**PROTOKOLL**for lab-exercise

***AVR ext. Interrupt, Timer and PWM Part 2***



|  |  |  |
| --- | --- | --- |
| Group / Class | Script writer | Signature |
| 6 / **4BHELS** | BIEHL S |  |
| Date of Exercise / Hand-in Date | Team member | Signature |
| 2. Dec. 2014  17. Dec. 2014 | HIRSCH L. |  |
| Teacher | Team member | Signature |
| CRHA | HOFSTÄTTER A. |  |
| Grade | Team member | Signature |
|  |  |  |
| ***AVR ext. Interrupt, Timer and PWM Part 2***  *ATmega32U4* | | |
| **Used devices**   |  |  |  |  |  | | --- | --- | --- | --- | --- | | Nr. | Gerätebezeichnung | Hersteller | Typ | Platznummer | | 1. | Oszilloskop | Tektronix | TDS 1001B | - |   **Used programs**   |  |  |  | | --- | --- | --- | | Nr. | Name | Version | | 1. | CodeBlocks | 13.12 | | 2. | DFU-Programmer | 1.2.2 | | | |

ÜBUNGS-/ABGABE-DATUM

Klasse /Gruppe

NOTE

LEHRER

# Index

[1 Index 2](#_Toc406480294)

[2 Task 3](#_Toc406480295)

[2.1 Timer 1 Phase and Frequency correct PWM 3](#_Toc406480296)

[2.2 LED pulsation 3](#_Toc406480297)

[2.3 Output Compare Modulator 3](#_Toc406480298)

[3 Timer 1 Phase and Frequency Correct PWM 4](#_Toc406480299)

[3.1 Oscillogram 4](#_Toc406480300)

[3.2 Programmlisting 4](#_Toc406480301)

[4 LED Pulsation 5](#_Toc406480302)

[4.1 Oscillosgram 5](#_Toc406480303)

[4.2 Programmlisting 5](#_Toc406480304)

[5 Output Compare Modulator 6](#_Toc406480305)

[5.1 Oscillogram 6](#_Toc406480306)

[5.2 Programmlisting 7](#_Toc406480307)

[6 Comment 8](#_Toc406480308)

# Task

The following 3 Task had to be done.

## Timer 1 Phase and Frequency correct PWM

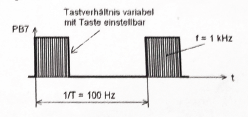
Generate a PWM signal with Timer 1 which is 90% Low and 10% of the time (fclk = 16 MHz)

## LED pulsation

The µC Board LED has to pulse from 0-100% in 1 second. A random PWM signal can be used.

## Output Compare Modulator

Generate following signal fclk = 8 MHz :



# Timer 1 Phase and Frequency Correct PWM

In the task it was asked to generate a simple signal at 1kHz and a prescaler 16MHz which can be seen in Figure 1. with 10% High 90% Low slope.

## Oscillogram

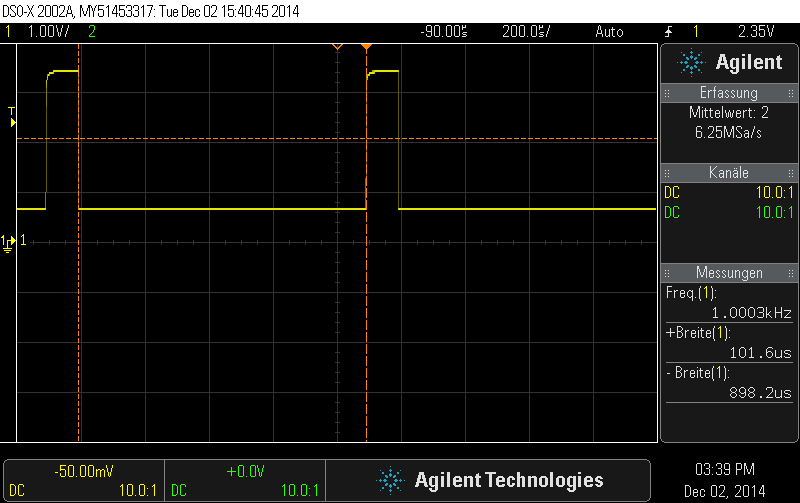
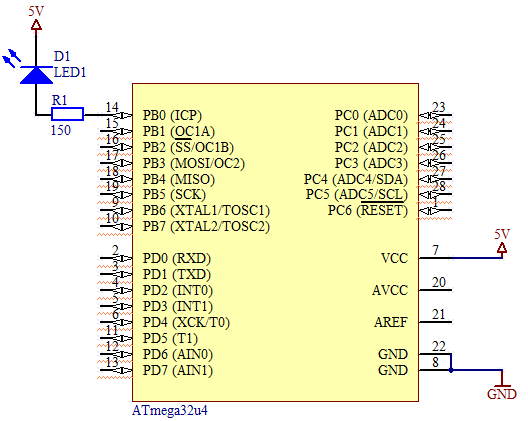


