Sisteme cu MicroProcesoare

Cursul 6

Module de tip timer

Conf. Gigel Măceșanu

Universitatea
Transilvania
din Brașov
FACULTATEA DE INGINERIE ELECTRICĂ
ȘI ȘTIINȚA CALCULATOARELOR

Cuprins

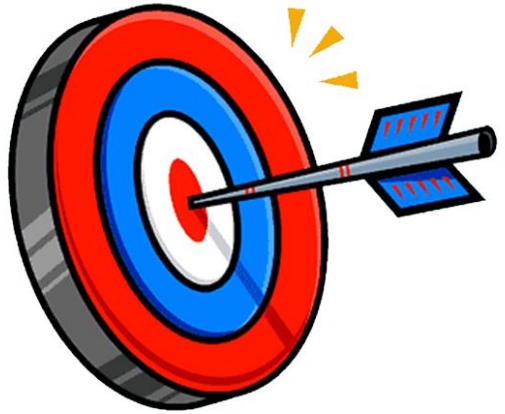
Universitatea
Transilvania
din Brașov
FACULTATEA DE INGINERIE ELECTRICĂ
ȘI ȘTIINȚA CALCULATOARELOR

- Prezentare generală
- Clasificarea modulelor de tip timer
- Modalități de accesare a datelor
- Circuit de prescalare
- Generarea de semnale cu factor de umplere variabil
- Capturi de evenimente externe

Obiectivul cursului

Universitatea Transilvania din Brașov
FACULTATEA DE INGINERIE ELECTRICĂ ȘI ȘTIINȚA CALCULATOARELOR

- Înțelegerea funcționării și utilizării modulelor de tip timer dintr-un microcontroler:
  - Modurile cum acesta poate fi configurat
  - Metodele de citire a datelor din acestea
  - Modalitățile de generare a semnalelor PWM și utilitatea acestora
  - Mecanismul prin care se pot asocia momente de timp specifice unor evenimente externe



Prezentare generală

Universitatea Transilvania din Brașov
FACULTATEA DE INGINERIE ELECTRICĂ ȘI ȘTIINȚA CALCULATOARELOR

- Timer vs Counter
  - Un timer monitorizează semnale care au o distribuție în timp constantă și uniformă (semnale de tact)
  - Un counter monitorizează semnale care au o distribuție aleatoare în timp (semnale de la dispozitive externe)
- Un timer poate fi utilizat pentru:
  - **Cronometrare**: timer-ele pot fi configurate pentru a număra într-un anumit interval de timp, ceea ce permite microcontrolerului să măsoare perioade specifice
  - **Generarea semnalelor de tact** (clock): modulele de timp pot genera semnale periodice (cu formă specifică), care sunt utilizate pentru a sincroniza alte componente ale sistemului (de ex. PWM)
  - **Generarea de întreruperi**: timer-ele pot fi configurate pentru a genera întreruperi la intervale de timp regulate
  - **Măsurarea frecvenței**: utilizat pentru a măsura frecvența unui semnal extern, ceea ce este important în aplicațiile unde frecvența trebuie monitorizată sau controlată
  - **Implementarea funcțiilor de temporizare sau contorizare**: utile la implementarea de funcții simple precum funcții de întârziere (delays) sau numărarea evenimentelor (de exemplu, impulsuri de la un senzor)

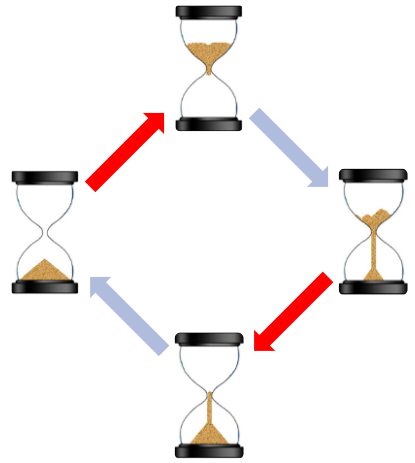
Prezentare generală

Universitatea Transilvania din Brașov
FACULTATEA DE INGINERIE ELECTRICĂ ȘI ȘTIINȚA CALCULATOARELOR

- Importanța modulelor de tip timer:
  - **Menținerea eficienței procesorului:** permit microcontrolerului să realizeze sarcini legate de timp fără a încărca procesorul. Astfel, CPU-ul poate continua să execute alte sarcini în timp ce modulele de timp operează în fundal
  - **Precizia controlului:** timer-ele oferă o precizie înaltă în măsurarea și generarea intervalelor de timp, ceea ce este esențial pentru aplicații care necesită sincronizare exactă. De exemplu: comunicațiile de date, controlul motoarelor sau generarea de semnale audio
  - **Flexibilitate:** prin configurarea modulelor de timp, microcontrolerele pot fi adaptate pentru o gamă largă de aplicații, de la controlul iluminării până la sisteme complexe de control industrial

Clasificare

- În funcție de caracteristici, funcționalități sau utilizare putem avea:
  - **Timer simplu**:
    - Numără un interval de timp bazat pe un ceas intern al microcontrolerului
    - Își incrementează valoarea (un registru) la fiecare ciclu de ceas (generat de la oscilator intern sau ext.) pe un anumit front (crescător sau descrescător)
    - Când este atins maximul se resetează și poate genera întreruperi, pentru a cronometra durate fixe
    - Numărarea poate sa fie în ambele sensuri: up sau down
    - Moduri de operare:
      - **Normal**: numără până la un număr maxim și se resetează
      - **CTC** (Clear Timer on Compare): pornește de la 0 și numără până când se atinge un prag, apoi se resetează



Clasificare

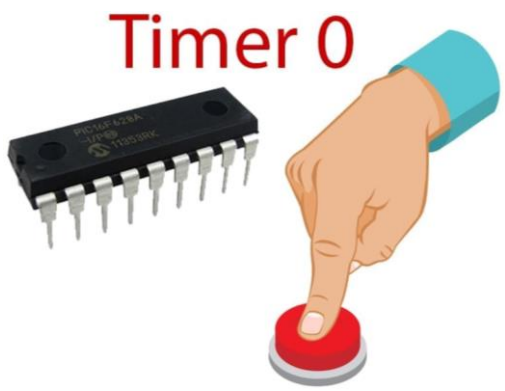
- Pentru o rezoluție de n biți intervalul de incrementare este:

$$[\mathbf{0}, \mathbf{2^n - 1}]$$

- În general timerele sunt de 8 sau 16 biți, astfel avem un interval de incrementare:

$$[\mathbf{0}, 255] \text{ sau } [\mathbf{0}, 65535]$$

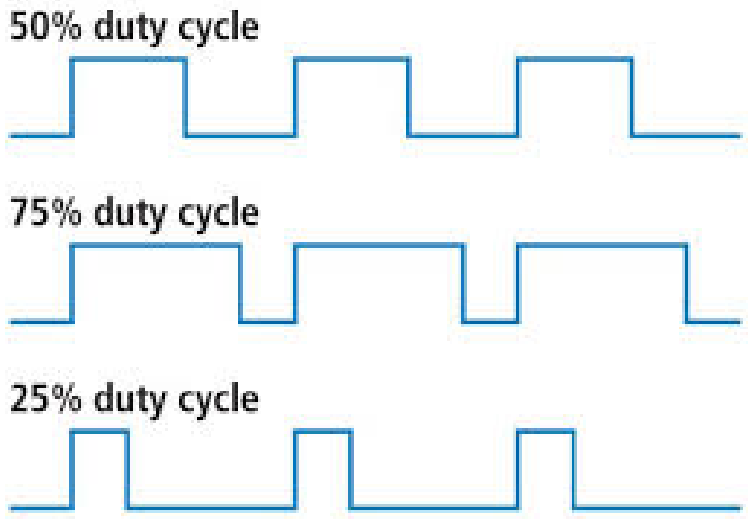
- **Counter** (numărător):
  - este un modul de timp specializat care nu numără pe baza unui ceas intern, ci pe baza unor impulsuri externe
  - când un eveniment extern (de exemplu, un impuls de la un senzor) are loc, counter-ul își incrementează valoarea
  - counter-ele pot fi configurate să numere fie în sus (incrementare), fie în jos (decrementare).



