Project Group 'Racing Car IT'

AVR Timer Configuration

Report of manual tests

Peer Adelt

University of Paderborn

Summer Semester 2012

Advisers:

Prof. Dr. Marco Platzner, Tobias Beisel, Sebastian Meisner, Lars Schäfers

Contents

1 Introduction

This document descibes the manual tests that have been performed to ensure correct operation of the AVR Timer Configuration Editor and the Code Generator.

It is assumed, that the reader is familiar with the hardware details of the timer modules of the AT90CAN microprocessor family. This document will not explain, what they can do or how they are configured. Please refer to the processor's datasheet for further details.

All tests were performed on the DVK90CAN1 board, which is equipped with the Atmel AVR processor AT90CAN128. The processor was clocked by an 8MHz crystal during all tests. The DVK90CAN1 was stacked on top of the STK500 development board, which acts as a port extender in the test scenarios described below. Since the DVK90CAN1 does not provide access to all processor pins, especially not all pins driven by the waveform generation modes of the timer modules, the use of the STK500 was mandatory.

The STK500 exposed many, but not all pins of the AT90CAN128 microprocessor. The Port E of the processor, which is the one driven by Timer 3, was not accessible at all. Tests for Timer 3 were performed the same way as for Timer 1. The results have been verified with the oscilloscope with its probe held down to the processor pin directly. In this situation though, it was impossible to take photographs of the measurement device. Nevertheless, all tests of Timer 3 passed equally well as those of Timer 1.

2 Oscilloscope tests

All tests were performed with an analoguous 60 MHz oscilloscope (Model: Oscilloscope 668, Brand: Voltcraft).

2.1 Normal / Overflow mode

In normal mode, the timer counts from 0 to MAX and then restarts. This happens in an endless loop. Therefore, the overflow frequency depends on the MAX value as well as the CPU frequency multiplied with the prescale divisor. Since no waveform generation takes place in Overflow mode, it needs to be simulated with software. All tests of this mode take place with interrupts enabled. The overflow interrupt routine will toggle the state of an output pin, which is measured with an oscilloscope to verify the toggle frequency. (See example code below.)

```
1 #include <avr/interrupt.h>
2
3 ...
4
5 ISR(TIMER0_OVF_vect)
6 {
7     // Toggle PB2:
8     PORTB ^= (1<<2);
9 }
```

2.1.1 8bit timer 0, prescale 1

Scenario:	test_overflow_t0
Mode:	Normal / Overflow
Timer:	0 (8bit)
Prescale:	1
Overflow period:	32 µs (Time to overflow: $\frac{Prescale*MAX}{Frequency}$)
Test description:	Output pin is toggled from overflow ISR. The toggle frequency is validated with an oscilloscope.

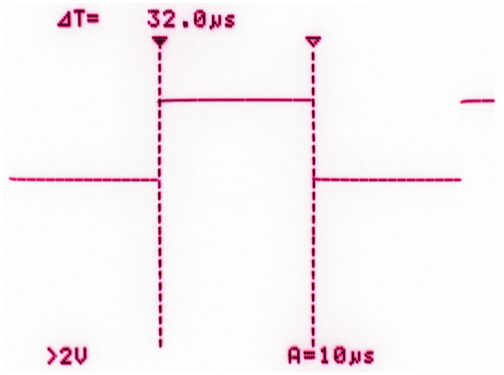


Figure 1: 32μs overflow rate verified

2.1.2 8bit timer 2, prescale 64

Scenario:	test_overflow_t2
Mode:	Normal / Overflow
Timer:	2 (8bit)
Prescale:	64
Overflow period:	2048 µs (Time to overflow: $\frac{Prescale*MAX}{Frequency}$)
Test description:	Output pin is toggled from overflow ISR. The toggle frequency is validated with an oscilloscope.

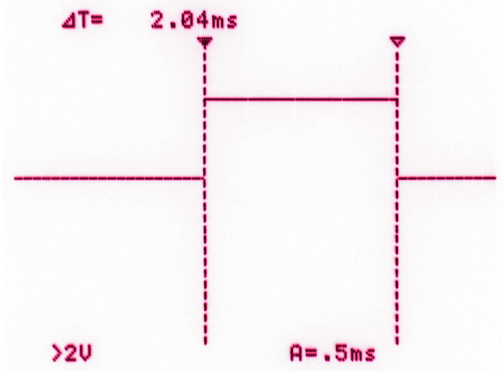


Figure 2: 2048μs overflow rate verified

2.1.3 16bit timer 1, prescale 8

Scenario:	test_overflow_t1
Mode:	Normal / Overflow
Timer:	1 (16bit)
Prescale:	8
Overflow period:	65536 µs (Time to overflow: $\frac{Prescale*MAX}{Frequency}$)
Test description:	Output pin is toggled from overflow ISR. The toggle frequency is validated with an oscilloscope.