Figure 1 – Phase and Frequency Correct PWM

The figure shows a PWM signal with ton=101.6µs and a toff = 898.2µs at 1kHz

## Programmlisting

#include <avr/io.h>

#include <avr/interrupt.h>

ISR**(**TIMER1\_COMPA\_vect**)**

**{**

PORTB **^=** **(**1**<<**PB0**);**

**}**

int main**(**void**)**

**{**

CLKPR**=**0x80**;**

CLKPR**=**0x00**;**

DDRB **|=** **(**1**<<**DDB0**);**

TCCR1B **|=** **(**1**<<**WGM13**);**

TCCR1B **|=** **(**1**<<**CS11**);**

ICR1H **=** 0x03**;**

Figure 2 – Circuit

ICR1L **=** 0xE8**;**

OCR1AL **=** 130**;**

TIMSK1 **|=** **(**1**<<**OCIE1A**);**

sei**();**

**while(**1**);**

**}**

**Listing 1** – C Code for Phase and Frequency correct PWM (Task 4)

# LED Pulsation

The task was to set the LED on the microcontroller board to change their brightness with a random PWM in 1 second from 0% to 100% and then start again from 0. So that a pulsating light signal is produced.

## Oscillosgram

The following figures show the process of the LED. The pictures were taken randomly just to show the course.

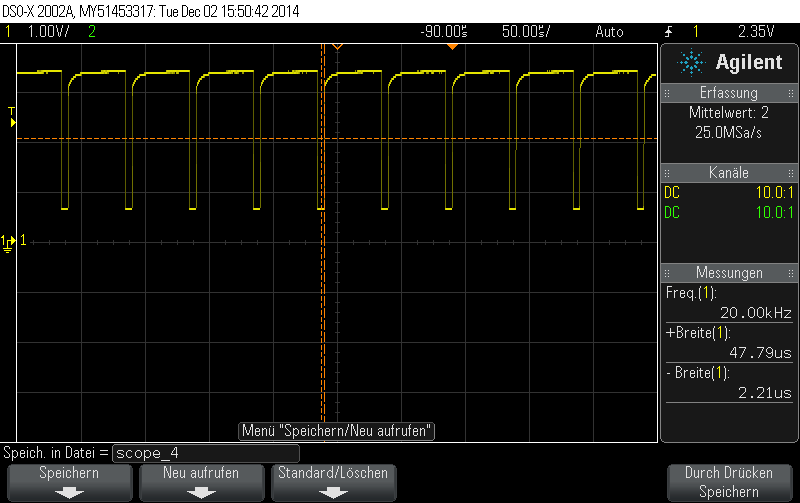
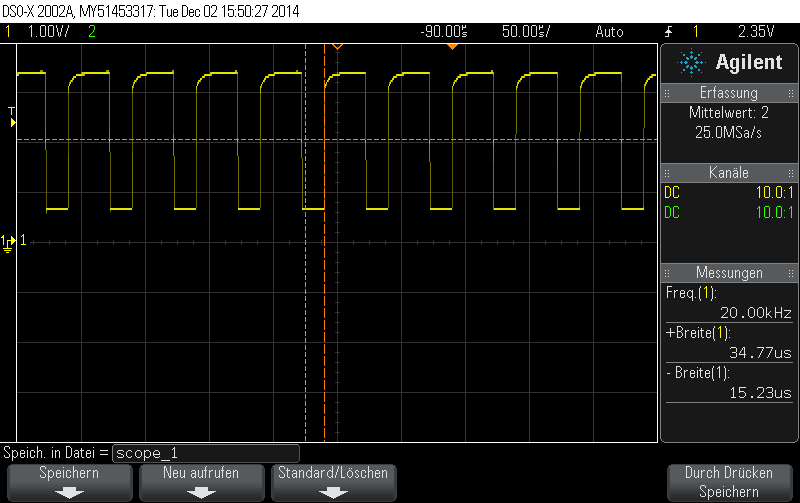
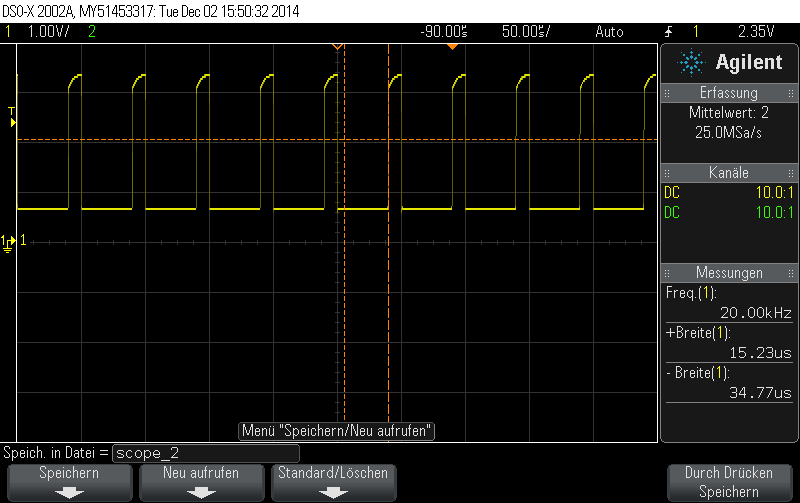
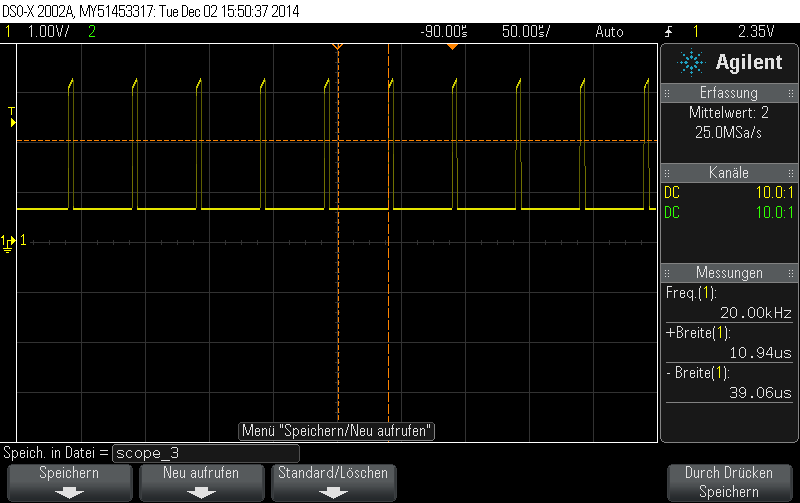
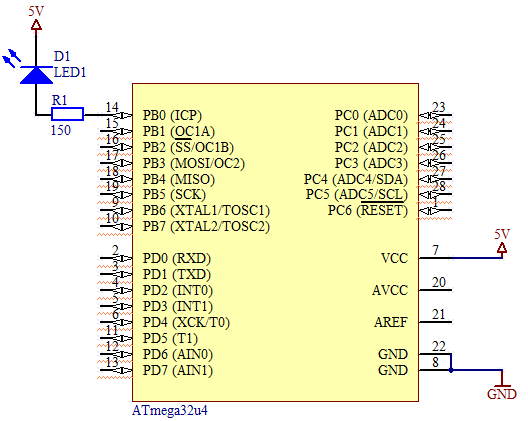