Clasificare

- **Timer PWM** (Pulse Width Modulation)
  - se generează un semnal digital cu o frecvență constantă, dar cu un factor de umplere (eng. duty cycle) variabil
  - Factorul de umplere reprezintă procentul dintr-o perioadă completă în care semnalul este activ

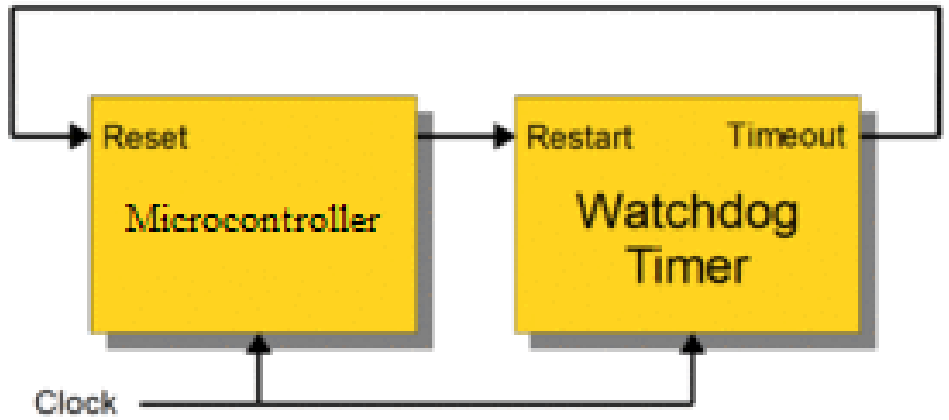
  - Se utilizeaza la:
    - Controlul motoarelor electrice
    - Generarea de sunete (muzică electronică)
    - Controlul sistemelor de iluminare
    - Module de comunicație
    - Module de conversie DAC



Clasificare

Universitatea Transilvania din Brașov
FACULTATEA DE INGINERIE ELECTRICĂ ȘI ȘTIINȚA CALCULATOARELOR

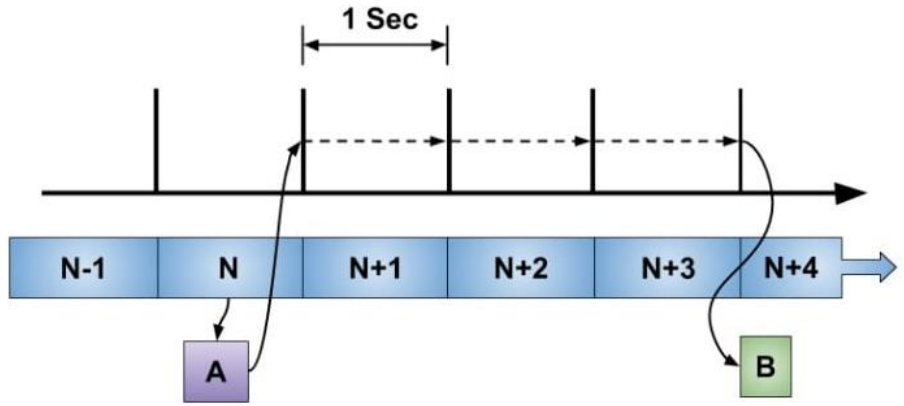
- **Watchdog** timer: responsabil pentru monitorizarea funcționării corecte a microcontrolerului
  - **Watchdog Timer**-ul *este un timer specializat care are rolul de a reseta microcontrolerul în cazul în care programul se blochează sau nu funcționează corect*
  - Dacă timer-ul watchdog nu este resetat de către programul principal înainte de expirarea sa, acesta va genera un semnal de reset care va reporni microcontrolerul.
  - Watchdog Timer-ul poate fi configurat pentru a ajusta perioada de timp înainte de resetare
  - Perioada poate varia de la câteva milisecunde până la câteva secunde, în funcție de nevoile aplicației
  - În aplicațiile embedded critice, Watchdog Timer-ul este esențial pentru a detecta blocaje ale sistemului și a corecta astfel de situații prin repornirea sistemului



Clasificare

Universitatea Transilvania din Brașov
FACULTATEA DE INGINERIE ELECTRICĂ ȘI ȘTIINȚA CALCULATOARELOR

- **Real-Time Clock** (RTC): *este un modul de timp specializat, integrat în microcontrolere sau disponibil ca un cip extern, care menține și oferă informații precise despre ora și data curente*
  - RTC-ul este conceput pentru a păstra timpul pe termen lung, chiar și atunci când microcontrolerul este în modul de repaus sau oprit
  - RTC-ul poate fi configurat pentru a genera alarme la ore sau date prestabilite, sau pentru a declanșa întreruperi periodice (de exemplu, la fiecare secundă, minut sau oră)
  - Un RTC nu se resetează
  - De obicei se incrementează de la o sursă internă de ceas, cu o frecvență de 32,768 Hz (multiplu de 2^15 → permite divizarea ușoară în secunde)
  - Se pot utiliza surse externe de alimentare, pentru a păstra funcționarea
  - Conține registrul de timp, alarme specifice și alimentare (poate chiar externă)



Accesarea datelor din timer

Universitatea Transilvania din Brașov
FACULTATEA DE INGINERIE ELECTRICĂ ȘI ȘTIINȚA CALCULATOARELOR

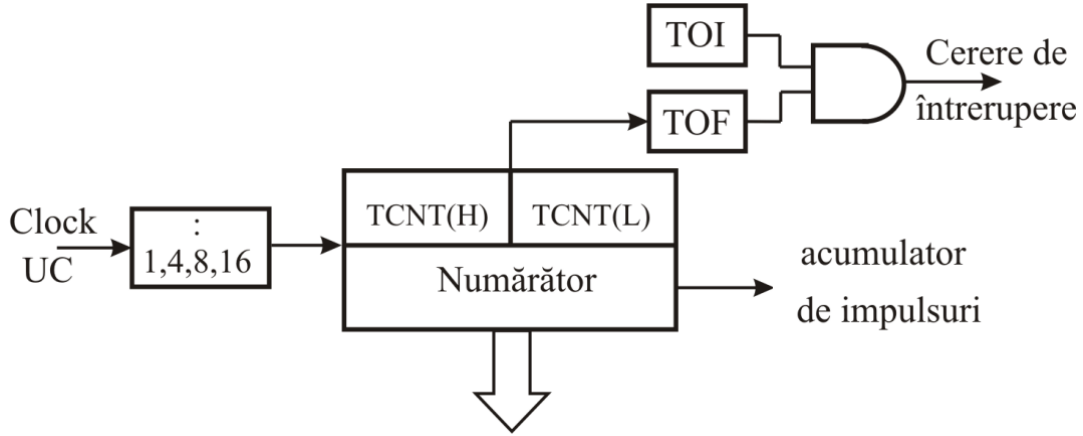
Citire directă:

- Este metoda cea mai simplă și directă de accesare a datele dintr-un timer
- Se citește direct valoarea curentă a contorului (registrul de contor), prin intermediul codului

```
uint16_t valoare_curenta = TCNT1;
```

Folosirea întreruperilor:

- Un flag de tip TOI trebuie să fie activat iar un flag TOF (timer overflow interrupt) este setat
- Întreruperea este validă dacă sistemul de întreruperi este setat corespunzător tipului de întrerupere



Accesarea datelor din timer

Universitatea Transilvania din Brașov
FACULTATEA DE INGINERIE ELECTRICĂ ȘI ȘTIINȚA CALCULATOARELOR

- Tipuri de întreruperi asociate:
  - **Overflow Interrupt** (întrerupere la depășire): contorul unui timer atinge valoarea maximă și revine la zero. Este utilă pentru a genera evenimente periodice, cum ar fi o întrerupere la fiecare x microsecunde
  - **Compare Match Interrupt** (întrerupere la comparare): când valoarea contorului timer-ului este egală cu o valoare prestabilită (stocată într-un registru de comparare). Aceasta permite generarea de evenimente precise în funcție de valoarea dorită
  - **Input Capture Interrupt** (întrerupere la captura de intrare): un semnal extern schimbă starea unui pin specific, iar valoarea curentă a contorului este salvată într-un registru de captură. Această întrerupere este esențială pentru măsurarea precisă a duratei pulsului sau a frecvenței semnalelor

```
ISR(TIMER1_COMPA_vect) {
// Cod care va fi executat la întrerupere }
```

Accesarea datelor din timer

Universitatea Transilvania din Brașov
FACULTATEA DE INGINERIE ELECTRICĂ ȘI ȘTIINȚA CALCULATOARELOR

- **Interogarea** (eng. polling):
  - Presupune citirea anumitor registre și așteptarea în bucle software a unor valori (flags-uri specifice din registrele de status ale timer-elor)
  - Avantaj: Acuratețea timer-ului este utilizată la maxim
  - Dezavantaj: sunt pierdute cicluri mașină pentru interogare

```
// interogare pentru o valoare într-o locație de memorie
while ( locație_interogare_curentă != VALOARE_DORITĂ )
  {}; // așteaptă ca locația să conțină o valoare specifică

// sau, interogareore pentru ca un bit/biți să devină true
while ( !(locație_interogare_curentă & Mască_biți) )
  {}; // aștept ca bit/biți din Mască_biți sa devină true
```

Aplicație – Utilizare întrerupere

Universitatea Transilvania din Brașov
FACULTATEA DE INGINERIE ELECTRICĂ ȘI ȘTIINȚA CALCULATOARELOR

- Exemplu de utilizare a unei întreruperi pentru timer
  - Cerință: să se implementeze un program care generează întreruperi pentru overflow. Se utilizează un timer de 8 biți de la ATmega8, cu un oscilator de 8MHz.

```
// Timer/Counter 0 initialization
// Clock source: System Clock
// Clock value: 8000,000 kHz
TCCR0=(0<<CS02) | (0<<CS01) | (1<<CS00);

// Timer(s)/Counter(s) Interrupt(s) initialization
TIMSK=(0<<OCIE2) | (0<<TOIE2) | (0<<TICIE1) |
(0<<OCIE1A) | (0<<OCIE1B) | (0<<TOIE1) | (1<<TOIE0);
```

Legendă:

TCCR0 – gestionează comportamentul și modul de operare al timerului

CS0x - controlează sursa de ceas și prescalerul pentru timer

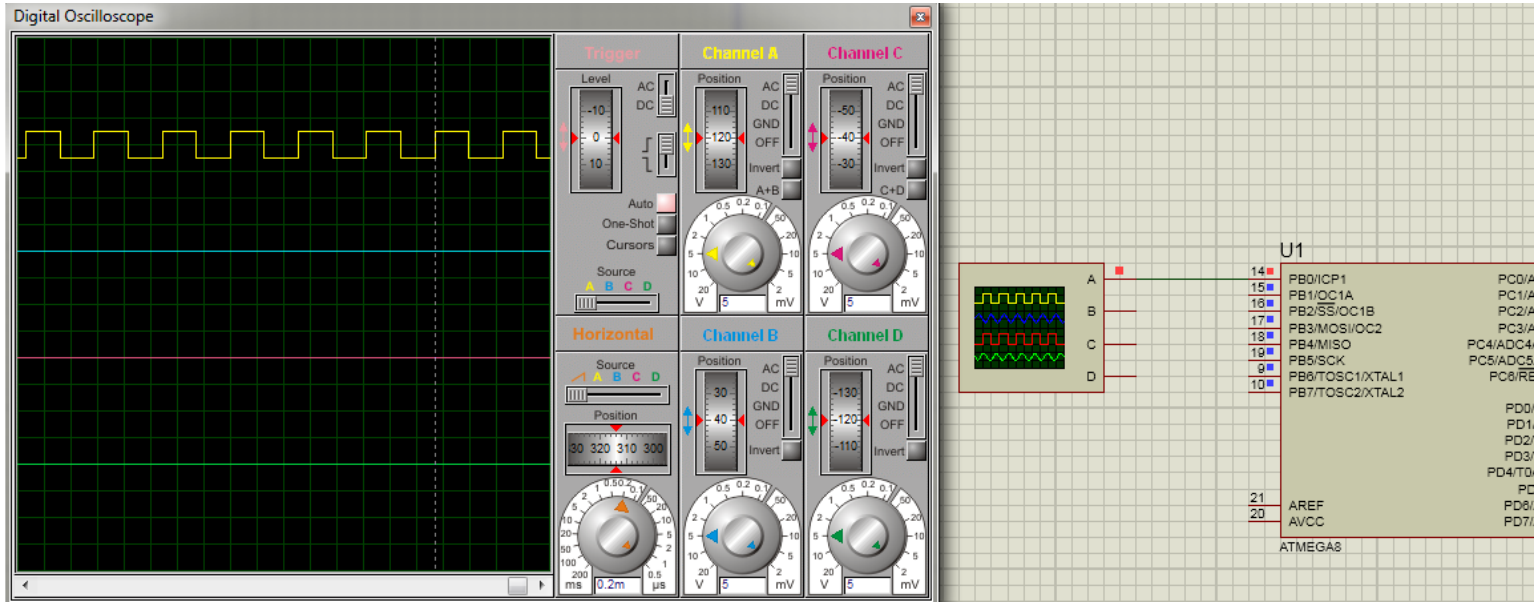
TIMSK – controlează activarea sau dezactivarea întreruperilor asociate cu timer-ele

TOIE0 - controlează activarea întreruperii de overflow de pe Timer0

Aplicație – Utilizare întrerupere

Universitatea Transilvania din Brașov
FACULTATEA DE INGINERIE ELECTRICĂ ȘI ȘTIINȚA CALCULATOARELOR

