

|  |
| --- |
| **Course: Embedded Electronic Devices and Programming** |
| Laboratory work № 2 |
| “Handling Interrupt Requests” |

|  |  |
| --- | --- |
| Student name and surname: | Nikita Goloviznins |
| Student code: | St78436 |
| Group: | 4103BDA |
| Lecturer: | Artjoms Ivanovs |

|  |  |
| --- | --- |
| Date performed: | 30.03.2023 |

Riga

2023

TABLE OF CONTENTS

[Ⅰ Task 2](#_Toc132356646)

[Ⅱ Individual variant 3](#_Toc132356647)

[Ⅲ Microcontroller I/O diagram 4](#_Toc132356648)

[Ⅳ Algorithm 5](#_Toc132356649)

[Ⅴ Program code 6](#_Toc132356650)

[Ⅵ Conclusion 8](#_Toc132356651)

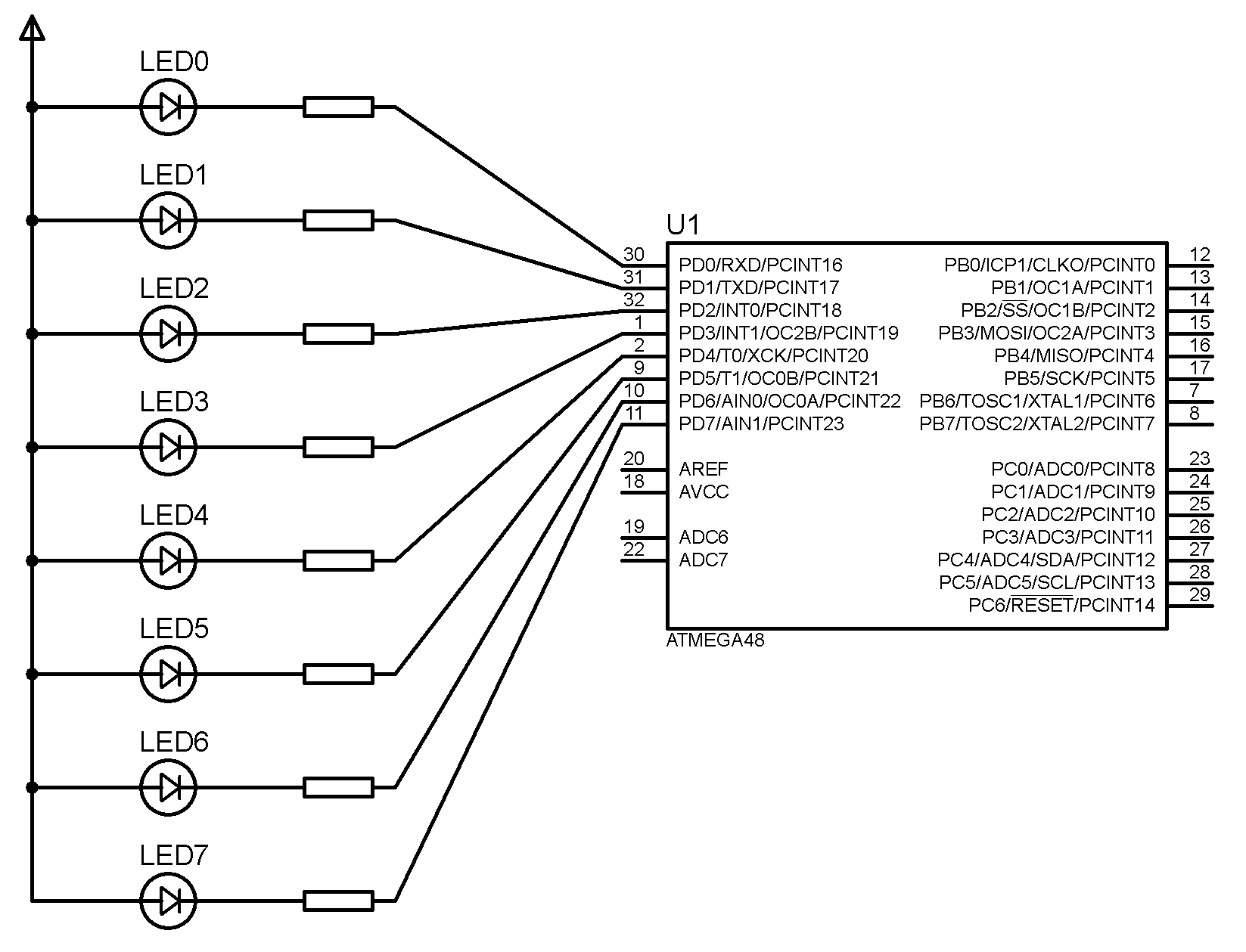
Ⅰ Task

* to develop of the microcontroller I/O diagram;
* to create an algorithm of main function and an algorithm(-s) of handling interrupt requests for the ATMega48 microcontroller in accordance with an individual task;
* to create program code in Atmel Studio 7 environment;
* to debug the program in Atmel Studio 7 environment;
