

Mikropočítačové Systémy
MIPS
 Distribuované vnorené počítačové systémy
 Distributed Embedded Computer System
 (Microcontrollers)

Cvičenie 11.
Meranie PWM.

Úloha:

- Vytvoríť simulátor PWM signálu snímača teploty: SMT 160-30, resp. SMT 172
 - TC0 bude na pine PD5 generovať PWM signál s $f_{PWM} = 1kHz$ a plnením v intervale DC = (0,1 až 0,9).
 - DC budeme zadávať v %.
- Na meranie parametrov PWM signálu: T1 - čas v log. 1 a T_opak - trvanie jednej periódy PWM signálu použijeme TC1 (16b). TC1 nastavíme do módu „odpamätanie“ stavu počítačidla pri výskyte hrany na pine PB.0. Využijeme ISR().
- Pomocou TC2, bude generovať presnú periódu vzorkovania. Periódu opakovania výpisov, ako aj zmeny plnenia o definovaný prírastok, napr. 1, budeme realizovať s krokom 0,5sek. Využijeme ISR().
- Výpis na LCD, resp. terminál treba realizovať vo formáte, xx.x. (Pevná rádova čiarka). Pri výpočtoch nemôže byť použitá float aritmetika.
- Vo výpisoch sa musí objaviť: Nastavené plnenie. Trvanie T_1, resp. T_opak v SC a us. Ako aj plnenie v % vo formáte: xx.x. Rovnako aj danému plneniu odpovedajúca teplota vo formáte +xx.x °C .
- Ako úloha naviac: Navrhnete filter typu kyžavy priemer z 8-ich vzoriek.

Na cvičení budeme meniť plnenie DC = (0,1 až 0,9), čo odpovedá meranej teplote v rozsahu -46,8 °C až 123,4 °C.

Frekvencia PWM signálu sa môže meniť od 1kHz po 4kHz. My budeme predpokladať len konštantnú hodnotu 1kHz
 ATMEGA 328P má $f_{Osc} = 16\text{ MHz}$.

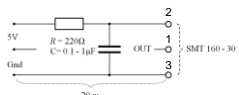
1

2

SMT 160-30 (172) snímač teploty:

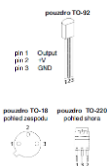
Snímač teploty s PWM výstupom. PWM vo funkcii D/A prevodníka. Je to prevodník teplota na šírko modulovaný signál PWM. Merací rozsah je (-45 °C až 150 °C). Vyrába sa v pázdrach, napr.: T018, T092, T0220.

Spôsob pripojenia:



Základné vlastnosti, parametre:

- Rozsah meranej teploty je -45 až 130°C
- Absolútna presnosť ±0,7°C
- Ochýľka prevodovej charakteristiky od lineárnej je < 0,2°C
- Výstupný signál je kompatibilný s TTL a CMOS logikou
- Spotreba obvodu je menšia ako 1 mW
- Snímač je kalibrovaný vo výrobe
- Výstup PWM signál s frekvenciou opakovania: $f_{op} = 1 + 4 [kHz]$, $T_{op}(pre\ 4kHz) = 250(\mu s)$



3

SMT 160-30, SMT 172 snímač teploty:

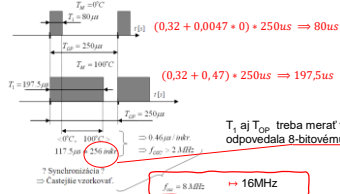
Plnenie ako funkcia meranej teploty:

$$pl = \frac{T}{T_{op}} = 0,32 + 0,0047 \cdot T_w \quad [%; \text{ } ^\circ\text{C}] \quad T_M^{max} = (1 - 0,32)/0,0047 \Rightarrow 144,68^\circ\text{C}$$

Spracovanie informácie:

Treba zmerať aj T₁ aj T_{op} počas jednej periódy opakovania.

