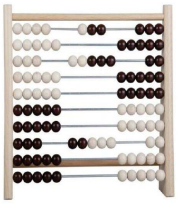
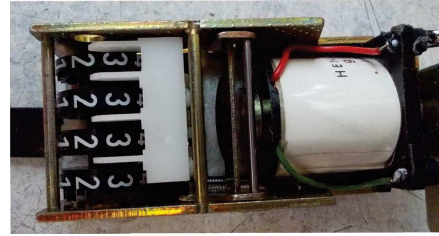


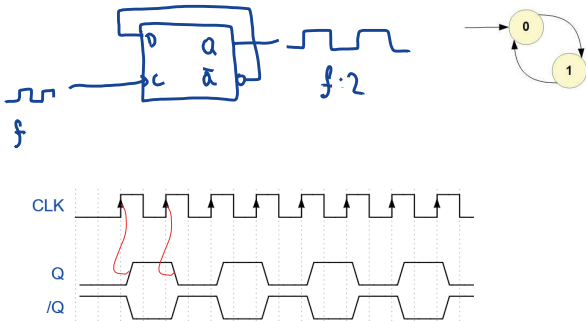
# Prednáška 5: Počítadlá / časovače Timer / Counter



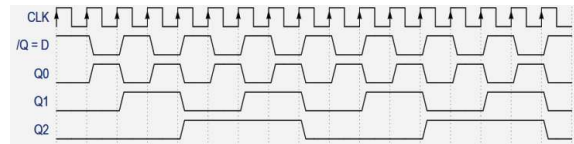
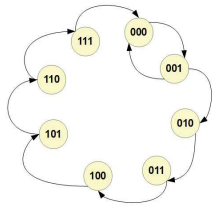
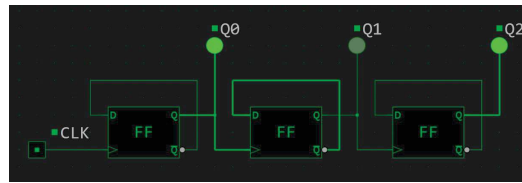
## Mechanické počítadlo



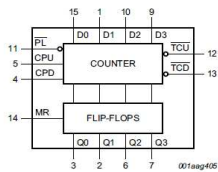
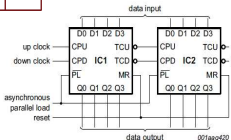
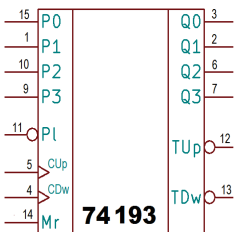
## Delička frekvencie



## Binárne 3-bitové počítadlo



## 74LS193



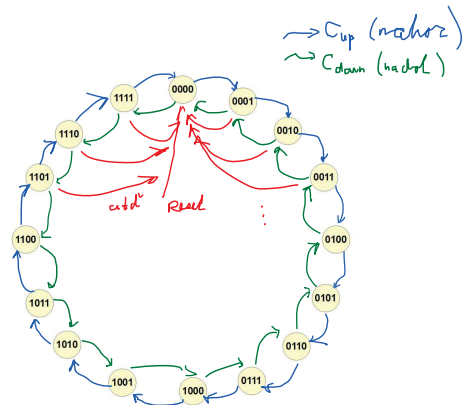
P0 – P3 data input  
Q0 – Q3 data output

