

# Analiza complexității unui algoritm Las Vegas pentru rezolvarea problemei Sudoku

**Autori: Cristian PAMPARĂU** <sup>1</sup>

[pamparaucristian0@gmail.com](mailto:pamparaucristian0@gmail.com)

**Coordonator: as.univ.dr.ing Ovidiu GHERMAN**<sup>2</sup>

<sup>1</sup> Universitatea Ștefan cel Mare Suceava, Facultatea Inginerie Electrică și Știința Calculatoarelor, specializarea: Știința și Ingineria Calculatoarelor, anul I

<sup>2</sup> Universitatea Ștefan cel Mare Suceava, Facultatea Inginerie Electrică și Știința Calculatoarelor, Departamentul: Calculatoare

## Rezumat

Prezentăm în această lucrare o abordare algoritmică aleatoare (*randomized algorithm*) Las Vegas pentru rezolvarea tabelelor  $n \times n$  Sudoku. Raportăm rezultatele empirice privind timpul de execuție al algoritmului nostru pentru mărimile de tabelă  $n = 4, 6, 8$  și, respectiv,  $9$ , corespunzând nivelelor ușoare și medii de dificultate. Algoritmul nostru utilizează diferite tehnici de randomizare pentru a calcula valori potrivite pentru celulele goale ale tabelului Sudoku pentru un număr maxim de 2000 iterații, care a fost calculat empiric pentru o performanță eficientă. Pe baza rezultatelor au fost emise concluzii cu privire la complexitatea teoretică a abordării algoritmice testate.

## Cuvinte cheie

*Las Vegas, Sudoku, complexitate, algoritm aleator, empiric*

### 1. Introducere

Sudoku este un joc de tip puzzle logic, combinatorial, ce folosește o tabelă de dimensiune  $n \times n$ , a cărei scop este umplerea grilei cu numere de la 1 la  $n$  respectându-se anumite condiții de plasare a acestor. O tabelă Sudoku (Subramania și Ponnuswamy, 2009) este de fapt o matrice pătratică de dimensiune  $n$ , alcătuită din  $n$  sub-matrici, cu următoarele proprietăți: fiecare linie și respectiv coloană din matrice conține toate numerele de la 1 până la  $n$ , iar fiecare sub-matrice conține, de asemenea, o singură dată, toate numerele de la 1 până la  $n$ . Inițial, tabela este completată parțial cu valori de la 1 la  $n$ , valoarea 0 însemnând poziție liberă. O asemenea tabelă se consideră rezolvată atunci când pe fiecare linie, fiecare coloană și în fiecare sub-matrice sunt numerele de la 1 la  $n$  o singură dată.

Cercetările și publicațiile actuale prezintă rezolvarea acestei probleme prin mai multe metode, precum metoda algoritmilor genetici (Mantere și Koljonen, 2006), metoda polenizării florilor sau *flower pollination* (Abdel-Raouf et al., 2014), metoda *backtracking* (Dhanya și Varghese, 2016), metoda coloniilor de albine (Pacurib et al., 2009) sau printr-un algoritm determinist-probabilistic (Parag și Himanshu, 2007), luând-se în considerare, în toate cazurile, table de dimensiunea  $9 \times 9$ . Suportul pentru lucrarea de față îl constituie un algoritm aleator Las Vegas (Vatavu, 2019), implementat în C# care permite rezolvarea facilă a tabelelor Sudoku de dimensiuni 4,6,8 și respectiv 9 nivelele. Pornind de la acest program, ne-am propus să calculăm empiric complexitatea algoritmului, prin rularea de 100 de ori a programului pentru pereche dimensiune-nivel de dificultate ( $4 \times 4$  dificultate ușoară de 100 ori,  $4 \times 4$  dificultate medie de 100 ori etc).

Prin nivel de dificultate ușor înțelegem o tabelă care are libere un număr de ~43% poziții din totalul de  $n \times n$ , adică pentru o tabelă  $9 \times 9$  vor fi libere 35 de poziții. Nivelul de dificultate mediu se exprimă printr-un număr de poziții libere aproximativ egal cu ~53% poziții din totalul de  $n \times n$ .

### 2. Materiale și metode de cercetare

Prima versiune a algoritmului presupunea inițial citirea tablei Sudoku dintr-un fișier de intrare. Acesta realizează o copie a matricii inițiale, asupra căreia face modificările dat fiind faptul că problema are soluție unică, iar o valoare inserată pe o poziție eronată, va duce la ciclarea tuturor operațiilor până va fi nevoie reluarea încercărilor pe matricea inițială. Astfel, într-o buclă infinită se vor genera aleator 3 valori, după cum urmează: un număr de la 0 la  $n$  reprezentând linia, un număr de la 0 la  $n$  reprezentând coloana (poziție aleatoare, care probabil deja este ocupată) și respectiv un număr de la 1 la  $n$  care se dorește a fi inserat în tabelă. Dacă valoarea generată aleator îndeplinește condițiile Sudoku, atunci poziția generată aleator se va popula cu această valoare, numărul de poziții libere fiind decrementat. Altfel, se va continua cu generarea altor 3 valori aleatorii. Dacă în 2000 de iterații (număr determinat pe baza testelor, valorile fiind alese din intervalul 400-100000) de generare a acestui set de 3 valori, tabela Sudoku nu este rezolvată, se va relua algoritmul de la copierea matricii inițiale. Așa după cum se poate observa și după cum s-a menționat, era de așteptat o complexitate mare a algoritmului, dat fiind faptul că exista probabilitatea ca poziția generată să fi fost deja ocupată, ca fiind dată de intrare pentru algoritm.

În vederea optimizării algoritmului, cu condiția ca acesta să fie în continuare un algoritm aleator, s-a introdus o etapă de preprocesare a datelor, adică programul a fost „ajutat” cu o serie de informații suplimentare despre datele problemei. Mai exact, s-a definit o structura care conține trei date: linia  $i$ , coloana  $j$  și o listă de întregi semnificând lista de valori posibile ce pot fi introduse în poziția determinată de această linie și respectiv coloană. Valorile posibile reprezintă acele valori care lipsesc simultan de pe linia  $i$ , coloana  $j$  și sub-matricea care conține celula determinată de

valorile  $i$  și  $j$ .

Cu valorile rezultate în urma preprocesării datelor, algoritmul va genera aleator un număr reprezentând o poziție disponibilă din tabelă, precum și o valoare aleatoare din lista de valori posibile asociate poziției alese. Din nou, dacă poziția aleasă îndeplinește condițiile populării ei cu respectiva valoare, algoritmul continuă să caute o altă valoare corectă; altfel, se va relua procesul de generare a datelor, contorizând numărul de seturi de generări.

Algoritmul implementat este descris mai jos în limbaj pseudocod. Funcția **determinePossibleValues()** este folosită pentru optimizarea algoritmului în etapa de preprocesare a datelor, scopul ei fiind să extragă informațiile suplimentare din datele de start ale problemei.

Implementarea de test a fost realizată în limbajul C#.

*\*) DIM reprezintă dimensiunea tablei Sudoku (n)*

*funcția main()*

*dim = sqrt(DIM)*

*\*) citește datele din fișierul de intrare*

*\*) pornește cronometrul*

*determinePossibleValues()*

*cât timp !LasVegas()*

*end\_cât\_timp*

*\*) oprește cronometrul*

*\*) afișează timpul*

*end\_main*

*funcția determinePossibleValues()*

*pentru x<-1, n execută*

*\*) n este dimensiunea vectorului de poziții libere; la citirea datelor, dacă valoarea din matrice este 0, se introduce în vector poziția (i,j)*

*\*) generează lista elementelor nule de pe rândul i, row*

*\*) generează lista elementelor nule de pe coloana j, column*

*\*) generează lista elementelor nule din subMatricea determinată de linia i și coloana j, subMatrix*

*\*) realizează intersecția celor 3 liste:*

*pentru y<-1, n execută*

*rar[y] = 0*

*end\_pentru*

*pentru y<-1, m execută*

*\*) m = dimensiunea listei row*

*rar[row[y]]++*

*end\_pentru*

*pentru y<-1, m execută*

*\*) m = dimensiunea listei column*

*rar[column[y]]++*

*end\_pentru*

*pentru y<-1, m execută*

*\*) m = dimensiunea listei subMatrix*

*rar[submatrix[y]]++*

*end\_pentru*

*pentru y<-1, DIM execută*

*daca rar[y] == 3 atunci*

*\*) s-a găsit o valoare posibilă pentru poziția (i,j)*

*\*) inserează y în lista de valori posibile asociate poziției (i,j)*

*end\_dacă*

*end\_pentru*

*end\_pentru*

*end\_determinePossibleValues*

*funcția LasVegas()*

*\*) copiază matricea de intrare în variabila matrix*

*\*) copiază vectorul de poziții libere în variabila current vector*

*\*) copiază numărul de poziții libere în variabila pos*

*cât timp adevărat*

*nr\_iterații++*

*\*) generează poziția aleatoare (i,j)*

*\*) generează valoarea aleatoare newValue*

```

    dacă !estePeRand(newValue,i,matrix) && !estePeColoana(newValue, j, matrix) &&
        !esteInSubMatrice(newValue, i, j, matrix) atunci
        matrix[i,j]=newValue
        *) elimina poziția generata din current_vector
        pos--
    end_dacă
    dacă pos==0 atunci
        întoarce adevărat
    end_dacă
    dacă nr_iteratii>2000
        întoarce fals
        *) rezultatul fals întors duce la reapelarea funcției LasVegas()
    end_dacă
end_cât_timp
end_LasVegas

```

**funcția estePeRand(newValue, row, matrix)**

```

    pentru i<-1,DIM execută
        dacă matrix[row,i] == newValue atunci
            întoarce adevărat
        end_dacă
    end_pentru
    întoarce fals
end_estePeRand

```

**funcția estePeColoana(newValue, coloana, matrix)**

```

    pentru i<-1,DIM execută
        dacă matrix[i,column] == newValue atunci
            întoarce adevărat
        end_dacă
    end_pentru
    întoarce fals
end_estePeColoana

```

**funcția esteInSubMatrice(newValue, i, j, matrix)**

```

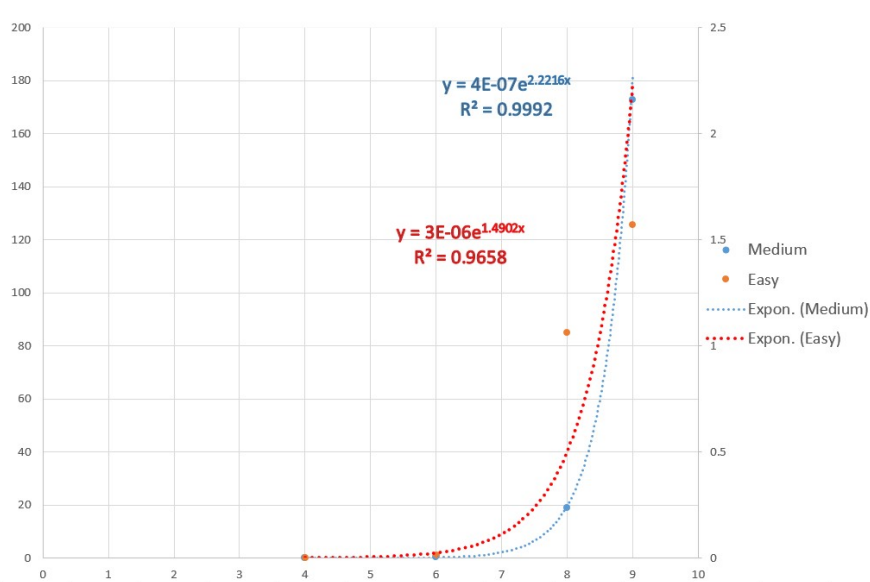
    minI = (i/dim)*dim
    *) dim=sqrt(DIM) - calculat în metoda main() anterior
    maxI = minI + dim + (DIM==6?1:0)
    minJ = (j / (DIM==6? dim+1 : (DIM==8? dim*2 : (DIM == 10 ? 5 : dim)))) *
        (DIM==6?dim+1:(DIM==8?dim*2:(DIM==10?5:dim)))
    maxJ = minJ dim
    pentru ii<-minI, maxI execută
        pentru jj<-minJ, maxJ execută
            dacă matrix[ii,jj] == newValue atunci
                întoarce adevărat
            end_dacă
        end_pentru
    end_pentru
    întoarce fals
end_esteInSubMatrice

```

### 3. Evaluare și rezultate

Experimentul a fost efectuat pe o mașină cu o cantitate de memorie RAM de 16 GB și procesor Intel(R) Core(TM) i7-7700HQ CPU cu frecvența de 2,8 GHz.

În Figura 1 se prezintă graficul complexității obținute pentru prima versiune a algoritmului. Complexitatea este calculată ca și raport al dimensiunii tablei de joc (bazată pe  $n$ ) în raport cu timpul de execuție al algoritmului pe tabla respectivă. Pentru prima versiune a algoritmului complexitatea este exponențială, dovedind că algoritmul este ineficient (nefavorabil) pentru rezolvarea tablei Sudoku.



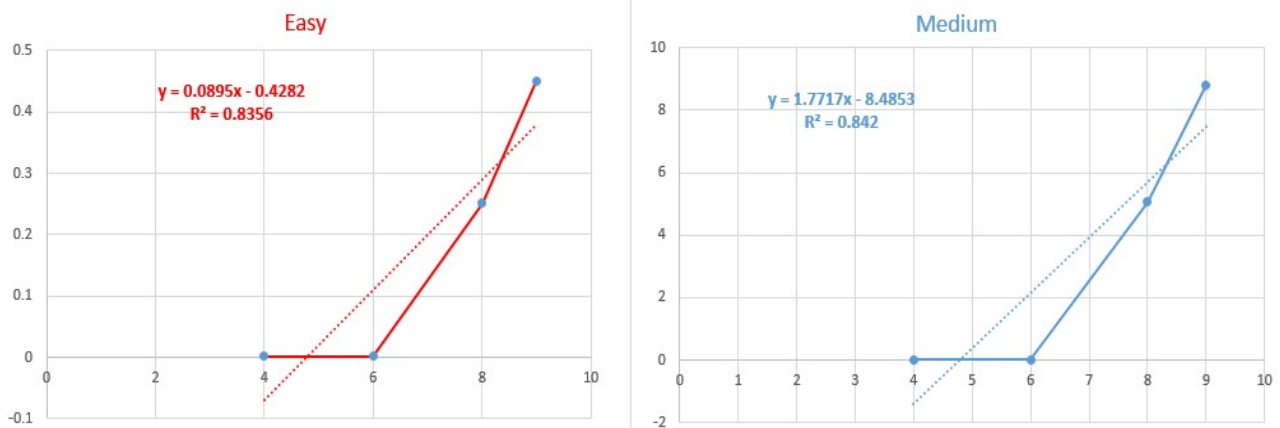
**Fig. 1.** Complexitate exponențială pentru prima versiune a algoritmului (dificultate ușoară și medie).

Figura 1 a fost realizată pe baza datelor experimentale prezentate în Tabelul 1.

**Tabelul 1.** Media timpilor de execuție pentru prima versiune a algoritmului

Dimensiune tabelă (n)	Nivel de dificultate	
	Mediu (secunde)	Ușor (secunde)
4	0.002429	0.001635
6	0.282467	0.013319
8	18.87608	1.064026
9	172.7323	1.573507

În urma optimizării algoritmului prin adăugarea etapei de preprocesare, complexitatea acestuia s-a modificat substanțial, devenind liniară, așa după cum se observă în Figura 2.



**Fig. 2.** Complexitate liniară pentru versiunea finală a algoritmului (dificultate ușoară și medie).

Analog, în Tabelul 2 se va prezenta media timpilor de execuție pe baza cărora a fost determinată complexitatea algoritmului de rezolvare.

**Tabelul 2.** Media timpilor de execuție pentru versiunea finală a algoritmului

Dimensiune tabelă (n)	Nivel de dificultate	
	Mediu (secunde)	Ușor (secunde)
4	0.002015	0.001547
6	0.017601	0.001637
8	5.068611	0.250977
9	8.807018	0.449795

#### 4. Concluzii

Un algoritm aleator se constituie dintr-o finitudine de pași care presupun o anumită probabilitate în logica sa de logică sa de funcționare (în engleză, *randomized algorithm*). Acești algoritmi sunt de două tipuri: Las Vegas sau Monte Carlo. Un algoritm Las Vegas va rezolva totdeauna problema, însă timpul de execuție nu poate fi determinat nici aproximativ, spre deosebire de un algoritm Monte Carlo care va rezolva problema într-un timp infinit de execuție. Cum timp infinit de execuție nu există, se va rezolva problema într-o anumită măsură, după un anumit timp de execuție. După cum se observă, cu un algoritm aleator de tip Las Vegas s-a obținut o complexitate liniară, lucru ce nu se poate obține prin metodele de programare clasice, deterministe, precum *backtracking*. Totuși, algoritmi Las Vegas pot rezolva problema doar dacă li se oferă o serie de „informații” despre datele problemei. În acest sens, în exemplul de față, complexitatea algoritmului în situația în care nu exista etapa de preprocesare a datelor era destul de mare (exponențială). Se observă așadar faptul că viteza de execuție a algoritmului s-a îmbunătățit substanțial după momentul în care au fost furnizate mai multe informații despre datele problemei, rezultate în faza inițială de preprocesare sau pregătire a datelor.

#### Bibliografie:

1. Abdel-Raouf O., El-henawy I., Abdel-Baset M., (2014), *A Novel Hybrid Flower Pollination Algorithm with Chaotic Harmony Search for Solving Sudoku Puzzles*, în I.J. Modern Education and Computer Science, nr. 3, pag. 38-44, DOI: 10.5815/ijmecs.2014.03.05.
2. Dhanya J., Varghese P., (2016), *Recursive Backtracking for Solving 9\*9 Sudoku Puzzle*, în Bonfring International Journal of Data Mining, vol. 6, nr. 1, ISSN 2277-5048, DOI: 10.9756/BIJDM.8128.
3. Mantere T., Koljonen J., (2006), *Solving and Rating Sudoku Puzzles with Genetic Algorithms*, în New Developments in Artificial Intelligence and the Semantic Web, Proceedings of the 12th Finnish Artificial Intelligence Conference STeP 2006, Espoo, Finlanda.
4. Pacurib J.A., Seno G.M.M., Yusiong J.P.T., (2009), *Solving Sudoku Puzzles Using Improved Artificial Bee Colony Algorithm*, în Fourth International Conference on Innovative Computing, Information and Control (ICICIC), DOI: 10.1109/ICICIC.2009.334, Taiwan.
5. Parag H.D., Himanshu B.D., (2007), *An Implementation of Sudoku-Solver Using Randomized Algorithm*, disponibil online la [https://www.academia.edu/9969706/An\\_Implementation\\_of\\_Sudoku-Solver\\_using\\_Randomized\\_Algorithm](https://www.academia.edu/9969706/An_Implementation_of_Sudoku-Solver_using_Randomized_Algorithm), mai 2019.
6. Subramani J., Ponnuswamy K.N., (2009), *Construction and Analysis of Sudoku Designs*, în Model Assisted Statistics and Applications, nr. 4, DOI: 10.3233/MAS-2009-0139, IOS Press.
7. Vatavu R., (2019), *Algoritmi Avansați, Probabilistici și Tehnici Metaeuristice* (curs), disponibil online la <http://www.eed.usv.ro/~vatavu/teaching/aaptm/Curs7.pdf>, mai 2019.

# CONSIDERAȚII TEORETICE CU PRIVIRE LA UTILIZAREA ȘI SECURITATEA DISPOZITIVELOR SMART ÎN CONTEXTUL INTERNET OF THINGS (IoT)

**Autori:** Alina ȘOLTOIANU <sup>1</sup>, Ioana-Dariana TĂMĂIOGĂ <sup>1</sup>  
cosminrus@upet.ro

**Coordonatori:** Conf.univ.dr.ing. Nicolae Pătrășcoiu <sup>2</sup>, Asist.cercet.drd.ing. Cosmin Rus <sup>2</sup>

<sup>1</sup> Universitatea din Petroșani, Facultatea I.M.E., specializarea: Calculatoare , anul II

<sup>2</sup> Universitatea din Petroșani, Facultatea I.M.E., Departamentul: A.C.I.E.E.

## Rezumat

În această lucrare se prezintă mai multe considerente teoretice cu privire la Internetul Lucrurilor (IoT) și dispozitivele interconectate de tip smart. Se fac mai multe aprecieri critice asupra funcționalității unor echipamente de tip smart și se prezintă în sinteză o evaluare a riscurilor de securitate la care aceste dispozitive pot fi expuse.

**Cuvinte cheie:** dispozitive inteligente, smart things, IoT

## 1. Generalități despre Internetul Lucrurilor

Cu toții cunoaștem Internetul așa cum este la momentul actual, un Internet al oamenilor – Internet of People (IoP), cu toate datele, imaginile, înregistrările video, jocurile, cărțile și produsele on-line, pe care le putem accesa oricând și de aproape oriunde. Internetul, una din cele mai importante tehnologii ce a reușit să ne influențeze stilul de viață, a fost creat de oameni pentru oameni. Dinamica schimbărilor duce însă Internetul la un nou nivel, noul Internet nerezumându-se doar în a conecta oamenii, ci și lucrurile. Acest Internet al lucrurilor – Internet of Things (IoT) este capabil să conecteze toate dispozitivele inteligente cu scopul de a fi monitorizate și controlate de la distanță. Vom ajunge, nu peste mult timp, să avem un Internet al tuturor lucrurilor – Internet of Everything (IoE) – care să conecteze oameni, dispozitive și locuri într-un tot unitar, pentru a ne asista viața de zi cu zi. Acest lucru este posibil prin implementarea de senzori și abilități de comunicare tuturor dispozitivelor ce ne înconjoară [1, 2].

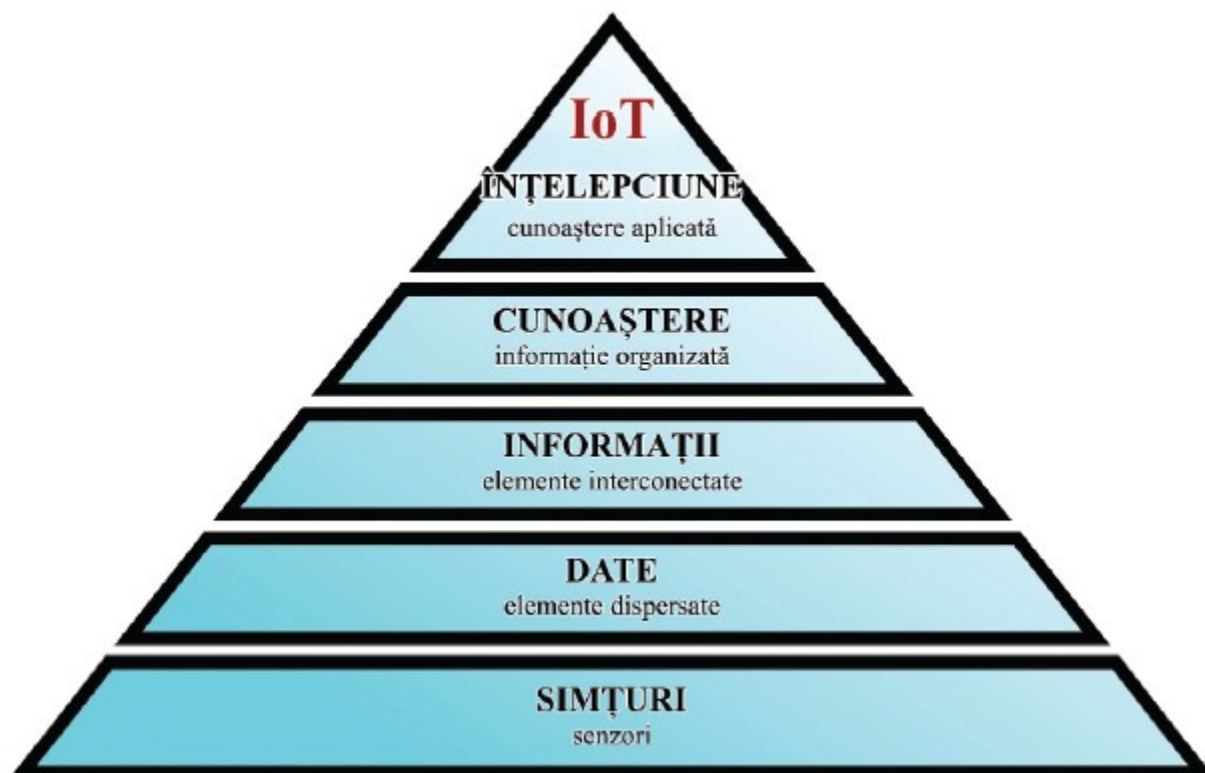


Fig. 1 Ierarhia informației în Internetul lucrurilor (IoT)

Prin senzorii implementați, dispozitivele vor culege date din mediul înconjurător, se vor conecta între ele, vor schimba datele obținute, iar informațiile astfel coroborate vor ajunge la utilizator, acesta putând lua decizii și controla activitatea dispozitivelor.

