Nadir Osman Al-Wattar
225796

Lab Assignment 7

Link to my Github

$$ADC = \frac{V_{PC0[A0]}}{V_{ref}} * 2^n - 1 =$$

Table 1: Voltage Divider, calculated, and measured ADC values for all buttons

Push button	PC0[A0] voltage	ADC value (calculated)	ADC value (measured)
Right	0 V	0	0
Up	0.495 V	101	101
Down	1.203 V	246	245
Left	1.970 V	403	402
Select	3.182 V	651	650
none	5 V	1023	1022

Table 2: Analog to Digital Converter Description

Operation	Register(s)	Bit(s)	Description
Voltage Reference	ADMUX	REDS1:0	01: AVcc voltage reference, 5V
Input channel	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 clock 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

Table 3: 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)

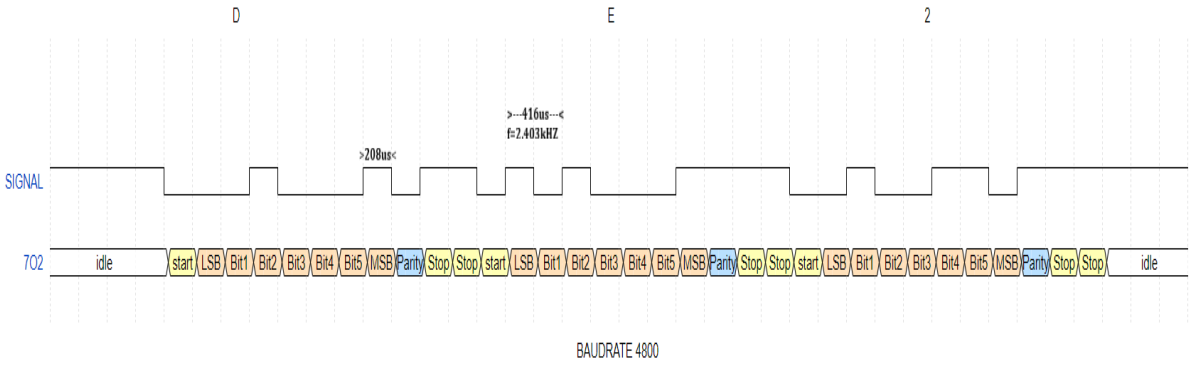

Figure 1: UART Signal when transmitting data DE2 in 4800 702 mode

Listing of ADC vect interrupt routine

```
ISR(ADC_vect)
{

    uint16_t value = 0;
    char lcd_string[4] = "0000";
    // int i = 0;
    // int j = 0;
    // uint8_t count_ones = 0;
    // uint16_t reminder[16];

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


    itoa(value, lcd_string, 10);    // Convert to string in decimal
    lcd_gotoxy(8,0);
    // Clear the position
    lcd_puts("    ");
    lcd_gotoxy(8,0);
    // Update the position
    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 hex
    lcd_gotoxy(13,0);
    lcd_puts("   ");
    lcd_gotoxy(13,0);
    lcd_puts(lcd_string);

    // Approximately +- 5 Interval
    if ((value >= 0) | (value < 5))
    {
        lcd_gotoxy(8,1);
        lcd_puts("      ");
        lcd_gotoxy(8,1);
        lcd_puts("Right");
    }
    if((value > 95) && (value < 105) )
    {
        lcd_gotoxy(8,1);
        lcd_puts("      ");
        lcd_gotoxy(8,1);
        lcd_puts("Up");
    }
    if((value > 240) && (value < 250))
    {
        lcd_gotoxy(8,1);
        lcd_puts("      ");
        lcd_gotoxy(8,1);
        lcd_puts("Down");

    }
```

```
    if((value > 395) && (value < 410))
    {
        lcd_gotoxy(8,1);
        lcd_puts("      ");
        lcd_gotoxy(8,1);
        lcd_puts("Left");


    }
    if((value > 645) && (value < 655))
    {
        lcd_gotoxy(8,1);
        lcd_puts("      ");
        lcd_gotoxy(8,1);
        lcd_puts("Select");

    }
    if (value >= 1015)
    {
        lcd_gotoxy(8,1);
        lcd_puts("      ");
        lcd_gotoxy(8,1);
        lcd_puts("None");

    }
}
```