/PL – async. Parallel Load  
MR – async. Master Reset

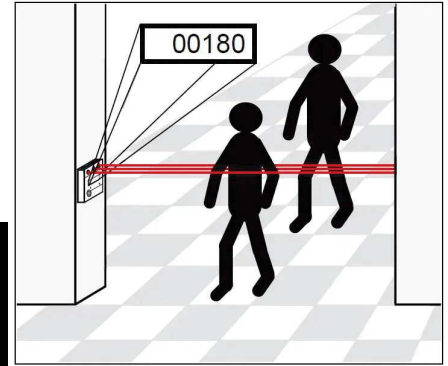
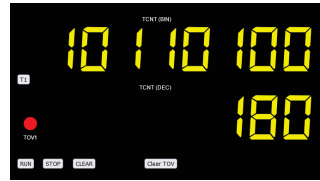
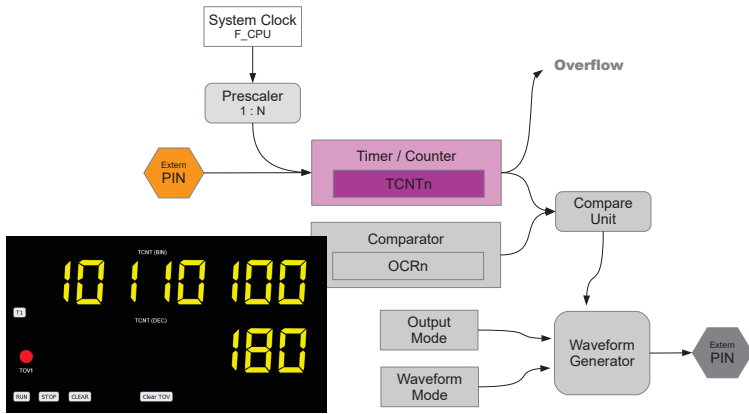
Cup – count UP  
Cdw – count DOWN

Tup – Terminal (carry) UP  
Tdw – Terminal (carry) DOWN

## 74LS193



Stavový diagram



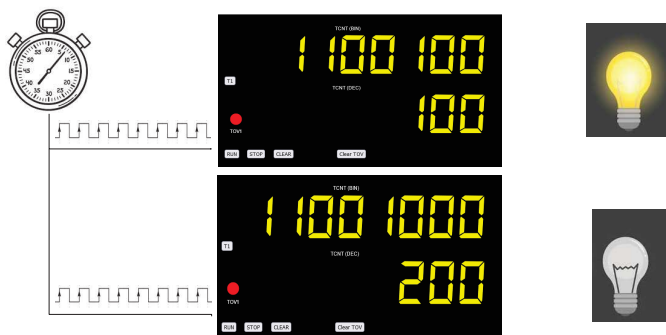
## AVR Timer Overview

	Timer T0	Timer T1	Timer T2
rozlíšenie	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
preddeľič	1, 8, 64, 256, 1024	1, 8, 64, 256, 1024	1, 8, 32, 64, 128, 256, 1024

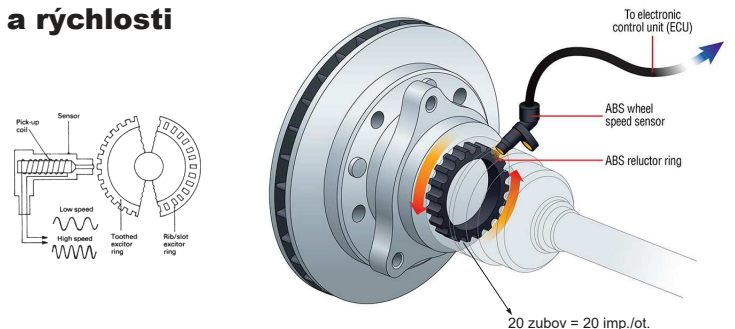
\*T2 má navyše možnosť pracovať s externým 32kHz kryštálom nezávisle od Fosc - RTC

## AVR Timer Overview

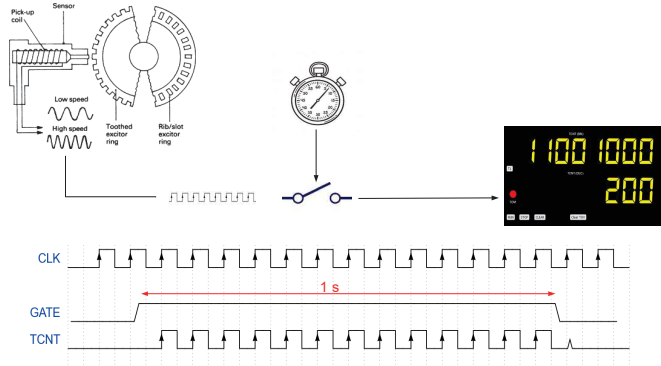
Atmega328	
(PCINT14/RESET) PC6	1
(PCINT16/RXD) PD0	2
(PCINT17/TXD) PD1	3
(PCINT18/INT0) PD2	4
(PCINT19/OC2B/INT1) PD3	5
(PCINT20/XCK/T0) PD4	6
VCC	7
GND	8
(PCINT6/XTAL1/TOSC1) PB6	9
(PCINT7/XTAL2/TOSC2) PB7	10
(PCINT21/OC0B/T1) PD5	11
(PCINT22/OC0A/AIN0) PD6	12
(PCINT23/AIN1) PD7	13
(PCINT0/CLKO/ICP1) PB0	14
28	PC5 (ADC5/SCL/PCINT13)
27	PC4 (ADC4/SDA/PCINT12)
26	PC3 (ADC3/PCINT11)
25	PC2 (ADC2/PCINT10)
24	PC1 (ADC1/PCINT9)
23	PC0 (ADC0/PCINT8)
22	GND
21	AREF
20	AVCC
19	PB5 (SCK/PCINT5)
18	PB4 (MISO/PCINT4)
17	PB3 (MOSI/OC2A/PCINT3)
16	PB2 (SS/OC1B/PCINT2)
15	PB1 (OC1A/PCINT1)



