**LAB6 home assignment Lukáš Kaleta**

1. Preparation tasks (done before the lab at home). Submit:
   * 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 | Choosing between reading (R/W = 1) or writing (R/W = 0 or GND) |
| E | PB1 | On the falling edge of E signal data transfers ito the LCD |
| D[3:0] | - | Lower 4 data pins. Used only for 8bit communication |
| D[7:4] | PD[7:4] | Upper 4 data pins. Usable for both 4bit and 8bit communication |

| **Function name** | **Function parameters** | **Description** | **Example** |
| --- | --- | --- | --- |
| lcd\_init | LCD\_DISP\_OFF LCD\_DISP\_ON LCD\_DISP\_ON\_CURSOR LCD\_DISP\_ON\_CURSOR\_BLINK | display off  display on, cursor off display on, cursor on disp on, cursor flashing | lcd\_init(LCD\_DISP\_OFF); lcd\_init(LCD\_DISP\_ON);  lcd\_init(LCD\_DISP\_ON\_CURSOR);  lcd\_init(LCD\_DISP\_ON\_CURSOR\_BLINK); |
| lcd\_clrscr | - | clear screen | lcd\_clrscr(); |
| lcd\_gotoxy | x, y | Set cursor to specified position. | lcd\_gotoxy(1,0); |
| lcd\_putc | c | Wtrite one symbol on current position | lcd\_putc(a); |
| lcd\_puts | “string“ | Wtrite given string on current position | lcd\_putc(“DE\_2“); |
| lcd\_command | cmd | Send LCD controller instruction command | lcd\_command(uint8\_t cmd); |
| lcd\_data | data | Send data byte to LCD controller | lcd\_data(uint8\_t data) |

* + ASCII values.

A = $41

B = $42

…

Z = $5A

a = $61

…

z = $7A

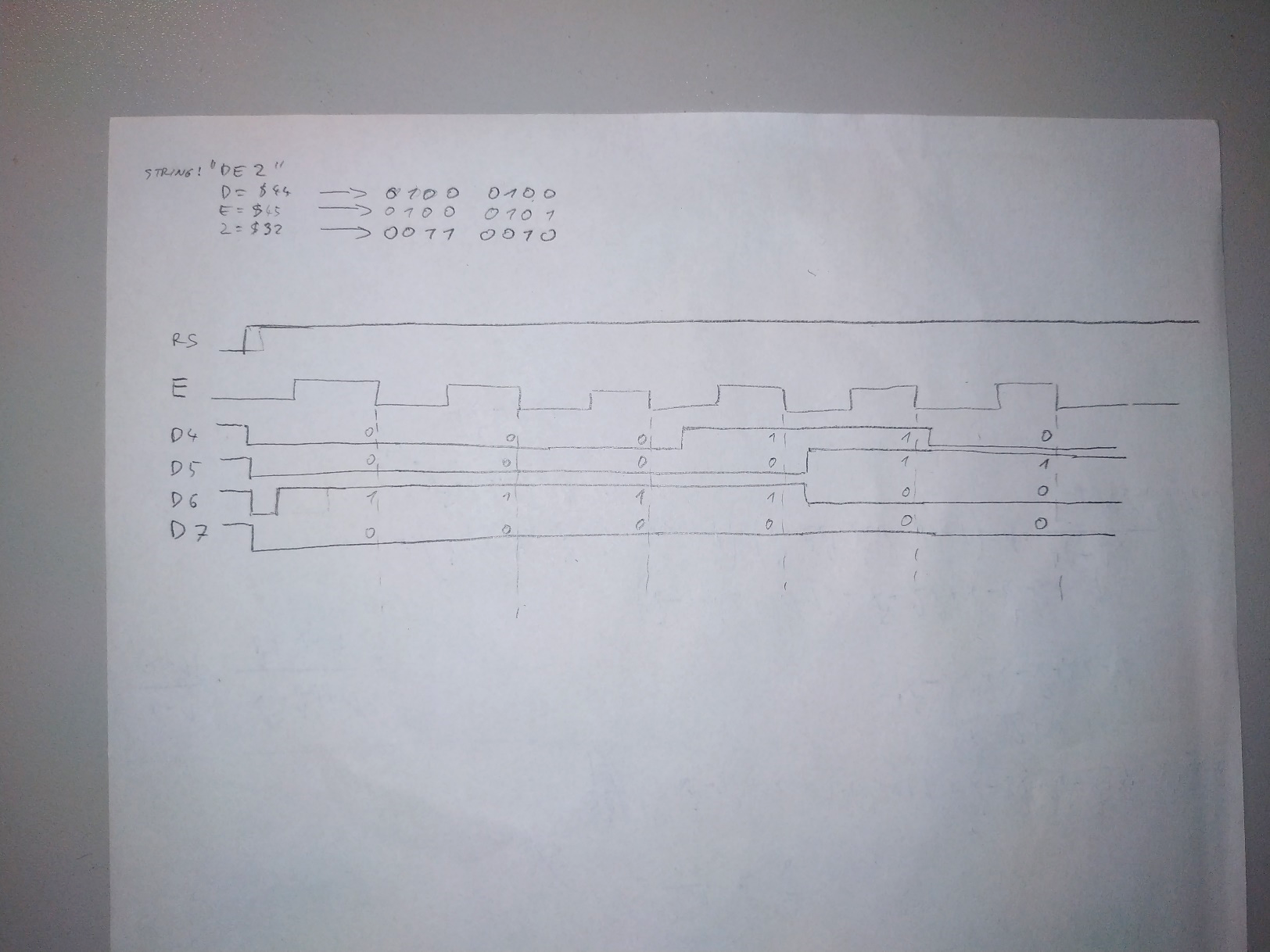
0 = $30

1 = $31

…

9 = $39

1. HD44780 communication. Submit:
   * Picture of time signals between ATmega328P and HD44780 (LCD keypad shield) when transmitting data DE2.



