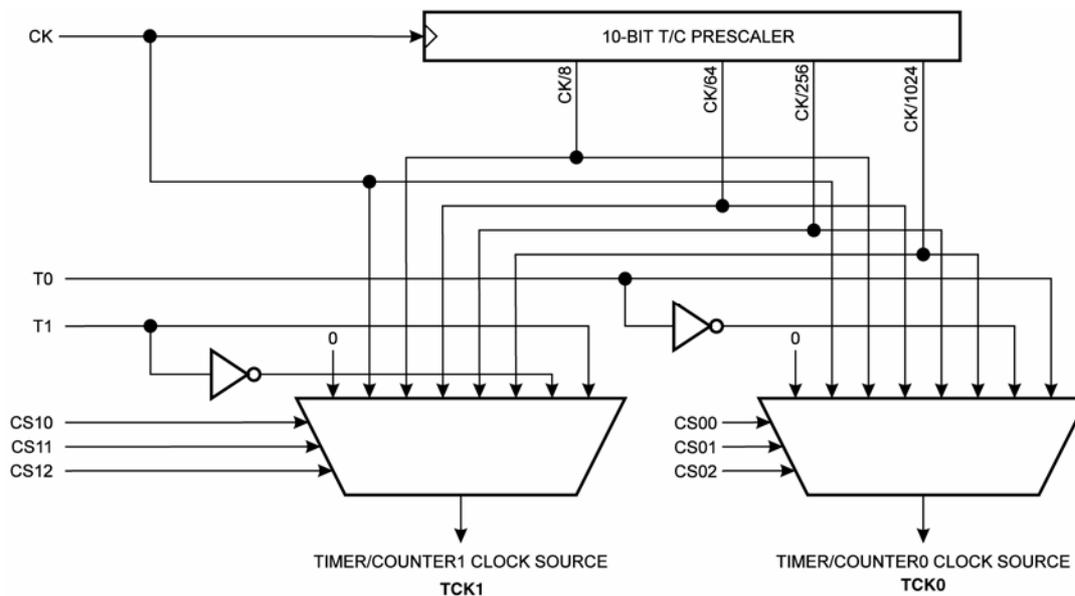


5 Die Zeitgeber/Zähler (Timer/Counter)

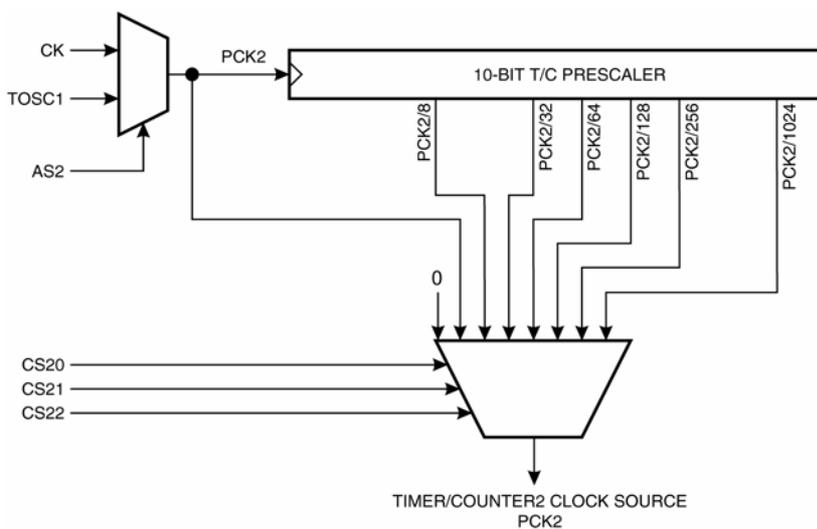
5.1 Der Vorteiler (Prescaler) und die Timer-Steuerlogik

Nachfolgende Abbildung zeigt den Aufbau der Steuerlogik der AVR - Familie. In der Betriebsart Timer können die Taktraten für T/C0, T/C1 mit Hilfe eines Vorteilers und je eines Multiplexers jeweils individuell eingestellt werden.

Dazu dient ein 10 – stufiger Teiler mit Ausgängen bei $\Phi/8$, $\Phi/64$, $\Phi/256$ und $\Phi/1024$. Weiters kann Φ direkt oder die externen Takteingänge T/C0 oder T/C1 über den jeweiligen MUX an den Takteingang von T/C0 oder T/C1 geschaltet werden.



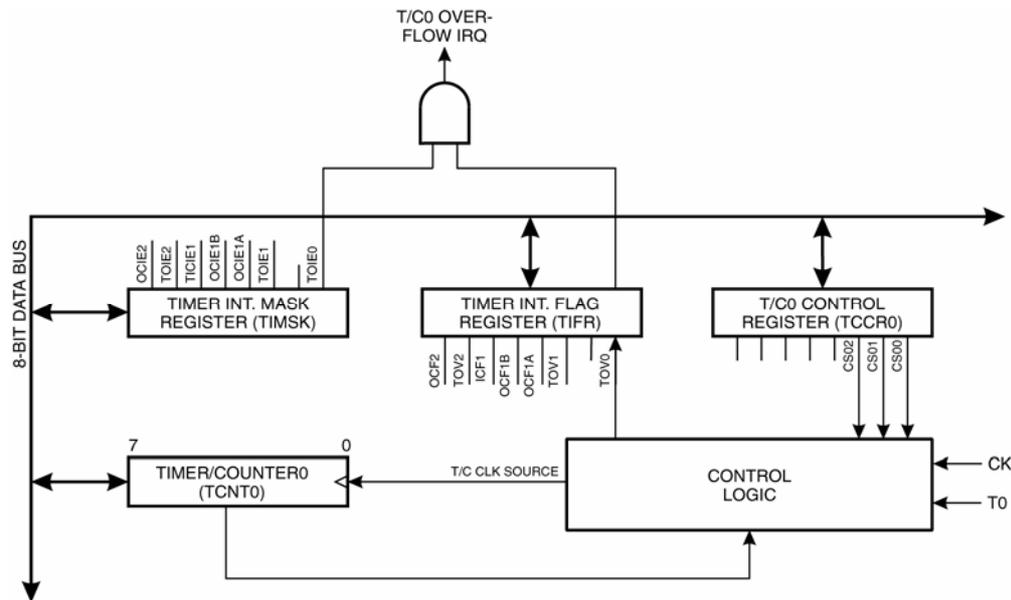
Für den T/C2 gilt nachfolgende Abbildung.



