VUT FEKT

[BPA-DE2] Digital Electronics 2

Assignment 2

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

Name and Suriname: *Kreshnik Shala*

Person ID: *226108*

Date: Tuesday, October 06, 2020

1 Table of Contents

Lab assignment 2

VUT FEKT

[BPA-DE2] Digital Electronics 2

Assignment 2

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

Name and Suriname: *Kreshnik Shala*

Person ID: *226108*

Date: Tuesday, October 06, 2020

1.1 LED example. Submit:

1.1.1 Tables for DDRB, PORTB, and their combination

1.1.1.1 Table DDRB:

DDRB	Description
0	Input pin
1	Output pin

1.1.1.2 Table PORTB:

PORTB	Description
0	Output low value
1	Output high value

1.1.1.3 Table Combination of DDRB and PORTB

DDRB	PORTB	Direction	Internal pull-up resistor	Description
0	0	input	No	Tri-state, high-impedance
0	1	Input	Yes	Tri-state (Hi-Z)
1	0	Output	No	Output Low (Sink)
1	1	Output	No	Output High (Source)

VUT FEKT

Name and Suriname: *Kreshnik Shala*

[BPA-DE2] Digital Electronics 2

Person ID: *226108*

Assignment 2

Date: Tuesday, October 06, 2020

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

1.1.2 Table with input/output pins available on ATmega328P,

Port	Pin	Input/output usage?
A	X	Microcontroller ATmega328P does not contain port A
B	0	Yes (Arduino pin 8)
B	1	Yes (Arduino pin ~9)
B	2	Yes (Arduino pin ~10)
B	3	Yes (Arduino pin ~11)
B	4	Yes (Arduino pin 12)
B	5	Yes (Arduino pin 13)
B	6	No, [From DATASHEET of ATmega328P]
B	7	No, [From DATASHEET of ATmega328P]
C	0	Yes (Arduino pin A0)
C	1	Yes (Arduino pin A1)
C	2	Yes (Arduino pin A2)
C	3	Yes (Arduino pin A3)
C	4	Yes (Arduino pin A4)
C	5	Yes (Arduino pin A5)
C	6	No [From DATASHEET of ATmega328P]
C	7	X, [From DATASHEET of ATmega328P]
D	0	Yes (Arduino pin RX<-0)
D	1	Yes (Arduino pin TX->1)
D	2	Yes (Arduino pin 2)
D	3	Yes (Arduino pin ~3)
D	4	Yes (Arduino pin 4)
D	5	Yes (Arduino pin ~5)
D	6	Yes (Arduino pin ~6)
D	7	Yes (Arduino pin 7)

VUT FEKT

Name and Suriname: *Kreshnik Shala*

Person ID: *226108*

[BPA-DE2] Digital Electronics 2

Assignment 2

Date: Tuesday, October 06, 2020

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

1.1.3 C code with two LEDs and a push button,

```
/* Defines -----------------------------------------------------------*/
#define LED_GREEN PB5 // AVR pin where green LED is connected
#define LED_RED PC0
#define BTN PD0
#define BLINK_DELAY 500
#ifndef F_CPU
#define F_CPU 16000000 // CPU frequency in Hz required for delay
#endif

/* Includes ----------------------------------------------------------*/
#include <util/delay.h> // Functions for busy-wait delay loops
#include <avr/io.h> // AVR device-specific IO definitions

/* Functions ---------------------------------------------------------*/
/**
* Main function where the program execution begins. Toggle two LEDs
* when a push button is pressed.
*/
int main(void)
{
       /* GREEN LED */
       // Set pin as output in Data Direction Register...
       DDRB = DDRB | (1<<LED_GREEN);
       // ...and turn LED off in Data Register
       PORTB = PORTB & ~(1<<LED_GREEN); // Turn OFF

       /*RED LED*/
       DDRC = DDRC | (1<<LED_RED); // Output
       PORTC = PORTC & ~(1<<LED_RED); // Turn ON
       /*PUSH BUTTON*/
       DDRD = DDRD & ~(1<<BTN); // Input
       PORTD = PORTD | (1<<BTN); // enable internal pull-up


       // Infinite loop
       while (1)
       {
              // Pause several milliseconds
              _delay_ms(BLINK_DELAY);


              if (bit_is_clear(PIND, BTN))
              {

                     PORTB = PORTB ^ (1<<LED_GREEN);
                     PORTC = PORTC ^ (1<<LED_RED);
              }

       }

       // Will never reach this
       return 0;
}
```