1. Stopwatch. Submit:
   * 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

static *uint16\_t* secs\_square = 0; // squared number of secs on position A

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

number\_of\_overflows++;

if (number\_of\_overflows >= 5)

{

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

number\_of\_overflows = 0;

// WRITE YOUR CODE HERE

tens++;

if(tens > 9)

{

tens = 0;

secs++;

if(secs > 9)

{

if(secs > 59)

{

secs = 0;

lcd\_gotoxy(5, 0);

lcd\_puts("0");

mins++;

if(mins > 9)

{

if(mins > 59)

{

mins = 0;

lcd\_gotoxy(2, 0);

lcd\_puts("0");

}

// displaying > 9 minutes

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

lcd\_gotoxy(1, 0);

lcd\_puts(lcd\_string);

}

else

{

// displaying < 10 minutes

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

lcd\_gotoxy(2, 0);

lcd\_puts(lcd\_string);

}

// clearing square of secs

lcd\_gotoxy(11, 0);

lcd\_puts(" ");

}

// displaying > 9 seconds

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

lcd\_gotoxy(4, 0);

lcd\_puts(lcd\_string);

}

else

{

// displaying < 10 seconds

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

lcd\_gotoxy(5, 0);

lcd\_puts(lcd\_string);

}

// computing square of secs

secs\_square = secs \* secs;

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

lcd\_gotoxy(11, 0);

lcd\_puts(lcd\_string)

}

// displaying tenths of seconds

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

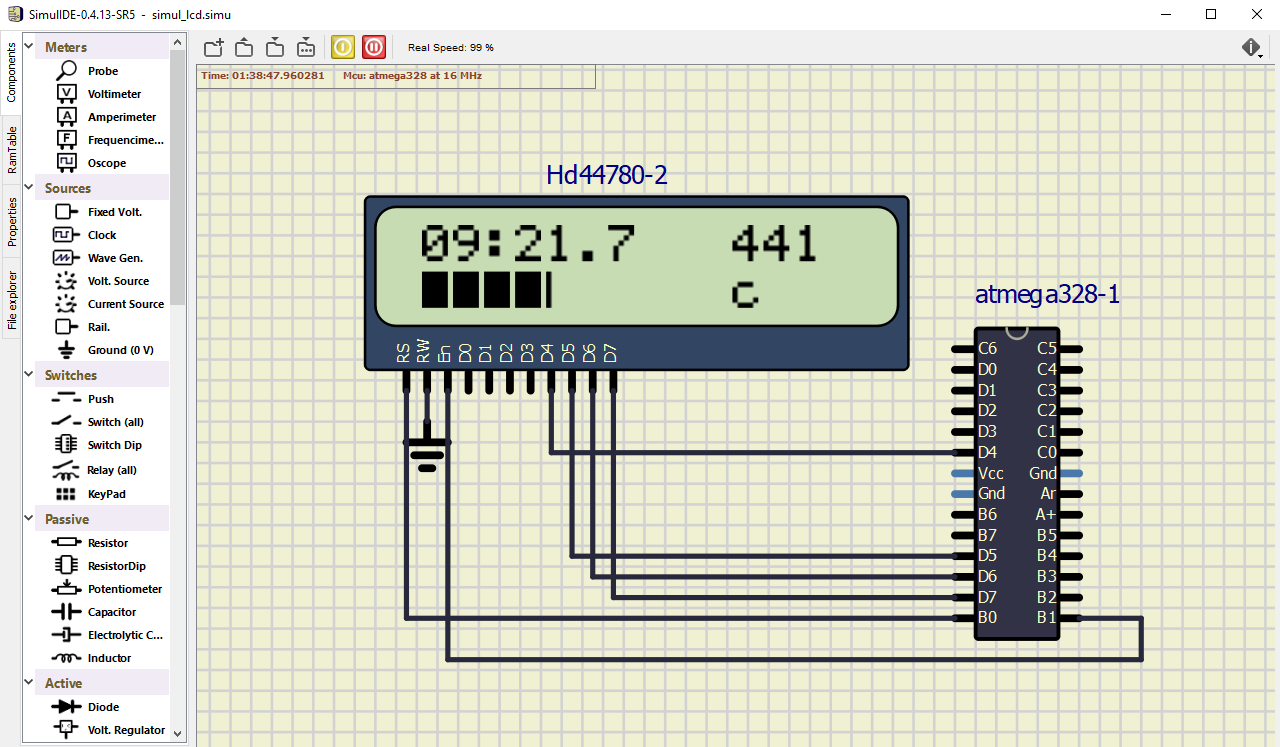
lcd\_gotoxy(7, 0);

lcd\_puts(lcd\_string);

}

}

* + Screenshot of SimulIDE circuit when "Power Circuit" is applied.



1. Progress bar. Submit:
   * Listing of TIMER0\_OVF\_vect interrupt routine with a progress bar,

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

/\*\*

\* 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 = 1;

number\_of\_overflows++;

if (number\_of\_overflows >= 6)

{

number\_of\_overflows = 0;

symbol++;

if (symbol > 5) // one bar is full

{

symbol = 1;

position++;

if (position > 9)

{

position = 1; // clear position

lcd\_gotoxy(1, 1); // clear all bars

lcd\_puts(" ");

}

}

// display bar

lcd\_gotoxy(position, 1);

lcd\_putc(symbol);

}

}

* + Screenshot of SimulIDE circuit when "Power Circuit" is applied.

