Lab assignment 6: LCD

# Masauso Lungu 209533,

## Preparation task:

1. **Table with LCD signals**

| **Signal(s)** | **Pin(s)** | **Purpose** |
| --- | --- | --- |
| RS | PB0 | Register selection signal. Selection between *Instruction register* (0) and *Data register* (1) |
| R/W | GND | Selecting reading or writing. GND means only writing is enabled |
| E | PB1 | Enable signal for communication |
| D[3:0] | N/A | Data transfer in 8-bit mode. |
| D[7:4] | PD[7:4] | Data transfer in both 8 and 4-bit modes. |

1. **ASCII values**

**CHAR DEC HEX CHAR DEC HEX CHAR DEC HEX**

0 48 30 A 65 41 a 97 61

1 49 31 B 66 42 b 98 62

2 50 32 C 67 43 c 99 63

3 51 33 D 68 44 d 100 64

4 52 34 E 69 45 e 101 65

5 53 35 F 70 46 f 102 66

6 54 36 G 71 47 g 103 67

7 55 37 H 72 48 h 104 68

8 56 38 I 73 49 i 105 69

9 57 39 J 74 4A j 106 6A

.. .... .... K 75 4B k 107 6B

.. .... .... L 76 4C l 108 6C

.. .... .... M 77 4D m 109 6D

.. .... .... N 78 4E n 110 6E

.. .... .... O 79 4F o 111 6F

.. .... .... P 80 50 p 112 70

.. .... .... Q 81 51 q 113 71

.. .... .... R 82 52 r 114 72

.. .... .... S 83 53 s 115 73

.. .... .... T 84 54 t 116 74

.. .... .... U 85 55 u 117 75

.. .... .... V 86 56 v 118 76

.. .... .... W 87 57 w 119 77

.. .... .... X 88 58 x 120 78

.. .... .... Y 89 59 y 121 79

.. .... .... Z 90 5A z 122 7A

## 2. HD44780 communication.

Picture of Time signals between ATmega328p and LCD keypad shield when transmitting “DE2”

DE2 => 0100 0100 \_ 0100 0101\_0011 0010

A picture containing timeline

Description automatically generated

## 3. STOPWATCH

i) **Listings of TIMER2\_OVF\_vect**

/\* Interrupt service routines ----------------------------------------\*/

/\*\*

\* ISR starts when Timer/Counter2 overflows. Update the stopwatch on

\* LCD display every sixth overflow, ie approximately every 100 ms

\* (6 x 16 ms = 100 ms).

\*/

ISR(TIMER2\_OVF\_vect)

{

static *uint8\_t* number\_of\_overflows = 0;

static *uint8\_t* tens = 0; // Tenths of a second

static *uint8\_t* secs = 0; // Seconds

static *uint8\_t* mins = 0; // Minutes

char lcd\_string[] = " "; // String for converting numbers by itoa()

*uint16\_t* secs\_sq = secs\*secs; // Square of seconds

number\_of\_overflows++;

if (number\_of\_overflows >= 6)

{

// Do this every 6 x 16 ms = 100 ms

number\_of\_overflows = 0;

tens++;

if (tens > 9)

{

tens = 0;

secs++;

if (secs > 59)

{

secs = 0;

mins++;

if (mins > 59)

{

mins = 0;

}

}

}

*itoa*(tens, lcd\_string, 10);

lcd\_gotoxy(7, 0);

lcd\_putc(lcd\_string[0]);

// Displaying Seconds

*itoa*(secs, lcd\_string, 10);

lcd\_gotoxy(4, 0);

if (secs < 10)

{

lcd\_putc('0');

lcd\_putc(lcd\_string[0]);

}

else

{

lcd\_puts(lcd\_string);

}

//Displaying minutes

*itoa*(mins, lcd\_string, 10);

lcd\_gotoxy(1, 0);

if (mins < 10)

{

lcd\_putc('0');

lcd\_putc(lcd\_string[0]);

}

else

{

lcd\_puts(lcd\_string);

}

//Displaying square of seconds

*itoa*(secs\_sq, lcd\_string, 10);

lcd\_gotoxy(11, 0);

lcd\_puts(lcd\_string);

if (secs == 0)

{

// Clears the position and reset square of secs back to 0

secs\_sq = 0;