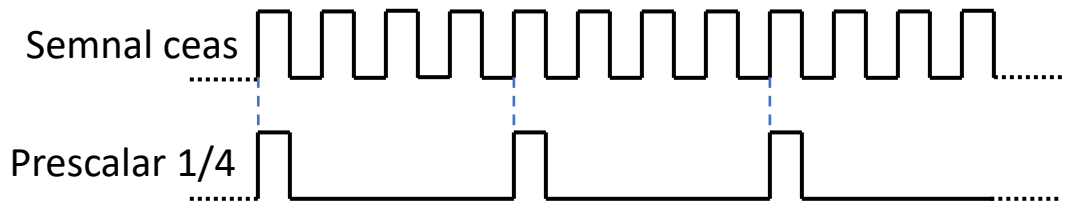
```
// Activare întreruperi globale
#asm("sei") //sau sei() pentru compilatoare AVR

// Timer 0 overflow interrupt service routine
interrupt [TIM0_OVF] void timer0_ovf_isr(void)
{ PORTB.0 = ~PORTB.0;//vizualizare rezultat }
```

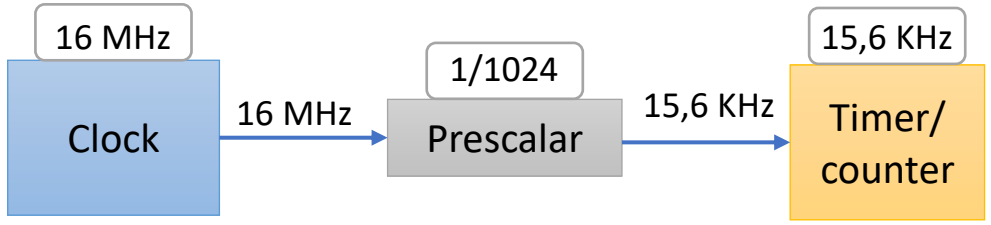


Circuit de prescalare

Universitatea Transilvania din Brașov
FACULTATEA DE INGINERIE ELECTRICĂ ȘI ȘTIINȚA CALCULATOARELOR

- Principalele caracteristici/funcții ale acestuia:
  - Preia semnalul de ceas și îl împarte la anumite valori, presetabile (din registrul de prescalare corespunzător)
  - Scopul principal al prescaler-ului este de a permite timerelor să funcționeze la o frecvență mai mică decât frecvența de ceas a microcontrolerului
  - Are valori de forma 2^P, unde P este nr. biți de prescalare
  - Nu se poate folosi orice număr pentru prescalare, ci se verifică foaia de catalog
  - Se recomandă utilizarea celei mai mici valori de prescalare





Aplicație – Utilizare prescalar

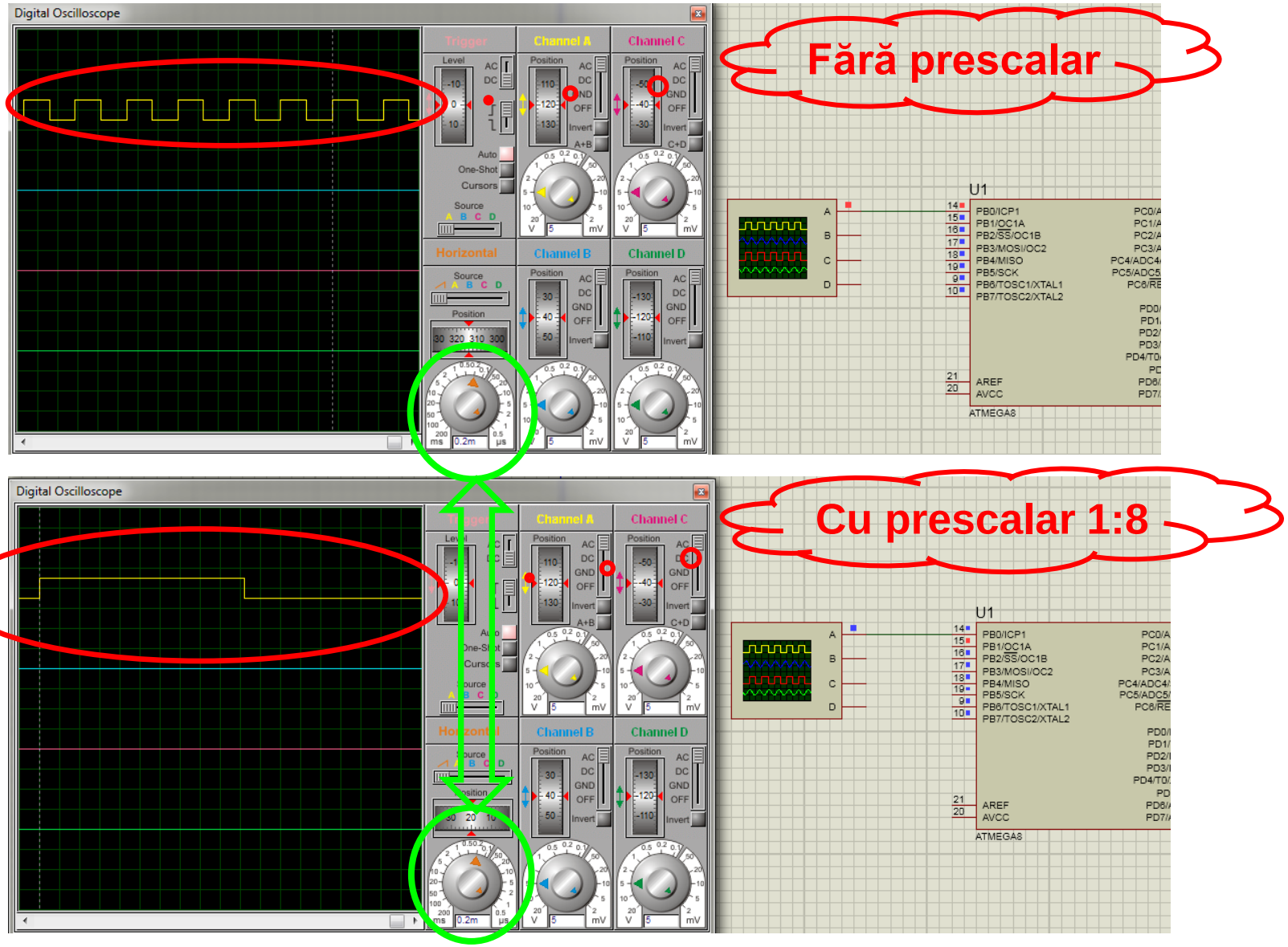
Universitatea Transilvania din Brașov
FACULTATEA DE INGINERIE ELECTRICĂ ȘI ȘTIINȚA CALCULATOARELOR

- Exemplu de utilizare a unei timer cu setare de prescalar
  - Cerință: să se implementeze un program care generează întreruperi pentru overflow. Se utilizează un timer de 8 biți de la ATmega8, cu un oscilator de 8MHz și prescalar 1/8.

```
// Timer/Counter 0 initialization
// Clock source: System Clock
// Clock value: 8000,000 kHz
TCCR0=(0<<CS02) | (1<<CS01) | (0<<CS00); //prescalar 1:8
TCNT0=0x00;

// Timer(s)/Counter(s) Interrupt(s) initialization
TIMSK=(0<<OCIE2) | (0<<TOIE2) | (0<<TICIE1) | (0<<OCIE1A) |
(0<<OCIE1B) | (0<<TOIE1) | (1<<TOIE0);
```