Normalerweise ist die Taktquelle für T/C2 der Systemtakt. In Sonderfällen kann er aber auch asynchron getaktet werden; durch Anschluss einer externen Taktquelle an TOSC1 und TOSC2. Das ermöglicht den Einsatz von T/C2 als RTC (Real Time Clock).

5.2 Der 8-Bit Zeitgeber / Zähler T/C0

Für T/C0 sind insgesamt 4 Register im I/O – Bereich relevant, wie die folgende Abb. zeigt. Es sind dies das eigentliche Zählregister TCNT0, das Steuerregister TCCR0 zur Einstellung des MUX, das TIFR mit dem Überlauf – Flag TOV0 und das TIMSK zur Freigabe/Sperrung des Interrupts mittels des TOIE0 Bits.



Mit den drei Bits b0 bis b2 des Registers TCCR0 kann die Taktquelle gewählt werden; siehe dazu Table 11. Diese Tabelle gilt natürlich auch für T/C1 und T/C2.

Table 11. Clock 0 Prescale Select

CS2	CS1	CS0	Description
0	0	0	Stop, Timer/Counter0 is stopped.
0	0	1	CK
0	1	0	CK / 8
0	1	1	CK / 64
1	0	0	CK / 256
1	0	1	CK / 1024
1	1	0	External Pin T0, falling edge
1	1	1	External Pin T0, rising edge

Sobald über die Bits CS00, CS01 und CS02 im Control-Register TCCR0 für Mux0 eine Multiplexeradresse ungleich 0b000 eingestellt wird, beginnt der Zeitgeber / Zähler T/CO mit jedem an seinem Takteingang eintreffenden Taktimpuls seinen Zählerstand (Register TCNT0) um eins zu inkrementieren.

Wenn der Zählerstand in TCNT0 von \$FF nach \$00 kippt, wird das TIC0-Overflow-Flag TOV0 im Timer/Counter Interrupt Flagregister TIFR gesetzt. Der Programmierer hat zwei Möglichkeiten, diesen Zählerüberlauf auszuwerten:

1. Ständiges Abfragen des Overflow-Flags TOV0 in einer Schleife (Polling). Sobald TOV0 gesetzt ist, ist die gewünschte Zeitspanne abgelaufen und das Programm kann fortgesetzt werden. Vorher sollte TOV0 allerdings durch Einschreiben einer **log. 1** in Bit 1 des TIFR wieder rückgesetzt werden.
2. Wenn das 1-Flag zur globalen Freigabe von Interrupts (Bit 7 im Statusregister SREG) sowie das TOIE0-Flag zur Freigabe des T/C0-Interrupts (Bit 1 im Timer/Counter-Interruptmaskenregister TIMSK) gesetzt sind, wird immer dann ein T/C0-Interrupt ausgelöst, wenn TOV0 durch das Überlaufen des Zählerregisters TCNT0 von \$FF nach \$00 gesetzt wird. In diesem Fall wird TOV0 hardwaremäßig automatisch rückgesetzt, wenn die T/C0- Interruptadresse angesprungen wird. Zu beachten ist, dass die Zeit vom Auftreten der Interruptbedingung TOV0 bis zur Ausführung des ersten Befehls der Interruptroutine wie bei allen Interrupts mindestens 4 Taktzyklen beträgt (Speichern der Returnadresse auf dem Stack und relativer Sprung zur Interruptroutine). Wird während der Interruptanforderung gerade ein Befehl ausgeführt, der mehr als einen Taktzyklus zur Ausführung benötigt, so wird dieser Befehl in jedem Fall noch beendet. Diese Zeit muss bei der Bemessung des Startwertes für TCNT0 ebenfalls berücksichtigt werden.

Anwendungsgebiete für T/C0

T/C0 eignet sich sehr gut zur Erfassung von Zeitabständen. Hierzu wird im Programmablauf TCNT0 mit einem Anfangswert geladen. Anschließend kann T/C0 durch Auswahl des gewünschten Taktes mittels Mux0 gestartet werden. Das Programm wartet entweder im Polling-Betrieb in einer Schleife auf das Overflow-Flag TOV0 im I/O-Register TIFR oder es gibt den T/C0-Interrupt frei. TOV0 zeigt an, dass die gewählte Verzögerungszeit abgelaufen ist und das Programm fortgesetzt werden kann.

