

Introducere Arduino

A. Scopul lucrării:

- Înțelegerea structurii unei plăci Arduino (Placă de studiu: Arduino UNO)
- Înțelegerea schemei electrice
- Înțelegerea funcționării
- Testarea unui program. Dezvoltarea, încărcarea și rularea

B. Prezentare generală:

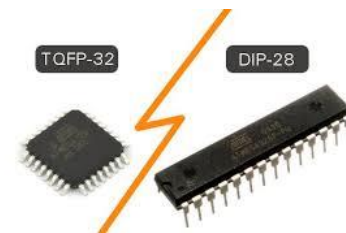
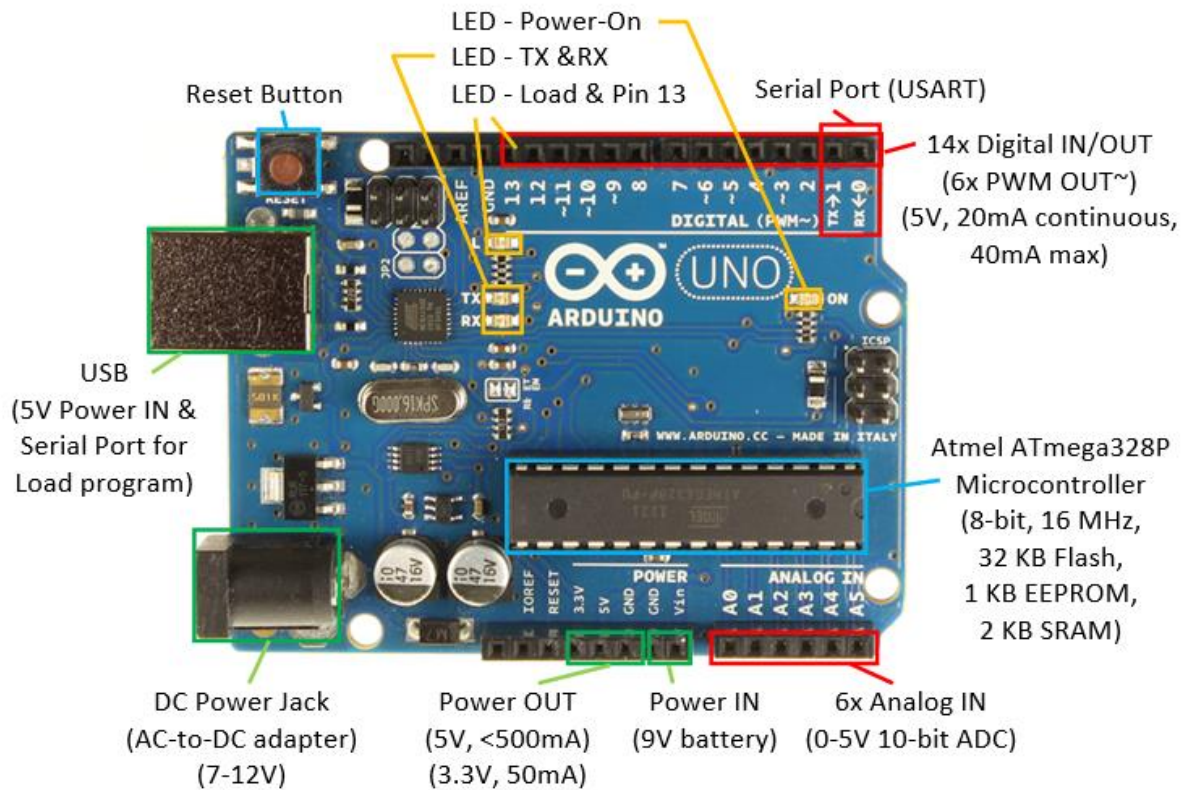
În cele ce urmează se vor prezenta caracteristicile generale ale plăcii Arduino UNO.

Arduino UNO este o placă de dezvoltare open-source realizată pe baza microcontrolerului ATmega328P. Acesta din urmă prezintă un procesor AVR cu o arhitectură de tip RISC (set restrâns de instrucțiuni). Figura 1 prezintă caracteristicile microcontrolerului:

- Memorie program de 32kb (non-volatilă)
- Viteza de 20 MIPS (20 milioane de instrucțiuni pe secundă)
- Memorie RAM (de lucru) și ROM (non-volatilă pentru stocarea de informații și după oprirea alimentării)
- Interfețe seriale de comunicare cu alte dispozitive: UART, SPI, I²C
- Numărătoare și comparatoare

Name	Value
Program Memory Type	Flash
Program Memory Size (KB)	32
CPU Speed (MIPS/DMIPS)	20
SRAM Bytes	2,048
Data EEPROM/HEF (bytes)	1024
Digital Communication Peripher...	1-UART, 2-SPI, 1-I2C
Capture/Compare/PWM Periph...	1 Input Capture, 1 CCP, 6PWM
Timers	2 x 8-bit, 1 x 16-bit
Number of Comparators	1
Temperature Range (C)	-40 to 85
Operating Voltage Range (V)	1.8 to 5.5
Pin Count	32
Low Power	Yes

C. Structura plăcii



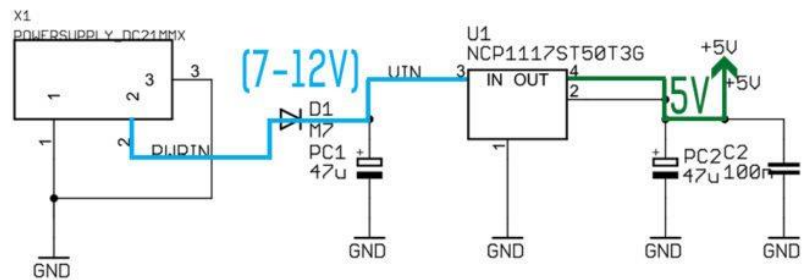
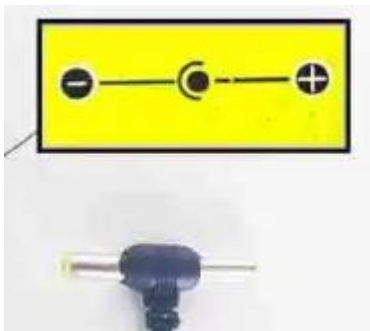
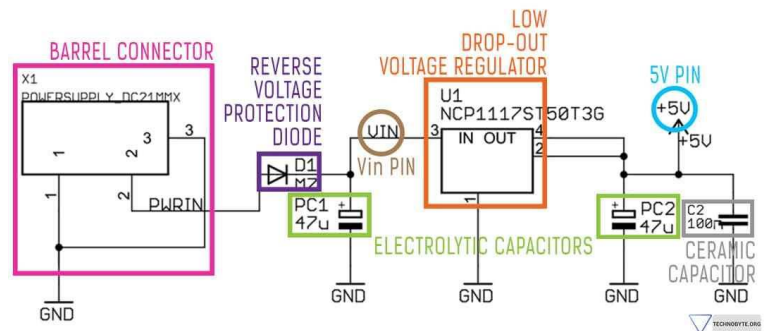
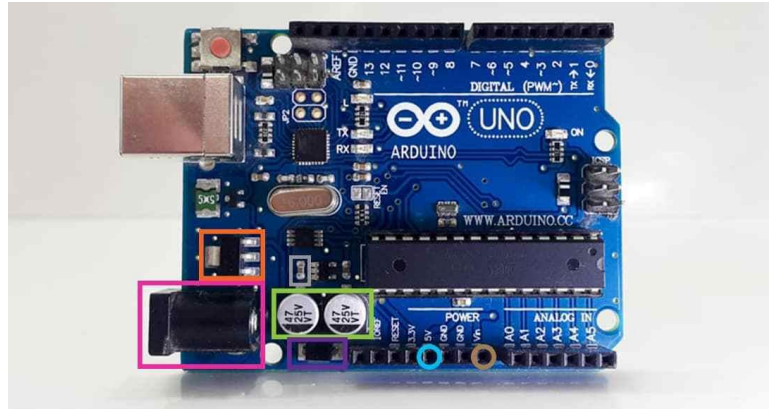
- Microcontroler (posibil să difere între diferite versiuni)
- Buton reset
- LED indicator funcționare
- LED-uri Tx (Indicator transmisie date), Rx (Indicator recepție date)
- LED Load. Se poate controla de la Pin 13
- USB (Sursă de alimentare și port serial pentru încărcarea programului)
- DC Power Jack (Mufă pentru alimentare placă)
- Pini pentru alimentare dispozitive externe (3.3V, 5V, GROUND)
- Pini pentru alimentare placă de la baterie de 9V (Vin, GROUND)
- A0-A5. 6 pini analogici. ADC (Analog to Digital Converter). Tensiunile între 0-5V aplicate la intrarea oricăruia dintre acești pini sunt citite ca valori între 0-1023 (2^{10} biți)
- 14 pini digitali IN/OUT. Pini notați cu (~) sunt capabili să genereze semnal PWM (Pulse-width modulation)
- AREF. Folosit pentru setarea unei limite superioare de referință pentru pini analogici

D. Schema electrică

Pentru analiza schemei electrice a plăcii se utilizează fișierul [Schematic Arduino UNO.pdf](#) sau [Schematic Arduino UNO.jpg](#)

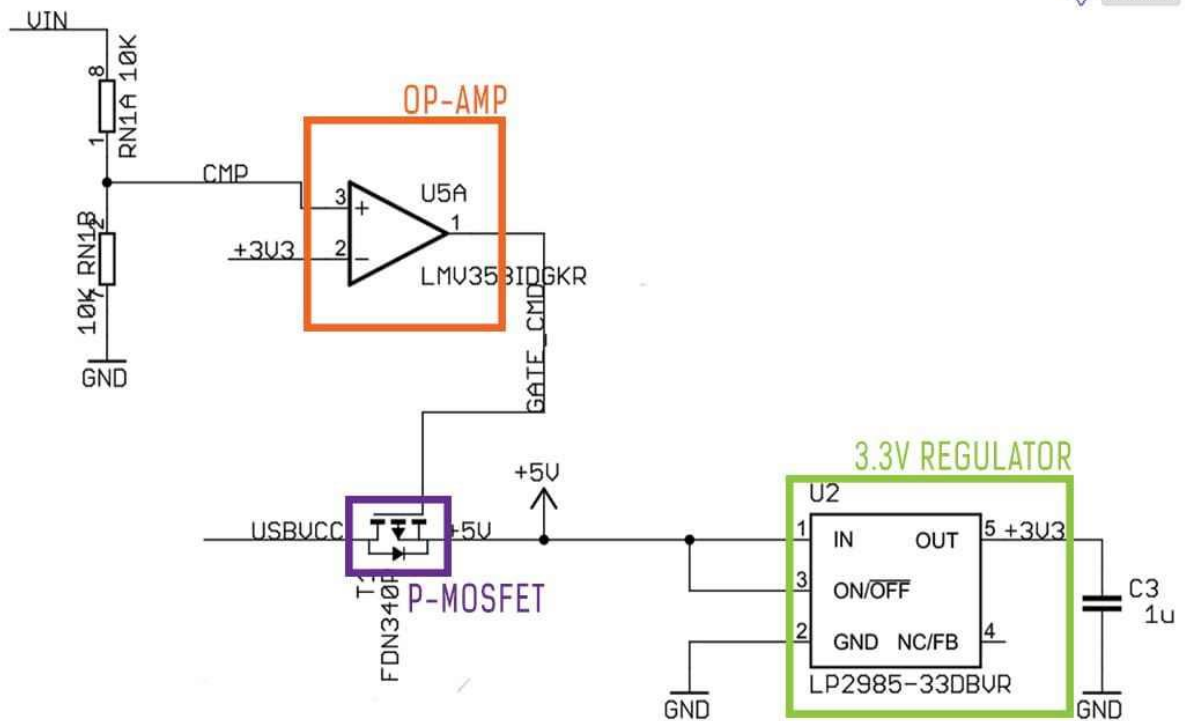
a. Alimentarea cu electricitate

Pentru alimentarea de la mufa Jack se utilizează pentru obținerea unei tensiuni constante de 5V un regulator de tensiune. Observăm de asemenea dioda de protecție. Alimentatorul trebuie să aibă în centru tensiune pozitivă iar pe suprafața exterioară tensiuni negative sau masă. În caz contrar dioda oprește trecerea unei polarități inverse. Condensatorii filtrează semnalele de curent alternativ.

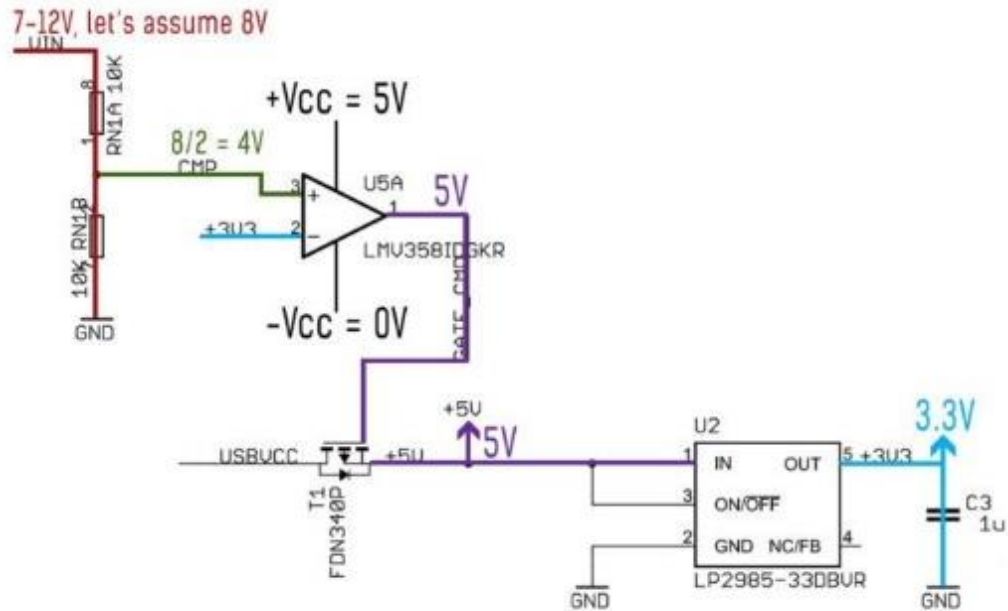


Variantă:

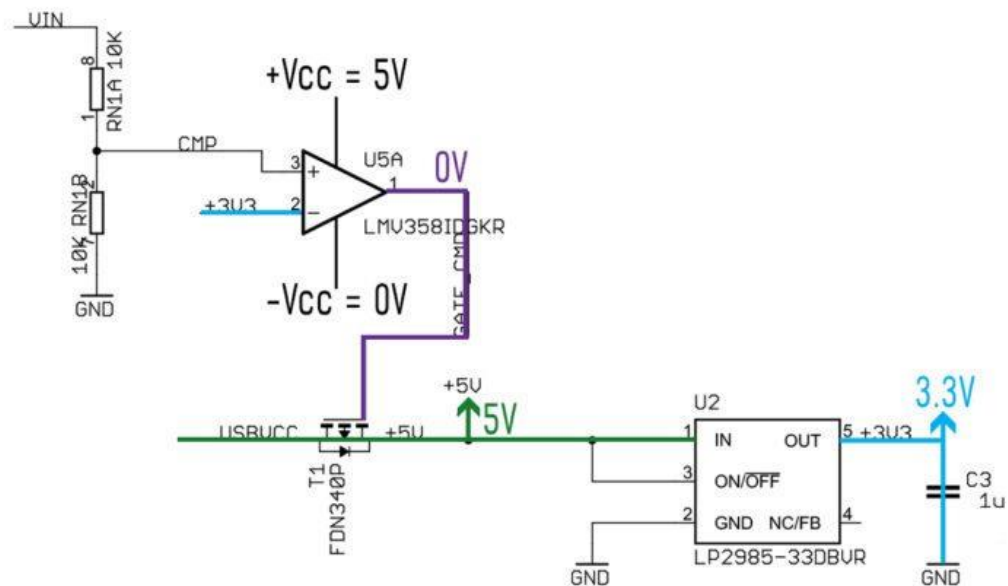
În cazul în care în timpul alimentării de la DC Power Jack se conectează placa la USB, se realizează închiderea căii pentru alimentarea de la USB și utilizarea celei de la Power Jack pentru evitarea unei supratensiuni.



- i) Comparatorul OP-AMP compară tensiunea de la Power Jack cu valoarea de 3.3V. Dacă tensiunea este mai mare atunci la ieșirea amplificatorului vor fi 5V care vor închide tranzistorul MOSFET cu canal P. Astfel este blocat curentul furnizat de USB și este utilizat în continuare cel de la Power Jack.



- ii) Dacă Mufa Power Jack nu furnizează curent atunci comparatorul va avea o tensiune de 0V la ieșire ceea ce va genera deschiderea tranzistorului și trecerea curentului de la USB către alimentarea cu 5V și mai departe la regulatorul de 3.3V.

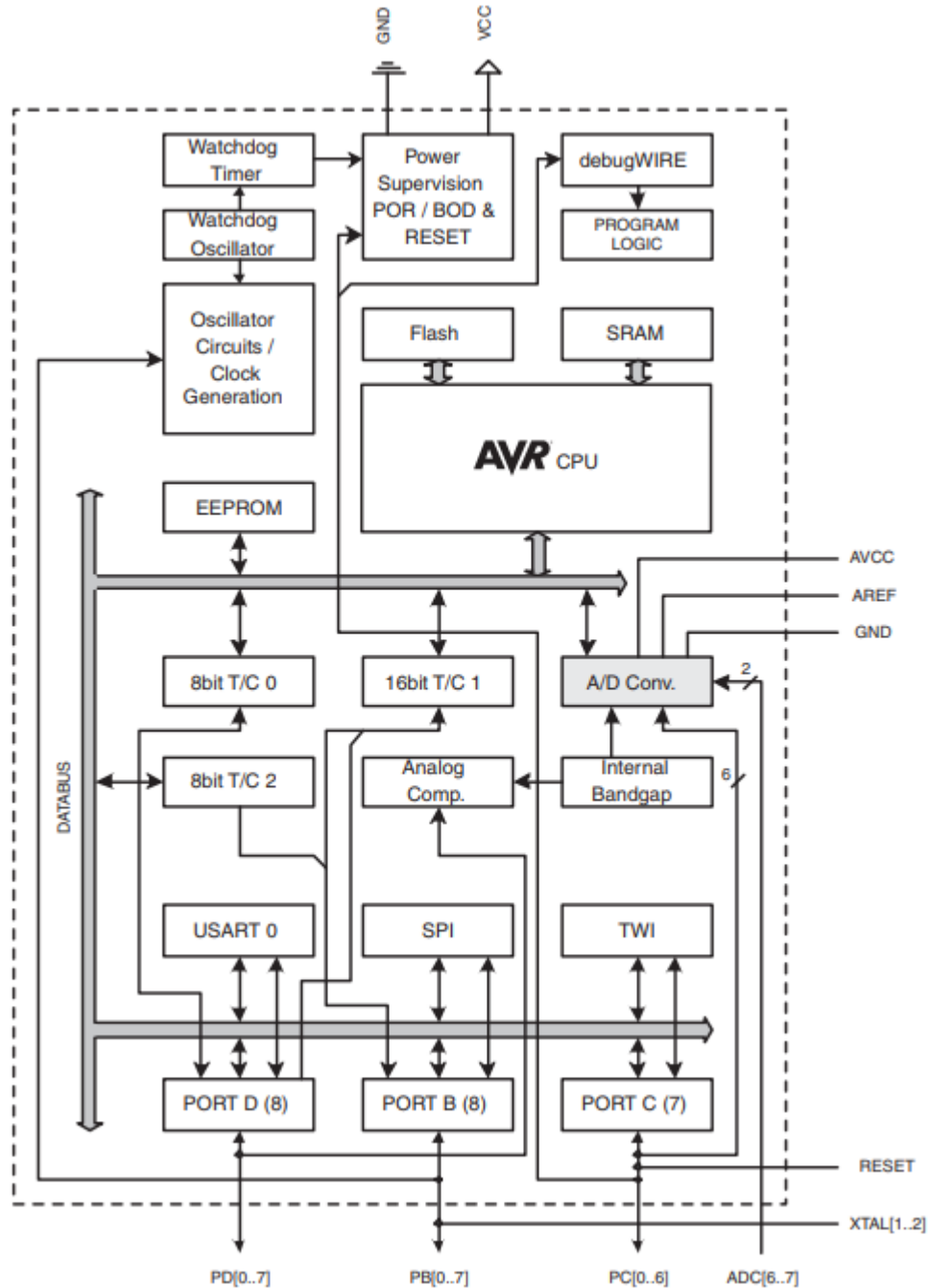


b. Microcontrolerul

Se prezintă schema bloc a microcontrolerului.

Observăm memoriile FLASH pentru program, SRAM de lucru și EEPROM.

Se observă porturile de intrare aferente. Pe porturile de intrare sunt implementate interfețele de comunicare serial SPI, TWI(Two Wire Interface sau I²C), și USART (utilizat și pentru încărcarea codului program). În continuare se găsesc ilustrate maparea porturilor Arduino la microcontroler. De asemenea se poate observa portul ADC pentru pinii analogici.



Atmega168 Pin Mapping

Arduino function				Arduino function
reset	(PCINT14/RESET) PC6	1	28	PC5 (ADC5/SCL/PCINT13) analog input 5
digital pin 0 (RX)	(PCINT16/RXD) PD0	2	27	PC4 (ADC4/SDA/PCINT12) analog input 4
digital pin 1 (TX)	(PCINT17/TXD) PD1	3	26	PC3 (ADC3/PCINT11) analog input 3
digital pin 2	(PCINT18/INT0) PD2	4	25	PC2 (ADC2/PCINT10) analog input 2
digital pin 3 (PWM)	(PCINT19/OC2B/INT1) PD3	5	24	PC1 (ADC1/PCINT9) analog input 1
digital pin 4	(PCINT20/XCK/T0) PD4	6	23	PC0 (ADC0/PCINT8) analog input 0
VCC	VCC	7	22	GND GND
GND	GND	8	21	AREF analog reference
crystal	(PCINT6/XTAL1/TOSC1) PB6	9	20	AVCC VCC
crystal	(PCINT7/XTAL2/TOSC2) PB7	10	19	PB5 (SCK/PCINT5) digital pin 13
digital pin 5 (PWM)	(PCINT21/OC0B/T1) PD5	11	18	PB4 (MISO/PCINT4) digital pin 12
digital pin 6 (PWM)	(PCINT22/OC0A/AIN0) PD6	12	17	PB3 (MOSI/OC2A/PCINT3) digital pin 11(PWM)
digital pin 7	(PCINT23/AIN1) PD7	13	16	PB2 (SS/OC1B/PCINT2) digital pin 10 (PWM)
digital pin 8	(PCINT0/CLKO/ICP1) PB0	14	15	PB1 (OC1A/PCINT1) digital pin 9 (PWM)

Digital Pins 11, 12 & 13 are used by the ICSP header for MOSI, MISO, SCK connections (Atmega168 pins 17, 18 & 19). Avoid low-impedance loads on these pins when using the ICSP header.

A se observa pe schema circuitului electric pinii de intrare de la placa Arduino la microcontroler în partea din dreapta jos a schemei. De asemenea, identificați următoarele componente:

- Rezonator sau oscilator ceramic. Oferă un semnal electric oscilator cu frecvența de 16Mhz. Conectat la pinii XTAL1 și XTAL2. Observați de asemea conectivitatea în schema bloc a microcontrolerului.
- Pinul de RESET. Valoarea pinului este ridicată prin rezistența de 10KΩ la 5V deoarece la tensiuni mici poate exista zgomot în circuit care să reseteze microcontrolerul. De asemenea pentru protecția împotriva descărcărilor electrostatice se adaugă dioda D2.
- Condensatorii C4 și C6. Aceștia filtrează semnalele zgomotoase. Conform formulei impedanței capacitive frecvențele mari vor scădea impendata iar curentul va trece mai ușor spre masă.
- ICSP (In-Circuit Serial Programming) legat la interfața SPI a microcontrolerului pentru programarea acestui. De obicei nu se utilizează deoarece programul este încărcat prin utilizarea portului USB. Se poate utiliza pentru reconfigurarea microcontrolerului.

c. Puntea USB-UART

Pe schema circuitului se poate observa microcontrolerul ATmega16U2. Acesta are rolul de a transforma datele primite de la portul USB în date care vor fi încărcate în microcontrolerul principal prin UART utilizându-se de o serie de instrucțiuni preîncărcate din fabrică. Pe placă această componentă se găsește lângă portul de intrare USB.

Se observă următoarele componente pe schema circuitului electric:

- Terminatorii rezistivi RD- și RD+ care asigură impedanțele corespunzătoare standardului USB
- Rezistențele dependente de tensiune Z1 și Z2 care protejează liniile USB împotriva descărcărilor electrostatice
- Condensatorul C5 care transformă semnalul de tip declanșare la nivel (level-triggered) DTR (Data Terminal Ready) generat de microcontrolerul de conversie într-un semnal de tip declanșare la tranziție (edge-triggered).

E. Configurarea Arduino IDE

Urmați următorii pași:

1. Descărcați și instalați Arduino IDE.

Link-ul pentru installer: <https://www.arduino.cc/en/Main/Software>

Urmăriți chenarul roșu pentru instalarea IDE-ului.

Download the Arduino IDE



 **ARDUINO 1.8.7**

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software.

This software can be used with any Arduino board. Refer to the [Getting Started](#) page for installation instructions.

Windows Installer, for Windows XP and up
Windows ZIP file for non-admin install

Windows app Requires Win 8.1 or 10
Get

Mac OS X 10.8 Mountain Lion or newer

Linux 32 bits
Linux 64 bits
Linux ARM

Release Notes
Source Code
Checksums (sha512)

Contribute to the Arduino Software

Consider supporting the Arduino Software by contributing to its development. (US tax payers, please note this contribution is not tax deductible). [Learn more on how your contribution will be used.](#)

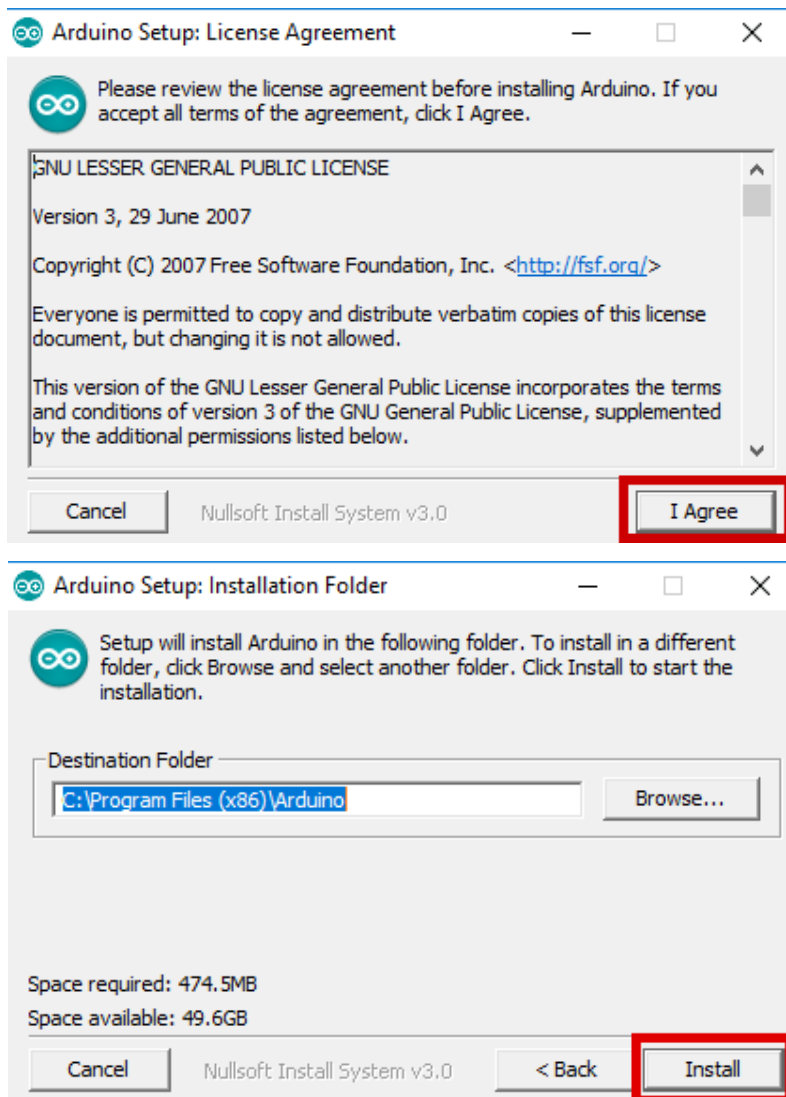


SINCE MARCH 2015, THE ARDUINO IDE HAS BEEN DOWNLOADED **28,170,698** TIMES. (IMPRESSIVE!) NO LONGER JUST FOR ARDUINO AND GENUINO BOARDS, HUNDREDS OF COMPANIES AROUND THE WORLD ARE USING THE IDE TO PROGRAM THEIR DEVICES, INCLUDING COMPATIBLES, CLONES, AND EVEN COUNTERFEITS. HELP ACCELERATE ITS DEVELOPMENT WITH A SMALL CONTRIBUTION! REMEMBER: OPEN SOURCE IS LOVE!

\$3 \$5 \$10 \$25 \$50 OTHER

JUST DOWNLOAD CONTRIBUTE & DOWNLOAD

2. Deschideți fișierul descărcat și urmați instrucțiunile



Arduino Setup: License Agreement

Please review the license agreement before installing Arduino. If you accept all terms of the agreement, click I Agree.

GNU LESSER GENERAL PUBLIC LICENSE
Version 3, 29 June 2007
Copyright (C) 2007 Free Software Foundation, Inc. <<http://fsf.org/>>
Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.
This version of the GNU Lesser General Public License incorporates the terms and conditions of version 3 of the GNU General Public License, supplemented by the additional permissions listed below.

Cancel Nullsoft Install System v3.0 **I Agree**

Arduino Setup: Installation Folder

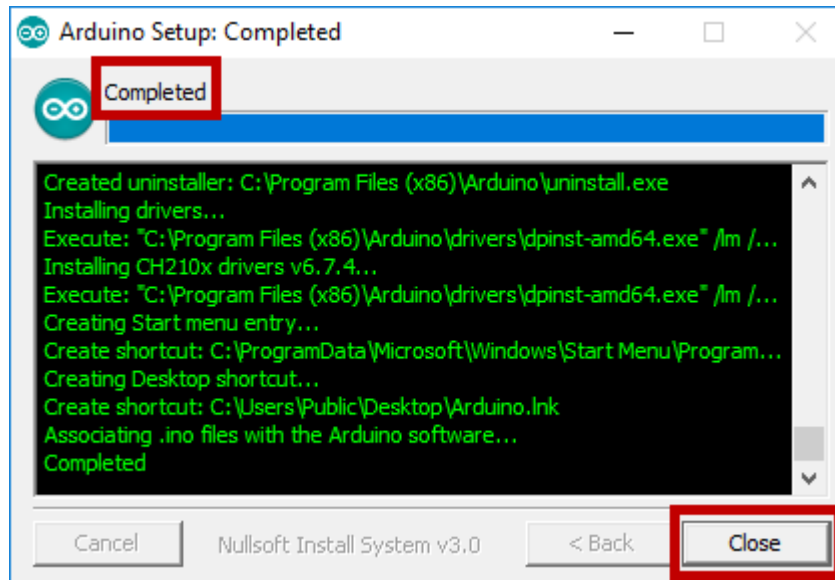
Setup will install Arduino in the following folder. To install in a different folder, click Browse and select another folder. Click Install to start the installation.

Destination Folder
 Browse...

Space required: 474.5MB
Space available: 49.6GB

Cancel Nullsoft Install System v3.0 < Back **Install**

! Dacă vă cere acordul pentru instalarea unui driver pentru USB Arduino acceptați.

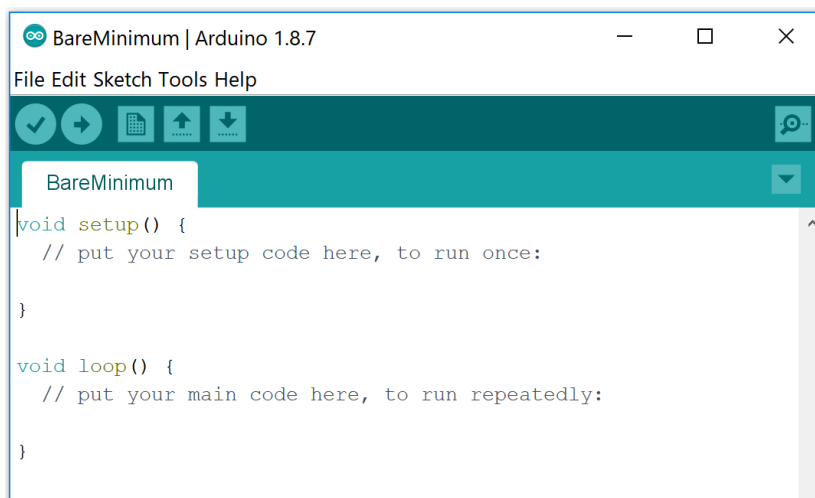


F. Rularea unui program (sketch) simplu

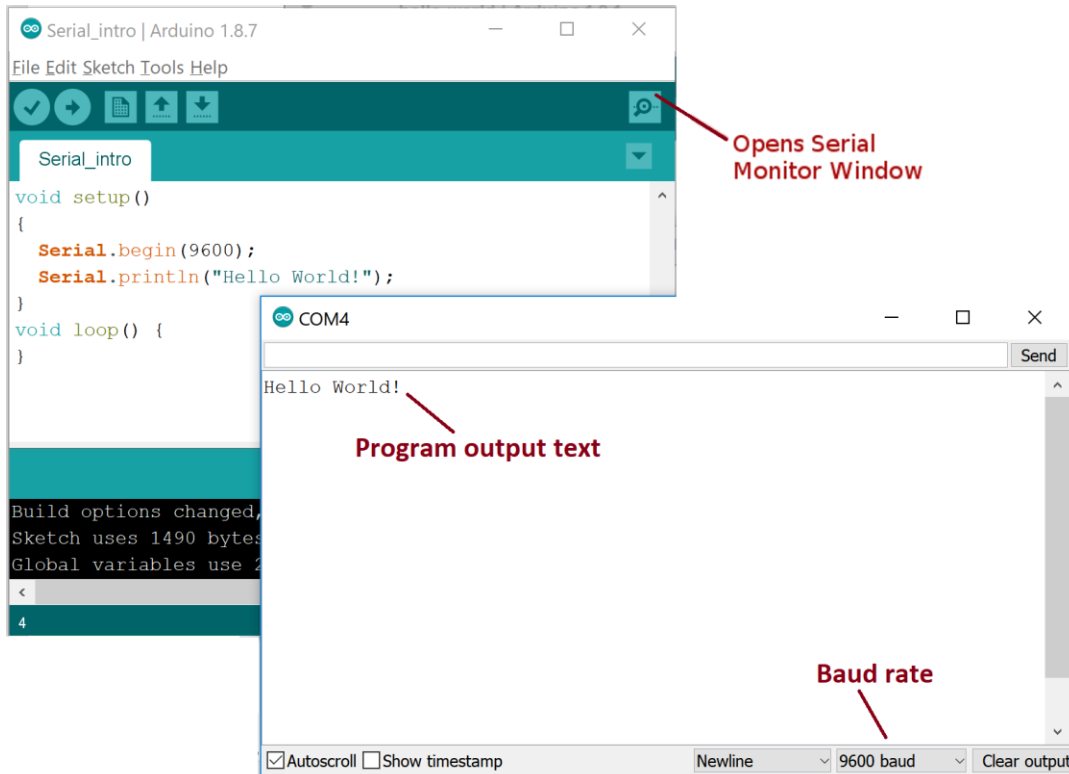
Structura de bază a unui program Arduino (denumit sketch) este alcătuită din două funcții numite `setup()` și `loop()`.