Aplicație – Utilizare prescalar

Universitatea Transilvania din Brașov
FACULTATEA DE INGINERIE ELECTRICĂ ȘI ȘTIINȚA CALCULATOARELOR



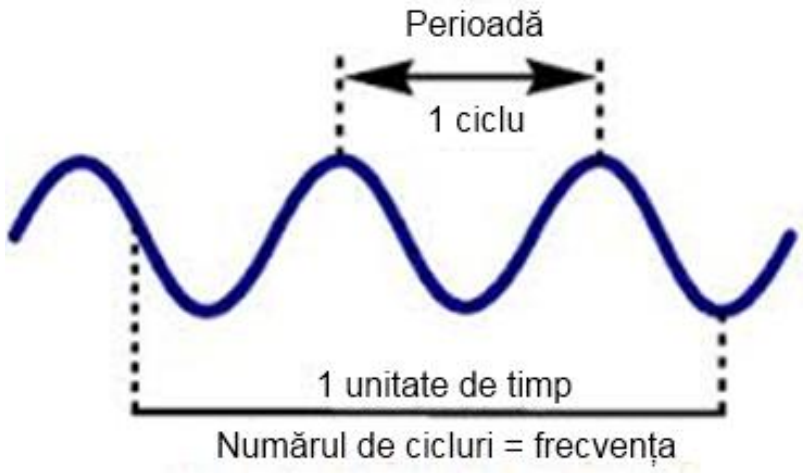
Caracteristicile ale semnalelor periodice

Universitatea Transilvania din Brașov
FACULTATEA DE INGINERIE ELECTRICĂ ȘI ȘTIINȚA CALCULATOARELOR

- Terminologie legată de module timer:
  - **Frecvență**:
    - Un semnal $x(t)$ este periodic, cu perioada T dacă:

$$x(t)=x(t+T)$$

    - Frecvența reprezintă numărul de apariții ale unui eveniment în cadrul unei unități de timp (Hertz sau cicluri/sec)
  - **Perioada** este inversul frecvenței
    - $f=1/T$
    - Frec = 1Hz T = 1s

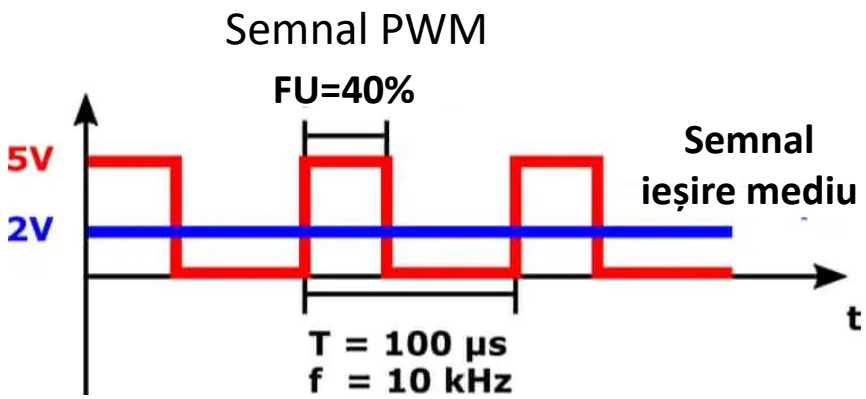


Generare semnal cu factor de umplere variabil (PWM)

Universitatea Transilvania din Brașov
FACULTATEA DE INGINERIE ELECTRICĂ ȘI ȘTIINȚA CALCULATOARELOR

- Timer-ul poate fi utilizat pentru generarea unui semnal digital cu factor de umplere și perioadă ajustabile
- Parametrii ai semnalului PWM:
  - **Perioada** (T): reprezintă durata totală a unui ciclu complet al semnalului PWM. Aceasta include timpul în care semnalul este „on” (activ) și timpul în care este „off” (inactiv)
  - **Frecvența** (f): indică de câte ori pe secundă se repetă un ciclu complet al semnalului PWM
  - **Amplitudine**: reprezintă tensiunea maximă în starea activă („on”) a semnalului PWM
  - **Factorul de Umplere** (FU): proporția din perioada totală în care semnalul PWM este „on”. Este exprimat ca procentaj și determină puterea medie livrată la sarcină. Se calculează:

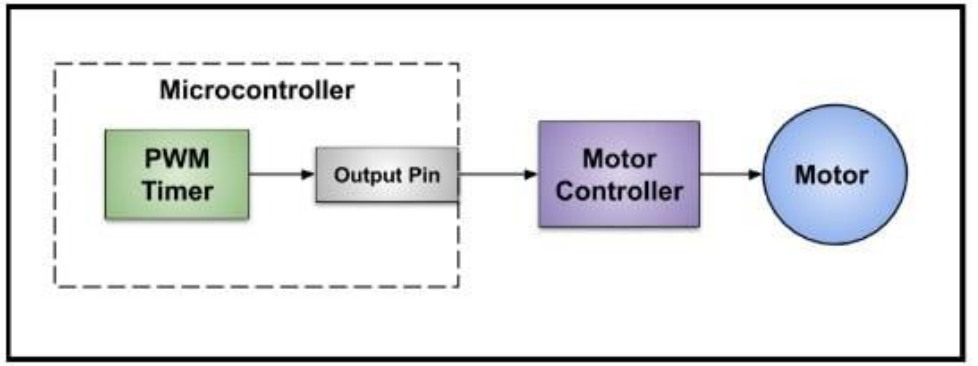
    $FU = \frac{T_{on}}{T} \times 100\%$, unde T_{on} este timpul în care semnalul este activ („on”) și T este perioada totală

