Timers

ATmega1284P

Networks and Embedded Software

Module 4.3.4

by Wolfgang Neff

Timers (1)

- Implementation
  - Two 8-bit timers/counters
    - TC0 and TC2
  - Two 16-bit timers/counters
    - TC1 and TC3
  - Each timer has a PWM module
  - Each timer has two compare registers

Timers (2)

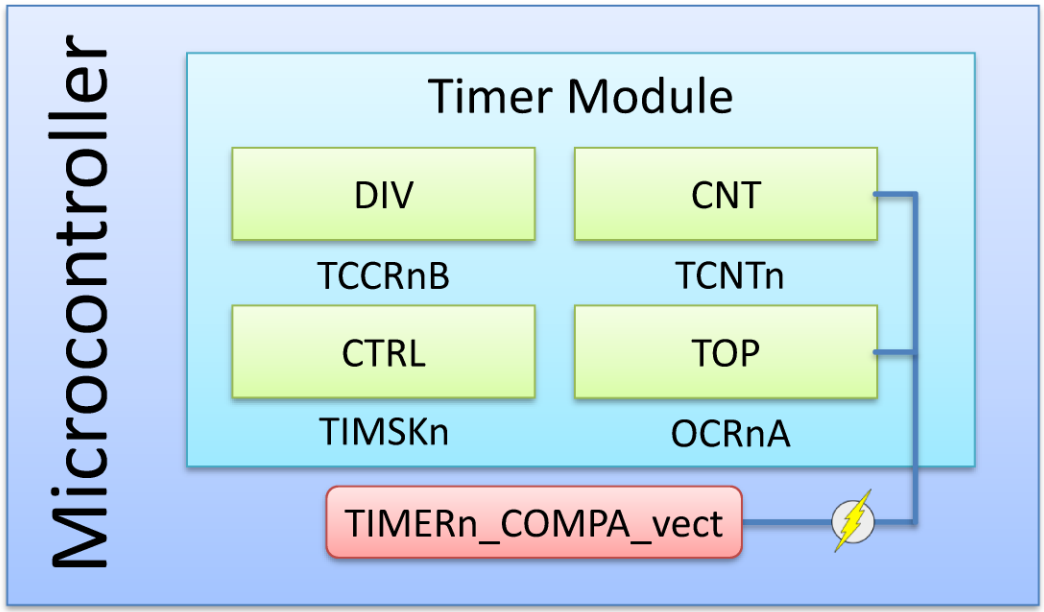
- Modes of Operation
  - Normal Mode
    - BOTTOM = 0x00 resp. 0x0000
      - Can be changed in ISR (software → less precise)
    - TOP = MAX = 0xFF resp. 0xFFFF
    - Timer/Counter Overflow Interrupt
    - Basic Formulas (n: clock divider)
      - $T = (MAX - BOTTOM + 1) = \frac{n \cdot (MAX - BOTTOM + 1)}{f_{CPU}}$
      - $F = \frac{1}{T} = \frac{f_{CPU}}{n \cdot (MAX - BOTTOM + 1)}$

Timers (3)

- Modes of Operation (continued)
  - Clear Timer on Compare Match (CTC)
    - BOTTOM = 0x00 resp. 0x0000
    - TOP = OCRnA
    - Timer/Counter Compare Match Interrupt
    - Basic Formulas (n: clock divider)
      - $T = (OCRnA + 1) = \frac{n \cdot (OCRnA+1)}{f_{CPU}}$
      - $F = \frac{1}{T} = \frac{f_{CPU}}{n \cdot (OCRnA+1)}$

Timers (4)

- Register Mapping (CTC)



n: 0, 1, 2, 3

Timers (5)

- Register Description
  - TCNTn – Counter Value
  - OCRnA – Output Compare Register A
  - TIMSKn – Timer Interrupt Mask Register
    - Bit 1 – OCIEA: Compare A Match Interrupt Enable
    - Bit 0 – TOIE: Overflow Interrupt Enable

Timers (6)

