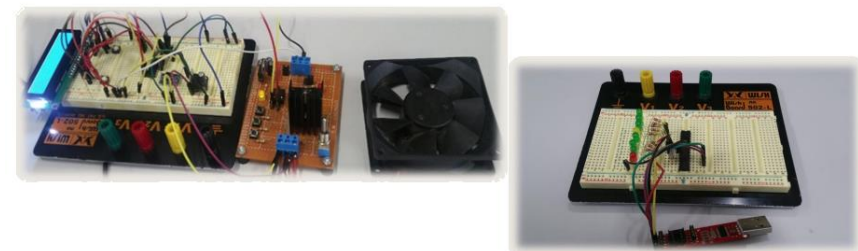


## **SISTEME CU MICROPROCESOARE**

### **Ședința de laborator nr. 1**

#### Noțiuni introductive Microcontrolerul ATMega 328P

SISTEME EMBEDDED ÎN INGINERIE ELECTRICĂ  
GHID DE APLICAȚII



<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)

**epe.utcluj.ro**



# **Cuprins**

- 1. Scopul lucrării**
- 2. Noțiuni introductive**
- 3. Integrarea microcontrolerului ATmega 328P într-o aplicație**
- 4. Metode de programare a microcontrolerului ATmega 328P**
- 5. Etape de implementare a codului program la nivel de microcontroler**
- 6. Concluzii**
- 7. Bibliografie**

# 1. 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 <sup>[1]</sup>

## 2. Noțiuni introductive

**Microcontrolerul** reprezintă un sistem de calcul micro-programabil specializat care poate efectua operații logice și aritmetice pe baza semnalelor electrice achiziționate sau furnizate la nivelul etajelor de circuit periferice [2].

Ca și exemple de **etaje de circuit periferice specializate** la nivel de microcontroler pot fi considerate următoarele:

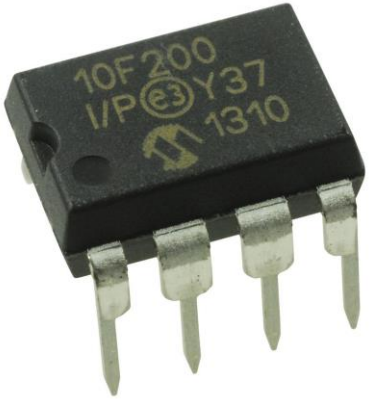
- Intrări și ieșiri de uz general (eng. General Purpose Input or Output - GPIO)
- Convertoare analog – digitale (eng. Analog to Digital Converter – ADC)
- Interfețe fizice de comunicare (ex. UART, SPI, I<sup>2</sup>C, Parallel Port etc.)

## 2. Noțiuni introductive

În practică există mai multe variante constructive de microcontrolere, de la diverși producători precum:

- ✓ MicroChip PIC 10F200 [3]
- ✓ Infineon XMC1400 [4]
- ✓ STMicroelectronics STM32 F205 [5]
- ✓ **Atmel, AVR, MicroChip ATMega 328P** [1]

## 2. Noțiuni introductive



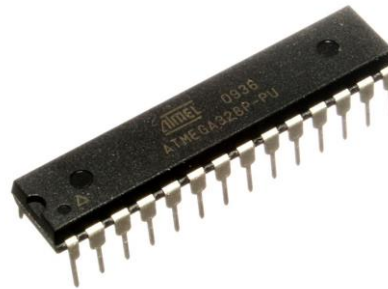
MicroChip PIC 10F200



Infineon XMC1400



STMicroelectronics STM32 F205



Atmel, AVR, MicroChip ATmega 328P

<https://upload.wikimedia.org/wikipedia/commons/thumb/0/0c/ATMEGA328P-PU.jpg/1200px-ATMEGA328P-PU.jpg>

[https://canada.newark.com/productimages/large/en\\_US/96K3749-40.jpg](https://canada.newark.com/productimages/large/en_US/96K3749-40.jpg)

[https://www.infineon.com/export/sites/default/\\_images/product/microcontroller/XMC/lowres-XMC1400-Family.jpg\\_811405727.jpg](https://www.infineon.com/export/sites/default/_images/product/microcontroller/XMC/lowres-XMC1400-Family.jpg_811405727.jpg)

<https://www.ic-crack.com/wp-content/uploads/2022/09/STM32F205VBT6.jpg>

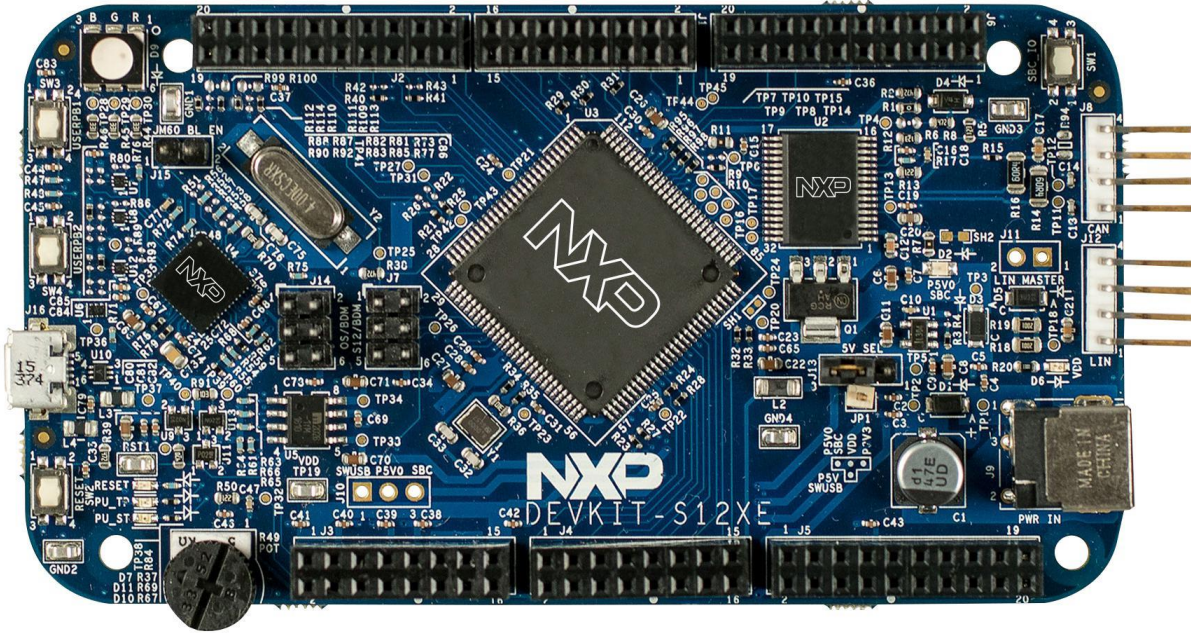
### **3. Integrarea microcontrollerului ATmega 328P într-o aplicație**

Există două tehnologii de integrare sau încorporare a dispozitivului central de procesare (în cazul de față al microcontrollerului):

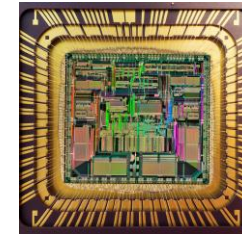
- ✓ Tehnologia (eng.) „On board” (integrarea modulelor pe un singur cablaj)
- ✓ Tehnologia (eng.) „System On a Chip” (integrarea modulelor într-o capsulă)



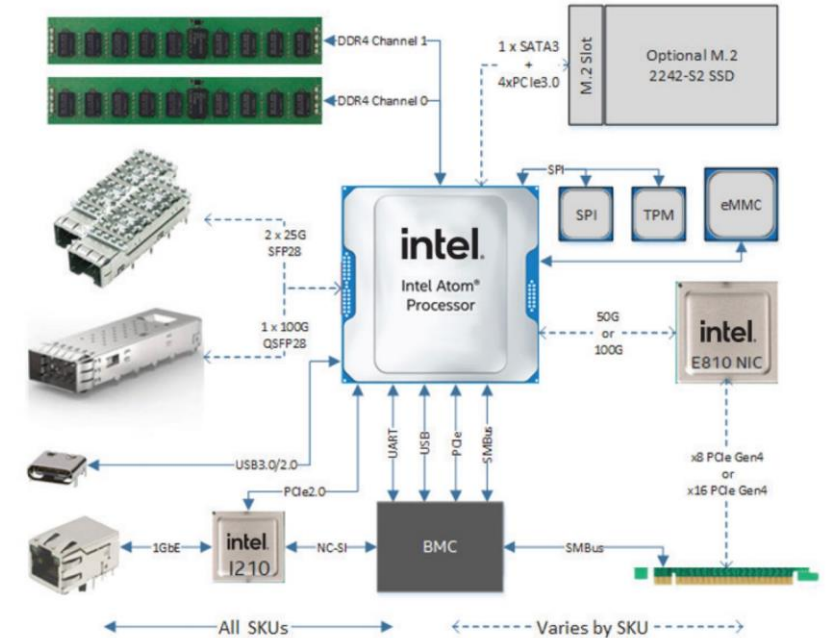
### 3. Integrarea microcontrolerului ATmega 328P într-o aplicație



Tehnologia (eng.) „On board”



Tehnologia (eng.) „System On a Chip”



<https://www.servethehome.com/wp-content/uploads/2022/06/Intel-NetSec-Accelerator-Reference-Design-Architecture-800x612.jpg>

<https://opengameart.org/sites/default/files/Motorola68040die-Farwestern-CCBYSA3.jpg>

[https://ro.farnell.com/productimages/large/en\\_GB/2820405-40.jpg](https://ro.farnell.com/productimages/large/en_GB/2820405-40.jpg)

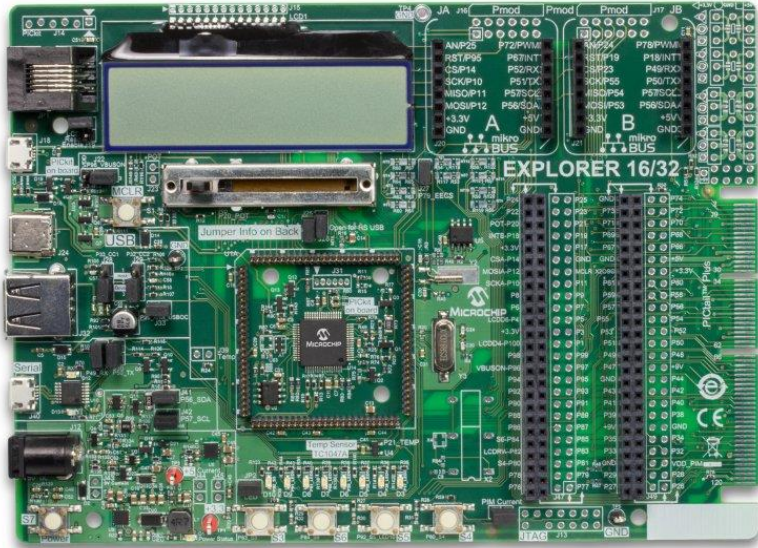


### 3. Integrarea microcontrolerului ATMega 328P într-o aplicație

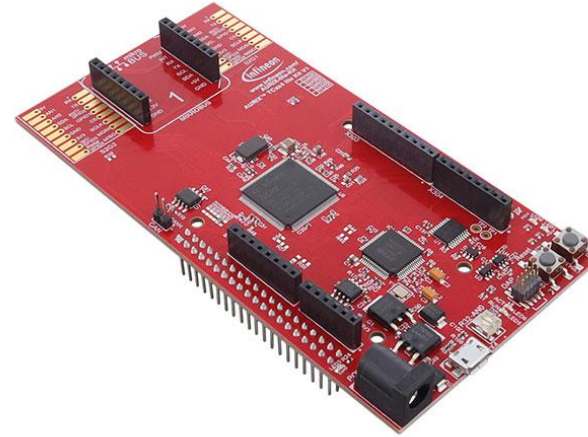
Pe baza tehnologiei „**On Board**” au fost concepute mai multe așa – zise **platforme de dezvoltare** în jurul dispozitivului central de calcul de tip microcontroler. Câteva exemple ar fi:

- ✓ MicroChip Explorer 16/32 Development Board [6]
- ✓ Infineon AURIX TC275 ARDSBT0B01 [7]
- ✓ STMicroelectronics NUCLEO - L010RB [8]
- ✓ Arduino UNO [9]

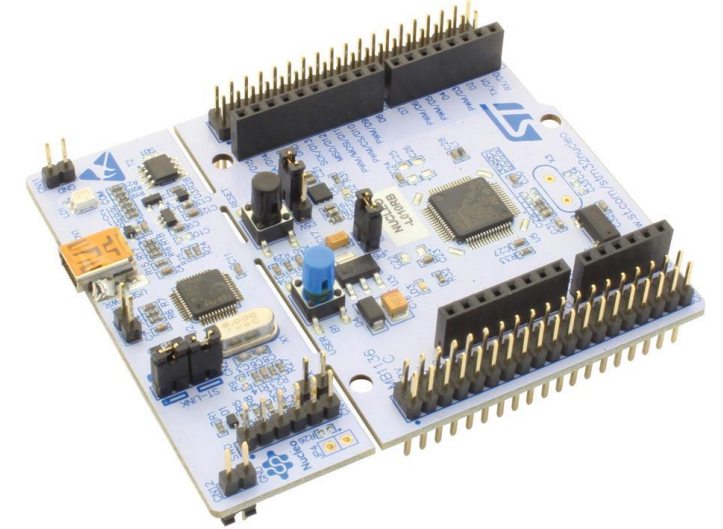
# 3. Integrarea microcontrolerului ATmega 328P într-o aplicație



MicroChip Explorer 16 / 32



Infineon AURIX TC275  
ARDSBT0B01



STMicroelectronics  
NUCLEO - L010RB



Arduino UNO

[https://m.media-amazon.com/images/I/61ac06V0ljL.\\_AC\\_UF1000,1000\\_QL80\\_.jpg](https://m.media-amazon.com/images/I/61ac06V0ljL._AC_UF1000,1000_QL80_.jpg)

[https://ro.farnell.com/productimages/large/en\\_GB/2845534-40.jpg](https://ro.farnell.com/productimages/large/en_GB/2845534-40.jpg)

<https://mm.digikey.com/Volume0/opasdata/d220001/medias/images/4089/KITA2GTC334LITETOBO1.jpg>

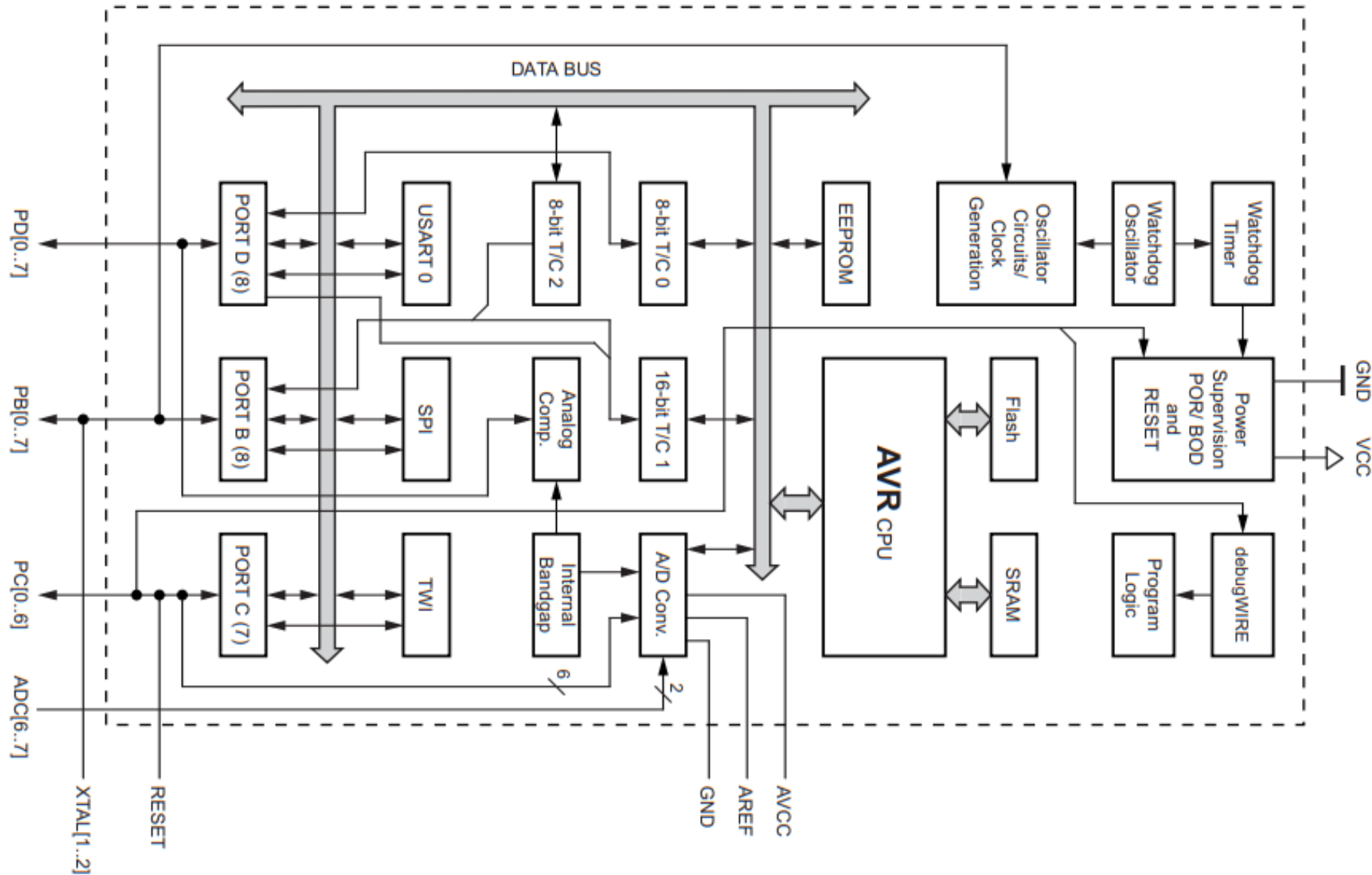
<https://www.microchip.com/content/dam/mchp/mrt-dam/devtools/1488-explorer16-32-developmentboard.jpg>

### 3. Integrarea microcontrolerului ATmega 328P într-o aplicație

Microcontrolerele, fiind sisteme de calcul micro-programabile cu un grad mare de integrare, prezintă în mod preponderent o construcție de tip „**System on a chip**”, fără a necesita introducerea modulelor externe de memorie sau introducerea unui modul de procesare extern. Toate componentele specifice funcționării unui sistem de calcul (ex. memorie ROM, RAM sau unitate centrală de procesare) sunt **integrate** în cadrul arhitecturii interne.