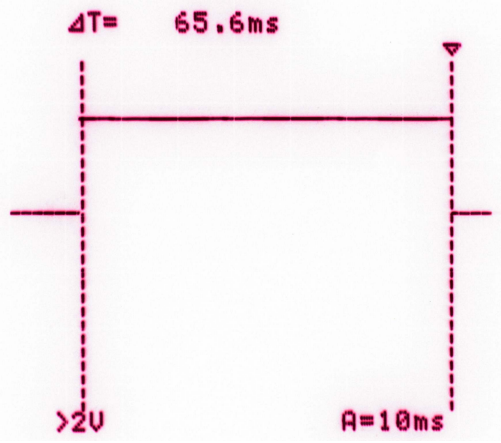


Figure 3: 65536μs overflow rate verified

2.1.4 16bit timer 3, prescale 1

Scenario:	test_overflow_t3
Mode:	Normal / Overflow
Timer:	3 (16bit)
Prescale:	1
Overflow period:	8192 µs (Time to overflow: $\frac{Prescale*MAX}{Frequency}$)
Test description:	Output pin is toggled from overflow ISR. The toggle frequency is validated with an oscilloscope.

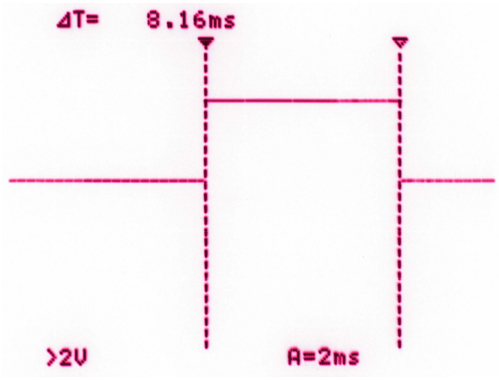


Figure 4: 8192μs overflow rate verified

2.2 CTC mode

In CTC mode, the timer runs from 0 to a user defined TOP value (instead of MAX, like in Overflow mode). This happens in an endless loop. Therefore, the compare match frequency depends on the TOP value and CPU frequency multiplied with the prescale divisor. Like in the overflow tests, an output pin will be toggled. The toggle frequency will be validated with an oscilloscope.

2.2.1 8bit timer 0, prescale 1, desired period: 12.4 µs

Scenario:	test_ctc_t0
Mode:	CTC
Timer:	0 (8bit)
Prescale:	1
Top register:	OCR0A
Top period:	12.375 µs (99 ticks) (desired value: 12.4 µs, number of timer ticks will get quantized from 99.2 to 99)
Test description:	Output pin OC0A (PB7) is toggled by timer hardware. The toggle frequency is validated with an oscilloscope.

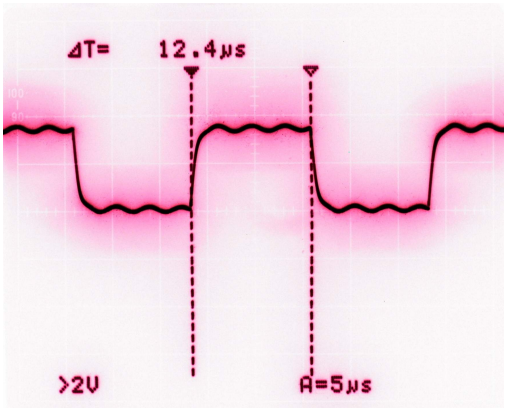


Figure 5: 12.375μs toggle rate verified

2.2.2 8bit timer 2, prescale 64, desired period: 1ms

Scenario:	test_ctc_t2
Mode:	CTC
Timer:	2 (8bit)
Prescale:	64
Top register:	OCR2A
Top period:	1ms (125 ticks)
Test description:	Output pin OC2A (PB4) is toggled by timer hardware. The toggle frequency is validated with an oscilloscope.

