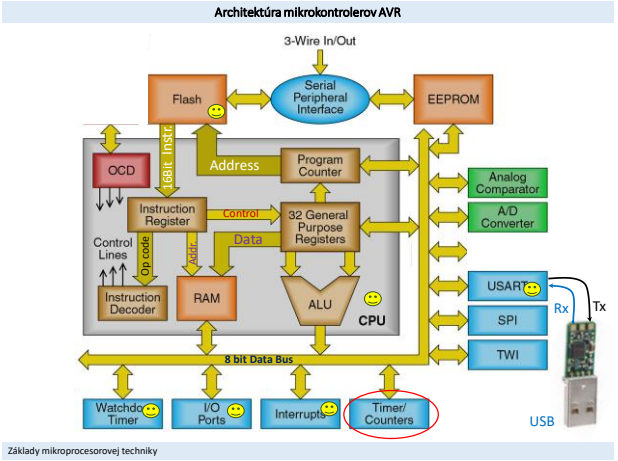


Základy mikroprocesorovej techniky		
Týždeň	Prednášky	Cvičenia
1.	Úvod do predmetu. Číselné systémy v elektronike; kódovane čísel. Jazyk C. Úvod do digitálnej elektroniky.	Instalácia: — IDE Microchip Studio — driver pre Arduino-Kit
2.	Polovodičové pamäte	Práca s Microchip Studio, definovanie projektu
3.	Atmega328, pamäť, generovanie hodinových signálov, manažment napájania, reset	Práca s Microchip Studio preklad, simulácia projektu
4.	Vstupy-výstupy	Práca so vstupmi a výstupmi
5.	Externé a interné prerušenia, Watchdog časovač	Práca s prerušeniami Výber zadania na zápočet
6.	Sériové komunikačné prostriedky-USART—Tx	Práca s-USART—Tx. Práca na zadani
7.	Sériové komunikačné prostriedky-USART—Rx	Práca s-USART—Rx. Práca na zadani
8.	TCO—Normal-mode	Práca s-TCO. Práca na zadani
9.	TCO – Fast PWM mode, Fázovo korektná a CTC PWM	Práca s TCO. Práca na zadani
10.	A/D prevodník	Programovanie A/D. Práca na zadani
11.	EEPROM, FUSE bity	Programovanie EEPROM. Práca na zadani
12.	I2C zbernica	Zápočet - Praktická práca s MCU
13.	Predtermín	Zápočet - Praktická práca s MCU

Základy mikroprocesorovej techniky

1



2

Atmega328P – 8 bitový časovač / počítadlo TCO – Režimy činnosti (19.7)

- Režim činnosti určuje správanie vývodov časovač/počítadlo a vývodov porovnávania výstupu.
- Režim činnosti je definovaný bitmi pre generovanie krivky TCCR0A.WGM0[2:0] a bitmi režimu porovnávania výstupu TCCR0A.COMA[1:0] a TCCR0A.COMB[1:0].
- Bity režimu porovnávania výstupu neovplyvňujú postupnosť počítania, zatiaľ čo bity režimu generovania krivky áno.

- Režimy činnosti TCO:
 - Normal Mode 0
 - Fast PWM Mode 3, 7
 - Phase Correct PWM Mode 1, 5
 - Clear Timer on Compare Match Mode (CTC) 2

Mode	WGM02	WGM01	WGM00	Timer/Counter Mode of Operation	TOP	Update of OCRx at	TOV Flag Set on
0	0	0	0	Normal	0xFF	Immediate	MAX
1	0	0	1	PWM, Phase Correct	0xFF	TOP	BOTTOM
2	0	1	0	CTC	OCRA	Immediate	MAX
3	0	1	1	Fast PWM	0xFF	BOTTOM	MAX
4	1	0	0	Reserved	-	-	-
5	1	0	1	PWM, Phase Correct	OCRA	TOP	BOTTOM
6	1	1	0	Reserved	-	-	-
7	1	1	1	Fast PWM	OCRA	BOTTOM	TOP

Table 19-3. Compare Output Mode, non-PWM

COMB1	COMB0	Description
0	0	Normal port operation, OCA disconnected
0	1	Toggle OCA on Compare Match
1	0	Clear OCA on Compare Match
1	1	Set OCA on Compare Match

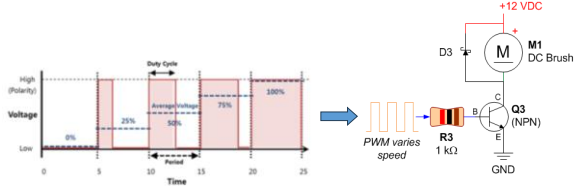
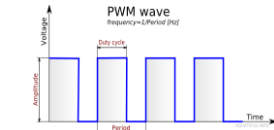
Table 19-4. Compare Output Mode, Fast PWM

COMB1	COMB0	Description
0	0	Normal port operation, OCA disconnected
0	1	WGM02 = 0: Normal Port Operation, OCA Disconnected
0	1	WGM02 = 1: Toggle OCA on Compare Match
1	0	Clear OCA on Compare Match, set OCA at BOTTOM (non-inverting mode)
1	1	Set OCA on Compare Match, clear OCA at BOTTOM (inverting mode)

Základy mikroprocesorovej techniky

3

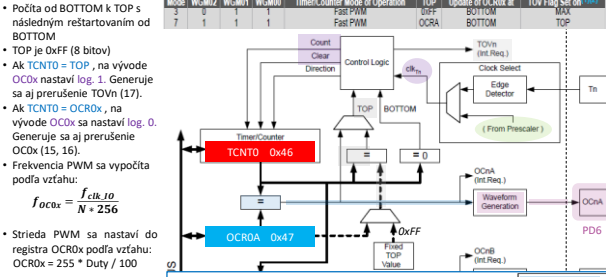
Riadenie výkonu pomocou PWM (napr. otáčky DC motora)



Základy mikroprocesorovej techniky

4

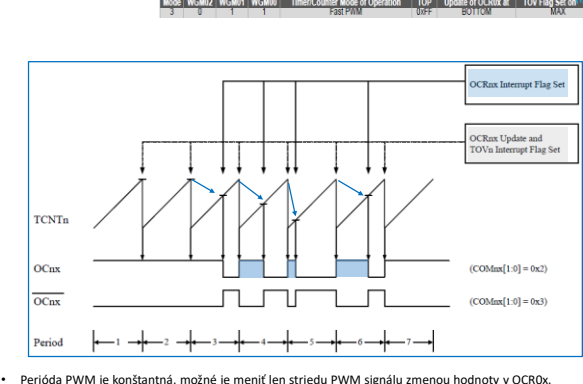
Atmega328P – 8 bitový časovač / počítadlo TCO – Fast PWM mode (19.7.3) (71-89)



Základy mikroprocesorovej techniky

5

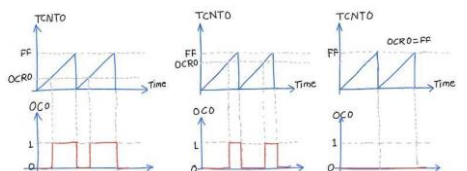
Atmega328P – 8 bitový časovač / počítadlo TCO – Fast PWM mode (19.7.3) (71-89)



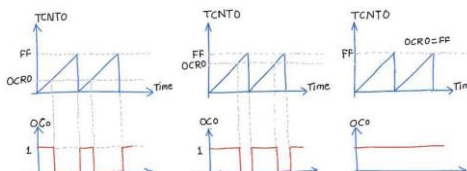
Základy mikroprocesorovej techniky

6

ATmega328P – 8 bitový časovač / počítadlo TC0 – Fast PWM mode



Obr. 63 Jednoduché znázornenie invertujúceho Fast PWM signálu



Obr. 64 Jednoduché znázornenie neinvertujúceho Fast PWM signálu

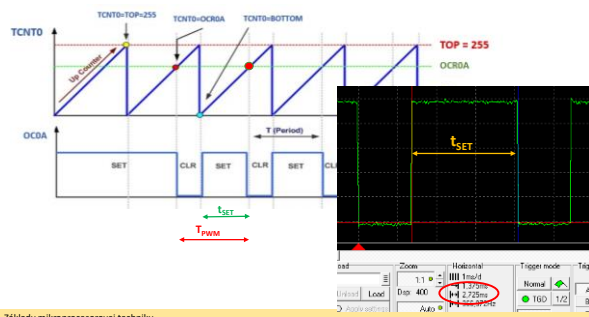
Základy mikroprocesorové techniky

7

TC0c.c ATmega328P – 8 bitový časovač / počítadlo TC0 – Fast PWM mode

Príklad inicializácie mód 3 (Fast PWM), generovanie PWM na OC0A,

```
TCCR0A |= (1<<WGM01)|(1<<WGM00); //Set Fast PWM - mode3
TCCR0A |= (1<<COM0A1); //Set Non Inverting Mode
TCCR0B |= (1<<CS2); //Set prescaler 256 (Tperm = 4,096ms)
TCNT0 = 0;
OCR0A = 170; //tSET = 170 * (4,096ms/256) = 2,72ms
```



Základy mikroprocesorové techniky

8

TC0d.c Príklad TC0 – Fast PWM mode

```

#define BOTTOM 6 //Ass PAM 250 drivnt
#define PERIODA 4 //ms
#define TOP 255 //ms

//Obiliza prerusena TCO

TSM (TIMING_OUT_VST);

PORTIO = !((PORTIO7)); //test osciloskop
ticlear; ticmax=0;
if (ticled < (50*PERIODA)) PORTIO |= (!cPORTS);
if (ticled < (250*PERIODA) & ticled < (250*PERIODA)) PORTIO &= (!cPORTS);
if (ticled < (250*PERIODA) & ticled < (300*PERIODA)) PORTIO |= (!cPORTS);
if (ticled < (250*PERIODA) & ticled < (300*PERIODA)) PORTIO &= (!cPORTS);
if (ticled < (300*PERIODA)) ticled = 0;

if (ticmax > (50*PERIODA)) {
    ticmax = 0;
    if (OCMA < 100) OCMA++;
    else OCMA = BOTTOM;
    if (OCMA < 100) OCMA++;
    else OCMA = BOTTOM;
}

TCNTA = BOTTOM;

int main (void) {
    //inicializacija TIMERA
    //inicializacija I/O
    DDIO |= (1<<OCMA);
    DDIO |= (1<<OCMA);
    TCCR0B |= (1<<CS2);
    TCCR0B |= (1<<CS2);
    TCCR0A |= (1<<ICF1);
    TIMSK0 |= (1<<TOIE1);
    TCNTA = BOTTOM;
    //TCR - Fast PAM
    TCCR0A |= (1<<WGM01)|(1<<WGM00); //Fast fast PAM (TOP=4095)
    TCCR0A |= (1<<COM01); //Fast non-inverting mode OCMA (PAM)
    TCCR0A |= (1<<COM00)|(1<<COM01); //Fast inverting mode OCMA (PDS)
    OCMA = BOTTOM;
    OCMA = BOTTOM;
    sei();

    while(1) {
        _writetest();
    }
}

```

Základy mikroprocesorovej techniky

9

Príklad TC0 – Fast PWM mode

1. Režim LED diódy na výstupe PB5: Svieti, Nesvieti, Bliká (500ms/500ms)
2. Regulovanie jas externej LED diódy 0% až 100% na vývode OC0A alebo OC0B

1. Počítadlo **TC0**, $f_{clk} = 16\text{ MHz}$.
 - Blikanie LED na vývode PB5 bude realizované v obsluhu prerušenia **TOV0** (**TIMER0_OVF_vect**). Prerušenie sa bude generovať každé **2ms**. Čas svietenia a zhasnutia LED diódy bude riadiť softvérové počítadlo **tic_led = 250** (pretože $500\text{ms} / 2\text{ms} = 250$).
 - Ak perióda generovania **TOV0** má byť každé **2ms**, potom $f_{TOV0} = 500\text{Hz}$.
Je potrebné zvoliť vhodný deliaci pomer **N** (1, 8, 64, 256, 1024) a vypočítať hodnotu **TCNT0**, pre generovanie **2 ms** prerušenia (celé kladné číslo).
$$\text{Zvolíme } N=256.$$
$$\text{TCNT0} = 256 - \frac{16\text{ MHz}}{256 \cdot 500\text{ Hz}} = \mathbf{131}$$

(ak $N=64$, potom je $\text{TCNT0} = -244$)
(ak $N=1024$, potom je $\text{TCNT0} = 1,25$)
 - Voľba režimu LED diódy na vývode PB5 sa bude realizovať komunikáciou MCU a PC pomocou UARTu a následných príkazov:

<i>Príkaz</i>	<i>Terminal.exe</i>	<i>Data Visualizer</i>
Zapni	\$02Zapni\$03	\x02Zapni\x03
Vypni	\$02Vypni\$03	\x02Vypni\x03
Blikaj	\$02Blikaj\$03	\x02Blikaj\x03

Základy mikroprocesorovej techniky

10

TC0d.c Príklad TC0 – Fast PWM mode

- Maximálna frekvencia generovaného prerušenia TOV0 je pre $N=256$ a $TCNT0 = 255$,

$$f_{TOVO} = \frac{f_{clkIO}}{256 * (256 - 255)} = 62500\text{Hz} \rightarrow T_{\min TOV} = 0,000016\text{s} = 16\text{us}$$

- V zadani úlohy je generovanie prerušenia každé 2ms → počet impulzov počítadla = $2\text{ms} / 16\mu\text{s} = 125$.
- 8 bitové počítadlo TCNT0 umožňuje napočítať 255 „impulzov“ a nasledujúci „impulz“ vynuluje TCNT0, čo spôsobí generovanie prerušenia TOVF (**žltá značka**). Aby sa generovalo prerušenie, musí dôjsť k pretečeniu počítadla TCNT0.
- Ak bude TC0 počítat 125 impulzov od hodnoty 0, nikdy nedôjde k pretečeniu. Preto je potrebné nastaviť počiatočnú hodnotu počítadla TCNT0.SET = 131 (modulácia 125 + 6).

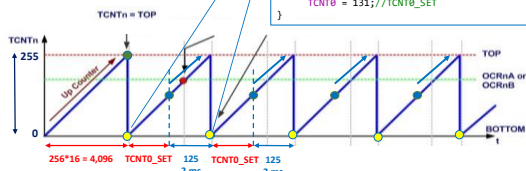
- Toto nastavenie je v obsluhu prerušenia TOV0.

```

{
    if (BLKANIE) {
        if (++ticled == 250) {
            ticled = 0; TOG_LED;
        }
    }
    TCNT0 = 131; //TCNT0_SET
}

```

TCNT0 = TOP



Základy mikroprocesorovej techniky

11

TC0d.c Příklad TC0 – Fast PWM mode

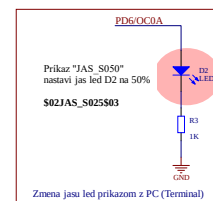
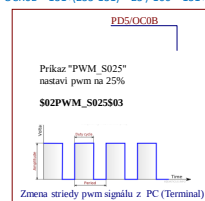
2. Regulovanie jasú externej LED diódy 0% až 100% na vývode OC0A a pwm signálu na vývode OC0B.