**Din punct de vedere al utilizatorului**, gândiți-vă cum ar fi ca dimineața, în momentul în care vă sună alarma, lumina să se aprindă automat treptat, televizorul să pornească pe canalul preferat, espressoorul să vă pregătească cafeaua, toasterul să vă prăjească pâinea, mașina să fie pregătită în momentul în care doriți să plecați și pe drum să vă indice calea optimă pentru a ajunge la serviciu cât mai repede și să informeze în legătură cu principalele știri din domeniul

preferat. Odată ajuns în parcare, aerul condiționat și calculatorul din birou să pornească automat. Toate aceste lucruri sunt aproape posibile, făcându-ne viața mai ușoară și mai interactivă.

**Locuințele inteligente**, dotate cu astfel de dispozitive, pot ajuta foarte mult la economisirea energiei prin reducerea consumurilor inutile, asigurând încălzirea și iluminarea optimă în funcție de activitățile locatarilor și de condițiile meteo-climatice. Sensorii amplasați aceste locuințe pot avertiza în timp real despre diverse avarii sau incidente precum inundație, incendiu sau intrarea unui străin prin efracție.

**Mașinile inteligente** vor fi conectate la sistemele de monitorizare al traficului și vor ști să evite ambuteiajele și să găsească cele mai apropiate locuri de parcare disponibile. În cazul apariției unor defecțiuni tehnice, acestea vor fi detectate de senzori și vor fi raportate în timp real proprietarului.

**Orașele inteligente** vor pune la dispoziția cetățenilor servicii ce pot contribui major la protejarea mediului înconjurător. Resursele importante (apa, gazele, energia electrică sau termică) vor putea fi monitorizate și controlate prin automatizare și vor fi distribuite mai eficient, în conformitate cu necesitățile reale, iar companiile de distribuție pot fi alertate imediat dacă apar probleme de infrastructură.



Fig. 2 Multitudinea de senzori dintr-un oraș inteligent

**Sistemul medical** va putea beneficia de asemenea de avantajele tehnologiei IoT. Monitorizarea continuă a semnelor vitale ale organismului pacienților poate contribui la siguranța acestora și la informarea la timp a doctorilor atunci când apar probleme de sănătate. Sensorii pot ajuta la administrarea medicamentelor prescrise, aducând un aport enorm în spitale, cămine sau aziluri de bătrâni, acolo unde monitorizarea atentă a persoanelor îngrijite este cel puțin la fel de importantă ca și intervențiile efectuate la timp.

**Probleme de conectivitate.** Tehnologia IoT presupune dispozitive interconectate și dependente unele de altele. În momentul în care un dispozitiv este compromis, cedează sau livrează date eronate, efectul se va propaga tuturor sistemelor care depind de el direct sau indirect. Efectul poate fi relativ minor (atunci când aplicația de planificare nu mai transmite ceasului o alarmă privind o întâlnire) sau major (când un senzor ce monitorizează funcțiile vitale ale unui pacient nu informează doctorul privind un incident medical).

**Probleme de securitate.** Dornici de a lansa cât mai repede un produs nou pe piață, proiectanții echipamentelor IoT neglijează de cele mai multe ori aspectele de securitate. Din acest motiv tot mai multe dispozitive inteligente (telefoane, televizoare, camere de supraveghere sau frigider) sunt implicate în atacuri cibernetice de amploare.



Fig. 3 Securitatea dispozitivelor inteligente

Multe companii au analizat sistemele inteligente disponibile în acest moment pe piață și au ajuns la concluzia îngrijorătoare că securitatea acestora este complet nesatisfăcătoare. Mai mult, câteva dintre aceste sisteme sunt promovate drept soluții inteligente de securitate, ceea ce face ca situația să fie cu atât mai ironică și mai gravă. Pe lângă vulnerabilitățile detectate, aceste dispozitive nu dețin rutine software pentru actualizare automată și folosesc canale de comunicație necriptate sau prost criptate.

**Probleme de confidențialitate a datelor stocate.** Cele mai multe dintre aceste sisteme sunt vulnerabile la atacurile informatice, neavând aplicații de protecție necesare sau politici de securitate care să impună parole cu o complexitate satisfăcătoare și nu protejează cum trebuie datele stocate. Concepute prost, aceste sisteme pot permite unor eventuali intruși să arunce o privire la datele înregistrate fără ca aceștia să poată fi depistați.

**Risipă de energie.** Într-un studiu publicat de Agenția Internațională a Energiei se arată că cele aproximativ 14 miliarde de dispozitive conectate la Internet, existente în prezent la nivel mondial, risipesc o cantitate enormă de energie electrică din cauza unor tehnologii ineficiente, iar această problemă se va agrava până în 2020, când electricitatea risipită de aceste dispozitive va crește cu 50%. Risipa de energie se datorează faptului că dispozitivele utilizează mai multe electricitate decât ar trebui pentru a menține conexiunea și a comunica cu rețeaua.

## 2. Riscuri de securitate

Atunci când ne gândim la soluții de securitate cibernetică, nu ne vine în minte că ar trebui să avem un software care să ne protejeze peria de păr (dacă aceasta este un device smart), iar specialiștii atrag atenția că, deși inteligente, device-urile IoT nu ar trebui considerate neapărat sigure, iar vulnerabilitatea unui device conectat prin rețea la restul device-urilor din casă devine o vulnerabilitate pentru întreaga casă [3-7].

Internet of Things (IoT) a devenit un termen recunoscut la nivel global, iar într-un sens foarte larg ar putea fi utilizat pentru a descrie orice este conectat la internet. Însă ce device-uri se încadrează de fapt în IoT? În această categorie, oamenii tind să încadreze în sens general aproape orice, de la telefoane mobile, becuri inteligente, brățări de fitness, boxe smart și mașini de spălat vase inteligente până la senzori care testează calitatea apei înainte de distribuție, potrivit unui studiu publicat de compania slovacă de securitate cibernetică ESET asupra tehnologiilor IoT și a conceptului de smart home.

Când predicțiile cu privire la răspândirea IoT au început să apară, analiștii au supraestimat capacitatea de absorbție a pieței și anunțau cifre imense: 50 de miliarde de device-uri IoT în lume până în 2020 a fost numărul citat într-o prezentare oficială în 2010. Opt ani mai târziu, entuziasmul s-a mai temperat în jurul sectorului, iar estimările jucătorilor din industrie sunt mult mai conservatoare. Astăzi, Ericsson oferă o perspectivă mult mai nuanțată, estimând că aproximativ 29 de miliarde de device-uri conectate vor exista în lume până în 2022, dintre care circa 18 miliarde vor avea o conectivitate specifică IoT.

În timp ce numărul device-urilor la nivel global este, cu siguranță, în creștere, iar acestea aduc beneficii atât caselor oamenilor cât și vieților acestora în general, majoritatea consumatorilor au tendința de a ignora amenințările care vin odată cu tehnologia sau de a nu se proteja corespunzător de potențialele atacuri cibernetice.

Toți senzorii din componența unui produs specific smart home — cu microfoane, camere, interfață cu GPS și interoperabilitate la nivelul sistemului - sunt ținte atractive pentru atacuri de tip malware (tip de software utilizat pentru deteriorarea datelor dintr-un device) sau chiar ransomware (tip de atac cibernetic ce blochează accesul utilizatorului la sistem dacă nu plătește o recompensă). Dacă obțin controlul asupra acestor device-uri, criminalii ciberneticici pot să atace alte device-uri conectate la rețea cu device-ul utilizatorului. Mai mult, aceștia pot spiona și pot obține date personale sau date confidențiale.

Exemplul oferit în studiul ESET de la începutul anului care demonstrează că un singur device personal poate prezenta un risc enorm vine de la un student australian, Nathan Ruser, de 20 de ani, care studiază securitate internațională în cadrul Universității Naționale din Australia. În cadrul unei postări pe rețeaua socială Twitter în 27 ianuarie 2018, Ruser a evidențiat o problemă de securitate operațională generată de personalul militar care folosește aplicația de fitness Strava. Aplicația utilizează locația GPS din telefon pentru a urmări și înregistra rutele de jogging ale utilizatorului. Astfel, Strava înregistrează trasee. Exemplul evidențial de Ruser a fost baza aeriană Bagram din Afganistan, în dreptul căreia orice utilizator al aplicației putea vedea pe hartă traseul de antrenament utilizat de militarii staționați în bază. Acest incident nu a avut urmări, însă poate fi un exemplu al riscurilor cu care vin chiar și device-urile personale [8].

Primul telefon smart a fost iPhone-ul lansat în SUA pe 29 iunie 2007. În perioada octombrie- noiembrie 2009, virusul Dutch Hack deja făcea victime printre utilizatori. Acest virus bloca accesul utilizatorilor la dispozitiv dacă aceștia nu plăteau o recompensă de 5 euro pentru deblocare.

La 22 octombrie 2008, a fost lansat în SUA primul telefon cu sistem de operare tip Android, iar în august 2010, virusul FakePlayer.A, care se infiltra în sistem prin accesarea unor aplicații de redare media false și trimitea mesaje text cu plată de pe dispozitivul afectat.

Se susține, în baza unui articol publicat în platforma BleepingComputer, că aproape 90% dintre televizoarele smart sunt vulnerabile la a fi hackuite de la distanță [9]. Atacul cibernetic prin care s-a ajuns la această statistică a fost dezvoltat de Rafael Scheel, un cercetător pe segmentul de securitate în cadrul companiei elvețiene de securitate cibernetică Oneconsult, și a fost utilizat doar pentru a testa gradul de risc la care sunt expuse datele colectate de smart TV-uri. Mai mult, se vorbește despre protecția cibernetică a saltețelor smart, un subiect ce poate părea discutat pentru un scenariu ireal. Cu toate acestea, saltețele smart există, iar tehnologia lor le permite să memoreze tipare de somn, forma corpului, comportamentul utilizatorului în timpul somnului, iar acestea se pot adapta. Pentru a face însă acest lucru, saltețele trebuie să stocheze date, iar aceste date pot fi atacate. „Să spunem că cineva îți atacă cibernetic saltețeaua și să spunem că nevasta te înșală, sau tu pe ea, iar cineva poate folosi datele extrase din saltea doar pentru a te șantaja. Însă există și produse complete de protecție pentru smart home care pot preveni atacurile cibernetice”, afirmă un specialist. Chiar și jucăriile pentru copii și stadiul tehnologic avansat în care au ajuns unele dintre acestea pot prezenta un risc la adresa intimității și a securității datelor. Potrivit CNN, în 2017, în Germania, organele de reglementare s-au autosesizat și au interzis vânzarea unei păpuși numite My Friend Cayla. Această jucărie se conecta la internet prin intermediul Bluetooth-ului și putea răspunde copiilor la întrebări simple precum „Care este cel mai înalt munte din lume?”. Instituțiile germane de reglementare au oprit vânzările după câteva luni și au transmis tuturor părinților care au achiziționat aceste păpuși să le distrugă deoarece au microfoane neprotejate cibernetic în componența lor.

Însă vulnerabilitatea dispozitivelor inteligente conectate la internet poate avea și implicații foarte serioase și



poate afecta chiar și dispozitivele medicale foarte avansate din punct de vedere tehnologic. În vara anului 2017, Administrația Medicamentelor și a Alimentației din SUA (FDA) a anunțat că 465.000 de pacienți cu stimulator cardiac (denumit pacemaker, este un dispozitiv medical care emite impulsuri electrice, transmise prin electrozi ce sunt în contact cu mușchii inimii, cu scopul de a regulariza bătăile inimii) au nevoie de update-uri la dispozitivele lor pentru că ar putea fi atacați cibernetic, întrucât și stimulatoarele sunt conectate la internet.

Noile dispozitive precum periile de păr inteligente mențin apetitul de consum pentru tehnologie. Periile de tip Hair Coach sunt dispozitive care pot învăța textura părului unei persoane și pot personaliza caracteristicile periei, acesta fiind un dispozitiv conectat, de asemenea, la internet. Chiar și acestea pot fi atacate cibernetic. Care este pericolul dacă cineva îți hackuiește peria de păr?

La prima vedere poate niciunul, deoarece utilizatorul nu consideră textura părului o informație sensibilă. Însă specificul tehnologiilor IoT stă în conectivitate. Atât conectivitatea la internet, cât și interconectivitatea. Accesul la un device neprotejat atât de banal precum o perie de păr pe care o utilizezi acasă, conectată la același router de internet nesecurizat precum celelalte dispozitive din casă, poate asigura unui atacator accesul la întreg sistemul smart home, caz în care cu cât mai multe dispozitive IoT, cu atât mai bine pentru atacator. Iar multitudinea de device-uri poate depăși imaginația consumatorilor care sunt „la început de drum” cu tehnologia IoT. O companie numită HAPI a dezvoltat produsul HAPIfork, o furculiță care promite utilizatorului că ajută la scăderea în greutate, prin caracteristicile smart pe care le are încorporate. Această furculiță învață comportamentul utilizatorilor plecând de la parametri precum durata unei mese sau de câte ori a îndreptat utilizatorul furculița spre gură, informații ce pot fi uploadate ulterior în computer și interpretate prin intermediul platformei HAPI pentru a oferi sfaturi de nutriție.

### 3. Concluzii

În concluzie, putem spune că orice evoluție a tehnologiei este benefică, dar trebuie să conștientizăm și riscurile aferente. Atât timp cât securitatea Internetului lucrurilor (IoT) este prezentă mai mult la nivel declarativ, riscul de compromitere al tuturor sistemelor inteligente interconectate și expunerii datelor înregistrate este destul de mare. Vor fi însă și momente în care viața noastră ar putea depinde de un dispozitiv inteligent ce ne monitorizează funcțiile vitale și poate da alarma într-un moment de criză, făcând posibilă o intervenție de urgență. Astfel de gadgeturi pot simplifica viața unei persoane și pot chiar îmbunătăți activitățile cotidiene, însă specialiștii atrag atenția că fără o protecție adecvată, chiar și cel mai nesemnificativ device, conectat la internet, poate reprezenta o amenințare dacă nu este protejat corespunzător. Cât de smart sunt, de fapt, IoT-urile?

#### Bibliografie:

1. Xu B., Zheng J., Wang Q., (2016) Analysis and Design of Real-Time Micro-Environment Parameter Monitoring System Based on Internet of Things, IEEE, pp. 368-371.
2. Sun Y., Xia Y., Song H., Bie R., (2014) Internet of Things Services for Small Towns, International Conference on Identification, Information and Knowledge in the Internet of Things, Beijing, pp. 92-95.
3. Dongare A., Hesling C., Bhatia K., Balanuta A., Pereira R. L., Iannucci B., Rowe A., (2017), OpenChirp: A Low-Power Wide-Area Networking architecture, 2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops), pg. 569–574.
4. Pătrășcoiu N., Rus C., (2016), Study on the use of Arduino boards to monitor power consumption, Annals of the University of Petrosani, Electrical Engineering, vol.18, ISSN 1454-8518.
5. Pătrășcoiu N., Rus C., Barbu I.-C., (2017), Virtual instrumentation for data acquisition and remote control, 18th International Carpathian Control Conference, Sinaia, INSPEC ACCESSION NUMBER: 17014220 DOI:10.1109/CarpathianCC.2017.7970378 Publisher: IEEE.
6. Rus C., (2016), Arduino webserver used for electrical energy monitoring, International Multidisciplinary Symposium "Universitaria SIMPRO 2016", Petrosani, ISSN 1454-8518.
7. Rus Cosmin, Leba Monica., Ionica Andreea, Road traffic control in the context of a smart city, The national technical-scientific conference (international participation) - the 17th edition, "Modern technologies for the 3rd millennium", <http://www.arhiconoradea.ro/Conferinta/HOME.htm>, Oradea, 2018.
8. <https://www.eset.com/us/>
9. <https://www.bleepingcomputer.com/>

# STUDIU PRIVIND CASA INTELIGENTĂ AUTONOMĂ ENERGETIC

**Autori:** Robert-Cristian PARLAPAN <sup>1</sup>, Izabela-Maria Șurlea <sup>2</sup>  
cosminrus@upet.ro

**Coordonatori:** Conf.univ.dr.ing. Nicolae Pătrășcoiu <sup>3</sup>, Asist.cercet.drd.ing. Cosmin Rus <sup>3</sup>

<sup>1</sup> Universitatea din Petroșani, Facultatea I.M.E., specializarea: Calculatoare, anul IV

<sup>2</sup> Universitatea din Petroșani, Facultatea I.M.E., specializarea: Electromecanică, anul I

<sup>3</sup> Universitatea din Petroșani, Facultatea I.M.E., Departamentul: A.C.I.E.E.

## Rezumat

Această temă a fost dezbătută în mai multe rânduri și odată cu evoluția tehnologiei s-a revenit asupra ei, cum este și firesc, cu îmbunătățiri, dar ideea de bază păstrându-se de-a lungul anilor: confort cât mai sporit, risipă cât mai puțină. De aceea mi-am propus și eu să dezvolt acest subiect dintr-o perspectivă proprie, bazându-mă pe cunoștințele dobândite în facultate dar și din experiența personală. Aportul personal adus acestui proiect este folosirea tehnologiei de ultimă generație în materie de surse de energie, microcontrolere și senzorică, ținând cont de nevoile unei familii moderne.

**Cuvinte cheie:** energie regenerabilă, domotică, acumulatori, gestionare energetică

## 1. Introducere

Se propune să se dimensioneze o casă tipică a unei familii moderne compusă din 4 membri, să se calculeze un consum mediu lunar și să se dimensioneze diversele surse de energie regenerabilă astfel încât să acopere necesarul creat. Cât pentru partea de domotică, se vor implementa diverse funcții și servicii, folosind microprocesoare de tip Arduino UNO și Arduino MEGA alături de numeroși senzori pentru a ușura activitățile întreprinse atât în casă cât și în împrejurul acesteia.

Din cele mai vechi timpuri și până în zilele noastre, atât oamenii cât și toate animalele și plantele s-au adaptat și au fost nevoiți să ia diferite măsuri pentru a-și asigura existența și pentru a o îmbunătăți. Oamenii, în special, au fost cei mai ingenioși și am ajuns în acest prag al evoluției când supraviețuirea nu mai este o problemă atât de mare, ci doar îmbunătățirea calității vieții pe cât mai mult posibil, dar protejând și mediul care ne înconjoară; această ultimă idee fiind și scopul principal al lucrării.

## 2. Descriere generală

În continuare se va prezenta calculul estimativ al casei ținând cont de un număr de consumatori, consumul specific al acestora și un interval de funcționare exprimat în ore/zi.

Tot aici va fi prezentată o dimensionare a surselor regenerabile de energie ce vor fi implementate, modul lor de funcționare și punctual fiecare etapă în ceea ce privește distribuția energiei către consumatori luând în calcul și alte aparate ce fac acest lucru posibil.

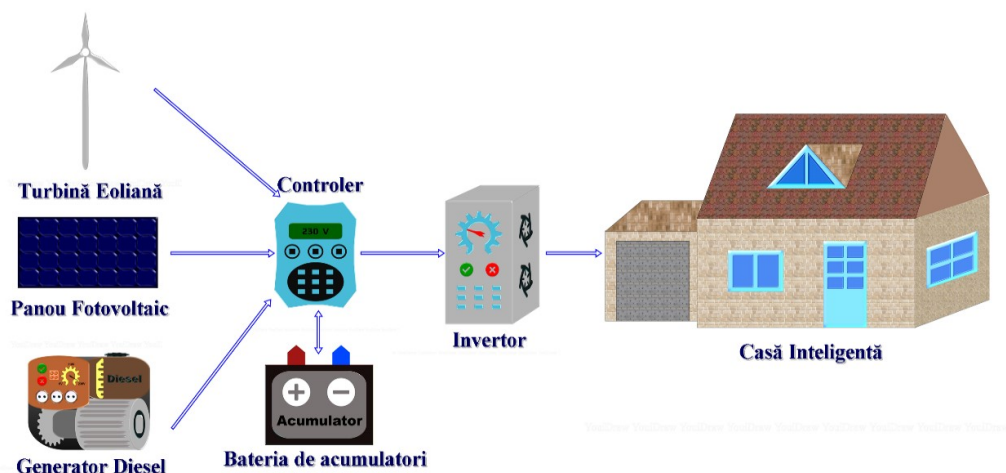


Fig.1. Schema ansamblului de elemente constructive

### Legendă:

*Turbina eoliană:* transformă energia mecanică a vântului în energie electrică

*Panoul Fotovoltaic:* transformă radiația solară în energie electrică

*Genertor Diesel:* transformă energia produsă prin arderea combustibilului prin explozii controlate în energie electrică-rol de a suplini sau în caz de urgență

*Controler hibrid:* comandă acumulatorii, îi încarcă și descarcă în siguranță

*Bateria de acumulatori:* înmagazinează energia produsă pentru a fi consumată ulterior

*Invertor:* transformă curentul continuu de la surse în curent alternativ

*Consumatori(casa inteligentă):* aparatele electrice din clădire alimentate la invertor (curent alternativ) sau la acumulatori (curent continuu).

După cum se poate observa și în *Tabelul 1* am împărțit consumatorii pe categorii pentru a putea studia care are un necesar mai mare de energie, acest lucru fiind folositor și în proiectarea cablurilor din rețeaua electrică a casei, dar și un mijloc de prevenție și de economisire prin reducerea timpului efectiv de funcționare sau a numărului de dispozitive și totodată prin achiziționarea de noi aparaturi care au aceeași funcționalitate dar un consum mai scăzut.

*Tabelul 1.*Consum lunar total pe categorii de consumatori

Iluminat	=	36,45	KWh/Luna	Procent:	8%
Electrocasnice:	=	237,9643	KWh/Luna	Procent:	49%
Climatizare:	=	60	KWh/Luna	Procent:	12%
Comunicatii:	=	132,1393	KWh/Luna	Procent:	27%
Scule:	=	18,32143	KWh/Luna	Procent:	4%
Altele:	=		KWh/Luna	Procent:	
	=				
<b>Total:</b>	=	<b>484,875</b>	<b>KWh/Luna</b>	Procent:	<b>100%</b>

În *Tabelul 2.* se poate observa mai detaliat prin exemplificarea categoriilor consumatorilor folosiți pentru iluminat și consumatorilor electrocasnici, cum s-a ajuns la rezultatele din *Tabelul 1.* și anume prin enumerarea anumitor tipuri de consumatori, numărul lor și orele acestora de funcționare pe zi, dar și frecvența săptămânală. Toti coeficienții introduși au un rol estimativ și nu reflectă cu exactitate programul de funcționare propriu zis.