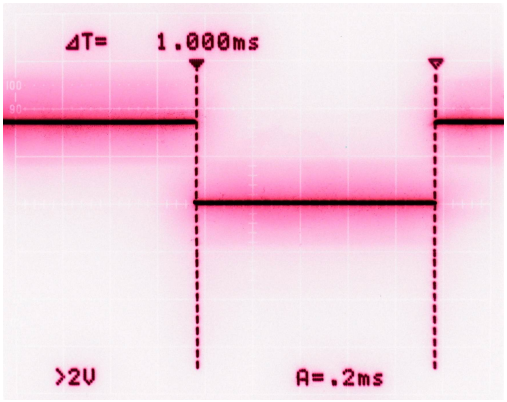


Figure 6: 1ms toggle rate verified

2.2.3 16bit timer 1, prescale 256, desired period: 1s

Scenario:	test_ctc_t1
Mode:	CTC
Timer:	1 (16bit)
Prescale:	256
Top register:	OCR1A
Top period:	1s (31250 ticks)
Test description:	Output pin OC1B (PB6) is toggled by timer hardware. The toggle frequency is validated with an oscilloscope.

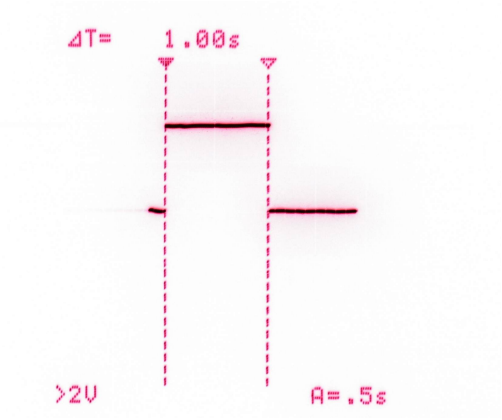


Figure 7: 1s toggle rate verified

2.2.4 16bit timer 3

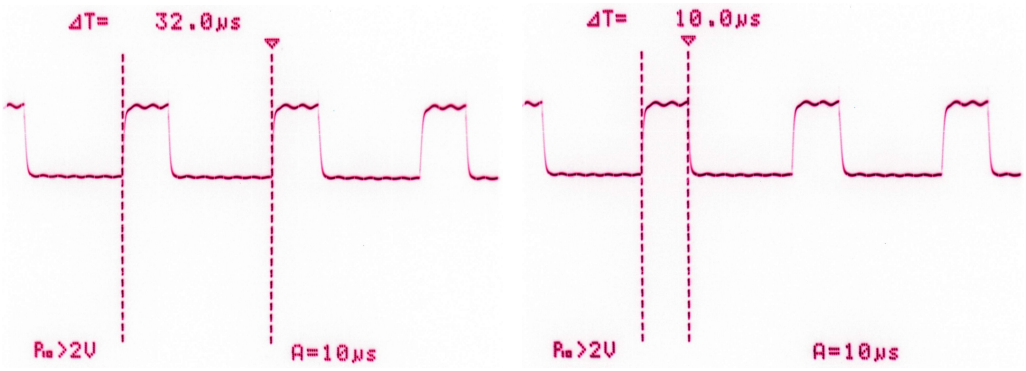
The same test procedure as for timer 1 was run. Although now photographs have been taken, the test has passed (see section 1 for details).

2.3 Fast PWM mode

In Fast PWM mode, the timer runs from 0 to TOP (or MAX, when using 8bit Timers). This happens in an endless loop. Therefore, the PWM base rate depends on the TOP value and CPU frequency multiplied with the prescale divisor. The output pins can be set up to be either set on TOP and cleared on compare match (normal PWM) or cleared on TOP and set on compare match (inverted PWM). The generated PWM waveforms are validated with an oscilloscope.

2.3.1 8bit timer 0, prescale 1, base rate 32 µs

Scenario:	test_fpwm_t0
Mode:	Fast PWM
Timer:	0 (8bit)
Prescale:	1
Top register: **Top period:**	N/A (fixed value 255) 32 µs (256 ticks)
Duty cycle A: **Duty cycle B:** **Duty cycle C:**	10 µs (80 ticks) N/A N/A
Test description:	Output pin OC0A (PB7) is controlled by timer hardware to output a non-inverted 10 µs/32 µs PWM signal. This PWM signal is validated with an oscilloscope.

