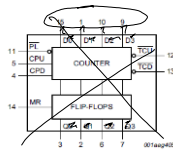
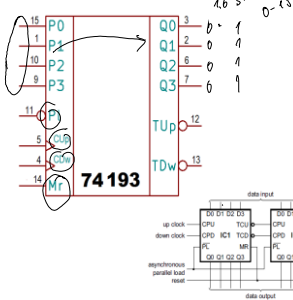
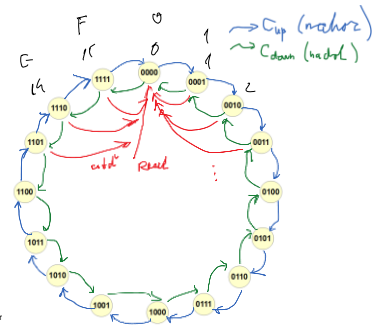


74LS193

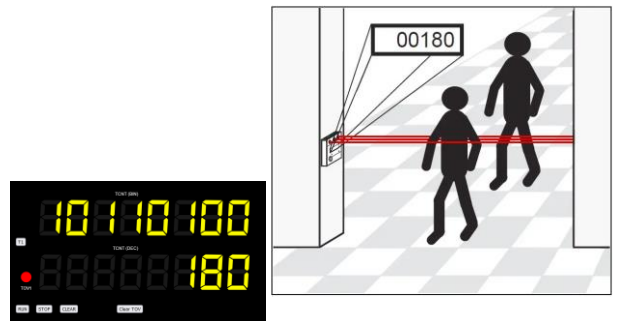
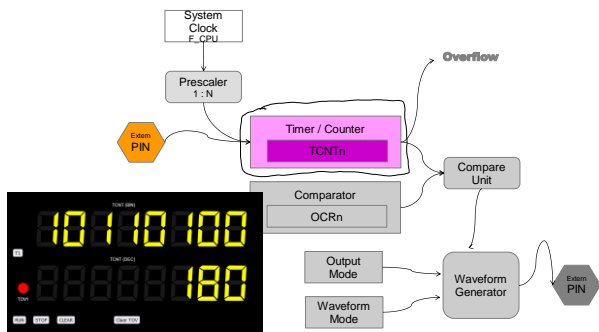


P0 - P3 data input
 Q0 - Q3 data output
 /PL - async. Parallel Load
 MR - async. Master Reset
 Cup - count UP
 Cdw - count DOWN
 TUp - Terminal (carry) UP
 TDw - Terminal (carry) DOWN

74LS193



Stavový diagram

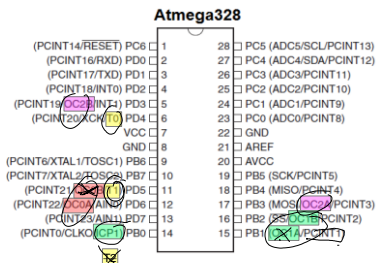


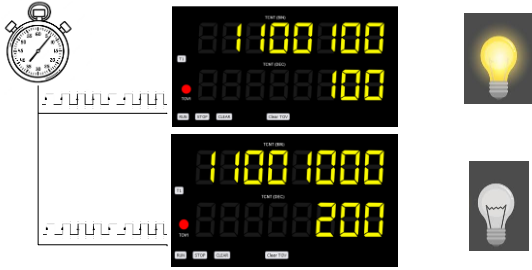
AVR Timer Overview

	Timer T0	Timer T1	Timer T2
rozišenie	8-bit	16-bit	8-bit
min-max	0-255	0-65535	0-255
ext. vstup	1	1	0
PWM výstup	2	2	2
Input capture	0	1	0
preddelič	1, 8, 64, 256, 1024	1, 8, 64, 256, 1024	1, 8, 32, 64, 128, 256, 1024