* to issue a report on laboratory work;
* to form time intervals, use functions that form delays;
* when debugging a program in the Atmel Studio 7 environment, the call of functions that generate delays must be commented out;
* the clock frequency of the ATMega48 microcontroller is 8 MHz;
* when the button is pressed on the input/output line of the port - logical "0", when the button is released on the input/output line of the port - logical "1";
* the LED is turned on by logical "0" on the I/O line of the port, and turned off by logical "1".

Ⅱ Individual variant

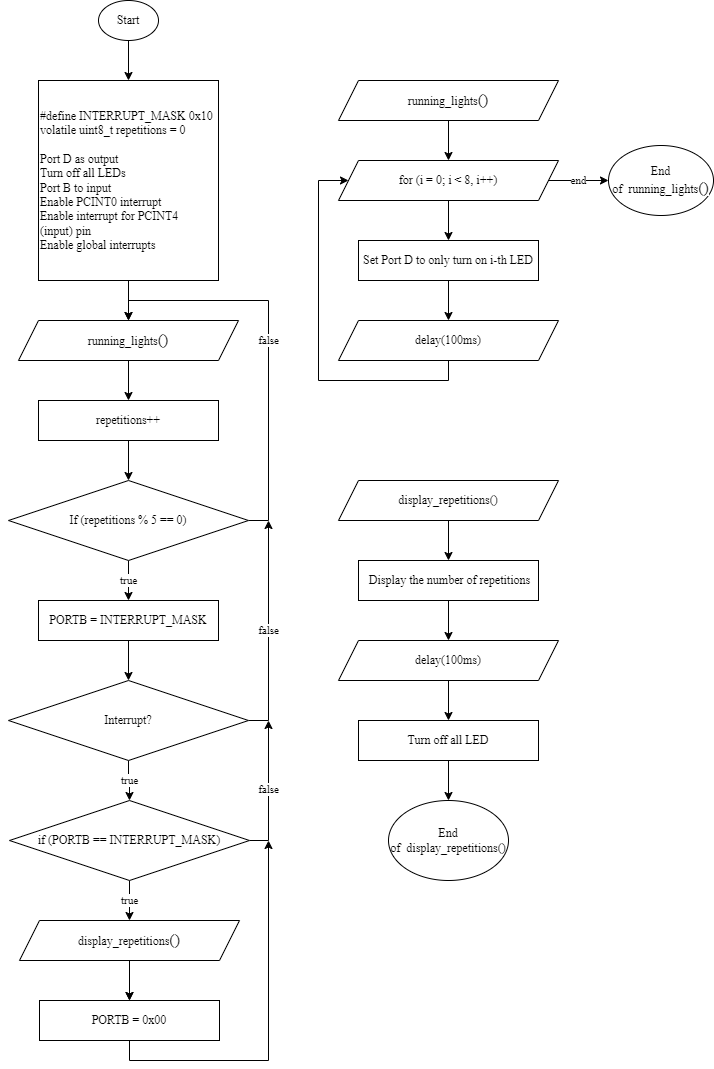
*Variant № 17.* LEDs are connected to port D. Buttons are not used. The LEDs light up in turn, creating the effect of running lights from left to right. After every 5 repetitions, a PCINT0 interrupt request is called by applying the value 0x10 to the input line of port B. The PCINT0 interrupt handler must be programmed to display the number of repetitions of running lights for 6 seconds.

Ⅲ Microcontroller I/O diagram



**Figure 1. ATMEGA48 I/O diagram**

Ⅳ Algorithm



**Figure 2. ATMEGA48 algorithm**

Ⅴ Program code

#define *F\_CPU* 8000000UL // the clock frequency of the ATMega48 microcontroller is 8 MHz

#include <avr/io.h> // Including I/O library

#include <avr/interrupt.h> // Including interrupt library

#include <util/delay.h> // Including a library with delay functions

#define LED\_PORT PORTD // Port D controls the LEDs

#define LED\_DDR DDRD // Data Direction Register for Port D

#define INTERRUPT\_PORT PORTB // Port B controls the interrupt

#define INTERRUPT\_DDR DDRB // Data Direction Register for Port B

#define INTERRUPT\_MASK 0x10 // Mask for interrupt pin

#define REPETITIONS\_BEFORE\_INTERRUPT 5 // After every 5 repetitions of running lights, an interrupt is triggered

volatile *uint8\_t* repetitions = 0; // Global variable to count the number of repetitions

void setup() {

LED\_DDR = 0xFF; // Set up the LED pins as outputs

LED\_PORT = 0xFF; //Turn off all LED

INTERRUPT\_DDR = INTERRUPT\_MASK; // set desired pins on Port B as input

INTERRUPT\_PORT = 0x00; // initialize all pins on Port B to low

PCICR |= (1 << PCIE0); // enable PCINT0 interrupt

PCMSK0 |= (1 << PCINT4); // enable interrupt for PCINT4 (input) pin

sei(); // Enable global interrupts

}

void running\_lights() {

for (*uint8\_t* i = 0; i < 8; i++) {

LED\_PORT = ~(1 << i); // Turn on LED at i-th position

*\_delay\_ms*(100); // Delay 100 ms

}

}

void display\_repetitions() {

LED\_PORT = ~repetitions; // Display the number of repetitions

*\_delay\_ms*(6000); // Delay 6 second

LED\_PORT = 0xFF; //Turn off all LED

}

ISR(PCINT0\_vect) { // PCINT0 interrupt

if (INTERRUPT\_PORT == INTERRUPT\_MASK) { // check if PORTB is 0x10

display\_repetitions();

INTERRUPT\_PORT = 0x00; // Clear Port B

}

}

int main() {

setup();

while (1) {

running\_lights();

repetitions++;

if (repetitions % REPETITIONS\_BEFORE\_INTERRUPT == 0) {

INTERRUPT\_PORT = INTERRUPT\_MASK; // Trigger interrupt by setting PinB to mask

}

}

return 0;

}

Ⅵ Conclusion

In this laboratory work, we have developed a program for the ATMega48 microcontroller to control LEDs connected to Port D. The program uses a running lights effect, where the LEDs light up in turn from left to right. After every five repetitions of the running lights, a PCINT0 interrupt request is called by applying the value 0x10 to the input line of Port B. The PCINT0 interrupt handler is programmed to display the number of repetitions of the running lights for six seconds.

The program has been successfully implemented in Atmel Studio 7 environment and debugged.

The program uses a global variable to count the number of repetitions and a function to display the number of repetitions on the LEDs. The program also uses an interrupt to trigger the display function after every five repetitions.

In conclusion, this laboratory work has provided us with practical experience in developing a program for a microcontroller and utilizing interrupts to handle events in the program. The resulting program effectively controls the LEDs and displays the number of repetitions, demonstrating the successful implementation of the given task.