Creșterea abilităților de programare cu microcontrolere

Conceput de Mirela TIBU

Liceul Teoretic de Informatică ,,Grigore Moisil'', Iași

Un set de instrumente pentru formatori pentru a stimula utilizarea abilităților STEM
Aplicații pentru microcontroler

Co-funded by the Erasmus+ Programme of the European Union

Project No. 2019-1-RO01-KA202-063965
This project has been funded with support from the European Commission. The content reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Continut

Informatica

Project No. 2019-1-RO01-KA202-063965
This project has been funded with support from the European Commission. The content reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Co-funded by the Erasmus+ Programme of the European Union

Scop

Exprimarea unui mod creativ de gândire, în structurarea și rezolvarea problemelor

Formarea de obiceiuri de a utiliza concepte și metode algoritmice specifice computerului în abordarea unei varietăți de probleme

Manifestarea atitudinilor față de știință și cunoaștere

Manifestarea de inițiativă și dorință de a aborda diverse sarcini

Project No. 2019-1-RO01-KA202-063965
This project has been funded with support from the European Commission. The content reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Co-funded by the Erasmus+ Programme of the European Union

Descriere

- Abordare ITC - Microprocessor VS Microcontroller
  - ✓ identificarea zonelor în care computerele/sistemele încorporate sunt utilizate în viața de zi cu zi
  - ✓ descrierea arhitecturii hardware pentru un computer și un sistem încorporat
  - ✓ compararea caracteristicilor microprocesoarelor și microcontrolerelor
- Abordare programare - Înțelegerea conceptelor de programare prin utilizarea aplicațiilor Microcontrolere
  - ✓ instrucțiuni de programare structurată – decizii, bucle (IF, WHILE, FOR)
  - ✓ declararea și apelarea funcțiilor void și non-void
  - ✓ folosind matrice în aplicații
  - ✓ analizarea funcționalității dispozitivelor Arduino pentru a recunoaște etapele de execuție a instrucțiunilor de programare

Project No. 2019-1-RO01-KA202-063965
This project has been funded with support from the European Commission. The content reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Co-funded by the Erasmus+ Programme of the European Union

Obiectivele de invatare

- Identificarea aplicațiilor informatice în viața socială - conștientizarea impactului sistemelor încorporate în viața de zi cu zi
  - Identificarea asemănărilor și diferențelor dintre un microprocesor și un microcontroler în arhitectura sistemelor de calcul și încorporate
  - Exersarea implementării elementelor de programare structurată - decizii, bucle, funcții; reprezentarea și utilizarea datelor matrice
  - Vizualizarea efectului executării diverselor secvențe de programe prin dispozitive bazate pe microcontrolere

Project No. 2019-1-RO01-KA202-063965
This project has been funded with support from the European Commission. The content reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Co-funded by the Erasmus+ Programme of the European Union

Metodologii de invatare

- Explicatii
- Demonstratie
- Conversatie
- Algoritmizare
- Implementare

Project No. 2019-1-RO01-KA202-063965
This project has been funded with support from the European Commission. The content reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Co-funded by the Erasmus+ Programme of the European Union

Grup tinta

Elevi de liceu – Clasele a 9a si a 10a

Project No. 2019-1-RO01-KA202-063965
This project has been funded with support from the European Commission. The content reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Co-funded by the Erasmus+ Programme of the European Union

Calculatoare si sisteme incorporate

Project No. 2019-1-RO01-KA202-063965
This project has been funded with support from the European Commission. The content reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Co-funded by the Erasmus+ Programme of the European Union