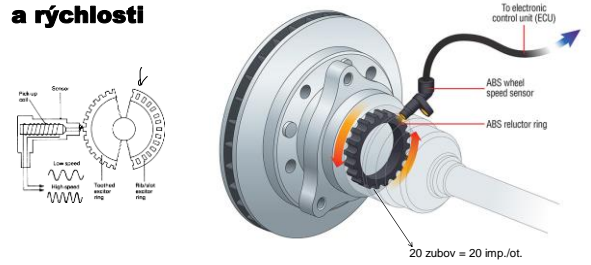
*T2 má navyše možnosť pracovať s externým 32kHz kryštálom nezávisle od Fosc - RTC

AVR Timer Overview

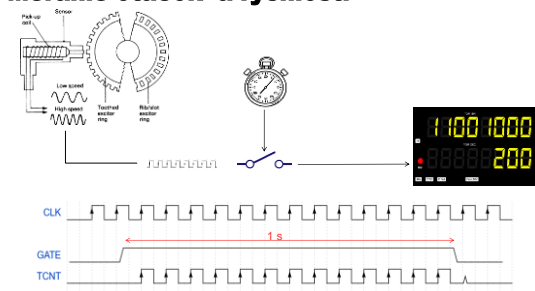




Meranie otáčok a rýchlosti



Meranie otáčok a rýchlosti



Spracovanie vstupného signálu

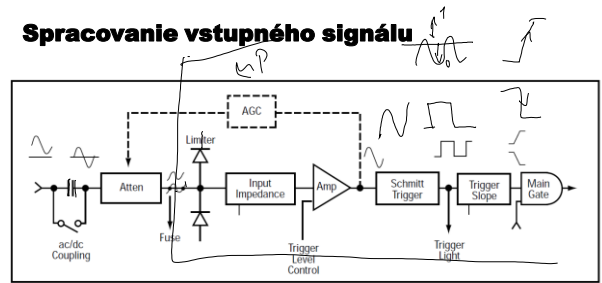
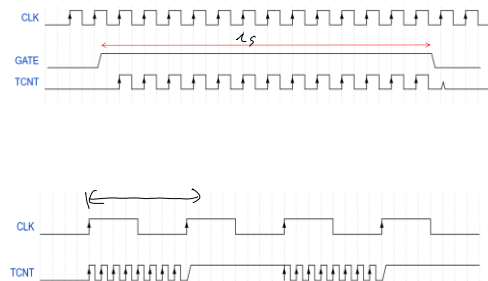
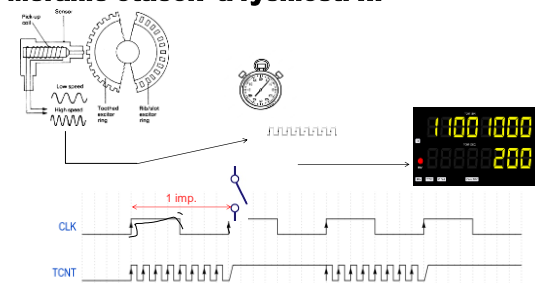
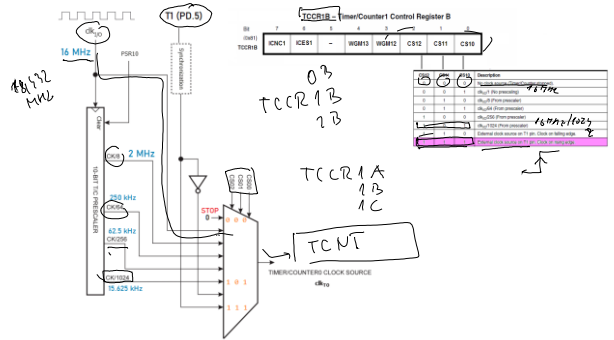
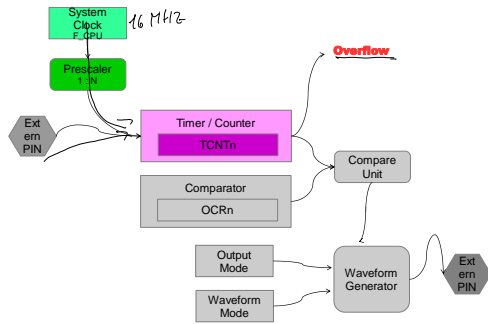


Figure 13. Input Signal Conditioning

Meranie otáčok a rýchlosti II.