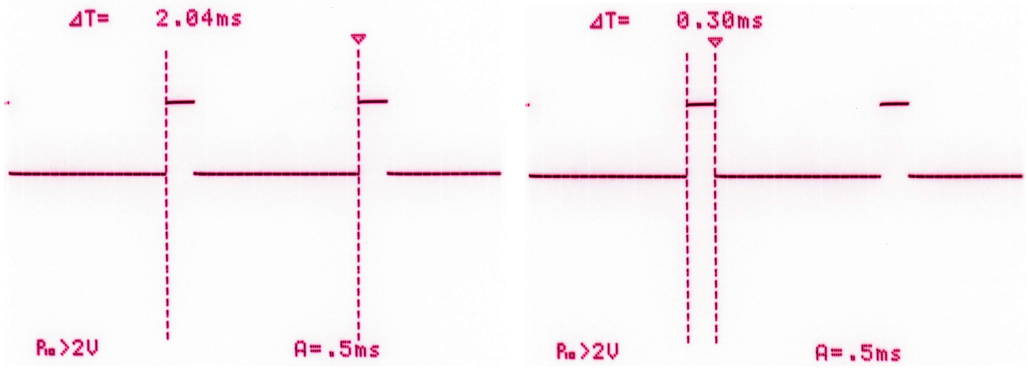


(a) base rate (b) duty cycle

Figure 8: 32 µs base rate and 10 µs duty cycle verified

2.3.2 8bit timer 2, prescale 64, base rate 2048 µs

Scenario:	test_fpwm_t2
Mode:	Fast PWM
Timer:	2 (8bit)
Prescale:	64
Top register: **Top period:**	N/A (fixed value 255) 2048 µs (256 ticks)
Duty cycle A: **Duty cycle B:** **Duty cycle C:**	304 µs (38 ticks) (desired value: 300 µs, number of timer ticks will get quantized from 37.5 to 38) N/A N/A
Test description:	Output pin OC2A (PB4) is controlled by timer hardware to output a non-inverted 304 µs/2048 µs PWM signal. This PWM signal is validated with an oscilloscope.

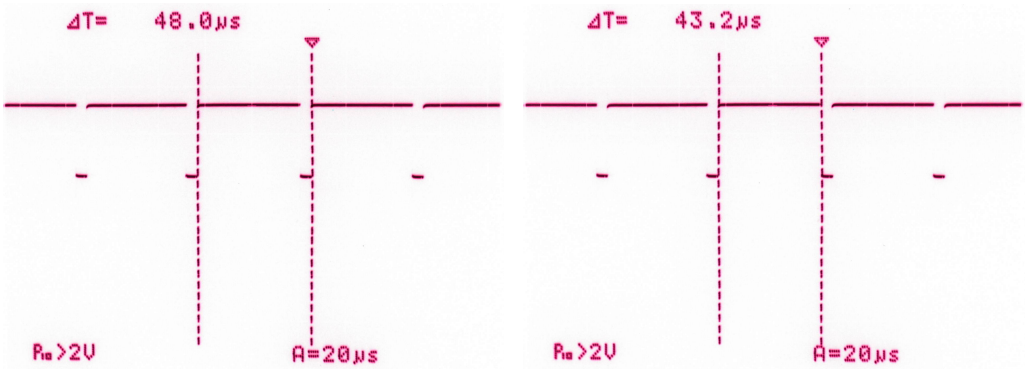


(a) base rate (b) duty cycle

Figure 9: 2048 µs base rate and 304 µs duty cycle verified

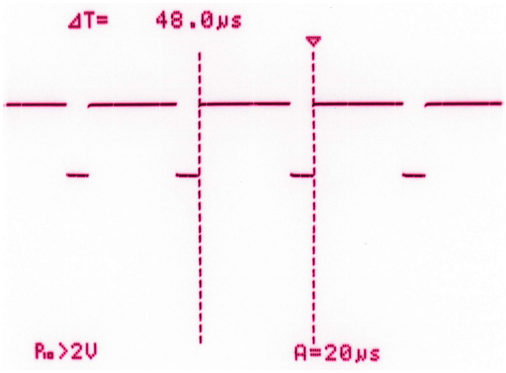
2.3.3 16bit timer 1, prescale 1, base rate 48 µs

Scenario:	test_fpwm_t1
Mode:	Fast PWM
Timer:	1 (16bit)
Prescale:	1
Top register: **Top period:**	ICR1 48 µs (384 ticks)
Duty cycle A:	43.25 µs (346 ticks) (desired value: 43.2 µs, number of timer ticks will get quantized from 345.6 to 346)
Duty cycle B:	38.375 µs (307 ticks) (desired value: 38.4 µs, number of timer ticks will get quantized from 307.2 to 307)
Duty cycle C:	33.625 µs (269 ticks) (desired value: 33.6 µs, number of timer ticks will get quantized from 268.8 to 269)
Test description:	Output pins OC1A (PB5), OC1B (PB6) and OC1C (PB7) are controlled by timer hardware to output the non-inverted PWM signals described above. These PWM signals are validated with an oscilloscope.