Register:

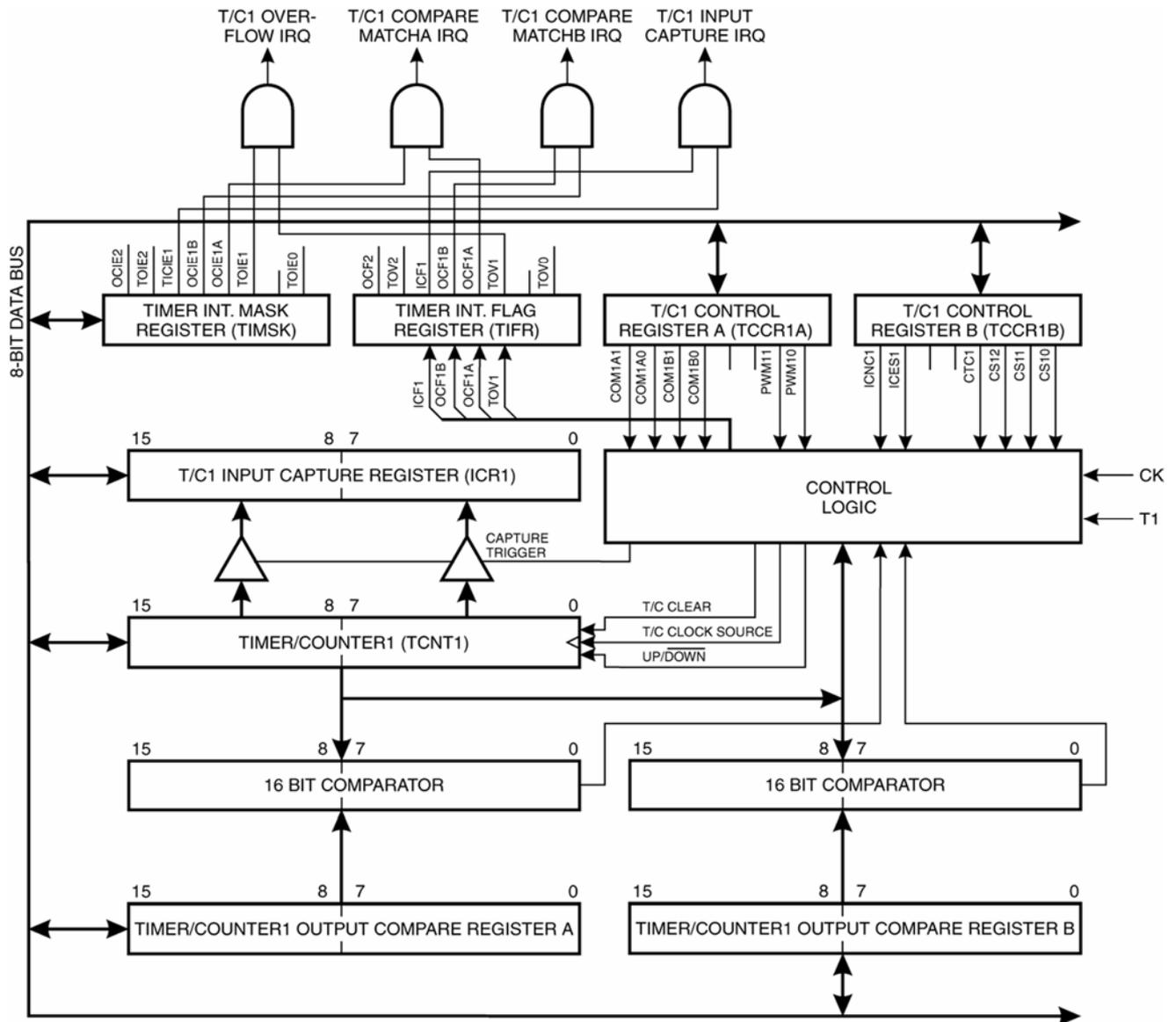
Timer/Counter0 Control Register TCCR0	R/W für b2..b0	0x00 nach Reset
Timer Counter0 Zählregister TCNT0	R/W	0x00 nach Reset

5.3 Der 16-Bit Zeitgeber / Zähler T/C1

Die Bedeutung der einzelnen Register in der folgenden Abb.: Blockdiagramm des T/C1

TCNT1	Das eigentliche Zählregister
TCCR1A	Control-Register zur Festlegung der Reaktion der Ausgangspins OC1A bzw. OC1B im Fall des Übereinstimmens des Zählerstandes in TCNT1 mit dem Compare-Register OCR1A bzw. OCR1B sowie zur Anwahl der Pulsweitenmodulation
TCCR1B	Control-Register zur Einstellung des Multiplexers Mux1 (siehe Abb. 4. 1. 1), zur Freigabe eines Reset-Signals für TCNT1 und zur Einstellung des Capture-Triggers
TIMSK	Timer/Counter-Interruptmaske TIMSK mit dem Bit TOIE1 zur Freigabe des T/C1-Overflow Interrupts, den Bits OCE1A und OCE1B zur Freigabe der Compare-Interrupts A und B und dem Bit TICIE1, das den Capture-Interrupt freigibt
TIFR	Timer/Counter Interrupt Flagregister TIFR mit dem T/C1-Overflow-Flag TOV1, das den Überlauf des Zählerstandes in TCNT1 auf \$0000 anzeigt, den Flags OCF1A und OCF1B zur Anzeige der Übereinstimmung von Zählerstand und Compare-Registern A bzw. B und dem Flag ICF 1, das den Übertrag (Capture) des Zählerstandes in das ICR 1 - Register anzeigt
ICR1	Input Capture Register; beim Auftreten der als aktiv definierten Flanke des Eingangssignals an Pin ICP wird der aktuelle Zählerstand in dieses Register übertragen

OCR1A, OCR1B Compare Register; der Inhalt dieser Register wird kontinuierlich mit dem Zählerstand verglichen. Bei Gleichheit werden die durch TCCR1A definierten Aktionen ausgelöst.



Mit T/C1 können weit komplexere Aufgaben gelöst werden als mit dem bereits vorgestellten T/C0. Dem Anwender stehen hier neben einem Capture-Register auch ein (AT90S2313) bzw. zwei Output-Compare-Register A und B (AT90S4414 und AT90S8515) zur Verfügung.

