

comun de activare a digitului (enable) și se legă la un potențial LOW. Segmentele individuale luminează aplicând potențial HIGH pe fiecare anod individual. ( AC pinii de anod - HIGH : Lumineață-HIGH  
 Pentru a activa afişajul 7-segmente de pe placă este nevoie să avem pozitionat switchul SW8 ca și în figura 2.5. Primele 4 poziții ale switchului permit conexiunea pinilor PA0-PA3 cu linia de selecție a digitului.

## 2.3 DESFASURAREA LUCRARII

să afișăm cifre de la 0-9, sunt controlate de registrul PORTC

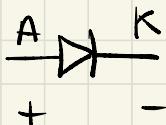
$$\boxed{1} \Rightarrow \text{PORTC} = 0b00111111_{\text{;}}$$

## LAB 3

$$\text{bit} \rightarrow 1 \rightarrow \text{DDRA} \ |= (1 \ll 3)$$

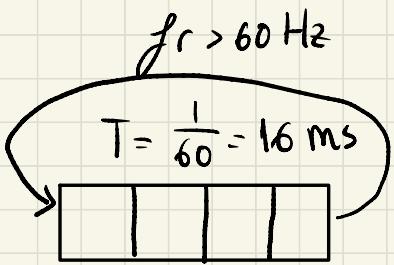
$$\text{bit} \rightarrow 0 \rightarrow \text{DDR B} \ &= \sim(1 \ll 4)$$

$$\text{test bit} \rightarrow 1 \text{ if } (\text{PINA} \ \& \ (1 \ll 5))$$



PA0-3 → ENABLE DIGIT

PC0-7 → ENABLE SEGMENT



## Conспект 3

~ MODULE DE TEMPORIZARE ~  
~ TIMERE ~

### 3.1 SCOPUL LUCRĂRII

- prezentare structură internă a unui timer
- modulul de configurare al registrilor
- o aplicație (ceas digital) 8 biți

### 3.2 PREZENTAREA CIRCUITELOR DE TEMPORIZARE

- construite dintr-un numărător, având unul sau mai multe timere pe 8 / 16 biți
- sunt folosite pt diferite aplicații :
  - măsurarea unei perioade de timp
  - generarea de seminale cu o frecvență impusă
- mecanismul de comparare permite controlul unor semnale de ieșire sau chiar generare de semnale PWM (pulse width modulation)

#### 3.2.1 NUMĂRĂTORUL

- fiecare timer este la bază un numărător care este incrementat la fiecare semnal de ceas, dlr. de numărare fiind fixă sau configuriabilă
- pentru un timer cu o rezoluție de  $n$  biți, numărarea se face în intervalul  $[0, 2^n - 1]$ . Numărătorul și resetează când atinge max ( $2^n - 1$ ). Totuși limita se poate seta la oricăr din interval

### 3.2.2 TIMER AVAND CA SEMNAL DE TACT CEASUL SISTEMULUI

- modul predefinit, timer++ la fiecare front ascendent

### 3.2.3 TIMER CU PRESCALARARE (PRESALER)

- in acest mod se foloseste ceasul intern dar acesta este divizat de un prescaler (un numarator de o lungime variabila  $\approx 8$  sau 10 biti), care este incrementat odata cu ceasul sistemului.

- timer-ul va folosi ca semnal de ceas, semnalul provenit de la unul din bitii numaratorului. Daca prescaler-ul este ++ la fiecare front crescator al ceasului sistemului, atunci cel mai nesemnificativ bit va avea perioada dubla fata de cea a ceasului intern. In acest caz se foloseste o valoare a prescaler-ului egală cu 2 iar timer-ul va opera la jumătate din frecventa ceasului sistemului.

### 3.2.4 TIMER AVAND CA SEMNAL DE TACT SEMNALE EXTERNE

- timerul primește semnalul de ceas ca un semnal de ceas ca un semnal extern care este conectat pe un pin de intrare specific microcontrolerului

### 3.2.5 TIMER AVAND CA SEMNAL DE TACT UN OSCILATOR CU CRISTAL DE CUART EXTERN

- timerul foloseste semnalul de ceas obtinut cu ajutorul unui cristal de quart, conectat pe 2 pini predefiniti ai controlerului.

- cristal de quart ( $32.768 \text{ kHz}$ )

### 3.2.6 CAPTURA LA INTRARE

- este folosita pentru a marca in timp un eveniment, care poate sa fie constituit de un front  $\nearrow$  sau  $\downarrow$  sau de un anumit nivel logic.

- cand se intampla, pe pinul ICn, timer-ul copiază automat valoarea curentă in ICR (input capture register)

### 3.2.7 COMPARARE LA IEȘIRE

- OC (output compare) este opusul capturii la intrare.

- pt ICR, marcarul de timp este memorat atunci cand are loc un eveniment extern pe linia de intrare

- la OC se va genera un eveniment la ieșire atunci cand s-a

implină o perioadă de timp prestabilită

- controare OCR (output compare register) în care poti seta timpul de OC. Acest eveniment poate sa schimbe nivelul logic de ieșire (H sau L)

### 3.2.8 GENERARE PWM

- caz particular al modului de comparare la ieșire
- permite temporului să genereze un semnal de ieșire digital periodic al cărui timp de HIGH și perioadă pot fi configurate în program

## 3.3 DESFASURAREA LUCRARII

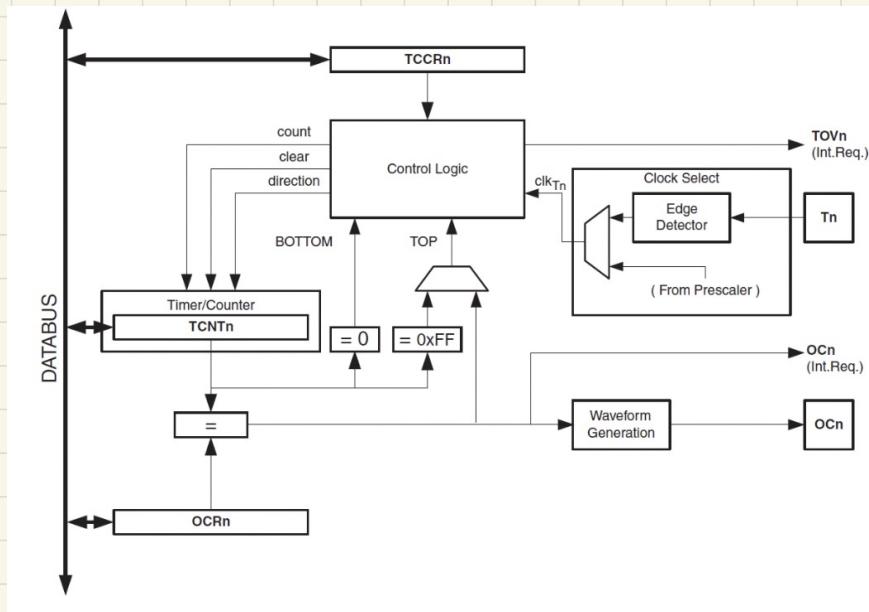


Figura 3.1. Diagrama bloc a unui timer pe 8 biți

- 3 registri : TCCR0 , TCNT0 și OCR0

- TCCR0 (time / counter control register) - stăruie diferențe funcționale

- TCNT0 - registrul de numărat

- OCR0 - registrul de comparare [0,255]

- vom folosi timer-ul în regim normal de numărat (WGM01 : 0 = 0) dacă trece de 255 se întoarce la 0

- in modul CTC (Clear time on Compare Match), registrul OCRD este folosit pt a manipula rezolutia numaratorului si de asemenea valoarea maxima de numarare. (WGM01:0 = 2)

$$f_{OCO} = \frac{f_{elk\_I10}}{N \cdot (1 + OCRO)}$$

f<sub>ck</sub>-I10 - frecvenția ceasului intern al sistemului

N - valoarea preschimbului (1, 8, 64, 256, 1024)

OCRD - val. scrisă în acest registru

## **DESCREREA REGISTRILOR**

TCCR0 – permite controlul modului de funcționare a timerului

Bitu WGM01 :00 - schimbarea modului de operare (Normal , CTC,PWR)

Bîtu COM 01 : 00 - schimbarea funcționalității primului OCÖ

Bidu CS02 : 00 - oferă posibilitatea selectari semnalului de ceas intern filtrat prin presele

TCNTO - permite accesul direct atât pentru scrierea cat și pentru citirea numărătorului pe 8 biți

OCRO - conține o val. pe 8 biti care este comparată mereu cu registrul de numărare (TCNT0)

# ~Conспект 4~

## ~SISTEMUL DE INTRERUPERI~

### 4.1 SCOPUL LUCRĂRII

- înțelegerea mecanismului de intreruperi prezent la microcont.
- aplicații la care se folosesc intreruperile.

### 4.2 CE SUNT INTRERUPERILE

- microcontrolele trebuie să reacționeze la evenimente.
- evenimente = schimbarea unor stări ale sistemului, făcând microcont. să reacționeze.
  - intreruperi interne (din microcont.)
  - intreruperi externe (int. cu mediul înconjur.)

CUM MONITORIZEAZĂ MICROCONT. ACESTE EVENIMENTE ??

1. să verifice periodic în program dacă au avut loc schimbări. Verificarea trebuie să se facă de mai multe ori cu o anumită perioadă de timp maxim.
2. folosirea sistemului de intreruperi (sistem implementat hardware pt a detecta evenimentele) ISR - Interrupt Service Routine

### 4.3 CONTROLUL INTRERUPERILOR

- fiecare intrerupere poate fi configurată pe 2 biți.

#### 1. Un bit de **Interrupt Enable**

- setat de app. să activeze sau dezactiveze intreruperea

- indică faptul că o funcție ISR trebuie apelată

#### 2. Un bit **Interrupt Flag**

- este setat în mom. când are loc un event.

- este resetat automat sau manual (**tine**) după ce se apelaază ISR.

- sursele care generă interrupții pot fi atât pini fizici cât și componente precum timer, conversor analog-digital etc.
- bitul IE și bitul de IE Global trebuie să fie activate ca o interrupție să poată fi luată în considerare

## 4.4 ORGANIZAREA ÎNTRERUPERILOR

Întrerupările sunt organizate în 2 mari categorii

1. Interne: provenite de la componente precum timer, memoria EEPROM, Conversor Analog-Numerice
2. Externe: provenite de la pinii de intrare/iesiri digitali

## 4.5 PRIORITĂȚILE ÎNTRERUPERILOR

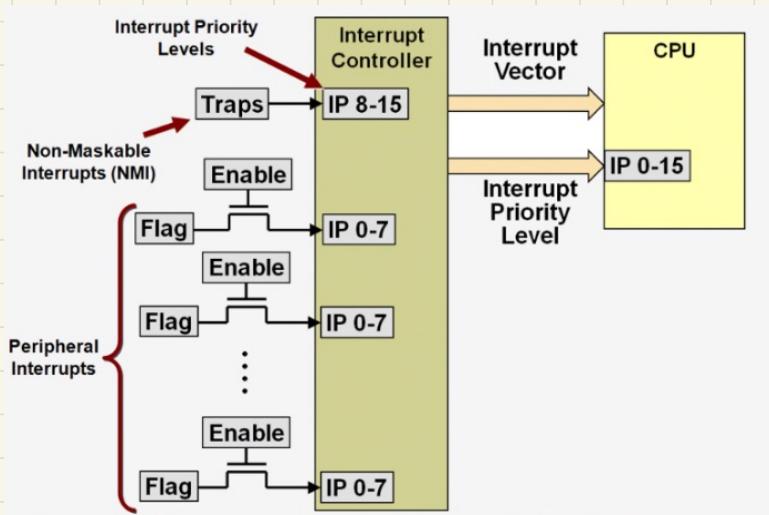


Figura 4.2. Controlerul de întreruperi, priorități

- dacă se întâmplă două interrupții în același moment sistemul de prioritare intră în funcțiune

## 4.6 LIBRARIA <avr/interrupt.h>

- sei() - set enable interrupts
- cli() - clear inputs

# ~ Conspect 5 ~

## ~ CONVERTORUL ANALOG-DIGITAL ~

### 5.1. SCOPUL LUCRARII

- principiul de functionare a unui modul de conversie analog-digitală
- structura generală împreună cu descrierea registrilor asociati convertorului
- modul de configurare a modulului de conversie pt citirea unei valori analogice provenite de la un potentiometru si senzor de temperatură.

### 5.2. PREZENTAREA MODULELOR DE CONVERSIE A/D

- I/O digitale au două valori posibile  $\downarrow 0$  LOW  
 $\downarrow 1$  HIGH
- în teorie e necesar să lucrăm cu mărimi analogice, astăzi că avem nevoie de 2 mărimi care se modifică constant între două limite. Aceste mărimi sunt transformate într-un semnal de tensiune.
- fiind un circuit digital microcont. trebuie să convertească semnalul analogic în semnal digital, acest lucru fiind realizat de convertorul analog-digital incorporat în microcontrolere

#### 5.2.1. CONVERSIA ANALOG-DIGITALA

- prezența modulelor A/D și D/A în structura unui MC este de o importanță majoră, deoarece e nevoie să prelucrăm sau elaborăm mărimi analogice
- convertoarele A/D integrate pe chip sunt cu aproximativ succesiive.
- convertoarele integrate în MC sunt relativ lente
- Fig. 5.1 prezintă principiul de funcționare a unui convertor A/D cu o rezoluție de 3 biți.
  - $r$  = rezoluție a convertorului
  - = numărul de biți pe care se va reprezenta valoarea digitală

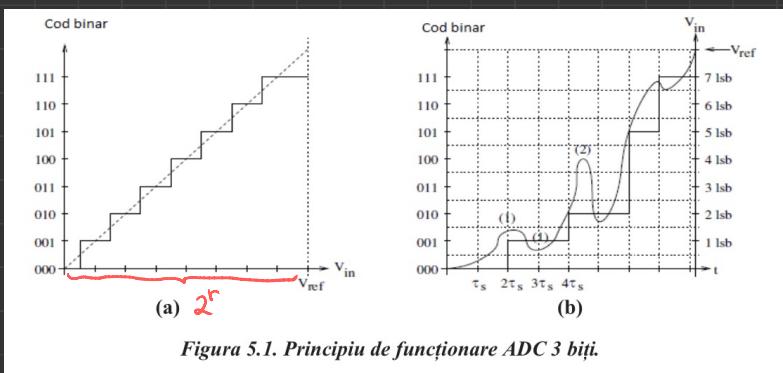


Figura 5.1. Principiu de funcționare ADC 3 biți.

- Fig 5.2 prezintă modul de operare al convertorului care funcționează pe baza aprox. successive.
- Fig 5.3 prezintă modul de funcționare al acestuia

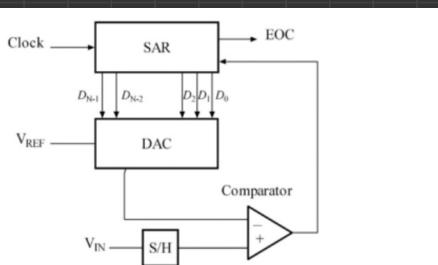


Figura 5.2. Schema de principiu a unui ADC cu aproxiimări successive

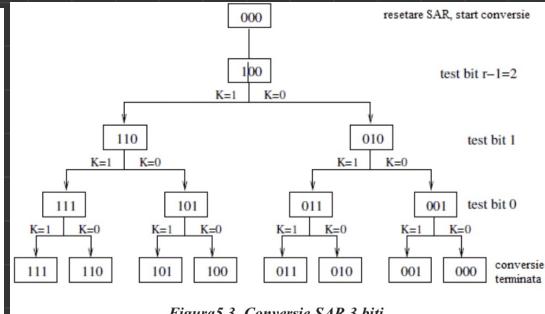
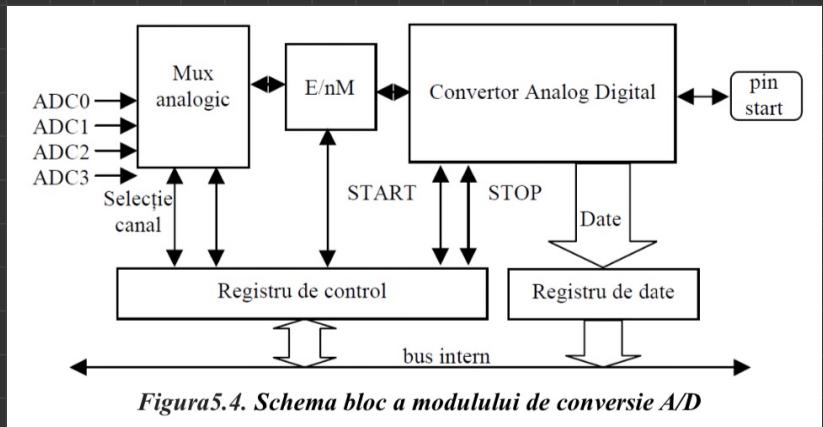


Figura 5.3. Conversie SAR 3 biți.

In Fig 5.2:

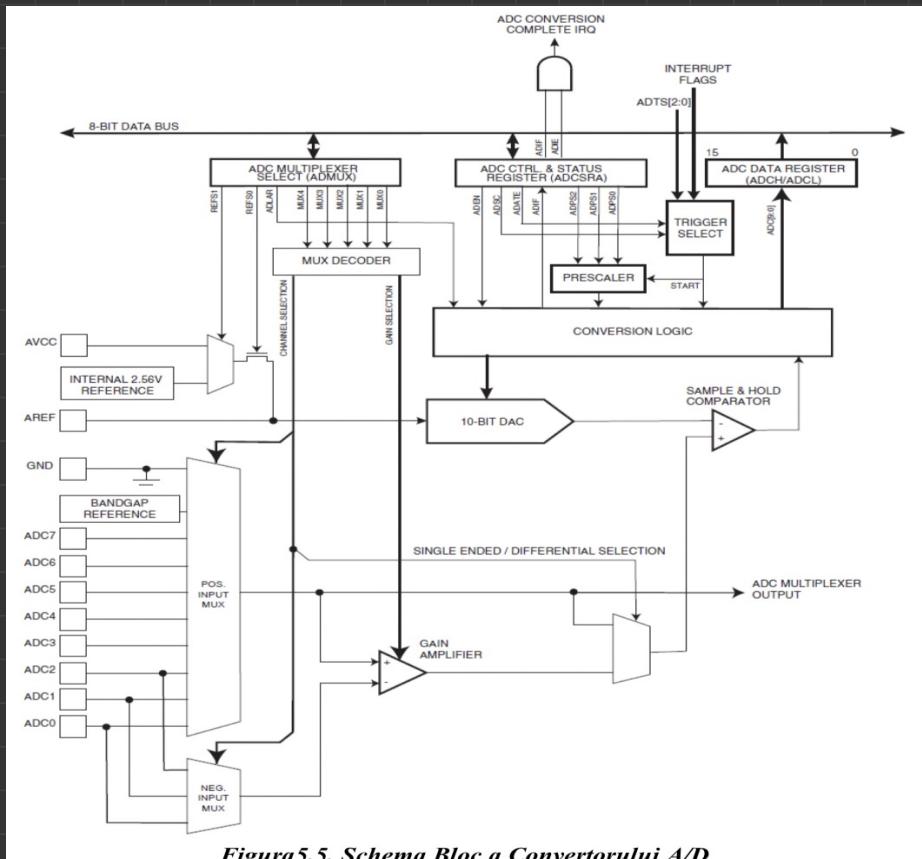
- $V_{IN}$  - semnalul măsurat
- $V_{ref}$  - tensiunea de referință
- DAC - convertor numeric-analogic
- SAR - registrul de aprox. successive
- S/H - circuit de esantionare și memorare a tensiunii ce se măsoară în acel moment

- In Fig 5.4 este prezentată o schema simplă bloc a unui modul de conversie A/D. Circ. analogic de intrare constă într-un multiplexor analogic, un circuit de memorare și un convertor A/D cu aprox. successive. Rezultatul conversiei este stocat în registrul de date (conține rezultatul ultimei conversii).



*Figura 5.4. Schema bloc a modulului de conversie A/D*

### 5.3 DESFASURAREA LUCRARII



*Figura 5.5. Schema Bloc a Convertorului A/D*

## DESCREREA REGISTRILOR

# LAB 6

0 - 5V

10 bits  $0 \rightarrow 2^{10} - 1 = 1023$

$$V_{ref} = 5V$$

$$D_g = 1, D_{g-0} = 0$$

$$SAR = 512$$

$$V_{DAC} = \frac{512}{1024} \cdot 5V = 2.5V$$

$$V_n = 3V$$

$$\text{if } V_{DAC} > V_n \Rightarrow D_g = 0$$

$$\text{else } D_g = 1$$

# ~ Conспект 6 ~

## - GENERAREA SEMNALELOR PWM -

### 6.1 SCOPUL LUCRARII

- înțelegerea mecanismului prin care se generează semnalul PWM cu ajutorul modulului de temporizare de pe un microcontroler ATmega32
- folosim semnalul PWM pt obținerea unui convertor digital-analogic simplu

### 6.2 CE ESTE UN SEMNAL PWM

- Semnalul Pulse Width Modulation este un semnal periodic dreptunghiular la care se poate modifica în mod controlat factorul de umplere
- Factorul de umplere este un procent din perioada semnalului pentru care semnalul se află în stare HIGH

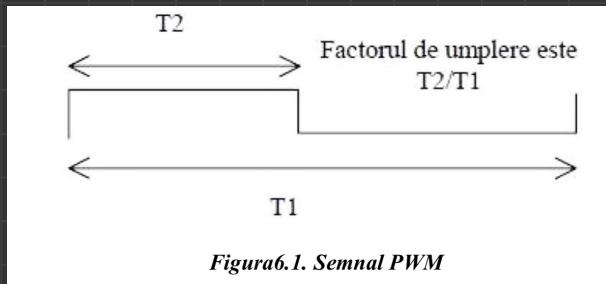


Figura 6.1. Semnal PWM

- Modul de operare a timer-ului în regim PWM este un curent particular al comparării la ieșire. Semnalul digital periodic generat de timer poate fi obs. pe un pin numit pin de comparație la ieșire ( $OC_n$ ). Timerul numără (0-255)

$$f_{PWM} = \frac{f_{clk\_I10}}{N \cdot 256}$$

frecvență internă →  
valoarea preseleului