- **TCNT0** počíta od **TCNT0 SET** do 256. čo je **125** úrovní počítania, ktoré je možné použiť pre generovanie PWM (N=256).

$$f_{\text{COXPWM}} = \frac{f_{\text{clk}_{\text{IO}}}}{N * (256 - \text{TCNT0})} = \frac{16 \text{ MHz}}{256 * 125} = 500 \text{ Hz}$$

- Zmena striedy sa realizuje zmenou obsahu **OCRO0a** alebo **OCRO8**. **OCROx = TCNT0_SET+(255-TCNT0_SET) * Duty / 100**.
 Duty = 100%: $OCROx = 131 + (255 - 131) * 100 / 100 = 131 + 124 = 255$
 Duty = 20%: $OCROx = 131 + (255 - 131) * 20 / 100 = 131 + 24,8 = 156$
 Duty = 0%: $OCROx = 131 + (255 - 131) * 0 / 100 = 131$
 $\Delta PWM = 100\% / (256 - 131) = 0,8 \%$

- jas = 50% $OCROA = 131 + (255 - 131) \cdot 50 / 100 = 131 + 62 = 193$
- pwm = 25% $OCROB = 131 + (255 - 131) \cdot 25 / 100 = 131 + 31 = 162$



Základy mikroprocesorovej techniky

12

TC0d.c Príklad TCO – Fast PWM mode

jas = 30%, pwm = 30%

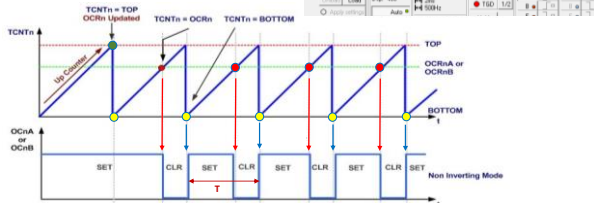
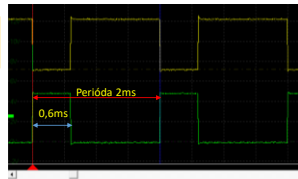
Invertujúci mód

$$TCCR0A = (1 \ll \text{COM0B1}) | (1 \ll \text{COM0B0});$$

$$OCR0B = \text{TCNT0_SET} + (255 - \text{TCNT0_SET}) * \text{pwm} / 100$$

Neinvertujúci mód

$$TCCR0A = (1 \ll \text{COM0A1});$$

$$OCR0A = \text{TCNT0_SET} + (255 - \text{TCNT0_SET}) * \text{pwm} / 100$$


Základy mikroprocesorovej techniky

13

ATmega328P – 8 bitový časovač / počítadlo TCO – Fázovo korektná PWM (19.7.4)

- Môžeme ju označiť ako symetrickú PWM so stredom symetrie v BOTTOM.
- Opakovane počítá od BOTTOM k TOP a potom od TOP k BOTTOM.
- TOP je 0xFF alebo OCR0x.
- Vždy keď $\text{TCNT0} = \text{BOTTOM}$, nastaví sa príznak prerušenia TIFR0.TOV0.
- Ak sa počítadlo inkrementuje a $\text{TCNT0} = \text{OCR0x}$, potom sa OCR0x nastaví na log. 0.
- Ak sa počítadlo dekrementuje a $\text{TCNT0} = \text{OCR0x}$, potom sa OCR0x nastaví na log. 1.
- Frekvencia PWM na vývode OCR0x sa vypočíta podľa vzťahu:

$$f_{\text{OCxPWM}} = \frac{f_{\text{clk_IO}}}{N * 510}$$

$$255 * 2 = 510$$

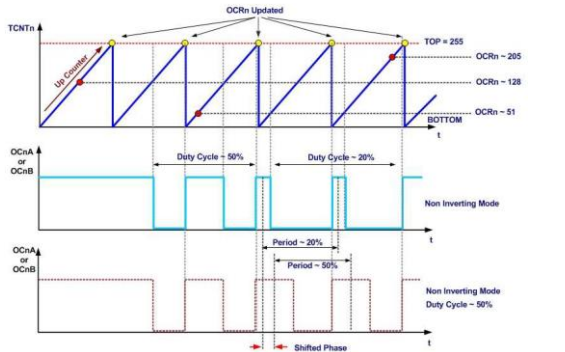
Tento mód je vhodný pre riadenie motorov.

