

# Session 6

-

## Display devices, LCD display

**Author:** Guillermo Cortés Orellana

**Teacher:** Tomáš Frýza

# Lab assignment

## 1. Preparation tasks

- Table with LCD signals

LCD signal (s)	AVR pin(s)	Description
RS	PB0	Register selection signal. Selection between Instruction register (RS=0) and Data register (RS=1)
R/W	GND	Pin writing/reading to/from - LCD
E	PB1	Enabling pin. When this pin is set to logical low, the LCD does not care what is happening with R/W, RS, and the data bus lines. When this pin is set to logical high, the LCD is processing the incoming data
D[3:0]	-	We won't use them. They would only be used if we worked in 8 bits mode
D[7:4]	PD4, PD5, PD6, PD7	Four high order bidirectional tristate data bus pins. Used for data transfer and receive between the MPU and the LCD

- ASCII values

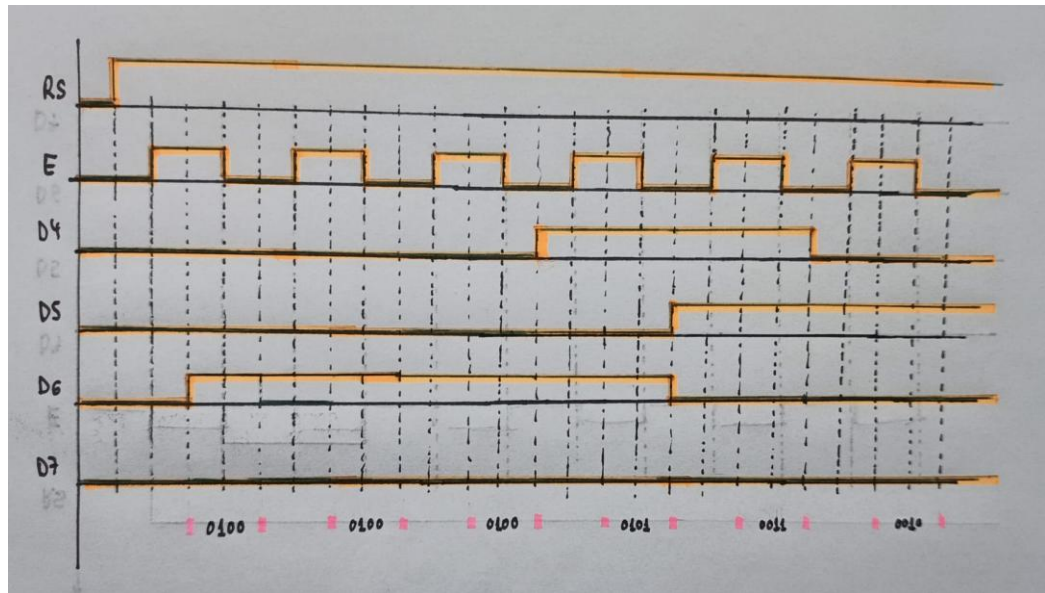
Representation	Binary	Decimal	Hexadecimal
A	0100 0001	65	41
B	0100 0010	66	42
C	0100 0011	67	43
D	0100 0100	68	44
E	0100 0101	69	45
F	0100 0110	70	46
G	0100 0111	71	47
H	0100 1000	72	48
I	0100 1001	73	49
J	0100 1010	74	4A
K	0100 1011	75	4B
L	0100 1100	76	4C
M	0100 1101	77	4D
N	0100 1110	78	4E
O	0100 1111	79	4F
P	0101 0000	80	50
Q	0101 0001	81	51
R	0101 0010	82	52
S	0101 0011	83	53
T	0101 0100	84	54
U	0101 0101	85	55
V	0101 0110	86	56
W	0101 0111	87	57
X	0101 1000	88	58
Y	0101 1001	89	59
Z	0101 1010	90	5A

Representation	Binary	Decimal	Hexadecimal
a	0110 0001	97	61
b	0110 0010	98	62
c	0110 0011	99	63
d	0110 0100	10	64
e	0110 0101	101	65
f	0110 0110	102	66
g	0110 0111	103	67
h	0110 1000	104	68
i	0110 1001	105	69
j	0110 1010	106	6A
k	0110 1011	107	6B
l	0110 1100	108	6C
m	0110 1101	109	6D
n	0110 1110	110	6E
o	0110 1111	111	6F
p	0111 0000	112	70
q	0111 0001	113	71
E	0111 0010	114	72
s	0111 0011	115	73
t	0111 0100	116	74
u	0111 0101	117	75
v	0111 0110	118	76
w	0111 0111	119	77
x	0111 1000	120	78
y	0111 1001	121	79
z	0111 1010	122	7A

Representation	Binary	Decimal	Hexadecimal
0	0011 0000	48	30
1	0011 0001	49	31
2	0011 0010	50	32
3	0011 0011	51	33
4	0011 0100	52	34
5	0011 0101	53	35
6	0011 0110	54	36
7	0011 0111	55	37
8	0011 1000	56	38
9	0011 1001	57	39

- Picture of time signals between ATmega328P and HD44780 (LCD keypad shield) when transmitting data **DE2**

**DE2 = 0100 0100 – 0100 0101 – 0011 0010**



## 2. Stopwatch

- Listing of **TIMER2\_OVF\_vect** interrupt routine with complete stopwatch code (minutes:seconds.tenths) and square value computation

```
/* 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[2] = " ";
    char lcd_sqr[2] = " ";

    number_of_overflows++;

    if (number_of_overflows >= 6)
    {
        // Do this every 6 x 16 ms = 100 ms
        number_of_overflows = 0;

        /*TENTHS*/
        itoa(tens, lcd_string, 10);    // Convert decimal value to string
        lcd_gotoxy(7, 0);
        lcd_puts(lcd_string);

        tens++;

        if(tens >= 10){
            tens = 0;

            secs++;

            /*SQUARE OF SECONDS*/
            itoa(secs*secs, lcd_sqr, 10);    // Convert decimal value to string
            lcd_gotoxy(11, 0);
            lcd_puts(lcd_sqr);

            /*SECONDS*/
            if (secs >= 10){
                itoa(secs, lcd_string, 10);    // Convert decimal value to string
                lcd_gotoxy(4, 0);
                lcd_puts(lcd_string);
            }else{
                itoa(secs, lcd_string, 10);    // Convert decimal value to string
                lcd_gotoxy(5, 0);
                lcd_puts(lcd_string);
            }
        }
    }
}
```

```

/*MINUTES*/
if (secs >= 60){
    secs = 0;
    itoa(secs, lcd_string, 10);    // Convert decimal value to string
    lcd_gotoxy(4, 0);
    lcd_puts(lcd_string);

    itoa(secs*secs, lcd_sqr, 10);    // Convert decimal value to string
    lcd_gotoxy(11, 0);
    lcd_puts(lcd_sqr);

    lcd_gotoxy(12, 0);
    lcd_data(0x20);
    lcd_gotoxy(13, 0);
    lcd_data(0x20);
    lcd_gotoxy(14, 0);
    lcd_data(0x20);

    mins++;
    if (mins >= 60){
        mins = 0;

        lcd_gotoxy(1, 0);
        lcd_putc('0');
        lcd_gotoxy(2, 0);
        lcd_putc('0');

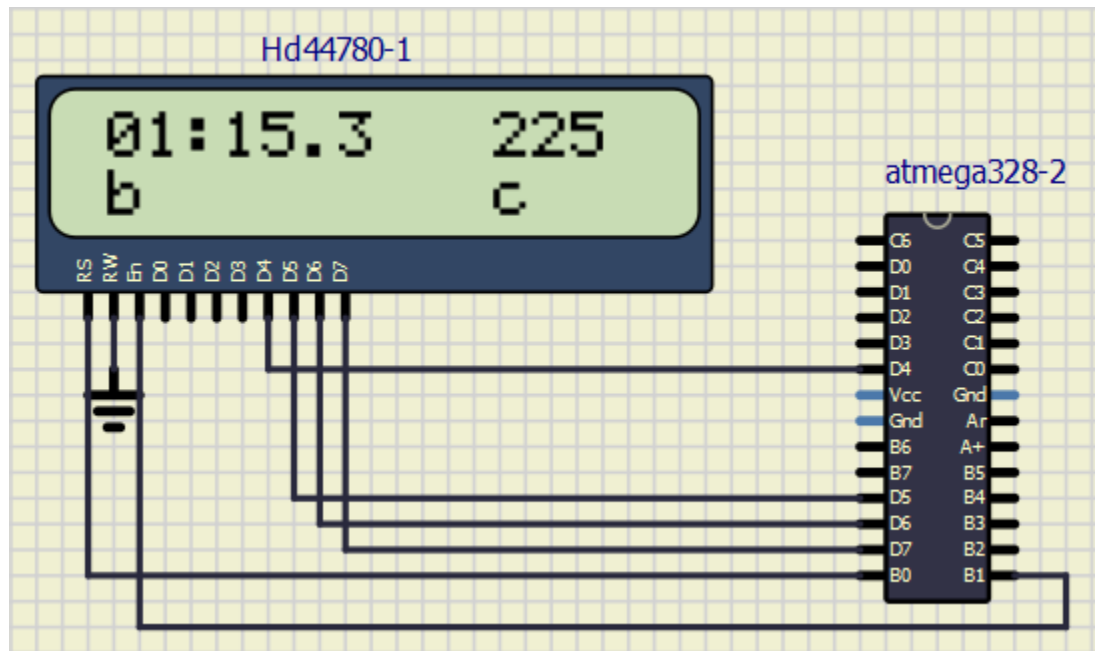
        }else if (mins >= 10){
            itoa(mins, lcd_string, 10);    // Convert decimal value to string
            lcd_gotoxy(1, 0);
            lcd_puts(lcd_string);
        }else {
            itoa(mins, lcd_string, 10);    // Convert decimal value to string
            lcd_gotoxy(2, 0);
            lcd_puts(lcd_string);
        }
    }
}
}
}
}
}

```

You can find the code on my GitHub:

<https://github.com/GuicoRM/Digital-Electronics-2>

- Screenshot of SimulIDE circuit when “Power Circuit” is applied





### 3. Progress bar

- Listing of **TIMER0\_OVF\_vect** interrupt routine with a progress bar

```
/**
 * ISR starts when Timer/Counter0 overflows. Shows
 * bar state, ie approximately every 100 ms
 * (6 x 16 ms = 100 ms).
 */
ISR(TIMER0_OVF_vect)
{
    static uint8_t symbol = 0;
    static uint8_t position = 1;

    symbol++;

    if(symbol >= 6){
        symbol = 0;

        if(position == 11){
            lcd_gotoxy(0, 1);
            lcd_data(0x20);
            lcd_gotoxy(1, 1);
            lcd_data(0x20);
            lcd_gotoxy(2, 1);
            lcd_data(0x20);
            lcd_gotoxy(3, 1);
            lcd_data(0x20);
            lcd_gotoxy(4, 1);
            lcd_data(0x20);
            lcd_gotoxy(5, 1);
            lcd_data(0x20);
            lcd_gotoxy(6, 1);
            lcd_data(0x20);
            lcd_gotoxy(7, 1);
            lcd_data(0x20);
            lcd_gotoxy(8, 1);
            lcd_data(0x20);
            lcd_gotoxy(9, 1);
            lcd_data(0x20);
            lcd_gotoxy(10, 1);
            lcd_data(0x20);
        }

        if(position < 11){
            lcd_gotoxy(position, 1);
            lcd_data(0xff);
        }else{
            position = 1;

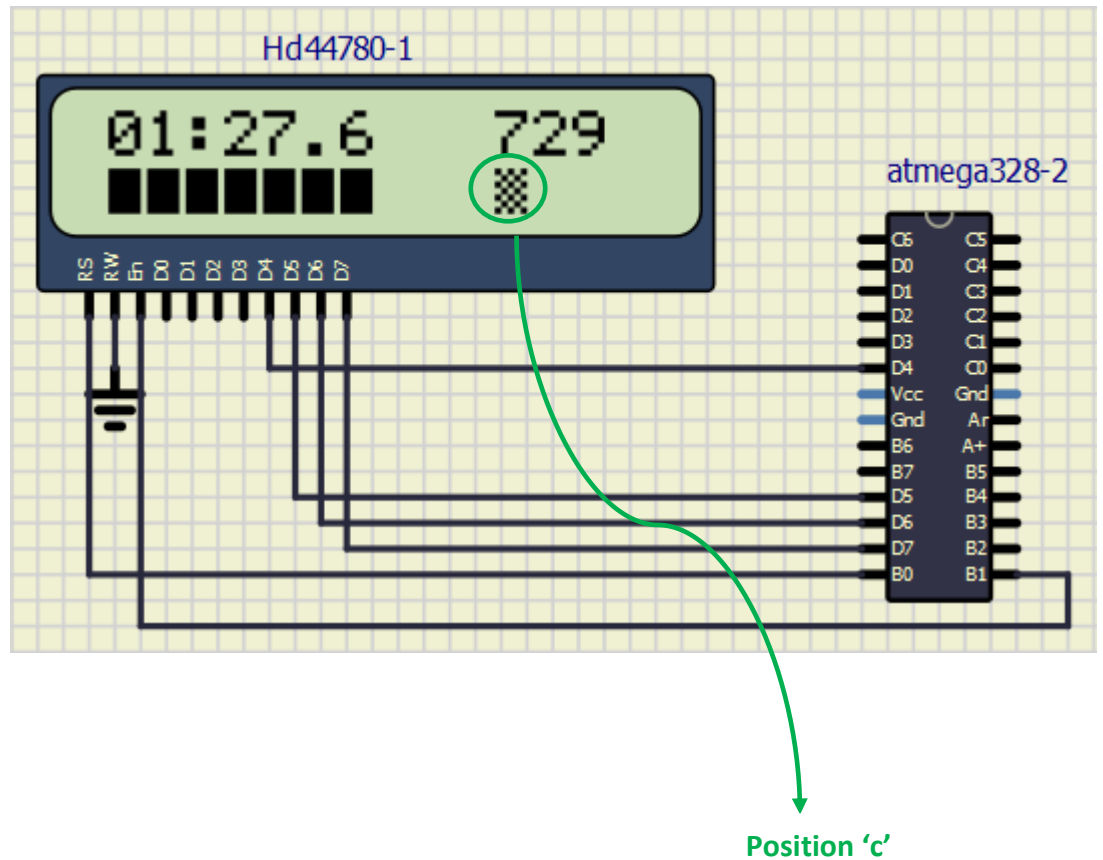
            lcd_gotoxy(position, 1);
            lcd_data(0xff);
        }
        position++;
    }
}
```

You can find the code on my GitHub:

<https://github.com/GuicoRM/Digital-Electronics-2>

- Screenshot of SimulIDE circuit when "Power Circuit" is applied

**Note:** I also added one custome character in position 'c' in order to test it



You can find the code of **custome character** on my GitHub:

<https://github.com/GuicoRM/Digital-Electronics-2>