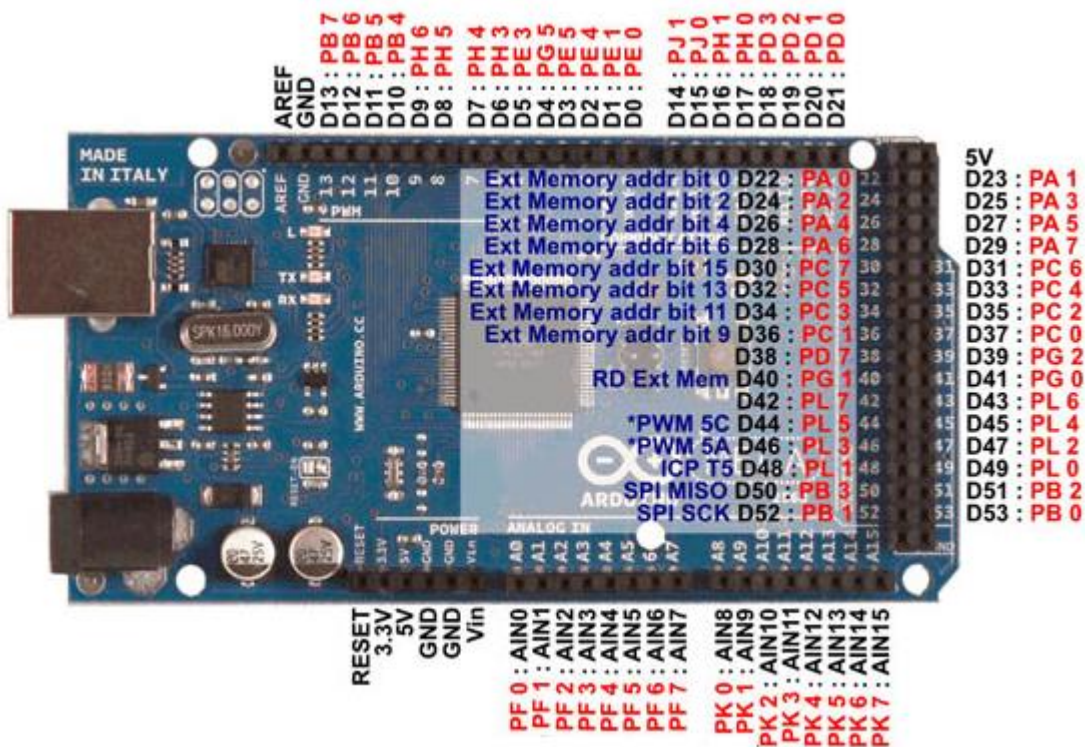


Laborator 1 – Introducere în utilizarea plăcilor Arduino

Familia de unelte de dezvoltare Arduino include plăci cu microcontroller, accesorii, și componente software open source, care permit utilizatorilor să realizeze proiecte folosind o abordare unificată, de nivel înalt, care se dorește a fi independentă de microcontrollerul folosit. Plăcile Arduino sunt echipate în principal cu microcontrollere Atmel AVR, dar există și plăci echipate cu microcontrollere de tip ARM, sau din familia x86. În afara plăcilor Arduino oficiale, există o gamă largă de clone, de obicei cu preț redus (și performanță discutabilă), precum XDduino, Freeduino, etc.

Placa pe care o vom folosi pentru activitatea de laborator este Arduino Mega 2560, bazată pe microcontrollerul Atmel AVR ATmega2560, pe 8 biți. Placa dispune de 54 de pini digitali pentru intrare/ieșire, și 16 pini pentru preluare de semnale analogice. Unii pini pot avea funcție multiplă, constituind semnale pentru diferite interfețe de comunicație (UART, SPI, I2C). Frecvența microcontrollerului este de 16 MHz. Placa se alimentează prin cablul USB cu care se conectează la calculator, sau poate fi alimentată cu o sursă de tensiune continuă, de 7 ... 12 V, care poate furniza o intensitate a curentului de minim 0.25 A. A doua opțiune este necesară când placa trebuie să alimenteze periferice mari consumatoare, precum motoare, shield-uri GSM. Etc.



1. Utilizarea unei plăci de dezvoltare Arduino: primul exemplu.

Pentru o mai bună desfășurare a activității, placa de dezvoltare a fost fixată pe un suport de plastic, împreună cu o placă de prototipizare (“breadboard”).

Pentru început, vom încerca pe placa exemplul cel mai elementar, “Blink”, disponibil în directorul de instalare al mediului de dezvoltare Arduino (de obicei C:\Program

Files\Arduino\examples\01.Basics\Blink). Pentru acest lucru, copiați directorul “Blink” în directorul vostru de lucru (pe care îl veți crea în D:\Studenti\Grupa30xxx ! În orice altă locație directorul dvs. va fi ȘTERS periodic). Verificați ca după copiere directorul și fișierul din interiorul acestuia, blink.ino, să nu fie cu atributele “Read only”.

Regulă: fiecare proiect Arduino, chiar dacă are doar un singur fișier sursă, trebuie plasat într-un director cu același nume ca fișierul sursa.

După copierea directorului, lansați în execuție mediul Arduino, executând dublu-click pe fișierul sursa **blink.ino**. Fereastra deschisă ar trebui să arate așa:



```
Arduino IDE - Blink | Arduino 1.0.6
File Edit Sketch Tools Help

Blink
/*
 * Blink
 * Turns on an LED on for one second, then off for one second, repeatedly.
 *
 * Most Arduinos have an on-board LED you can control. On the Uno and
 * Leonardo, it is attached to digital pin 13. If you're unsure what
 * pin the on-board LED is connected to on your Arduino model, check
 * the documentation at http://arduino.cc
 *
 * This example code is in the public domain.
 *
 * modified 8 May 2014
 * by Scott Fitzgerald
 */

// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin 13 as an output.
  pinMode(13, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(13, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000); // wait for a second
  digitalWrite(13, LOW); // turn the LED off by making the voltage LOW
  delay(1000); // wait for a second
}

13 Arduino Mega 2560 or Mega ADK on COM4
```

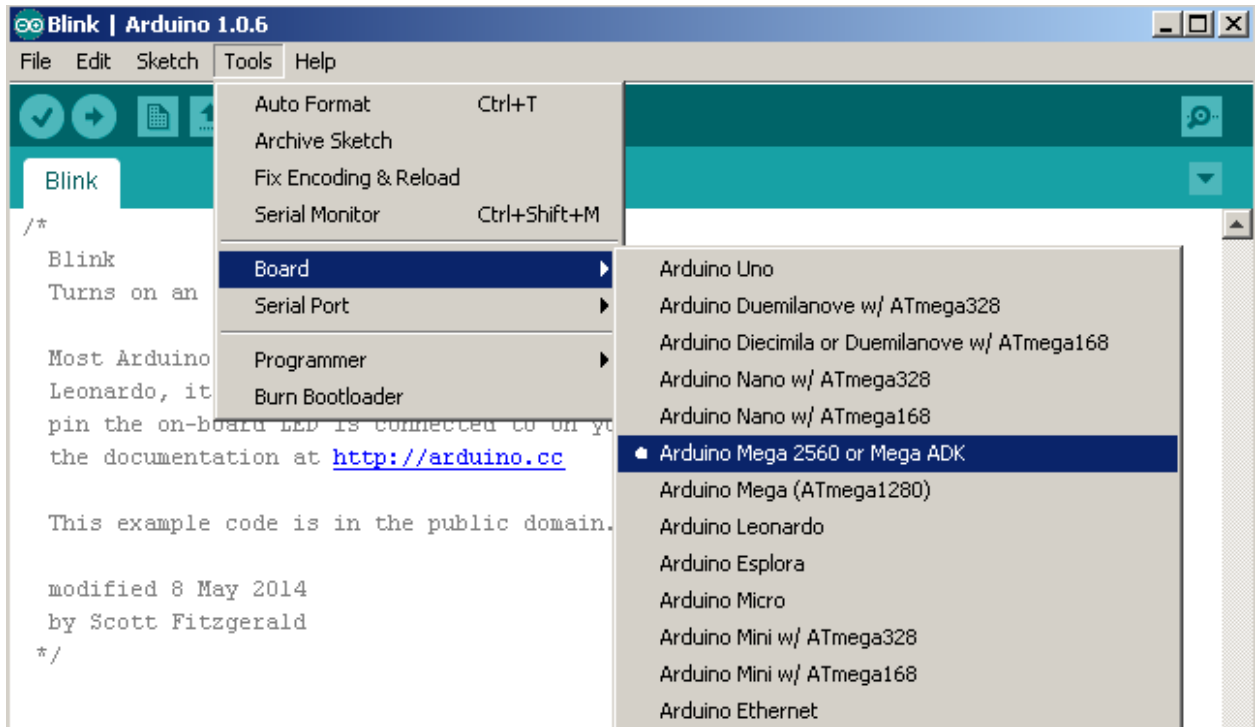
Dacă mediul de dezvoltare nu se deschide, este posibil ca asocierea dintre mediul și tipul de fișier .ino să nu se fi putut realiza cu succes. În acest caz, deschideți mediul de dezvoltare din Start/Programs, și alegeți din meniu opțiunea File->Open.

În acest moment, puteți conecta la PC placa Arduino Mega 2560, prin cablul USB. Folosiți, dacă se poate, portul USB din panoul frontal al calculatorului, deoarece veți fi nevoiți să deconectați și să re-conectați cablul de mai multe ori în timpul unui laborator.

După conectare, este posibil ca sistemul să ceară instalarea unui driver. Indicați sistemului de operare calea spre driver-ul Arduino, de obicei “C:\Program Files\Arduino\drivers”. Dacă nu vă descurcați, sau aveți probleme cu drepturile utilizator, apelați la ajutorul cadrului didactic.

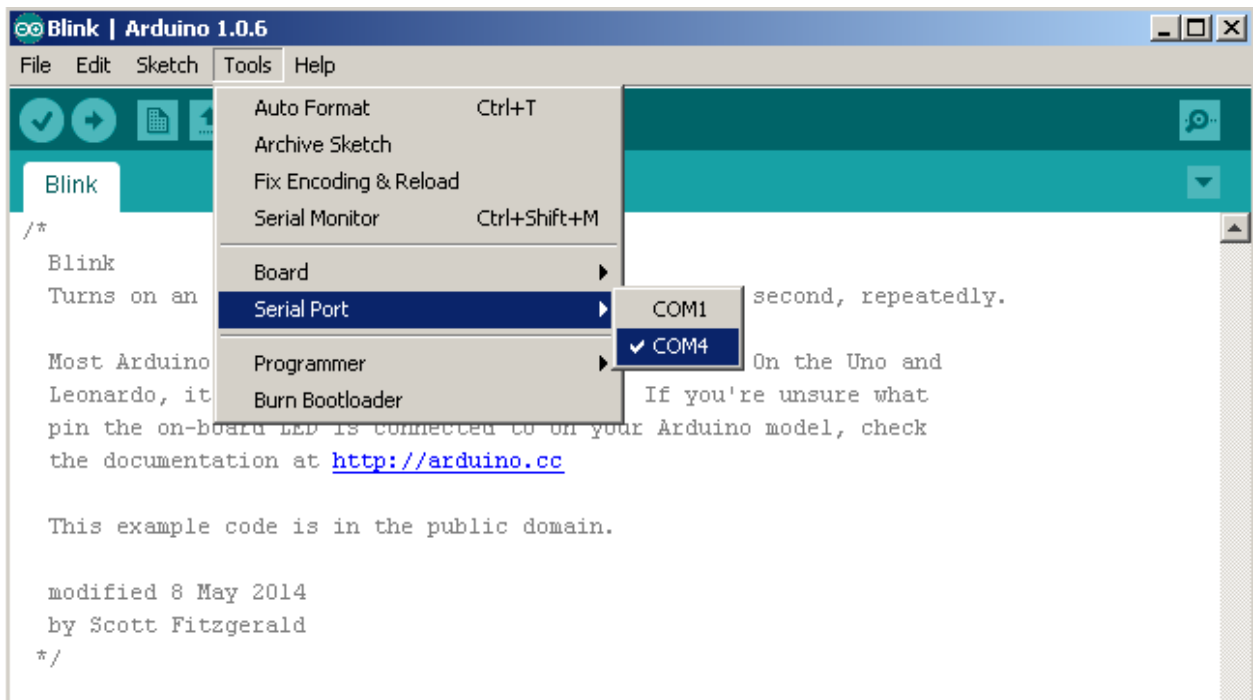
Dacă instalarea driverului a decurs cu succes, și placa Arduino este în funcțiune (este aprins un LED verde pe placă), puteți merge mai departe.

Înainte de a încerca să programăm placa, trebuie să ne asigurăm că mediul este configurat în mod corect. **Trebuie aleasă placa corectă, din meniul Tools->Board, conform imaginii de mai jos.**



Dacă folosiți o altă placă, precum Arduino Uno, trebuie să alegeți setările corespunzătoare.

De asemenea, trebuie configurat portul serial prin care se face comunicarea cu placa. Conexiunea USB va instala un port serial virtual, cu un nume ce începe cu COM, urmat de o cifra. Calculatoarele pe care lucrați au unul sau doua porturi seriale pe placa de baza, numite COM1 sau COM2. Porturile seriale virtuale primesc un număr mai mare, de la 3 în sus. Configurați portul serial din meniul Tools->Serial Port, ca în figura de mai jos:



După configurare, puteți compila și programa primul exemplu pe placă, apăsând butonul “Upload”, cel indicat în figura de mai jos. Dacă toți pașii au fost executați corect, programul va rula pe placă, aprinzând și stingând led-ul conectat la pin-ul digital 13.



2. Al doilea exemplu. Pini digitali de intrare, și folosirea interfeței seriale.

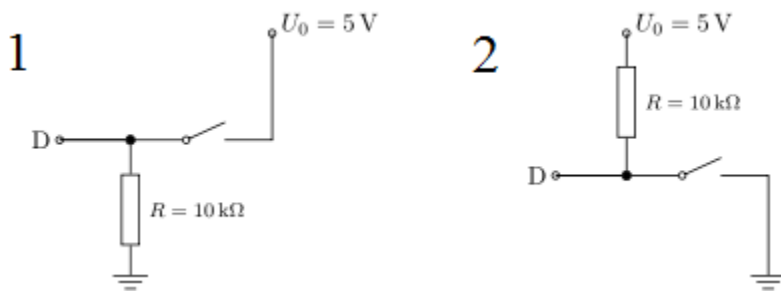
Ca dispozitiv elementar de intrare, vom utiliza blocul de butoane vizibil în figura de mai jos.



Blocul de butoane are 5 pini:

- GND (masă)
 - 4 pini de date (K1, K2...K4), indicând starea butoanelor (logic 0 = buton apăsat)
- Numele corespunzător apare atât în dreptul fiecărui pin cât și în dreptul fiecărui buton.

Schemele generale de conectare a unui buton la un microcontroller sunt prezentate mai jos. Prima schemă ilustrează folosirea unui resistor pull down pe când în a doua schemă se folosește un resistor de pull up.



Ne vom întrepta atenția asupra folosirii rezistoarelor de tip pull up intrucat sunt mai comune decat cele de tip pull down. In principiu functionarea lor este similara, diferenta fiind ca rezistorii pull up sunt conectati la 5V(sau VCC) pe cand rezistorii de tip pull down sunt conectati la masa (GND).

Cand se foloseste un rezistor de tip pull up, valoarea citita la pinul de intrare D cand butonul nu e apasat va fi 1 logic sau HIGH. Cu alte cuvinte, o cantitate mica de curent se scurge intre VCC si pinul de intrare (nu GND), asadar acest pin de intrare citeste o valoare aproape de VCC. In momentul in care apasam butonul, pinul de intrare se conecteaza direct la ground. Curentul trece prin rezistor si ajunge la ground facand pinul de intrare sa citeasca valoarea 0 sau LOW. Daca nu am folosi rezistenta de pull up, butonul ar conecta VCC la GND, lucru care ar rezulta intr-un scurt circuit (lucru pe care nu il dorim).

Blocul de butoane utilizat nu are rezistente, asadar pentru utilizarea lui avem 2 optiuni:

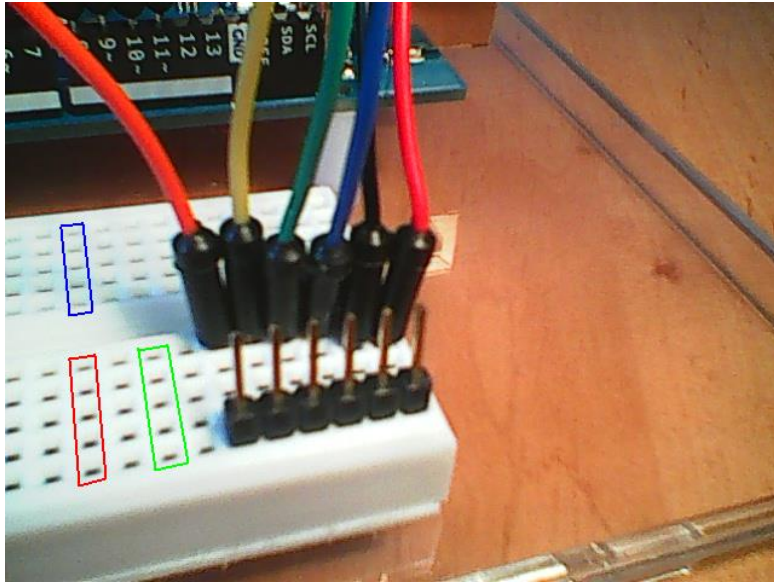
- Lipim rezistente auxiliare fiecarui buton
- Folosim rezistentele de tip pull up/ pull down prezente pe microcontroller

In prezenta lucrare vom folosi rezistentele oferite de placa de dezvoltare arduino. Acestea sunt activate folosind optiunea INPUT_PULLUP in stadiul de configurare a pinilor.

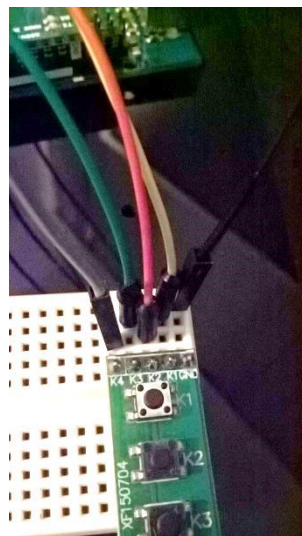
Pentru ieșire, vom utiliza interfața serială, putând astfel monitoriza ieșirile plăcii direct de pe calculator. **Toate conexiunile între un dispozitiv periferic, componenta, Bread-Board si placa Arduino le faceți doar cu cablul USB decuplat de la calculator!**

Pentru conectarea blocului de butoane, vom folosi placa de prototipizare (“Bread-Board”). Vom conecta cei patru pini de date la pinii digitali 4,5,6,7 si GND. Pentru a evita situația in care firele se pot desprinde la orice mica mișcare, ducând la funcționare greșită sau la defecțiuni (montaj de tip “paianjen”), vom folosi placa de prototipizare pentru a rigidiza montajul.

Placa de prototipizare are conectorii conectați electric în grupuri de câte 5, pe jumătăți de coloană, cum arată figura de mai jos.

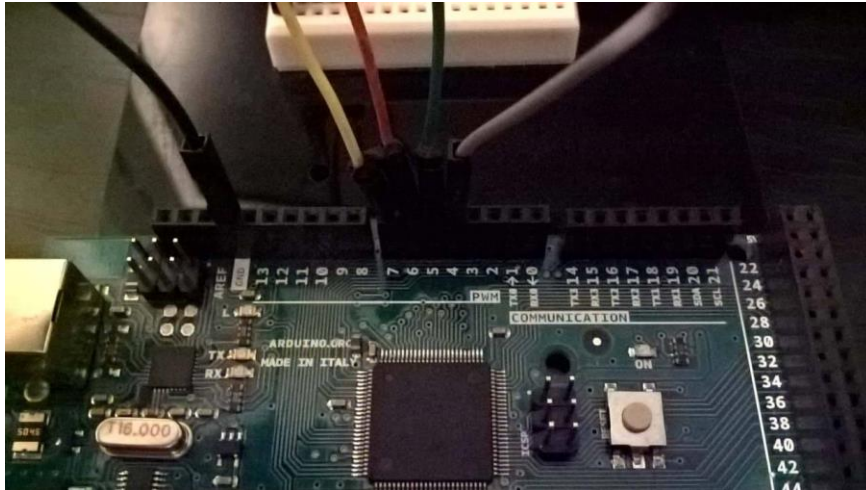


Plasati blocul de butoane in breadboard si apasati, astfel incat pini blocului de butoane sa intre in gaurile breadboardului. Introduceți apoi fire, pe fiecare semi-coloană în care aveți pini ai blocului de butoane, ca în figura de mai jos. Firul negru din figura reprezintă masa (GND), iar celelalte fire, de culori diferite, corespund butoanelor.



Capetele libere ale firelor le veți conecta la placa Arduino, astfel:

- Firele de semnal le conectați la pini digitali 4,5,6,7.
- Firul negru la GND.



In acest moment montajul fizic este gata. In continuare, deschidem mediul Arduino și creăm un program nou (File->New), ce va conține următorul cod:

```
// Citirea starii butoanelor conectate la pinii 4,5,6,7  
// afisarea prin interfata seriala  
// se transmite un numar care are ultimele 4 cifre starea butoanelor apasate  
// variabile pentru starea butoanelor
```

```
int b1;  
int b2;  
int b3;  
int b4;
```

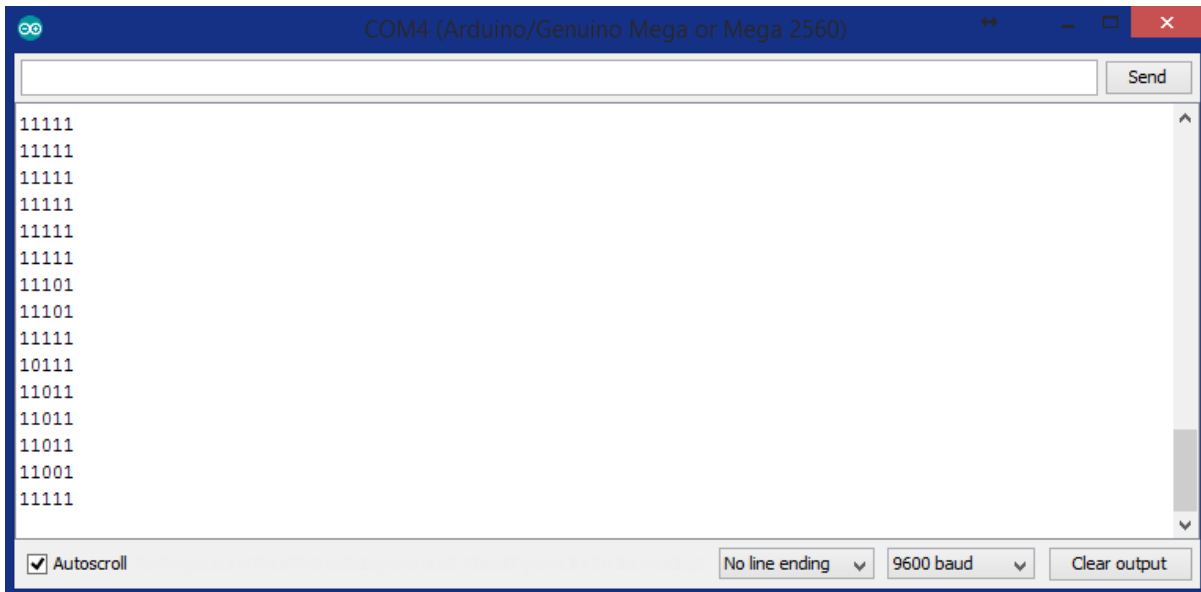
```
// variabila pentru compunerea numarului de transmis  
int stat = 0;
```

```
void setup() {  
  // configurare pini pentru butoane, intrare  
  pinMode(4, INPUT_PULLUP);  
  pinMode(5, INPUT_PULLUP);  
  pinMode(6, INPUT_PULLUP);  
  pinMode(7, INPUT_PULLUP);  
  // activare comunicatie seriala  
  Serial.begin(9600);  
}
```

```
void loop() {  
  // citire stare butoane  
  b1 = digitalRead(4);  
  b2 = digitalRead(5);  
  b3 = digitalRead(6);  
  b4 = digitalRead(7);  
  // compunere rezultat  
  stat = 10000 + b4 * 1000 + b3 * 100 + b2 * 10 + b1;  
  // transmisie  
  Serial.println(stat);  
}
```

```
// asteptare 0.5 sec  
delay(500);  
}
```

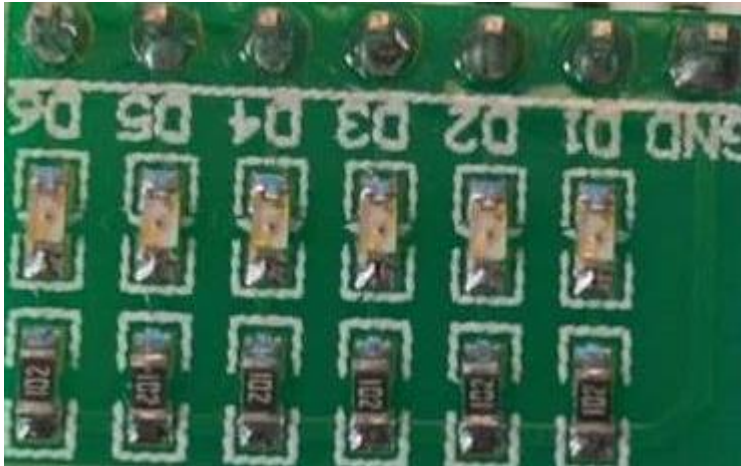
Conectați placa Arduino, și apăsați butonul “Upload”. Pentru vizualizarea rezultatului, deschideți utilitarul “Serial Monitor”, din meniul Tools->Serial Monitor. In fereastra utilitarului va fi afișat, la fiecare 0.5 secunde, un numar de 5 cifre (1XXXX), ultimele patru cifre fiind starea butoanelor, ca in figura de mai jos. Cand apasati pe un buton starea lui se transforma din 1 in 0.



Atenție! Serial Monitor trebuie închis înainte de a deconecta placa Arduino de la calculator. In caz contrar, este posibil ca portul serial virtual sa rămână blocat, și comunicarea ulterioară cu placa sa nu mai fie posibilă, decât după restartarea calculatorului!

3. Al treilea exemplu. Folosirea blocului de LED-uri pentru ieșire. Folosirea porturilor.

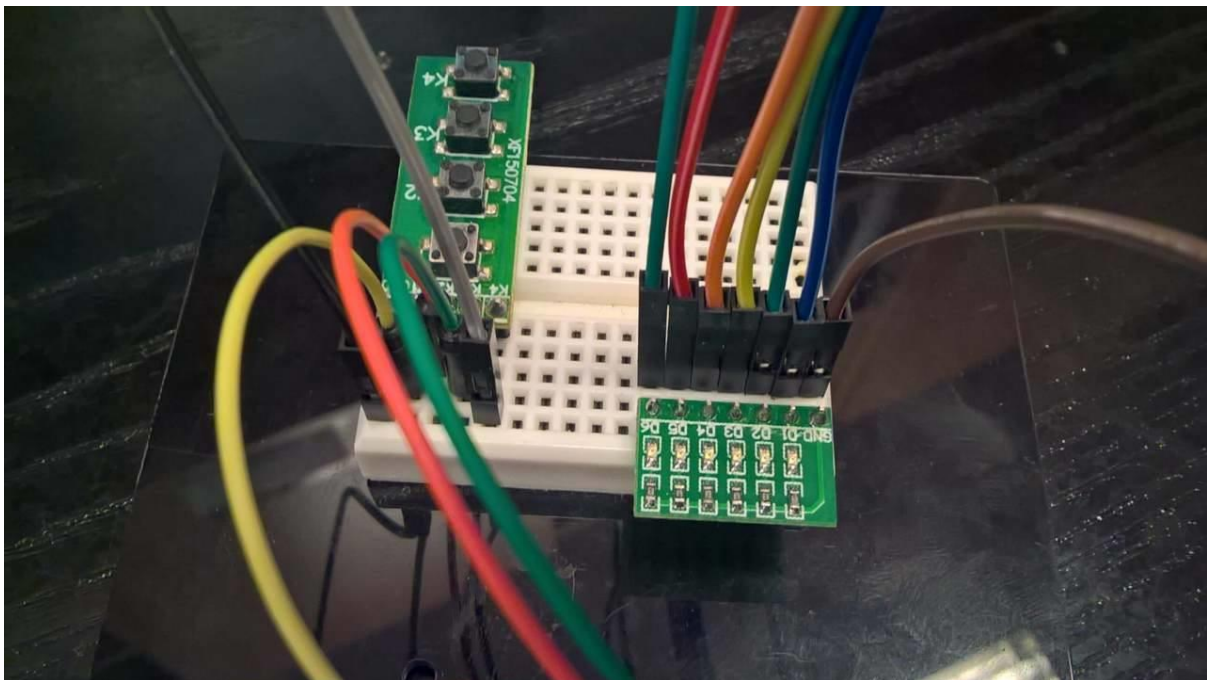
In al treilea exemplu, vom utiliza pentru ieșire un bloc de leduri. Acesta are 7 pini, dintre care unul este de alimentare (GND), și șase sunt semnalele pentru cele șase led-uri (valoarea logică 1 = led aprins). Observati ca ledurile au atasate rezistente pentru a le proteja sa nu fie arse.



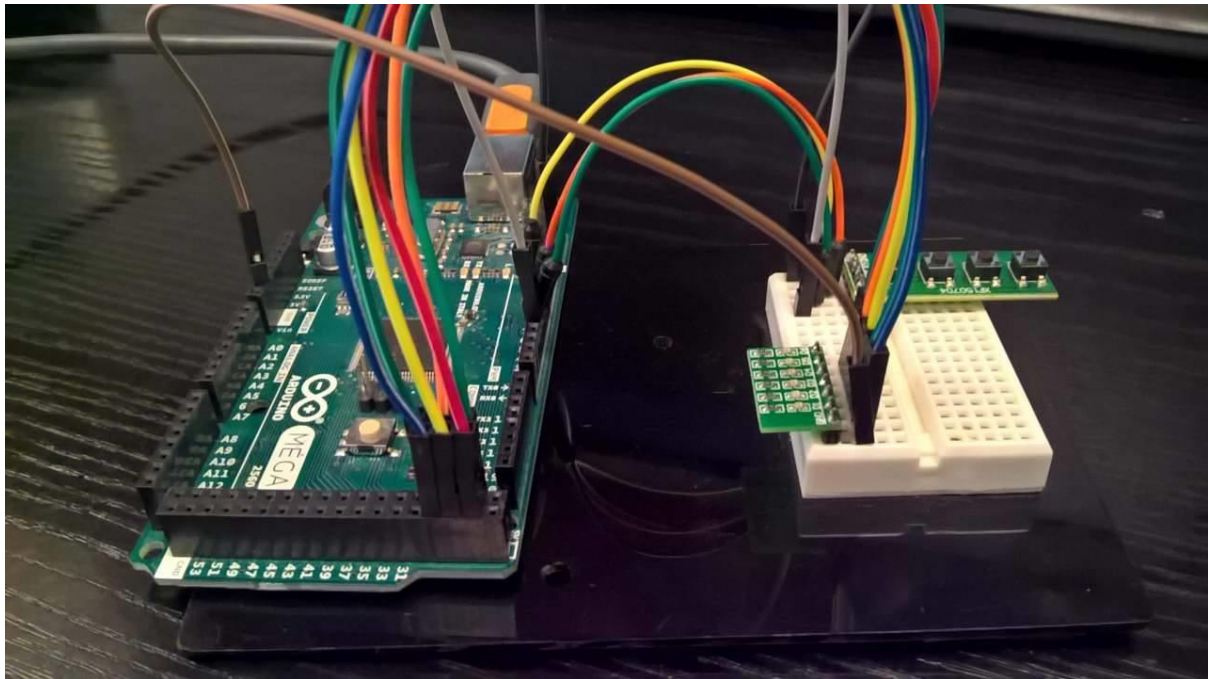
Pentru comanda mai rapidă a acestor led-uri, vom folosi porturile microcontrollerului ATmega 2560.

Primul pas constă în pregătirea celor șapte poziții pe breadboard, corespunzătoare celor 7 pini ai blocului de led-uri (D6...D1, GND). Vom folosi 7 fire, pe care le vom conecta ca în figura de mai jos.

Vom introduce blocul de led-uri direct în breadboard, ca în figura de mai jos. **Apăsați ferm, fără a lovi.**



Firele de semnal le vom conecta la pinii digitali 22, 23, 24, 25, 26, 27 ai plăcii Arduino, corespunzătorii biților din portul A (PA5, PA4, PA3, PA2, PA1 și PA0), ca în figura de mai jos.



Corespondența dintre pini și porturi este descrisă în documentul “ATmega2560-Arduino Pin Mapping”, ce poate fi vizualizat aici: <http://arduino.cc/en/Hacking/PinMapping2560> , sau în prima figură a acestui document. Pentru portul A, avem:

PA7 (AD7)	Digital pin 29
PA6 (AD6)	Digital pin 28
PA5 (AD5)	Digital pin 27
PA4 (AD4)	Digital pin 26
PA3 (AD3)	Digital pin 25
PA2 (AD2)	Digital pin 24
PA1 (AD1)	Digital pin 23
PA0 (AD0)	Digital pin 22

Pentru pornirea și oprirea ledurilor se folosește montajul cu butoane realizat în exercitiul precedent.

Nota:

In viitor când vom avea mai multe componente vom încerca să identificăm firele cu semnale comune între componente. Acestea vor fi grupate pe breadboard (puse pe aceeași coloană). În exemplul nostru semnalele de ground ale celor două module puteau fi grupate, putând astfel să folosim un singur fir de masă.

După realizarea montajului fizic, creați un nou proiect Arduino, și adăugați în el următorul cod:

```
// Citirea stării butoanelor conectate la pinii 4,5,6,7
// afisarea pe LED-uri conectate la PORTA
```

```

// variabile pentru starea butoanelor
int b1;
int b2;
int b3;
int b4;

// variabila pentru compunerea numarului de transmis
unsigned char stat = 0;

void setup() {
  // configurare pini pentru butoane, intrare
  pinMode(4, INPUT_PULLUP);
  pinMode(5, INPUT_PULLUP);
  pinMode(6, INPUT_PULLUP);
  pinMode(7, INPUT_PULLUP);
  // activare PORTA , ca iesire,
  DDRA = 0b11111111;
}

void loop() {
  // citire stare butoane
  b1 = digitalRead(4);
  b2 = digitalRead(5);
  b3 = digitalRead(6);
  b4 = digitalRead(7);
  // compunere rezultat
  // fiecare LED e controlat de 1 buton, unele butoane sunt duplicate
  stat = (b4<<5) | (b3<<4) | (b4<<3) | (b3<<2) | (b2<<1) | b1;
  // afisare pe LED-uri, legate la port a
  PORTA = stat;
  // asteptare 50 ms
  delay(50);
}

```

Lucru individual:

1. Calculati valoarea rezistentei de pull up, in cazul in care nu am folosi optinea INPUT_PULLUP din arduino ci o rezistenta externa, si am dori sa limitam curentul la 1 mA.
2. Rulați exemplele 1 si 2.
3. Modificați exemplul 2, pentru ca placa să transmită la calculator informații diferite în funcție de butonul apăsat. De exemplu, la apăsarea unui buton puteți transmite numărul de milisekunde de când a fost pornit programul, apelând funcția millis(), la apăsarea unui alt buton puteți transmite numărul de secunde (millis()/1000), la apăsarea altui buton un mesaj text, etc.
4. Rulați exemplul 3.
5. Modificați exemplul 3 pentru a afișa pe led-uri o animație (aprindere din 2 in 2, baleiere, etc), care se modifica la apăsarea unui buton.

Materiale suplimentare: http://users.utcluj.ro/~rdanescu/teaching_pmp.html
Maparea pinilor la Arduino Mega 2560: <http://arduino.cc/en/Hacking/PinMapping2560>