Un exemplu de astfel de sistem de calcul specializat este microcontrolerul ATmega 328P. Acesta poate fi integrat în cadrul unei platforme de dezvoltare (Arduino UNO), sau poate să funcționeze în componența unor aparate sau dispozitive precum electrocasnice, mașini – unelte sau orice produse electronice de larg consum.

### 3. Integrarea microcontrolerului ATmega 328P într-o aplicație



Arhitectura internă a microcontrolerului ATmega 328P



# 3. Integrarea microcontrolerului ATmega 328P într-o aplicație

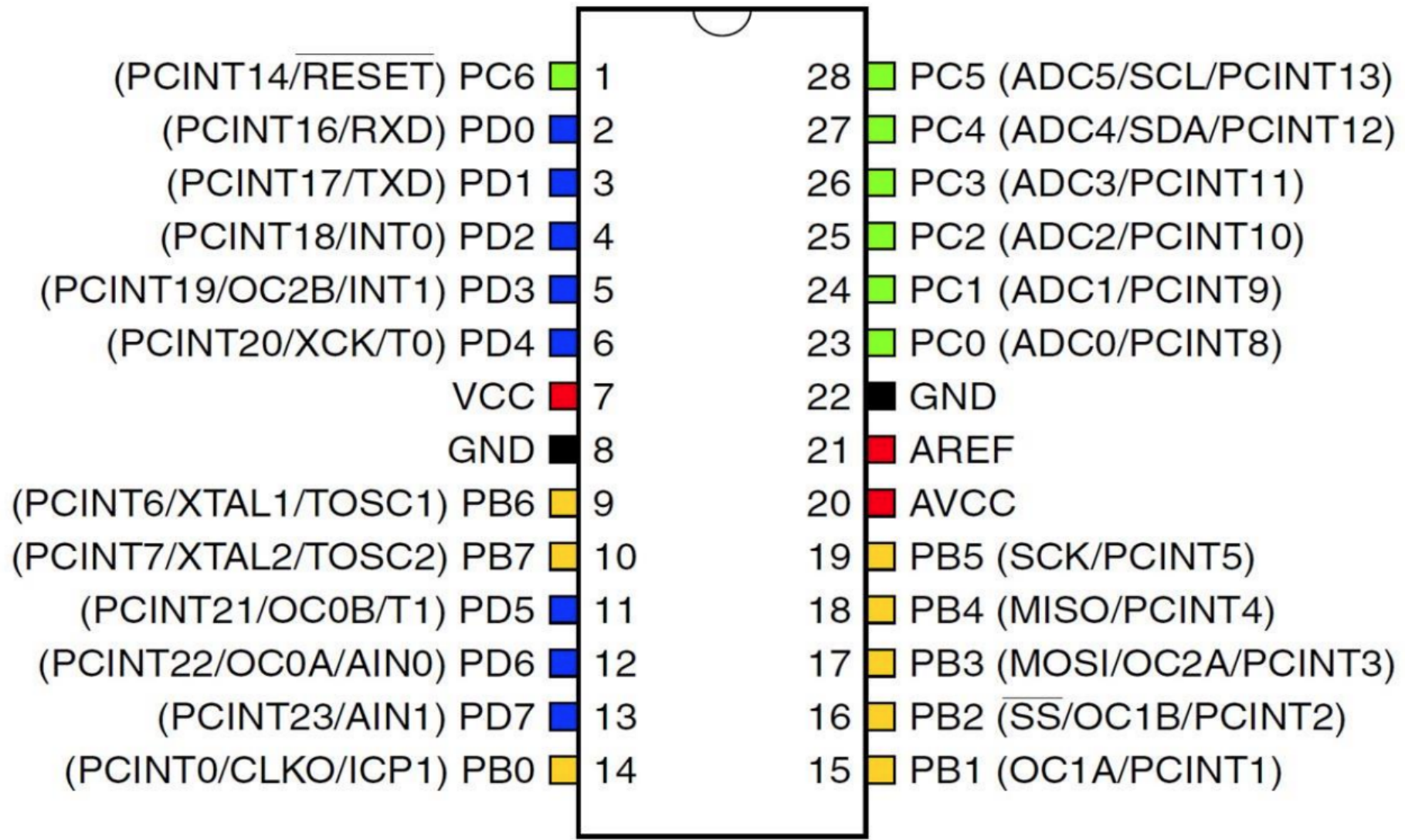


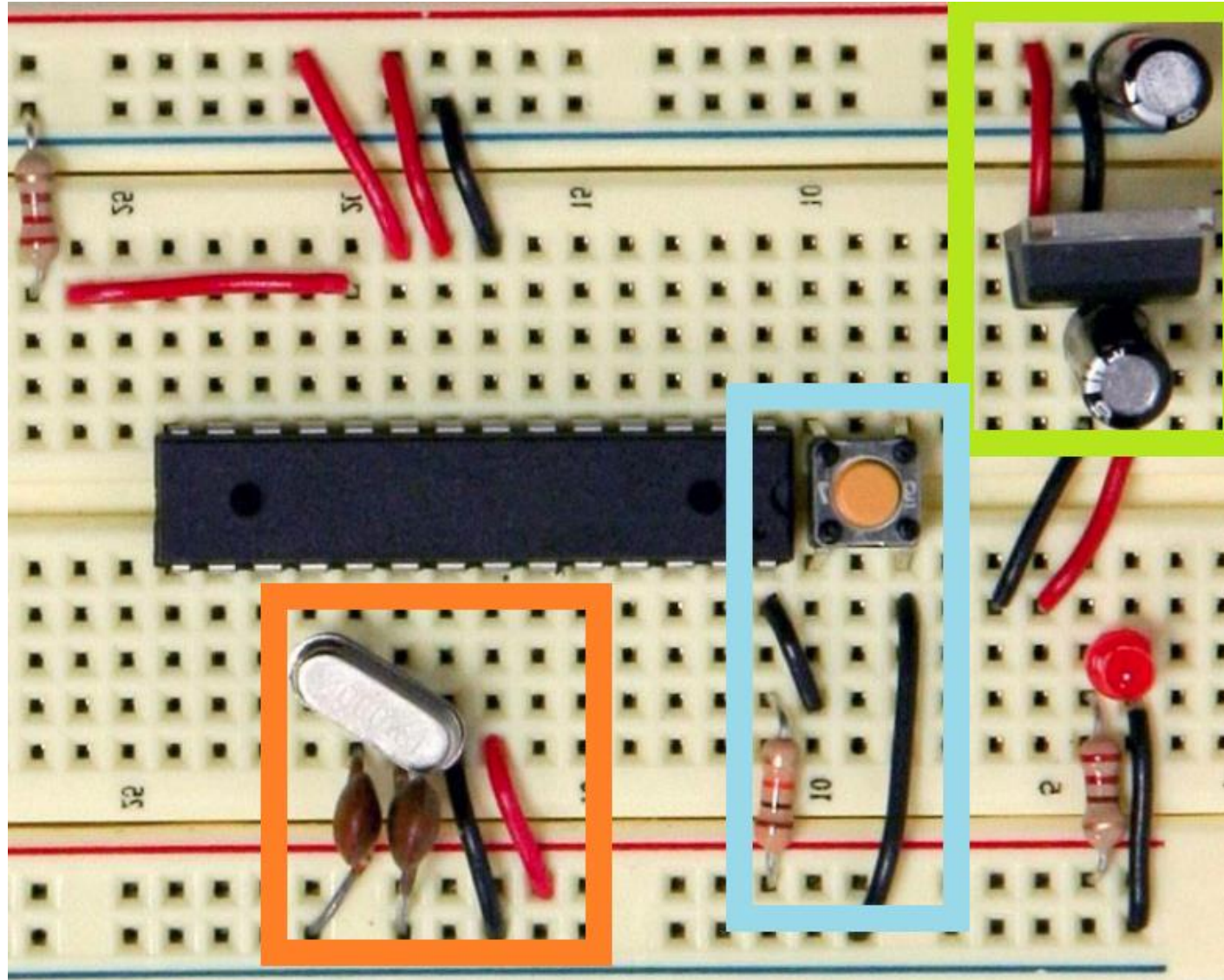
Diagrama de distribuție a terminalelor și funcțiile alocate acestora la nivel de microcontroler

### 3. Integrarea microcontrolerului ATMega 328P într-o aplicație

Pentru a integra microcontrolerul ATMega 328P într-o anumită aplicație, este necesară introducerea a cel puțin **trei etaje de circuit fundamentale:**

- ✓ 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 (marcaj albastru)

### 3. Integrarea microcontrolerului ATmega 328P într-o aplicație



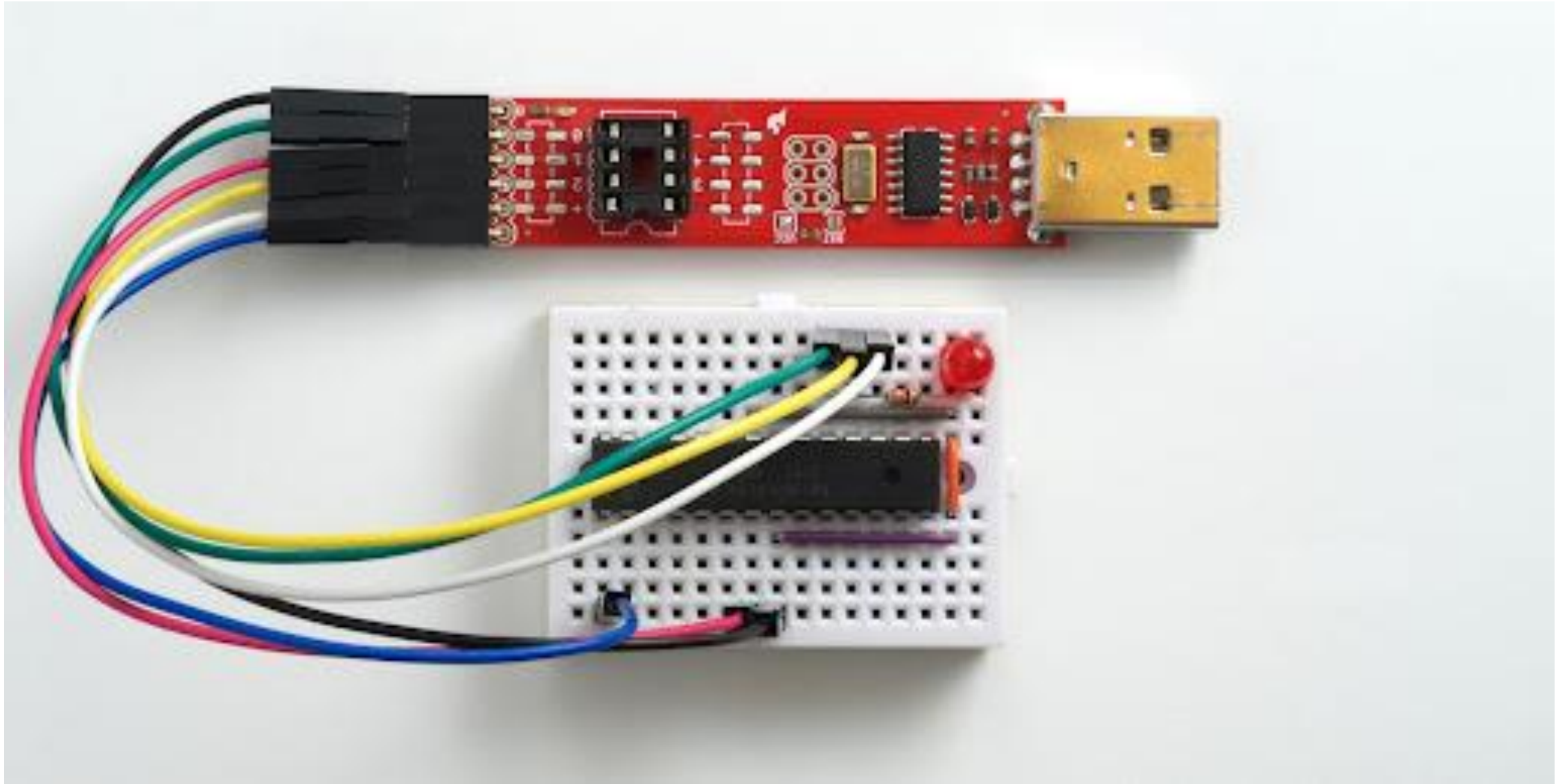
Integrarea microcontrolerului ATmega 328P într-o aplicație



### 3. Integrarea microcontrollerului ATmega 328P într-o aplicație

- ✓ În vederea transferării inițiale a informației numerice în memoria microcontrollerului 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 microcontrollerului.
- ✓ Procedura utilizată în această situație, reprezintă un protocol de comunicare direct (între calculatorul gazdă și microcontroller) prin **interfață USB – SPI** (eng. Serial to Peripheral Interface).

### 3. Integrarea microcontrolerului ATMega 328P într-o aplicație

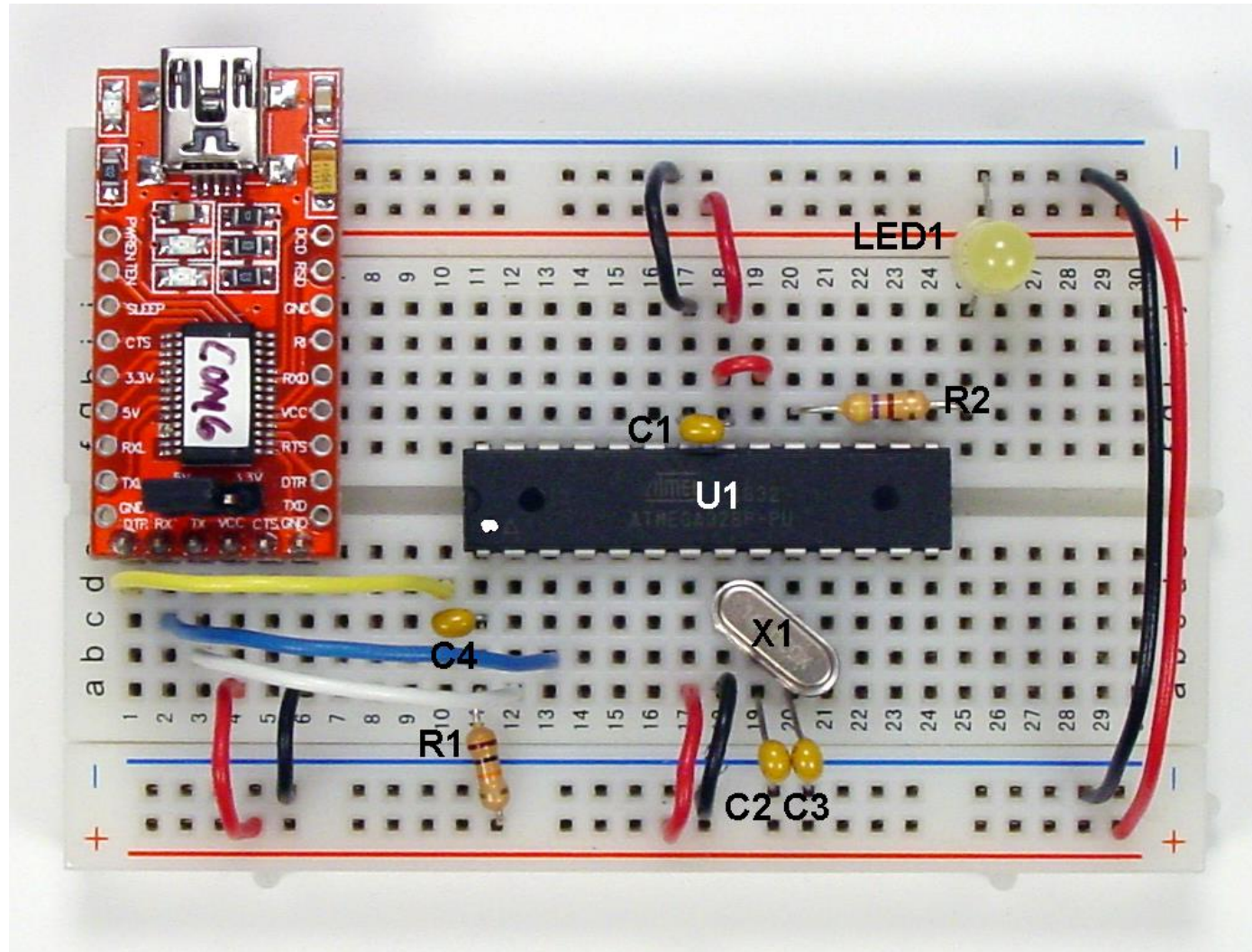


Atașarea dispozitivului programator USB – SPI la microcontrolerul ATMega 328P

### 3. Integrarea microcontrolerului ATMega 328P într-o aplicație

- ✓ Există, de asemenea, posibilitatea de a programa microcontrolerul ATMega 328P și prin intermediul unui dispozitiv adaptor de tip **USB – TTL – Serial** (ex. FTDI FT232RL).
- ✓ 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 calculator.
- ✓ 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.

### 3. Integrarea microcontrolerului ATmega 328P într-o aplicație



Atașarea dispozitivului programator USB – Serial la microcontrolerul ATmega 328P

## 4. Metode de programare a microcontrolerului ATMega 328P

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

Procesul de **compilare** reprezintă „**translatarea**” codului program de la un nivel inteligibil al factorului uman înspre un nivel compatibil cu sistemul de calcul dat.



# 4. Metode de programare a microcontrolerului ATMega 328P



## 4. Metode de programare a microcontrolerului ATMega 328P

- ✓ 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 microcontrolerului ATMega 328P este organizată **tabelar** în **zone** numite **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 microcontrolerului



## 4. Metode de programare a microcontrolerului ATMega 328P

Există de asemenea, la nivelul memoriei microcontrolerului ATMega 328P, două zone speciale (registre) rezervate **terminalelor specifice intrărilor și ieșirilor digitale**, anume:

- ✓ Registrul „DDRD” care controlează **modul de lucru** al unui grup de opt terminale digitale aflate la periferia carcusei microcontrolerului. Acronimul „DDRD” semnifică (eng.) „Data Direction Register of PORT D”
- ✓ Registrul „PORTD” **gestionează diferența de potențial** sau **starea logică** la nivelul celor opt terminale din grupul „PORTD” aflate la periferia carcusei microcontrolerului

## 4. Metode de programare a microcontrolerului ATMega 328P

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

Bit	7	6	5	4	3	2	1	0	
0x0A (0x2A)	<b>DDD7</b>	<b>DDD6</b>	<b>DDD5</b>	<b>DDD4</b>	<b>DDD3</b>	<b>DDD2</b>	<b>DDD1</b>	<b>DDD0</b>	<b>DDRD</b>
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	

Structura registrului pentru stabilirea modului de lucru „DDRD”

## 4. Metode de programare a microcontrolerului ATMega 328P

### 13.4.8 PORTD – The Port D Data Register

Bit	7	6	5	4	3	2	1	0	
0x0B (0x2B)	<b>PORTD7</b>	<b>PORTD6</b>	<b>PORTD5</b>	<b>PORTD4</b>	<b>PORTD3</b>	<b>PORTD2</b>	<b>PORTD1</b>	<b>PORTD0</b>	<b>PORTD</b>
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	

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

## 4. Metode de programare a microcontrolerului ATMega 328P

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

✓ Pentru stabilirea **sensului curentului** prin terminale din grupul PORTD:

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

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

✓ Pentru stabilirea **stării logice** a terminalelor din grupul PORTD:

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

```
PORTD = B11111111; // (terminalele au starea logic „1” și furnizează tensiune);
```

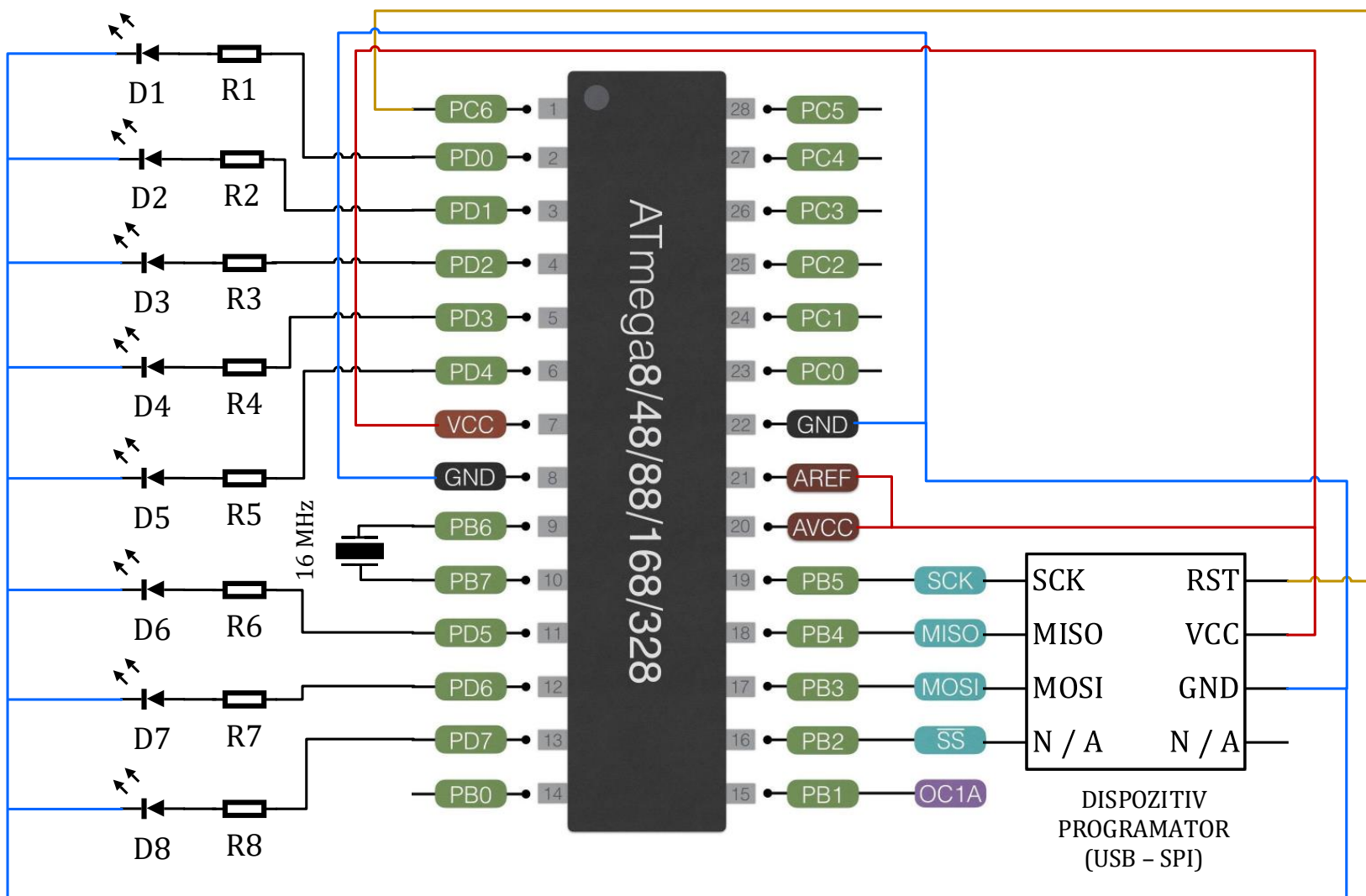
## 5. Etape de implementare a codului program la nivel de microcontroler