Príklad: Pomocou snímača SMT 160 - 30 meráme teplotu v rozsahu T_M = (T₀ až 100 °C), T₀ = (0 až 20 °C).



T_M počítame zo vzťahu:

$$T_M = \frac{T_1}{T_{op}} - 0,32$$

$$T_M = \frac{0,0047}{0,0047} [^\circ\text{C}; -; -]$$

T₁ aj T_{op} treba merať tak, aby presnosť merania odpovedala 8-bitovému prevodníku.

Symbolizácia
 ⇒ Časujte vzorkov. $f_{op} = 8kHz \Rightarrow 16MHz$

4

Informácia je prvotne spracovaná pomocou C/T mikropočítača.

Vlastnosti určité takto: Počítadlo viacej napačíta ak horší prípad odpovedá: T_{op} (1kHz) = 1000 µs. A $f_{osc} = 16MHz \Rightarrow T_{osc} = 0,0625\mu s$

Za čas T_{op} počítačilo napačíta $\frac{1000\mu s}{0,0625\mu s} = 16000\text{ SC}$

⇒ Treba použiť 16 bitové počítačilo.

Čas spracovania: 1 [ms]

Ak nechceme použiť aritmetiku pohyblivej rádovej čiarky, upravíme vzťah do tvaru:

$$T_M * 10 = \frac{T_1 \cdot 10^5 - 32000}{47}$$

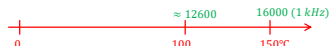
Výsledok predpokladáme v tvare napr.: 995 ± 99,5. Správny výsledok získame, len ak je vzorec naprogramovaný v „správnom“ poradí, a ak zvolíme správne dátové typy.

Rozsahy čísel:

T₁ ∈ (0 + "16000")

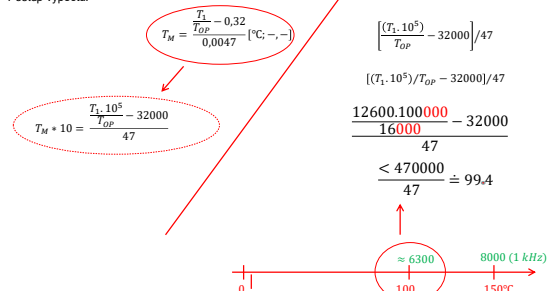
T_{op} ≈ 16000

1kHz



5

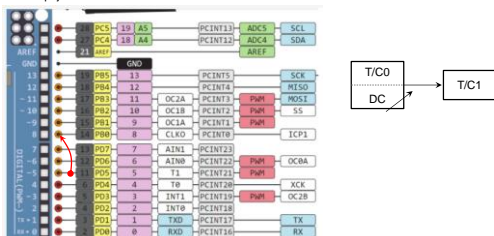
Postup výpočtu:



6

Meranie plnenia PWM signálu:
 Frekvencia opakovania PWM signálu bude konštantná a meníme plnenie.
 Meraná veľičina sa prenáša ako plnenie impulzu -ov.
 Našou úlohou je správne vypočítať veľkosť meranej veľičiny.

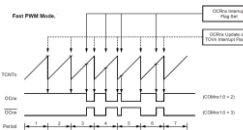
Zapojenie:



Nastavenie TCO:

Namiesto snímača SMT 172 použijeme TCO ako generátor PWM signálu s premenlivým plnením. Port D, pin 5 output. TCO je nastavený v móde 7 (Fast PWM Mode). Plnenie DC_PWM_percent (zadávané v percentách) je premenlivé a nastavuje sa pomocou registra OCR0B. Delička je nastavená na deleno N = 22. Frekvencia PWM signálu je daná hodnotou registra OCR0A.