VUT FEKT

Name and Suriname: *Kreshnik Shala*

Person ID: *226108*

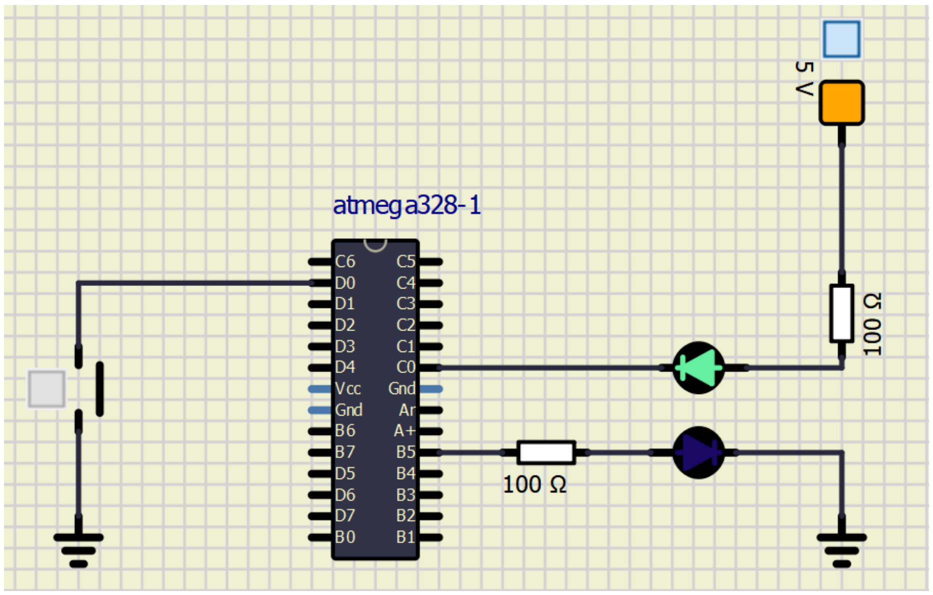
[BPA-DE2] Digital Electronics 2

Assignment 2

Date: Tuesday, October 06, 2020

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

1.1.4 Screenshot_1 of SimulIDE circuit.



1.1.5 Knight Rider application. Submit:

1.1.5.1 C code.

As requested in the code below I have realized the lighting of the lights to work in the "Knight Rider" style.
The realization is done in such a way that as soon as the button is pressed the lighting starts and also as soon as the finger is removed from the button the lighting stops immediately.

```
/*
VUT FEKT                                             Name and Surname: Kreshnik Shala
[BPA-DE2] Digital Electronics 2                      Person ID: 226108
Date: Tuesday, October 06, 2020
GitHub: https://github.com/ShalaKreshnik
*/




/* Defines -----------------------------------------------------------*/
#define LED_GREEN PC0 // AVR pin where green LED is connected
#define LED_RED PC1  // AVR pin where red LED is connected
```

VUT FEKT

[BPA-DE2] Digital Electronics 2

Assignment 2

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

Name and Suriname: *Kreshnik Shala*

Person ID: *226108*

Date: Tuesday, October 06, 2020

```
#define LED_BLUE PC2 // AVR pin where blue LED is connected
#define LED_RED2 PC3 // AVR pin where red2 LED is connected
#define LED_GREEN2 PC4  // AVR pin where green2 LED is connected
#define BTN PD0
#define BLINK_DELAY 500
#define SMALL_DELAY 250
#ifndef F_CPU
#define F_CPU 16000000 // CPU frequency in Hz required for delay
#endif

/* Includes ----------------------------------------------------------*/
#include <util/delay.h> // Functions for busy-wait delay loops
#include <avr/io.h> // AVR device-specific IO definitions

/* Functions ---------------------------------------------------------*/
/**
* Main function where the program execution begins. Toggle two LEDs
* when a push button is pressed.
*/
int main(void)
{
	int LED =0;
	int back=0;
	/* GREEN LED */
	// Set pin as output in Data Direction Register...
	//DDRB = DDRB | (1<<LED_GREEN);
	// ...and turn LED off in Data Register
	//PORTB = PORTB & ~(1<<LED_GREEN); // Turn OFF

	/*RED LED*/
	DDRC = DDRC |(1<<LED_GREEN) | (1<<LED_RED) | (1<<LED_RED2) | (1<<LED_BLUE) |
(1<<LED_GREEN2); // Outputs
	PORTC = PORTC & ~(1<<LED_GREEN); // TURN OFF
	PORTC = PORTC & ~(1<<LED_RED); // Turn OFF
	PORTC = PORTC & ~(1<<LED_RED2); // Turn OFF




	PORTC = PORTC & ~(1<<LED_BLUE); // Turn OFF
	PORTC = PORTC & ~(1<<LED_GREEN2); // Turn OFF
	/*PUSH BUTTON*/
	DDRD = DDRD & ~(1<<BTN); // Input
	PORTD = PORTD | (1<<BTN); // enable internal pull-up

	// Infinite loop
	while (1)
	{
		LED =0;
		back=0;
		// Pause several milliseconds
		//_delay_ms(BLINK_DELAY);
		while (bit_is_clear(PIND,BTN))
		{

			PORTC = PORTC ^ (1<<LED); // TURN ON THE LED
```