Pentru a vizualiza un sketch de bază se vor urma pașii:

1. Deschideți Arduino IDE.
2. Selectați: **File** → **Examples** → **01.Basics** → **BareMinimum**
3. Observați cele două funcții: `setup` și `loop`. Funcția `setup` rulează o dată la pornirea programului. Funcția `loop` rulează continuu în buclă.



4. Se va modifica codul programului BareMinimum prin adaugarea codului de mai jos și salvarea programului cu denumirea **hello_world** (File → Save As...)



5. Din meniul **Tools** la rubrica **Board** setăm "Arduino/Genuino Uno".
6. Conectăm placa la un port USB al calculatorului.
7. Din meniul **Tools** la rubrica **Port** setăm portul activ care corespunde plăcii.
8. Se apasă butonul **Upload** din interfața Arduino IDE sau combinația de taste **Ctrl+U**.
9. Se observă funcționarea programului in fereastra Serial Monitor Window.

G. Rularea unui program pentru controlul unui LED onboard

1. Vom dezvolta un program care să stingă și să aprindă LED-ul de pe placă la un interval de câte o secundă.
2. În funcția setup vom seta ca pin-ul 13, cel de LED_BUILTIN să fie de tip OUTPUT.
3. În funcția loop setăm pin-ul aferent LED-ului la valoarea HIGH, apoi introducem o întârziere de o secundă. După setăm pin-ul la valoarea LOW și introducem din nou o întârziere de o secundă.

```
void setup() {  
    pinMode(13, OUTPUT);  
}  
  
void loop() {  
    digitalWrite(13, HIGH);  
    delay(1000);  
    digitalWrite(13, LOW);  
    delay(1000);  
}  
}
```

SAU

```
void setup() {  
    pinMode(LED_BUILTIN, OUTPUT);  
}  
  
void loop() {  
    digitalWrite(LED_BUILTIN, HIGH);  
    delay(1000);  
    digitalWrite(LED_BUILTIN, LOW);  
    delay(1000);  
}  
}
```

4. Din meniul **Tools** la rubrica **Board** setăm “Arduino/Genuino Uno”.
5. Conectăm placa la un port USB al calculatorului.
6. Din meniul **Tools** la rubrica **Port** setăm portul activ care corespunde plăcii.
7. Se apasă butonul **Upload** din interfața Arduino IDE sau combinația de taste **Ctrl+U**.
8. Se observă modul în care funcționează LED-ul.
9. Se modifică timpii de întârziere și se încarcă din nou programul pe placă.
10. Se urmărește modul în care funcționează LED-ul.
11. Se deconectează placa de la USB și se conectează din nou. Se observă că programul este în continuare încărcat pe placă și rulează.

H. Rularea unui program pentru controlul unui LED onboard prin comenzi pe portul Serial

1. Vom dezvolta un program care să comande LED-ul de pe placă, prin comenzile utilizatorului pe portul Serial

```

void setup()
{
  pinMode(13, OUTPUT);
  Serial.begin(9600);
  Serial.println("Input:\n 1 for LED ON\n 0 for LED OFF");
}
void loop()
{
  if (Serial.available() > 0)
  {
    char state = Serial.read();

    if (state == '1' )
    {
      digitalWrite(13, HIGH);
      Serial.println("LED --> ON");
    }

    if (state == '0')
    {
      digitalWrite(13, LOW);
      Serial.println("LED --> OFF");
    }
  }
}

```

2. Din meniul **Tools** la rubrica **Board** setăm “Arduino/Genuino Uno”.
3. Conectăm placa la un port USB al calculatorului.
4. Din meniul **Tools** la rubrica **Port** setăm portul activ care corespunde plăcii.
5. Se apasă butonul **Upload** din interfața Arduino IDE sau combinația de taste **Ctrl+U**.
6. Se observă funcționarea programului în fereastra Serial Monitor Window.

Studiu suplimentar și bibliografie:

- <https://store.arduino.cc/arduino-uno-rev3>
- <https://www.microchip.com/wwwproducts/en/ATmega328P>
- <http://ww1.microchip.com/downloads/en/DeviceDoc/ATmega48A-PA-88A-PA-168A-PA-328-P-DS-DS40002061A.pdf>
- <https://www.allaboutcircuits.com/technical-articles/understanding-arduino-uno-hardware-design/>
- <https://i2.wp.com/www.technobyte.org/wp-content/uploads/2016/08/Arduino-Uno-multiple-input-management-circuit-op-amp-comparator-min.jpg>
- <http://www.ntu.edu.sg/home/ehchua/programming/arduino/arduino.html>
- <https://learn.sparkfun.com/tutorials/what-is-an-arduino/whats-on-the-board>
- <https://www.arduino.cc/en/Hacking/PinMapping168>
- https://en.wikipedia.org/wiki/Ceramic_resonator
- <http://users.utcluj.ro/~baruch/sie/labor/Magistrala-USB.pdf>

- <https://components101.com/microcontrollers/arduino-uno>
- Getting started with Arduino Uno: <https://www.youtube.com/watch?v=zusVAX7bgtA>
- <https://electronics.stackexchange.com/questions/61537/what-is-the-cap-for-in-arduino-reset-circuit>
- <https://arduino.stackexchange.com/questions/986/whats-the-2nd-icsp-header-for-in-arduino-uno-r3>
- <https://startingelectronics.org/software/arduino/learn-to-program-course/01-program-structure-flow/>