Der Inhalt der Output-Compare-Register wird hardwaremäßig ständig mit dem Zählerinhalt verglichen, bei Übereinstimmung lassen sich vorwählbare Reaktionen der zugeordneten Ausgangspins aktivieren. Darüber hinaus ist es möglich, Compare-Register A auch zum Rücksetzen des Zählers bei einem gewünschten Zählerstand einzusetzen.

Mit Hilfe der Compare-Register von T/C1 lassen sich zwei pulswidenmodulierte Ausgangssignale erzeugen. Die Auflösung kann dabei zu 8 Bit, 9 Bit oder 10 Bit programmiert werden.

Mit der Input-Capture-Funktion von T/C1 kann - durch ein externes Signal am EingangsPin ICP getriggert - der momentane Zählerstand von T/C1 zum Triggerzeitpunkt im Register ICR1 »eingefroren« werden. Alternativ zum Eingangs-Pin ICP kann auch der Analogkomparator als Triggerquelle für das Input-Capture-Register Verwendung finden. Näheres hierzu auch in Kapitel 9 >Der integrierte Analog-Komparator<

Der 16-Bit-Timer/Counter T/C1 arbeitet im Normalfall als Aufwärtszähler. Einzige Ausnahme ist der Einsatz als Pulsweitenmodulator, bei dem er als Auf- und Abwärtszähler betrieben wird.

Sobald über die Bits CS10, CSU und CS12 im Control-Register TCCR1B (I/O-Adresse \$2E) für Mux1 eine Multiplexeradresse ungleich 0b000 eingestellt wird, beginnt der Zeitgeber / Zähler T/C1 mit jedem an seinem Takteingang eintreffenden Taktimpuls den Inhalt des 16 Bit - Registers TCNT1 um eins zu inkrementieren.

Wenn der Zählerstand in TCNT1 von \$FFFF nach \$0000 kippt, wird das T/C1-OverflowFlag TOV1 im Timer/Counter Interrupt Flagregister TIFR (I/O-Adresse \$38) gesetzt. Der Programmierer hat zwei Möglichkeiten, diesen Zählerüberlauf auszuwerten:

1. Ständiges Abfragen des Overflow-Flags TOV1 in einer Schleife (Polling). Sobald TOV1 gesetzt ist, ist die gewünschte Zeitspanne abgelaufen und das Programm kann fortgesetzt werden. Vorher sollte TOV1 allerdings durch Einschreiben einer log. 1 wieder rückgesetzt werden.
2. Wenn das 1-Flag zur globalen Freigabe von Interrupts (Bit 7 im Statusregister SREG auf der I/O-Adresse \$3F) sowie das TOIE1-Flag zur Freigabe des T/C1-Interrupts (Bit 7 im Timer/Counter -Interruptmaskenregister TIMSK auf der I/O-Adresse \$39) gesetzt sind, wird immer dann ein T/C1-Overflow-Interrupt ausgelöst, wenn TOV1 durch das Überlaufen des Zählerregisters TCNT1 von \$FFFF nach \$0000 gesetzt wird. In diesem Fall wird TOV1 hardwaremäßig automatisch rückgesetzt, wenn die T/C1- Interruptadresse angesprungen wird. Zu beachten ist, dass die Zeit vom Auftreten der Interruptbedingung TOV1 bis zur Ausführung des ersten Befehls der Interruptroutine wie bei allen Interrupts mindestens 4 Taktzyklen beträgt (Speichern der Returnadresse auf dem Stack und relativer Sprung zur Interruptroutine). Wird während der Interruptanforderung gerade ein Befehl ausgeführt, der mehr als einen Taktzyklus zur Ausführung benötigt, so wird dieser Befehl in jedem Fall noch beendet. Diese Zeit muss bei der Bemessung des Startwertes für TCNT1 ebenfalls berücksichtigt werden.

Das Timer/Counter1 Zählregister TCNT1

Das 16 Bit umfassende Timer/Counter1 Zählregister TCNT1 ist in zwei Bytes unterteilt. Das Low-Byte liegt auf dem I/O-Speicherplatz \$2C (RAM: \$4C), das High-Byte auf dem I/O-Speicherplatz \$2D (RAM: \$4D). Beide Bytes können gelesen und beschrieben werden. Nach einem RESET werden sie mit \$00 initialisiert.

TCNT1 stellt den eigentlichen Zähler dar. Solange an Mux1 über die Steuereingänge CS10 ... CS12 eine Adresse ungleich 0b000 anliegt, wird sein Zählerstand mit jedem eintreffenden Taktimpuls um 1 erhöht.

Damit trotz des internen 8-Bit-Busses beide Bytes gleichzeitig in den 16-Bit-Zähler geladen werden können, wird für den Zugriff auf TCNT1 ein internes Register namens TEMP benutzt.

TCNT1-Schreibvorgang:

Bei der Initialisierung von TCNT1 muss zuerst das High-Byte auf die I/O-Adresse \$2D geschrieben werden. Dieses Byte gelangt aber nicht sofort nach TCNT1H, sondern wird vorerst intern in dem Hilfsregister TEMP zwischengespeichert. Anschließend schreibt das Anwenderprogramm das Low-Byte auf die I/O-Adresse \$2C. Dieses Byte wird intern mit dem zwischengespeicherten kombiniert und beide Bytes gelangen simultan in das 16-Bit Register.

TCNT1-Lesevorgang:

Beim Auslesen von TCNT1 muss zuerst das Low-Byte auf der I/O-Adresse \$2C gelesen werden. Beim Lesezugriff auf dieses Byte wird automatisch auch das aktuelle High-Byte des Zählerstandes im Hilfsregister TEMP zwischengespeichert. Beim anschließenden Lesezugriff auf TCNT1H wird das in TEMP zwischengespeicherte Byte zurückgeliefert.