*Tabelul 2.*Consum lunar estimativ a categoriilor de consumatori pentru Iluminat și Electrocasnici

Iluminat	Putere	Ore/zi	Cantitate	Zile/ Sapt.	KWH/zi	KWh/Luna
Economic	11 W	x 5,0 h	x 15	x 7	= 0,8 KWh	24,8 KWh
Flourescent	13 W	x 3,0 h	x 10	x 7	= 0,4 KWh	11,7 KWh
Electrocasnice	Putere	Ore/zi	Cantitate	Zile/ Sapt.	KWH/zi	KWh/Luna
Mixer	300 W	x 1,0 h	x 1	x 2	= 0,1 KWh	2,6 KWh
Uscator	1000 W	x 0,5 h	x 1	x 7	= 0,5 KWh	15,0 KWh
Ventilator	50 W	x 0,0 h	x 0	x 0	= 0,0 KWh	0,0 KWh
Uscator haine (electric)	4000 W	x 0,0 h	x 0	x 0	= 0,0 KWh	0,0 KWh
Uscator Haine (incalzire gas)	400 W	x 0,0 h	x 0	x 0	= 0,0 KWh	0,0 KWh
Cafetiera	1000 W	x 1,0 h	x 1	x 7	= 1,0 KWh	30,0 KWh
Masina spalat vase (uscare)	700 W	x 0,0 h	x 0	x 0	= 0,0 KWh	0,0 KWh
Masina spalat vase (turbo)	1450 W	x 0,5 h	x 1	x 7	= 0,7 KWh	21,8 KWh
Dehidrator	600 W	x 0,0 h	x 0	x 0	= 0,0 KWh	0,0 KWh
Mixer solide	400 W	x 0,0 h	x 0	x 0	= 0,0 KWh	0,0 KWh
Fier de calcat	1000 W	x 0,5 h	x 1	x 7	= 0,5 KWh	15,0 KWh
Cuptor cu microunde	1500 W	x 0,3 h	x 1	x 7	= 0,5 KWh	13,5 KWh
Aragaz electric mare	2100 W	x 3,0 h	x 1	x 3	= 2,7 KWh	81,0 KWh
Aragaz electric mic	1250 W	x 0,0 h	x 0	x 0	= 0,0 KWh	0,0 KWh
Frigider (6 ani vechime)	500 W	x 5,0 h	x 0	x 7	= 0,0 KWh	0,0 KWh
Frigider (nou, economic)	200 W	x 5,0 h	x 1	x 7	= 1,0 KWh	30,0 KWh
Aspirator (manual)	100 W	x 0,0 h	x 0	x 0	= 0,0 KWh	0,0 KWh
Aspirator (vertical)	700 W	x 1,0 h	x 1	x 2	= 0,2 KWh	6,0 KWh
Masina de spalat (ax orizontal)	250 W	x 0,0 h	x 0	x 0	= 0,0 KWh	0,0 KWh
Masina de spalat (ax vertical)	900 W	x 3,0 h	x 1	x 2	= 0,8 KWh	23,1 KWh

Având în vedere că se cunoaște un consum energetic estimativ al casei, putem trece la partea de dimensionare și proiectare a surselor regenerabile de energie, astfel managerial, acest studiu are un caracter ciclic invers față de modul în care energia este prelucrată și ajunge către consumator după cum este prezentat și în *Figura 2.*

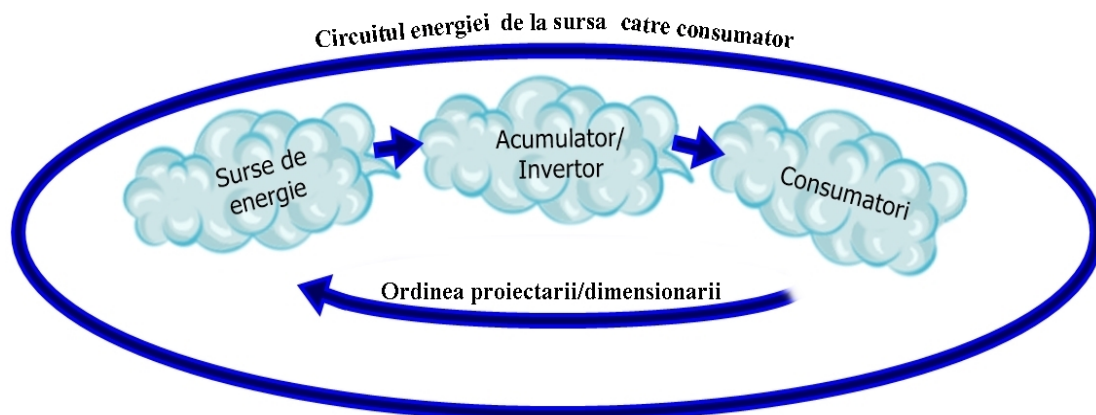


Fig.2 Schema ciclică a ordinii proiectării în funcție de circuitul energiei.

În ceea ce privește modul în care sursele de energie regenerabilă pot fi dispuse, acestea se combină în mod uzual cu un generator pe combustibil, deoarece în cazul în care sursa regenerabilă este în incapacitate să producă energie, alimentarea să nu fie întreruptă niciodată. Astfel se vor trata din 3 cazuri posibile: Turbina Eoliană asociată cu Generator Diesel; Panou Fotovoltaic asociat cu Generator Diesel; Hibrid: Turbina Eoliană+Panou Fotovoltaic+Generator Diesel; doar cel Hibrid după cum s-a figurat și în *Figura 1*.

Din *Tabelul 1*, în care este prezentat consumul lunar total pe categorii de consumatori putem observa un consum lunar de aproximativ: 484.9 kWh, astfel într-un an de zile se apreciază un consum de 5820 kWh. Cu aceste date putem trece direct la dimensionarea surselor de energie regenerabilă, acumulatorului și implicit invertorului.

## 2.1 Bateria de acumulatori

Luând în considerare rolul acumulatorilor în sistem și anume acela de a fi capabili să susțină rețeaua electrică pentru câteva ore la consum mediu sau chiar câteva zile pentru un consum minimizat de energie acesta a fost dimensionat să suporte aproximativ 200 KWh/lună pentru un număr de 3 zile a câte 10 ore după cum este prezentat și în *Figura 3*.

Introduceți Timpul de Rezerva dorit		Calculatie & Rezultate	
Introduceți nr de zile	3	Total ore de rezerva necesare	82Ore
Introduceți nr de ore	10	KWh utilizati pe ora	0.27Kwh
<b>Introduceți informatii despre sistemul protejat</b>		Total KWh necesar	22.77Kwh
Putere consumatori (W)	500	Nr de baterii necesare in serie	2
KWh pe Luna	200	KWh disponibil pe set	3.84Kwh
*Voltaj Sistem	24 Volti ▼	Seturi necesare	5.93
<b>Alegeți tipul de sistem dorit (click pt detalii)</b>		Rotunjire	6
** <a href="#">Select Tip Sistem</a>	Sistem SIMPLU ▼	<b>Sumar</b>	
<b>Selectează un tip de acumulator</b>		Nr baterii necesare	12Buc
Selectează Model:	Exide A512/200 A ▼	Pret Total baterii	5.280,00 €
		KWh disponibili	23.04Kwh
		<b>Acumulatorii alesi au rezerva disponibila pentru un:</b>	
		Numar de zile	3zile
		Numar de ore	10ore

Figura 3. Lista cu informații pentru proiectarea și dimensionarea acumulatorilor

## 2.2 Sursa hibridă de energie

Având în vedere faptul că modalitatea de furnizare a energiei necesare consumului casei dimensionate este printr-o modalitate hibrida, alcătuită din panouri fotovoltaice, turbine eoliene și un generator Diesel, dimensionarea acestora trebuie făcută cu mare atenție, deoarece totalitatea energiei create trebuie să fie suficientă și ideal să și depășească, dar nu cu mult, necesarul casei.

## 2.3 Panoul fotovoltaic

A fost selectat spre folosință 4 panouri fotovoltaice cu celule policristaline de ultimă generație ale cărui putere maximă este de 320 W și s-a calculat astfel încât acestea să producă energie în regim nominal aproximativ 8 ore pe zi. Cunoscând că acestea produc aproximativ o treime din puterea lui nominală, și anume 0.1066 kWh, făcând calculele,

acestea ar trebui să producă 3.41 kWh într-o zi, 105.82 kWh într-o luna de zile. Panoul fotovoltaic acoperă în proporție de 21.8% din necesarul de energie al casei, dar această cifră este orientativă.

## 2.4 Turbina eoliană

În alegerea și dimensionarea turbinei eoliene s-a luat în considerare că aceasta împreună cu panoul fotovoltaic să producă împreună o cantitate de energie suficientă cât să satisfacă necesarul casei și să producă și în exces în regim nominal, astfel încât în cazul în care condițiile meteorologice nu le permite o funcționare medie calculată, energia în exces produsă înmagazinată în bateria de acumulatori să suplinească nevoia de energie a casei.

Astfel s-a ales o turbină eoliană cu o putere de 1000 W care să funcționeze în medie 10 ore pe zi, astfel producând 10 kWh pe zi, 310 kWh pe luna. Turbina eoliană acoperă în proporție de 63.91% din necesarul de energie al casei, dar această cifră este de asemenea orientativă. Menționez că în procesul de calcul am adoptat o perspectivă realist-pesimistă, astfel turbina poate funcționa și mai mult de 10 ore pe zi, pentru 16 ore, spre exemplu, aceasta poate produce până la 496 kWh pe lună, acoperind de una singură tot necesarul de energie al casei.

## 2.5 Generatorul diesel

Generatorul Diesel este singura sursă de energie din sistem care nu funcționează cu energie regenerabilă și nici nu se dorește acest lucru. Acest generator are rolul de sursă de rezervă de producere a energiei și va fi pus în funcțiune doar în cazuri excepționale.

Acesta a fost dimensionat ca având o putere nominală de 4.4 kW, având un consum de 1.3 litri/oră cu o capacitate a rezervorului de combustibil de 5 litri și furnizează o tensiune de 230V la o frecvență de 50 Hz.

## 2.6 Domotică

Acest capitol va îngloba toată partea de inteligență și funcții implementate atât în interiorul cât și în exteriorul casei pentru a facilita experiența beneficiarilor printr-o interfață cât mai ușor de folosit.

Pentru a asigura un confort cât mai sporit, dar cu un minim de consum energetic, sistemul de climatizare a fost gândit astfel încât fiecare încăpere să fie încălzită pe intervale orare în funcție de cum vor fi ocupate, în celelalte camere păstrându-se pe timp de iarnă o medie de 15°C, iar pe timpul verii de 20°C.

Va fi implementat un sistem inteligent de hibernare a casei atunci când ocupanții acesteia vor fi plecați, astfel încât sistemul de iluminat va fi întrerupt automat precum și alimentarea prizelor, astfel doar anumiți consumatori rămânând branșați, și sistemul de securitate va fi armat.

Totodată în cazul în care condițiile meteorologice nu sunt favorabile, fie se înregistrează o intensitate a vântului mai puternică, fie plouă, geamurile se vor închide automat. Tot în acest subsistem se implementează și un sistem ce permite aerisirea casei.

Posibilitățile de îmbunătățire și de a adăuga funcții noi sistemului sunt numeroase având în vedere faptul că pentru implementarea lor se folosesc microcontrolere Arduino și senzori din aceeași platformă a căror costuri de hardware sunt foarte mici, software-ul fiind gratuit.

## 3. Concluzii

Pe baza unor calcule estimative se poate dimensiona orice model de casă sau imobil astfel încât să se poată proiecta pentru acestea o modalitate de obținere a energiei electrice necesare, mai prietenoasă cu mediul și independentă de orice furnizor de astfel de servicii. Fie că este din dorința proprie, ori infrastructura nu permite cuplarea la rețeaua unui furnizor de energie, această opțiune este una rentabilă având costuri de investiție ce se amortizează în timp.

În ceea ce privește partea de inteligență a casei, aceasta poate fi dezvoltată de asemenea după nevoile proprii, scopul principal fiind acela de a îmbunătăți calitatea vieții ocupanților.

## Bibliografie:

1. [http://www.lpelectric.ro/ro/support/calculator\\_ro.html](http://www.lpelectric.ro/ro/support/calculator_ro.html)
2. <https://egeneratoare.ro/>
3. <http://alternativepureenergy.ro/>
4. <http://www.alaska-energies.ro/>
5. <http://www.transselectrica.ro/web/tel/home>
6. <https://dexonline.ro/>
7. <https://e-panouri.eu>
8. [https://en.wikipedia.org/wiki/Autonomous\\_building](https://en.wikipedia.org/wiki/Autonomous_building)
9. <https://hobbytronica.ro/>

# STUDIU PRIVIND UN VEHICUL ELECTRIC CU MOTOR ASINCRON

**Autori:** Alexandru BUGESCH<sup>1</sup>, Alexandru-Mihai Chiuda<sup>1</sup>  
cosminrus@upet.ro

**Coordonatori:** Prof.univ.dr.habil.ing. Monica Leba<sup>2</sup>, Asist.cercet.drd.ing. Cosmin Rus<sup>2</sup>

<sup>1</sup> Universitatea din Petroșani, Facultatea I.M.E., specializarea: Calculatoare , anul II

<sup>2</sup> Universitatea din Petroșani, Facultatea I.M.E., Departamentul: A.C.I.E.E.

## Rezumat

Această lucrare oferă o imagine de ansamblu asupra vehiculelor electrice. Lucrarea descrie dezvoltarea și compararea diferitelor părți ale componentelor acestor tipuri de vehicule. Se examinează componentele principale ale tehnologiei bateriei, designului încărcătorului, motorului, direcției și frânării.

**Cuvinte cheie:** vehicul electric, control, acumulatori

## 1. Introducere

Semnalul de alarmă în dezvoltarea automobilului cu motor cu ardere internă, l-a dat poluarea atmosferei în zona urbană. Raportul anual din 9 Feb 1989, al World Watch Institute din Washington atrage atenția omenirii că, dacă nu se va reduce nivelul eliberării în atmosferă a substanțelor poluante există riscul ca locuitorii Terrei să declanșeze fenomenul de încălzire globală. Cantitățile mereu sporite de anhidridă carbonică (cu o creștere anuală de 0,4%), oxizi de azot, oxizi de carbon, vapori de plumb, clorofluorocarburi (gaze cu spor anual de 5%) contribuie la încălzirea atmosferei prin distrugerea stratului de ozon care protejează planeta de radiațiile ultraviolete. Pe lângă poluarea chimică a atmosferei un alt mare produs nociv de care se face vinovat automobilul față de mediul ambiant este zgomotul. Acționând asupra scoarței cerebrale, zgomotul provoacă iritații nervoase accelerează procesul de oboseală, slăbește atenția și reacțiile psihice. Apare așadar, evidenta oportunitate a utilizării unui vehicul nepoluant și silențios în circulația urbană această concluzie este cea care, la începutul anilor 60 a redeschis era automobilelor electrice. Cercetările au primit apoi, un nou impuls odată cu atingerea apogeului unui nou fenomen: criza energetică, fenomen de exploatare, care la nivelul anului 1973 a pus în evidența problema disponibilităților de rezerve de țiței, determinând statele lumii să treacă la programe concrete de cercetare și dezvoltare. Fiecare țară încearcă alternative pentru țiței, în funcție de sursele de energie autohtone, gaze naturale (rezerve epuizabile aproape în aceeași măsură ca și țițeiul), cărbune precum producerea de metanol, hidrogen, energii regenerabile, etc. Utilizarea energiei electrice este favorabilă prin faptul că infrastructura necesară, sub forma sistemelor de distribuție publică este deja disponibilă și poate fi folosită cu mici modificări. De asemenea electricitatea, spre deosebire de alcoolii sau benzină sintetică poate fi produsă din toate sursele primare de energie cunoscute, ceea ce reprezintă un factor esențial în eventualitatea că una din aceste forme devine deficitară. În comparație cu vehiculele propulsate prin motoare cu combustie internă, VEM autonome alimentate de la acumulatori electrice prezintă un randament electric mai superior. În timp ce la vehiculele din prima categorie numai 14% din energia totală este folosită pentru tracțiune, la VEM energia utilizată este de 50 % din energia totală. Dezvoltarea vehiculelor electrice este strâns legată de evoluția electricității și a tehnologiilor de conversie a energiei electromecanice. La începutul anilor 1800, mijloacele de transport erau propulsate doar prin puterea aburului, deoarece legile inducției electromagnetice, și prin urmare, motoarele și generatoarele electrice nu erau încă descoperite. Faraday a descoperit în 1831 legea inducției electromagnetice și la scurt timp a inventat motorul de curent continuu. Primele vehicule echipate cu baterii nereîncărcabile au fost testate în 1834 și 1851. Bateriile electrice cu plumb au fost realizate în 1859, iar în 1874 au fost utilizate pe un vehicul electric [1, 2, 3].

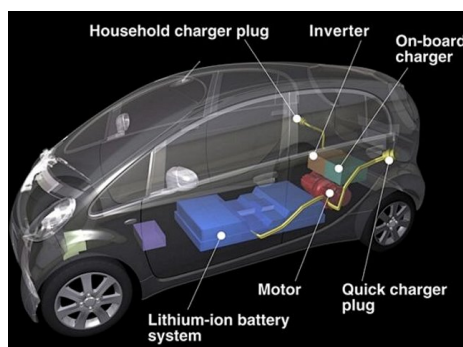


Fig. 1 Modul de amenajare a echipamentului electric

## 2. Schema bloc a instalației electrice la electromobil

Pedala de accelerație este conectată la două potențiometre, care traduc mișcarea de translație a pedalei într-un

semnal variabil ce comanda inverterul. Semnalul dat de potențiometre spune inverterului cat de multa putere sa furnizeze motorului electric. Sunt doua potențiometre din motive de siguranță, inverterul citește ambele potențiometre si se asigura ca ambele dau același semnal. Daca nu, el nu funcționează.

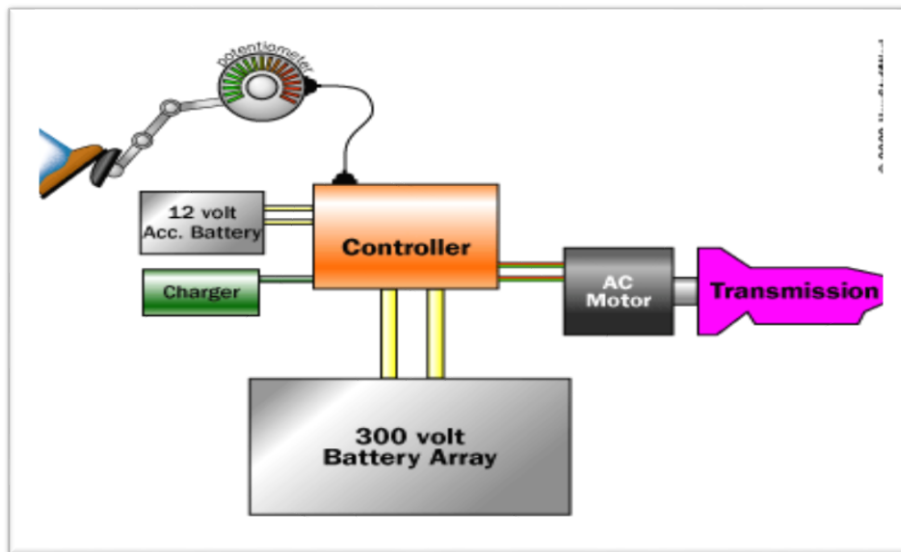


Fig. 2 Schema bloc a electromobilului cu motor asincron

### 3. Invertorul

Invertorul transforma curentul continuu furnizat de baterii, in curent alternativ trifazat. Motorul de curent alternativ va avea o tensiune de alimentare cuprinsa intre 240 – 300 V. Motoarele de curent alternativ sunt folosite aproape in intreaga industrie, deci se poate gasi mult mai ușor unul potrivit si pentru un vehicul electric. In timpul frânării, motorul se transforma in generator si încarcă bateriile prin intermediul inverterului, care poate funcționa si ca redresor. Construcție tipică a unității inverter ar fi: partea formată din semiconductoarele de putere cu elementele de detecție a curentului și temperaturii, interfața și alte circuite de control . În figura 3 este prezentată o schemă bloc a unui etaj de putere, tipic pentru un inverter trifazat . Invertorul de putere este realizat pe baza tehnologiei IGBT. Cu tehnici avansate de ambalare se poate reduce volumul unității de control [3, 6].

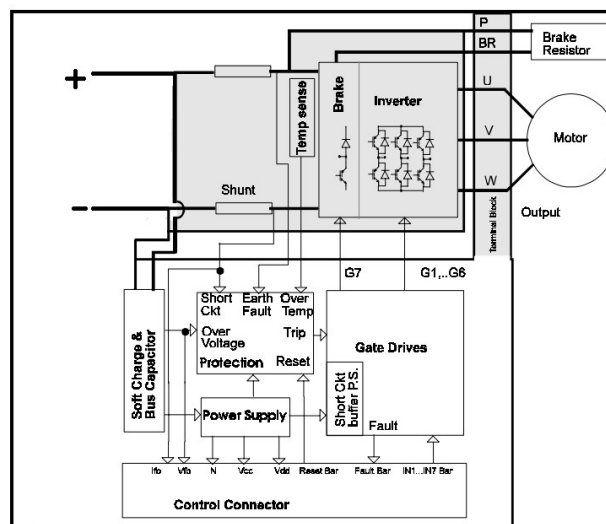


Fig. 3 Schema electrică principală a inverterului

### 4. Reglarea vitezei

La pornire se asigură creșterea vitezei mașinii asincrone cu menținerea constantă a raportului  $U_1/f_1$ , astfel că fluxul magnetic al mașinii se păstrează nemodificat. Însușind frecvența rotorică (sau de alunecare) cu frecvența de rotație, (măsurată) a mașinii asincrone se obține frecvența statorică. După atingerea valorilor nominale ale tensiunii și frecvenței, reglarea vitezei se realizează la tensiune constantă, prin modificarea frecvenței. La creșterea frecvenței are loc micșorarea fluxului în mașină. Prin menținerea constantă a frecvenței rotorice, caracteristicile mecanice ale mașinii asincrone devin elastice, similare cu cele ale mașinii de c.c. cu excitație serie [6].

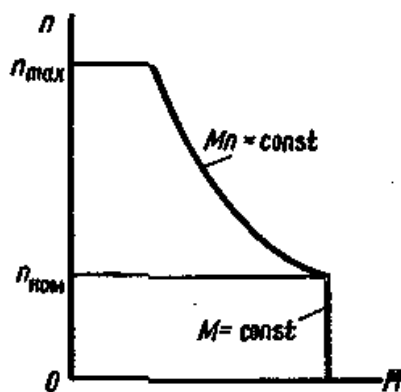


Fig. 4 Caracteristicile mecanice ale motorului pentru tracțiune

În figura 5 reprezintă dependența cuplului electromagnetic în unități relative  $M / M_{nom}$  la  $f_2$  frecvență rotorului pentru diferite valori ale tensiunii. În regim de motor cuplul maxim este mărit esențial cu ajutorul reducerii semnificative a frecvenței.

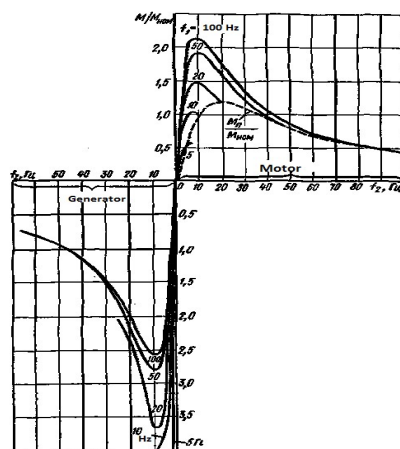


Fig. 5 Dependența  $M = f(f_2)$  la reglarea  $U_1 / f_1 = \text{const}$  pentru diferite frecvențe  $f_1$

### 5. Modulul de încărcare a bateriilor

Bateria auto este un element esențial pentru confortul călătoriilor noastre. De calitatea ei depinde punctualitatea noastră, dar și buna dispoziție și siguranța.

Modulul de încărcare a bateriei este realizată pe o schemă clasică ce prezintă un transformator de coborîre cu tensiunea pe secundar de 25 V. Tensiunea este redresată cu ajutorul unor ventile necomandate, apoi filtrată cu un condensator electrolitic. Pentru stabilizarea tensiunii se va folosi microschema L200 care va comanda cu 3 tranzistoare bipolare MJ15004, legate în paralel. Curentul de încărcare se va calcula în funcție de capacitatea bateriilor, se va lua 0,1 unități din capacitatea totală a bateriilor [8-13].

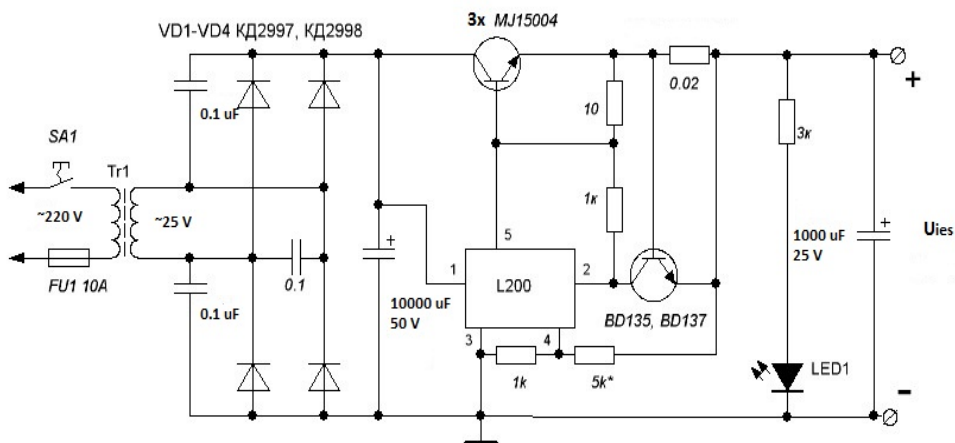


Fig. 6 Schema electrică a modului de încărcare a bateriilor



## 6. Tipuri de baterii

- acumulatori plumb-acid;

Caracteristici teoretice:

- tensiunea unei celule : 2,1 V;
- densitatea de energie : masica 161 Wh/kg; volumica 686 Wh/l.

În cazul acumulatorilor cu plumb-acid uzuali, apa din electrolit se pierde în timp (prin evaporare, precum și prin descompunerea ei în  $H_2$  și  $O_2$  care se degajă la electrozi); de aceea, ea trebuie completată periodic. Pentru evitarea sulfatării electrozilor, acumulatorii trebuie reîncărcați imediat după descărcarea lor. De asemenea, are loc o autodescărcare (descărcare fără a avea conectată o sarcină) a acumulatorilor, ceea ce impune reîncărcarea lor după perioade mari de neutilizare. Rezultă necesitatea unei întrețineri pretentioase a acestor acumulatori.

Tehnologiile moderne permit realizarea acumulatorilor cu plumb-acid în așa numita variantă “fără întreținere”. În acest caz acumulatorii sunt capsulați, au o construcție adecvată a electrozilor, iar electrolitul este “solidificat”, fie fixat într-un gel, fie absorbit în materiale poroase de tip “vată”. La aceste acumulatori, pierderile de apă sunt neglijabile, autodescărcarea este foarte redusă (circa 35 % din capacitatea nominală, după 12 luni), iar sensibilitatea lor la descărcări ocazionale outernice, la vibrații și la temperaturi joase este mult scăzută. Deși folosesc electrozi solidificați, densitatea de energie mai ales la descărcări rapide și temperaturi scăzute este mai mare decât a acumulatorilor uzuali. Desigur toate acestea se obțin cu prețul unui cost mai ridicat.

### Avantaje:

- sunt relativ ieftine;
- au un număr relativ mare de cicluri de încărcare – descărcare;
- folosesc pentru electrozi plumbul care este disponibil în cantități mari;
- se poate realiza în varianta “fără întreținere”;
- există producție în serie;
- la ieșirea lor din funcțiune, plumbul din electrozi poate fi ușor reciclat.

### Dezavantaje:

- au densitate redusă a energiei, ceea ce implică greutate și volume mari;
- puterea lor scade odată cu descărcarea;
- capacitatea și energia lor scad, la temperaturi scăzute;
- oferă posibilități limitate de încărcare rapidă (încărcarea lor completă se face pe timp de câteva ore);
- sunt sensibile la supracurenți (de încărcare sau descărcare).

- acumulatori Ni-Cd;

Caracteristici teoretice:

- tensiunea unei celule : 1,3 V;
- densitatea de energie : masica 209 Wh/kg; volumica 693 Wh/l.

Dintre celelalte tipuri de acumulatori, perspectiva de a fi utilizați în tractiune, datorită energiilor lor specifice ridicate, o au cele cu sodiu-sulf (Na-S), precum și cele bazate pe litiu (Li). Pentru a obține o conductivitate suficientă a electrolitului solid este necesară o temperatură ridicată (între 300 – 400 °C; uzual 330 °C). Funcționarea la o astfel de temperatură necesită o izolare termică compactă și eficientă. De regulă se folosește o izolație termică vacuumată care acoperă complet celulele acumulatorului, lăsându-se doar locuri pentru realizarea conexiunilor electrice. În încălțarea termică se instalează un încălzitor electric precum și un schimbător de căldură lichid (pentru răcirea celulelor în cazul unor descărcări puternice).

### Avantaje:

- au un număr foarte mare de cicluri de încărcare descărcare;
- puterea lor se menține relativ constantă după o descărcare parțială;
- au o comportare foarte bună la temperaturi scăzute;
- oferă posibilitatea reîncărcării rapide (se pot realiza încărcări cu curenți de până la 10 In);
- nu necesită întreținere (acumulatori capsulați);
- există producție în serie.

### Dezavantaje:

- sunt scumpe (datorită costului ridicat al cadmiului);
- pun probleme ecologice, cadmiul fiind toxic;
- încărcarea lor pune probleme la temperaturi ridicate;

- acumulatori Na-S;

Caracteristici teoretice:

- tensiunea unei celule : 2,1 V;
- densitatea de energie : masica 792 Wh/kg; volumica 1196 Wh/l.

Principalele obstacole în utilizarea acestor baterii pe automobilele electrice sunt: numărul redus de cicluri încărcare – descărcare și puterea specifică scăzută la funcționarea continuă.

## 7. Utilizarea ultracondensatoarelor

Conectând în paralel cu bateria de acumulare condensatoare cu capacități ridicate (actualmente se realizează condensatoare cu dimensiuni rezonabile, având capacități de ordinul a 600 F, la 3 V), acestea din urmă pot asigura energiile necesare în cazul pornirilor vehiculului sau al altor suprasarcini, menajând, astfel, bateria de acumulare și asigurând o autonomie mai ridicată a autovehiculului.

## 8. Concluzii

Avantaje și dezavantaje ale folosirii autovehiculelor electrice în locul autovehiculelor cu motoare clasice cu combustie internă.

### Avantaje:

- au un cost de exploatare aproape gratuit (circa 1 euro/100km);
- se pot conduce ușor, (unele nu au comenzi cu pedale pot fi utilizate și de către anumite persoane cu handicap fizic);
- întreținere simplă, ușor de manevrat în parcare și de încărcat bateriile;
- nu fac zgomot și nu emana fum, oxid de carbon, metale grele și alte noxe daunătoare mediului și vieții;
- nu consumă suplimentar la staționarea temporară - în trafic - la semafoare, etc;
- posibilitatea de utilizare a unor sisteme de acționare sofisticate, oferite de cele mai moderne realizări în domeniul acționării electrice;
- posibilitatea realizării comode a sistemelor de frânare antiblocante, prin utilizarea frânării electrice; dacă frâna este recuperativă, se face și o importantă economie de energie;
- posibilitatea de acționare individuală a roților (eventual, prin înglobarea motoarelor de tracțiune în roți, realizând așa-numitele motoroti); aceasta conduce la simplificarea sistemelor de transmisie, cea mai importantă fiind eliminarea diferențialului mecanic;
- au cuplu constant la variații ale turatiei în limite mari.

### Dezavantaje:

- principalul dezavantaj este acela al autonomiei reduse;
- au cost de achiziție ridicat;
- bateria de acumulatori are o densitate de energie scăzută, o durată de viață relativ mică limitată de numărul de cicluri încărcare – descărcare, timpi mari de încărcare;
- cheltuielile pentru întreținerea bateriilor sunt destul de ridicate;
- sunt necesare stații de încărcare a bateriilor de acumulare; acestea pot fi dotate fie cu acumulatori preîncărcate, care să le schimbe pe cele descărcate de pe automobile - ceea ce pune probleme de depozitare și de asigurare a unei diversități de baterii, în funcție de tipurile de automobile existente -, fie cu instalații de încărcare a bateriilor direct pe automobile; în ultimul caz, se pune problema timpului de încărcare, acesta fiind de ordinul orelor pentru o încărcare completă normală (doar pentru încărcări parțiale, el poate fi redus la ordinul minutelor-zecilor de minute);
- sunt necesare investiții inițiale mari, dacă producția automobilelor este de serie mică

## Bibliografie

- [1] Rus C, Leba M and Ionica A 2018 Proc. 17th edition International Technical-Scientific Conference, Modern Technologies for the 3rd Millennium (March, Oradea, Romania) p 351
- [2] Rus C, Negru N, Leba M and Ionica A 2018 Proc. Int. Conf. GLOREP (Timisoara, Rom.) p 237
- [3] Patrascoiu N and Rus C 2019 Quality - Access to Success 20 347
- [4] Rebrisoreanu M, Rus C, Leba M and Ionica A 2018 International Journal Of Systems Applications, Engineering & Development 12 164
- [6] Patrascoiu N and Rus C 2018 Proc. 19th International Carpathian Control Conference p 318
- [7] <http://ro.wikipedia.org>
- [8] [www.venturifetish.fr](http://www.venturifetish.fr)
- [9] [www.teslamotors.com](http://www.teslamotors.com)
- [10] [www.wrightspeed.com](http://www.wrightspeed.com)
- [11] <http://www.econogics.com/ev/evhistory.htm>
- [12] <http://www.newton.mec.edu>
- [13] <http://www.dassault.fr>
- [14] <http://sloan.stanford.edu>
- [15] <http://mikes.railhistory.railfan.net>
- [16] <http://inventors.about.com>

# APLICAȚIE SOFTWARE PENTRU RECUNOAȘTEREA UNEI PERSOANE

**Autori:** Andrei-Gabriel BERCEANU <sup>1</sup>, Radu-Alexandru PÎRCIU <sup>2</sup>  
berceanu.andreigabriel@yahoo.com/ alexandrur37@gmail.com

**Coordonatori:** Conf. Univ. Dr. Ing. Egri Angela <sup>3</sup>

<sup>1</sup> Universitatea din Petroșani, Facultatea I.M.E., specializarea: Automatică, anul III

<sup>2</sup> Universitatea din Petroșani, Facultatea I.M.E., specializarea: Calculatoare, anul II

<sup>3</sup> Universitatea din Petroșani, Facultatea I.M.E., Departamentul: A.C.I.E.E.

## Rezumat

*Această lucrare este formată din două părți: una hardware și un software. Pentru partea hardware folosim o placă Raspberry Pi 3, care oferă funcționalități specifice unui sistem PC, având unitate de procesare, memorie. Pentru această aplicație este necesară și o cameră Raspberry Pi Module V2. Partea software este realizată utilizând OpenCV, format din librării ce au incluse diferite unelte folosite în prelucrarea imaginilor.*

## Cuvinte cheie

Raspberry Pi, sistem de recunoaștere, OpenCV.

### 1. Introducere

Un sistem de recunoaștere facială este o aplicație capabilă să identifice o persoană dintr-o imagine digitală și să compare diferitele trăsături ale persoanei cu unele stocate în prealabil. Există mai multe modalități de a analiza cele două imagini și a determina dacă există o compatibilitate între acestea. Majoritatea algoritmilor analizează anumite trăsături faciale cum ar fi poziția și forma ochilor, a nasului, a gurii, etc.

O problemă controversată în ceea ce privește sistemele de recunoaștere facială este cea a eficienței sistemelor de acest tip, care în comparație cu alte tehnici biometrice nu este tocmai eficientă. Este pusă sub semnul întrebării eficiența sistemelor de acest tip în locuri precum aeroporturi, gări, însă recunoașterea facială permite identificarea persoanelor în mulțime, lucru care nu este posibil în cazul altor sistemelor biometrice.

Recunoașterea facială este din ce în ce mai populară, fiind multe aplicații care o folosesc. Ca exemple putem da Iphone X de la Apple, care folosește un sistem de acest tip care permite utilizatorului să-și deblocheze telefonul prin intermediul unei amprente digitale captate de camera acestuia, Facebook ce folosește recunoașterea facială pentru a eticheta persoane în fotografii, deoarece la fiecare etichetare, soft-ul reține informații despre trăsăturile persoanei, pentru ca mai apoi să poată eticheta în mod automat persoana respectivă.

Există numeroase soft-uri gratuite pe care dezvoltatorii le pot folosi pentru a crea aplicații de recunoaștere facială. Printre acestea putem numi Rekognition, un soft dezvoltat de Amazon, cei de la Google pun la dispoziția utilizatorilor Google Cloud Vision API. Alte soft-uri asemănătoare sunt BioID, un serviciu web pentru recunoaștere facială și vocală, Microsoft Project Oxford, un algoritm ce poate detecta și recunoaște fețele și OpenCV, cel utilizat în acest proiect și despre care vom discuta mai pe larg în cele ce urmează.

### 2. Raspberry Pi

Raspberry Pi 3 este o placă de dezvoltare de tip SBC (Single Board Computer) – sistem de calcul nemodular implementat pe un singur cablaj electronic. Chiar dacă dimensiunile sunt reduse (85mm x 56mm), Raspberry Pi este un calculator complet care permite funcționalități obișnuite precum rularea unui sistem de operare (Linux sau Windows) și rularea de aplicații utilizator (jocuri, editoare de text, medii de programare, redarea de muzică și filme, aplicații de teleconferință, aplicații Internet).

Diferențele între o placă Raspberry Pi și calculatorul personal sau laptop constau atât în dimensiunea redusă a plăcii cât și în puterea mai mică de calcul a acesteia – nu are aceleași performanțe de calcul precum un PC desktop sau laptop care au un cost și o dimensiune de câteva ori mai mari. Putem compara placa Raspberry Pi cu o tabletă sau cu un sistem de tip NetBook dar fără a dispune de ecran și tastatură. În plus, placa Raspberry Pi oferă posibilitatea de a conecta diverse componente electronice specifice sistemelor embedded: senzori, butoane, ecrane LCD sau pe 7 segmente, drivere de motoare, relee etc.

Posibilitatea de a personaliza sistemele de programe (sistemul de operare, aplicațiile) și posibilitatea de interconectare cu alte componente electronice fac din placa Raspberry Pi un sistem de calcul ce poate sta la baza unor proiecte personale extrem de interesante și de puternice – un calculator ce poate fi integrat în sisteme electronice și mecanice proiectate și realizate de utilizator.

Placa are următoarele specificații:

- Procesor SoC pe 64 de biți din familia ARMv8-A, Broadcom BCM2837, ce lucrează la o frecvență de 1.2GHz și dispune de 4 nuclee de tip ARM Cortex-A53;
- 1GB de memorie RAM (folosită și ca memorie video, partajată cu procesorul grafic);
- Procesor grafic Broadcoam VideoCore IV 3D integrat pe aceeași pastilă de siliciu ca și procesorul principal;
- Ieșire digitală video / audio HDMI;
- Ieșire analogică video (composite video) / audio mixtă prin intermediul unei mufe jack 3.5mm;
- Mufă de rețea RJ45 Ethernet 10/100 Mbit/s;
- Conectivitate WiFi 802.11n;
- Conectivitate Bluetooth 4.1 / BLE;
- 4 porturi USB 2.0;
- 40 de pini de intrare / ieșire (GPIO);
- Slot card de memorie microSD (utilizat pentru instalarea sistemului de operare);
- Conectori dedicați pentru cameră video (CSI) și afișaj (DSI);

Pentru conectarea plăcii Raspberry Pi avem nevoie de următoarele componente suplimentare:

• Cablu HDMI și un monitor HDMI. În cazul în care nu avem un monitor cu intrare HDMI putem utiliza un adaptor HDMI-DVI sau un adaptor HDMI-VGA, în funcție de monitorul pe care îl folosim.

• Alimentator cu ieșire de 5V, minim 2.5A și mufă microUSB. Se recomandă utilizarea unui alimentator original sau a unui alimentator de calitate pentru a asigura o tensiune corectă și un curent suficient alimentării plăcii. Dacă utilizăm un alimentator ieftin putem distruge placa din cauza fluctuațiilor de tensiune sau putem întâmpina probleme în utilizare din cauza curentului insuficient dat de sursa de tensiune.

• Mouse și o tastatură USB. Necesare pentru instalarea și configurarea inițială a sistemului. Dacă, ulterior, se utilizează sistemul de la distanță, tastatura, mouse-ul și monitorul nu mai sunt necesare. O variantă mai simplă este utilizarea unui tastatură cu TouchPad.

• Card de memorie microSD, capacitate de 16GB recomandat. Cardul de memorie stochiază sistemul de operare la fel ca și hard-disk-ul în cazul unui sistem de calcul de tip PC sau laptop. Este importantă utilizarea unui card microSD de calitate și de viteză deoarece utilizarea unui card de calitate îndoielnică poate conduce la probleme de funcționare extrem de neplăcute: blocări în funcționare, resetarea aleatorie a sistemului, pierderi de date etc.

• Dacă utilizăm sistemul într-o rețea locală pe cablu este necesar și un cablu de rețea UTP –patch-cord. Dacă placa este utilizată într-o rețea locală WiFi nu este necesar. Conectivitatea de rețea (conectivitatea Internet) nu este obligatorie pentru funcționarea plăcii dar este recomandată deoarece altfel nu se realizează actualizările de securitate ale sistemului de operare, nu se pot menține corect setările de dată și oră și, bineînțeles, se pierde o funcționalitate importantă a sistemului.

• Opțional putem utiliza o carcasă pentru placa Raspberry Pi. Dacă nu este folosită o carcasă pentru placa Raspberry Pi riscăm deteriorarea acesteia datorită descărcărilor electrostatice (descărcarea sarcinii electrice statice a corpului uman în circuitul electronic prin atingerea componentelor conductoare a acestuia), a șocurilor mecanice și a murdăriei (praf, lichide, grăsimi).

Pentru instalarea plăcii Raspberry Pi3 avem nevoie de cadrul microSD pe care se instalează sistemul de operare. Placa Raspberry Pi 3 poate rula diverse distribuții ale sistemului de operare Linux și o versiune minimală a sistemului de operare Microsoft Windows 10 (prin versiune minimală înțelegem o variantă ce nu poate fi folosită ca platformă desktop ci doar ca platformă pentru aplicații –Windows 10 IOT Core). Vom exemplifica instalarea distribuției Linux Raspbian, sistemul de operare oficial al plăcii Raspberry Pi, sistemul de operare este foarte ușor de utilizat și recomandat pentru începători.

Dacă se utilizează card-ul oficial microSD al plăcii Raspberry Pi instalarea sistemului de operare este foarte simplă deoarece cardul de memorie conține utilitarul NOOBS (New Out Of Box Software) pentru instalarea diverselor sisteme de operare specifice plăcii Raspberry Pi. Trebuie să ne asigurăm că avem toate echipamentele conectate corect: tastatură, mouse și cablu HDMI, pentru pornirea instalării sistemului de operare se inserează cardul în slotul microSD al plăcii și se pornește sistemul. După inițializare utilitarul NOOBS permite alegerea sistemului de operare pe care dorim să-l instalăm.

Dacă placa Raspberry Pi 3 nu este conectată la Internet (prin cablu sau prin WiFi) singura opțiune de instalare va fi sistemul de operare Raspbian al cărui kit de instalare se află deja pe cardul de memorie. Toate celelalte sisteme de operare necesită conectivitate la internet pentru instalare. Conexiunea la internet se realizează prin conectarea cu un cablu la o rețea ce oferă setări automate (DHCP) sau configurând accesul WiFi (opțiunea Wifi Networks din partea de sus a ferestrei anterioare). După confirmarea sistemului de operat dorit, utilitarul NOOBS va instala automat sistemul de operare – procesul durează câteva zeci de minute.

Dacă nu dispunem de un card microSD cu NOOBS preinstalat putem utiliza un card microSD pe care copiem utilitarul NOOBS – se downloadează și se dezarchivează pe cardul microSD. După această operație instalarea decurge ca în cazul precedent.

Alternativă la utilizarea programului NOOBS este copierea directă a sistemului de operare pe cardul microSD. Copierea nu se poate face direct, ca în cazul NOOBS, ci se realizează prin transferul unui fișier imagine cu ajutorul unui utilitar specializat, de exemplu:

Etcher sau Win32DiskImager. Downloadăm imaginea sistemului de operare dorit (variantea LITE nu include interfața grafică) și o copiem pe cardul de memorie. După această operație cardul de memorie va conține sistemul de operare deja instalat, gata de utilizare. Cardul de memorie se introduce în slotul plăcii și pornim sistemul.

După ce avem instalat sistemul de operare, trebuie să-l configurăm. Pentru aceasta avem două opțiuni: folosind utilitarele în linie de comandă (Terminal) sau folosind utilitarele puse la dispoziție de interfața grafică a sistemului de operare Raspbian. Utilitarul principal de configurare se numește raspi-config acesta poate fi accesat atât prin intermediul interfeței grafice cât și în linie de comandă.

### 3. OpenCV

Acum că placa este configurată, putem să trecem la crearea propriu-zisă a sistemului de recunoaștere facială. Așa cum am precizat mai devreme, vom realiza aceasta utilizând OpenCV. Aceasta este cea mai populară bibliotecă ce conține funcții pentru astfel de sisteme. Acesta a fost dezvoltat de firma Intel, fiind scris inițial în C/C++, dar recent suportă și versiune pentru Python.

OpenCV folosește algoritmi de învățare automată pentru căutarea și recunoașterea feței într-o fotografie. Aceasta nu este o sarcină ușor de îndeplinit, de aceea trebuie făcute mai multe teste pentru diferite șabloane și trăsături verificate în vederea stabilirii unei potriviri între imaginia originală și cea captată.

Pentru a realiza proiectul în sine, am parcurs mai mulți pași. În primă instanță, trebuie să instalăm userland. Acesta conține toate fișierele și codul sursă necesare manipulării OpenCV-ului. După ce am descărcat și am copiat fișierele în /opt/vc, folosim următoarele comenzi pentru build și compilare:

```
sudo mkdir build
cd build
sudo cmake -DCMAKE_BUILD_TYPE=Release ..
sudo make
sudo make install
```

După ce am terminat, trebuie să înlocuim conținutul fișierului CmakeLists.txt cu:

```
project( camcv )
SET(COMPILER_DEFINITIONS -Werror)
include_directories(/opt/vc/userland/host_applications/linux/libs/bcm_host/include)
include_directories(/opt/vc/userland/interface/vcos)
include_directories(/opt/vc/userland)
include_directories(/opt/vc/userland/interface/vcos/threads)
include_directories(/opt/vc/userland/interface/vmcs_host/linux)
add_executable(camcv RaspiCamControl.c RaspiCLI.c RaspiPreview.c camcv.c)
target_link_libraries(camcv /opt/vc/lib/libmmal_core.so /opt/vc/lib/libmmal_util.so
/opt/vc/lib/libmmal_vc_client.so /opt/vc/lib/libvcos.so /opt/vc/lib/libbcm_host.so )
```

Acum camera este gata de utilizare și putem face asta compilând următoarele linii de cod:

```
cmake .
make
./camcv -t 1000
```

Acum trebuie să creăm o legătură cu librăriile OpenCV și pentru aceasta copiem următorul cod în CmakeLists.txt:

```
project( camcv )
SET(COMPILER_DEFINITIONS -Werror)
#OPENCV
find_package( OpenCV REQUIRED )
link_directories( /home/pi/libfacerec-0.04 )

include_directories(/opt/vc/userland/host_applications/linux/libs/bcm_host/include)
include_directories(/opt/vc/userland/interface/vcos)
include_directories(/opt/vc/userland)
include_directories(/opt/vc/userland/interface/vcos/threads)
include_directories(/opt/vc/userland/interface/vmcs_host/linux)
add_executable(camcv RaspiCamControl.c RaspiCLI.c RaspiPreview.c camcv.c)
target_link_libraries(camcv /opt/vc/lib/libmmal_core.so /opt/vc/lib/libmmal_util.so
/opt/vc/lib/libmmal_vc_client.so /opt/vc/lib/libvcos.so /opt/vc/lib/libbcm_host.so /home/pi/
libfacerec-0.04/libopencv_facerec.a ${OpenCV_LIBS})
```

Acum putem prelua imagini sau video utilizând camera. Pentru aceasta vor mai fi necesare câteva mici ajustări.

#### 4. Concluzii

Această aplicație poate fi utilizată în multe domenii, precum: sisteme de supraveghere, protejarea datelor (în combinație cu alte sisteme biometrice precum recunoașterea amprentelor).

Am dori ca pe viitor să dezvoltăm această aplicație pentru a putea fi folosită la sisteme ATM. Am dori să introducem și un senzor pentru captarea sunetelor, iar în momentul în care este declanșat un semnal de alarmă (un țipăt), se vor alerta autoritățile, trimițând și imagini de la fața locului.

O altă posibilă aplicație ar fi un sistem de evidență a prezenței studenților: în momentul în care aceștia intră la curs, se vor apropia de cameră, care va consemna prezența acestora. Datele vor putea fi înregistrate și prelucrate ulterior de către profesor.

#### Bibliografie:

1. A. Egri and V. Sirb, „Aplicatii software inteligente”, Petrosani, 2014
2. <https://www.raspberrypi.org>
3. <https://github.com>
4. <https://stackoverflow.com>
5. <https://en.wikipedia.org>

# FASHION STYLE – PLATFORMĂ PENTRU VÂNZARE ONLINE DE PREDUSE VESTIMENTARE

**Autor: Andreea-Alexandra ITU**<sup>1</sup>

[deeaandreea466@yahoo.com](mailto:deeaandreea466@yahoo.com)

**Coordonator Conf.univ.dr.ing. EGRI Angela**

**Coordonator Șef.lucr.dr.ing. SÎRB Vali**

<sup>1</sup>*Universitatea din Petroșani, Facultatea de Inginerie Mecanică și Electrică, Specializarea: Calculatoare, anul 4*

## **Rezumat:**

*Many of the businesses today for sell products become online to accesible to a high number of people. Because of this reason we decided to build a online platform which can be easy to use and work fast. Fasion Style is a platform which is dedicated to sell products from clothing industry for the moment. This platform can be accesed from any computer who have acces to internet. The advanced of this platfom is the high speed of work and have a friendly interface. The reason why this platfom work fast is behind angularJS which use the MVC. With help of angularJS the store need load just one time the web aplication and the customer have acces to all pages. To say all pages is much because this platfom have 1 / one page but we saw many pages the reason is that we use angularJS. For build this platform we used: mongoDB, nodeJS, HTML, Bootstrap, JavaScript and of corse AngularJS.*

## **Cuvinte cheie:**

*AngularJS HTML CSS MVC NodeJS*

## **Capitolul 1 Introducere**

Fashion Style este o platformă care folosește arhitectura AngularJS-MVC( model view controller). Această arhitectură izolează logica de business față de considerentele interfeței cu utilizatorul. Prin așa ceva rezultă o aplicație unde aspectul vizual și nivelele inferioare ale regulilor de business sunt mai ușor de modificat, fără a afecta alte nivele.[1] Cele trei elemente ale lui MVC au următorul rol:

- Model- este nivelul cel mai jos responsabil cu menținerea datelor[3]
- View- este responsabil cu vizualizarea datelor de către utilizator[3]
- Controller- este codul software care controlează interacțiunea dintre model si view[3]

Arhitectura AngularJS-MVC devine destul de mult utilizată cu timpul datorită la ceea ce am zis mai sus. Multe platforme online caută să treacă pe această arhitectură datorită avantajelor acestuia.

Fashion Style vinde produse din domeniul vestimentar sub forma unei aplicații web. Acesta utilizează o singură pagină HTML deși pentru un utilizator care nu are cunoștințe despre AngularJS și MVC pare că are o multitudine de pagini.

## **Capitolul 2 Realizarea**

În cadrul realizării platformei avem patru părți: baza de date, serverul local pentru pagini, aplicația pentru administratorul platformei și aplicația pentru client.

### **2.1 Baza de date și serverul local pentru pagini**

Pentru ca platforma să funcționeze are nevoie de două servere.

Primul și cel mai ușor de realizat pentru exemplul nostru este cel pe care va rula paginile. Acest server a fost realizat utilizând NodeJS. Este ușor de realizat deoarece este local astfel nu suntem întâmpinați de diversele probleme care apar atunci când hostăm în rețea.

Al doilea server reprezintă baza de date. Acest servăr a fost realizat cu ajutorul bazei de date MongoDB. Am ales MongoDB pentru că are precizia de control foarte bună când vine vorba de plasarea globală a datelor deci putem asigura performanțe foarte bune din punct de vedere al vitezei de lucru cu datele. În figura de mai jos se poate vedea modul în care sunt pornite cele două servere și codul pentru serverul local pe care rulează aplicația web.

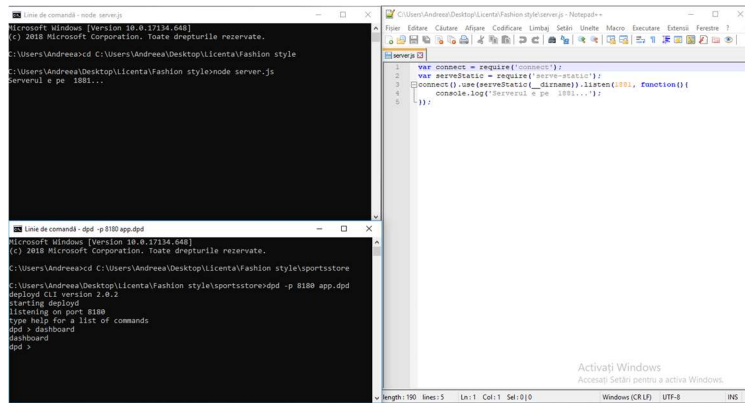


Fig.1 Pornirea celor două servare

După cum am zis cu ajutorul lui NodeJS am realizat destul de ușor serverul local pentru aplicația web, se poate vedea că avem doar 5 linii de cod restul fiind făcut în spate de NodeJS. Tot ce avem de făcut este să folosim comanda „node” alături de numele fișierului care conține codul.

## 2.2 Aplicația pentru client și administrator

Aplicația în sine a fost realizată utilizând: HTML, CSS (în special bootstrap), JavaScript și AngularJS.

HTML și CSS au rolul să realizeze interfața cu utilizatorul. Pentru a face un aspect cât mai plăcut am utilizat bootstrap. Bootstrap este o bibliotecă CSS care are diverse modele prin care putem da un aspect plăcut paginii noastre.

Cât despre JavaScript acesta cuprinde AngularJS. AngularJS este o bibliotecă în care avem diverse directive ce o să ne ajute în realizarea platformei.

AngularJS este folosit pentru a realiza ceea ce se întâmplă în spatele aplicației. AngularJS este open source, complet gratis și utilizat destul de mult în lumea noastră.

Cu ajutorul lui noi am creat o singură pagină HTML. În aceea pagină noi am încărcat diverse views. Este un avantaj destul de bun pus la dispoziție de AngularJS deoarece ne împarte aplicația în cât mai multe părți astfel că dacă vrem să modificăm ceva nu trebuie să modificăm aplicația în diverse locuri, decât într-un singur loc. Ca și exemplu în aplicație pe pagina principală avem un coș de cumpărături atunci când dăm click pe el se încarcă un nou view care afișează cu totul alt ceva. Clientul crede că acesta a trecut la o altă pagină când de fapt se află pe aceeași pagină. La fel se întâmplă și pentru cel ce deține platforma el are o singură pagină cu care controlează toată platforma. O să vedem așa ceva în următorul capitol unde o să vedem cum interacționează utilizatorul cu platforma.

## Capitolul 3 Întreținere și utilizarea

### 3.1 Întreținerea aplicației

Întreținerea aplicației poate fi realizată în două moduri prezentate mai jos.

#### 3.1.1 Întreținerea cu ajutorul unui programator

Pentru a putea întreține această platformă o să avem nevoie de un programator care să adauge produse în baza de date și să modifice platforma în funcție de ceea ce dorește managerul platformei. În cadrul adăugării produselor programatorul are la dispoziție servărul cu datele despre produse și comenzi ca în figura de mai jos:

A id	A name	A description	A category	price
c5b76536a990841	Botine	Botine de culoare maro, inaltime	Incaltaminta	14.86
e03741e2902895a	Sandale	Sandale de culoare neagra, mar...	Incaltaminta	12.64
ed1a853a532878f	Sandale	Sandale de culoare neagra, lum...	Incaltaminta	16.47
18a599d4148d33	Sandale	Sandale de culoare neagra cu p...	Incaltaminta	14.98
18b3ecc07a59f4	Adidas cu platforma	Adidas de culoare neagra se in...	Incaltaminta	15.37
eb3e8ec4276a928	Adidas cu platforma	Adidas de culoare crem se inch...	Incaltaminta	12.76
22151060647b72	Ghete de lama	Ghete de lama de culoare neag...	Incaltaminta	17.53
9338b3a359aa8b	Adidas	Adidas de culoare alba, stil spo...	Incaltaminta	16.97
e13ca46042299d2	Fusta	Fusta de culoare neagra, poat...	Fusta	14.74

Fig.2 Serverul pentru baza de date



### 3.1.2 Întreținerea de către managerul platformei

Ca să apelăm la un programator de fiecare dată când dorim să adăugăm produse sau să verificăm anumite comenzi poate fi destul de costisitor dar să lăsăm managerul care cel mai probabil nu are experiența cu programarea să efectueze schimbări în baza de date poate fi riscant. Așadar am creat și o pagină pentru administrator unde se poate loga cu ajutorul unui username și o parolă. Scopul este acela de a putea adăuga/șterge/edita produse sau verifica comenzi fără a fi nevoie să se atingă managerul de baza de date. Astfel o să evităm ca managerul care dacă nu știe programare de cele mai multe ori să poate să strice baza de date și eliminăm și costul pentru care trebuie să plătim un programator pentru a face acest lucru. Tot ce rămân este să apelăm la un programator doar dacă dorim să schimbăm aspecte legate de modul de funcționare sau de dizainul aplicației web.

În figurile de mai jos este prezentată pagina la care are acces managerul pentru a efectua modificări platformei:

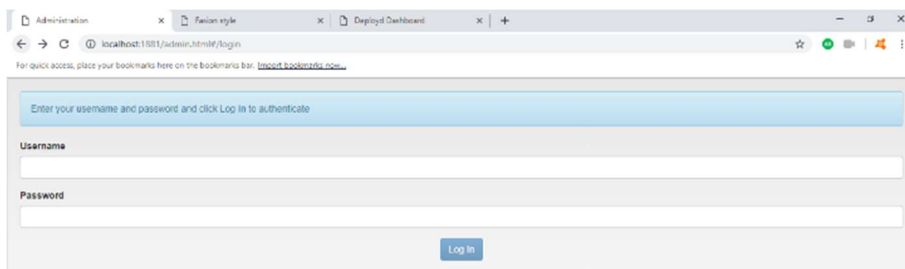


Fig.3 Pagina unde se logheaza managerul

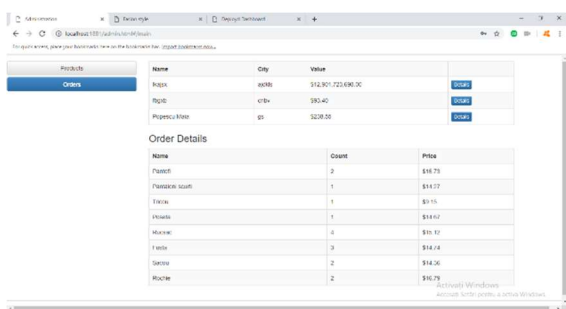


Fig.4 Pagina pentru vizualizarea

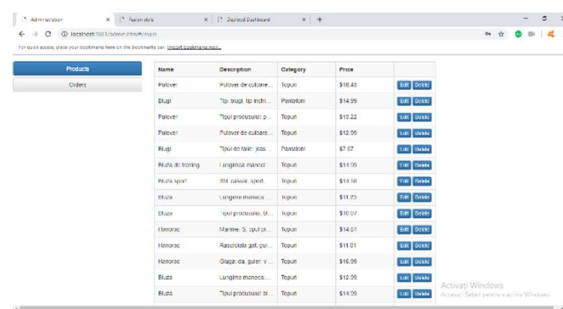


Fig.5 Pagina pentru editarea/ștergerea și comenzilor adăugarea produselor

### 3.2 Utilizarea din punctul de vedere al clientului

În momentul când utilizatorul intră pe aplicația noastră acesta are la dispoziție următoarea pagină( figura 6):

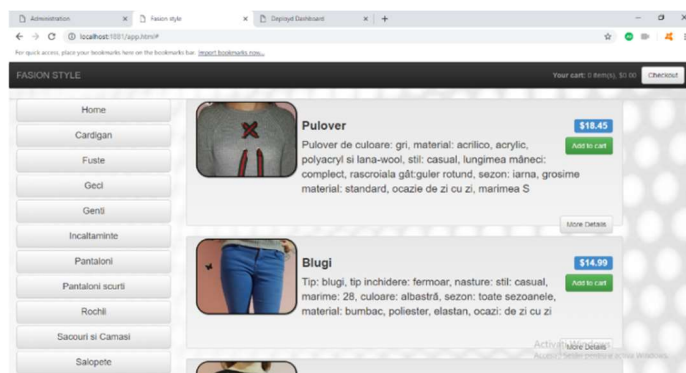


Fig.6 Prima pagină a aplicației web

Utilizatorul poate căuta produse de pe pagina „Home” sau poate selecta una din categorii pentru a fi afișate doar produsele dintr-o anumită categorie. Avem pentru fiecare produs numele acestuia, o poză cu produsul, prețul produsului, un buton pentru adăugare în coșul de cumpărături și un buton pentru mai multe detalii „More Details” . În cazul în care clientul dă click pe butonul pentru mai multe detalii se va încarță un nou view dar el rămâne pe aceeași pagină și va avea în față următoarea pagină(figura 7):

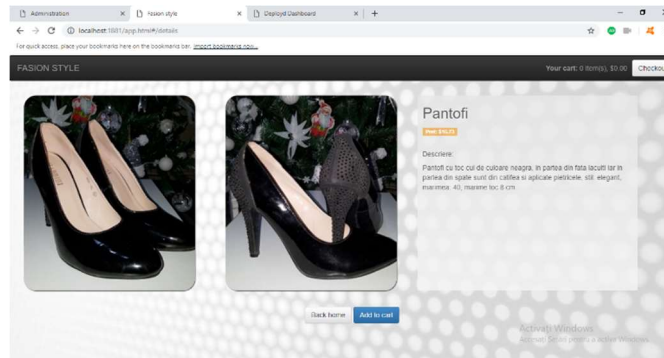


Fig.7 Pagina cu mai multe detalii

Acesta are două poze cu produsul și cu mai multe detalii legate de el. Pe pagină mai sunt observate două butoane unul pentru a se întoarce la view-ul original și unul pentru adăugarea produsului în coșul de cumpărături.

Să presupunem că ne-am plimbat puțin pe website ne-am adăugat produse în coșul de cumpărături și vrem să vedem tot ceea ce avem în coș eventual poate să renunțăm la anumite produse pe care nu le mai dorm pentru a face așa ceva apăsăm pe butonul „Checkout” din colțul din dreapta sus. Un nou view va fi încărcat și va arăta ca în figura 8:

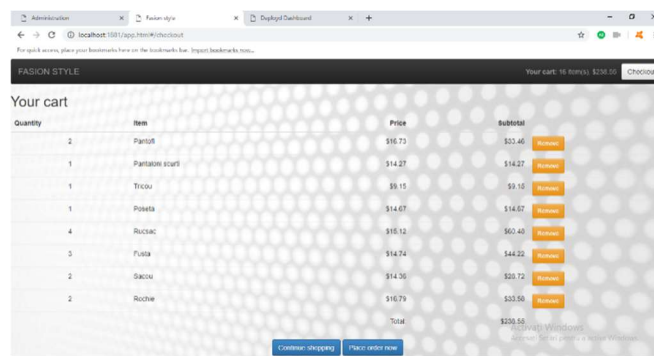


Fig.8 Interiorul coșului de cumpărături

Clientul are pe pagină cantitatea dintr-un anumite produs achiziționat( dacă cumpărăm două bluze de același fel) numele produsului și subtotalul( este rezultatul înmulțiri dintre prețul produsului si cantitatea achiziționată) alături de un buton „Remove” în caz că vrea să renunțe la un produs. Acesta mai are două butoane „Continue shopping” și „Place order now”. Dacă dă click pe „Continue Shopping” se va întoarce la primul view al aplicației(figura 6). Dacă dă click pe „Place order now” atunci va fi încărcat un nou view după cum se poate observa în figura 9:

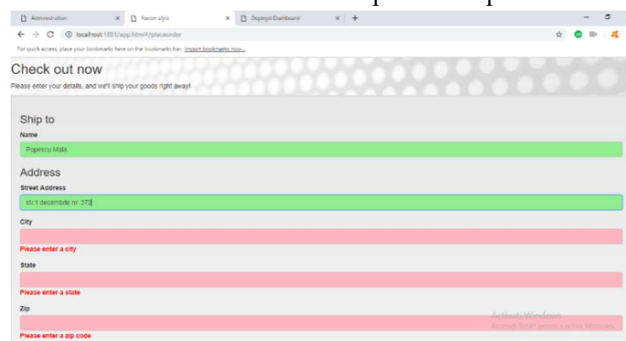


Fig.9 Completarea cu detalii despre locul unde să fie făcută livrarea

Câmpurile roșii semnifică faptul că trebuie să fie complete cu ceva de către client iar verde dacă a completat corect. Acesta mai are la dispoziție pe pagină un buton „Order”, când acesta va fi apăsat i se va arăta următorul view în care i se arată un mesaj cu „Thanks!” iar coșul de cumpărături va fi resetat la 0 întru-cât a achiziționat produsele.

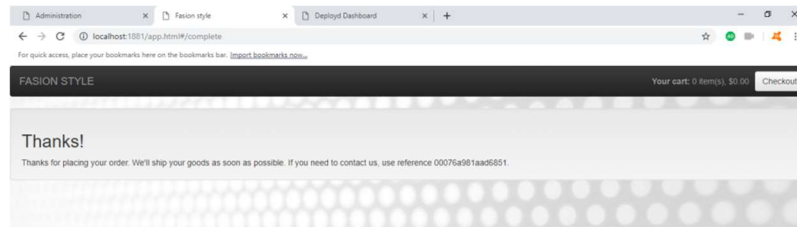


Fig.10 Finalizarea comenzii

#### Capitolul 4 Concluzii:

Prin realizarea acestei platforme am reușit să vedem ce este în spatele unei afaceri în mediul online și cât de ușor este să ne ocupăm de ea. În cadrul unui magazin mic această platformă poate fi administrată de un grup mic de persoane. Dacă pe lângă haine mai aveam și alte domenii cum ar fi: electronice, mobilieri, piese auto etc. atunci este bine să avem angajații care se ocupă de fiecare domeniu în parte. Avantajele unei platforme online sunt destul de mari și putem a face bani destul de rapid. Partea cu banii în general este cea mai importantă, degeaba ai o platformă și o întreții dacă nu aduce un profit. Un exemplu prin care această platformă poate face bani ar fi următorul: noi suntem managerul platformei, iar noi ne oferim să îi ajutăm pe anumiți furnizorii să vândă online produsele. Produsele rămân în magazin astfel noi nu le cumpărăm, dar dacă cineva le cumpără de pe pagina noastră web clientul va cumpăra produsul la prețul inițial la care mai adăugăm noi 5%-10% din valoarea lui + taxa de transport pana la client. Profitul nostru va fi acel 5%-10%. Deși noi nu am vândut nimic ci altcineva. Aceasta este o tehnică destul de des întâlnită în zilele noastre.

#### Bibliografie

1. Buschmann, Frank (1996) *Pattern-Oriented Software Architecture*
2. Freeman and Adam, Pro AngularJS
3. \*\*\*[https://www.tutorialspoint.com/angularjs/angularjs\\_mvc\\_architecture.htm](https://www.tutorialspoint.com/angularjs/angularjs_mvc_architecture.htm)

# SISTEM PENTRU CONTORLUL VOCAL A UNEI OGLINZI INTELIGENTE

**Autor: Dan-Dumitru MANU<sup>1</sup>**

[raiku.doku@yahoo.com](mailto:raiku.doku@yahoo.com)

**Coordonatorii: Conf.univ.dr.ing. EGRI Angela**

**Şef.lucr.dr.ing. SÎRB Vali**

<sup>1</sup>*Universitatea din Petroşani, Facultatea de Inginerie Mecanică şi Electrică, Specializarea: Calculatoare, anul 4*

## **Rezumat:**

In this paper you will find how to build a magic mirror/ smart mirror witch can understand what are you say. The smart mirror was develop first time for raspberrry pi but after a while people start to want more functions from the mirror so the one who come with the idea of a magic mirror start to build modules for it. It's nice to have a magic mirror but with every new module the sistem will use more resource and raspberrry pi boards are not so strong so you can't have all the modules that you want even with a last raspberrry pi that come out in special when we speak about voice recognition. I decide to build a smart mirror but this time the hardware sistem not to be a raspberrry pi but to be a normal PC. I want use a old computer because people don't want old PC form the lowest resources. If I take that low resources form an old pc and do a corporation with a raspberrry pi then the PC would win. I will build a smart mirror, this mirror will be more good that the original mirror because it will running on all operating systems (windows or linux), the smart mirror work only on raspberrry pi 2/3 nothing more now it will have a voice recognition and will be usefully for any old computer.

## **Cuvinte cheie:**

*Smart, mirror, voice control*

## **1. Introducere**

Ideea din spatele acestui proiect a venit atuncea când am dorit să instalez programul pentru o oglindă magică pe raspberrry pi –ul pe care îl deţin. Am rămas puţin surprins că dupa ce am reuşit să instalez programul nu am reuşit să instalez şi modulul de recunoaştere vocală motivul fiind: versiunea nouă de raspberrry pi pe care o deţin şi modulele iniţiale ne fiind updatate la zi. Dacă voiam să implementez şi comanda vocală trebuie să aştept un nou update fie să achiziţionez un modul separate de google recognizer pe care să îl leg la raspberrry pi-ul meu. Cei ce au construit acest proiect de oglindă inteligentă denumită mai târziu MagicMirror, doar au făcut open source munca lor însă nu au mai realizat updateuri de multă vreme drept urmare şi acuma dacă dorim să instalăm programul acestora pe un raspberrry pi o să primim diverse mesaje de eroare iar la final oglinda va funcţiona la un nivel mediu, basic ne având acces la funcţii mai complexe.

La început am zis să imi creez eu un modul de recunoaştere vocală doar ca să îl leg la modulul lor trebuie să am permisiune şi de la ceea ce au creat programul de oglindă. Programul nu acceptă module suplimentare şi să intru în el ca să adaug un modul nou + să mai creez modul în sine durează foarte mult. Soluţia a fost să creez eu un program de oglindă magică la care să îi creez un program de recunoaştere vocală. Cu această idee am făcut doi paşi importanţii: primul este acela că programul meu de oglindă nu depinde de sistemul de operare oferit de raspberrry pi( Debian linux) iar cel de al doilea este timpul de realizare efectiv al proiectului, acesta fiind foarte mic. Primul pas este uşor de înţeles în schimb al doilea se referă la faptul că în loc să caut printe mii de linii de cod şi să editez un cod pe care nu îl cunosc( codul este împărţit în diferite fişiere când ajung la o legătură nici nu termin un sfert din fişierul actual că trebuie să merg la alt fişier să îl înţeleg doar pentru o linie de cod din fişierul actual) este o muncă destul de grea şi necesită mult timp.

## **2. Modul de construire al oglinzii**

Oglinda este realizată în două părţi: hardware şi software.

### **2.1 Partea Hardware**

Partea hardware este alcătuită din: geam reflectorizant, un ecran şi un computer. Ecranul poate să fie orice, eu am ales să folosesc o plasma. Geamul reflectorizant poate fi achiziţionat de la un magazin care lucrează pe această temă(ex.: magazine de termopane). După ce suprapunem geamul de ecranul ales trebuie să conectăm computerul la ecran, pentru aşa ceva eu am folosit un cablu HDMI în ambele capete.

Geamul reflectorizant are două suprafeţe. Una reflectă 20% din lumină şi una reflectă 80% din lumină. Partea care reflectă 20% din lumină trebuie suprapusă peste ecran iar ceea ce reflectă 80% să fie partea exterioară.

### **2.2 Partea Software**

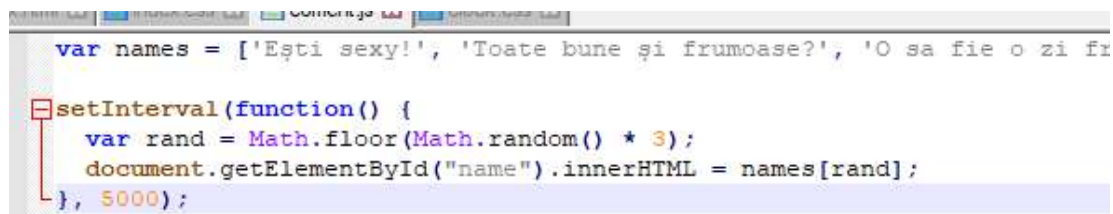
Programul original al oglinzii inteligente este alcătuit din html, css și java script. Exact asta o să folosesc și eu în creerea oglinzii, pot fi folosite și alte limbaje de programare dar acestea le cunosc eu mai bine când vine vorba de economisirea resurselor. Programul pe care o să îl realizez este de la 0 făcut. Tot ce am luat de la oglinda inițială este ideea de realizare. Ca exemplu oglinda este o mașină, eu știu că mașina are 4 roți, 4 uși, un motor, scaune etc. doar că eu nu știu cum sunt făcute acele piese component ale mașini sau cum sunt legate între ele, așa că o să realizez și eu scaunele mele, roțile mele și așa mai departe în funcție de cum știu eu să programez.

Primul pas pentru a face software-ul este să creem o pagină HTML și să facem zona de body a pagini HTML să fie neagră. Fundalul pagini trebuie să fie negru deoarece dacă în spatele geamului este întuneric acesta o să reflecte exact ca o oglindă reală ceea ce este în fața lui. Dacă în spate sunt alte culori/imagini noi o să le putem vedea. Pentru a realiza așa ceva recomand să faceți un fișier css și să puneți în el următorul cod: `body{ background-color:black; font-family:helvetica; color:white; align:center; }`. În acest cod specific culoarea zone body și tipul, culoarea și alinierea în pagină pentru text.

Acum că avem fundalul pot adăuga diverse detalii pe el. Oglinda originală vine instalată cu: ceas, dată, vremea orașului, mesaje pozitive și diverse știri.

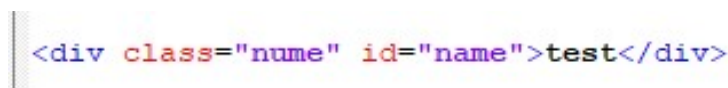
Eu am ales să adaug data, ora, vremea orașului Petroșani și mesaje ce o să fie afișate la un anumit interval de timp. O să am 3 elemente la final.

Pentru a afișa mesaje pozitive cum ar fi: "O să fie o zi frumoasă ca tine!" sau "O să fie bine" o să creez un fișier de tipul js în care o să pun următorul cod ca în figura nr.1. La iar în pagina principală o să adaug legătura către scriptul meu js și o diviziune în care o să specific ce dimensiuni să aibă textul cum să fie el( recomand să îl lăsați centrat). urmat de un cuvânt oare care ca în figura nr. 2. Acesta va fi schimbat la un interval de 5 secunde.



```
var names = ['Ești sexy!', 'Toate bune și frumoase?', 'O să fie o zi fr  
setInterval(function() {  
    var rand = Math.floor(Math.random() * 3);  
    document.getElementById("name").innerHTML = names[rand];  
}, 5000);
```

Fig. nr.1 Codul pentru afișarea mesajelor pozitive



```
<div class="nume" id="name">test</div>
```

Fig.nr.2 Codul ce trebuie pus în pagina HTML

Am specificat că o să am 3 elemente, pentru fiecare element am o diviziune de aceea în figura nr 2.o să vedeți `class="nume"` iar `id="nume"` este variabila inițială iar textul din interior va fi schimbată cu cel specificat în fișierul .js. Acolo apare nume și nu mesaje deoarece la început l-am pus să afișeze cele două prenume și numele pe care îl am eu. L-am lăsat așa ca să știu de la ce idee am început și anume cu modificarea unui șir de caractere.

Pentru a vedea data și ora curentă am realizat un cod în pagina HTML în zona de head, codul poate fi pus și într-un fișier separate dacă doriți dar eu am ales să îl las în pagină pentru moment deoarece îmi este mai la îndemână. codul este cel din figura nr.3. Acesta cod ia data și ora curentă( ziua, luna și anul). Acuma că avem datele trebuie să le afișăm. afișarea este ca în figura nr.4.

```

8     window.onload = setInterval(clock,1000);
9
10    function clock()
11    {
12        var d = new Date();
13
14        var date = d.getDate();
15
16        var month = d.getMonth();
17        var montharr=["Ianuarie","Februarie","Martie","Aprile","Mai","Iunie","Iulie","August",
18                    "Septembrie","Octombrie","Noiembrie","Decembrie"];
19        month=montharr[month];
20
21        var year = d.getFullYear();
22
23        var day = d.getDay();
24        var dayarr=["Duminică","Luni","Marți","Miercuri","Joi","Vineri","Sâmbătă"];
25        day=dayarr[day];
26
27        var hour =d.getHours();
28        var min = d.getMinutes();
29        var sec = d.getSeconds();
30
31        document.getElementById("date").innerHTML=date+" "+date+" "+month+" "+year;
32        document.getElementById("time").innerHTML=hour+" "+min+" "+sec;
33    }
34    </script>

```

Fig.nr.3 Codul pentru data și ora curentă

```

<div class="clock">
    <p id="date"></p>
    <p id="time"></p>
</div>

```

Fig. nr.4 Codul pentru pagina HTML

Acum am data, ora și mesaje pozitive ce pot fi afișate la un anumit interval de timp. Acuma trebuie să implementez vremea meteorologică de la orașul meu. Realizarea unui astfel de cod este extreme de complexă și durează mult timp și doar pentru o singură localitate drept urmare nu o să mai pierd timpul pentru a realiza un astfel de cod, puteți căuta pe internet un "weather" pentru pagina voastră web pe care să îl descărcați și să îl implementați. În cazul în care vreți să îl faceți de la 0, trebuie să luați un API de la un website care oferă vremea pentru orașul/orașele pe care le doriți și să realizați un program care să înțeleagă API-ul respectiv și să știe ce afușează. Sunt destule tutorial pe internet pentru așa ceva, tutorialele bune au undeva la 3-4 ore, partea de funcționalitate la care adăugăm și timpul pentru partea de grafică să arate bine în pagină. Eu am creat o diviziune și am luat codul de la [1], este gratis mai rămâne să realizați dizainul pe culorile alb-negru și să îl adăugați în pagină.

La final după ce deschideți pagina html, utilizați google chrom pentru a o deschide și apăsați F11 pe tastatură astfel pagina devine full screen. În figura nr.5 putem vedea oglinda realizată de mine iar în figura nr.6 cea disponibilă pe raspberry pi 2/3.

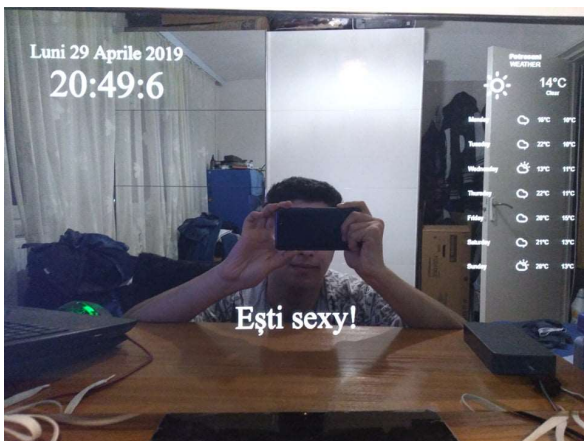


Fig nr.5 Oglinda realizată cu programul meu



Fig. nr. 6 Oglinda realizată cu programul de pe internet

Destul de asemănătoare cele două. Acum că avem oglinda gata este timpul să realizăm partea de comandă vocală.

### 3. Modul de construire al sistemului de recunoaștere vocală

În cadrul sistemului de recunoaștere vocală am ales să folosesc python 3 pentru al realiza. Python vine cu multe librării care reprezintă unele foarte utile în realizarea programului.

Programul o să îl împart în 4 părți:

- Recunoașterea vocală( speech to text);
- Răspunsul vocal( text to speech);
- Condiții pentru a decide un răspuns;
- Rularea programului „non-stop”.

Înainte de a începe scrierea codului propriu zis va trebui să instalăm dependențele programului nostru( bibliotecile pe care o să le folosim). Recomand instalarea de la început pentru a putea lucra liniștiți mai târziu, bibliotecile pe care o să le folosim sunt: speech\_recognition, ctimer, time, gtts și os. La aceste două biblioteci recomand să instalați și programul VLC pe calculator și să îl setați ca fiind programul implicit pentru deschiderea fișierelor audio, aceasta în cadrul la windows. Dacă sunteți pe linux nu mai folosiți os, folosiți playsound, programul rămâne același și nu este nevoie să mai instalați vlc player, aici [2] este linkul de unde puteți descărca vlc player. În mod normal python când este instalat vine deja cu time și os, în caz că nu vine instalațiile cu pip. Restul de ctimer și speech\_recognition se instalează tot cu pip în command prompt, dacă întâmpinați probleme sau nu știți cum se instalează doar căutați pe google ” python install” și numele de la bibliotecile pe care vi le-am lăsat mai sus. Este foarte ușor și puteți vedea și exemple cu ele. De aici [3] puteți descărca și python3.

După ce ați instalat bibliotecile de care depinde programul puteți merge mai departe. **NU RECOMAND SĂ FACEȚI CODUL DE MAI JOS DACĂ NU AȚI INSTALAT BIBLIOTECILE!!!!**

#### 3.1 Recunoașterea vocală

Pentru a face recunoașterea vocală o să folosesc speech\_recognizer, una din bibliotecile menționate mai sus. O să scriu o definiție în care am o variabilă sr, variabila sr o să apară în prima linie din programul în python așa: import speech\_recognition as sr. Sr reprezintă speech\_recognizerul nostru. O să pun aici în figura nr.7 și o să îl explic.

```
def recordaudio():
    r = sr.Recognizer()
    with sr.Microphone() as source:
        print ('Attention! Say something')
        audio = r.listen(source)
    try:
        data='something'
        data = r.recognize_google(audio,language='ro-RO')
        print('You said: ' + data + '\n')
    except sr.UnknownValueError:
        print ('Attention ! Google could not understand audio')
        data='Could not understand'
    except sr.RequestError as e:
        print ('Attention ! Could not request results from Google service.')
    return data
```

Fig. nr. 7 Codul pentru recunoașterea audio

În acest cod vedem variabila noastră sr care folosește microfonul ca sursă de date, afișează ceea ce înregistrează după care variabila audio ascultă sursa noastră. Variabila data este o variabilă care o să o folosim mai târziu, pentru început aceasta o să primească un string oarecare. Variabila noastră data o să primească valoarea r.recognize\_google(audio, language=ro-RO) adică merge la google\_recognizer îi dă variabila audio realizată cu două linii de cod mai sus și îi zice că în acea variabilă audio există o voce care zice cuvinte românești și să îi dea din acel fișier textul înregistrat. În continuare funcția noastră va afișa mesajul: You said: ” ce am zis noi”. Programul nostru va recunoaște limba română. Restul de mai jos sunt erori ce pot apărea, lăsați acele erori acolo exact cu textul care este în el. În caz că programul vă dă eroare dar nu știți ce eroare este sau va da eroare care a fost propagată să știți exact care este problema. Acele două erori sunt singurele care pot apărea dacă o întâmpinați pe a doua căutați rezultate pe google pentru ea în funcție de sistemul de operare pe care lați ales linux/ windows.

#### 3.2 Răspunsul vocal

Dupa ce am realizat la 3.1 transformarea vocii în text acum să face și invers text to voice. Pentru așa ceva am reat o nouă definiție unde se primește o variabilă de tipul string. Am folosit gtts( am specificat mai sus că trebuie instalată această bibliotecă) pentru a transforma textul în voce cu comanda tts = gTTS(text=textulMeu, lang='ro'), această comandă o să o pun într-o variabilă denumită tts pentru ca după ce am sunetul să pot să îl salvez cu comanda:

tts.save("audio1.mp3"). Destul de simplu, aici mai adăugați o linie de cod și anume: os.system("audio1.mp3"), această linie de cod spune calculatorului să folosească media playerul implicit( VLC la noi, am zis mai sus să îl instalați, puteți să îl instalați și acum linkul este la [2]) ca să deschidă fișierul nostru audio salvat mai cu linia de cod de dinainte. Definiția mea aici se numește "speak".

### 3.3 Condiții pentru a decide un răspuns

Această parte este destul de simplă creți o nouă definiție care primește ca parametru "data", variabila noastră de mai sus. Acuma doar realizăm condiții cum ar fi cea din figura nr. 8.

```
if "salut" in data:  
    speak("Buna ziua")
```

Fig. nr. 8 Exemplu de condiție

Dacă variabila data creată la 3.1 conține doar "salut" și nimic mai mult, atunci se va apela definiția creată la 3.2 și va spune "Bună ziua". Destul de simplu acum doar trebuie să adăugăm cât de multe condiții dorim iar programul nostru o să ne răspundă. Aici puteți să faceți și întrebări mai complexe ca și exemplu: există mailuri necitite și programul să zică dacă da sau nu și câte sunt necitite sau să creați o agendă în care să înregistrați programul pe o zi și programul să vă răspundă. Aici aveți varianta "basic" dar acesta este punctul de plecare către proiectul nostru final.

### 3.4 Rularea programului „non-stop”

Această parte este cea mai ușoară la finalul programului scrieți: while True: data=recordaudio() "numele de la definiția creată la 3.3"(data).

Acesta este programul de recunoaștere vocală. Acum trebuie doar să deschidem aplicația creată la punctul 2, deschideți și programul creat în python și dați run după care dați click pe pagina creată și apăsați F11. Acuma aveți o oglindă magică care poate recunoaște cuvinte în limba română.

## 4. Bibliografie

- [1] \*\*\*<https://weatherwidget.io/>
- [2] \*\*\*<https://www.videolan.org/vlc/index.ro.html>
- [3] \*\*\*<https://www.python.org/downloads/>



# APLICAȚIE PENTRU GESTIONAREA SENZORILOR

**Autor: Hajdu <sup>1</sup>, Prenume Iosif-Emanuel <sup>2</sup>**

[ioji\\_h\\_2011@yahoo.com](mailto:ioji_h_2011@yahoo.com)

**Coordonator: șef lucrări drg. ing. Badea Alin <sup>3</sup>**

<sup>1</sup> *Universitatea Petroșani, Facultatea de Inginerie Mecanică și Electrică, specializarea: Calculatoare, anul IV*

## Rezumat

Lucrarea este împărțită în 5 părți, *Introducere, Componente software folosite, Componente hardware folosite, Descrierea aplicației și Bibliografie*. Introducerea descrie un scurt istoric despre necesitatea de comunicare a oamenilor și despre apariția Internetului Lucrurilor. A doua parte conține toate componentele și tehnologiile software folosite pentru realizarea aplicației, cu o scurtă descriere pentru fiecare, similar în partea a treia unde sunt descrise componentele hardware utilizate. În *Descrierea aplicației* se explică cum toate tehnologiile menționate în părțile anterioare sunt folosite pentru realizarea aplicației. Ultima parte conține toate referințele bibliografice utilizate în lucrare.

## Cuvinte cheie

*IoT, API, REST, IoT Hub*

### 1. Introducere

Comunicarea de la distanță a reprezentat un lucru foarte important pentru omenire, de la semnalele cu fum, porumbeii călători la telefonie și la televiziune, abilitatea de a transmite mesaje a devenit una esențială pentru orice societate modernă, apariția internetului a reprezentat un pas revoluționar. Internetul a permis transmiterea de date de la un capăt la altul al lumii într-un timp foarte scurt, astfel nu este de mirare că mare parte din dispozitivele moderne se pot conecta la internet. Pe lângă laptop-uri, tablete și smartphone-uri, care sunt în mare parte folosite pentru comunicarea între oameni, au apărut dispozitive electronice care pot fi controlate de la distanță, fără intervenția omului, rețeaua pe care o alcătuiesc poartă numele de IoT(Internet of Things) sau Internetul Lucrurilor.

Conform CIBSG(Cisco Internet Business Solutions Group), IoT a prins viață între anii 2008-2009 când numărul de dispozitive sau obiecte care se pot conecta la internet a depășit numărul de oameni. S-a aproximat că pentru numărul de oameni de pe planetă, care prin 2010 era de 6.8 miliarde, ar fi 12.5 miliarde de dispozitive conectate la internet. În 2008 Consiliul Național de Informații al SUA a declarat IoT ca una din cele 6 tehnologii disruptive. În 2011 a fost lansat în mod oficial IPV6, care spre deosebire de IPV4 care avea 32 de biți aceste era compus din 128 de biți și permite un număr de 2128 de adrese.<sup>[1]</sup>

Principala utilitate al IoT este facilitarea automatizării proceselor și permiterea controlului la distanță ale dispozitivelor, fără intervenția omului. Acesta presupune existența unor aplicații software care să poată să recepțeze date de la senzori să le stocheze într-o bază de date de unde pot fi transmise către o interfață grafică și să poată transmite comenzi către senzori, pentru toate acestea este necesară utilizarea unui API(Application Programming Interface).

API-ul reprezintă un set de subrutine, protocoale de comunicații și unelte pentru construirea de programe software. În esență acesta este un intermediar între comunicațiile între mai multe componente care pot fi realizate pe în cadrul unui sistem de operare, în cadrul unui sistem de baze de date, sau prin Internet, cel din urmă fiind cel mai relevant pentru IoT.

### 2. Componente software folosite

ASP.NET Web API 2 este un tip de proiect al platformei ASP.NET, care furnizează aproape toată funcționalitatea necesară pentru a adera la constrângerile arhitecturii REST(Representational State Transfer). Acest lucru se datorează în mare parte proprietății de routing a URL-urilor(Uniform Resource Locator) dată de către platformă.

MSSQL(Microsoft Structured Query Language Server) este un sistem de gestionare de baze de date relaționale. MSSQL folosește o variantă SQL numită T-SQL(Transact-SQL) care adaugă o sintaxă adițională pentru proceduri stocate și pentru tranzacții. Totuși crearea și administrarea bazelor de date reprezintă o grijă în plus pentru dezvoltatori pe lângă familiarizarea cu SQL, ei trebuie să petreacă mult timp gândindu-se despre relațiile dintre tabele, numele și parametrii procedurilor, schematică bazelor de date etc.

Entity Framework a apărut pentru prima oară în Iunie 2008 ca parte a Visual Studio 2008 Service Pack 1 și .NET 3.5 Service Pack 1. ADO.NET Entity Framework a devenit o platformă de bază pentru aplicații scrise în .NET. Principalul avantaj al lui Entity Framework este faptul că eliberează dezvoltatorii de grija administrării și structurării bazelor de date. Tot accesul la date și stocarea se realizează pe baza unui model conceptual care reflectă clase de obiecte create în cod. Entity Framework transformă obiectele create în cod, spre exemplu C#, în tabele și coloane, astfel preluarea, adăugarea, modificarea și ștergerea rândurilor și coloanelor din baza de date se va face prin accesarea acestor

clase de obiecte, denumite entități.<sup>[2]</sup>

Azure IoT Hub este un serviciu care rulează într-un cloud care are rolul de punct central pentru comunicația bidirecțională între o aplicație IoT și dispozitivele pe care le administrează. Serviciul permite mai multe tipuri de transmitere a mesajelor cum ar fi device-to-cloud, file upload și request-reply. IoT Hub permite întreținerea stării soluțiilor sale prin monitorizarea evenimentelor de tipul: crearea unui dispozitiv, eșuare unui dispozitiv și conectare unui dispozitiv.<sup>[3]</sup>

Funcțiile Azure sau AF(Azure Functions) sunt servicii care funcționează pe baza conceptului de serverless, adică permit hosting-ul și rularea de programe fără a mai fi necesară crearea și administrarea unei infrastructuri. Caracteristicile lor de bază sunt: suport pentru limbajele C#, F# sau JavaScript, costurile sunt calculate doar pentru momentele în care codul rulează, securitate integrată, integrare simplificată, dezvoltarea flexibilă, open-source.

VS(Visual Studio Community 2019) este un IDE(Integrated Development Environment) gratuit dezvoltat de Microsoft pentru crearea aplicațiilor pentru Windows, Android, iOS, aplicații web și care rulează pe cloud.

Postman este o aplicație care permite transmiterea de cereri HTTP, perfectă pentru testarea API-urilor de tip REST.