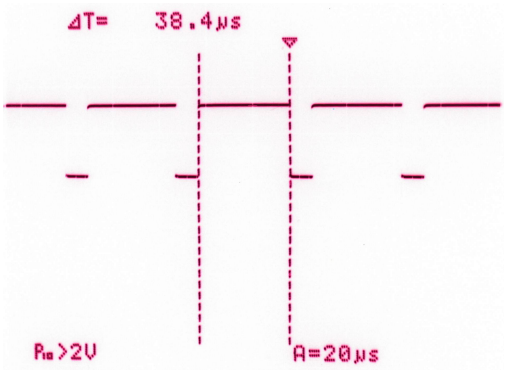


(a) base rate (channel A) (b) duty cycle (channel A)

Figure 10: 48 µs base rate and 43.25 µs duty cycle verified

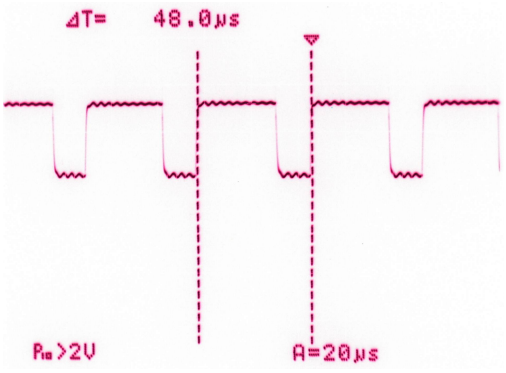


(a) base rate (channel B)

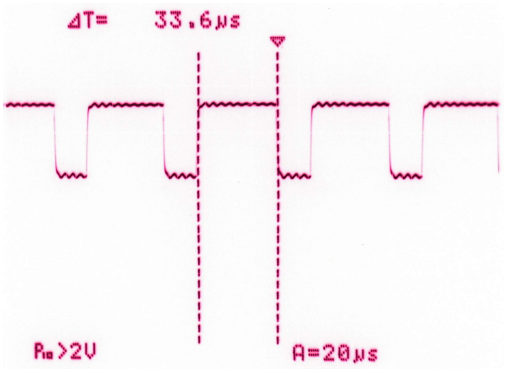


(b) duty cycle (channel B)

Figure 11: 48 µs base rate and 38.375 µs duty cycle verified



(a) base rate (channel C)



(b) duty cycle (channel C)

Figure 12: 48 µs base rate and 33.625 µs duty cycle verified

2.3.4 16bit timer 3

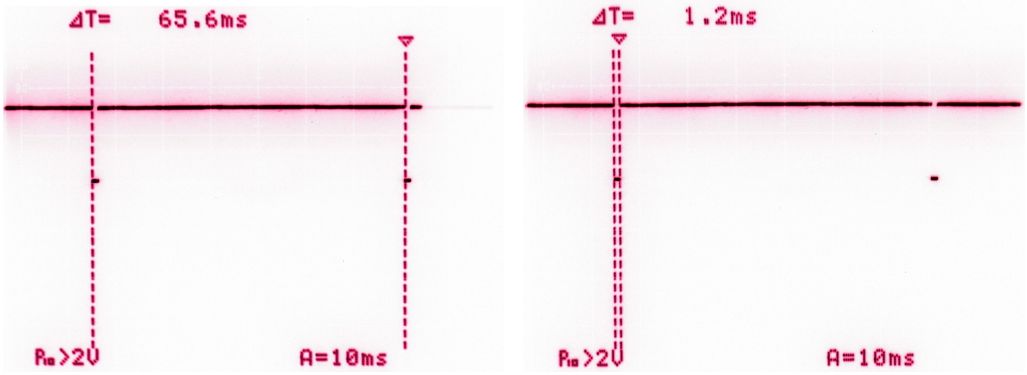
The same test procedure as for timer 1 was run. Although now photographs have been taken, the test has passed (see section 1 for details).