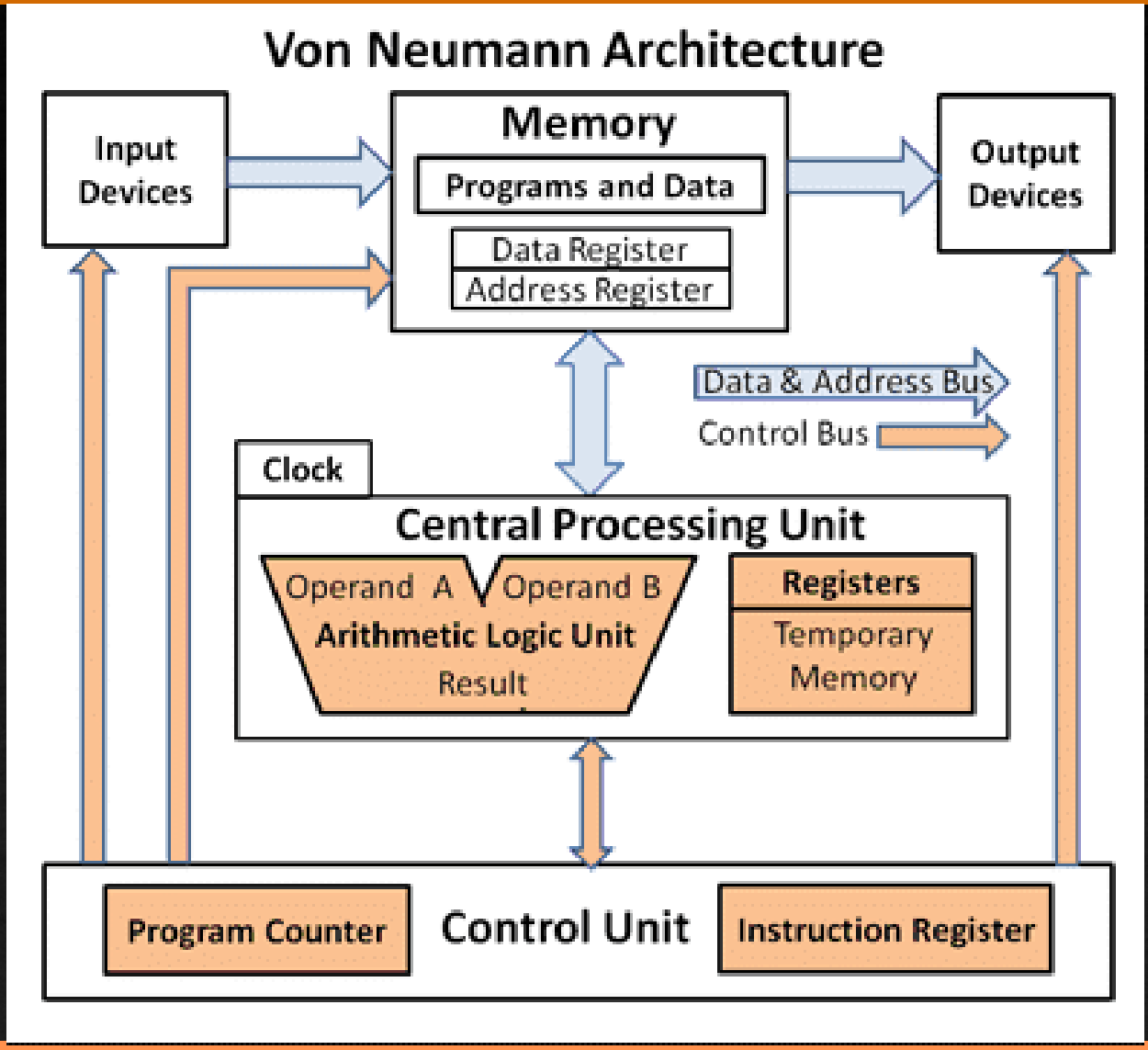
Computere si viata de zi cu zi

- Calculatoarele fac parte din viața noastră de zi cu zi.
- Computer = Hardware + Software
  - ✓ Hardware – component fizice
  - ✓ Software - programe, proceduri și rutine care îi spun unui computer ce trebuie să facă
  - ✓UNDE folosim computere?

Project No. 2019-1-RO01-KA202-063965
This project has been funded with support from the European Commission. The content reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.
Co-funded by the Erasmus+ Programme of the European Union

Computer Hardware Architecture

- ✓ CPU = Microprocessor - “creierul” computerului nostru – efectuează toate operațiile aritmetice și logice (ALU) și controlează toate activitățile sistemului
- ✓ Memory Unit – stocati date si programe
  - RAM – Memorie cu acces aleator
  - ROM – Memorie numai pentru cifre
- ✓ Dispozitive de intrare si iesire



Project No. 2019-1-RO01-KA202-063965
This project has been funded with support from the European Commission. The content reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Co-funded by the Erasmus+ Programme of the European Union

Sisteme incorporate in viata de zi cu zi

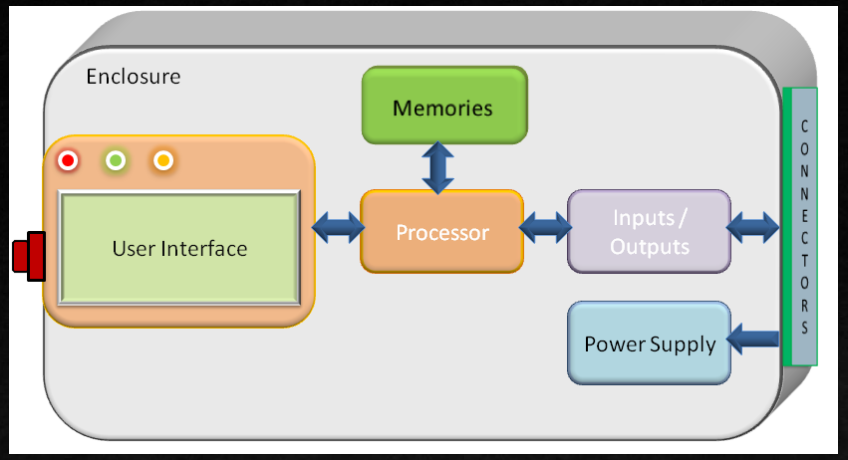
- Un sistem incorporat este un computer cu scop special care este utilizat in interiorul unui dispozitiv
- se bazează pe microcontroler care este un cip optimizat pentru a controla dispozitivele electronice; este stocat într-un singur circuit integrat, dedicat îndeplinirii unei anumite sarcini și executării unei aplicații specifice
- Unde folosim sistemele incorporate

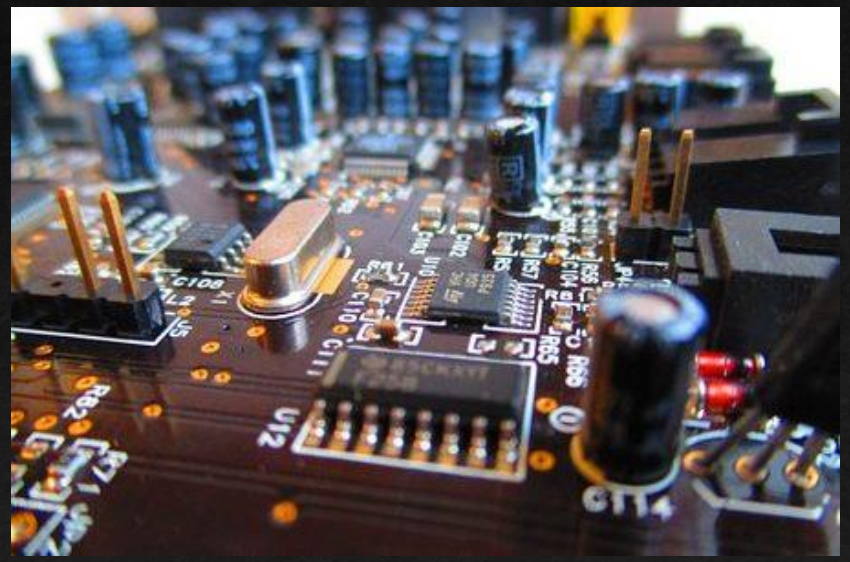
Project No. 2019-1-RO01-KA202-063965
This project has been funded with support from the European Commission. The content reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