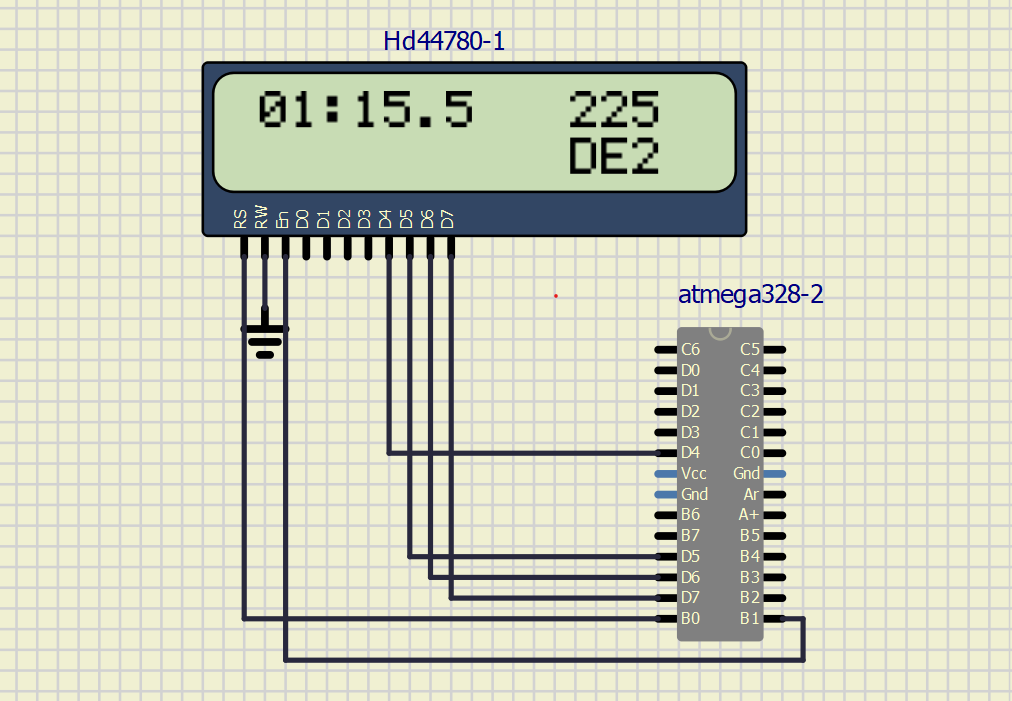
lcd\_puts(" ");

}

}

}

ii**) Screenshort of SimulIDE circuit for stopwatch**



## 4. Progress bar

**i) Listings of TIMER0\_OVF\_vect**

/\*--------------------------------------------------------------------\*/

/\*\*

\* ISR starts when Timer/Counter0 overflows. Update the progress bar on

\* LCD display every 16 ms.

\*/

ISR(TIMER0\_OVF\_vect)

{

static *uint8\_t* number\_of\_overflows = 0;

static *uint8\_t* symbol = 0;

static *uint8\_t* position = 0;

number\_of\_overflows++;

if (number\_of\_overflows >=12) // It takes approximately 12 cycles to fill 1 bar

{

number\_of\_overflows = 0;

symbol++;

if(symbol > 5)

{

symbol = 0;

position++;

if ((position > 7))// Resetting the progress bar when 7th bar filled

{

position = 0;

lcd\_gotoxy(1, 1);

lcd\_puts(" ");

}

}

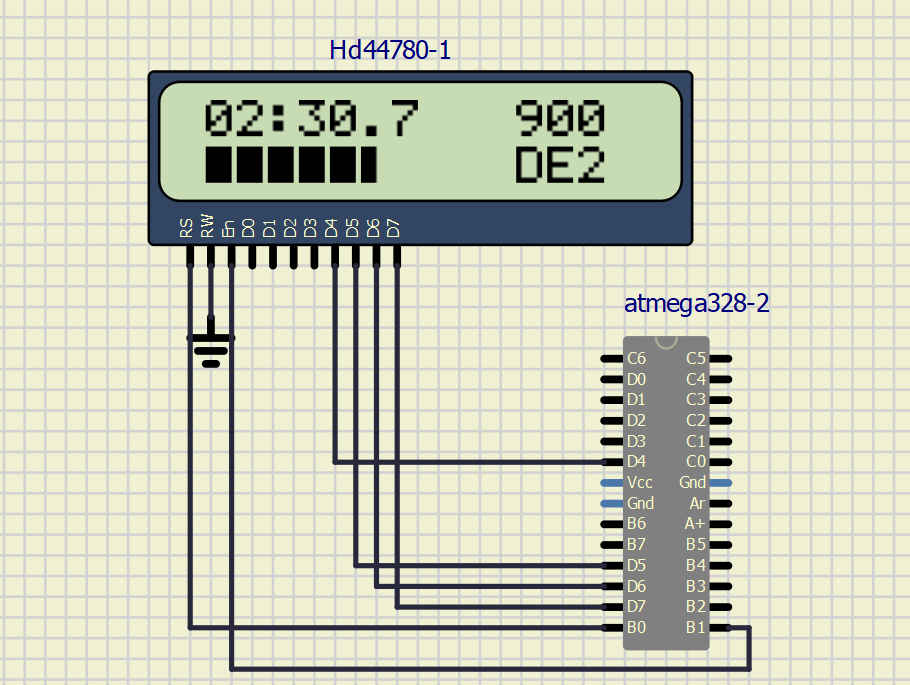
}

lcd\_gotoxy(1 + position, 1);

lcd\_putc(symbol);

}

**ii) Circuit simulation when with progress bar.**



Link to repository: