VUT FEKT

[BPA-DE2] Digital Electronics 2

Assignment 6

GitHub: *https://github.com/ShalaKreshnik*

Name and Surname: *Kreshnik Shala*

Person ID: *226108*

Date: Monday, November 09, 2020

1 Table of Contents

Lab assignment 6

VUT FEKT

[BPA-DE2] Digital Electronics 2

Assignment 6

GitHub: *https://github.com/ShalaKreshnik*

Name and Surname: *Kreshnik Shala*

Person ID: *226108*

Date: Monday, November 09, 2020

1.1 Preparation tasks. Submit:

1.1.1 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	Write data signal (R/W=0), read data signal (R/W=1), pin is GND -> only write
E	PB1	Enable signal, falling edge starts communication
D[3:0]	Not used	Data signals, possible for 8 bit communication
D[7:4]	PD7:PD4	Data signals, 4 bit communication, words are sent in 2 halves (2 E signals needed)

1.1.2 Table With Library for HD44780 based LCDs

Function name	Function parameters	Description	Example
lcd_init	LCD_DISP_OFF LCD_DISP_ON LCD_DISP_ON_CURSOR LCD_DISP_ON_CURSOR_BLINK	Initialize display and select type of cursor.	lcd_init(LCD_DISP_OFF);
lcd_clrscr	none	Clear display and set cursor to home position.	lcd_clrscr();
lcd_gotoxy	x horizontal position (0: left most position) y vertical position (0: first line)	Set cursor to specified position.	lcd_gotoxy(x,y);
lcd_putc	c character to be displayed	Display character at current cursor position.	lcd_putc(c);
lcd_puts	s string to be displayed	Display string without auto linefeed.	lcd_puts(s);
lcd_command	cmd instruction to send to LCD controller, see HD44780 data sheet	Send LCD controller instruction command.	lcd_command(cmd);
lcd_data	data byte to send to LCD controller, see HD44780 data sheet	Send data byte to LCD controller.	lcd_data(data);

Name and Surname: *Kreshnik Shala*
Person ID: *226108*

Date: Monday, November 09, 2020

1.1.3 ASCII values:

ASCII stands for **A**merican **S**tandard **C**ode for **I**nformation **I**nterchange. Computers can only understand numbers, so an ASCII code is the numerical representation of a character such as 'a' or '@' or an action of some sort.

Decimal	Octal	Hexadecimal	Character
048	060	30	0
049	061	31	1
050	062	32	2
051	063	33	3
052	064	34	4
053	065	35	5
054	066	36	6
055	067	37	7
056	070	38	8
057	071	39	9

Name and Surname: *Kreshnik Shala*

Person ID: *226108*

Date: Monday, November 09, 2020

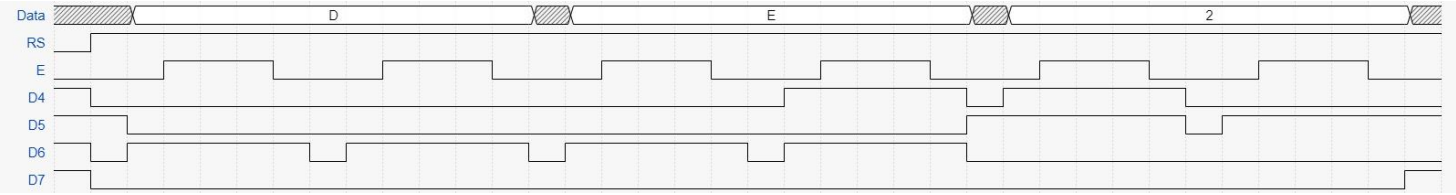
Decimal	Octal	Hexadecimal	Character	Decimal	Octal	Hexadecimal	Character
065	101	41	A	097	141	61	a
066	102	42	B	098	142	62	b
067	103	43	C	099	143	63	c
068	104	44	D	100	144	64	d
069	105	45	E	101	145	65	e
070	106	46	F	102	146	66	f
071	107	47	G	103	147	67	g
072	110	48	H	104	150	68	h
073	111	49	I	105	151	69	i
074	112	4A	J	106	152	6A	j
075	113	4B	K	107	153	6B	k
076	114	4C	L	108	154	6C	l
077	115	4D	M	109	155	6D	m
078	116	4E	N	110	156	6E	n
079	117	4F	O	111	157	6F	o
080	120	50	P	112	160	70	p
081	121	51	Q	113	161	71	q
082	122	52	R	114	162	72	r
083	123	53	S	115	163	73	s
084	124	54	T	116	164	74	t
085	125	55	U	117	165	75	u
086	126	56	V	118	166	76	v
087	127	57	W	119	167	77	w
088	130	58	X	120	170	78	x
089	131	59	Y	121	171	79	y
090	132	5A	Z	122	172	7A	z

Name and Surname: *Kreshnik Shala*
Person ID: *226108*

Date: Monday, November 09, 2020

1.2 HD44780 communication. Submit:

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



1.3 Stopwatch. Submit:

1.3.1 Listing of TIMER2_OVF_vect interrupt routine with complete stopwatch code (minutes:seconds.tenths) and square value computation:

```
ISR(TIMER2_OVF_vect)
{
    static uint8_t number_of_overflows = 0;
       static uint8_t tens = 0; // Tenth of a second
       static uint8_t secs = 0; // Seconds
       static uint8_t mins = 0; // Minutes
       char lcd_string[2] = "00"; // Strings for converting numbers by itoa


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

              tens++; // Update tenth of a second
              if(tens >= 10)
              {
```

```
tens = 0;



// Updating secs
secs++;
if (secs >31) // Square is of 4 digits (i.e. 32*32 = 1024)
{
	lcd_gotoxy(COL2, 0); lcd_putc(((secs*secs)/1000)+48);
	lcd_gotoxy(COL2+1, 0); lcd_putc((((secs*secs)/100)%10)+48);
	lcd_gotoxy(COL2+2, 0);
	lcd_putc(((((secs*secs)/10)%10)%10)+48);
	lcd_gotoxy(COL2+3, 0);
	lcd_putc(((((secs*secs)%100)%10)%10)+48);
}

else if(secs>9) // Square is of 3 digits (i.e. 13*13 = 169)
{
	lcd_gotoxy(COL2, 0); lcd_putc(((secs*secs)/100)+48);
	lcd_gotoxy(COL2+1, 0); lcd_putc((((secs*secs)/10)%10)+48);
	lcd_gotoxy(COL2+2, 0); lcd_putc((((secs*secs)%10)%10)+48);
}

else if(secs>3) // Square is of 2 digits (i.e. 4*4 = 16)
{
	lcd_gotoxy(COL2, 0); lcd_putc(((secs*secs)/10)+48);
	lcd_gotoxy(COL2+1, 0); lcd_putc(((secs*secs)%10)+48);
}

else
{
lcd_gotoxy(COL2, 0); lcd_putc(((secs*secs)+48));
}

if(secs >= 60)
{
	secs = 0;
	// Updating minutes
	mins++;
	if(mins >= 60)
	{
		mins = 0;
	}

	// Display minutes
	lcd_gotoxy(COL1, 0);
	if(mins < 10)
	{
		lcd_putc('0');
	}
	itoa(mins, lcd_string, 10);
	lcd_puts(lcd_string);
```

Name and Surname: *Kreshnik Shala*
Person ID: *226108*

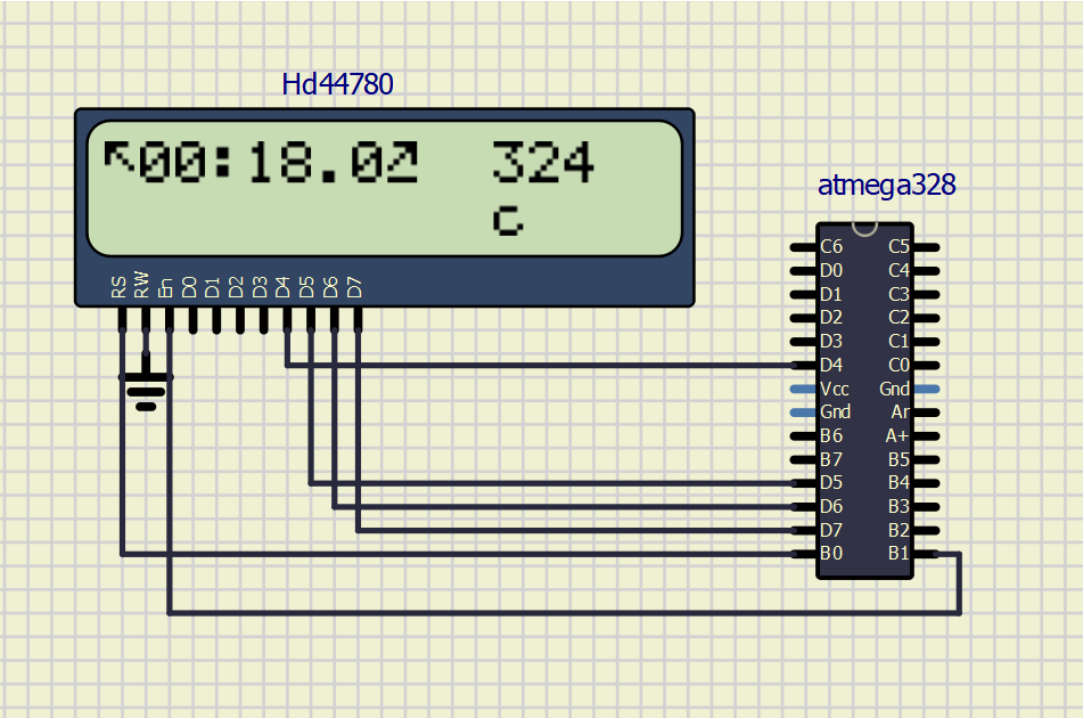
Date: Monday, November 09, 2020

```
                        //Clearing seconds^2
                        lcd_gotoxy(COL2, 0);
                        lcd_puts("    ");

                        lcd_gotoxy(COL2, 0);


                        lcd_putc(((secs*secs)+48));
                }
                // Display seconds
                lcd_gotoxy(COL1+3,0);
                if (secs < 10)
                {
                        lcd_putc('0');
                }
                itoa(secs, lcd_string, 10);
                lcd_puts(lcd_string);
        }

        // Display hundredth of a second
        lcd_gotoxy(COL1+6, 0);

        // Convert cnt0 in decimal to string
        itoa(tens, lcd_string, 10);
        lcd_puts(lcd_string);

    }
}
```

Name and Surname: *Kreshnik Shala*
Person ID: *226108*

Date: Monday, November 09, 2020

1.3.2 Screenshot of SimulIDE circuit when "Power Circuit" is applied:



Name and Surname: *Kreshnik Shala*
Person ID: *226108*

Date: Monday, November 09, 2020

1.4 Progress bar. Submit:

1.4.1 Listing of TIMER0_OVF_vect interrupt routine with a progress bar:

```
ISR (TIMER0_OVF_vect)

{
	static uint8_t symbol = 0;
	static uint8_t position = 0;

	lcd_gotoxy(COL1 + position, 1);
	lcd_putc(symbol);

	symbol++;
	if (symbol >= 6)
	{
		symbol = 0;
		// Moving to next position
		position++;
		if (position >= 10)
		{
			position = 0;
			// Clearing progress bar
			lcd_gotoxy(COL1, 1);
			lcd_puts("          ");
		}
	}
}
```

Name and Surname: *Kreshnik Shala*
Person ID: *226108*

Date: Monday, November 09, 2020

1.4.2 Screenshot of SimulIDE circuit when "Power Circuit" is applied.