Das Timer/Counter1 Control-Register A TCCR1A

Das Timer/Counter1 Control-Register A (TCCR1A) liegt auf dem I/O-Speicherplatz \$2F (RAM: \$4F). Nach einem RESET wird es mit \$00 initialisiert.

Bei den Prozessoren AT90S8515 und AT90S4414 werden nur die Bits 0 ... 1 und 4 ... 7 des TCCR1A-Registers benutzt. Sie können gelesen und beschrieben werden. Die Bits 2 ... 3 sind von Atmel für spätere Erweiterungen reserviert und können nur gelesen werden (immer log. 0).

Beim AT90S2313 werden nur die Bits 0 ... 1 und 6 ... 7 des TCCR1A-Registers benutzt, da bei diesem Typ Comparator B fehlt. Diese Bits können gelesen und beschrieben werden. Die Bits 2 ... 5 sind von Atmel für spätere Erweiterungen reserviert und können nur gelesen werden (immer log. 0).

Die Bits COM1A1 und COM1A0 für Comparator A bzw. COM1B1 und COM1B0 für Comparator B legen die Reaktion des dem Comparator A zugeordneten Ausgangspins OC1A bzw. des dem Comparator B zugeordneten Ausgangspins, OC1B auf eine Übereinstimmung von Compare-Register und Zählerstand fest. Bei OC1A handelt es sich um Bit 5 / PortD (AT90S8515, AT90S4414) bzw. Bit 3 / PortB (AT90S2313). Dieser Pin muss bei Verwendung als Compare-Funktionsausgang durch das Einschreiben einer log. 1 in das zugeordnete Datenrichtungsregister als Ausgang konfiguriert werden. Die nachfolgende Tab. 4.3.1 zeigt die möglichen Einstellungen für den Compare1-Modus.

Im Falle eines aktiven PWM-Modus haben die Bits 4 ... 7 in TCCR1A eine andere Bedeutung, als in Tab. 4.3.1 angegeben. Näheres hierzu im Abschnitt Pulsweitenmodulation.

Die Bits PWM11 und PWM10 aktivieren den Pulsweitenmodulationsmodus und legen die Auflösung der PWM fest:

Das Timer/Counter1 Control-Register B TCCR1B

Das Timer/Counter1 Control-Register B (TCCR1B) liegt auf dem I/O-Speicherplatz \$2E (RAM: \$4E). Nach einem RESET wird es mit \$00 initialisiert.

Bei den Prozessoren der AVR Basic Line werden nur die Bits 0 ... 3 und 6...7 des TCCR1B-Registers benutzt. Sie können gelesen und beschrieben werden. Die Bits 4 ... 5 sind von Atmel für spätere Erweiterungen reserviert und können nur gelesen werden (immer log. 0).

Die Bits CS10 ... CS12 legen, wie bereits in Kapitel 4.1 >Der Vorteiler (Prescaler) und die Timer-Steuerlogik< ausführlich dargestellt, die Eingangstaktfrequenz für T/C1 fest.

Ist das CTC1-Bit gesetzt, wird T/C1 mit dem nächsten einer Übereinstimmung von Zählerstand und Vergleichsregister A folgenden Systemtaktimpuls auf \$0000 rückgesetzt. Im PWM-Modus hat dieses Bit keine Auswirkungen.

Sobald am Ausgang X von Mux 1 in Abb. 4.3.2 die steigende bzw. fallende (abhängig von Bit ICES1 in Register TCCR1B) Flanke entdeckt wird, wird der augenblickliche Wert von TCNT1 in das ICR1-Register übertragen. Gleichzeitig wird das Input Capture Flag ICH in Register TIFR zur Anzeige der Übernahme gesetzt.

Damit trotz des internen 8-Bit-Busses beide Bytes gleichzeitig aus dem 16-Bit-Register ICR1 gelesen werden können, wird für den Zugriff auf ICR1 ein internes Register namens TEMP benutzt.

Beim Auslesen von ICR1 muss zuerst das Low-Byte ICR1L auf der I/O-Adresse \$24 gelesen werden. Beim Lesezugriff auf dieses Byte wird automatisch auch das High-Byte ICR1H des Registerinhalts im Hilfsregister TEMP zwischengespeichert. Beim anschließenden Lesezugriff auf ICR1H wird das in TEMP zwischengespeicherte Byte zurückgeliefert.

Die Timer/Counter1 Output Compare-Register OCR1A und OCR1B

Die 16 Bit umfassenden Timer/Counter1 Output Compare-Register OCR1A und OCR1B sind in zwei Bytes unterteilt.

Das Low-Byte von OCR1A liegt auf dem I/O-Speicherplatz \$2A (RAM: \$4A), das High-Byte auf dem I/O-Speicherplatz \$2B (RAM: \$4B), das Low-Byte von OCR1B liegt auf dem I/O-Speicherplatz \$28 (RAM: \$48), das High-Byte auf dem I/O-Speicherplatz \$29 (RAM: \$49). Alle vier Bytes können gelesen und beschrieben werden. Nach einem RESET werden sie mit \$00 initialisiert.

Damit trotz des internen 8-Bit-Busses beide Bytes gleichzeitig in die 16-Bit-Register geladen werden können, wird für den Schreibvorgang in die Output Compare-Register ein internes Register namens TEMP benutzt.

Zuerst muss das High-Byte nach OCR1AH bzw. OCR1BH geschrieben werden. Dieses Byte gelangt aber nicht sofort nach OCR1AH bzw. OCR1BH, sondern wird vorerst intern in dem Hilfsregister TEMP zwischengespeichert. Anschließend schreibt das Anwenderprogramm das Low-Byte nach OCR1AL bzw. OCR1BL. Dieses Byte wird intern mit dem zwischengespeicherten kombiniert und beide Bytes gelangen simultan in das 16-Bit Register.

Timer/Counter1 als Pulsweitenmodulator

Wenn T/C1 über das Control-Register TCCR1A als Pulsweitenmodulator konfiguriert wird, bilden bei den Controllern AT90S4414 und AT90S8515 die beiden Output Compare Register OCR1A und OCR1B und TCNT1 einen zweikanaligen, freilaufenden, glitchfreien und phasensynchronen Pulsweitenmodulator mit einer programmierbaren Auflösung von 8, 9 oder 10 Bit. Das PWM-Signal liegt an den beiden Ausgangs-Pins OCR1A bzw. OCR1B. Beim AT90S2313 ist nur 1 Kanal mit dem Ausgangs-Pin OCR1A vorhanden.

T/C1 arbeitet in diesem Modus als Auf- / Abwärtszähler, der zyklisch von \$0000 bis zu seinem Höchstwert TOP und anschließend wieder zurück auf \$0000 zählt. Bei einer programmierten PWM-Auflösung von N Bit errechnet sich TOP zu $TOP = 2^N - 1$;

Die Frequenz f_{PWM} , mit der sich die PWM-Zyklen wiederholen, errechnet sich zu $f_{PWM} = f_{T/C1} / (2^{N+1} - 2)$,

wobei $f_{T/C1}$ die über die Bits CS10 ... CS12 in Register TCCR1B gewählte Eingangsfrequenz von T/C1 und N die über die Bits PWM10 und PWM11 in Register TCCR1A gewählte PWM-Auflösung ist. Tab. 4.3.3 zeigt diese Zusammenhänge.

<i>PWM11</i>	<i>PWM10</i>	<i>Auflösung</i>	<i>TOP-Wert</i>	<i>PWM-Frequenz</i>
0	1	8 Bit	\$00FF (255)	$f_{T/C1} / 510$
1	0	9 Bit	\$01FF (511)	$f_{T/C1} / 1022$
1	1	10 Bit	\$03FF (1023)	$f_{T/C1} / 2046$

Tab. 4.3.3: TOP-Werte und PWM-Frequenz in Abhängigkeit von der PWM-Auflösung

Wenn der Zählerstand in TCNT1 mit dem Wert der 10 niederwertigsten Bits in OCR1A bzw. OCR1B übereinstimmt, werden die Ausgangspins OC1A bzw. OC1B mit dem folgenden Taktimpuls entweder gesetzt oder rückgesetzt, abhängig vom Zustand der Bits COM1A1/COM1A0 bzw. COM1B1/COM1B0 in Register TCCR1A. Tab. 4.3.4 zeigt diese Zusammenhänge.

COM1A1 [COM1B1]	COM1A0 [COM1B0]	Aktion im Fall der Übereinstimmung: [Werte in eckigen Klammern für Comparator B]
0	0	OC1A [OC1B] nicht mit T/C1 verbunden
0	1	OC1A [OC1B] nicht mit T/C1 verbunden
1	0	Nichtinvertierender Pulsweitenmodulator: Beim Aufwärtszählen wird OC1A [OC1B] im Fall der Übereinstimmung auf log. 0 gesetzt, Beim Abwärtszählen wird OC1A [OC1B] im Fall der Übereinstimmung auf log. 1 gesetzt.
1	1	Invertierender Pulsweitenmodulator: Beim Aufwärtszählen wird OC1A [OC1B] im Fall der Übereinstimmung auf log. 1 gesetzt, beim Abwärtszählen wird OC1A [OC1B] im Fall der Übereinstimmung auf log. 0 gesetzt.

Tab. 4.3.4: Auswahlmöglichkeiten für den Compare1-Modus bei PWM-Betrieb

Beim nichtinvertierenden Pulsweitenmodulator entspricht der Tastgrad g des Rechtecksignals am PWM-Ausgangspin dem Wert $n / (2N - 1)$ bei einem angenommenen Wert n im zugeordneten Output Compare Register und einer PWM-Auflösung von N Bit.

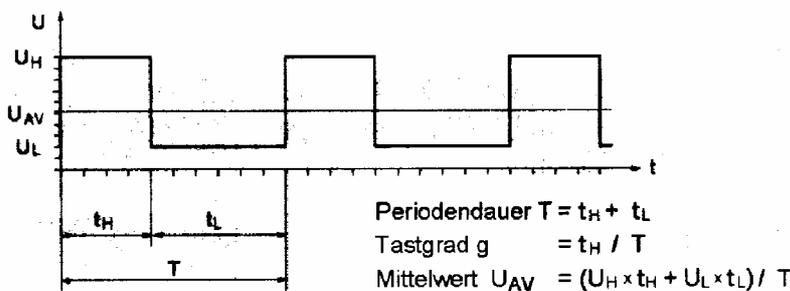


Abb. 4.3.4: Definition der Periodendauer T , des Tastgrades g und des arithmetischen Mittelwertes U_{AV} einer Rechteckspannung U

Hat das Vergleichsregister OCR1A bzw. OCR1B den Inhalt TOP oder 0, so wird der entsprechende Ausgangspin gemäß den nachfolgenden Regeln in Tab. 4.3.5 ständig auf log. 0 oder log. 1 gehalten:

COM1A1 [COM1B1]	COM1A0 [COM1B0]	OCR1A [OCR1B]	Ausgang OC1A [OC1B]
1	0	0	0
1	0	TOP	1
1	1	0	1
1	1	TOP	0

Tab. 4.3.5: PWM-Ausgang für die Sonderfälle $OCR1A[B] = TOP$ oder $OCR1A[B] = 0$

Abb. 4.3.5 zeigt anhand einer fiktiven 3 Bit - PWM die Generierung eines nichtinvertierten und eines invertierten PWM-Ausgangssignals für den PWM-Ausgang OC1B. An diesem vereinfachten Beispiel lassen sich die Verhältnisse anschaulich darstellen. Diagramm A zeigt exemplarisch das Stufensignal des Zählerinhalts TCNT1, Diagramm B zeigt das nichtinvertierte und Diagramm C zeigt das invertierte PWM-Ausgangssignal.

Die Periodendauer T_{PWM} ist in diesem Beispiel gemäß der o.a. Beziehung $T_{PWM} = T_{T/C1} \cdot (2^{N+1} - 2)$. Mit $N = 3$ besteht eine Periode des PWM-Signals damit aus 14 Perioden des Taktsignals $f_{T/C1}$ am Eingang von TCNT1. In diesem Beispiel hat das Output Compare Register OCR1B den Inhalt 5. Ausgehend von seinem Anfangswert 0 weist TCNT1 nach 5 Taktimpulsen den gleichen Zählerstand auf. Mit dem nächsten Taktimpuls nach Erkennen der Übereinstimmung wird in Diagramm B der Ausgang OC1B gemäß Tabelle 4.3.4 auf log. 0 gesetzt. TCNT1 wird weiter inkrementiert, bis der Wert TOP erreicht ist, der bei einer 3 Bit - PWM nach obiger Gleichung 7 beträgt. Sobald der Wert TOP erreicht ist, kehrt sich die Zählrichtung um und TCNT1 zählt abwärts. Nach dem insgesamt 9. Taktimpuls vom Startwert 0 aus gerechnet, weist TCNT1 erneut den gleichen Inhalt wie OCR1B auf. Mit dem nächsten Taktimpuls nach Erkennen der Übereinstimmung wird der Ausgang OC1B gemäß Tabelle 4.3.4 auf log. 1 gesetzt. TCNT1 wird weiter dekrementiert, bis der Wert 0 wieder erreicht ist. Dies ist der Fall nach dem insgesamt 14. Taktimpuls vom Startwert 0 aus gerechnet. Damit ist eine Periode des PWM-Signals beendet und die Zählrichtung kehrt sich abermals um, TCNT1 zählt wieder aufwärts. Wie Diagramm B in Abb. 4.3.5 zu entnehmen ist, beträgt der High-Anteil des Ausgangssignals 10 und der Low-Anteil 4 Taktperioden. Damit ergibt sich ein Tastgrad von $g = 10/14$ bzw. $g = 5/7$.

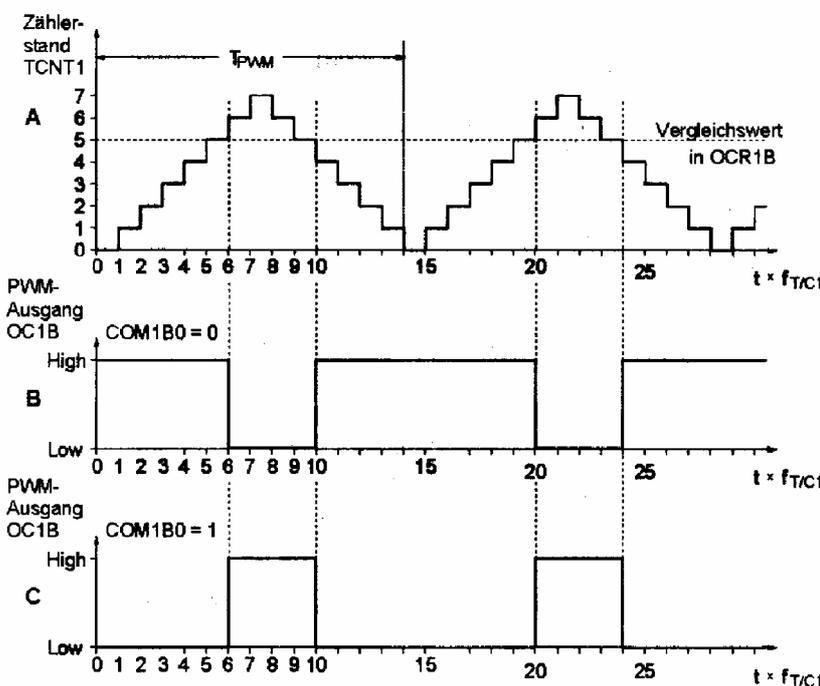


Abb. 4.3.5: Verfahren zur Generierung des nichtinvertierten und des invertierten PWM-Ausgangssignals

Diagramm C in Abb. 4.3.5 zeigt analog dazu die Verhältnisse für das invertierte PWM-Ausgangssignal nach Tab. 4.3.4. Wie bereits zu Beginn dieses Abschnitts erwähnt, ist das mittels T/C1 erzeugte PWM-Signal phasensynchron. Damit wird das Auftreten von undefinierten Impulslängen des Ausgangssignals bei einer Änderung des Inhalts der Vergleichsregister verhindert. Dies wird dadurch erreicht, dass die niederwertigsten 10 Bit bei einem Schreibzugriff auf OCR1A bzw. OCR1B nicht sofort in das Register gelangen, sondern vorerst temporär zwischengespeichert und erst beim Zählerstand TOP in die Register eingeschrieben werden. Abb. 4.3.6 zeigt hierfür ein Beispiel für ein nichtinvertiertes PWM-Ausgangssignal.

