Krzysztof Piotrowski

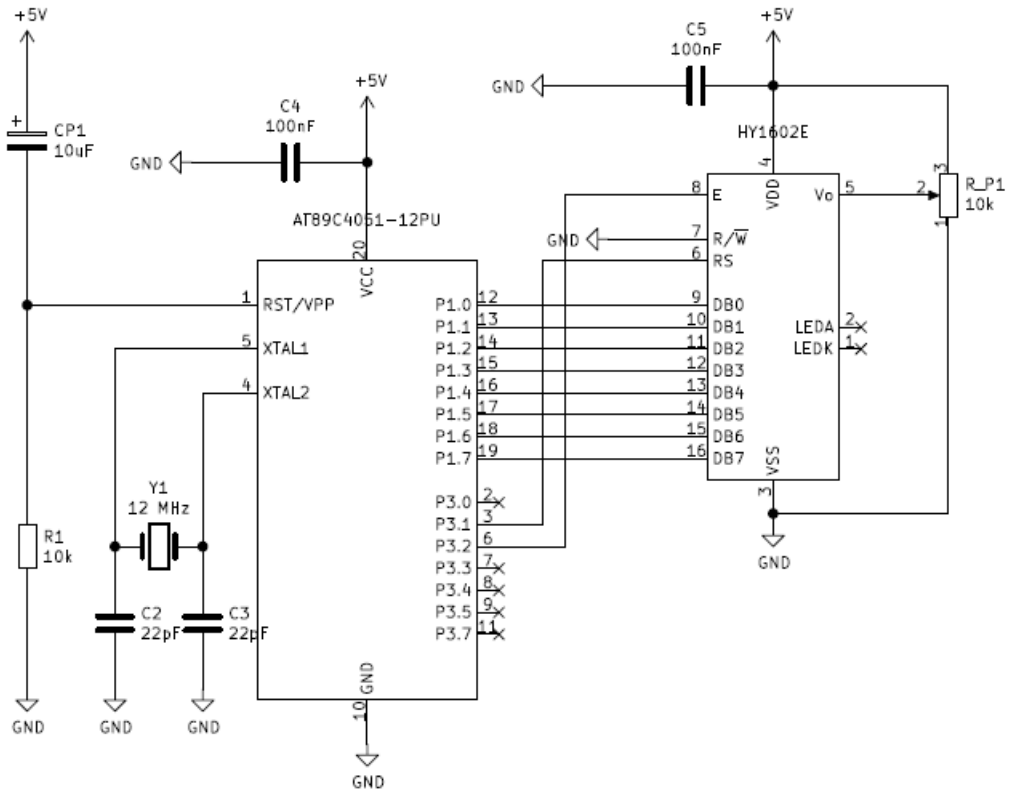
EMISY LAB 1

21st April 2021, 16:00

Index no.: 300175

Introduction

The main goal of this laboratory is to get familiar with 2x16 HD44780 LCD Display using 8051 microcontroler. The hardware scheme for this laboratory can be seen below however, for this task I will use EDSIM51 simulator to check the correctness of the code.



Task 1.

The main goal of this task was to send and display first letter of my name on LCD display using EDSIM51 simulator.

```
LCD_BUS equ P1
LCD_RS equ P3.1
LCD_E equ P3.2
```

Clock is set for 12 Khz. Pin R/W is tied low to work in write mode. Labels were created for E, RS, and data bus pins according to the scheme above for better clarity and visibility in code

Code is divided into several subroutines:

- **main:** used to call other subroutines. Thanks to that it is easy to see order and process of execution as well as delay between each other subroutine.
- **initializtion:** used to initialize LCD display by sending proper values to data bus. Divided into four groups: function set, display on/off control and entry mode set. During this process LDC_RS is set to 0, giving information that instructions are sent to module. All elements of initialization were set according to display manual.
- **send_command:** subroutine used set LCD_E on negative edge. Only then instructions will be accepted by display.
- **send_data:** subroutine used to send data (text) to the display. During this procedure LCD_RS is set to 1.
- **us_delay:** created for multiplicity of 1 µs delay, used in initialization and sending data process.
- **ms_delay:** created for multiplicity of 1 ms delay, used at the beginning to wait for proper voltage setting.

Us_delay precision is 1 µs and ms_delay error precision is about 0.3% which is enough for the purpose of this tasks.

When all information is sent to the display the microcontroller goes into Infinite loop.

Despite using one line in LCD display, two line were set as EDSIM51 simulator is not able to work on one line. Same goes with font. Both of them are set in „function set" stage

Task was supposed to present in two version: 8 bit bus and 4 bit bus. Main difference between them is that 4 bit bus can process 4 bits at once, the one that are located in so called high nibble (DB4 – DB7). When simulator starts, 8-bit operation is automatically selected. That is why, when first instruction is sent, it must be done twice in order to change operation mode.

Task 2.

The main goal of this task was to send and display my index number in first line of LCD display starting from 5th position and my name in second line starting from 2nd position.

Program works similarly to 8 bit version from previous task. However, due to the fact that not one letter but whole string need to be sent and cursor was moved, several changes were made:

- **edited send_data:** - each letter is taken from RAM. Both string are separeted by 0 to distinguish them and to tell instruction to which moment it should read the letters. One register is pointing to first letter and it is incrementing when subroutine finish sending letter to display. When 0 value is occured return from this subroutine
- **move_cursor:** new subroutine to move cursor to certain position on display by taking proper value of it from memory map. LCD_RS is set to 0 and BD7 must be set to 1. This base on HD44780 display function: „Set DDRAM address".

Final Questions

1) **How is 4 bit data bus configured in 2x16 LCD display?**

   When 4-bit data mode is used all information are sent in two 4-bit nibbles first high then low using DB4 – DB7. As by default 8 bit mode is used, initialization set must be sent twice in order to tell module to change into 4 bit mode.

2) **Why is it convenient to use 12 MHz system clock with 8051 microcontrollers?**

   8051 core ran 12 clock cycles per one processor cycle (due to cristal frequency). Thanks to that most instruction run in one or two cycles which in our case was 1 μs or 2 μs. It enable to generate time intervals user want to achieve.

3) **Are delay methods presented in the tutorial part of the instruction optimal? Is it possible to generate precise time intervals using loops and NOP commands?**

   No, it is not optimal, Djnz instruction last for two cycles (2 μs) which allows only for even delay and each instruction like mov also take one cycles which decreases precision of such delay. Usage of NOP and loops is not enough as it requires shifting bits (to divide by 2) and substraction to estimate number of cycles which are used to perfrom delay.

4) **How many 2x16 displays can you connect to the AT89C4051 microcontroller without using any additional ICs?**

   To AT84C4051 we can connect up to 10 LCD displays. This microcontroller has 15 pins and display require 6 pins to work. However, all displays can share the bus, RS pin and R/W pin and only E pin must be individual for each display (which allows to use 10 screens).

I declare that this piece of laboratory work which is the basis for recognition of achieving learning outcomes in the Microprocessor Systems (EMISY) course was completed on my own.

Krzysztof Piotrowski
300175