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
- ✓ 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 experimental conform următoarei scheme electronice

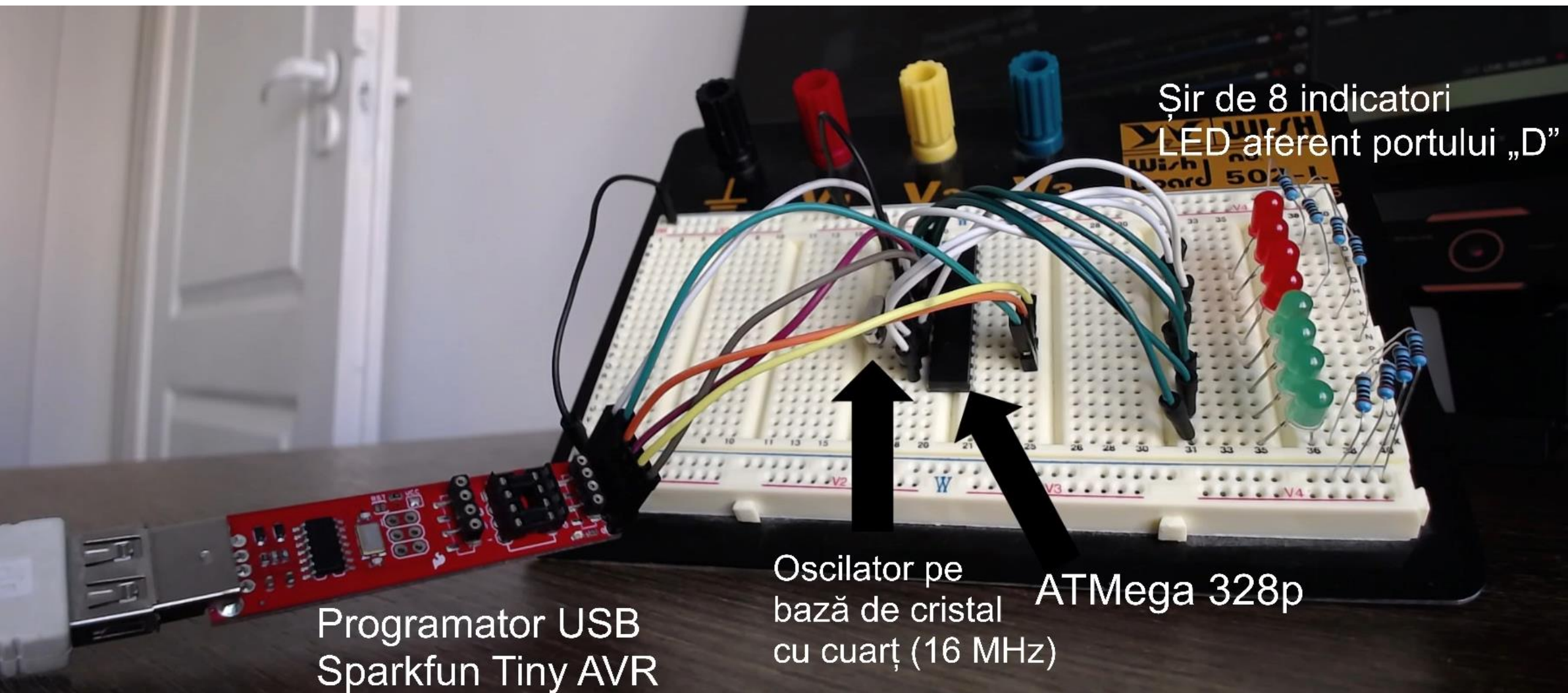
# 5. Etape de implementare a codului program la nivel de microcontroler



Schema electronică a montajului



## 5. Etape de implementare a codului program la nivel de microcontroler



Montajul experimental



## **5. Etape de implementare a codului program la nivel de microcontroler**

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

- ✓ Crearea unui proiect nou în mediul Arduino IDE
- ✓ Salvarea proiectului nou creat
- ✓ Alegerea tipului de microcontroler
- ✓ Alegerea variantei constructive
- ✓ Alegerea frecvenței oscilatorului cu cuarț atașat la microcontroler
- ✓ Alegerea nivelului critic al tensiunii de alimentare
- ✓ Alegerea tipului de dispozitiv programator
- ✓ Încărcarea secvenței de cod pentru inițializare (eng. Boot Loader)

## 5. Etape de implementare a codului program la nivel de microcontroler

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 1 specifică studierii circuitelor logice combinaționale (statice)
- ✓ Aplicația 2 specifică studierii circuitelor logice secvențiale (dinamice)
- ✓ Aplicația 3 specifică instrucțiunilor iterative (numărător binar)

## 5. Etape de implementare a codului program la nivel de microcontroler

Pentru **prima aplicație** se va implementa următorul cod program:

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

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 binare.

## 5. Etape de implementare a codului program la nivel de microcontroler

Pentru **a doua aplicație** 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), iar timpul exprimat cu ajutorul funcției **„delay()”** este de ordinul **milisecundelor**.

Totodată, se remarcă faptul că, în cazul unui **circuit logic secvențial**, există **cel puțin două stări logice distincte** care se manifestă **la un anumit interval de timp**.

## 5. Etape de implementare a codului program la nivel de microcontroler

Pentru **a treia aplicație** 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 de tip **numărător**, care parcurge de la 0x00 la 0xFF toate valorile în format **hexazecimal**, urmând ca la ieșirile digital grupate în registrul „PORTD” să fie transferate **toate combinațiile numerice** în **format binar** ale valorilor rezultante, la un interval de timp de **o secundă** pentru fiecare combinație numerică.

## 6. Concluzii

- ✓ **Manipularea în mod direct** a registrelor de memorie în cadrul unui microcontroller 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 microcontroller (ex. modificarea frecvenței semnalului modulat în lățime – eng. PWM)

# 7. 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”
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”