- Ak $\text{OCR0x} = \text{BOTTOM}$, potom OCR0x bude v log. 0.
- Ak $\text{OCR0x} = \text{MAX}$, potom OCR0x bude log. 1.

Základy mikroprocesorovej techniky

14

ATmega328P – 8 bitový časovač / počítadlo TCO – Fázovo korektná PWM

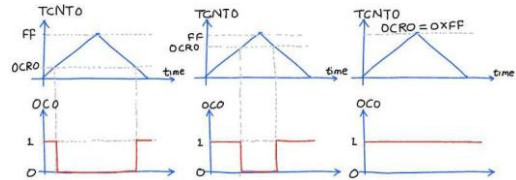


Obr. 67 Princíp vzniku posunutia fázy PWM signálu v režime Fast PWM pri zmene Duty Cycle

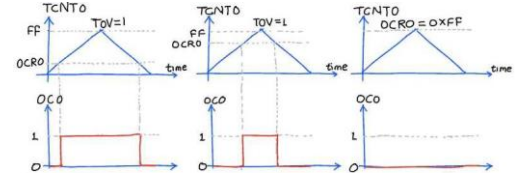
Základy mikroprocesorovej techniky

15

ATmega328P – 8 bitový časovač / počítadlo TCO – Fázovo korektná PWM



Obr. 68 Jednoduché znázornenie neinvertujúceho Phase Correct PWM signálu

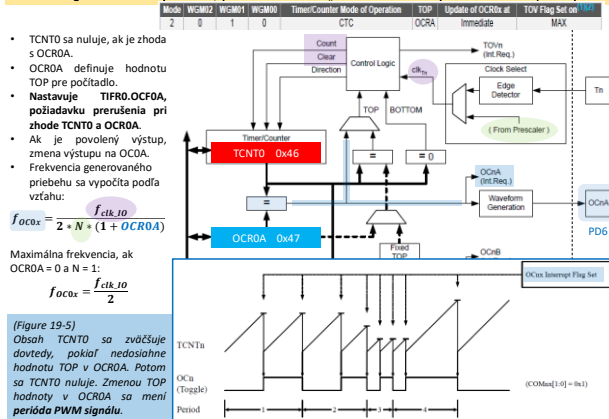


Obr. 69 Jednoduché znázornenie invertujúceho Phase Correct PWM signálu

Základy mikroprocesorovej techniky

16

ATmega328P – 8 bitový časovač / počítadlo TCO – CTC „Clear Timer on Compare match“ mód (19.7.2)

Maximálna frekvencia, ak $\text{OCR0A} = 0$ a $N = 1$:

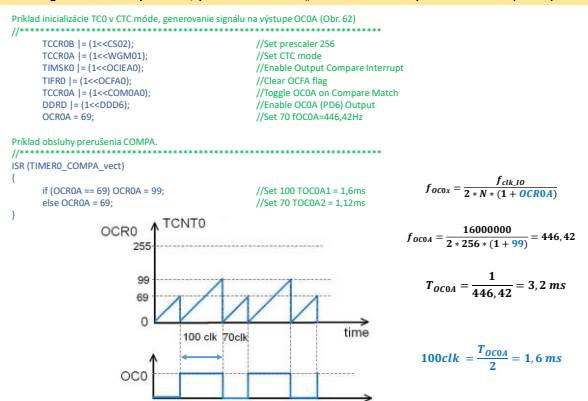
$$f_{\text{OCx}} = \frac{f_{\text{clk_IO}}}{2}$$

(Figure 19-5)
Obsah TCNT0 sa zväčšuje dovtedy, pokiaľ nedosiahne hodnotu TOP v OCR0A. Potom sa TCNT0 nuluje. Zmenou TOP hodnoty v OCR0A sa mení perióda PWM signálu.

Základy mikroprocesorovej techniky

17

ATmega328P – 8 bitový časovač / počítadlo TCO – CTC „Clear Timer on Compare match“ mód (19.7.2)



Obr. 62 Príklad priebehu signálu na vývode OCR0A v CTC móde TCO

Základy mikroprocesorovej techniky

18