VUT FEKT

Name and Surname: *Kreshnik Shala*

[BPA-DE2] Digital Electronics 2

Person ID: *226108*

Assignment 7

Date: Tuesday, November 17, 2020

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

1 Table of Contents

Lab assignment 7

1.1 Preparation tasks. Submit:

1.1.1 Table with voltage divider, calculated, and measured ADC values for all buttons.

Push button	PC[A0] voltage	ADC value (calculated)	ADC value (measured)
Right	0V	0	0
Up	0.495V	101	101
Down	1.203V	246	245
Left	1.970V	403	402
Select	3.182V	651	650
none	5V	1023	1022

1.1.2 Analog to Digital Converter Description

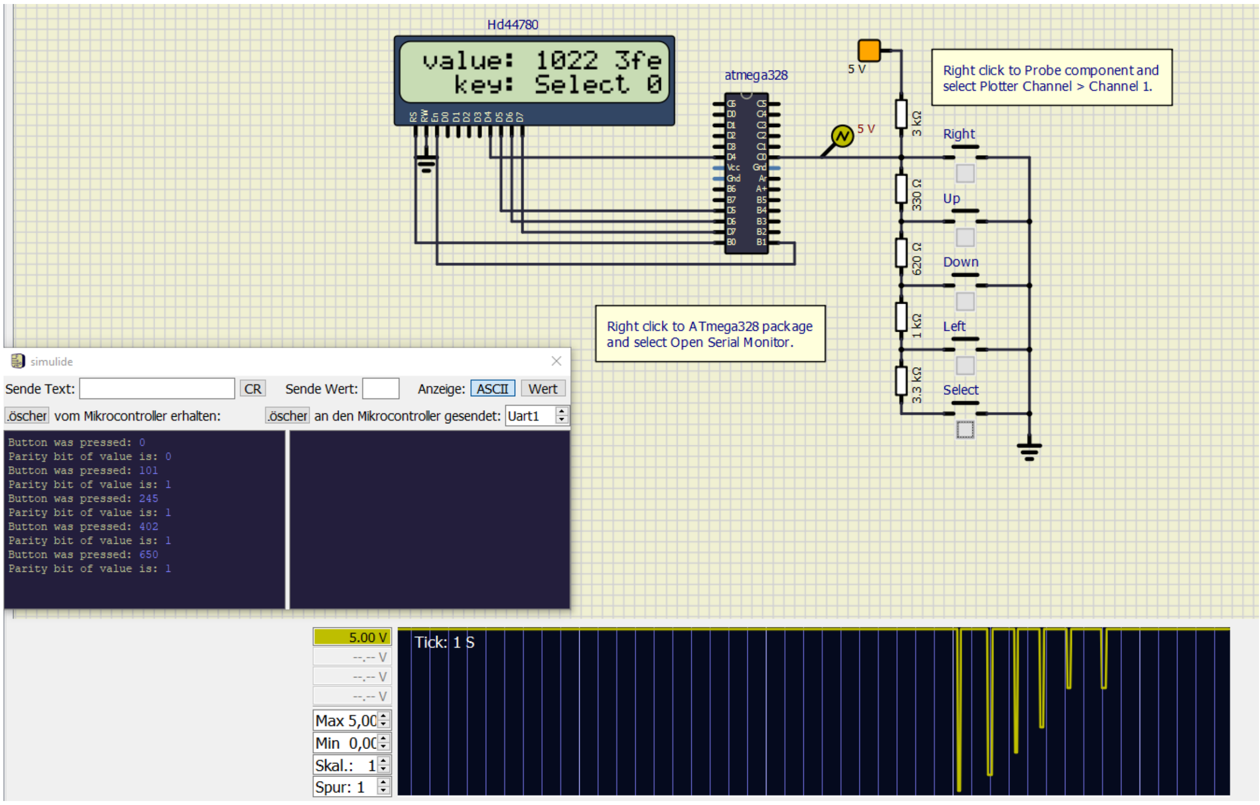
Operation	Register(s)	Bit(s)	Description
Voltage Reference	ADMUX	REDS1:0	01:AVcc voltage reference, 5V
Input Chanel	ADMUX	MUX3:0	0000: ADC0, 0001: ADC1, …
ADC enable	ADCSRA	ADEN	Writing this bit to one/ zero enable/ disable the ADC
Start conversion	ADCSRA	ADSC	This bit has to write to one to start conversion
ADC interrupt enable	ADCSRA	ADIE	When this bit is written to one and the I-bit in SREG is set, the ADC Conversion Complete Interrupt is activated.
ADC clcok prescaler	ADCSRA	ADPS2:0	000: Division factor 2,001:2; 010:4, …
ADC result	ADCL and ADCH		When ADCL is read, the ADC Data Register is not updated until ADCH is read. ADC results