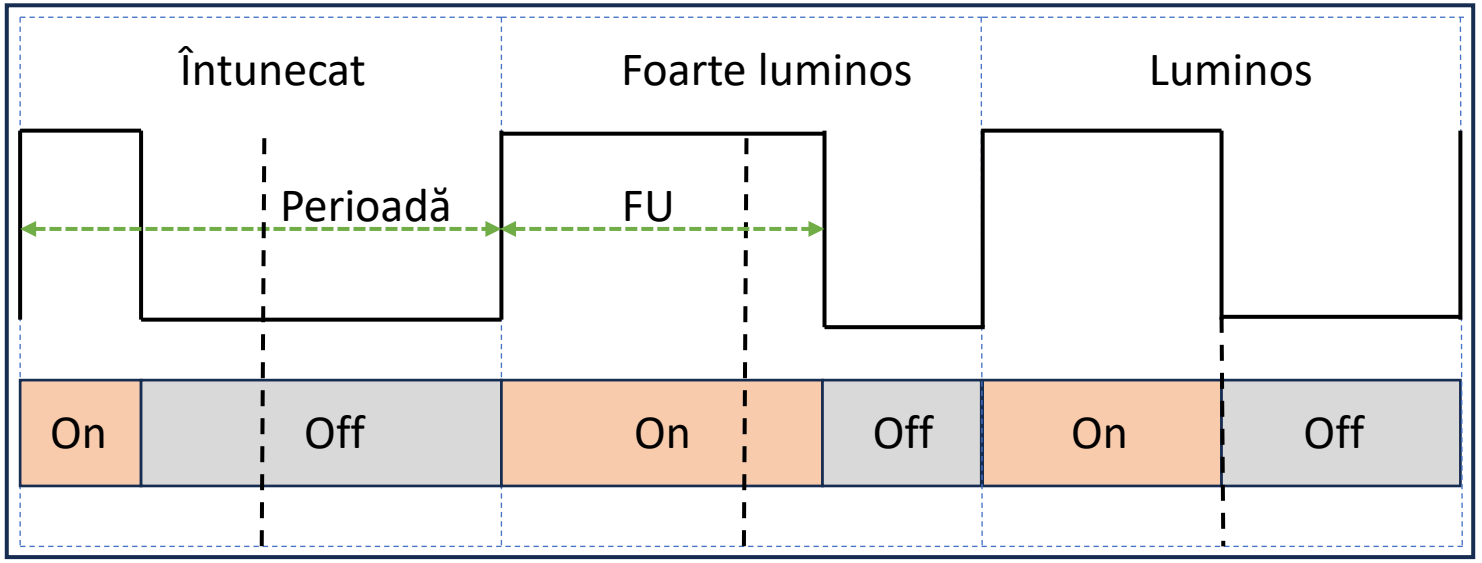


Generare semnal cu factor de umplere variabil (PWM)

Universitatea Transilvania din Brașov
FACULTATEA DE INGINERIE ELECTRICĂ ȘI ȘTIINȚA CALCULATOARELOR

- Aplicabilitate:
  - Controlul Motoarelor: pentru a controla viteza motoarelor DC. Prin varierea FU se controlează puterea medie aplicată motorului, ceea ce duce la modificarea vitezei acestuia
  - Dimarea LED-urilor: PWM permite controlul precis al intensității luminii emise de un LED. FU determină cât timp LED-ul este „on” în fiecare ciclu, controlând astfel luminozitatea percepută





Generare semnal cu factor de umplere variabil (PWM)

Universitatea Transilvania din Brașov
FACULTATEA DE INGINERIE ELECTRICĂ ȘI ȘTIINȚA CALCULATOARELOR

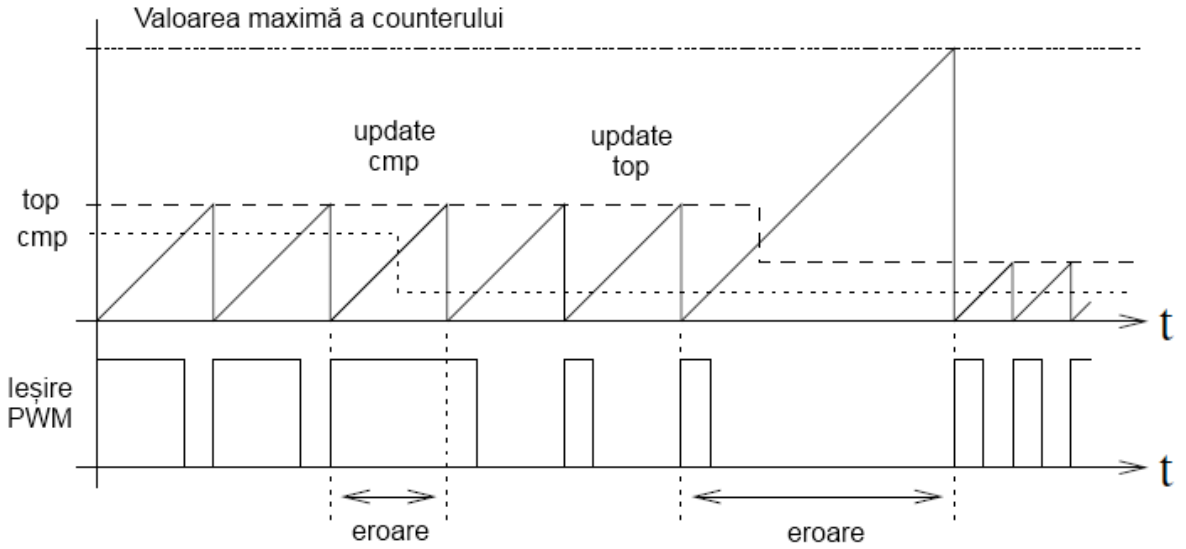
- Aplicabilitate:
  - Generarea Sunetelor: PWM este folosit pentru a genera semnale audio prin controlul difuzoarelor. Frecvența PWM poate fi ajustată pentru a produce diferite tonuri
  - Conversie Digital-Analog (DAC): PWM poate fi utilizat pentru a crea o tensiune medie variabilă, care, atunci când este filtrată, poate apropia o valoare analogică dintr-un semnal digital
- Modificarea frecvenței semnalului PWM se poate face prin:
  - Schimbarea valorii de prescalare
  - Schimbarea valorii de TOP: valoarea maximă la care contorul timer-ului ajunge înainte de a se reseta
  - Modificarea frecvenței ceasului de sistem: se poate modifica frecvența de tact (ceasul de sistem) a microcontrolerului. Aceasta afectează toate funcțiile bazate pe temporizare, inclusiv PWM

Implementare PWM

Universitatea Transilvania din Brașov
FACULTATEA DE INGINERIE ELECTRICĂ ȘI ȘTIINȚA CALCULATOARELOR

- Utilizând incrementare **sau** decrementare contor

- Valoarea maximă a contorului (top): limita superioară la care contorul se resetează. Valoarea se poate modifica, ceea ce schimbă frecvența semnalului PWM
- Valoarea de comparare (cmp): Este valoarea la care ieșirea PWM se comută între HIGH și LOW
- ieșirea PWM este un semnal pătrat ce variază în funcție de valorile cmp și top

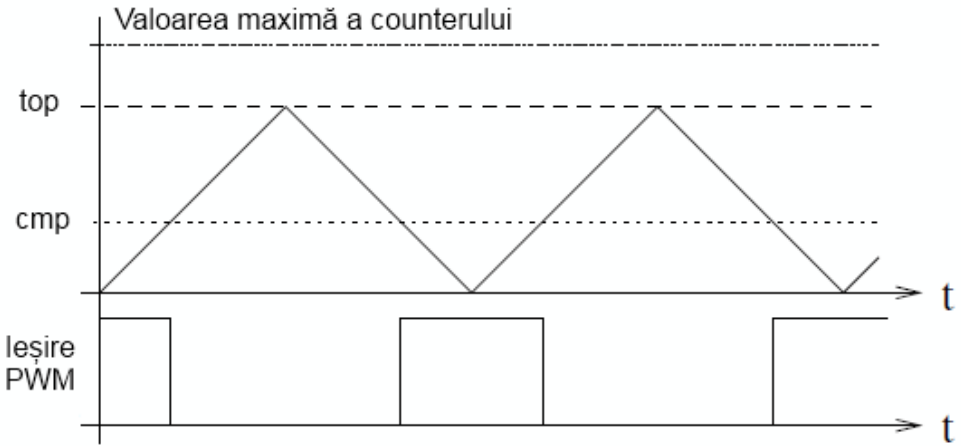


- Deoarece valorile TOP și cmp sunt actualizate în timpul funcționării, pot apărea mici erori în sincronizarea semnalului PWM
- Metoda se regăsește și sub denumirea de Fast PWM
- Se utilizează la la toate aplicațiile care au nevoie de PWM, mai puțin la cele care au nevoie de un PWM foarte precis (audio sau motoare fără perii (Brushless DC)