2.4 Phase correct PWM mode

In Phase correct PWM mode, the timer runs from 0 to TOP (or MAX, when using 8bit Timers) and then counts down to 0 again. This happens in an endless loop. Therefore, the PWM base rate depends on the TOP value and CPU frequency multiplied with the prescale divisor. The output pins can be set up for normal and inverted PWM. The generated PWM waveforms are validated with an oscilloscope.

2.4.1 8bit timer 0, prescale 1024, base rate 65280 µs

Scenario:	test_pcpwm_t0
Mode:	Phase correct PWM
Timer:	0 (8bit)
Prescale:	1024
Top register: **Top period:**	N/A (fixed value 255) 65280 µs (256 ticks)
Duty cycle A: **Duty cycle B:** **Duty cycle C:**	1280 µs (5 ticks) N/A N/A
Test description:	Output pin OC0A (PB7) is controlled by timer hardware to output output inverted PWM. The generated PWM waveform is validated with an oscilloscope.

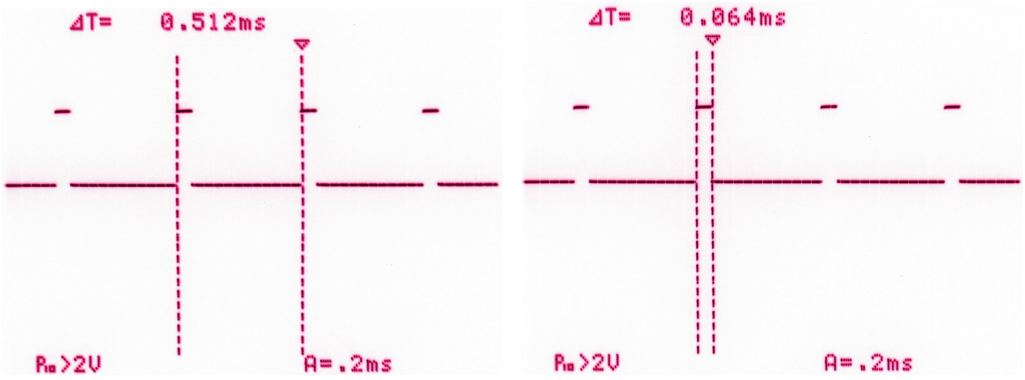


(a) base rate (b) duty cycle

Figure 13: 65280 µs base rate and 1280 µs duty cycle verified

2.4.2 8bit timer 2, prescale 8, base rate 510 µs

Scenario:	test_pcpwm_t2
Mode:	Phase correct PWM
Timer:	2 (8bit)
Prescale:	8
Top register: **Top period:**	N/A (fixed value 255) 510 µs (256 ticks)
Duty cycle A: **Duty cycle B:** **Duty cycle C:**	64 µs (32 ticks) N/A N/A
Test description:	Output pin OC2A (PB4) is controlled by timer hardware to output output non-inverted PWM. The generated PWM waveform is validated with an oscilloscope.