Name and Suriname: *Kreshnik Shala*

Person ID: *226108*

Date: Tuesday, October 06, 2020

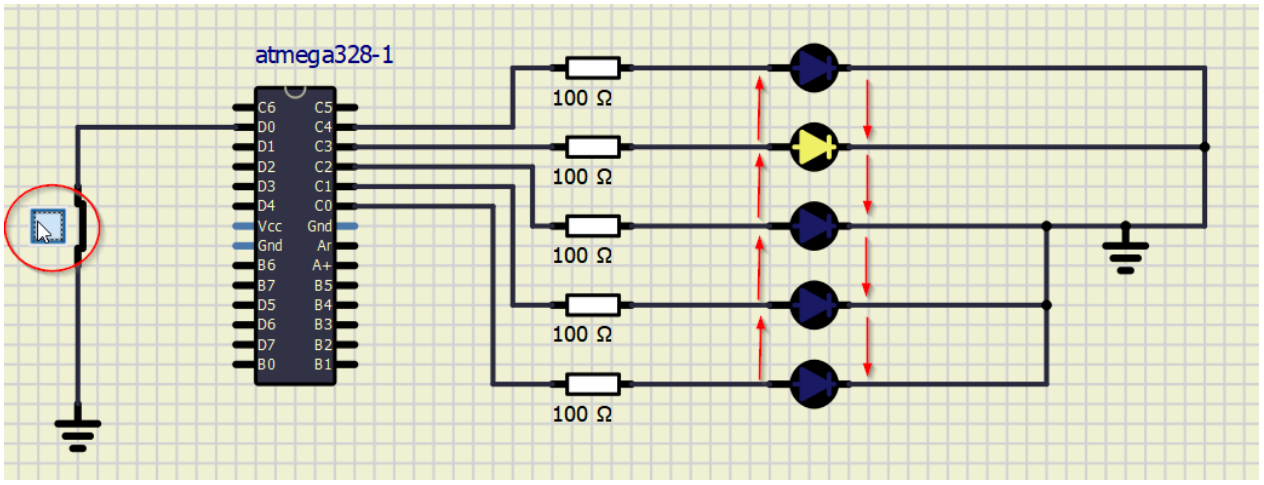
```
                        _delay_ms(SMALL_DELAY);
                        PORTC = PORTC & ~(1<<LED); // TURN OFF THE LED
                        if (LED==0) // CHECK FOR PC0
                        {
                                back =0; // IT HAS TO MOVE FORWARD
                        }
                        else if (LED == 4)
                        {
                                back = 1; // IT HAS TO MOVE BACKWARDS
                        }
                        if (back == 0)  // MOVING IN FORWARD DIRECTION
                        {
                                LED++;
                        }
                        else if (back == 1) // MOVING BACKWARDS
                        {
                                LED--;
                        }

                }

        }

        // Will never reach this
        return 0;
}
```

1.1.6 Screenshot_2 of SimulIDE circuit.


atmega328-1
C6
D0
D1
D2
D3
D4
Vcc
Gnd
B6
B7
D5
D6
D7
B0
C5
C4
C3
C2
C1
C0
Gnd
Ar
A+
B5
B4
B3
B2
B1
100 Ω
100 Ω
100 Ω
100 Ω
100 Ω