- Register Description (continued)
  - TCCR1B – Control Register B
    - Bit 4:3 – WGM: Waveform Generation Mode
      - 0: CTC off
      - 1: CTC on
    - Bit 2:0 – CS: Clock Select
      - 0: timer is off
      - 1: no prescaling
      - 2: prescaler division factor is 8
      - 3: prescaler division factor is 64
      - 4: prescaler division factor is 256
      - 5: prescaler division factor is 1024

Timers (7)

- Configuration Example 1
  - Timer 1, normal Mode, two interrupts per second
  - f_{CPU} = 11.0592 MHz, Prescaler *n* = 256
  - Calculate BOTTOM
    - $TPS = F = \frac{f_{CPU}}{n \cdot (MAX - BOTTOM + 1)}$
    - $BOTTOM = MAX - \frac{f_{CPU}}{n \cdot TPS} + 1 = 43937$

Timers (8)

- Configuration Example 1 (continued)
  - Set BOTTOM
    - `TCNT1 = 43937;`
  - Enable timer overflow interrupt
    - `TIMSK1 = (1 << TOIE1);`
  - Enable timer by setting prescaler
    - `TCCR1B = (4 << CS10);`

Timers (9)

- Configuration Example 1 (finished)
  - Implement interrupt service routine
    - ```
      ISR(TIMER1_OVF_vect) {
        …
        TCNT1 = 43937; // Set BOTTOM
      }
      ```
      ```

Timers (10)

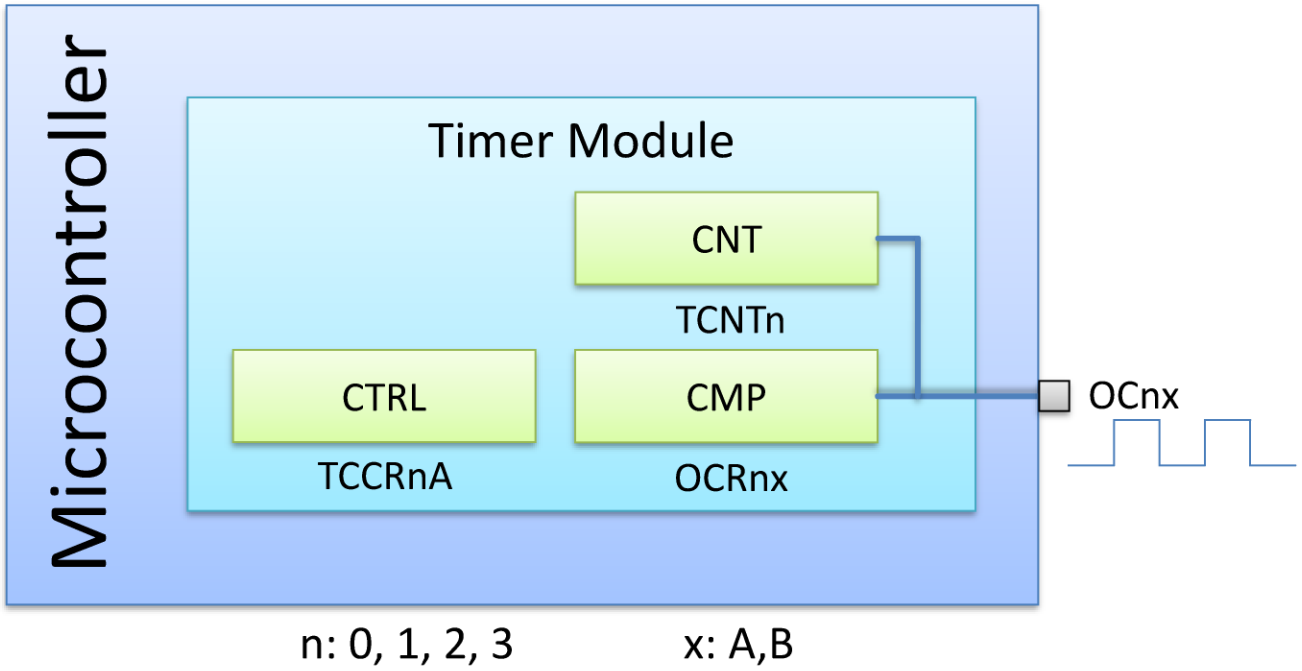
- Configuration Example 2
  - Timer 1, CTC, two interrupts per second
  - f_{CPU} = 11.0592 MHz, Prescaler *n* = 256
  - Calculate TOP
    - $TPS = F = \frac{f_{CPU}}{n \cdot (OCRnA+1)}$
    - $TOP = OCRnA = \frac{f_{CPU}}{n \cdot TPS} - 1 = 21599$