## Meranie otáčok a rýchlosti



## Meranie otáčok a rýchlosti



## Spracovanie vstupného signálu

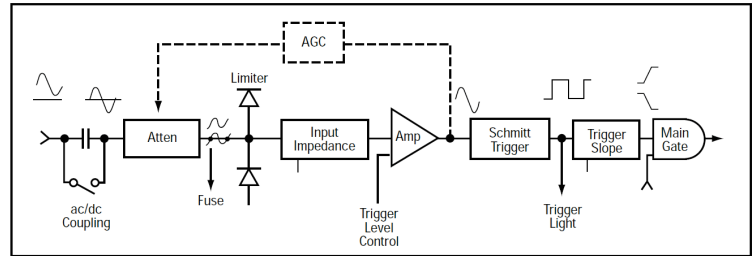
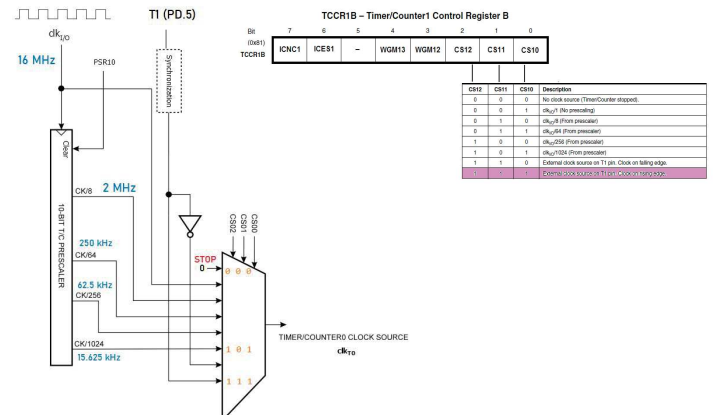
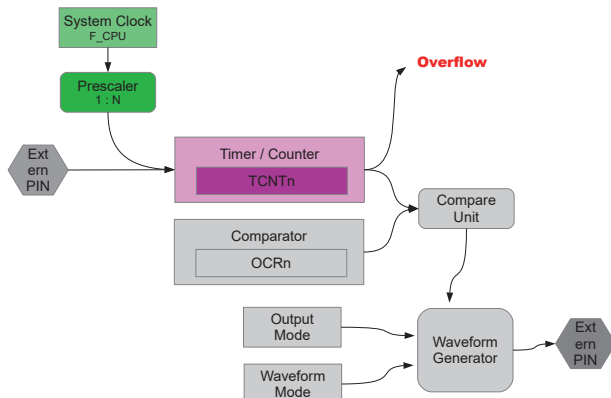
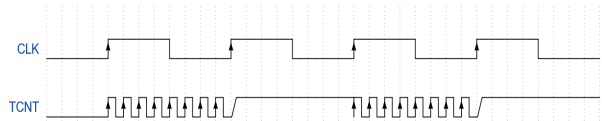
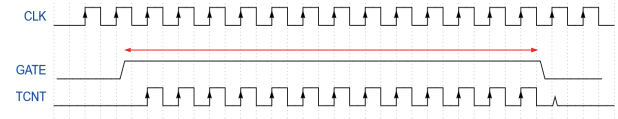
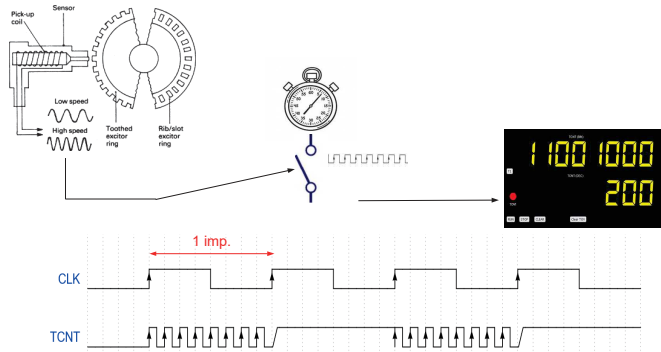