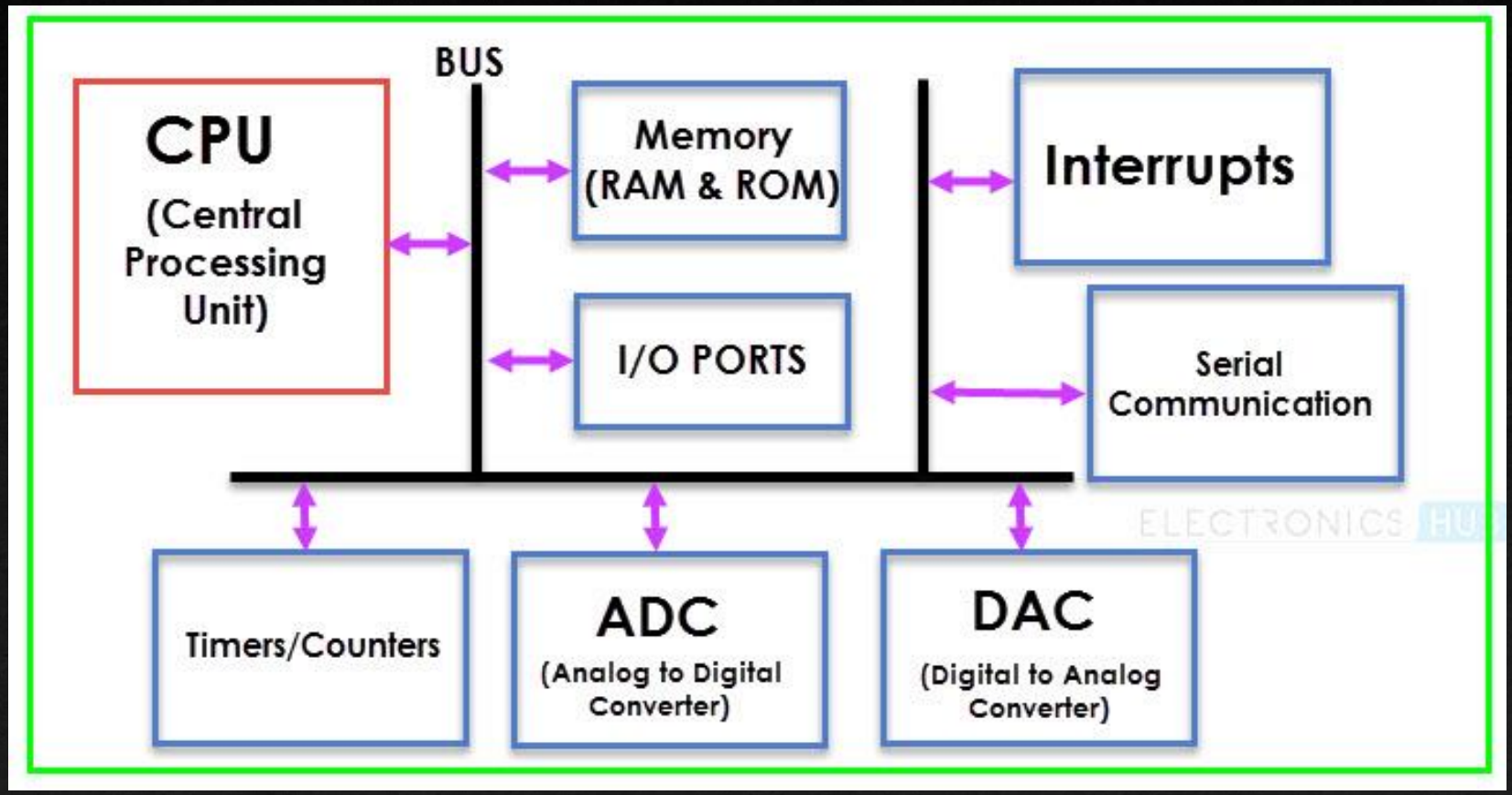
Co-funded by the Erasmus+ Programme of the European Union

Arhitectura sistemelor încorporate

- CPU – este un microcontroler sau un microprocesor







Project No. 2019-1-RO01-KA202-063965
This project has been funded with support from the European Commission. The content reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Co-funded by the Erasmus+ Programme of the European Union

Care este mai bun?

Microprocessor	Microcontroller
Este inima unui sistem de calculator .	Inima unui sistem incorporat
este doar un procesor, astfel încât componentele de memorie și I/O trebuie conectate extern	are un procesor împreună cu memorie internă și componente I/O
memoria și I/O trebuie conectate extern, astfel încât circuitul devine mare	Memoria și I/O sunt deja prezente, iar circuitul intern este mic
Poate fi folosit in sisteme compacte	Este folosit in sisteme compacte
Costul intregului sistem este ridicat	Costul intregului sistem este mic
datorită componentelor externe, consumul total de energie este mare. nu este ideal pentru dispozitivele care funcționează cu energie stocată, cum ar fi bateriile.	deoarece componentele externe sunt scăzute, consumul total de energie este mai mic. Deci poate fi folosit cu dispozitive care funcționează cu energie stocată, cum ar fi bateriile
majoritatea dintre ele nu au funcții de economisire a energiei	majoritatea oferă modul de economisire a energiei
Folosit in calculatoarele personale	utilizat în principal în sistemele încorporate.
Sunt bazate pe modelul lui Von Neumann	se bazează pe arhitectura Harvard

Project No. 2019-1-RO01-KA202-063965
This project has been funded with support from the European Commission. The content reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Co-funded by the Erasmus+ Programme of the European Union

Which one is better?

au un număr mai mic de registre, deci mai multe operațiuni sunt bazate pe memorie	au mai multe registre, astfel încât programele sunt mai ușor de scris
este o unitate centrală de procesare pe un singur cip integrat pe bază de siliciu	este un produs secundar al dezvoltării microprocesoarelor cu un procesor împreună cu alte periferice
nu are RAM, ROM, unități de intrare-ieșire, temporizatoare și alte periferice pe cip	are un procesor împreună cu RAM, ROM și alte periferice încorporate pe un singur cip
folosește o magistrală externă pentru interfața cu RAM, ROM și alte periferice	folosește o magistrală de control intern
Sistemele bazate pe microprocesoare pot rula la o viteză foarte mare datorită tehnologiei implicate	Sistemele bazate pe microcontroler rulează până la 200MHz sau mai mult, în funcție de arhitectură
este utilizat pentru aplicații de uz general care vă permit să gestionați o mulțime de date	este utilizat pentru sisteme specifice aplicației
Este complex și costisitor, cu un număr mare de instrucțiuni de procesat	Este simplu și ieftin, cu un număr mai mic de instrucțiuni de procesat

Project No. 2019-1-RO01-KA202-063965
This project has been funded with support from the European Commission. The content reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Co-funded by the Erasmus+ Programme of the European Union

Declarația IF, declarație și apel de funcții void și definiție const

Task: Programați un dispozitiv capabil să citească starea unui potențiometru (o intrare analogică) și aprinde un LED numai dacă potențiometrul.
Trebuie să imprime valoarea analogică indiferent de nivel.

```
const int analogPin = A0;
   // pin that the sensor is attached to
const int ledPin = 13;
   // pin that the LED is attached to
const int valmin = 400;
   // an arbitrary valmin level

void setup() {
   // initialize the LED pin as an output:
  pinMode(ledPin, OUTPUT);
   // initialize serial communications:
  Serial.begin(9600); }
```

```
void loop() {
  // read the value of the potentiometer:
  int analogValue = analogRead(analogPin);

  // if the analog value is high enough,
  // turn on the LED:
  if (analogValue > valmin) {
    digitalWrite(ledPin, HIGH);
  } else {
    digitalWrite(ledPin, LOW);
  }
  // print the analog value:
  Serial.println(analogValue);
  delay(1);    // delay in between reads
}
```


Project No. 2019-1-RO01-KA202-063965
This project has been funded with support from the European Commission. The content reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Co-funded by the Erasmus+ Programme of the European Union

Înțelegerea conceptelor de programare prin utilizarea aplicațiilor Microcontrolere

Variable definition

```
<typeVar> nameVar [= value];
```

declară o variabilă de un anumit tip, vrăjitoarea va determina mărimea valorilor, lungimea și tipul reprezentării memoriei
Usual types of variables used in Arduino apps

```
int = tip numeric pentru variabile/constante;
```
este reprezentat pe 4 octeți și poate stoca valori intre aprox. -2*109 ... 2*109

```
bool = tip boolean pentru variabile/constante;
```
este reprezentat în 1 octet și poate stoca valori false (0) și adevărate (1)

Constant definition

```
const <typeConst> nameConst = value;
```

setează o constantă de typeConst cu o valoare specifica

Decision Structure

```
if (Condition) {
  instructions_A
   // do instructions_A if the
   // Condition is true
} else {
 instructions_B
  // do instructions_A if the
  // Condition is false
}
```

void Functions – declaration

```
void nameFunction(list of formal parameters)
{ declaration of local variables
  instructions
}
```


Project No. 2019-1-RO01-KA202-063965
This project has been funded with support from the European Commission. The content reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Co-funded by the Erasmus+ Programme of the European Union

void Functions – declaration

```
void nameFunction(formal parameters)
{declararea variabilelor locale
  instrucțiuni}
```

non-void Functions - declaration

```
resultType nameFunction(formal parameters)
{ declararea variabilelor locale
  instrucțiuni
  expresie returnată;  //
}
```

unde parametrii formali este o listă de tipuri și denumiri de parametri utilizați în instrucțiunile funcției

Call of a function

```
nameFunction(list of actual parameters)
```

- funcții void - Apelul este o instrucțiune
- funcție non-void – Apelul este un operarand în expresie cu același tip ca resultType

! parametrii formali și actuali trebuie să aibă același tip, număr și trebuie să fie în aceeași ordine

Project No. 2019-1-RO01-KA202-063965
This project has been funded with support from the European Commission. The content reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Co-funded by the Erasmus+ Programme of the European Union

Înțelegerea conceptelor de programare prin utilizarea aplicațiilor Microcontrolere

Arduino specific functions

setup()

funcția void - apelată când începe o schiță și va rula o singură dată, după fiecare pornire sau resetare a plăcii Arduino, similar cu main(). Folosiți-l pentru a inițializa variabile, moduri de fixare, pentru a începe să utilizați biblioteci etc.

loop()

funcția void - bucle consecutiv, permițând programului Arduino să se schimbe și să răspundă Este apelată după funcția setup(), care inițializează și setează valorile inițiale.

pinMode(pin, mode)

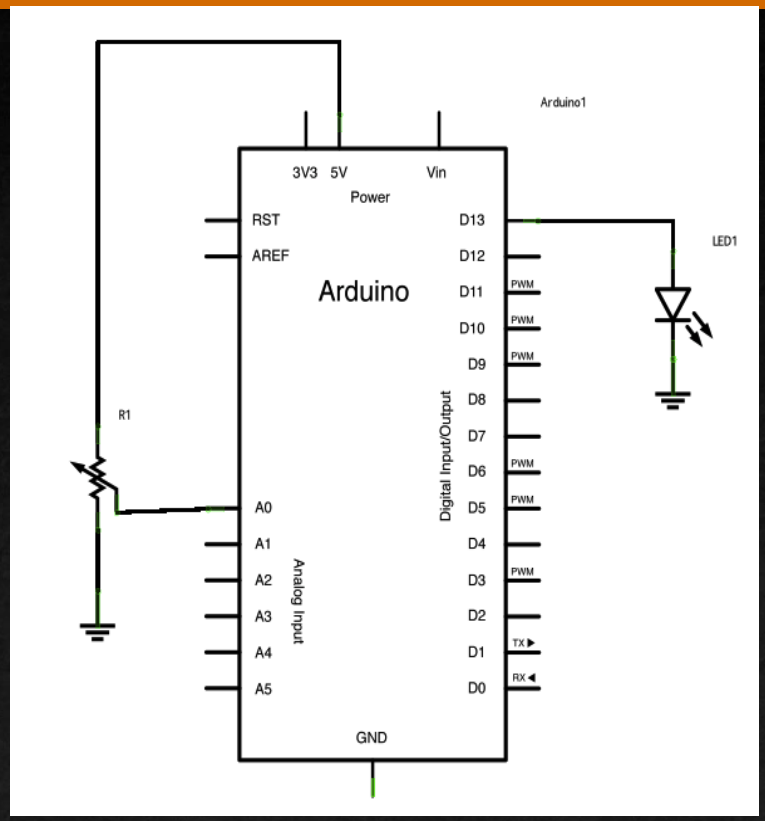
void function with parametrers

- pin: the Arduino pin number to set the mode off
- mode: INPUT, OUTPUT, or INPUT_PULLUP

delay(milisec)

void function with parametrers

- milisec: number of miliseconds to pause the program



electrical schema of the Arduino device

Project No. 2019-1-RO01-KA202-063965
This project has been funded with support from the European Commission. The content reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Co-funded by the Erasmus+ Programme of the European Union

Arduino specific functions

digitalWrite(pin, value)

void function with parametrers

- pin: the Arduino pin number.
- value: HIGH or LOW

digitalRead(pin)

function with parametrers

- pin: the Arduino pin num
- return value: HIGH or LOW

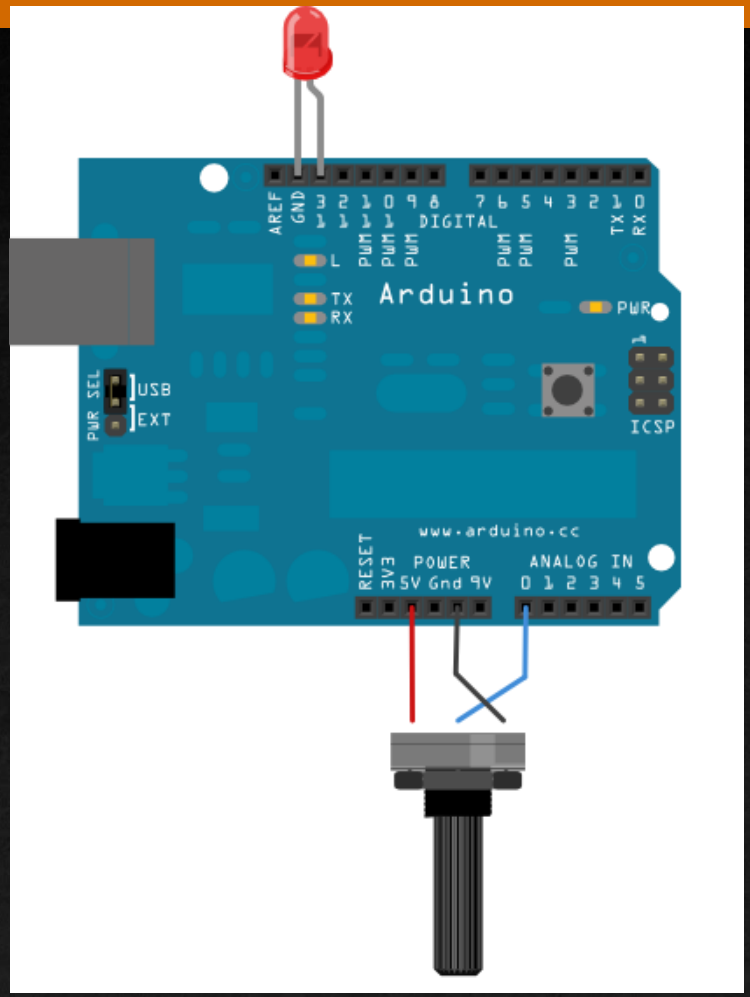
analogRead(pin)

int function with parametrer

- pin: the name of the analog input pin to read from (A0 to A5 on most boards)
- Returns the analog reading on the pin.

Arduino Pin Levels Constants HIGH and LOW

pin	INPUT	OUTPUT
HIGH	voltage > 3.0V	5V
LOW	voltage > 3.0V	0V



Arduino device

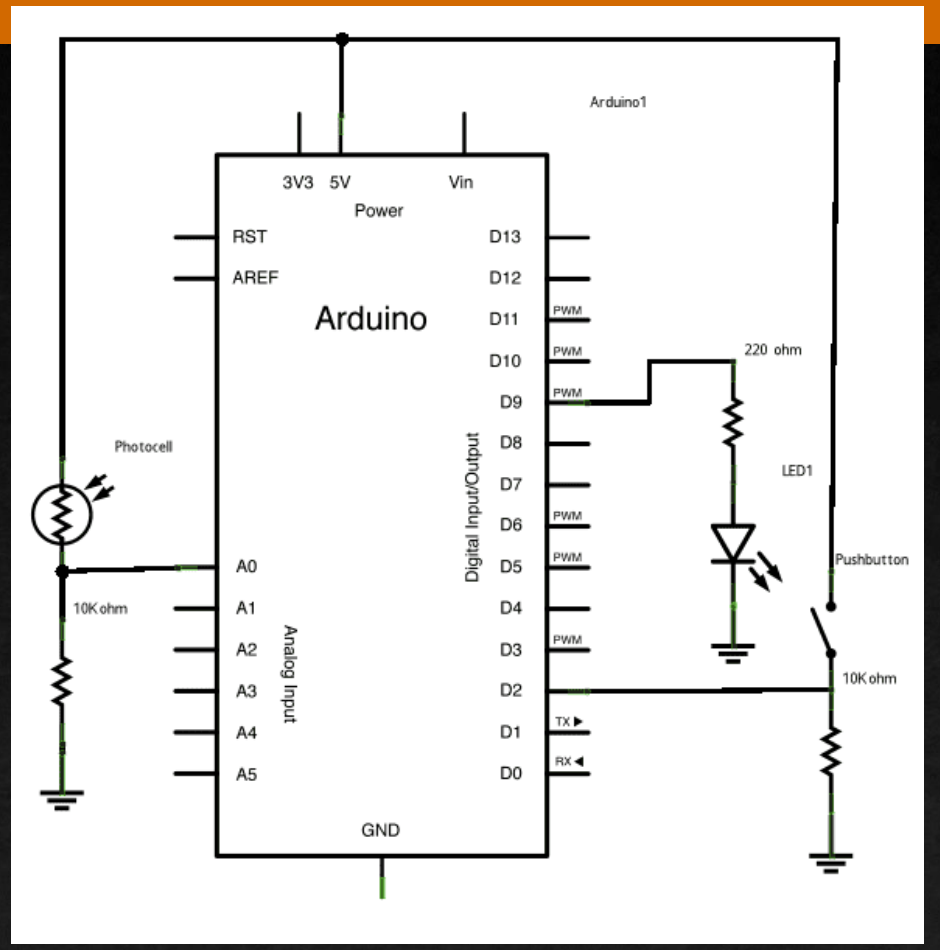
Project No. 2019-1-RO01-KA202-063965
This project has been funded with support from the European Commission. The content reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Co-funded by the Erasmus+ Programme of the European Union

Declaratia WHILE

Task: Programați un dispozitiv capabil să citească timp de cinci secunde intrarea senzorului și să calibreze prin definirea valorilor minime și maxime așteptate pentru citirile efectuate în timpul buclei.

```
const int sensorPin = A0;
     // pin that the sensor is attached to
const int ledPin = 9;
     // pin that the LED is attached to
     // variables:
int sensorValue = 0;
    // the sensor value
int sensorMin = 1023;
    // minimum sensor value
int sensorMax = 0;
    // maximum sensor value
```



Schema for Arduino device

Project No. 2019-1-RO01-KA202-063965
This project has been funded with support from the European Commission. The content reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Co-funded by the Erasmus+ Programme of the European Union

Declaratia WHILE

```
void setup() {
  // turn on LED to signal the start
  // of the calibration period:
     pinMode(13, OUTPUT);
     digitalWrite(13, HIGH);
  // calibrate during the first five seconds
  while (millis() < 5000) {
        sensorValue = analogRead(sensorPin);
        if (sensorValue > sensorMax){
            sensorMax = sensorValue;
        }
        if (sensorValue < sensorMin){
           sensorMin = sensorValue;
        }
   }
  digitalWrite(13, LOW);    // signal the end of the calibration period
}
```


Project No. 2019-1-RO01-KA202-063965
This project has been funded with support from the European Commission. The content reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Co-funded by the Erasmus+ Programme of the European Union

Determinați valoarea minimă și maximă dintr-un set de valori

Algorithm

Step 1. Setați variabilele pentru senzorMin cu valoarea maximă posibilă iar senzorul Max cu minim valoare posibilă
Step 2. Comparați valoarea curentă cu senzorMinand, dacă este mai mic, actualizare senzorMin
Step 3. Comparați valoarea curentă cu sensorMax și, dacă este mai mare, actualizare senzorMax

