

# Sisteme cu microprocesoare

## - Noțiuni introductive -

## - Microcontrolerul ATmega 328P -

### I. SCOPUL LUCRĂRII:

Lucrarea de laborator are ca scop:

- familiarizarea studentului cu noțiunile de bază introductive în domeniul programării și al operațiilor logice binare;
- familiarizarea studentului cu noțiunile legate de tehnologiile de integrare a dispozitivelor de calcul specializate;
- familiarizarea studentului cu noțiunea de sistem de calcul și conceptele fundamentale legate de sistemele încorporate (eng. Embedded Systems);
- studiul și analiza funcțională a arhitecturii microcontrolerului ATmega 328P, împreună cu metodele și procedurile de programare specifice, [1].

### II. INTRODUCERE:

În cadrul aplicațiilor specifice domeniilor ingineresti, (cu precădere cel al ingineriei electrice și electronice), de cele mai multe ori, este necesară proiectarea unor sisteme de comandă și control care au rolul de a deservi anumite procese tehnologice. Prin acest fapt a fost facilitată de utilizarea, pe scară largă a sistemelor microprogramabile de tip (eng.) „**microcontroller**”, [2]. Microcontrolerul există în echipamente precum:

- aparatura electrocasnică și dispozitive sau bunuri de larg consum (ex. imprimanta);
- sistemele de monitorizare și automatizare (ex. termostatul);
- aparatura specializată în domeniului de măsurare (ex. termometrul medicinal);
- aparatura industrială (ex. mașina pentru înfiletat);

În plus față de microcontroler mai există și alte tipuri de sisteme microprogramabile precum:

- sistemele monobloc cu microprocesoare dedicate (eng. System On a Chip – SOC);
- procesoarele digitale de semnal (eng. Digital Signal Processor – DSP);
- ariile de porți programabile (eng. Field Programmable Gate Array – FPGA);

În literatura de specialitate toate aceste dispozitive microprogramabile pot fi regăsite sub denumirea de „micro sisteme integrate sau încorporate” (eng. embedded systems). Integrarea sau încorporarea dispozitivului de procesare central (ex. microcontroler) conduce în final la obținerea unui echipament sau produs finit. În general, integrarea dispozitivului central de calcul se realizează pe baza unui cablaj imprimat care la rândul său inter-conectează o serie de **periferice** precum:

- componente electronice pentru semnalizare (ex. indicatori de tip LED sau difuzor);
- componente pentru preluarea comenzilor externe (ex. butoane sau senzori);
- dispozitive pentru comunicare cu alte echipamente (ex. adaptoare Bluetooth – Serial);

În cadrul proceselor de fabricare a echipamentelor electronice se pot distinge două tehnologii de integrare a dispozitivului central de procesare:

- tehnologia (eng.) „On board” (integrarea pe un cablaj imprimat);

<https://epe.utcluj.ro/index.php/sisteme-cu-microprocesoare/>

Conf. Dr. Ing. Ioana – Cornelia Gros – [Ioana.Gros@emd.utcluj.ro](mailto:Ioana.Gros@emd.utcluj.ro)

Asist. Dr. Ing. Lucian – Nicolae Pintilie – [Lucian.Pintilie@emd.utcluj.ro](mailto:Lucian.Pintilie@emd.utcluj.ro)

- tehnologia (eng.) „System On a Chip” (integrarea monobloc într-o singură capsulă);

Microcontrolerul reprezintă o soluție de integrare **System On a Chip (SOC)** a tuturor perifericelor **necesare** funcționării (ex. memorie RAM, ROM, Flash, procesor etc.).

În practică există mai multe modele și variante constructive (de la diverși producători) ale microcontrolerelor precum:

- MicroChip PIC 10F200, [3];
- Infineon XMC1400, [4];
- STMicroelectronics STM32 F205, [5];

**- Atmel, AVR, MicroChip ATmega 328P;**

Platformele de dezvoltare construite în jurul unui dispozitiv central de procesare reprezintă o soluție de integrare **On Board** a tuturor perifericelor opționale dar **nu neapărat necesare sau vitale** funcționării sistemului de calcul.

În practică există mai multe modele și variante constructive (de la diverși producători) ale platformelor de dezvoltare precum:

- MicroChip Explorer 16/32 Development Board [6];
- Infineon AURIX TC275 ARDSBTOBO1 [7];
- STMicroelectronics NUCLEO - L010RB [8];
- Arduino NANO, MEGA, UNO, DUE, Leonardo etc [9].**

### III. INTEGRAREA MICROCONTROLERULUI ATMEGA 328P ÎNTR-O APLICAȚIE:

În vederea evidențierii conceptelor teoretice fundamentale specifice funcționării sistemelor microprogramabile (cu preponderență cele cu microcontrolere), va fi analizată arhitectura microcontrolerului ATmega 328P.

Conform catalogului dispozitivului electronic ATmega 328P, acesta poate fi regăsit în trei variante constructive, anume, TQFP, MLF, PDIP-28 (Fig. 1).

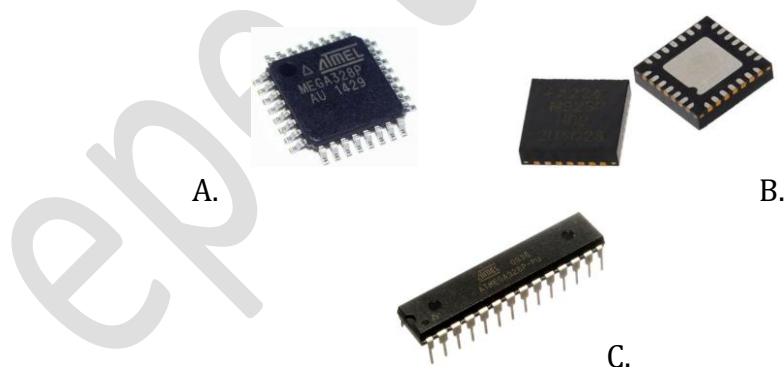


Fig. 1 – Microcontrolerul ATmega 328P (A. TQFP, B. MLF, C. PDIP-28)

Arhitectura internă a microcontrolerului ATmega 328P (Fig. 2), prezintă o serie de etaje constructive precum:

- unitate centrală de procesare cu frecvența maximă de lucru la 16 [MHz];
- memorie EEPROM având capacitatea de 1 [KB] și 100000 de cicluri de inscripționare;
- memorie SRAM având capacitatea de 2 [KB];
- memorie FLASH având capacitatea de 32 [KB] și 10000 de cicluri de inscripționare

<https://epe.utcluj.ro/index.php/sisteme-cu-microprocesoare/>

Conf. Dr. Ing. Ioana – Cornelia Gros – [Ioana.Gros@emd.utcluj.ro](mailto:Ioana.Gros@emd.utcluj.ro)

Asist. Dr. Ing. Lucian – Nicolae Pintilie – [Lucian.Pintilie@emd.utcluj.ro](mailto:Lucian.Pintilie@emd.utcluj.ro)

- numărător sau temporizator având rezoluția pe 8 și 16 biți;
- interfețe de comunicare UART / USART, I<sup>2</sup>C, TWI, SPI, debugWIRE;
- registre de intrare și ieșire de uz general (eng. General Purpose Input or Output – GPIO);
- convertoare analog – digitale (eng. Analog to Digital Converter – ADC);