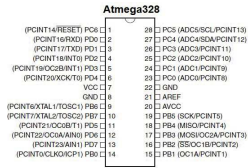
Figure 13. Input Signal Conditioning

## 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 T1 externý vstup:

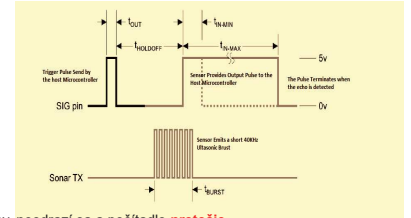
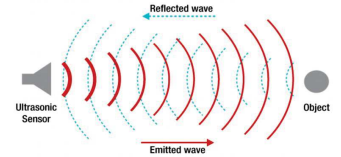
```

DDRD &= ~(1<PDS); // PORTD.5 (PDS) input
PORTD |= (1<PDS); // pull-up ON

TCNT1 = 0x0000; // initialize (CLEAR) counter
TCCR1B = 0b00001111; // T1 clk = external clock source on pin T1, rising edge

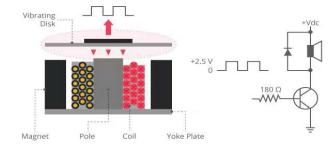
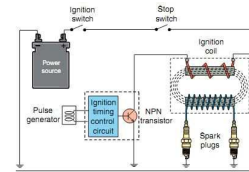
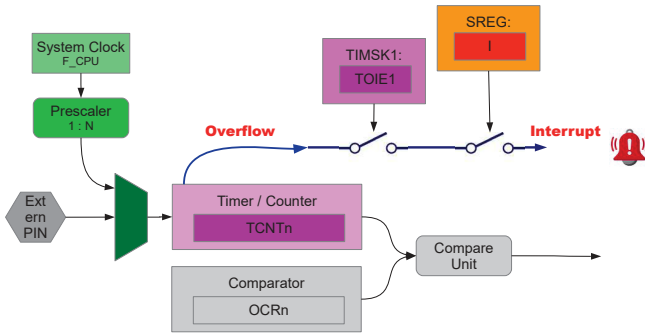
printf("Počítadlo: %04X", TCNT1);
    
```