Timers (11)

- Configuration Example 2 (continued)
  - Set TOP
    - `OCR1A = 21599;`
  - Enable timer overflow interrupt
    - `TIMSK1 = (1 << OCIE1A);`
  - Select mode and enable timer
    - `TCCR1B = (1 << WGM12) | (4 << CS10);`
  - Implement interrupt service routine
    - `ISR(TIMER1_COMPA_vect) { … }`

PWM (1)

- Register Mapping



PWM (2)

- Register Description
  - OCRnA – TCn Output Compare Register A
  - OCRnB – TCn Output Compare Register B
  - TCCRnA – TCn Control Register A
    - Bit 7:6 – COMA: Compare Output Mode
      - 0: OCnA disconnected
      - 2: non-inverting mode (clear OC on compare match, 1 → 0)
      - 3: inverting mode (set OC on compare match, 0 → 1)

PWM (3)

- Register Description (continued)
  - TCCR1A – TCn Control Register A (continued)
    - Bit 7:6 – COMB: Compare Output Mode
      - 0: OCnA disconnected
      - 2: non-inverting mode (clear OC on compare match, 1 → 0)
      - 3: inverting mode (set OC on compare match, 0 → 1)
    - Bit 1:0 – WGM: Waveform Generation Mode
      - 3: Fast PWM (8-bit timers, only)
      - Consult datasheet for details and 16-bit timers

PWM (4)

- Configuration Example (50% duty cycle, non-inv.)
  - Configure PWM output pin
    - `LED_DDR = (1 << DDB3);`
  - Turn non-inverting fast PWM on
    - `TCCR0A = ((2 << COM0A0) | (3 << WGM00));`
  - Set duty cycle to 50%
    - `OCR0A = 0x7F;`
  - Enable clock
    - `TCCR0B = (1 << CS00);`

Timers (12)

- Register Summary

Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TCCRnA	**COMnA1**	**COMnA0**	**COMnB1**	**COMnB0**	**-**	**-**	**WGMn1**	**WGMn0**
TCCRnB	ICNCn*	ICESn*	-	WGMn3*	**WGMn2**	**CSn[2:0]**		
TCCRnC*	FOCnA	FOCnB	-	-	-	-	-	-
TCNTnL	**TCNTn[7:0]**							
TCNTnH*	**TCNTn[15:8]**							
ICRnL*	ICRn[7:0]							
ICRnH*	ICRn[15:8]							

* 16-bit timers, only. 8-bit timers: FOCnA → ICNCn, FOCnB → ICESn, TCNTnL → TCNTn

Timers (13)

- Register Summary (continued)

Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
OCRnAL	**OCRnA[7:0]**							
OCRnAH*	**OCRnA[15:8]**							
OCRnBL	OCRnB[7:0]							
OCRnBH*	OCRnB[15:8]							
TIMSKn	**-**	-	ICIE*	-	-	OCIEB	**OCIEA**	**TOIE**
TIFRn			ICF*			OCFB	OCFA	TOV

* 16-bit timers, only. 8-bit timers: OCRnAL → OCRnA, OCRnBL → OCRnB

Timers (14)

- Interrupt Summary

Source	Description
TIMERn_OVF_vect	**Timer/Counter1 Overflow Interrupt**
TIMERn_COMPA_vect	**Timer/Counter Compare Match A Interrupt**
TIMERn_COMPB_vect	Timer/Coutner Compare Match B Interrupt
TIMERn_CAPT_vect*	Timer/Counter Capture Event Interrupt

* 16-bit timers, only