SSMS(SQL Server Management Studio) este un IDE pentru administrarea bazelor de date SQL, poate fi folosit pentru accesarea, configurarea și dezvoltarea tuturor componentelor de pe SQL Server, Azure SQL Database și SQL Data Warehouse. SSMS furnizează o singură utilitate, ușor de înțeles, care combină o arie vastă de ustensile grafice și un număr mare de editoare de script pentru a permite accesul dezvoltatorilor și administratorilor de orice nivel de cunoștințe.<sup>[4]</sup>

Azure Cosmos DB Emulator furnizează un mediu local care emulează serviciul ACDB(Azure Cosmos DB) pentru a facilita dezvoltare. Utilizarea emulatorului permite testarea aplicațiilor local fără a mai fi necesară o abonare la un serviciu real.<sup>[5]</sup>

Arduino este un IDE open-source care facilitează încărcarea programelor pe diverse plăci electronice. Mediul este scris în Java deci poate rula pe Windows, MAC OS X și Linux.

### 3. Componente hardware folosite

ESP8266 NodeMcu V3 caracteristici:

- hardware asemănător plăcilor Arduino;
- poate fi programat folosind IDE-ul Arduino;
- integrează GPIO(General-Purpose Input/Output), PWM(Pulse-Width Modulation), IIC(Integrated Inductive Components), 1-Wire și ADC(Analog-Digital Converter) într-o singură placă
- 4Mb memorie flash
- antenă încorporată

Diagrama microcontroller-ului:

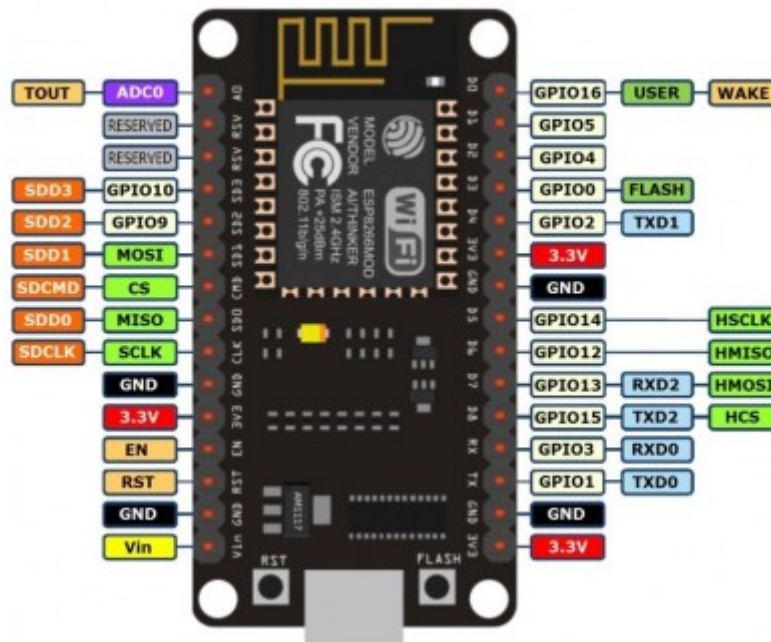


Fig. 1. ESP8266 NodeMcu V3

## Descrierea aplicației

Scopul aplicației este recepționarea datelor de la senzori și expunerea lor către clienți REST. Componentele aplicației sunt: Web API-ul REST, funcțiile Azure, platforma IoT Hub și două plăci NodeMcu V3.

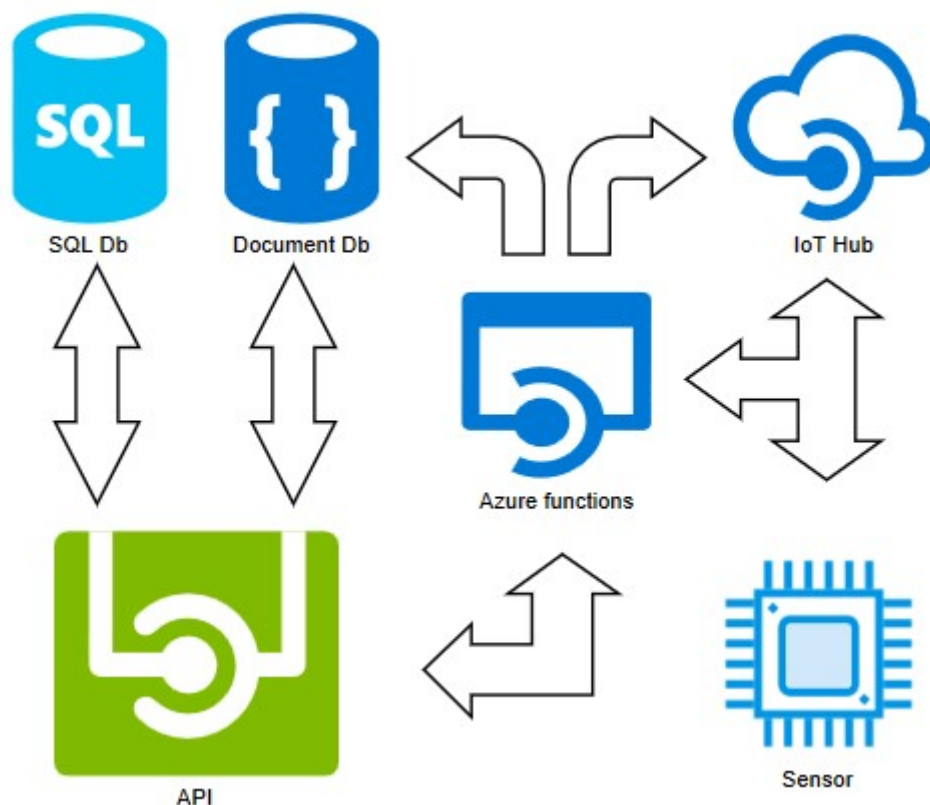


Fig. 2. Imagine de ansamblu

Web API reprezintă nivelul cel mai înalt al aplicației. Acesta recepționează datele transmise de la senzori prin intermediul IoT Hub și le face disponibile diverselor aplicații client. Server-ul a fost creat folosind IDE-ul VS 2019 și platforma ASP.NET, limbajul principal de programare este C#. Soluția conține trei proiecte: *SensorsManager.DataLayer*, *Sensor.DomainClasses* și *SensorsManager.Web.Api*, primele două fiind de tip bibliotecă .NET iar cel din urmă ASP.NET Web API 2.

*SensorsManager.DataLayer* este locul unde se configurează Entity Framework, mai exact în clasa *DataContext*, care moștenește din *DbContext*, se menționează toate tabelele care vor fi create în baza de date. Tot în acest proiect se mai găsește un folder intitulat *Migrations*, în care vor fi stocate toate modificările aduse modelului pentru a putea fi aplicate asupra bazei de date.

În al doilea proiect tip bibliotecă se află entitățile după care se vor construi tabelele din baza de date, acestea sunt:

*Measurements* - reprezintă unitățile de măsură ale senzorilor;

*SensorType* - reprezintă tipurile de senzori, printre proprietățile sale numim *MeasureId* care este o cheie străină și *Measurement* care este o referință prin care se stabilește o relație de unu la mai mulți între cele două tabele, cu tabelul de tipuri fiind tabelul copil;

*Users* - reprezintă tabelul în care vor fi stocate date despre utilizator, pe baza acestor date se va realiza și autentificarea;

*Sensors* - reprezintă tabelul de senzori, are o adresă generată automat în momentul creării și două referințe la tipuri și la utilizatori;

*SensorReadings* - reprezintă citirile transmise de senzori, spre deosebire de celelalte clase aceasta reprezintă entitățile din baza de date Cosmos DB;

În urma executării programului baza de date creată de Entity Framework va arăta așa:

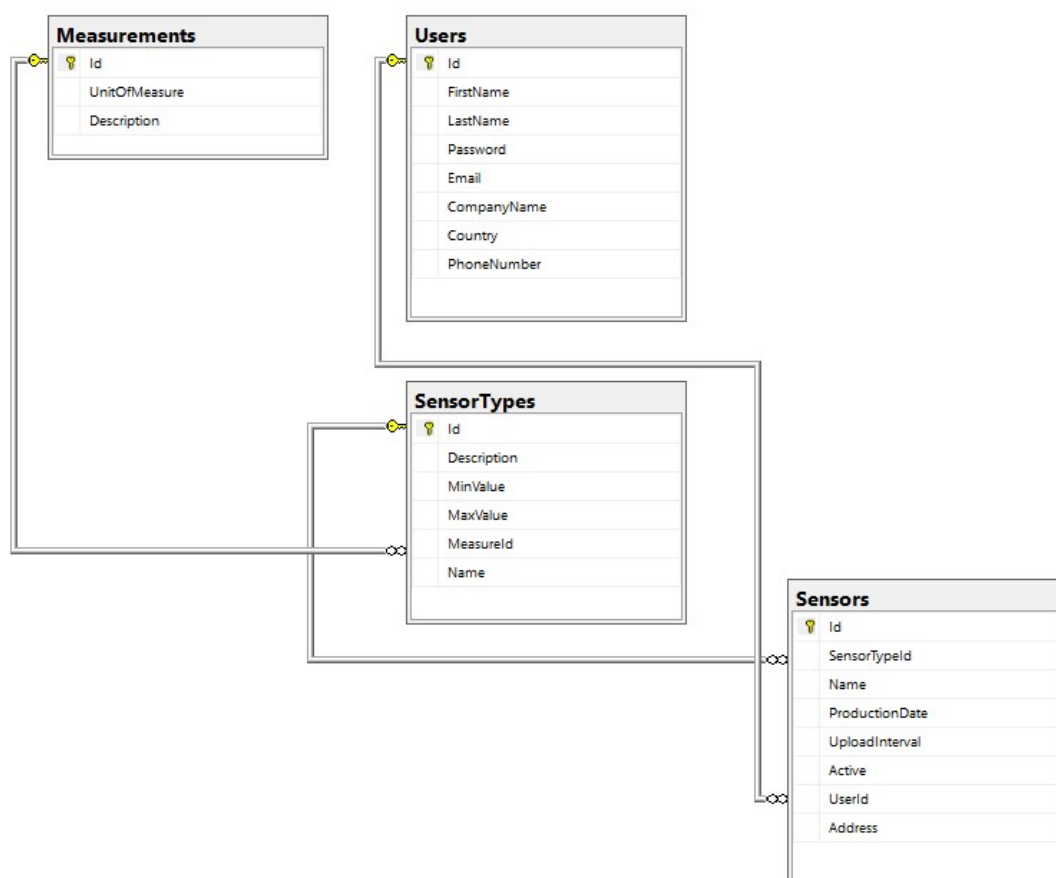


Fig. 3. Baza de date SQL

*SensorsManager.Web.Api* reprezintă server-ul în sine, el are referințe la celelalte două proiecte. Principalele sale elemente sunt *App\_Start*, *Repository*, *Models*, *IoT Hub*, *Controllers*, *Global.asax* și *Web.config*.

*App\_Start* este folderul în care se găsește tot codul care trebuie executat în momentul pornirii aplicației. Aici se au fost create trei clase: *WebApiConfig* unde se configurează rutele, se selectează metoda de serializare și se pornește CORS(Cross Origin Resource Sharing). În *ActivityCheck* se monitorizează activitatea senzorilor prin intermediul unui background worker care selectează data ultimei citiri și verifică dacă a depășit un anumit interval, specific fiecărui senzor. Ultima clasă din acest folder este *UnityConfig*, care reprezintă un IoCC(Inversion of Control Container) care este esențial pentru injectarea dependențelor într-un controller.

În *Repository* se realizează comunicarea directă cu baza de date prin intermediul clasei *DbContext*, din primul proiect. Pentru fiecare tabel în parte este o interfață tip repository și o clasă derivată cu același nume, care conține metode pentru adăugare, returnare, modificare și ștergere. Clasa *SensorReadingRepository* comunică cu baza de date *CosmosDb* prin intermediul unui obiect *DocumentClient*, în acest repository se află două metode, una de preluarea a citirilor și una de ștergere a citirilor, ultima fiind o metodă asincronă.

Folderul *Models* conține modele pentru fiecare entitate și două clase *ModelFactory* și *ModelMapper* care transferă date de la și respectiv către entități.

După cum îi spune denumirea, în folderul *IoT Hub* se găsește codul care realizează comunicarea cu IoT Hub, mai exact va realiza comunicarea cu funcțiile Azure, care au rol de intermediar între cele două elemente. Folderul conține o clasă în care se găsesc metode de transmitere a cererilor http (*HttpRequestFactory*) și o clasă *HubCommunicationManager* care conține metodele care apelează fiecare funcție în parte pe baza unor url-uri.

Nivelul server-ului care este expus aplicațiilor externe se găsește în folderul denumit *Controllers*. Structura acestuia este asta: *BaseApiController*, este clasa de bază din care derivă celelalte controlere, aceasta moștenește din clasa *ApiController* și conține proprietățile comune ale tuturor claselor derivate, cum ar fi clasele de transfer și suprascrieri pentru diferite funcții *IHttpActionResult*. La fel ca și la folderul *Repository*, se găsește câte un controller pentru fiecare model în parte, cu metodele respective fiecărei clase repository. În fiecare controller sunt validate datele primite din exterior, înainte de a fi transferate către baza de date.

*Global.asax* este locul în care se apelează toate funcțiile din *App\_Start*.

*Web.config* este un fișier XML(Extensible Markup Language) unde sunt menționate legăturile către cele două baze de date, dependențele aplicației și alte specificații.

Funcțiile Azure reprezintă elementul intermediar al aplicației, ele fac legătura dintre API, Cosmos DB și senzori. Numărul lor este patru în total: *CreateDeviceIdentity*, *DeleteDeviceIdentity*, *ReadingsReceiver*, *UpdateUploadInterval*.

Prima funcție este apelată din controller-ul *Sensors* al API-ului în momentul când se dorește adăugarea unui nou senzor, server-ul va trimite o cerere post către această funcție prin care îi transmite o adresa senzorului, generată automat la nivelul API-ului, care va determina identitatea dispozitivului. O dată adăugat un senzor pe server-ul IoT Hub, acesta îi va genera o legătură prin care dispozitivul fizic se va conecta la hub.

Cea de a doua funcție este apelată tot din controller-ul *Sensors*, aceasta va șterge identitatea selectată a senzorului, ca și consecință acesta nu va mai putea să se conecteze la hub.

*ReadingsReceiver* are rolul foarte important de a monitoriza toate citirile transmise de senzori către IoT Hub, apoi le stochează în baza de date Cosmos și trimite o cerere http către API pentru a-l anunța că senzorul a trimis citiri.

Folosind proprietatea de device twins, ultima funcție modifică intervalul la care transmite fiecare senzor.

Cele două plăci NodeMCU reprezintă nivelul cel mai de jos al aplicației, ele sunt programate în mod similar folosind IDE-ul Arduino, care folosește o variație a limbajului C++. Programul este salvat într-un fișier cu extensia .ino. Structura unui astfel de fișier este simplă deoarece conține doar două funcții una denumită *setup*, în care se realizează configurările inițiale (spre exemplu rata baud) și una denumită *loop* care este o buclă infinită. Înaintea funcției *setup* se includ bibliotecile necesare și se declară toate variabilele care vor fi folosite.

Primul lucru care trebuie setat la începutul oricărui program este rata baud, aceasta diferă de la circuit la circuit. În cazul plăcii NodeMCU, rata baud recomandată este de 9600. Al doilea lucru care trebuie realizat este conectarea plăcii la internet, pentru acest lucru este necesară numele și parola rețelei, care au fost stocate în două variabile statice *ssid* și *pass*. Placa va încerca să se conecteze la rețea folosind cele două variabile prin intermediul funcției *WiFi.begin*, cât timp conexiunea eșuează, programul va afișa adresa mac a circuitului împreună cu un mesaj de avertizare. În schimb, dacă conexiunea se realizează cu succes atunci se va afișa mesajul *Connected to* plus numele rețelei.

Următorul pas este configurarea timpului, pentru aceasta este folosită funcția *configTime*, care are ca parametrii zona temporală, dacă se folosește ora de iarnă sau de vară și două url-uri *pool.ntp.org* și *ime.nist.gov*, de unde vor fi luate informații necesare pentru determinarea timpului.

Conectarea la IoT Hub se va face folosind o instanță a clasei *IOTHUB\_CLIENT\_LL\_HANDLE*, denumită în acest context ca *IoTHubClientHandle*, care necesită legătura către imaginea virtuală a dispozitivului și specificații ale protocolului MQTT(Message Queing Telemetry Transport). Următorii pași pentru conectarea la IoT Hub sunt realizați prin apelarea a 4 funcții, fiecare având ca parametru obiectul instanțiat mai sus:

1. *IoTHubClient\_LL\_SetOption* - setează opțiuni adiționale ca trimiterea mesajelor în pachete, timpul permis de inactivitate până la încheierea sesiunii
2. și informații despre dispozitiv;
3. *IoTHubClient\_LL\_SetMessageCallback* - folosită pentru primirea mesajelor de la hub, are ca al doilea parametru funcția *receiveMessageCallback* al cărui scop este crearea unui buffer pentru recepționarea mesajelor de la server;
4. *IoTHubClient\_LL\_SetDeviceMethodCallback* - scopul ei este de a permite declanșarea anumitor metode cum ar fi activarea dispozitivului sau dezactivarea transmisiei de date, care vor fi specificate în funcția *deviceMethodCallback*;
5. *IoTHubClient\_LL\_SetDeviceTwinCallback* - folosind funcția *twinCallback* poate fi modificată valoarea intervalului de transmitere a datelor.

Funcția *loop* conține tot codul care trebuie executat în mod continuu până la dezactivarea dispozitivului. Aici se găsește o metodă adițională pentru activarea/dezactivarea transmisiei de mesaje, care folosește monitorul serial. Dacă utilizatorul introduce valoarea 1 atunci dispozitivul va începe să transmită citiri, iar dacă valoarea este 0 atunci dispozitivul va înceta activitatea. Tot în funcția *loop* se găsește vectorul *messagePayload* de tip char, a cărui lungime este determinată de către variabila *MESSAGE\_MAX\_LEN*. Vectorul este trecut mai întâi prin funcția *readMessage* pentru a fi serializat în formatul JSON(JavaScript Object Notation) și pentru a i se defini exact ce trimite: adresa senzorului, data citirii și o valoare generată în mod aleatoriu pentru a simula citirea de valori. Obiectul JSON trebuie să fie identic cu entitatea *SensorReadings* de la nivelul API-ului. După finalizarea serializării buffer-ul va fi introdus ca parametru, alături de *IoTHubClientHandle* în funcția *sendMessage*.

Primul element din funcția *sendMessage* este un obiect tip *IOTHUB\_MESSAGE\_HANDLE* denumit *messageHandle*, acesta reprezintă mesajul care va fi transmis către IoT Hub și este creat prin funcția *IoTHubMessage\_CreateFromByteArray* care are ca parametrii buffer-ul și lungimea lui. Dacă mesajul este creat cu succes atunci se vor mapa proprietățile mesajului în obiectul properties de tip *MAP\_HANDLE* folosind metoda *IoTHubMessage\_Properties* cu parametrul *messageHandle*. Procedura *IoTHubClient\_LL\_SendEventAsync* este responsabilă cu trimiterea asincronă a mesajelor, ea primește ca parametrii *IoTHubClientHandle*, *messageHandle* și *sendCallback*. Apelul acestei funcții este realizat într-un if în care rezultatul este comparat cu *IOTHUB\_CLIENT\_OK*, dacă egalitatea este adevărată atunci înseamnă că mesajul poate fi trimis, lucru semnalat de setarea la adevărat al variabilei boolene *messagePending*. Fie că mesajul a fost sau nu trimis cu succes, se va apela funcția *IoTHubMessage\_Destroy* pentru eliberarea resurselor alocate.

Ultima funcție de comunicare cu IoT Hub din *loop* este *IoTHubClient\_LL\_DoWork* aceasta este apelată atunci când se pot realiza primiri sau trimiteri de mesaje de către clientul *IoTHub*. Ea este responsabilă de toate interacțiunile

de legate de traficul de rețea ale dispozitivului.

#### 4. Concluzii

În concluzie folosind tehnologiile ASP.NET Web API, funcțiile Azure, IoT Hub și cele două baza de date (SQL și non-SQL) se poate realiza o aplicație care se folosește de viteza și siguranța oferită de protocolul MQTT pentru transmiterea de mesaje și pentru recepționarea de comenzi și instrucțiuni, de ușurința de implementare a arhitecturii REST pentru accesarea datelor de către utilizator, de structura arborescentă bine definită a bazei de date SQL utilă pentru definirea relațiilor complexe între senzori și utilizatori și de viteza și scalabilitatea foarte mare a bazei de date non-SQL, aceasta fiind compusă din obiecte JSON, perfectă pentru date telemetrice.

#### Bibliografie:

1. <https://www.postscapes.com/internet-of-things-history/>
2. Julie Lerman, (2010), *Programing Entity Framework*, editura O'REILY,
3. <https://docs.microsoft.com/en-us/azure/iot-hub/about-iot-hub>
4. <https://docs.microsoft.com/en-us/sql/ssms/sql-server-management-studio-ssms?view=sql-server-2017>

# SISTEM DE COMUNICARE WIRELESS PRIN SPAȚIUL DE LUMINĂ VIZIBILĂ

**Autori:** Iosif-Iulian MUREȘAN<sup>1</sup>, Henssel STEPANEK<sup>1</sup>, Alexandru HALIȚ<sup>1</sup>

[Iulianmuresan2014@gmail.com](mailto:Iulianmuresan2014@gmail.com); [hensselstepanek@yahoo.com](mailto:hensselstepanek@yahoo.com), [alexandruhalit@icloud.com](mailto:alexandruhalit@icloud.com)

**Coordonator:** Șef lucr.dr.ing. Simona RÎUREAN<sup>2</sup>

*1* Universitatea din Petroșani, Facultatea IME, specializarea Automatică și informatică aplicată, anul II

*2* Universitatea din Petroșani, Facultatea IME, Departamentul ACIEE

## Rezumat:

Tehnologiile de comunicații fără fir au proliferat și au devenit esențiale foarte repede în ultimele decenii ale secolului XX și începutul secolului XXI. Odată cu creșterea comunicațiilor de date fără fir, cererea pentru spectrul de frecvențe radio (RF) depășește oferta, determinând companiile să ia în considerare opțiunile de utilizare a unor părți ale spectrului electromagnetic, altele decât cele de radio frecvență. Această lucrare prezintă accesibilitatea și disponibilitatea unui sistem simplu LiFi de utilizat în viața de zi cu zi, realizat din echipamente cu costuri reduse, la îndemâna studenților. De asemenea, în lucrare sunt prezentați diferiți parametri, cum ar fi distanța, vizibilitatea și unghiul care afectează comunicarea LiFi cu ajutorul unor panouri solare. Utilizarea celulelor solare permit sistemului să recolteze energia împreună cu semnalul de recepție simultan. De asemenea, folosirea acestei tehnologii în viitor, va fi necesară și utilă.

**Cuvinte cheie** LiFi, RF, panou solar, comunicații wireless.

## 1. Introducere

Pentru că este în natura umană să fim curioși și aspirăm spre mai mult și mai bine, în ultimele două decenii, sistemele de comunicații wireless s-au răspândit în mod neașteptat de rapid datorită creșterii continue atât al numărului dar și a diverselor tipuri de dispozitive inteligente.

Dezvoltarea pe scară largă a tehnologiilor de comunicare în spectrul de undă de radiofrecvență a reprezentat un factor cheie în extinderea dispozitivelor și sistemelor wireless. Cu toate acestea, partea spectrului electromagnetic utilizat de sistemele fără fir este limitată în ceea ce privește capacitatea, iar licențele de utilizarea unor părți din spectru sunt deja atribuite dar și costisitoare.

Comunicațiile wireless optice (OWC) reprezintă o formă de comunicare optică în care lumina vizibilă, infraroșu (IR) sau lumină ultravioletă (UV) neagră este utilizată pentru a transporta un semnal.

Sistemele OWC care funcționează în banda lumină vizibilă (390-750 nm) sunt denumite în mod obișnuit comunicare prin spectru de lumină vizibilă (VLC).

