

Laborator 1

PREZENTAREA MODULUI DE LUCRU CU PLĂCILE DE DEZVOLTARE ȘI MEDIUL DE PROGRAMARE ARDUINO

1. Prezentarea mediului de programarea Arduino.

Arduino este o companie “open-source” care a dezvoltat partea hardware reprezentată de plăci de dezvoltare, module de extensie care poartă de numirea de “shield-uri” care se conectează hardware „stivuite” iar software de obicei prin protocolul I2C pentru a putea fi conectate mai multe în paralel. De asemenea sunt dezvoltați diverși senzori ce se pot conecta la sistemele de dezvoltare hardware. Totodată pentru dezvoltarea software-ului a fost dezvoltată o platformă de dezvoltare “Arduino” [1].

Plăcile de dezvoltare asigură prin mufa USB:

- tensiunea de alimentare necesară funcționării plăcii de dezvoltare (microcontrolerul),

- tensiunea de alimentare pentru modulele conectate la placa de dezvoltare, însă trebuie să se țină cont de curentul maxim debitat de portul USB al calculatorului/laptopului la care este conectat,

- scrierea în microcontroler a fișierului programului rezultat în urma compilării. Programul poate fi scris în limbaj C sau C++.

Arduino ca platformă și sistem de dezvoltare, poate fi utilizat pentru a programa o serie de procesoare din familia Atmel. Se utilizează procesoare de tip SoC (System on Chip).

În cadrul laboratorului se vor utiliza platformele Arduino Uno R3. Acestea utilizează microcontrolerul Atmega328P, care are o arhitectură pe 8 biți, sunt alimentate la o tensiune de 5V (de la portul USB) și o frecvență a oscilatorului de 16MHz, memorie SRAM de 2kB, EEPROM 1kB și o memorie flash pentru programe de 32kB din care 0,5kB este utilizată de către bootloader (programul care asigură comunicația cu calculatorul, care are și rolul de a realiza transferul și programarea software-ului de la calculator).

Ca interfață placa de dezvoltare are:

- 14 pini digitali (pini de intrare-ieșire, niveluri logice, din care 6 pot fi ieșiri cu PWM- **Pulse width modulation**, este o tehnică folosită pentru a varia în mod controlat tensiunea dată unui dispozitiv electronic. Această metodă schimbă foarte rapid tensiunea oferită dispozitivului respectiv din ON în OFF și invers (trecuri rapide din HIGH în LOW, de exemplu 5V - 0V). Raportul dintre perioada de timp corespunzătoare valorii ON și perioada totală dintr-un ciclu ON-OFF se numește **factor de umplere (duty cycle)** și reprezintă, în medie, tensiunea pe care o va primi dispozitivul electronic. Astfel, se pot controla circuite analogice din domeniul digital. Practic, aceasta asta înseamnă că un LED acționat astfel se va putea aprinde / stinge gradual, iar în cazul unui motor acesta se va învârti mai repede sau mai încet),

- 6 pini analogici notați de A0 până la A5 (pot fi utilizați ca și intrări pentru convertorul analog-digital pe 10 biți, încorporat), de asemenea acești pini pot fi declarați și utilizați ca pini de digitali de intrare/ieșire .

Alimentarea microcontrolerului se realizează la 5V dar sistemul de dezvoltare poate fi alimentat de la un alimentator extern în intervalul maxim 6-20V, însă este recomandată o tensiune de 7-12V. Producătorul precizează că tensiunea optimă este de 9V. Comunicația cu calculatorul se realizează prin intermediul interfeței USB, se realizează un port serial virtual care permite programarea sistemului de dezvoltare dar și comunicația serială a eventualelor programe cu calculatorul (care nu mai este condiționat de existența unui port serial realizat cu circuite dedicate). Unele sisteme de dezvoltare din gama Arduino au implementat on-board această comunicație, altele necesită un circuit extern, pentru a asigura conversia USB-serial.

Toate pot fi programate prin intermediul unui port ICSP dedicat, acest port se utilizează îndeosebi când din diferite motive este necesară reprogramarea programului de inițializare bootloader.

Utilizarea mediului de programare ARDUINO

Lansarea programului se face fie din mediul “START” și căutarea pictogramei din figura 1 după care se lansează. A doua variantă de lansare a programului este “dublu click”, pe pictograma de pe desktop. Pictograma mediului de programare Arduino este prezentată în figura 1.



Fig. 1. Pictograma mediului de programare Arduino.

După lansarea programului va apărea o fereastră de editare ca în figura 2, în care apar funcțiile:

“SETUP” în care sunt setări și inițializări, instrucțiunile de aici se execută o singură dată, la pornire, inițializarea variabilelor, a comunicației

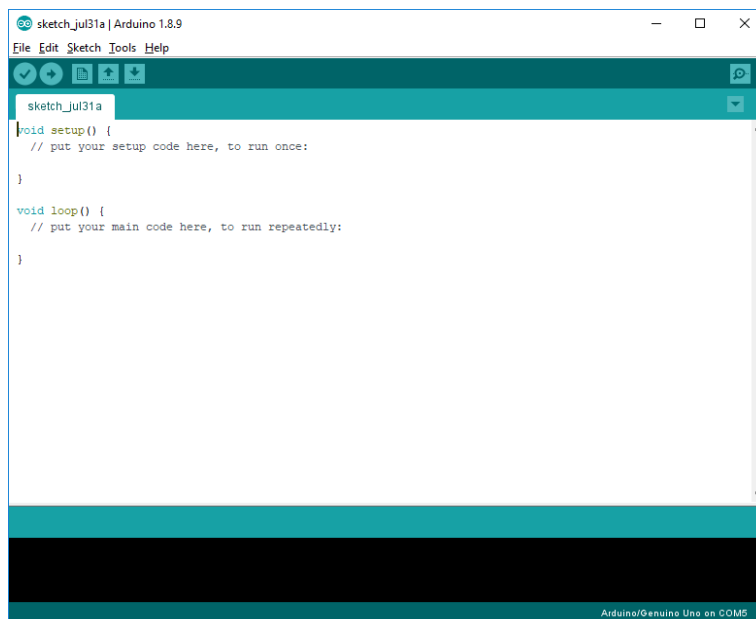


Fig. 2. Fereastra mediului de programare.

seriale (de exemplu baud rate), pentru declararea pinilor ca fiind de ieșire sau de intrare, diverse funcții pentru inițializarea unor senzori,
- “LOOP”, unde este scris programul ce se va executa. Instrucțiunile scrise se execută în buclă, de la prima la ultima instrucțiune. Acesta buclă se reia la infinit până intervine o condiție de oprire a buclei.

Apoi se va salva fișierul creat cu numele dorit în locația dorită.

Notă: Fiecare proiect Arduino, indiferent dacă este compus dintr-un singur fișier sursă sau mai multe fișiere sursă către care se face trimitere din fișierul principal trebuie plasat într-un director cu același nume ca fișierul sursă.

Selectarea plăcii de dezvoltare

După ce a fost scris codul în mediul de programare înainte de compilare trebuie selectată placa de dezvoltare pentru care va fi compilat software-ul, ca în figura 3.

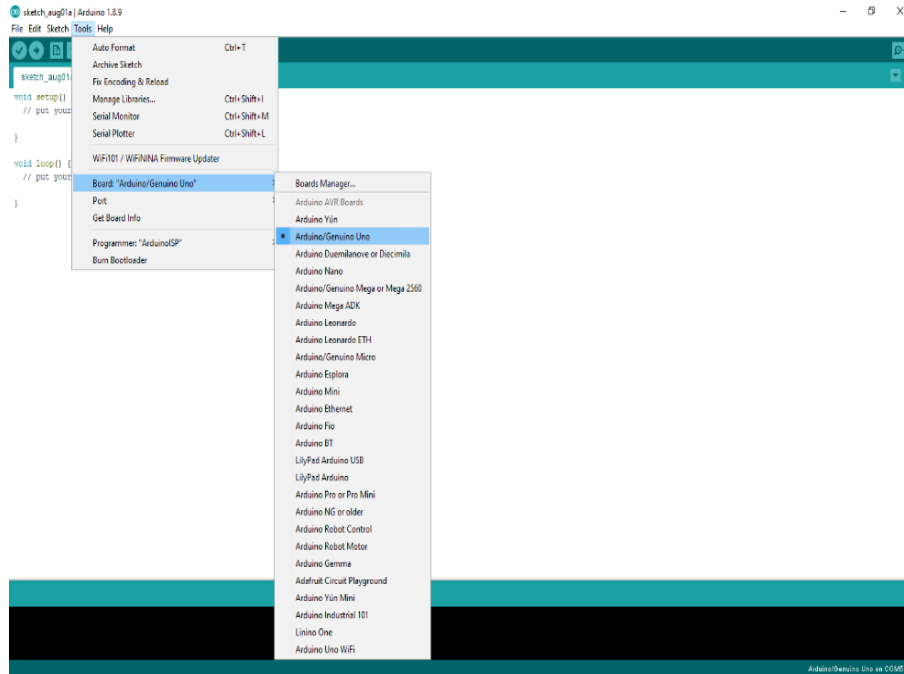


Fig. 3. Selectarea sistemului de dezvoltare pentru care se va compila software-ul.

Compilarea software-ului

După ce s-a selectat placa de dezvoltare trebuie verificat din punct de vedere al erorilor de sintaxă și apoi va fi compilat codul scris. În figura 4 se prezintă meniul de unde se face această operație, din meniu sau pictogramă.

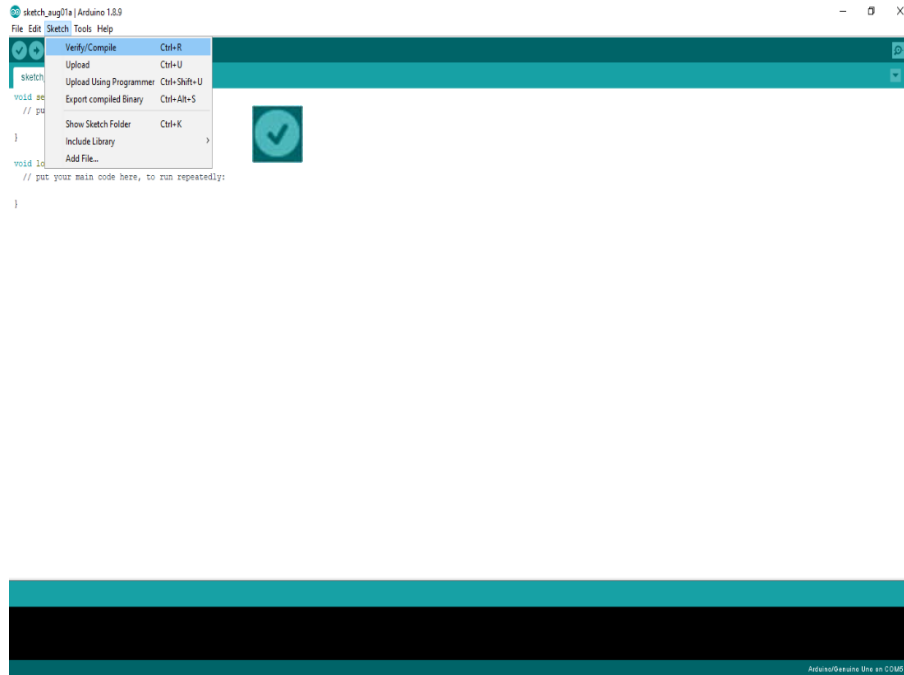


Fig. 4. Meniul în care se găsește submeniul de compilare.

După verificare și compilare dacă nu există greșeli de sintaxă sau erori în partea de jos a ferestrei va apărea un mesaj în care se va prezenta spațiul de memorie ocupat de program, figura 5 sau o eroare în cazul unei erori precum și tipul erorii, figura 6.

De asemenea în fereastra de eroare va fi detaliată eventuala greșeală de sintaxă.

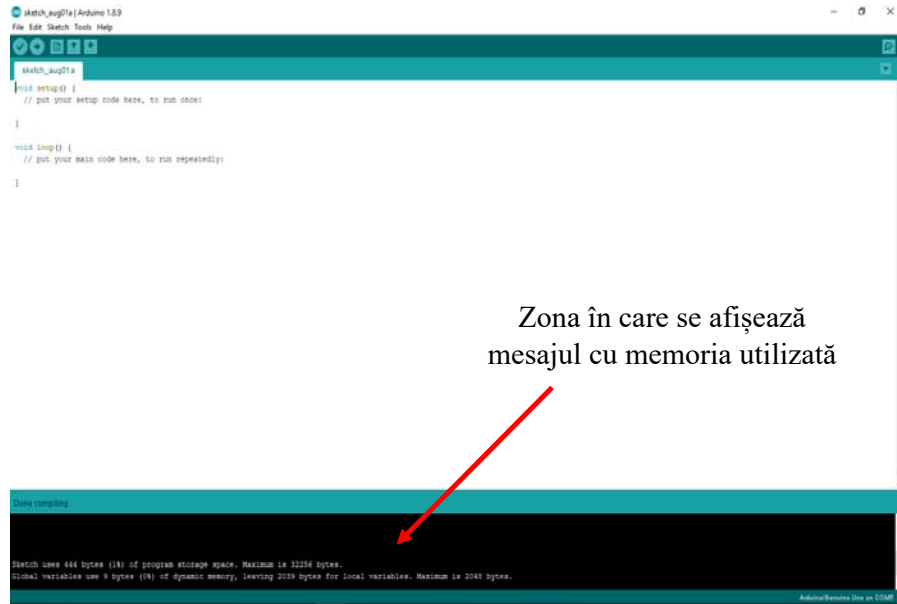


Fig. 5. Afișarea mesajelor când compilarea a fost realizată cu succes.

Este important să știm cât spațiu de memorie ocupă programul de aceea programul Arduino ne oferă această informație, ca în figura 5.

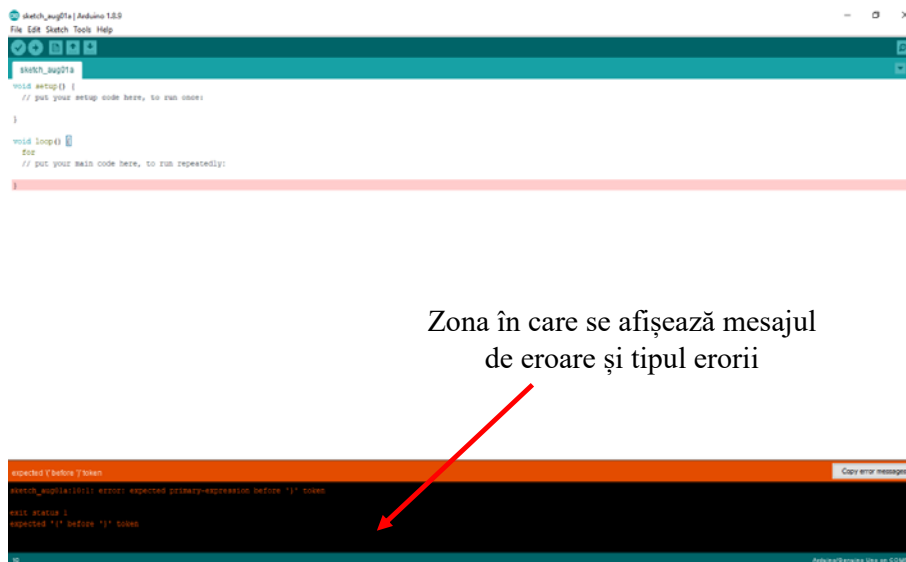


Fig. 6. Afișarea mesajelor când compilarea nu a fost realizată cu succes.

Sunt situații în care sunt erori de sintaxă în codul scris, caz în care apar erori care sunt afișate de mediul de programare, ca în figura 6.

Dacă nu există erori după compilare se poate scrie fișierul rezultat cu extensia “hex” în microcontroler. Această operație se face conform figurii 7 din meniu sau cu ajutorul pictogramei.

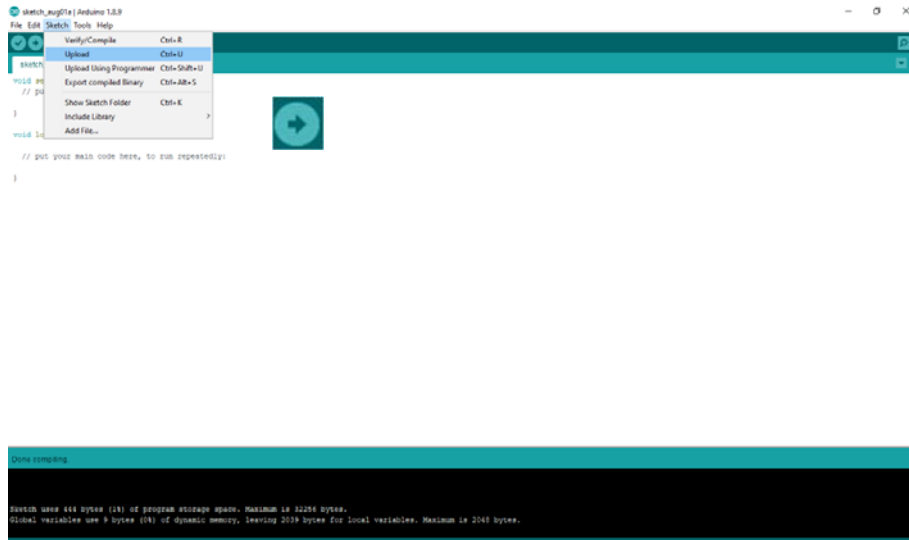


Fig. 7. Prezentarea meniului și submeniului pentru încărcarea software-ului în microcontroler

Totodată când se utilizează această comandă mediul de programare realizează și o verificare și compilare, după care dacă nu există erori se va scrie fișierul “hex” în microcontroler.

Alte pictograme ce apar în fereastra principală a mediului de programare.



Creează un nou proiect



Deschide un proiect



Salvează proiectul curent



Monitorizare serială – acest buton este folosit pentru comunicarea prin portul serial cu calculatorul (se află în partea dreaptă a ferestrei)

Instalarea librărilor

Mediul de programare ARDUINO fiind un mediu “open-source” dispune de numeroase librării dezvoltate de producătorul componentei hardware sau de diverși programatori. De aceea când utilizăm diverși senzori sau module trebuie să instalăm librăriile corespunzătoare. În figura 8 se prezintă meniul din care se instalează librăriile. În meniul “Tools” se selectează opțiunea “Manage Libraries ...”

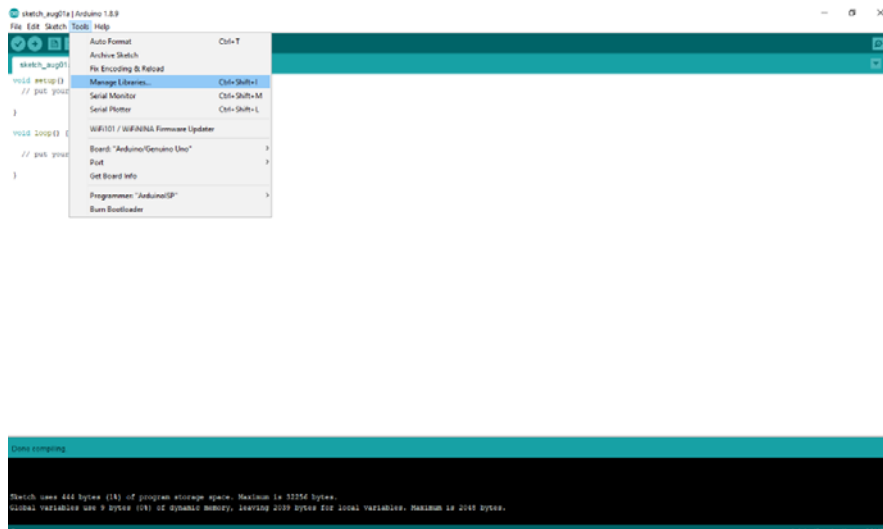


Fig. 8. Prezentarea meniului pentru instalarea librărilor.

După selectarea acestei opțiuni va apărea o fereastră ca în figura 9. În zona de editare se va scrie denumirea librăriei pe care doriți să o instalați apoi se apasă butonul “Install” din librăria dorită.

Tot din această fereastră în partea stângă sus la zona de tip selecție “Type” se poate selecta modul de lucru “Update”, caz în care se poate face update la librăriile deja instalate, în cazul în care au apărut diverse actualizări sau modificări ale librărilor.

De asemenea în partea stângă sus apare o zonă de tip selecție “Topic”, de unde se poate selecta categoria de librărie ce se dorește a se instala sau la care se dorește a se face update.

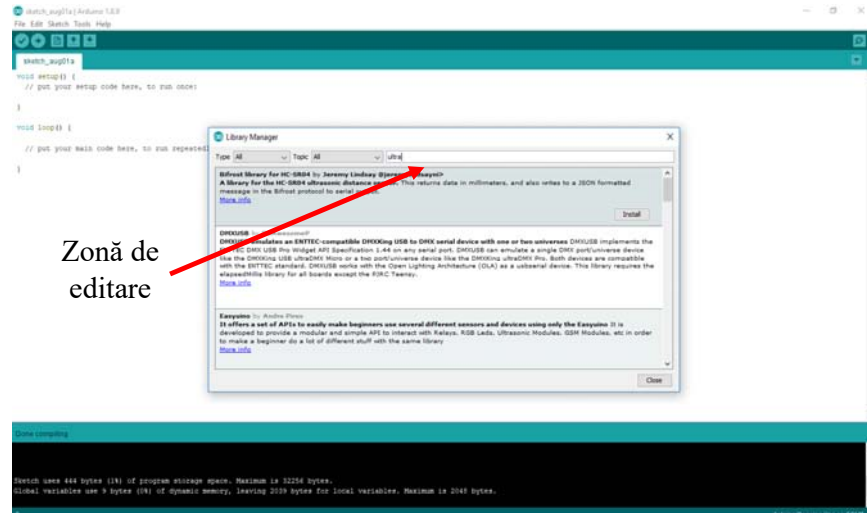


Fig. 9. Prezentarea modului de căutare a unei librării specifice.

Monitorizarea transmisiei pe portul serial

Modulul ARDUINO UNO R3 dispune de un port serial hardware la piniile digitale R_x și T_x . De asemenea se poate lucra și cu numeroase porturi seriale virtuale la care se declară de către utilizator funcțiile pinilor.

Pentru a putea monitoriza ceea ce se trimite pe portul serial hardware sau software se va utiliza din meniul “Tool”, submeniul “Serial Monitor”, ca în figura 10. Totodată trebuie selectat numărul portului serial. Este posibil ca la un moment dat să fie mai multe plăci de dezvoltare conectate la porturile USB ale calculatorului de aceea trebuie selectat portul dorit. Din fereastra nou deschisă se poate selecta “baud rate” (viteza de transfer pe portul serial), care trebuie să fie sincronizată cu viteza setată în codul C sau C++, ce a fost compilat și apoi încărcat în microcontroller, ca în figura 11.

Dacă se dorește să se trimită date către microcontroller în zona de editare din figura 10 se scrie textul dorit după care apasă butonul “Send”. Trebuie ca și programul încărcat în microcontroller să accepte prin software primirea de date prin portul serial pentru a avea efect funcția “Send”. Datele manipulate prin portul serial hardware sau software trebuie să fie în format caracter („char”).

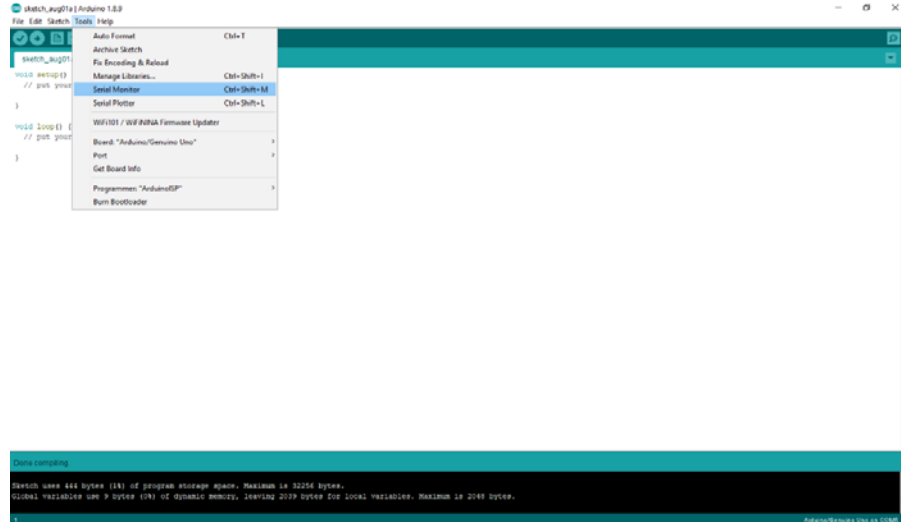


Fig. 10. Modul de afișare a ferestrei „Serial Monitor”

Datele afișate în această fereastră sunt de tip “char”, și ușor de interpretat de utilizator.

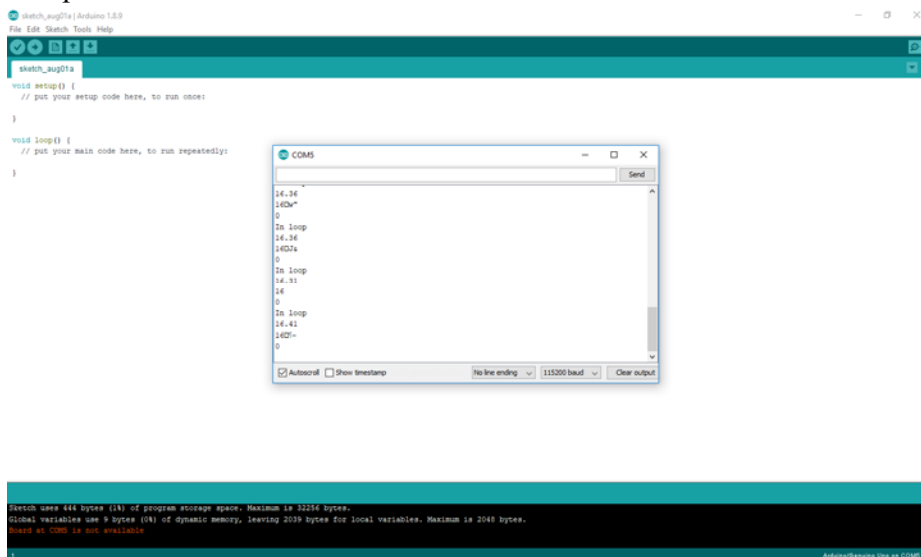


Fig. 11. Vizualizarea datelor transmise / recepționate pe portul serial.

Mediul de dezvoltare ARDUINO permite afișarea grafică a datelor transmise prin portul serial. Pentru această opțiune din meniul “Tools”. Se va selecta submeniul “Serial Plotter”, ca în figura 12. Va fi afișată fereastra din figura 13.

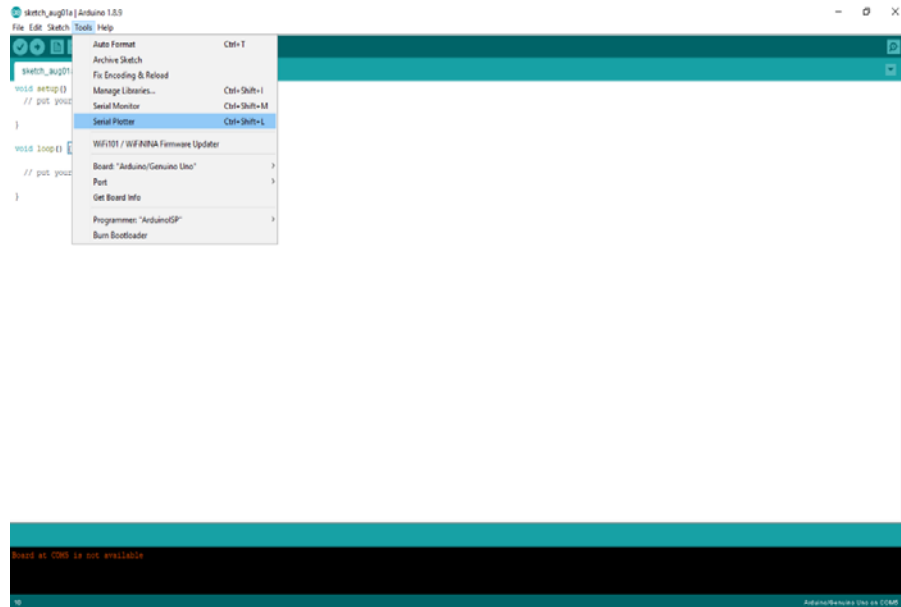


Fig. 12. Modul de afișare a ferestrei „Serial Plotter”

Pentru afișarea corectă trebuie setată viteza de transfer (“baud rate”). Dacă cele două rate de transfer nu sunt sincronizate ca valori, caracterele recepționate nu vor fi corect decodate și afișate.

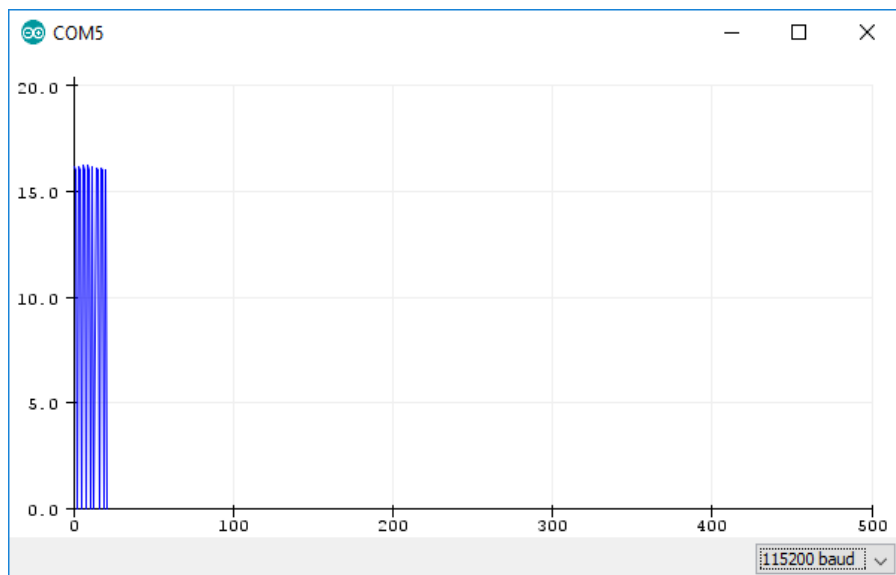


Fig. 13. Afișarea grafică a datelor recepționate pe portul serial în fereastra „Serial Monitor”

De studiat în laborator:

1. Se va lansa programul Arduino și se vor studia meniurile și comenzile necesare.
2. Se va încărca un exemplu din submeniul “Examples – Basics - Blink” și se va studia programul.
3. Se va realiza montajul corespunzător exemplului deschis ca în figura 14, se va încărca programul în microcontroler și se va urmări funcționarea lui. Rezistența va avea valoarea de 220 Ohmi [2].

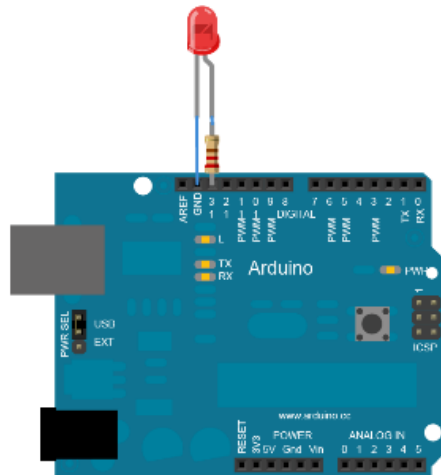


Fig. 14. Schema montajului de realizat.

4. Se va modifica pinul de ieșire digitală iar apoi se va modifica timpul de schimbare a stării ledului (ON/OFF)

Bibliografie

- [1] <https://www.arduino.cc/en/Tutorial/HomePage>
[2] <https://www.arduino.cc/en/Tutorial/Blink>