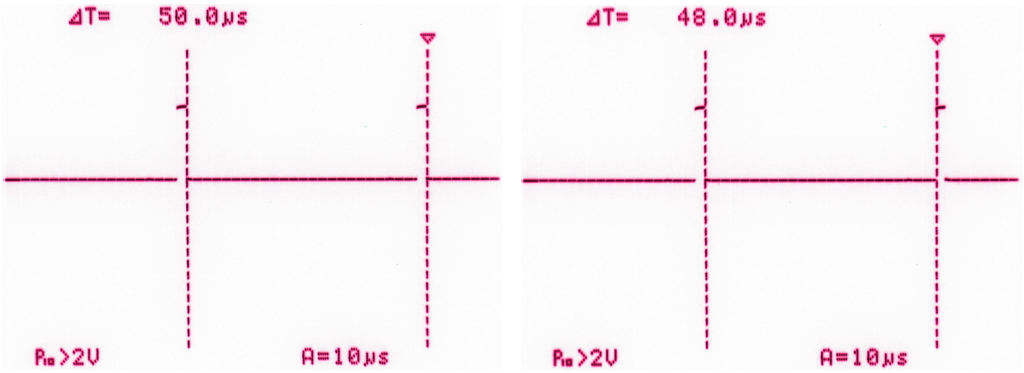


(a) base rate (b) duty cycle

Figure 14: 510 µs base rate and 64 µs duty cycle verified

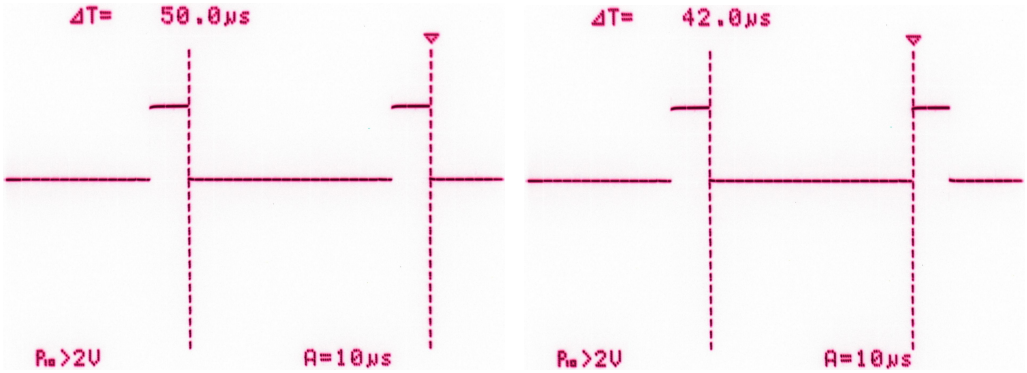
2.4.3 16bit timer 1, prescale 1, base rate 50 µs

Scenario:	test_pcpwm_t1
Mode:	Phase correct PWM
Timer:	1 (16bit)
Prescale:	1
Top register: **Top period:**	ICR1 50 µs (200 ticks)
Duty cycle A: **Duty cycle B:** **Duty cycle C:**	48 µs (192 ticks) 42 µs (168 ticks) 36 µs (144 ticks)
Test description:	Output pins OC1A (PE5), OC1B (PE6) and OC1C (PE7) are controlled by timer hardware to output the PWM signals described above. OC1A and OC1B output inverted PWM, while OC1C outputs non-inverted PWM. These PWM signals are validated with an oscilloscope.



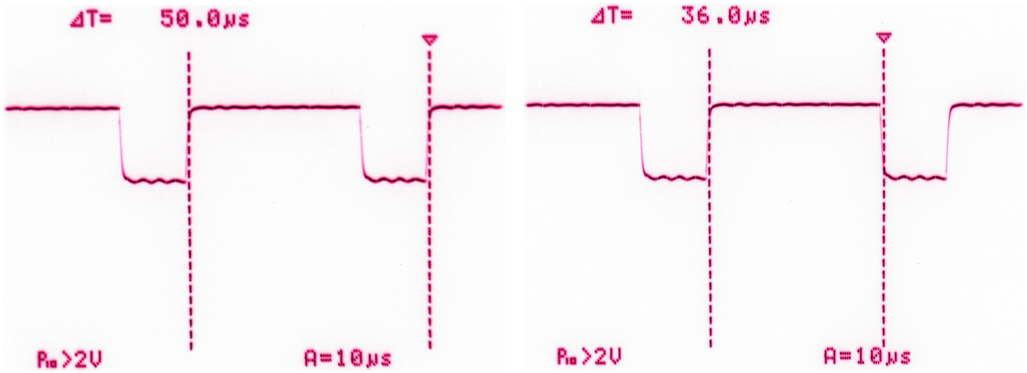
(a) base rate (channel A) (b) duty cycle (channel A)

Figure 15: 50 µs base rate and 48 µs duty cycle verified



(a) base rate (channel B) (b) duty cycle (channel B)

Figure 16: 50 µs base rate and 42 µs duty cycle verified



(a) base rate (channel C) (b) duty cycle (channel C)

Figure 17: 50 µs base rate and 36 µs duty cycle verified

2.4.4 16bit timer 3

The same test procedure as for timer 1 was run. Although now photographs have been taken, the test has passed (see section 1 for details).

3 Conclusions

With the manual oscilloscope tests it has been shown, that all timer modules work as expected in any supported mode of operation. The test values were chosen randomly to ensure a wide area of test cases being covered.

The measurement accuracy of the oscilloscope used for the tests was sufficiently high to prove, that the value calculation for processor registers as well as the quantization of time periods into register values works as expected.