as in Exercise 3.

  (Refer to Chapter 4, section 4.4, of the experiment guide for further guidance.)

## EXPERIMENT 2:

Connect additional I2C signals to the RTC module. Route the MFP signal from the RTC to an external interrupt pin. Add the following functionalities to Exercise 4:

- Initialize the RTC with the current time and enable the MFP output signal with a frequency of 1 Hz.
- Utilize the external interrupt of the AVR. Whenever an interrupt occurs, read the hour:minute:second time from the RTC and display it on line 2 of the LCD.
  Note: The functionalities from Exercise 4 will remain unchanged.

## EXPERIMENT 3:

a) Connect the necessary signals to control the LED matrix. (Remove the connections to the LCD, 7-segment LED displays, RTC, and UART).
b) Use the provided sample program and make necessary modifications to display the letter 'A' on the LED matrix. Scan the LED matrix using a timer interrupt with a scanning frequency of 25 Hz.
c) Modify the program to achieve a scanning frequency of 125 Hz.
d) Write a program to display the logo of the Bach Khoa University on the LED matrix.

## EXPERIMENT 4:

Requirements:

- Add to the experiment 4 in Lab 3-1 the functionality of measuring motor speed and displaying the speed and direction on an LCD.
- The signal from one channel of the encoder is fed into the clock input of Timer 2, using Timer 2 in external clock mode. When Timer 2 overflows, an interrupt is generated, and a counter is incremented to count the number of timer overflows.
- The two channels from the encoder are also connected to two port pins to determine the forward/reverse direction.
- A time period of 1 second is created using Timer 1 interrupt. When the interrupt occurs, count the number of pulses from the encoder within that 1-second period and calculate the motor speed, then display it on the LCD. Reset the counters to begin the measurement process again.

# LAB REPORT

*Class group:* *Group:*

*Subject:*

## EXPERIMENT 1:

1. Answer the questions:

   a. Describe the connections on the kit.

   b. Describe how to configure UART.

   c. How is the conversion from a count to a BCD number performed to write it into the LED display buffer?

2. Program source code with comments

   .....

## EXPERIMENT 2:

1. Answer the questions:

   a. Describe the connections on the kit.

   b. How is an external interrupt configured to occur once per second?

   c. What will happen if we configure an active-low level-triggered external interrupt?

2. Program source code with comments

   .....

## EXPERIMENT 3:

1. Answer the questions:

   a. Describe the connections on the kit.

   b. How long will a column LED remain lit to achieve a scanning frequency of 25Hz?

   c. What are the differences when scanning at frequencies of 25Hz and 125Hz?

2. Program source code with comments

   .....

## EXPERIMENT 4:

1. Answer the questions:

   a. Describe the connections.

b. For the motor in the experiment, what is the formula for calculating the speed?

c. Can Timer 1 be used to generate an interrupt every 1 second?

d. How is Timer 1 configured?

e. How many bits is the counter register of Timer 2?

f. In order to use Timer 2 to count the number of pulses in 1 second, is there a possibility of Timer 2 overflow?

Instructions: Calculate the maximum number of pulses in 1 second based on the maximum speed of the motor, transmission ratio, and pulses per revolution.

g. If there is a possibility of Timer 2 overflow, what should be done to ensure accurate counting of encoder pulses?

2. Program source code with comments

| ..... |
|---|