AVR Timer z programátorského hľadiska

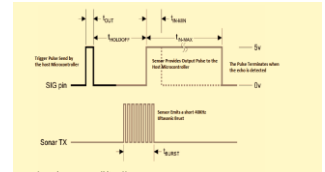
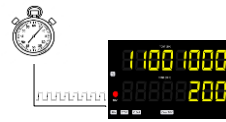
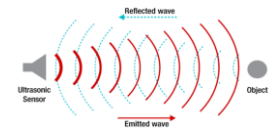
TCNTn - Timer/Counter Register
TCCRn - Timer/Counter Control Register
OCRn - Output Compare Register
ICRn - Input Capture Register
TIFRn - Timer Interrupt Flag Register
TOVn - Timer Overflow Flag
OCFn - Output Compare Match Flag
 % Počítadlo TI externý vstup:

```

DDRD |= ~(1<<PDS); // PORTD.5 (PDS) input pull-up ON
PORTD |= (1<<PDS); //
TCNT1 = 0x0000; // initialize (CLEAR) counter
TCCR1B = 0b00001111; // TI CLK = external clock source on pin TI, rising edge
printf("Počítadlo: %04X, TCNT1); // %X %d %d
          0003 600F          HEX %d %d
  
```

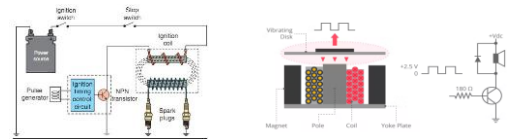
Handwritten notes: TCNT0 - 16b, TCNT1 - 16b, TCNT2 - 8b. A circuit diagram shows a push-button connected to PORTD.5 (PDS) with a pull-up resistor.

Ultrazvukový parkovací senzor



Ak ho namierim do oblchy, neodrazí sa o počítadlo

Generovanie impulzov / presnej frekvencie

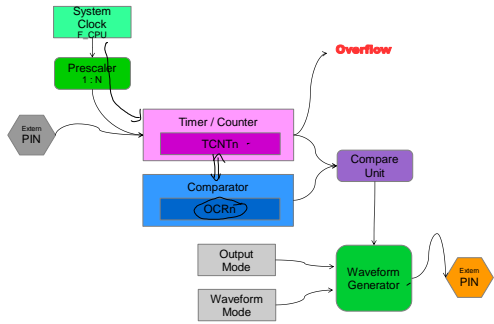


Softvérový generátor 500Hz

```

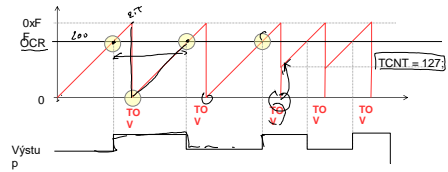
while(1)
{
  delay(1); // 1 ms delay
  set_bit(PORTB,PB1);
  delay(1); // 1 ms delay
  clear_bit(PORTB,PB1);
}
  
```





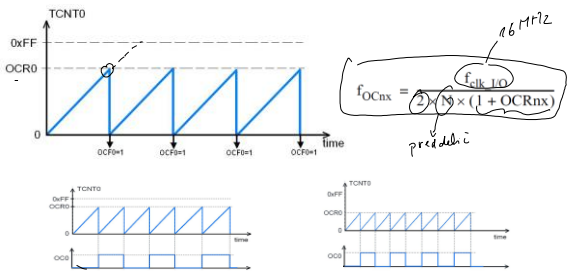
Mode 0: Normal mode

Togglé

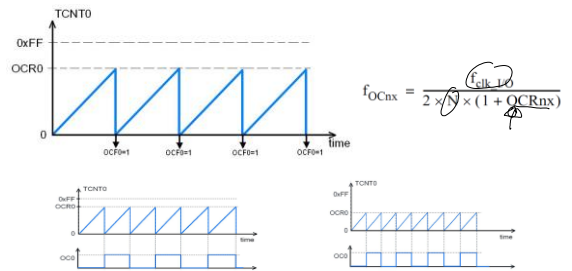


COM01	COM00	Description
0	0	The normal operation, $\overline{OC0}$
1	0	Toggle OC0 on compare match
1	1	Clear OC0 on compare match
1	1	Set OC0 on compare match