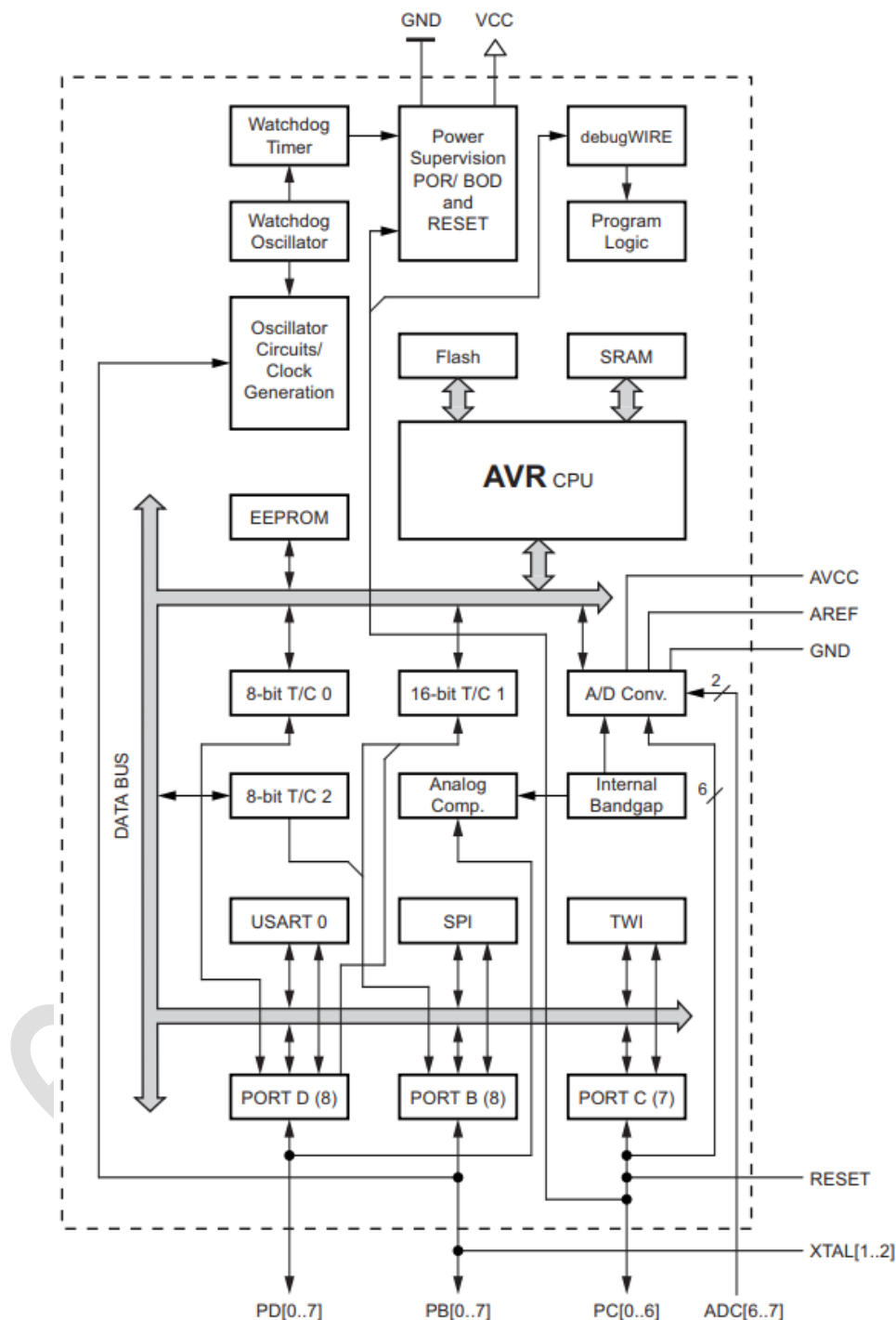


Fig. 2 – Arhitectura internă a microcontrolerului ATmega 328p (schema bloc)

Tot în cadrul datelor de catalog, se regăsește diagrama de distribuție a terminalelor (eng. pinout diagram) reprezentând funcțiile posibile ale acestora (Fig. 3) [10].

<https://epe.utcluj.ro/index.php/sisteme-cu-microprocesoare/>

Conf. Dr. Ing. Ioana – Cornelia Gros – [Ioana.Gros@emd.utcluj.ro](mailto:Ioana.Gros@emd.utcluj.ro)

Asist. Dr. Ing. Lucian – Nicolae Pintilie – [Lucian.Pintilie@emd.utcluj.ro](mailto:Lucian.Pintilie@emd.utcluj.ro)

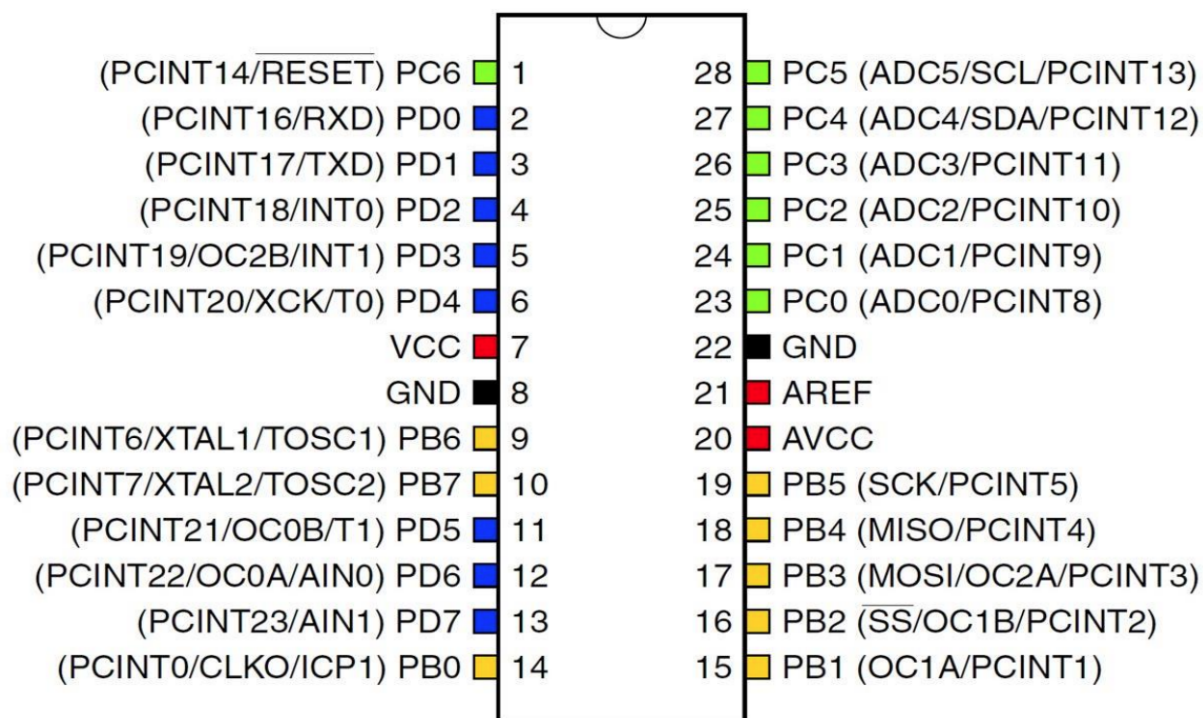


Fig. 3 – Diagrama de distribuție a terminalelor și funcțiile alocate acestora la nivel de microcontroler

Pentru a integra microcontrolerul ATmega 328P într-o anumită aplicație, este necesară introducerea a cel puțin trei etaje de circuit fundamentale (Fig. 4) [11]:

- etajul de alimentare pe bază de stabilizator liniar de tensiune (marcaj verde);
- etajul oscilatorului cu cuarț (marcaj portocaliu);
- etajul pentru tratarea rutinei de re-inițializare (eng. RESET), (marcaj albastru);

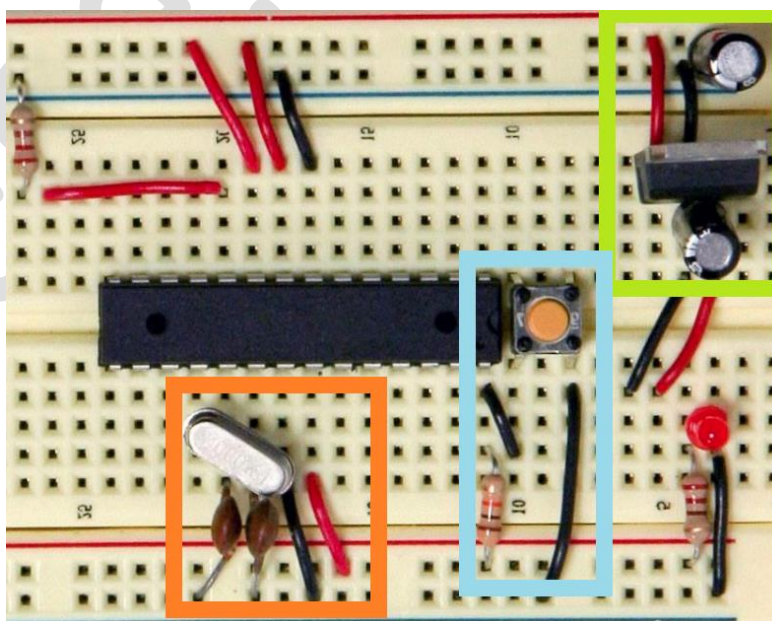


Fig. 4 – Integrarea microcontrolerului ATmega 328P într-o aplicație

<https://epe.utcluj.ro/index.php/sisteme-cu-microprocesoare/>

Conf. Dr. Ing. Ioana – Cornelia Gros – [Ioana.Gros@emd.utcluj.ro](mailto:Ioana.Gros@emd.utcluj.ro)

Asist. Dr. Ing. Lucian – Nicolae Pintilie – [Lucian.Pintilie@emd.utcluj.ro](mailto:Lucian.Pintilie@emd.utcluj.ro)



În vederea transferării inițiale a informației numerice în memoria microcontrolerului și în vederea programării acestuia, dispozitivul programator (ex. SparkFun USB Tiny AVR Prog) se va atașa la terminalele „VCC”, „AREF”, „AVCC”, „GND”, „SCK”, „MOSI”, „MISO” și „RESET” ale microcontrolerului (Fig. 5) [12]. Procedura utilizată în această situație, reprezintă un protocol de comunicare direct (între calculatorul gazdă și microcontroler) prin interfață USB – SPI (eng. Serial to Peripheral Interface).

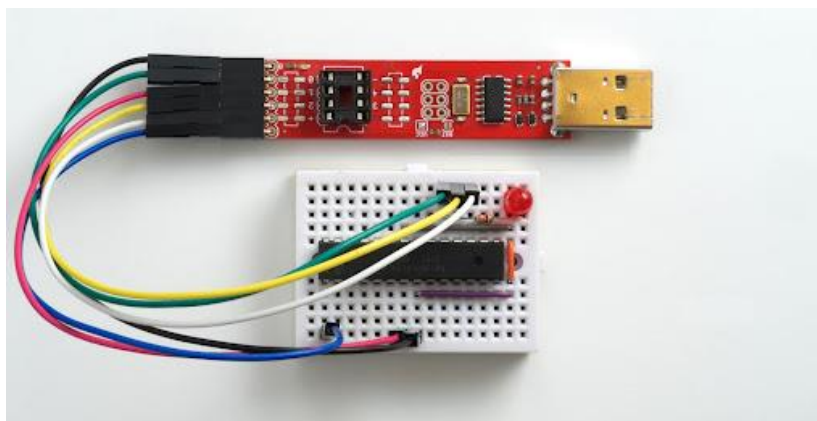


Fig. 5 – Atașarea dispozitivului programator USB – SPI la microcontrolerul ATmega 328P

Există, de asemenea, posibilitatea de a programa microcontrolerul ATmega 328P și prin intermediul unui dispozitiv adaptor de tip USB – TTL – Serial (ex. FTDI FT232RL) (Fig. 6) [13]. Această procedură de programare și comunicare cu calculatorul gazdă, (spre deosebire de metoda expusă anterior), necesită transferarea inițială (în memoria FLASH a microcontrolerului) a secvenței de cod pentru inițializarea a comunicației Serial dintre microcontroler și calculatorul gazdă. În literatura de specialitate această secvență de cod care inițializează comunicația Serial în vederea programării, poartă denumirea (eng.) „Boot Loader”. Aceasta trebuie inscripționată inițial cu ajutorul unui dispozitiv programator de tip USB – SPI.

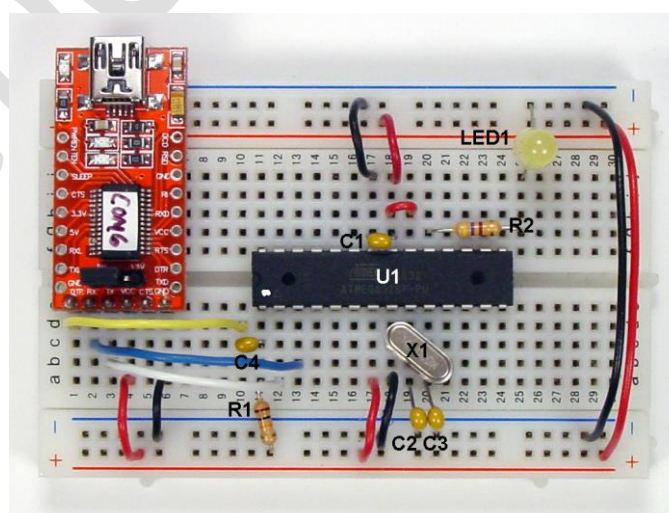


Fig. 6 – Atașarea dispozitivului programator USB – Serial la microcontrolerul ATmega 328P

#### IV. METODE DE PROGRAMARE A MICROCONTROLLERULUI:

Există patru metode de programare a unui sistem de calcul, anume:

- programarea în cod mașină;
- programare în limbaj de asamblare;
- programare în sintaxă;
- generare automată de cod pe baza mediilor grafice de programare;

Trecerea de la un nivel superior al limbajului de programare (precum mediul grafic), la un nivel inferior (precum cod mașină sau limbaj de asamblare), în literatura de specialitate poartă denumirea de „proces de compilare” (eng. compilation). Practic, procesul de compilare reprezintă „translatarea” codului program de la un nivel inteligibil al factorului uman înspre un nivel compatibil cu sistemul de calcul dat (Fig. 7), [14].

Correspondența dintre elementele de program regăsite în limbajele de programare se realizează prin intermediul funcțiilor și a bibliotecilor de funcții (eng. function libraries). Există biblioteci de funcții care asigură trecerea de la mediul grafic la sintaxă, de la sintaxă la limbaj de asamblare, de la limbajul de asamblare la codul specific arhitecturii sistemului de calcul, anume instrucțiuni în cod binar sau hexazecimal.

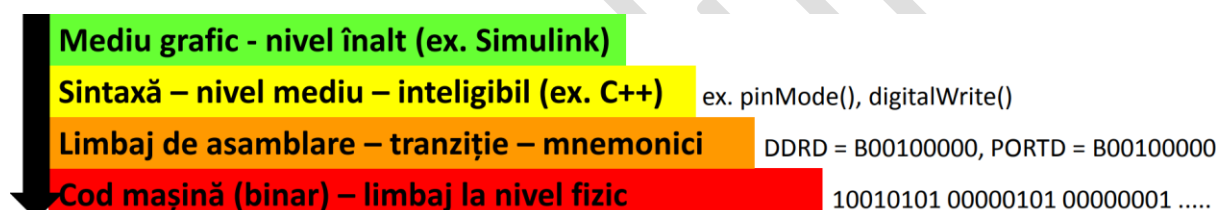


Fig. 7 – Metode și limbaje de programare specifice lucrului cu microcontrolere  
Ilustrarea procesului de compilare

Există de asemenea, posibilitatea de a manipula în mod direct conținutul registrelor de memorie prin așa zisa **metodă hibridă de programare în sintaxă și limbaj de asamblare**, denumită în literatura de specialitate (eng. Mixed assembly – syntax). Memoria microcontrollerului ATmega 328P este organizată tabelar în registre a câte opt unități fundamentale, anume opt biți, sau un octet (eng. byte).

Registrul reprezintă o zonă dedicată din memorie pentru a îndeplini o anumită funcție în cadrul microcontrollerului. Un exemplu de astfel de registru, poate fi „SREG” (Fig. 8), registrul de stare.

#### SREG – AVR Status Register

The AVR status register – SREG – is defined as:

Bit	7	6	5	4	3	2	1	0	
0x3F (0x5F)	I	T	H	S	V	N	Z	C	SREG
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Fig. 8 – Structura registrului de stare „SREG” al microcontrollerului ATmega 328P

Rolul registrului de stare SREG este de a consemna situațiile prevăzute sau neprevăzute care au loc în timpul efectuării unor operații aritmetice sau logice. În cadrul registrului de stare, există opt poziții binare, care descriu apariția unui anumit „eveniment de eroare de calcul”. Fiecare poziție din șirul de biți componenți ai registrului SREG are o anumită semnificație.

Există de asemenea, la nivelul memoriei microcontrolerului ATmega 328P, o zonă specială, rezervată terminalelor specifice intrărilor și ieșirilor digitale. Prin intermediul intrărilor și ieșirilor de uz general (GPIO) microcontrolerul poate furniza sau accepta din exterior un nivel de tensiune cuprins între zero și valoarea nivelului tensiunii de alimentare. Terminalele care au funcția de intrare și ieșire digitală, sunt grupate câte opt la nivelul carcasei microcontrolerului. Grupările a câte opt terminale se numesc „PORTx” (unde „x” reprezintă litera alocată în ordine crescătoare a grupărilor). Pentru exemplificare, se va alege gruparea de terminale „PORTD”.

Registrul care controlează modul de lucru al unei grupări de tip PORTx, este denumit „DDRD” (eng. Data Direction Register of PORT D) (Fig. 9).

Dacă valoarea tuturor pozițiilor de memorie din registrul „DDRB” va fi zero, atunci toate terminalele de pe carcasa microcontrolerului care aparțin grupului „PORTD” vor avea rolul de intrări digitale, și vor prelua o diferență de potențial. În cazul în care diferența de potențial este diferită de zero, la unul dintre terminale, în zona de memorie, în registrul „PORTD” se va modifica valoarea de la zero (0) la valoarea unu (1).

În cazul contrar, când valoarea tuturor pozițiilor din registrul DDRD va fi unu logic, atunci, terminalele de la nivelul carcasei grupate în PORTD, vor avea funcția de ieșire digitală, iar conform conținutului din memorie aflat în registrul de stare PORTD, potențialul la terminalele date, se va modifica conform secvenței memorate.

#### 13.4.9 DDRD – The Port D Data Direction Register

Bit	7	6	5	4	3	2	1	0	
0x0A (0x2A)	DDD7	DDD6	DDD5	DDD4	DDD3	DDD2	DDD1	DDD0	DDRD
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Fig. 9 - Structura registrului de stabilire a modului de lucru „DDRD”

Registrul „PORTD” gestionează diferența de potențial la nivelul terminalelor grupate în „PORTD” de la periferia carcasei microcontrolerului (Fig. 10).

#### 13.4.8 PORTD – The Port D Data Register

Bit	7	6	5	4	3	2	1	0	
0x0B (0x2B)	PORTD7	PORTD6	PORTD5	PORTD4	PORTD3	PORTD2	PORTD1	PORTD0	PORTD
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Fig. 10 - Structura registrului „PORTD” pentru stabilirea diferenței de potențial la nivelul terminalelor grupate în registrul „PORTD”

Manipularea în mod direct a informației binare aflată în conținutul registrelor din memoria microcontrolerului ATmega 328P presupune, cunoașterea temeinică a metodelor și procedeele de programare în limbaj de asamblare și gestionarea eficientă a valorilor numerice prin transformarea lor dintr-o bază de numerație în alta.

Pentru a gestiona valorile numerice dintr-un șir binar în mod corect, se vor respecta trei reguli de bază:

### **1. Într-un șir de valori scrise în sistem de numerație binar indicele de ordonare în șir începe de la ZERO tot timpul!**

Drept consecință a primei reguli, rezultă a doua regulă pentru gestionarea șirurilor de valori binare, anume:

### **2. Valoarea maximă care poate fi exprimată într-un șir binar are expresia $(2^n)-1$ !**

În cadrul expresiei  $(2^n)-1$ , „2” reprezintă baza de numerație iar „n” reprezintă numărul de termeni sau de biți din cadrul șirului binar utilizat în vederea reprezentării unei valori numerice. Rezultă faptul că, pentru determinarea valorii în sistemul de numerație zecimal a unui șir binar, se va utiliza metoda de exprimare sub forma unui polinom ponderat cu baza „2”.

Spre exemplu: Se dă șirul binar 1001, să se determine valoarea numerică zecimală exprimată prin intermediul șirului binar dat.

$$X_{(2)4} = b_{4-1} \cdot 2^{4-1} + b_{4-2} \cdot 2^{4-2} + b_{4-3} \cdot 2^{4-3} + b_{4-4} \cdot 2^{4-4}$$

$$X_{(2)4} = b_3 \cdot 2^3 + b_2 \cdot 2^2 + b_1 \cdot 2^1 + b_0 \cdot 2^0$$

$$X_{(2)4} = b_3 \cdot 2^3 + b_2 \cdot 2^2 + b_1 \cdot 2^1 + b_0 \cdot 2^0$$

$$X_{(2)4} = 1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0$$

$$X_{(2)4} = 8 + 0 + 0 + 1 = 9$$

Unde „b<sub>n-i</sub>” reprezintă elementul din șir (bitul) cu indicele de ordine „n-i”.

Totodată pe baza primelor două reguli, va fi posibilă întocmirea tabelului de adevăr, de corespondență între valorile binare, zecimale și hexazecimale. Spre exemplu, în vederea exprimării în binar, pe patru biți a unei valori zecimale, și în hexazecimal, tabelul de adevăr se va întocmi astfel:



N	Hex.	b3	b2	b1	b0
0	0	0	0	0	0
1	1	0	0	0	1
2	2	0	0	1	0
3	3	0	0	1	1
4	4	0	1	0	0
5	5	0	1	0	1
6	6	0	1	1	0
7	7	0	1	1	1
8	8	1	0	0	0
9	9	1	0	0	1
10	A	1	0	1	0
11	B	1	0	1	1
12	C	1	1	0	0
13	D	1	1	0	1
14	E	1	1	1	0
15	F	1	1	1	1

Ca și consecință celei de-a doua regulă, rezultă a treia regulă de transformare numerică din orice bază de numerație în zecimal, anume:

**3. Există o formula general valabilă pentru transformare în sistem zecimal din orice bază de numerație având o reprezentare pe un număr „n” de biți:**

$$X_{(z)n} = \sum_{\substack{i=1 \\ i \leq n}}^n b_{n-i} \cdot z^{n-i}$$

Unde, „z” reprezintă baza de numerație, „n” numărul de biți iar „i” reprezintă ordinea din șirul de valori binare.

Spre exemplu: Fie valoarea numerică „0xFE” exprimată în hexazecimal. Să se determine, valoarea numerică în sistem zecimal corespunzătoare numărului dat inițial.

$$X_{(16)2} = b_{2-1} \cdot 16^{2-1} + b_{2-2} \cdot 16^{2-2}$$

$$X_{(16)2} = b_1 \cdot 16^1 + b_0 \cdot 16^0$$

$$X_{(16)2} = 15 \cdot 16^1 + 14 \cdot 16^0 = 240 + 14 = 254$$

Efectuarea operațiilor în binar presupune o serie de „evenimente” care pot avea loc în memoria sistemului de calcul precum:

- rezultat zero (presupune activarea bitului „Z” din registrul SREG);
- rezultat negativ (presupune activarea bitului „N” din registrul SREG);
- transport pe jumătate de octet (presupune activarea bitului „H” din registrul SREG);
- transport pe întreg octetul (presupune activarea bitului „C” din registrul SREG);

<https://epe.utcluj.ro/index.php/sisteme-cu-microprocesoare/>

Conf. Dr. Ing. Ioana – Cornelia Gros – [Ioana.Gros@emd.utcluj.ro](mailto:Ioana.Gros@emd.utcluj.ro)

Asist. Dr. Ing. Lucian – Nicolae Pintilie – [Lucian.Pintilie@emd.utcluj.ro](mailto:Lucian.Pintilie@emd.utcluj.ro)

Spre exemplu, la însumarea valorii binare „0001” cu „0001”, are loc fenomenul de transport binar, de la poziția bitului zero la poziția bitului unu.

Fenomenul de transport binar are loc din cauza depășirii domeniului de reprezentare al bazei de numerație în poziția din șir considerată!

#### **BINAR:**

```
0001 +
0001
-----
0010
```

#### **ZECIMAL:**

```
99999999 +
      1
-----
E 00000000
```

#### **HEXAZECIMAL:**

```
0x0F +
0x01
-----
0x10 → BINAR: 0001 0000
```

În limbaj de asamblare, pentru a manipula conținutul registrelor de memorie corespunzătoare intrărilor și ieșirilor digitale, se vor utiliza următoarele instrucțiuni:

- sensul curentului prin terminale digitale grupate în PORTD:

DDRD = B00000000; (terminalele îndeplinesc funcția de intrări digitale);

DDRD = B11111111; (terminalele îndeplinesc funcția de ieșiri digitale);

- starea logică sau diferența de potențial a terminalelor grupate în PORTD:

PORTD = B00000000; (terminalele au starea logic „0” și nu furnizează tensiune);

PORTD = B11111111; (terminalele au starea logic „1” și furnizează tensiunea de alimentare);

#### **V. ETAPE DE IMPLEMENTARE:**

Pentru a evidenția modul de programare în limbaj hibrid de asamblare și sintaxă a microcontrolerului ATmega 328P, vor fi utilizată următoarele componente:

- placă pentru testare rapidă a circuitelor electronice (Wisher WBU-502L);
- microcontroler ATmega 328P;
- diode electro-luminiscente;

<https://epe.utcluj.ro/index.php/sisteme-cu-microprocesoare/>

Conf. Dr. Ing. Ioana – Cornelia Gros – [Ioana.Gros@emd.utcluj.ro](mailto:Ioana.Gros@emd.utcluj.ro)

Asist. Dr. Ing. Lucian – Nicolae Pintilie – [Lucian.Pintilie@emd.utcluj.ro](mailto:Lucian.Pintilie@emd.utcluj.ro)

- rezistențe cu valoarea de 100  $\Omega$ ;
- fire pentru conexiune rapidă compatibile cu placa de testare;
- dispozitiv programator SparkFun Tiny AVR Programmer – PGM 11801;
- oscilator pe bază de cristal de cuarț (având frecvența de 16 MHz);
- calculator gazdă având mediul Arduino IDE instalat;
- cablu adaptor sau prelungitor de la tip USB A la tip USB A;

Se va realiza montajul (Fig. 12), [16], conform diagramei ilustrată în figura 11, [15].

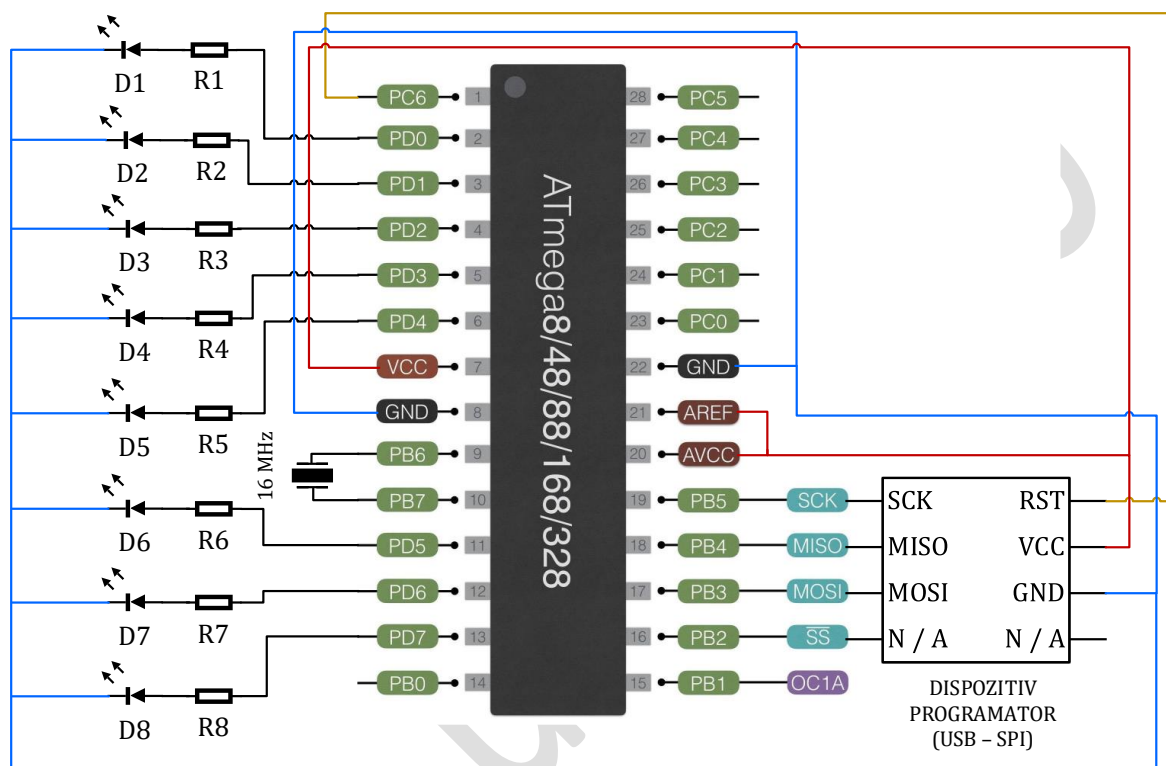


Fig. 11 – Schema electronică a montajului

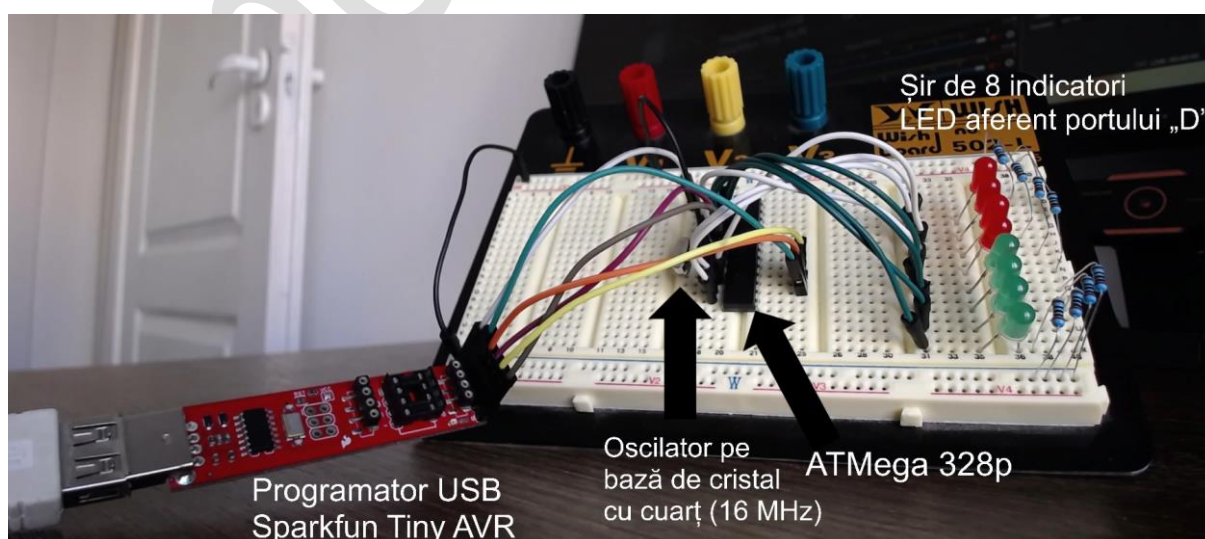


Fig. 12 – Montaj experimental

<https://epe.utcluj.ro/index.php/sisteme-cu-microprocesoare/>

Conf. Dr. Ing. Ioana – Cornelia Gros – [Ioana.Gros@emd.utcluj.ro](mailto:Ioana.Gros@emd.utcluj.ro)

Asist. Dr. Ing. Lucian – Nicolae Pintilie – [Lucian.Pintilie@emd.utcluj.ro](mailto:Lucian.Pintilie@emd.utcluj.ro)

Pentru a transfera informația numerică în memoria microcontrolerului ATmega 328P, se va utiliza dispozitivul SparkFun Tiny AVR Programmer – PGM 11801. Dispozitivul programator reprezintă un adaptor USB la SPI (eng. Serial Peripheral Interface). Pentru a instala pachetul de instrucțiuni aferent dispozitivului programator (eng. device driver) specific sistemului de operare Microsoft Windows, se va accesa pagina de la adresa următoare: <https://www.sparkfun.com/products/11801>.

De asemenea este necesar și setul de instrucțiuni pentru compilator în vederea programării microcontrolerului ATmega 328P în mediul Arduino IDE, anume „ATmega Breadboard”. Acest pachet de instrucțiuni poate fi regăsit la următoarea adresa: <https://github.com/technoblogy/atmegabreadboard>. Arhiva se va decompresa în locația „C:\Users\<numele\_utilizatorului>\Documents\Arduino\hardware”. Inițial directorul cu denumirea „hardware” nu există în locația indicată. Acesta se va crea în mod manual de către utilizatorul calculatorului. De asemenea, locația indicată nu există numai după prima rulare a mediului Arduino IDE. Versiunea mediului Arduino IDE utilizată pentru acest laborator este 1.8.19.

În urma pregătirii tuturor componentelor necesare în vederea programării microcontrolerului ATmega 328P, se vor parcurge următoarele etape:

A. Se va crea un proiect nou în cadrul mediului Arduino IDE 1.8.19, din meniul „File”, cu opțiunea „New” (Fig. 13).

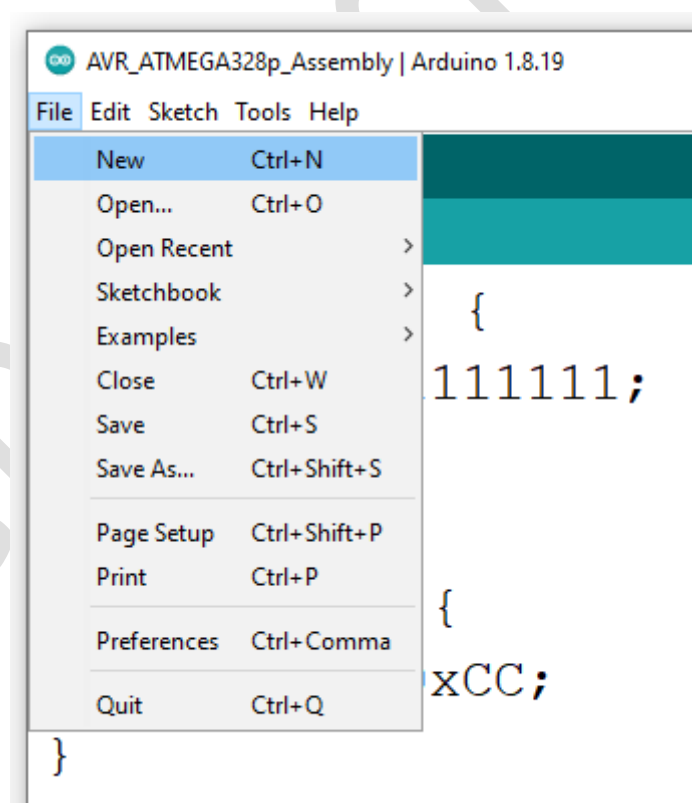


Fig. 13 – Crearea unui proiect nou în mediul Arduino IDE

B. Se va salva proiectul nou început într-o locație accesibilă, prin intermediul meniului „File” și a opțiunii „Save As...” (Fig. 14).

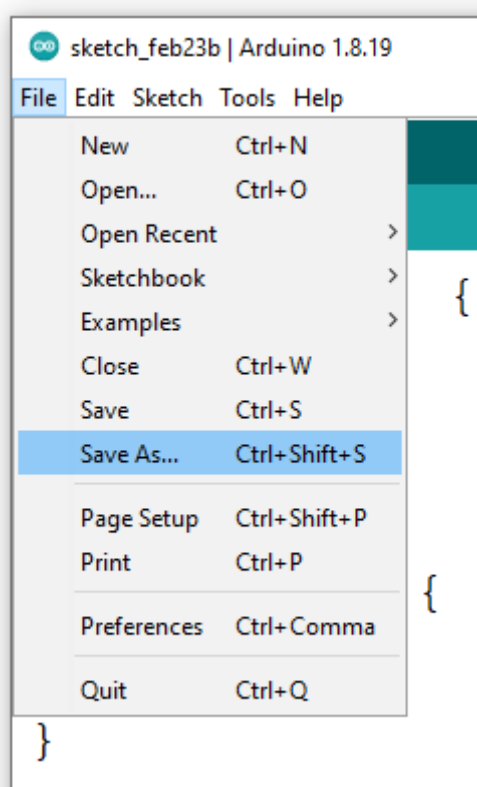


Fig. 14 – Salvarea proiectului nou creat

C. Se va alege tipul de microcontroler care urmează a fi programat cu ajutorul mediului Arduino IDE, din meniul „Tools”, opțiunea „Boards”, „ATmegaBreadboard (in sketchbook)”, „ATmega328” (Fig. 15).

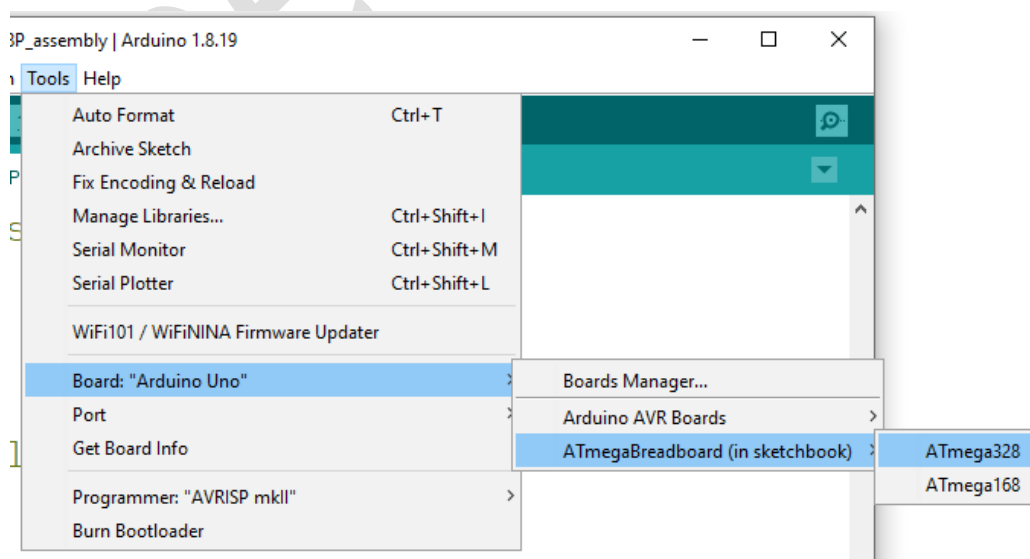


Fig. 15 – Alegerea tipului de microcontroler care urmează să fie programat

<https://epe.utcluj.ro/index.php/sisteme-cu-microprocesoare/>

Conf. Dr. Ing. Ioana – Cornelia Gros – [Ioana.Gros@emd.utcluj.ro](mailto:Ioana.Gros@emd.utcluj.ro)

Asist. Dr. Ing. Lucian – Nicolae Pintilie – [Lucian.Pintilie@emd.utcluj.ro](mailto:Lucian.Pintilie@emd.utcluj.ro)



D. Se va alege varianta constructivă a microcontrolerului, tot din meniul „Tools”, opțiunea „Chip”, „ATmega328P” (Fig. 16).

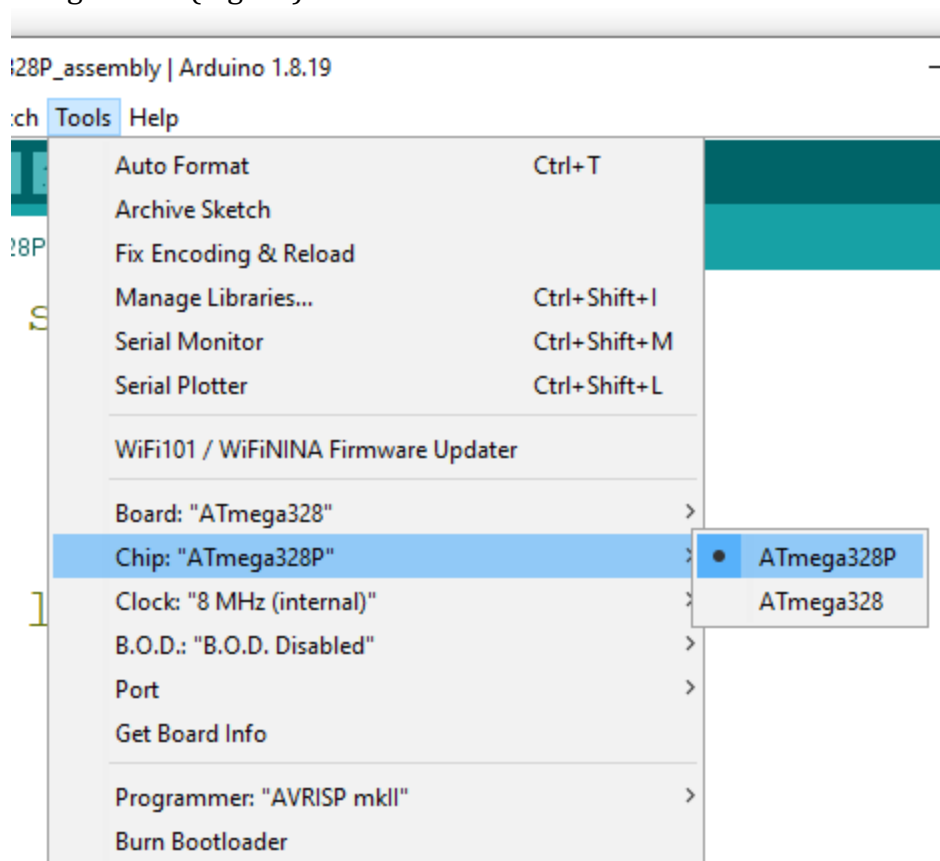


Fig. 16 – Alegerea variantei constructive a microcontrolerului care urmează să fie programat

E. Se va alege frecvența oscilatorului pe bază de cuarț atașat la terminalele „XTAL1” și „XTAL2” ale microcontrolerului ATmega 328P, tot din meniul „Tools” opțiunea „Clock”, „16 MHz (external)” (Fig. 17).

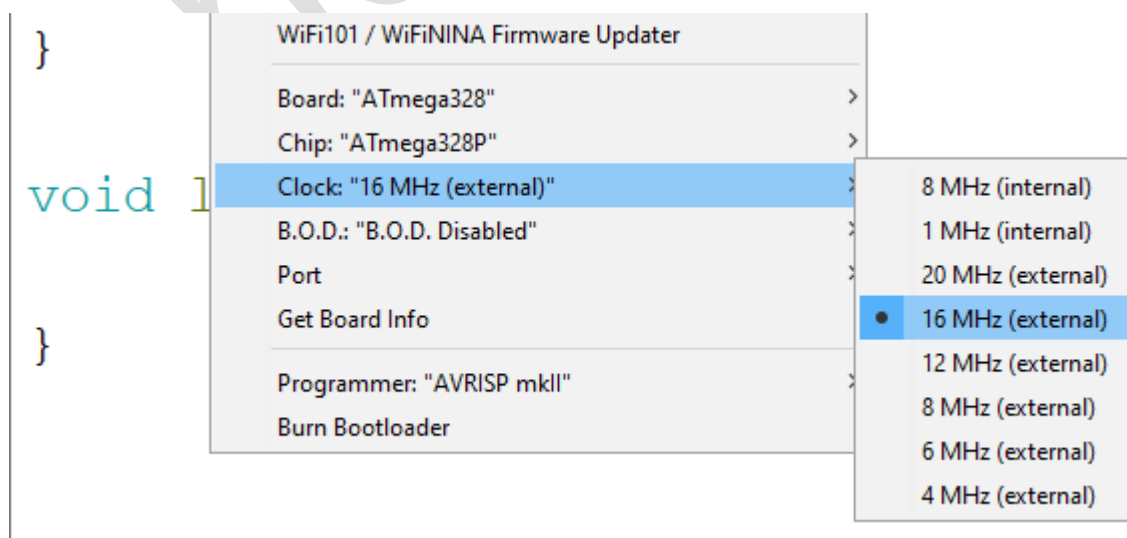


Fig. 17 – Alegerea frecvenței oscilatorului pe bază de cuarț atașat la microcontroler

F. Se va alege de asemenea, nivelul critic al tensiunii de alimentare, tot din meniul „Tools”, opțiunea „B.O.D.”, „B.O.D. Disabled” (eng. Brown Out Detection = determinarea nivelului minim sau critic al tensiunii de alimentare) (Fig. 18).



Fig. 18 – Alegerea nivelului minim al tensiunii de alimentare

G. Se va alege dispozitivul programator tot din meniul „Tools”, opțiunea „Programmer”, „USBtinyISP” (eng. In – System Programmer – ISP) (Fig. 19).

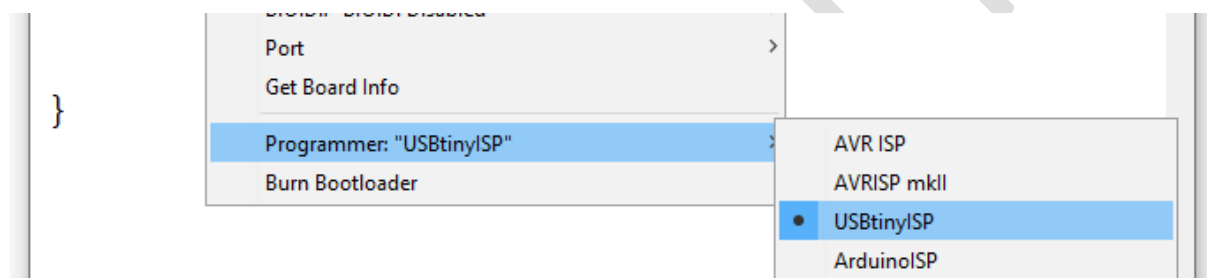


Fig. 19 – Alegerea dispozitivului programator

H. Pentru a încheia etapa de parametrizare inițială se va alege opțiunea „Burn Bootloader” (încărcare secvență de cod pentru inițializare), din meniul „Tools” (Fig. 20).

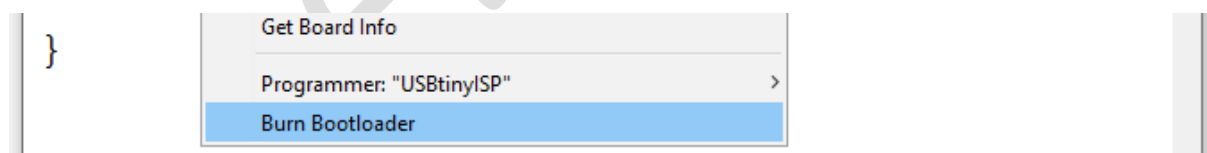


Fig. 20 – Încărcarea secvenței de cod pentru inițializarea codului program din memorie

În urma parcurgerii tuturor etapelor de configurare a microcontrolerului și a dispozitivului programator, se va trece la etapa de elaborare a codului program. Pentru a studia modul de manipulare directă a registrelor de memorie alocate intrărilor și ieșirilor digitale de uz general grupate în „PORTD”, se propune implementare a trei aplicații:

- aplicația specifică studierii circuitelor logice combinaționale (statice);
- aplicația specifică studierii circuitelor logice secvențiale (dinamice);
- aplicația specifică instrucțiunilor iterative (numărător binar);

#### OBSERVAȚII:

- structura „void setup()” reprezintă o secvență de inițializare care se va executa o singură dată la alimentarea cu tensiune sau la apăsarea butonului „RESET”;
- structura „void loop()” reprezintă o secvență repetitivă la infinit și este echivalentă cu expresia în sintaxă C standard „while(1)” sau „while(true)”;

#### APLICAȚIA 1:

Se va implementa următorul cod program:

```
void setup() {
    DDRD = B11111111;
}
void loop() {
    PORTD = B00001111;
}
```

În urma implementării codului program în fereastra de tip editor de text, din meniul „Sketch” se va alege opțiunea „Upload Using Programmer” (încărcarea codului program în memoria microcontrolerului cu ajutorul dispozitivului programator) (Fig. 21).