# Ultrazvukový parkovací senzor



Ak ho namierim do oblchy, neodrazi sa a počítačlo **pretečie**

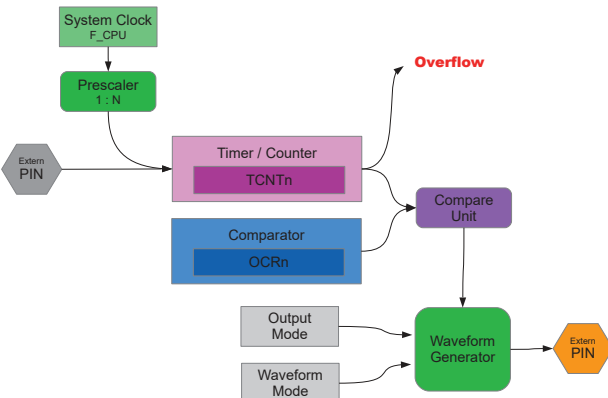
# Generovanie presnej frekvencie



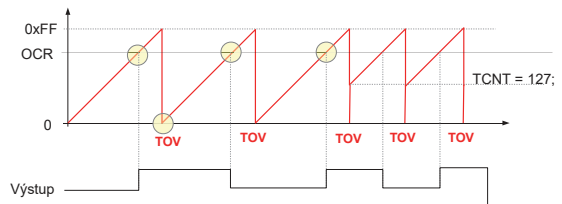
## Softvérový generátor 500Hz

```

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

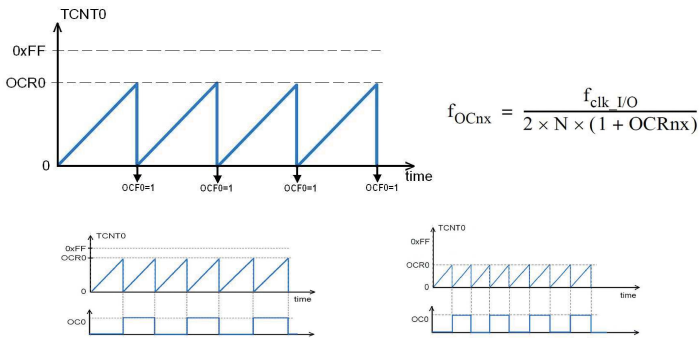


## Mode 0: Normal mode



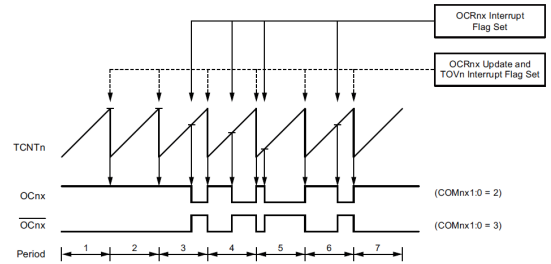
COM01	COM00	Description
0	0	The normal operation, OC0 disconnected
0	1	Toggle OC0 on compare match
1	0	Clear OC0 on compare match
1	1	Set OC0 on compare match

## Mode 1: Clear Timer on Compare Match (CTC)



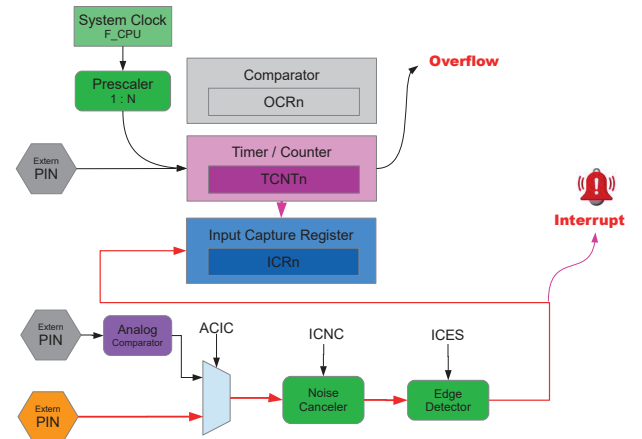
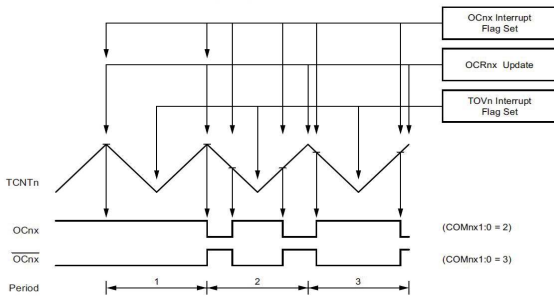
## Mode 2: Fast PWM

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)

$$f_{OCnx} = \frac{f_{clkIO}}{2 \cdot N \cdot (1 + OCRnx)}$$

$f_{clkIO} = 16 \text{ MHz}$

požadovaná  $f_{oc0A} = 440 \text{ Hz}$  (komorné A)

$N = ? / OCR0A = ?$

$N = \{1, 8, 64, 256, 1024\} / OCR0A = \{0, 1, 2, \dots, 255\}$

## Príklad 1:

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

$$f_{OCnx} = \frac{f_{clkIO}}{2 \cdot N \cdot (1 + OCRnx)}$$

$f_{clkIO} = 16 \text{ MHz}$

požadovaná  $f_{oc0A} = 440 \text{ Hz}$  (komorné A)

```

DDR0 |= (1 << PD6); // port D.6 pin ako vystup
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