$$f_{PWM} = \frac{f_{clk_1/0}}{N * (1 + OCR0A)} \quad f_{PWM} = 1000Hz \quad t_j, OCR0A = 10.$$



```
void ini_TCO(void) { // Nastavenie TCO
    set_bit(DDRD, PIN05); // OC0B PWM pin
    // 7 6 5 4 3 2 1 0
    // COM0A[1:0] COM0B[1:0]
    TCCR0A = 0x00; // OC0B PWM mod = 7
    // 7 6 5 4 3 2 1 0
    // WGM02 CS0[2:0]
    TCCR0B = 0x07; // fosc/2
    OCR0B = DC_PWM_percent; //
    // nastavenie frekvencie opakovania na 1ms
    OCR0A = OCR0A_f_opak_TCO;
}
```

Nastavenie TC2:

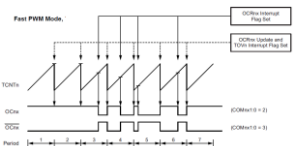
Plnenie PWM signálu budeme meniť s krokom (násobok 2ms). Tento bude generovaný TC2. Opäť použijeme Fast PWM Mode, tak aby sme mohli meniť podľa potreby hodnotu TOP. T.j. hodnotu pri ktorej TC2 pretečie. Povolieme prerušenie od pretečenia TC2. V obsluhu prerušenia budeme generovať čas pre výpis a zmenu plnenia.

Delička je nastavená na deleno N = 22. Frekvencia PWM signálu je daná hodnotou registra OCR0A.

$$f_{PWM} = \frac{f_{clk_1/0}}{N * (1 + OCR0A)} \quad f_{PWM} = 500Hz \quad t_j, OCR0A = 10.$$

```
void ini_TC2(void) {
    // Nastavenie TC2
    // 7 6 5 4 3 2 1 0
    // COM2A[1:0] COM2B[1:0]
    TCCR2A = 0x00; // OC2B PWM mod = 7
    // 7 6 5 4 3 2 1 0
    // WGM02 CS0[2:0]
    TCCR2B = 0x07; // fosc/2

    // nastavenie frekvencie opakovania na 2ms
    OCR2A = OCR2A_f_opak_TC2;
    // Enable interrupts @ overflow TC2 MOD 7
    TIMSK2 = 0x02;
    // sei(); pre všetky naraz
}
```



Obsluha prerušenia od pretečenia TC2:

Plnenie PWM signálu budeme meniť s krokom 0,5sek (násobok 2ms). Tento bude generovaný TC2. Na generovanie dlhších časových úsekov použijeme počítadlo poc_T_vypis, ktoré prednastavím tak, aby sme generovali požadovaný interval.

```
ISR(TIMER2_OVF_vect)
{ // tato slucka sa vykona kazde 2,0ms
    OCR2A = OCR2A_f_opak_TC2;
    //toggle_bit(PORTD, PIN02); //PD2 - change
    poc_T_vypis--;
    if (poc_T_vypis) {
        // prednastavim pocitadlo vypisov na 0,5s
        poc_T_vypis = T_05s;
        flag_vypisov = 1; // nastavim priznak vypisov
        //toggle_bit(PORTD, PIN02); //PD2 - change
    }
}
```

```
int main(void) {
    // Inicializácia I/O, prerušenia
    Konfigurácia UART: Tx, Rx
    ini_uart_0(MYUBRR);
    ini_TCO(); ini_TC1(); ini_TC2();
    sei();

    ...
    printf(Riadok, "Meranie DC impulzu \r\n");
    zob_text_UART(Riadok);
    /* main loop */
    while (1) {
        if (flag_vypisov) {
            Vypis();
            Zmena_plnenia(); // DC = (10 az 90 %)
        }
    }
}
```

PD.2 využijeme pri testovaní programu.

```
void Vypis(void) {
    // Formátovaný výpis
    //printf("Plnenie:????? ", DC_PWM_percent, T_log_1, T_opak );
    Vypis: Plnenie: dd%, ddd [SC], ddd [SC]
    Vypis: Plnenie: dd%, ddd [us], ddd [us]

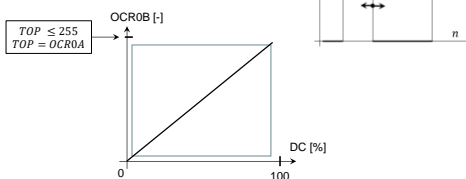
    printf("Plnenie:????? ", DC_PWM_percent, pl_PWM_pocitane_Percent );
    Vypis: Plnenie: dd%, dd, d%

    zob_text_UART(Riadok);
}
```

Plnenie impulzu menime raz za 500ms v rozsahu od 10 do 90%.