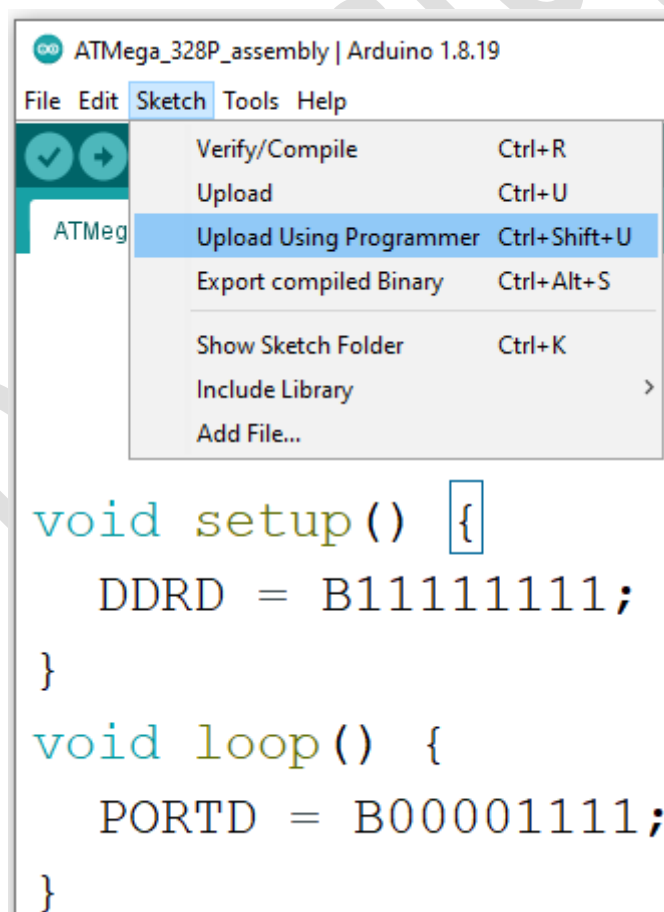


Fig. 21 – Încărcarea codului program în memoria microcontrolerului cu ajutorul dispozitivului programator

<https://epe.utcluj.ro/index.php/sisteme-cu-microprocesoare/>

Conf. Dr. Ing. Ioana – Cornelia Gros – [Ioana.Gros@emd.utcluj.ro](mailto:Ioana.Gros@emd.utcluj.ro)

Asist. Dr. Ing. Lucian – Nicolae Pintilie – [Lucian.Pintilie@emd.utcluj.ro](mailto:Lucian.Pintilie@emd.utcluj.ro)

În urma încărcării codului program în memoria microcontrolerului se va observa evoluția stării logice a grupării de terminale „PORTD” prin intermediul șirului de diode LED indicatoare atașate la microcontroler. Starea logică a grupării PORTD nu se va modifica în timp până la introducerea unei noi combinații.

Se vor introduce diverse valori numerice atât în format binar (B\*\*\*\*\*), cât și hexazecimal (0x\*\*) în registrul „PORTD”.

#### APLICAȚIA 2:

Se va implementa următorul cod program:

```
void setup() {  
    DDRD = B11111111;  
}  
void loop() {  
    PORTD = B10101010;  
    delay(100);  
    PORTD = ~B10101010;  
    delay(100);  
}
```

Spre deosebire de aplicația anterioară, în acest caz, avem declarat și parametrul „timp de întârziere sau așteptare” într-o anumită stare (eng. delay). Timpul exprimat cu ajutorul funcției „delay()” este de ordinul milisecundelor. Totodată, în cazul unui circuit logic secvențial se va constata faptul că, există cel puțin două stări logice distincte care se manifestă la un anumit interval de timp. A doua stare logică a registrului „PORTD” a fost impusă prin aplicarea operatorului de inversare sau negare logică „~”.

#### APLICAȚIA 3:

Se va implementa următorul cod program:

```
void setup() {  
    DDRD = B11111111;  
}  
void loop() {  
    for(byte i = 0; i <= 0xFF; i++) {  
        PORTD = i;  
        delay(1000);  
    }  
}
```

În cadrul ultimei aplicații a fost implementat un cod program numărător, care parcurge de la 0 la 255 toate valorile hexazecimale, urmând ca la ieșirile digitale grupate în registrul „PORTD” să fie transferate toate combinațiile în binar ale valorilor rezultante, la un interval de o secundă.

## VI. CONCLUZIE:

Manipularea în mod direct a registrelor de memorie în cadrul unui microcontroler reprezintă o soluție viabilă atât din punct de vedere al optimizării modului de execuție al unei aplicații cât și din punct de vedere al spațiului ocupat de aceasta în memoria sistemului de calcul dedicat implementării. Această metodă permite de asemenea, deblocarea unor funcții speciale la nivel de microcontroler (ex. modificarea frecvenței semnalului modulat în lățime – eng. PWM).

## VII. BIBLIOGRAFIE:

1. Atmel Corporation © 2015, Rev.: 7810D – AVR – 01 / 15 – „ATmega328P datasheet - 8-bit AVR Microcontroller with 32K Bytes In-System Programmable Flash”;
2. Conf. Dr. Ing. Ioana - Cornelia Gros, Asist. Dr. Ing. Lucian - Nicolae Pintilie, Prof. Dr. Ing. Teodor Crișan Pană – „SISTEME EMBEDDED ÎN INGINERIE ELECTRICĂ - GHID DE APLICAȚII”, Editura UTPRESS, Cluj-Napoca, 2020, ISBN 978-606-737-431-5, recenzie: Conf. Dr. Ing. Ioan Incze Iov, Șl. Dr. Ing. Călin Cenan;
3. MicroChip Technology Inc. © 2004 – 2014 – „PIC 10F200 / 202 / 204 / 206 datasheet”;
4. Infineon Technologies AG, 81726 Munich, Germany © 2017 – „XMC1400 AA-Step, Microcontroller Series for Industrial Applications, the XMC1000 Family with ARM® Cortex® - M0 32-bit processor core” – Datasheet;
5. STMicroelectronics © 2020, DS6329 Rev. 18 – „STM32F205xx, STM32F207xx – datasheet”, „ARM®-based 32-bit MCU, 150 DMIPs, up to 1 MB Flash / 128 + 4KB RAM, USB OTG HS / FS, Ethernet, 17 TIMs, 3 ADCs, 15 comm, interfaces and camera;
6. MicroChip Technology Inc. © 2004 – 2014 – „Explorer 16/32 Development Board (datasheet) - A Perfect Platform for Discovering the Full Capabilities of PIC® MCUs and dsPIC® DSCs”;
7. Infineon Technologies AG, 81726 Munich, Germany © 2017 – „AURIX™ TC275 lite Kit - Evaluation Board datasheet”;
8. STMicroelectronics © 2020 – „STM32 Nucleo-64 boards (MB1136) – user manual”;
9. Arduino® – „Arduino UNO R3 - Product Reference Manual”;
10. Protostack.com – „ATMEGA328P-PU Atmel 8 Bit 32K AVR Microcontroller”;
11. Arduino® forum – „Reducing power supply noise”;
12. Technoblogy.com – „Using an ATmega328 without a crystal”;

<https://epe.utcluj.ro/index.php/sisteme-cu-microprocesoare/>

Conf. Dr. Ing. Ioana – Cornelia Gros – [Ioana.Gros@emd.utcluj.ro](mailto:Ioana.Gros@emd.utcluj.ro)

Asist. Dr. Ing. Lucian – Nicolae Pintilie – [Lucian.Pintilie@emd.utcluj.ro](mailto:Lucian.Pintilie@emd.utcluj.ro)



13. Jet Support Service – „Arduino Parts DIY Arduino Guide Microcontroller Tutorials”;
14. epe.utcluj.ro – Prezentare laborator nr. II la disciplina Sisteme cu Microprocesoare – „Utilizarea platformei de dezvoltare Arduino”;
15. github.com – „MiniCore - DIP-28 package ATmega8 / 48 / 88 / 168 / 328 pinout”
16. epe.utcluj.ro – Material video demonstrativ la disciplina Sisteme cu Microprocesoare - „Programarea microcontrolerului ATmega 328p în limbaj hibrid sintaxă - asamblare”;