1.2 ADC. Submit:

1.2.1 Listing of ADC_vect interrupt routine with complete code for sending data to the LCD/UART and identification of the pressed button.

```
ISR(ADC_vect)
{
    // WRITE YOUR CODE HERE
    uint16_t value = 0;
    char lcd_string[4] = "0000";

       // Displaying parity bit of value
       parity_display(value);

       // Sending data to UART1
       uint8_t c;
       c = uart_getc();
       if(c != '\0')
       {
              if (c == '1')
              {
              uart_putc(c);
              }
       }


    value = ADC;                     // Copy ADC result to 16-bit variable

    itoa(value, lcd_string, 10); //Convert to string in decimal
    lcd_gotoxy(8, 0);
    lcd_puts("    ");
    lcd_gotoxy(8, 0);
    lcd_puts(lcd_string);

    if (value < 700)
    {
              uart_puts("Button was pressed: ");
              uart_puts(lcd_string);
              uart_puts("\r\n");
    }

    itoa(value, lcd_string, 16); //Convert to string in hexa
    lcd_gotoxy(13, 0);
    lcd_puts("    ");
    lcd_gotoxy(13, 0);

    lcd_puts(lcd_string);


       lcd_gotoxy(8, 1);
       if (value == 0) // If the right button was pressed
    {
        lcd_puts("      ");
        lcd_gotoxy(8, 1);
        lcd_puts("Right");
        parity_display(value);
    }
    if (value == 101) // If the right button was pressed
    {
        lcd_puts("      ");
        lcd_gotoxy(8, 1);
        lcd_puts("Up");
        parity_display(value);
    }
    if (value == 245) // If the right button was pressed
    {
        lcd_puts("      ");
        lcd_gotoxy(8, 1);
        lcd_puts("Down");
        parity_display(value);
    }
    if (value == 402) // If the right button was pressed
    {
        lcd_puts("      ");
        lcd_gotoxy(8, 1);
        lcd_puts("Left");
        parity_display(value);
    }
    if (value == 650) // If the right button was pressed
    {
        lcd_puts("      ");
        lcd_gotoxy(8, 1);
        lcd_puts("Select");
        parity_display(value);
    }

    // Displaying parity data on UART
    if (value < 700)
    {
        uart_puts("Parity bit of value is: ");
        uart_putc(parity+48);
        uart_puts("\r\n");
    }

}
```

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



```
value: 1022 3fe
  key: Select 0
```

Right click to Probe component and select Plotter Channel > Channel 1.

Right click to ATmega328 package and select Open Serial Monitor.

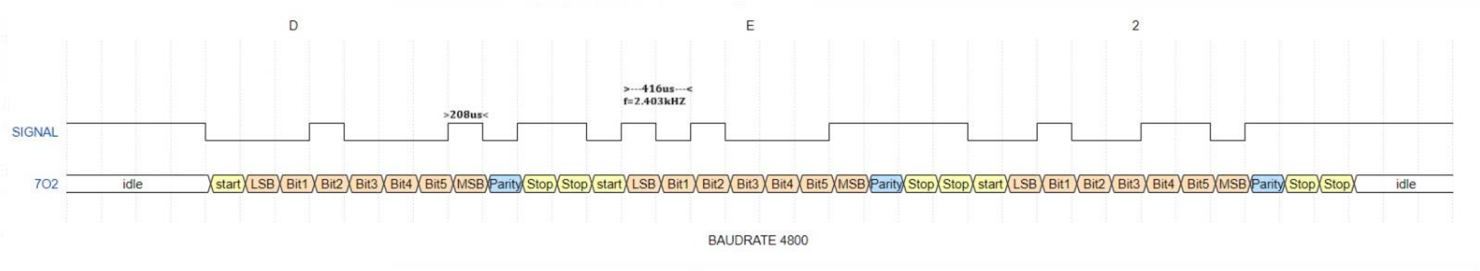
vom Mikrocontroller erhalten:

```
Button was pressed: 0
Parity bit of value is: 0
Button was pressed: 101
Parity bit of value is: 1
Button was pressed: 245
Parity bit of value is: 1
Button was pressed: 402
Parity bit of value is: 1
Button was pressed: 650
Parity bit of value is: 1
```

an den Mikrocontroller gesendet:

1.3 UART. Submit:

1.3.1 (Hand-drawn) picture of UART signal when transmitting data DE2 in 4800 7O2 mode (7 data bits, odd parity, 2 stop bits, 4800 Bd),



1.3.2 Description of UART functions

Function name	Function parameters	Description	Example
uart_init	UART_BAUD_SELECT (9600, F_CPU)	Initialize UART to 8N1 and set baudrate to 9600 Bd	uart_init(UART_BAUD_SELECT(9600, F_CPU));
uart_getc	void	Get received byte from ringbuffer	unsigned int uart getc(void)
uart_putc	unsigned char data	Put byte to ringbuffer for transmitting via UART	void uart putc(unsigned char data)
uart_puts	s string to be transmitted	Put string to ringbuffer for transmitting via UART	void uart puts(const char *s)

1.3.3 Listing of code for calculating/displaying parity bit.

```
uint8_t parity = 0;

// Hex values of every single bit  for example 0b0000000000000010 is defined as(0x0002)
and 0b1000000000000000 is defined as (0x8000)
uint16_t arr[16] = { 0x0001, 0x0002, 0x0004, 0x0008, 0x0010, 0x0020, 0x0040, 0x0080,
       0x0100, 0x0200, 0x0400, 0x0800, 0x1000, 0x2000, 0x4000, 0x8000
};
void parity_display(int a)
{

              uint8_t parity_count = 0;

                     for (uint16_t i =0; i<16; i++)
                     {
                            if ((arr[i]&a)!=0)
                            {
                                   parity_count++; // counting number of 1s in a value
                            }

                     }
                     if(parity_count == 0) // If the count is 0 (No 1 bit)
                     {
                                   parity = 0;
                     }
                     else if(parity_count%2 == 0) // If sum of 1's is an even number
                     {
                            parity = 1;
                     }

                     else if(parity_count%2 == 1) // If the sum of 1's is an odd number
                     {
                            parity = 0;
                     }

                     lcd_gotoxy(15, 1); // Parity display location
                     lcd_putc(parity+48); // Converting to character
};
```