Figure 3 – Process of the Pulsation

## Programmlisting



#include <avr/io.h>

#include <avr/interrupt.h>

#include <util/delay.h>

ISR**(**TIMER1\_COMPA\_vect**)**

**{**

PORTB **|=** **(**1**<<**PB0**);**

**}**

ISR**(**TIMER1\_OVF\_vect**)**

**{**

PORTB **&=** **~(**1**<<**PB0**);**

**}**

int main**(**void**)**

**{**

CLKPR**=**0x80**;**

CLKPR**=**0x00**;**

Figure 4 – Circuit

DDRB **|=** **(**1**<<**DDB0**);**

TCCR1A **|=** **(**1**<<**WGM11**);**

TCCR1B **|=** **(**1**<<**WGM12**)** **|** **(**1**<<**WGM13**);**

TCCR1B **|=** **(**1**<<**CS11**);**

ICR1L **=** 0x63**;**

OCR1A **=** 99**;**

TIMSK1 **|=** **(**1**<<**OCIE1A**)** **|** **(**1**<<**TOIE1**);**

sei**();**

**while(**1**)**

**{**

\_delay\_ms**(**10**);**

OCR1A **-=** 1**;**

**if(**OCR1A**<=**1**)**

**{**

OCR1A**=**99**;**

**}**

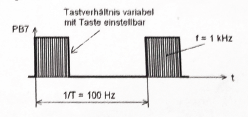
**}**

**}**

**Listing 2** – C Code for the LED pulsation (Task 5)

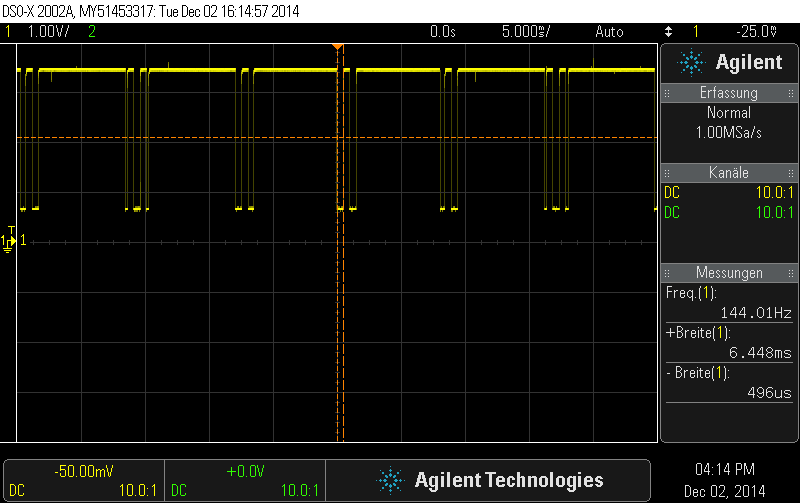
# Output Compare Modulator

Generate the following Signal:



A circuit is not necessary in this task because the signal was measured at PB7.

## Oscillogram



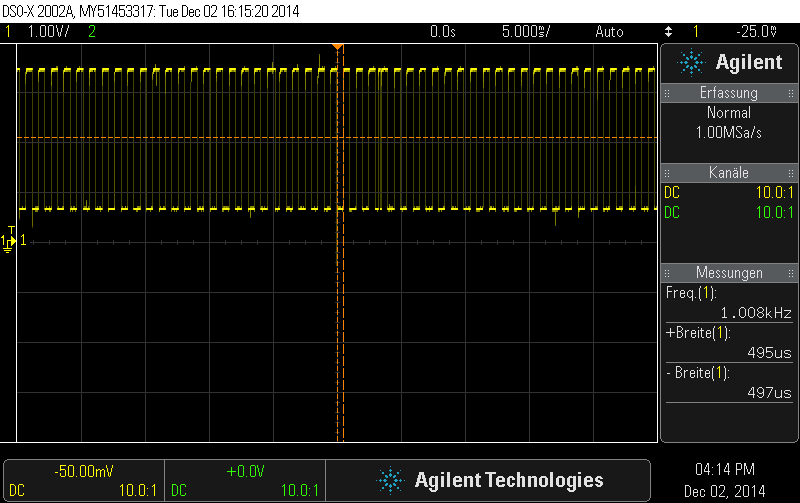


Figure 6,7 – Compare Modulator 0%-100%

## Programmlisting

int main **(**void**)**

**{**

// Įdern des internen CLK-Prescaler (int. RC-Oszillator)

CLKPR **=** 0x80**;** //䮤ern des internen CLK-Prescalers

CLKPR **=** 0x01**;** //16 MHz :2 = 8 MHz

// Portpins konfigurieren

DDRB **=** DDRB **|** **(**1**<<**DDB7**);** //PB7 als Ausgang

DDRD **=** DDRD **&~(**1**<<**DDD0**);** //PD0 als Input

PORTD **=** PORTD **|** **(**1**<<**PORTD0**);** //PD0 int. Pull Up ein

// Timer 1 Fast PWM am OC1C (PB7) mit 122Hz

TCCR1A **=** TCCR1A **|** **(**1**<<**WGM10**);**

TCCR1A **=** TCCR1A **|** **(**1**<<**WGM11**);**

TCCR1B **=** TCCR1B **|** **(**1**<<**WGM12**);** //WGM2:0=3

TCCR1B **=** TCCR1B **&~(**1**<<**WGM13**);** //WGM3:0=7 10-Bit Fast PWM

TCCR1A **=** TCCR1A **&~(**1**<<**COM1C0**);**

TCCR1A **=** TCCR1A **|** **(**1**<<**COM1C1**);** //COM1:2=2 Non Inverting

TCCR1B **=** TCCR1B**|(**1**<<**CS11**)|(**1**<<**CS10**);** //:64 Teiler, startet PWM

// Timer 0 CTC am OC0A (PB7) mit 1kHz

TCCR0A **=** TCCR0A**|(**1**<<**COM0A0**)|(**1**<<**WGM01**);** //COM1:0=1 , WGM2:0=2

OCR0A **=** 61**;** //Wert fr 1kHz

TCCR0B **=** TCCR0B**|(**1**<<**CS01**)|(**1**<<**CS00**);** //:64 Teiler => dt=64/8MHz=8us

**while(**1**)**

**{** **while(**PIND **&** **(**1**<<**PIND0**));** //warten auf neg. Flanke von Taste PB0

OCR1C **=** OCR1C **+** 100**;** //Tastverh䬴nis 䮤ern

**while(!(**PIND **&** **(**1**<<**PIND0**)));** //warten auf pos. Flanke von Taste PB0

**}**

**}**

**Listing 3** – C Code for the Compare Modulator (Task 7)

# Comment

The lab-exercise went pretty well. The only troubles happened at task 6. It has not been finished because the code did not work properly. Also the constitution has not been finished because the time ran out. All in the entire task was easy to understand and no major problem happened at the finished exercises.