Pulse With Modulation (PWM)

Universitatea Transilvania din Brașov
FACULTATEA DE INGINERIE ELECTRICĂ ȘI ȘTIINȚA CALCULATOARELOR

- Utilizând incrementare **și** decrementare contor
  - Valoarea top: valoarea la care se configurează counter-ul să înceapă decrementarea. Determină valoarea frecvenței
  - Valoarea cmp: determină valoarea FU
  - Perioada PWM este: $T_{PWM} = 2 \times (N \times T_{clk})$ unde:
    - N este valoarea maximă a contorului (de exemplu, 255 pentru un contor pe 8 biți)
    - T_{clk} este perioada semnalului de ceas al contorului (de exemplu, 1 ms dacă ceasul este de 1 kHz)



  - Frecvența PWM este: $f_{PWM} = \frac{1}{T_{PWM}}$
  - Utilizată în aplicații de control al motoarelor, reglatorii de intensitate pentru LED-uri și alte aplicații care necesită controlul puterii
  - Este frecvent întâlnită în microcontrolerele moderne care dispun de module PWM integrate
  - Se mai numește și Phase Correct PWM

Aplicație – Generare PWM

Universitatea Transilvania din Brașov
FACULTATEA DE INGINERIE ELECTRICĂ ȘI ȘTIINȚA CALCULATOARELOR

- Exemplu de utilizare a unei timer pentru generare PWM
  - Cerință: să se implementeze un program care generează semnal PWM (OCR2) cu factor de umplere de 10% și 50%

```
// Timer/Counter 2 initialization
// Clock source: System Clock
// Clock value: 1000,000 kHz
// Mode: Fast PWM top=0xFF
// OC2 output: Non-Inverted PWM
// Timer Period: 0,256 ms
// Output Pulse(s):
TCCR2=(1<<PWM2) | (1<<COM21) | (0<<COM20) |
(1<<CTC2) | (0<<CS22) | (1<<CS21) | (0<<CS20);
OCR2=0x19;//setare factor de umplere la ~10%
```

Legendă:

TCCR2 – gestionează comportamentul și modul de operare al timerului

1<<PWM2 – activare PWM

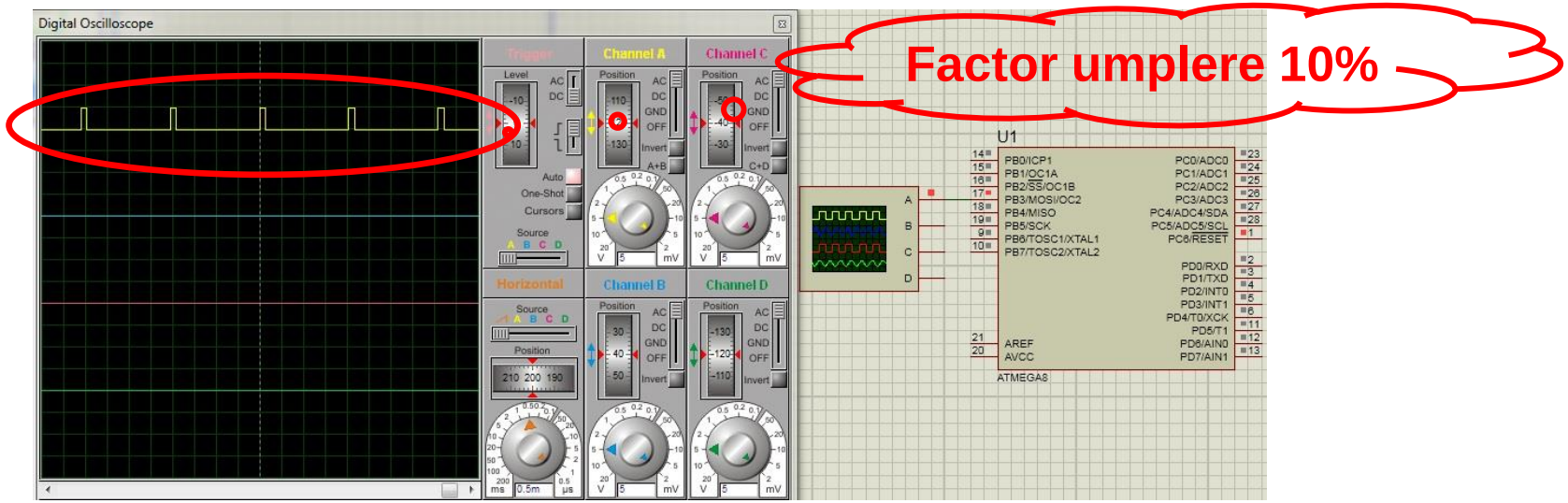
COM2x - controlează modul de comparare a ieșirii, pin OC2 resetat la o valoare atinsă

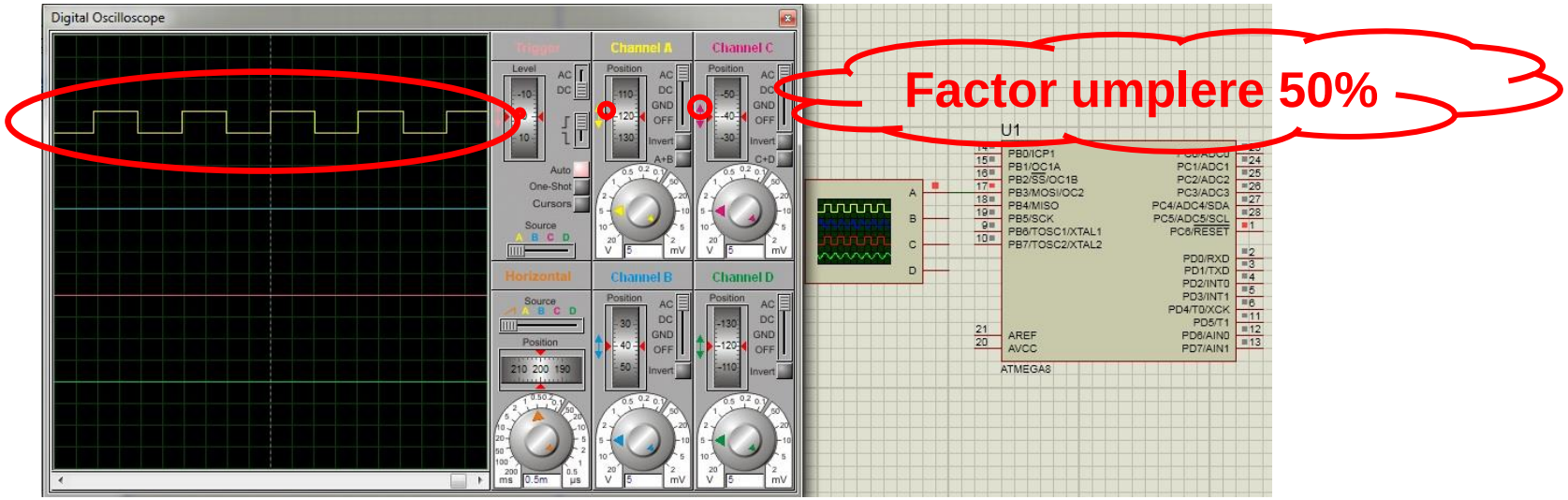
CTC2 - Clear Timer on Compare Match, timerul este resetat la zero atunci când atinge valoarea din OCR2