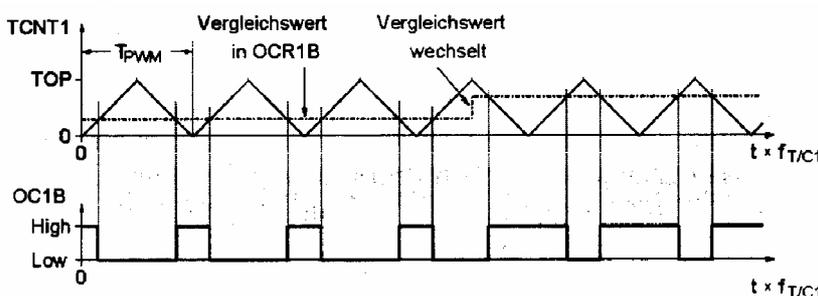


Abb. 4.3.6: Synchronisierter Wechsel des Inhalts des Vergleichsregisters OCR1B

Im PWM-Modus wird das Overflow-Flag TOV1 gesetzt, wenn der Zähler beim Erreichen des Zählerstandes 0 seine Richtung umkehrt. Dieser T/C1-Interrupt bei Überlauf wird wie im normalen Zählerbetrieb dann ausgelöst, wenn das globale Interruptflag I im Statusregister SREG und das TOIE1-Flag im Register TIMSK gesetzt sind. Entsprechend werden die Interrupts bei Übereinstimmung von TCNT1 und OCR1A bzw. OCR1B dann ausgelöst, wenn das globale Interruptflag I und das OCIE1A- bzw. das OCIE1B-Flag im Register TIMSK gesetzt sind.

Anwendungsgebiete für T/C1

T/C1 eignet sich sehr gut zur Erfassung und Messung von Zeitabständen und zur Ausgabe von genauen Frequenzen und Impulslängen. Im PWM-Modus kann mit T/C1 ein Rechteckimpulsgenerator mit einstellbarem Impuls-/ Pausenverhältnis realisiert werden. Bei nicht zu großen Anforderungen an die Geschwindigkeit lässt er sich auch sehr gut als einfacher D/A-Wandler verwenden. Zu diesem Zweck genügt es im einfachsten Fall ein passives Tiefpassfilter an den PWM-Ausgangspin anzuschließen.

Die Werte für R und C wählt man so, dass ihre Zeitkonstante $\tau = R \cdot C$ deutlich über der Periodendauer der PWM-Frequenz liegt, um die im Rechtecksignal enthaltene Grundwelle und ihre Oberwellen weitestgehend zu unterdrücken. Zu groß sollte man die Zeitkonstante jedoch auch nicht wählen, da dies die Einschwingzeit auf einen neuen Ausgangswert unnötig verlängert. In der Praxis wählt man $\tau = T_{\text{PWM}}$ (10...1000).

Große Anforderungen an die Genauigkeit dürfen an diesen D/A-Wandler in allerdings nicht gestellt werden. Grund dafür ist die Toleranz der Ausgangsspannung an OC1A, bei der es sich schließlich um ein digitales Signal handelt. U_{OC1A} erreicht weder ganz den Wert der Betriebsspannung V_{CC} noch den Wert 0V. Da U_{OC1A} von Materialeigenschaften, der Temperatur und der Belastung abhängig ist, kann sich ihr Wert in einem relativ großen Bereich bewegen. Abhilfe ließe sich hier mit der Schaltung in Abb. 4.3.11 schaffen. Hier wirkt die PWM-Ausgangsspannung nicht unmittelbar als Referenz für den D/A-Wandler, sondern es wird mittels eines CMOS-Schalters wie dem CD4053 zwischen einer stabilen Referenzspannung und einem sauberen Massesignal umgeschaltet.

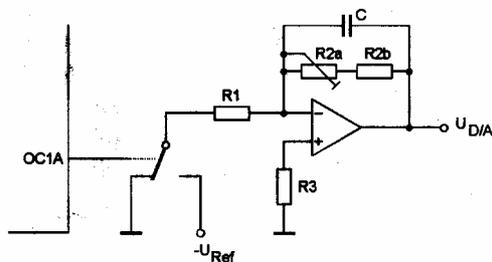


Abb. 4.3.11: Tiefpaßfilter 1. Ordnung mit Operationsverstärker bei der Verwendung von T/C1 als D/A-Wandler im PWM-Betrieb

Die 3 dB - Grenzfrequenz des Tiefpassfilters in Abb. 4.3.11 beträgt $f_g = 1 / 2\pi R_2 C$. Sie sollte im Bereich $f_{\text{PWM}} / (10 \dots 1000)$ liegen.

Die Verstärkung der Tiefpassschaltung liegt bei $A = - R_{2\text{ges}} / R_1$. Sie ist negativ, da sie auf der invertierenden Grundschaltung eines Operationsverstärkers basiert. Um positive Ausgangswerte des D/A-Wandlers zu erreichen, schaltet der CMOS-Schalter deshalb zwischen 0 V und einer negativen Referenzspannung um. Die Verwendung des Operationsverstärkers macht die Schaltung weitgehend belastungsunabhängig. Der Innenwiderstand des CMOS-Schalters liegt in Reihe zu R_1 und geht in die Verstärkung mit ein. Sein Wert liegt typischerweise bei 60...100 Q. Der Widerstand R_2 ist deshalb zweigeteilt und besteht aus der Reihenschaltung des Trimmers R_{2a} und des Festwiderstands R_{2b} . Mit R_{2a} lässt sich die Ausgangsspannung in einem gewissen Bereich zur Kompensation des Schalterwiderstandes und der Toleranzen der Referenzspannung kalibrieren. R_{2a} wählt man in der Regel zu 10 ... 20% von $R_{2\text{ges}}$. Der Widerstand R_3 dient zur Kompensation des Offsetstromes und sollte den Wert der Parallelschaltung von R_2 und R_1 aufweisen.

Quellenangabe:

Dieses Skriptum entstand durch eine komprimierte Darstellung der Inhalte des Buches AVR-RISC Mikrocontroller von Wolfgang Trampert aus dem Francis' Verlag.