Listing of code for calculating/displaying parity bit.

```
    // Calculating parity bit
    for(i=0; value>0; i++)
    {

        reminder[i]=value%2; // store the 1´s and 0´s
        value=value/2;

    }
    for (j=i-1 ;j>0; j--)
    {
        if (reminder[j] == 1)
        {
            count_ones++;  // we want to count only the 1´s
        }

    }

    lcd_gotoxy(15,1);
    if ((count_ones%2) == 1) // if the result of this is equal 1 then it has an ODD
number of 1´s
    {

        lcd_puts("1");

    }
    else
    {
        lcd_puts("0");    // EVEN

    }
```

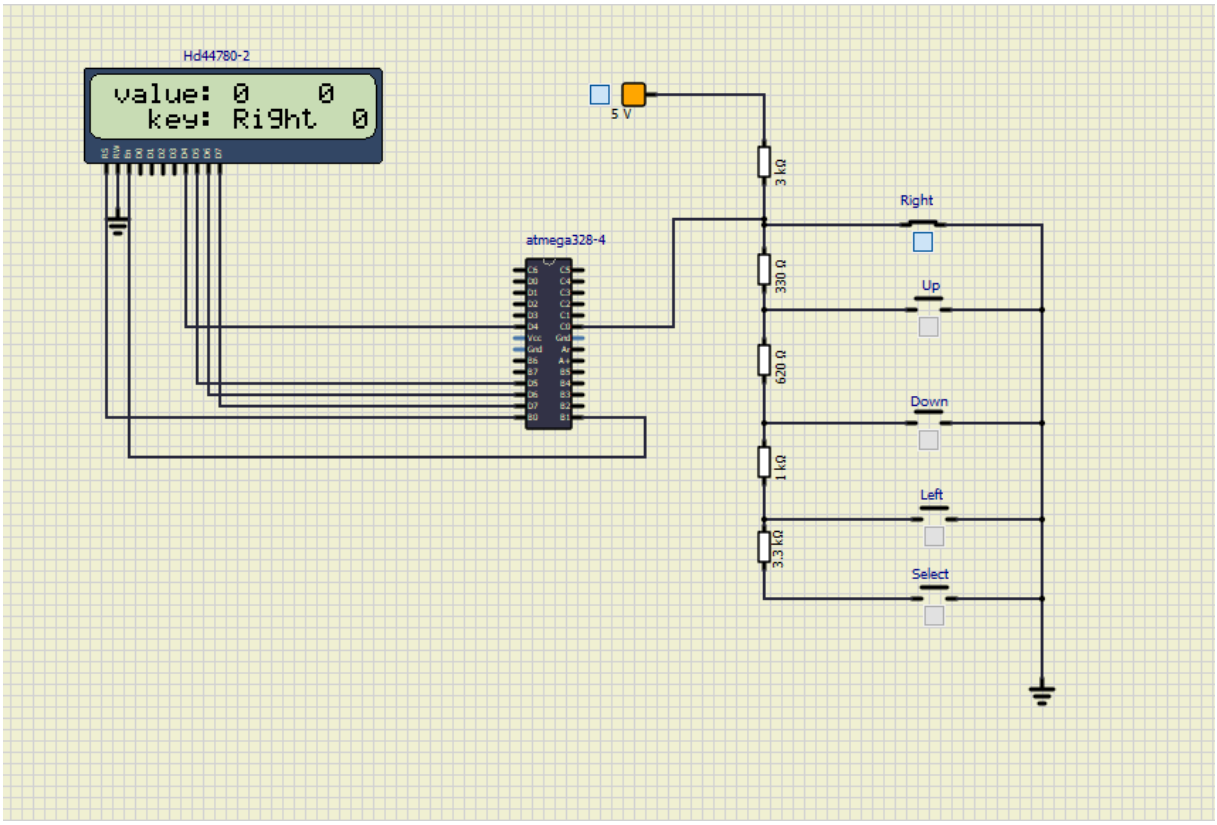

Figure 2: Screenshot of SimulIDE circuit when "Power Circuit" is applied.