CS2x – selectare prescalar

Aplicație – Generare PWM

Universitatea Transilvania din Brașov
FACULTATEA DE INGINERIE ELECTRICĂ ȘI ȘTIINȚA CALCULATOARELOR

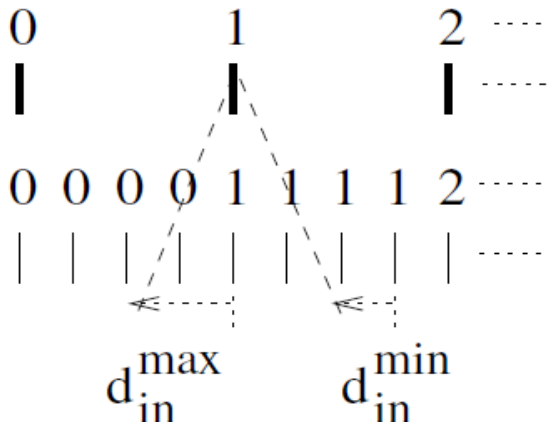




Capturi de evenimente externe utilizând un timer

Universitatea Transilvania din Brașov
FACULTATEA DE INGINERIE ELECTRICĂ ȘI ȘTIINȚA CALCULATOARELOR

- Permite evidențierea/marcarea apariției unui eveniment extern folosind valoarea curentă a unui modul timer
- Se poate configura condiția de apariție a evenimentului pe front crescător/descrescător al semnalului
- La apariția unui eveniment pe unul dintre pinii cu această funcționalitate:
  - Registrul Input Capture Register (ICR) este actualizat cu valoarea curentă a modulului timer utilizat
  - Flag-ul corespunzător este setat
  - Dacă întreruperea este configurată, se generează o întrerupere
- Evaluarea momentului de apariție a unui eveniment: $t_{eveniment} - t_{captură} \in (-d_{in}^{max}, P - 1 - d_{in}^{min})$



Valoare curentă registru timer

Semnal de ceas, aplicat modulului timer, configurat cu un prescalar ales de utilizator

Valoarea de timp salvata în ICR, când evenimentul a apărut

Clock-ul sistemului

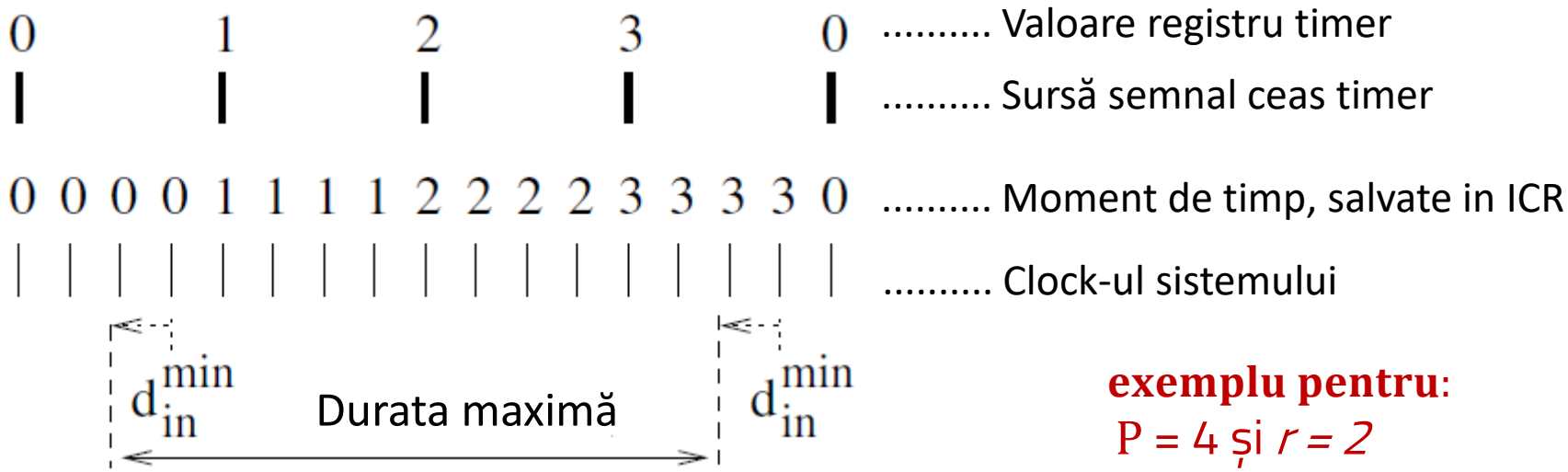
P = valoare prescalar

Capturi de evenimente externe utilizând un timer

Universitatea Transilvania din Brașov
FACULTATEA DE INGINERIE ELECTRICĂ ȘI ȘTIINȚA CALCULATOARELOR

- Eroarea maximă între două evenimente succesive este: $d_{in}^{max} + P - 1 - d_{in}^{min}$
  - Obs: trebuie avută în vedere o valoare minimă a prescalarului
- Intervalul maxim între două evenimente (până la overflow): $(2^r - 1) \cdot P$

unde, r reprezintă rezoluția timerului utilizat și configurat pentru captura de evenimente externe



..... Valoare registru timer

..... Sursă semnal ceas timer

..... Moment de timp, salvate in ICR

..... Clock-ul sistemului

exemplu pentru:
P = 4 și *r = 2*

Capturi de evenimente externe utilizând un timer

Universitatea Transilvania din Brașov
FACULTATEA DE INGINERIE ELECTRICĂ ȘI ȘTIINȚA CALCULATOARELOR

- Exemplu de măsurare a timpului între două evenimente
  - Configurare timer:
    - alegerea unui timer cu capabilitate de capturare evenimente externe
    - Alegerea unui prescalar care să faciliteze captura evenimentelor externe (relativ la viteza de apariție a evenimentelor externe)
  - Activare funcția de captură a mărimii de intrare:
    - Configurarea pinului de captură a intrării pentru a declanșa pe schimbarea frontului semnalului
  - Gestionarea întreruperii de captură a intrării:
    - Întreruperea respectivă trebuie configurată, la începutul programului
    - Când apare prima modificare a semnalului pe pinul de captură, valoarea curentă a timer-ului este salvată și stocată în registrul ICR
    - Când apare a doua modificare a semnalului (a frontului), are loc o altă captură, iar diferența dintre cele două valori capturate oferă timpul dintre cele două evenimente

Concluzii

- Modulele de tip timer facilitează dezvoltarea de funcționalități de monitorizare a timpului sau de generarea de evenimente la intervale exacte
- Aceste module se pot configura:
  - Timer, counter, module generator de PWM, watchdog timer sau RTC
- Accesul la datele dintr-un timer se poate face:
  - Folosind sistemul de întreruperi
  - Utilizând o metoda de interogare
- Timer-ul poate fi utilizat pentru captura de evenimente externe:
  - Schimbări de stare pe pinii specifici ai MC-ului
  - Determinarea timpului între două apariții succesive de semnale

Contact:
Email: gigel.macesanu@unitbv.ro
elearning.unitbv.ro - Sisteme cu Microprocesoare