Mode 1: Clear Timer on Compare Match (CTC)

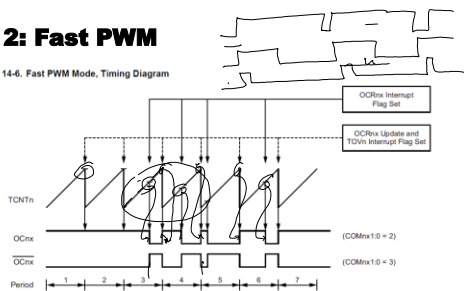


Mode 1: Clear Timer on Compare Match (CTC)



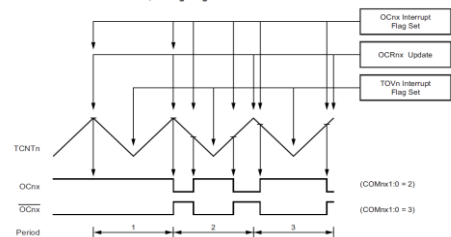
Mode 2: Fast PWM

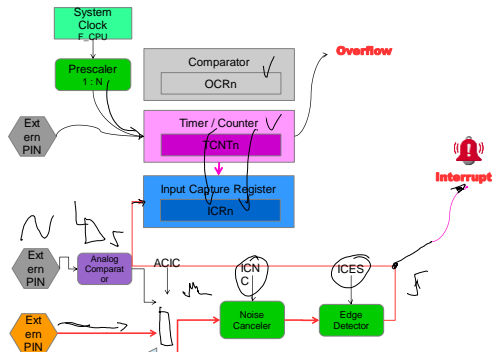
Figure 14-6. Fast PWM Mode, Timing Diagram



Mode 3: Phase Correct PWM

Figure 14-7. Phase Correct PWM Mode, Timing Diagram





Príklad 1:

Timer0 / výstup OC0A (pin 6 portu D) / režim Clear Timer on Compare Match (CTC)

Handwritten calculation for frequency:

$$f = 440 \text{ Hz} \Rightarrow \frac{16 \text{ MHz}}{2 \cdot N \cdot (1 + OCRn)} = 440$$

Handwritten table for OCRn values:

OCRn	255
OCRn	70
OCRn	16
OCRn	17

f_clkIO = 16 MHz

N = ? / OCR0A = ?
 N = (16 000 000 / 256) / OCR0A = (0,1,2,...,255)

Handwritten calculation for N:

$$2440 = \frac{16000000}{2 \cdot N \cdot (1 + OCRn)}$$

Handwritten calculation for OCRn:

$$\frac{1}{2 \cdot N \cdot OCRn} = \frac{1}{164}$$

Handwritten calculation for OCRn:

$$OCRn = \frac{16000000}{2 \cdot 64 \cdot 164} = 764,400$$

požadovaná f_oc0A = 440 Hz (komorné A) 441,02

Handwritten calculation for f_out:

$$f_{out} = \frac{16000000}{2 \cdot 256 \cdot (1 + 255)} = 440,14$$

Handwritten calculation for f_out:

$$f_{out} > \frac{16000000}{2 \cdot 64 \cdot (1 + 16)} = 459$$

Príklad 1:

Timer0 / výstup OC0A (pin 6 portu D) / režim Clear Timer on Compare Match (CTC)

$$f_{OCnHz} = \frac{f_{clkIO}}{2 \cdot N \cdot (1 + OCRn)}$$

f_clkIO = 16 MHz

požadovaná f_oc0A = 440 Hz (komorné A)

```

DDRD |= (1 << PD6); // port D.6 pin ako výstup
TCCR0A |= (1 << COM0A0) // toggle pin on match
|(1 << WGM01); // timer 0 in CTC mode
TCCR0B |= (1 << CS02); // set prescaler 256
OCR0A = 70; // set output freq. 440 Hz
  
```

N	OCR0A	f_out
64	283	>255!
256	70	440,14
	71	434,02
1024	16	459,56
	17	434,03

Príklad 2: