

|  |
| --- |
| **Course: Embedded Electronic Devices and Programming** |
| Laboratory work № 3 |
| “TIMERS” |

|  |  |
| --- | --- |
| Student name and surname: | Daniil Grammatikopulo |
| Group: | 4103BDA |
| Lecturer: | Artjoms Ivanov |

|  |  |
| --- | --- |
| Date performed: | 30.04.2023 |

Riga

2023

Table of Content

[Table of Content 2](#_Toc133765690)

[TASK FOR LABORATORY WORK 3](#_Toc133765691)

[The report should have followed structure: 3](#_Toc133765692)

[Individual variant of the task 4](#_Toc133765693)

[Microcontroller I/O diagram 5](#_Toc133765694)

[ALGORITHM 6](#_Toc133765695)

[PROGRAM CODE 7](#_Toc133765696)

[CONCLUSION 8](#_Toc133765697)

TASK FOR LABORATORY WORK

The report should have followed structure:

* 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".

Individual variant of the task

(16) Display sequentially after 1 second the numbers from 0 to 12 in binary code using LEDs. The display is controlled by timer T1.

Microcontroller I/O diagram

Изображение выглядит как диаграмма, схематичный

Автоматически созданное описание

**ATmega48 diagram 1**

ALGORITHM

Изображение выглядит как диаграмма

Автоматически созданное описание

**Algorithm diagram 1**

PROGRAM CODE

#include <avr/io.h>

#include <avr/interrupt.h>

int main(void)

{

// Enable DDRB to output mode and turn off LEDs

DDRB = 0xFF;

PORTB = 0xFF;

// Set up Timer 1 with a prescaler of 1024

TCCR1B |= (1 << CS12) | (1 << CS10);

// Enable Timer 1 overflow interrupt

TIMSK1|= (1 << TOIE1);

// Set initial value of Timer 1 counter to 57723

// TCNT1 = 65535 - F\_CPU / 1024

TCNT1 = 57723;

// Enable global interrupt

sei();

while (1) {}

}

// Interrupt service routine for Timer 1 overflow

ISR(TIMER1\_OVF\_vect) {

// If PORTB shows 12 in binary code, turn off LEDs

if (PORTB == 0xF3)

PORTB = 0xFF;

// Increase number by 1 for each occurrence of timer overflow

else

PORTB = ~(~PORTB + 1);

// Set initial value of Timer 1 counter to 57723

TCNT1 = 57723;

}

CONCLUSION

In conclusion, this lab was not difficult, generally because of the given task. Setting timer is quite difficult because different timers have different modes and setting. Anyway, I figured out how to work with them and the lab was completed successfully.

P.S work is also available on my GitHub: <https://github.com/Dast3X/ATmega48/tree/main/Timers>