Tento krát sme príklad nezostavili tak, aby čítač pretekol po 100 impulzov z preddeličky. To, znamená, že musíme „percentá“ (celé číslo) prepočítavať vzhľadom na TOP hodnotu pretečenia TCO.

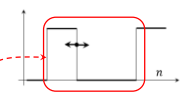
```
void Zmena_plnenia(void) {
    DC_PWM_Percent++;
    if (DC_PWM_Percent > 90) DC_PWM_Percent = 10;
    OCR0B = f(DC_PWM_Percent);
}
```



Plnenie impulzu meníme raz za 500ms v rozsahu od 10 do 90%.

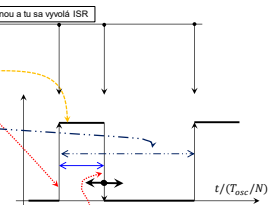
Meranie parametrov PWM signálu realizujeme pomocou TC1 (16b počítačlik)

```
void ini_TC1(void) { // Timer1 Capture
  DDRB &= ~(1<<DDB0); //ICP1 = PORTB.0, input
  PORTB |= (1<<PORTB0); // Pullup Rezistor
  // Nastavenie TC1
  // 7 6 5 4 3 2 1 0
  // WGM12 CSI[2:0]
  // fTC1 = f(fosc) chceme max. presnost merania
  TCCR1B |= 0x0001; // odpamätám nový "čas"
  TCCR1B |= 0x0001; // odchytenie na nabežnú hranu
  TIMSK1 |= 0x0001; // povolenie prerušenia od capture T1
  // sei(); pre všetky naraz
}
```



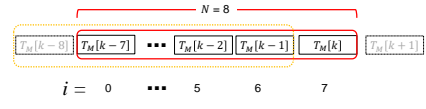
Meranie času trvania log.1, času trvania celého impulzu, ako aj výpočet plnenia v percentách (plnenie *100) sa realizuje v obsluhu prerušenia.

```
ISR(TIMER1_CAPT_vect) {
  // globálne prerušenie je zakázane
  // POČITADLO SA INKREMENTUJE KAŽDU ?us
  if (TESTUJEM HRANU) { // bola nábežná hrana
    // tu prepnem na dobežnú hranu
    T1_log_us = T1_us;
    T1_us = ICR1; // odpamätám nový "čas"
    T_opak = ...;
  } else { // bola dobežná hrana
    // tu prepnem na nábežnú hranu
    T_log_us = ICR1 - T1_us;
    T_log_us_us = f(...);
  }
  // odpamätám nový "čas"
}
```



V KL sa píše o „spravení“ merania teploty. Doporučujú filtrovanie. Navrhnete filter, výpočet priemernej hodnoty z N vzoriek: tzv. „kľzavý priemer“ Zvolíme N = 8.

$$T_{MF} = \frac{\sum_{i=0}^{N-1} T_M[i]}{N} = \frac{SUMA}{N}$$



Postup:

1. Vytvoríme pole $T_M[8]$ a vynulujeme.
2. Vytvoríme smerník $i = 0$ a pomocnú premennú $SUMA = 0$
3. Po odmeraní novej hodnoty T_M vykonáme:
 - * $SUMA = SUMA - T_M[i] + T_M;$
 - * $T_M[i] = T_M;$
 - * $i++;$ $i\&= 0x07;$
4. MÔŽEME vypočítať :
 - $T_{MF} = \frac{SUMA}{8} = SUMA \gg 3;$

DATASHEET SMT112

for voltage measurement 22, 2016
reference
page 4/10