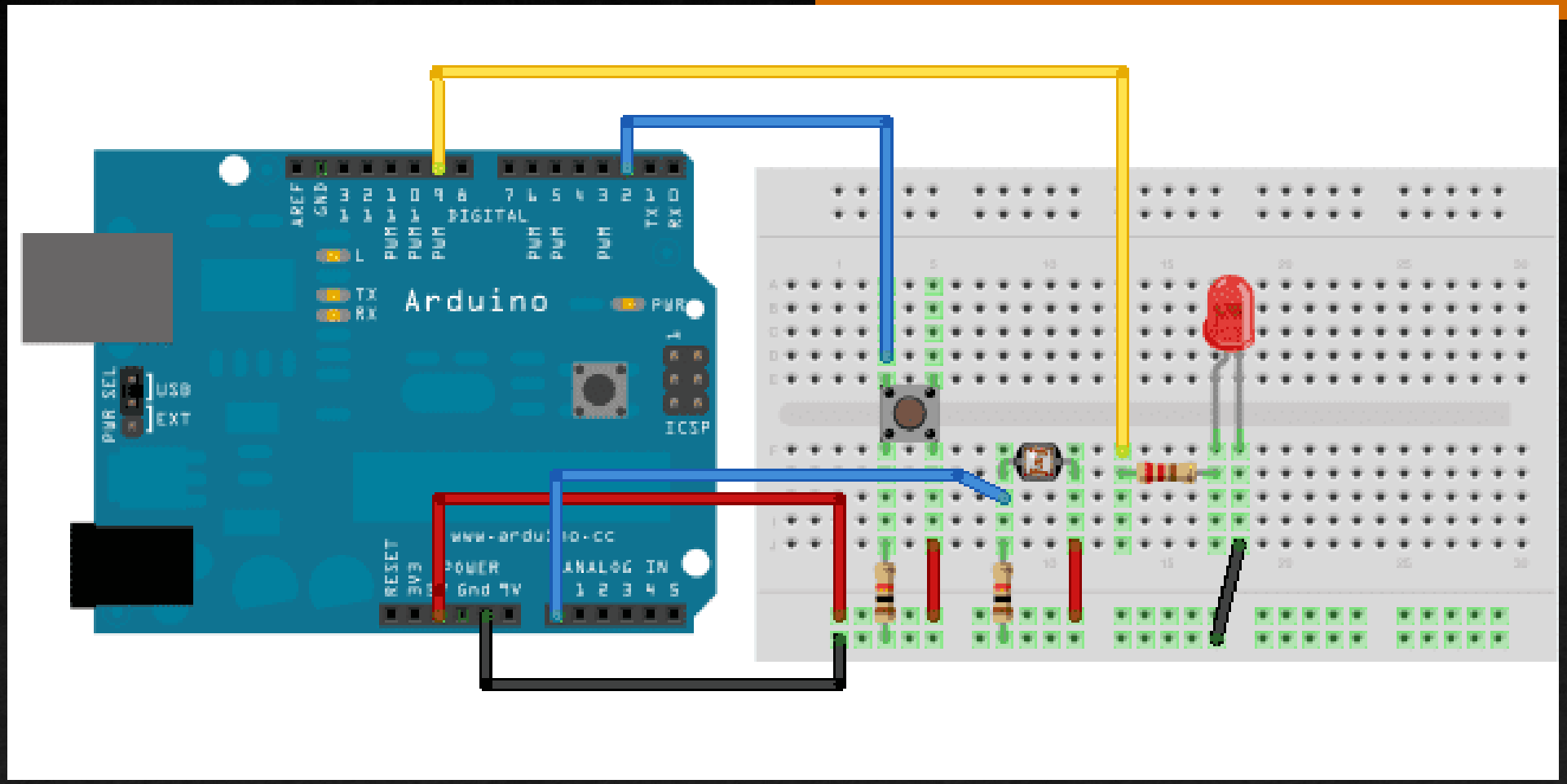
Declaratia WHILE **while (Conditia) {**
instructiunea _A
}

Executia

Step 1. Conditia este evaluata
Step 2. Daca conditia este adevarata
2.1. Instructiunea A va fi executata
2.2. Mergem la pasul 1.
Daca conditia este falsa, programul iese din WHILE

Project No. 2019-1-RO01-KA202-063965
This project has been funded with support from the European Commission. The content reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.
Co-funded by the Erasmus+ Programme of the European Union

Înțelegerea conceptelor de programare prin utilizarea aplicațiilor Microcontrolere



Arduino device explaining WHILE

Project No. 2019-1-RO01-KA202-063965
This project has been funded with support from the European Commission. The content reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Co-funded by the Erasmus+ Programme of the European Union

Înțelegerea conceptelor de programare prin utilizarea aplicațiilor Microcontrolere

manipularea instrucțiunilor FOR și ARRAY-urilor

```
int timer = 100;
   // Cu cât numărul este mai mare,
   // cu atât timpul este mai lent.
int ledPins[] = { 2, 7, 4, 6, 5, 3};
   // o serie de numere de pin la care sunt atașate LED-
urile
int pinCount = 6;
  // numărul de pini (lungimea matricei)
void setup() {
  // elementele matricei sunt numerotate de la 0 la
(pinCount-1)
  // utilizați o buclă for pentru a inițializa fiecare pin
ca ieșire:
  pentru (int thisPin = 0; thisPin < pinCount; thisPin++) {
    pinMode(ledPins[thisPin], OUTPUT);
  }
}
```

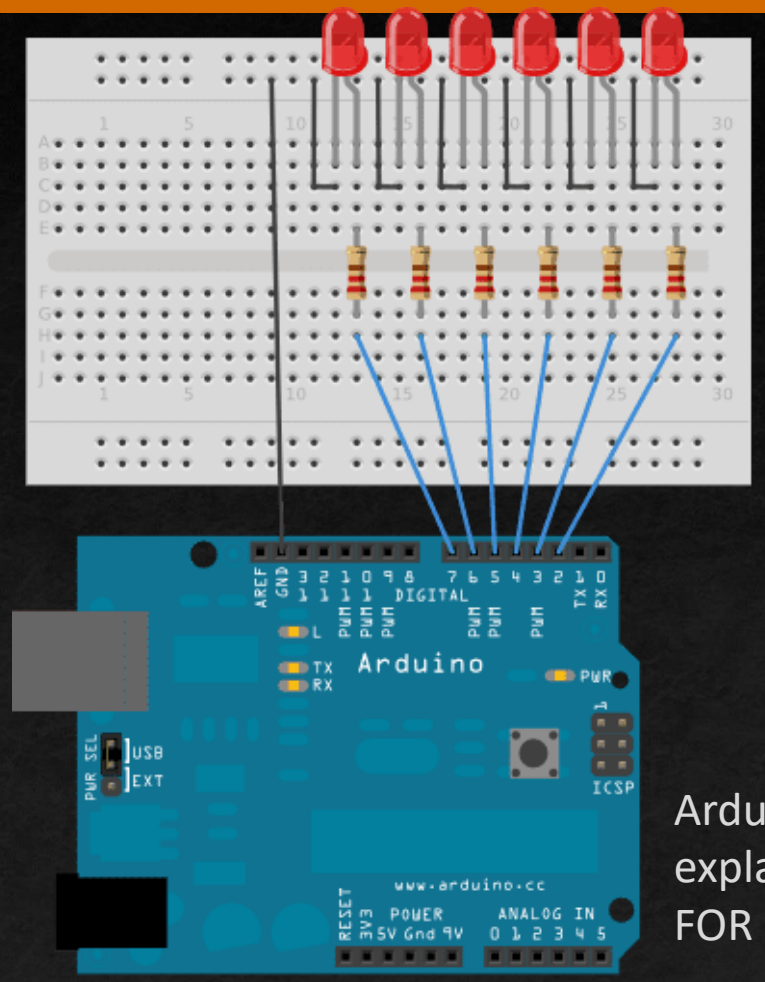
Task: Programați un dispozitiv capabil să aprindă a serie de LED-uri atașate la pini ale căror numere nu sunt nici învecinate, nici neapărat secvențiale. Pentru a face acest lucru, numerele PIN vor fi stocate într-un ARRAY și apoi folosiți buclele FOR pentru a repeta peste matrice.

Project No. 2019-1-RO01-KA202-063965
This project has been funded with support from the European Commission. The content reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Co-funded by the Erasmus+ Programme of the European Union

functia FOR si manipularea vectorilor

```
void loop() {
// buclă de la pinul cel mai de jos la cel mai înalt :
 for (int thisPin = 0; thisPin < pinCount; thisPin++) {
   digitalWrite(ledPins[thisPin], HIGH);
      // turn the pin on:
   delay(timer);
   digitalWrite(ledPins[thisPin], LOW);
      // turn the pin off:
 }
      // buclă de la cel mai înalt pin la cel mai de
jos :
 for (int thisPin=pinCount-1; thisPin >= 0; thisPin--){
      // turn the pin on:
    digitalWrite(ledPins[thisPin], HIGH);
    delay(timer);
      // turn the pin off:
    digitalWrite(ledPins[thisPin], LOW);
 }
}
```

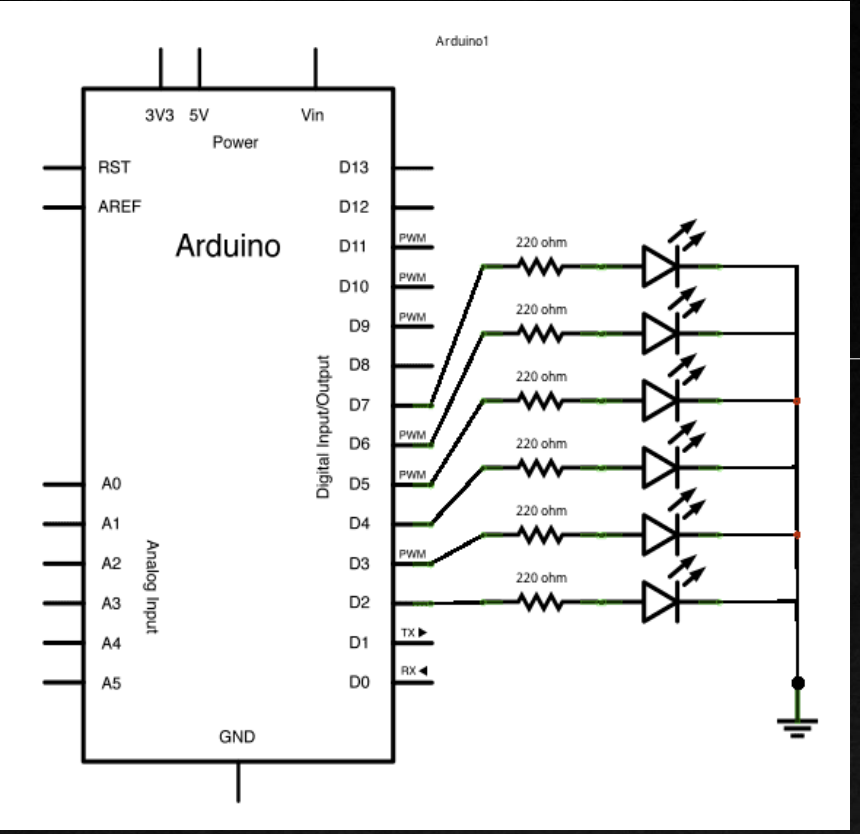


Arduino device explaining FOR loop

Project No. 2019-1-RO01-KA202-063965
This project has been funded with support from the European Commission. The content reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Co-funded by the Erasmus+ Programme of the European Union

Înțelegerea conceptelor de programare prin utilizarea aplicațiilor Microcontrolere



Schema for Arduino device

FOR statement

```
for(int counter = initialVal; counter <= finalVal; counter++) {
     instructions_A
}
```

Execution

Step 1. The counter is set with initialValue

Step 2. If counter <= finaValue is True

    2.1. Instructions A will be executed

    2.2. counter is increased with 1

    2.3. Go to Step 2

Step 3. If counter <= finaValue is False, the program execution leave the loop

Project No. 2019-1-RO01-KA202-063965
This project has been funded with support from the European Commission. The content reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Co-funded by the Erasmus+ Programme of the European Union

Înțelegerea conceptelor de programare prin utilizarea aplicațiilor Microcontrolere

Organizing data in ARRAYs

Vectori = o colecție de date de același tip, organizată într-o zonă de memorie adiacentă și referită cu un singur nume, care este un pointer (adresa de memorie) a primului element din matrice.

Declaratii:

```
valuesType arrayName[numberOfElements];
```

Initialization:

- Impreuna cu declaratiile
  `int ledPins[] = { 2, 7, 4, 6, 5, 3};`
- Dupa atribuire
  `digitalWrite(ledPins[thisPin], HIGH);`
- Dupa citirea valorilor de intrare

Referindu-se la o anumită valoare din matrice

A[expressionIndex]

expressionIndex este un număr întreg din [0,count-1], indicând poziția elementului în matrice

Analizarea ARRAY pentru a analiza și procesa elementele acestuia:

left-rigt

```
for (int it = 0; it < count; it++)
    process(A[it]);
```

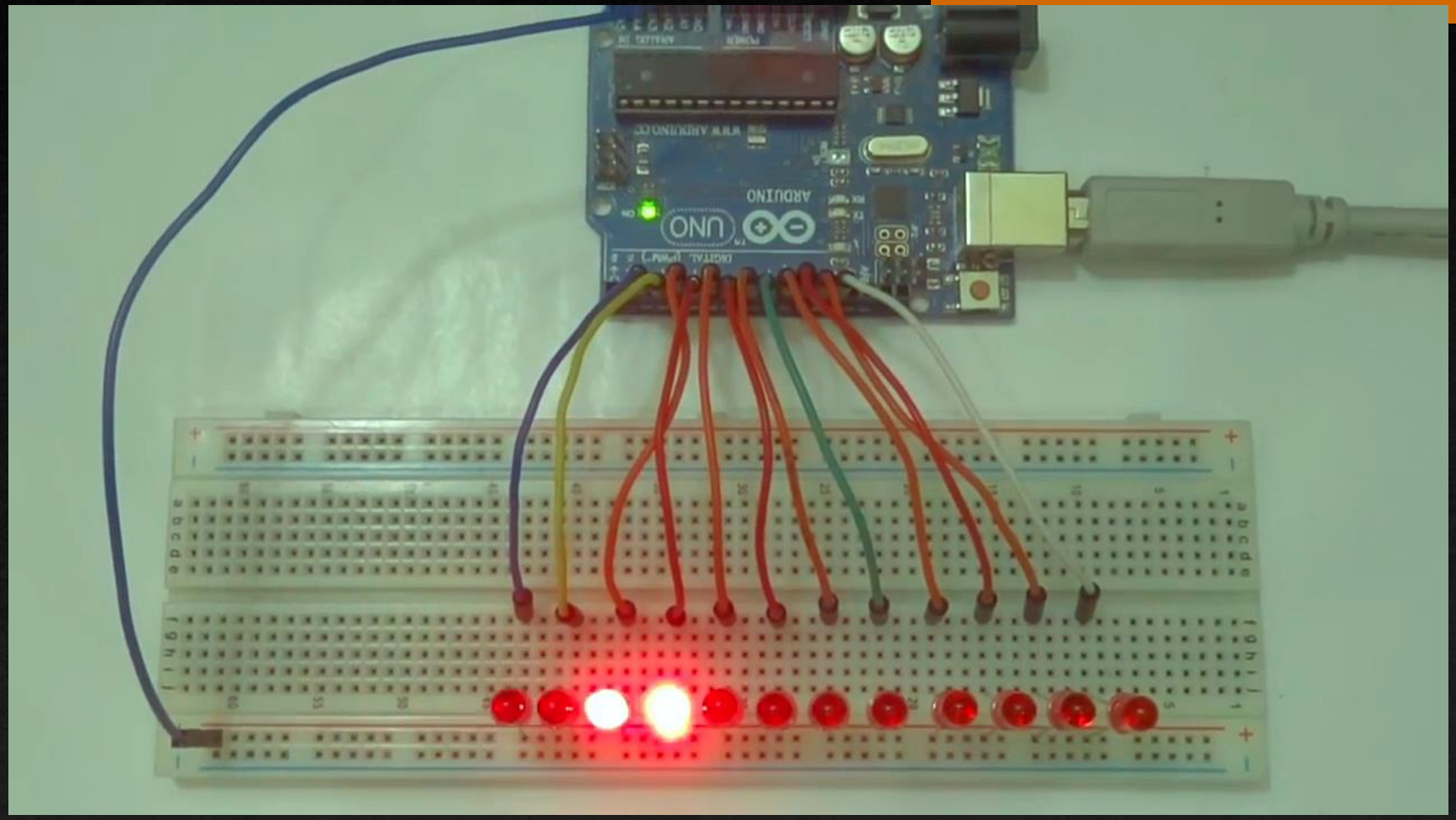
rigt-left

```
for (int it = count - 1; it >= 0; it--)
    process(A[it]);
```

unde count este numărul de elemente ale matricei și este un index folosit pentru analizarea matricei

Project No. 2019-1-RO02-KA202-063965
This project has been funded with support from the European Commission. The content reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.
Co-funded by the Erasmus+ Programme of the European Union

Înțelegerea conceptelor de programare prin utilizarea aplicațiilor Microcontrolere



Learning FOR loops and ARRAYs with Arduino device

Project No. 2019-1-RO01-KA202-063965
This project has been funded with support from the European Commission. The content reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Co-funded by the Erasmus+ Programme of the European Union

Domenii științifice acoperite

Arhitectură hardware pentru calculatoare și sisteme încorporate

Programare structurată – tipuri de date, statemets (IF, WHILE, FOR), funcții definite de utilizator

Programare dispozitive bazate pe microcontroler (ex. Arduino)

Project No. 2019-1-RO01-KA202-063965
This project has been funded with support from the European Commission. The content reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Co-funded by the Erasmus+ Programme of the European Union

Evaluare

- Test cu variante multiple
- mini-proiect într-o echipă de 2-3 studenți - programarea dispozitivelor Arduino care:
- descrieți funcționarea altor instrucțiuni specifice în programarea structurată
- de aplicat in situatii reale – de exemplu, functionarea unui semafor RGV

Project No. 2019-1-RO01-KA202-063965
This project has been funded with support from the European Commission. The content reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Co-funded by the Erasmus+ Programme of the European Union

Bibliografie

Webografie

- https://creativecommons.org/2008/10/22/wired-on-arduino-and-open-source-computing/
- https://www.arduino.cc/en/Tutorial/BuiltInExamples/WhileStatementConditional
- https://www.arduino.cc/en/Tutorial/BuiltInExamples/ForLoopIteration
- https://commons.wikimedia.org/wiki/Category:Fritzing
- https://www.arduino.cc/en/Tutorial/BuiltInExamples/ForLoopIteration
- https://commons.wikimedia.org/wiki/Category:Arduino_projects

Project No. 2019-1-RO01-KA202-063965
This project has been funded with support from the European Commission. The content reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Co-funded by the Erasmus+ Programme of the European Union