

Osnove mikroprocesorske elektronike

Vaja 8: Časovniki

Naloge:

- S pomočjo podatkovnega lista procesorja (datasheet) ugotovite, kako je nastavljen časovnik 0 v programu na naslednji strani. Izpišite binarne vrednosti obeh konfiguracijskih registrov in za spodaj navedene bite ali skupine bitov, napišite nastavljene vrednosti in z nekaj besedami pomen nastavitve. V graf vrišite vrednost števca v odvisnosti od časa. Pod ta graf narišite še graf stanja zastavice TOV0 iz registra TIFR0.

TCCR0A

--	--	--	--	--	--	--	--

TCCR0B

--	--	--	--	--	--	--	--

Waveform Generation Mode: _____

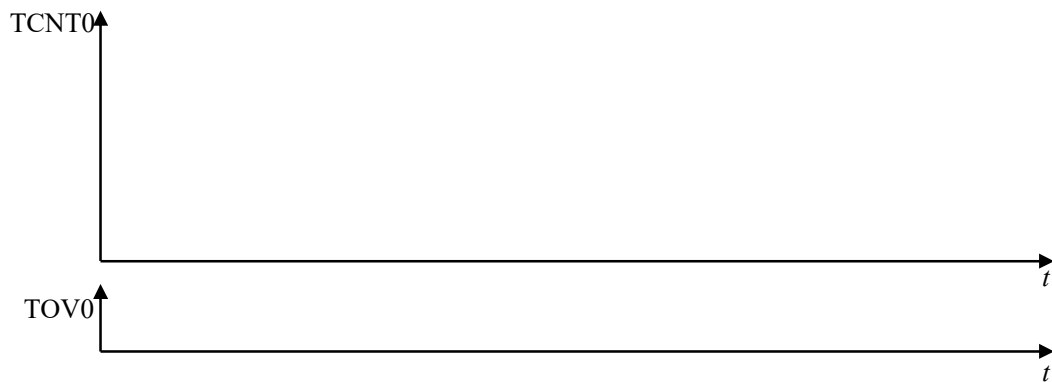
Compare Match Output A Mode: _____

Compare Match Output B Mode: _____

Force Output Compare A: _____

Force Output Compare B: _____

Clock select: _____



- Ugotovite, kaj počne program z naslednje strani in razložite njegovo delovanje.

- Nastavitve časovnika 0 spremenite tako, da bo začel šteti od začetka vsakih 18432 ciklov sistemske ure. Koliko časa je to? _____
- Program dopolnite tako, da bo časovnik prožil prekinitev. Prekinitev naj vsakič poveča 32-bitno globalno spremenljivko z imenom "systick" za ena. Ker sedaj prekinitev pobriše zastavico časovnika, je ne moremo več brati v glavnem programu. Namesto tega pripravite 8-bitno globalno spremenljivko "TIME_STATUS", v kateri naj 1. bit predstavlja zastavico, ki pomeni, da je minila 1 ms. Prekinitev naj to zastavico vsakič postavi na 1. Napišite funkcijo `char HasOneMillisecondPassed()`, ki to zastavico preveri in vrne "0", če je zastavica pobrisana in "1", če je zastavica postavljena. Če je zastavica postavljena, jo mora funkcija spet zbrisati. Glavni program popravite tako, da bo namesto časovnikove zastavice uporabljal novo funkcijo. (Opozorilo: Premislite o "volatile", "ATOMIC_BLOCK", ...)

- Časovnik 2 nastavite tako, da bo na izhodu PD7 pravokotni signal z impulznim razmerjem 25 %. Preddelilnik nastavite na 1:1.

Domača naloga

Napišite program, ki vsakih 100 ms spremeni impulzno razmerje pravokotnega signala na PD7. Impulzno razmerje naj se spreminja od 0 do 100 % (od 0 do 255) po sinusni krivulji s periodo 4 s.

```
#include <avr/io.h>
#include "kbd.h"

void Init_IO();
void Init_T0();

int main(void)
{
    int cnt1=0, cnt2=0;
    char cmd;

    Init_IO();
    Init_T0();
    while(1)
    {
        if (TIFR0 & (1<<TOV0))
        {
            TIFR0 = (1<<TOV0);
            ReadKBD();
            if (cnt1 == 100)
            {
                cnt1 = 0;
                PORTB ^=0x01;
            }
            cnt1++;
            if (cnt2 == 1000)
            {
                cnt2 = 0;
                cmd = GetKey();
                if (cmd == 1) PORTB += 0x10;
            }
            cnt2++;
        }
    }
}

void Init_IO()
{
    //init IO ports
    PORTA=0x08;
    PORTB=0x01;
    PORTC=0x3E;    //0011 1110
    PORTD=0x7D;    //0111 1101
    //1-output, 0-input
    DDRA=0xF0;
    DDRB=0xFF;
    DDRC=0xC1;
    DDRD=0x82;
}

void Init_T0()
{
    //Init timer0
    TCCR0A=0;
    TCCR0B=4;
}
```