Sistemele VLC folosesc diodele care emit lumină (LED-uri), care pot lucra (inchis/deschis) la frecvențe foarte mari fără efect vizibil asupra ochiului uman. VLC pot fi utilizate într-o gamă largă de aplicații, inclusiv rețele wireless locale, rețele wireless personale și rețele de autovehicule, printre altele.

Pe de altă parte, sistemele OWC terestre de tip punct-la-punct, cunoscute și ca sisteme optice libere (FSO), funcționează la frecvențele apropiate de IR (70-1600 nm). Aceste sisteme utilizează în mod obișnuit emițătoare laser și oferă o legătură eficientă din punct de vedere al costurilor cu rate mari de transfer de date, adică 10 Gbit/s pe distanțe apreciabile și oferă o soluție potențială pentru blocajul de backhaul.

De asemenea, există astăzi un interes tot mai mare în ceea ce privește comunicarea în spațiul de unde ultraviolete (UVC – UltraViolet Communication), ca urmare a progreselor recente ale surselor detectorilor optici în stare solidă care funcționează în spectrul UV cu fascicul solar (200-280 nm). În această așa-numită bandă UV profundă, radiația solară este neglijabilă la nivelul solului și acest lucru face posibilă proiectarea detectoarelor de numărare a fotonilor cu receptoare de câmp larg, care măresc energia recepționată cu zgomot de fundal suplimentar. Astfel de modele sunt deosebit de utile pentru configurațiile în aer liber care nu sunt în aceeași linie de comunicare (non-line-of-sight) pentru a susține UVC cu rază scurtă de acțiune, cum ar fi în rețelele de senzori wireless și ad-hoc.

## 2. Descrierea tehnologiei și cerințele comunicării prin spectrul vizibil de lumină

Sistemele VLC oferă avantaje tehnice și operaționale esențiale, cum ar fi cerințele de putere redusă, capacitatea de bandă mult mai mare (de la 380 la 750 nm, de la 430 la 790 THz care este de 10.000 mai mare decât spectrul RF).

Acesta poate fi util pe un domeniu larg de aplicații de comunicații, începând cu intervale de milimetri interconectate în cadrul circuitelor integrate prin legături în kilometri în aer liber. VLC este destinat să utilizeze LED-uri ca emițător și PD ca receptor.

Semnalele infraroșii au fost utilizate pentru transmisia de date fără fir aplicată rețelelor locale fără fir de interior. Lungimile de undă ale surselor de lumină infraroșie și vizibilă sunt aproape una de cealaltă, iar semnalele, prin urmare, dezvăluie calitativ un comportament similar de propagare. Ca o consecință, LED-urile albe au început să atragă atenția spre utilizare ca mijloc de comunicare a datelor.

În plus, LED-urile albe au o luminozitate foarte ridicată, consumuri de energie foarte scăzute și o durată lungă de viață. Din acest motiv, o caracteristică a LED-urilor albe este că pot servi îndouă scopuri în același timp: iluminarea și transmisia de date wireless de mare viteză

Canalul prin care semnalul se va propaga este prin spațiul de lumina vizibilă. Cu toate acestea, orice mediu care permite trecerea luminii poate fi considerat drept canal. Deoarece lumina vizibilă nu poate trece prin obiecte opace cum ar fi pereți, arbori etc. semnalul este limitat la o zonă locală și nu poate fi interceptat de nici un receptor nedorit, asigurând astfel o conexiune securizată.

LED-ul reprezintă o sursă de lumină mică, de cele mai multe ori însoțită de un circuit electric ce permite modularea formei radiației luminoase. De cele mai multe ori acestea sunt utilizate ca indicatori în cadrul dispozitivelor electronice, dar din ce în ce mai mult au început să fie utilizate în aplicații de putere ca surse de iluminare. Culoarea luminii emise depinde de compoziția și de starea materialului semiconductor folosit, și poate fi în spectrul infraroșu, vizibil sau ultraviolet. Pe lângă iluminare, LED-urile sunt folosite din ce în ce mai des într-o serie mare de dispozitive electronice.

În ceea ce privește panoul fotovoltaic principiul acestuia de funcționare se bazează pe efectul fotoelectric. Efectul fotoelectric, respectiv transformarea energiei solare ("foton") în energie electrică ("volt") a fost descoperit în 1839 de fizicianul A. Becquerel.

Acest efect fotoelectric se bazează pe trei fenomene fizice simultane, strâns legate între ele.

*Absorbția luminii de către materiale*

Fotonii compun lumina. Aceștia pot penetra anumite materiale, sau chiar să le traverseze. În general, o rază de lumină care atinge suprafața unui mediu, poate suporta trei fenomene optice:

- Reflexia: lumina este "întoarsă" de către suprafață;
- Transmisia: lumina traversează obiectul;
- Absorbția: lumina penetrează obiectul și nu îl mai părăsește, energia fiind restituită într-o altă formă.

Într-un material fotoelectric, o parte a energiei fluxului luminos va fi restituită sub forma de energie electrică. Trebuie deci ca materialul să aibă capacitatea de a absorbi lumina vizibilă, aceasta fiind cea ce se dorește a se converti: lumina solară sau a altor surse artificiale ca de exemplu LED-ul nostru din cadrul experimentului.

*Transferul de energie de la fotoni la sarcinile electrice*

Sarcinile elementare ce vor determina apariția unui curent electric în urma iluminării, sunt electronii. Fotonii vor ceda energia lor, electronilor periferici, ceea ce le va permite să se elibereze de atracția exercitată de nucleu. Acești electroni eliberați vor putea forma un curent electric, dacă sunt extrași din material.

### 3. Aplicația practică

Sistem Li-Fi din figura 1 este format dintr-un PC cu rol de preluare sunete din internet și comunicarea lui cu ajutorul driverului Li-Fi cu LED atașat, prin spațiul de lumina vizibilă către panoul fotovoltaic care are atașate boxe pentru redarea sunetului.

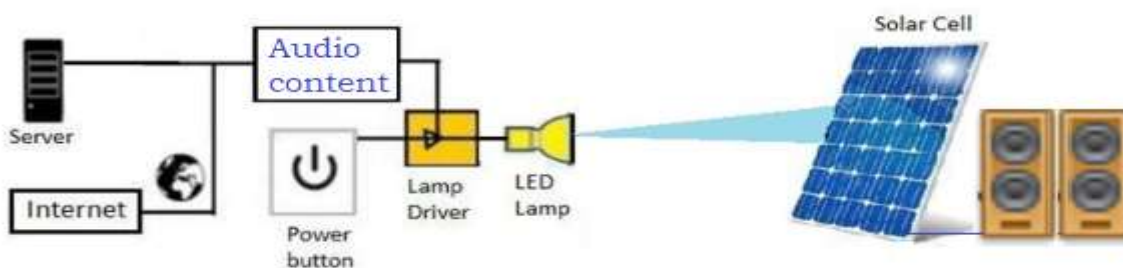


Fig 1. Schema generală a sistemului Li-Fi

Emițătorul sistemului Li-Fi dezvoltat, conectat la un sistem PC, conține un circuit electronic realizat cu tranzistori, rezistențe și condensatori care au ca scop oprirea și pornirea LED-ului cu frecvențe foarte mari (imperceptibile ochiului nostru).



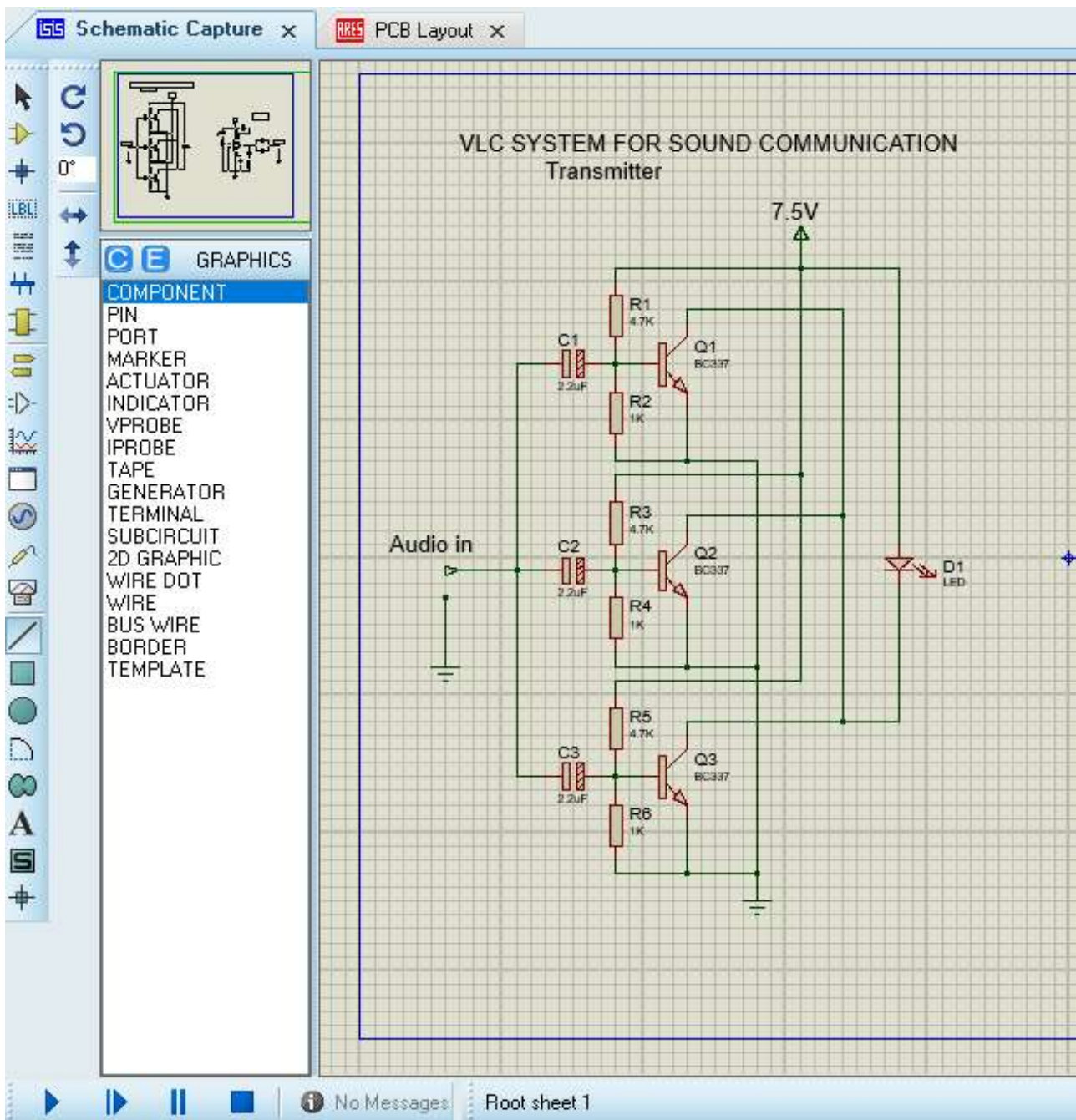


Fig. 2. Emitatorul sistemului LiFi dezvoltat pentru demonstrarea conceptului

Acesta transmite wireless, datele de tip sunet, prin intermediul luminii vizibile, la receptorul Li-Fi format dintr-un panou fotovoltaic și un sistem audio (boxe) care vor reda sunetele recepționate de la emițătorul Li-Fi.

### Concluzii

Am dezvoltat un prototip Li-Fi de comunicare wireless de date de tip audio. Transmisia se bazează pe topologia de aliniere directă între emițător și receptor, față în față, fără obstacole între cele două module.

S-a demonstrat că LED-ul alb, bazat pe sistemul de transmisie a datelor cu lumină vizibilă, pentru comunicarea fără fir a sunetului este într-adevăr fezabil din punct de vedere tehnic.

Li-Fi este o tehnologie potrivită pentru viitor, deoarece oferă o lărgime de bandă mai mare, o viteză crescută a datelor și o securitate sporită în comparație cu tehnologia Wi-Fi.

Succesul experimentului realizat mai sus arată că configurarea Li-Fi poate fi realizată cu materiale nu foarte scumpe, existente pe piață.

Pe baza experimentului, se poate concluziona că distanța dintre receptor și sursă, unghiul relativ al panoului solar receptiv și LED-ul de transmisie și suprafața panoului solar la lumină joacă un rol important în transmisie de date în sistemul Li-Fi.

Toți acești factori trebuie luați în considerare înainte de implementarea sistemului.

**Bibliografie**

1. Aayush Gupta, Syed Adeebu, Hafiz Iltesha, Wireless Data Transfer using Solar Panel in Li-Fi Technology International Journal of Advanced Research in Computer and Communication Engineering -
2. Riurean Simona, Stoica Razvan, Leba Monica, Visible Light Communication for Audio Signals, WSEAS Berlin 2017, IARAS International Journal of Communications ISSN: 2367-8887, Volume 2, 2017 <https://www.iaras.org/iaras/journals/caijoc/visible-light-communication-for-audio-signals>
3. Uysal M. and NouriH., "Optical Wireless Communications – An Emerging Technology", 16th International Conference on Transparent Optical Networks (ICTON), Graz, Austria, July 2014
4. "Visible Light Communication". [www.ntu.edu.sg](http://www.ntu.edu.sg). Retrieved 2015-12-24.

# ANALIZA VULNERABILITĂȚILOR DE SECURITATEA SITE-URILOR

**Autori:** Darius MOLDOVAN<sup>1</sup> și Dan-Dumitru MANU<sup>2</sup>  
t3jv11@gmail.com

**Coordonator:** Sef lucrări dr. Ing. Simona Mirela Rîurean<sup>3</sup>

<sup>1</sup> Universitatea din Petrosani, Facultatea IME, Specializare Calculatoare anul II

<sup>2</sup> Universitatea din Petrosani, Facultatea IME, Specializare Calculatoarea anul IV

<sup>3</sup> Universitatea din Petrosani, Facultatea IME, Departamentul ACIEE

## **Abstract**

*Aceasta lucrare are ca obiectiv realizarea unui studiu privind adoptarea HTTPS (Hypertext Transfer Protocol Secure) în cadrul site-urilor web din câteva instituții de învățământ postuniversitar, din Romania. Am analizat zece site-uri web din zece județe diferite pentru a vedea dacă certificatele și lanțul de certificare este corect instalat. Pe lângă analiza acestui certificat am găsit și o serie de vulnerabilități care pun în pericol securitatea datelor instituțiilor respective. Vulnerabilitățile găsite sunt de la nivel de importanță minoră până la critice.*

## **Cuvinte cheie:**

*HTTPS, vulnerabilitate, atacuri cibernetice.*

## **Introducere**

Marea majoritate a unitățile de învățământ preuniversitar (verificate în demersul nostru) au un site web oficial. De obicei, aceste site-uri web a fost realizate în mod tradițional utilizând protocolul HTTP (Hypertext Transfer Protocol). Cu toate acestea, site-urile nu include niciun mecanism de protecție împotriva atacurilor cibernetice, cum ar fi cele de tip "Man-in-the-middle".

Protocolul Hypertext Transfer Protocol Secure (HTTPS) asigură *autentificarea, integritatea și confidențialitatea* comunicării dintre un browser și un site web, care este considerat fundamental pentru o interacțiune demnă de încredere între utilizator și site-ul respectiv. În plus, protecția oferită de HTTPS este conformă cu propunerea europeană privind confidențialitatea în mediul electronic (Rîurean et al., 2010).

*Autentificarea* este demersul de confirmare a adevărului unui atribut al unei singure date revendicate de o entitate și se poate referi la confirmarea identității unei persoane prin validarea documentelor de identitate și verificarea autenticității unui site web cu un certificat digital.

*Integritatea* datelor este un concept care se referă la acțiuni consecvente, conform valorilor, metodelor și elementelor de măsură, precum și de raportare la principii și așteptări care pot fi verificate.

*Confidențialitatea* este un factor foarte important deoarece a proteja și a păstra confidențialitatea informațiilor înseamnă a garanta că ele nu sunt făcute disponibile sau dezvaluite entităților neautorizate.

Tinând cont că de câțiva ani încoace s-a migrat de la HTTP la protocolul HTTPS, aducând un plus în ceea ce privește partea de securitate, prin lucrarea de față ne propunem să investigăm utilizarea acestui protocol, cu grad de securitate sporit, în construcția site-urilor instituțiilor de învățământ preuniversitar.

În aceasta lucrare, pe lângă necesitatea de a utiliza protocolul HTTPS, prezentăm și alte norme de protecție a site-urilor web, cu referire la cele analizate din învățământului preuniversitar.

Există studii anterioare ce abordează acest subiect, privind implementarea și utilizarea protocolului și a certificatului HTTPS. Studiul de față tratează și modalitatea de implementarea cât mai corectă în ceea ce privește mentenanța datelor încărcate pe server.

Toate adresele Uniform Resource Locator (URL) au o ancoră HTTP, în loc de una HTTPS. Majoritatea URL-urilor conțin un nume Domain Name Server (DNS) care urmează o interfață omogenă și directă, cum ar fi "www.numa\_site.com". Am efectuat o serie de cereri și analize automatizate către toate cele 10 adrese URL, utilizând mai multe instrumente și apelând la analiza manuală cu un browser numai atunci când acest lucru s-a impus necesar.

## 2. Instrumentele folosite pentru verificare

Instrumentele folosite pentru verificarea prezenței site-urilor HTTPS, vulnerabilitățile existente, analiza porturilor și analiza de informații stric la nivel de fișiere ascunse (robots.txt, admin panel, Cpanel, phpadmin etc.): **openssl, nmap, gnutls-cli, dirb, wget, nikton, OpenWPM**. Ne referim, în lucrarea de față doar la câteva instrumente utilizate strict pentru analiza certificatului HTTPS și precum și a protocoalele de securitate.

Parte din instrumentele utilizate pentru analiza site-urilor web, sunt următoarele:

### Gnutls-cli

Acest instrument este folosit, de obicei, pentru a configura o conexiune Transport Layer Security (TLS) care trimite datele la intrarea standard către suportul securizat și invers. Acesta mai poate fi folosit și pentru analiza certificatelor HTTPS (vezi figura 1).

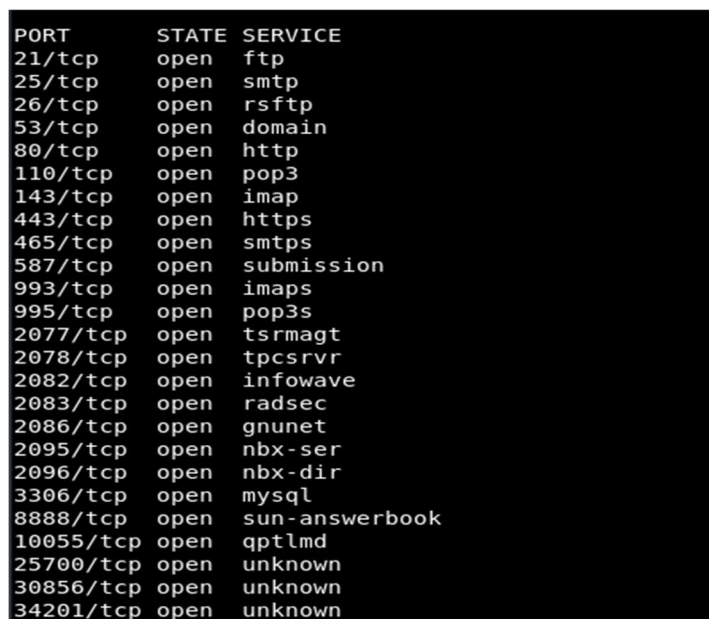
### Openssl

Este un instrument de tip linie de comanda - Command Line Interface (CLI) open-source, care este folosit în mod obișnuit pentru a genera chei private, pentru a instala certificatul Secure Socket Layer/ Transport Layer Security (SSL/TLS) și a identifica informațiile despre certificat. Acesta este foarte similar cu gnutls-cli.

### Nmap

Este un instrument gratuit, open-source pentru *descoperirea rețelei și auditul securității*. Administratorii de rețea le consideră de asemenea utile pentru sarcini precum inventarul de rețea, gestionarea programelor de actualizare a serviciilor și monitorizarea timpului de funcționare a gazdei sau a serviciului. Nmap utilizează pachete IP brut în moduri noi pentru a determina ce gazde sunt disponibile în rețea, ce servicii le oferă acelor gazde, ce sisteme de operare sunt folosite, ce tipuri de filtre de pachete / firewall-uri sunt în uz, precum și o serie de alte caracteristici.

Acesta a fost proiectat pentru a scana rapid rețele mari, dar funcționează bine și pentru gazde unice (figura 2).



PORT	STATE	SERVICE
21/tcp	open	ftp
25/tcp	open	smtp
26/tcp	open	rsftp
53/tcp	open	domain
80/tcp	open	http
110/tcp	open	pop3
143/tcp	open	imap
443/tcp	open	https
465/tcp	open	smtps
587/tcp	open	submission
993/tcp	open	imaps
995/tcp	open	pop3s
2077/tcp	open	tsrmagt
2078/tcp	open	tpcsrvr
2082/tcp	open	infowave
2083/tcp	open	radsec
2086/tcp	open	gnunet
2095/tcp	open	nbx-ser
2096/tcp	open	nbx-dir
3306/tcp	open	mysql
8888/tcp	open	sun-answerbook
10055/tcp	open	qptlmd
25700/tcp	open	unknown
30856/tcp	open	unknown
34201/tcp	open	unknown

Fig 1. Scanarea Porturilor

Analiza realizată se referă la:

**MITM – Man-in-the-middle** - Este un tip de atac declanșat în mod secret prin care atacatorul schimbă modul de comunicare dintre două părți care sunt conectate direct între ele. Un exemplu de atac MITM este interceptarea activă, în care atacatorul face legături independente cu victimele și transmite mesaje între ele pentru a le face să creadă că vorbesc direct unul cu celălalt în legătură cu o conexiune privată, atunci când de fapt întreaga conversație este controlată de atacatorul. Atacatorul trebuie să fie capabil să intercepteze toate mesajele relevante care trec între cele două victime și să injecteze altele noi.

```

Processed 133 CA certificate(s).
Resolving 'www.avramiancucluj.ro:443'...
Connecting to '89.40.33.64:443'...
- Certificate type: X.509
- Got a certificate list of 2 certificates.
- Certificate[0] info:
  - subject `CN=avramiancucluj.ro', issuer `CN=Let's Encrypt Authority X3, O=Let's Encrypt, C=US', serial 0x03240f35fa4e39ad9db93e7f9306d57d5640, RSA key 2048 bits, signed using RSA-SHA256, activated `2019-03-18 02:57:43 UTC', expires `2019-06-16 02:57:43 UTC', pin-sha256="dsXWIn1JuxbzYVUP8xzEZLiNjqd9ttBfTJ4/filqD6Q="
    Public Key ID:
      sha1:3237f6cb638e5828b46b423f4bc3f8b1b047eb45
      sha256:76c5d6227d49bb16f3615b8ff31cc464b88d8ea77db6d05f4c9e3f7e296a0fa4
    Public Key PIN:
      pin-sha256:dsXWIn1JuxbzYVUP8xzEZLiNjqd9ttBfTJ4/filqD6Q=
    Public key's random art:
      +--[ RSA 2048 ]-----+
      |
      |          . E S
      |        .+.0 * 0
      |       .ooB.o . .
      |      .=*B 0 00.
      |     .=*0. .0+.
      +-----+
- Certificate[1] info:
  - subject `CN=Let's Encrypt Authority X3, O=Let's Encrypt, C=US', issuer `CN=DST Root CA X3, O=Digital Signature Trust Co.', serial 0x0a0141420000015385736a0b85eca708, RSA key 2048 bits, signed using RSA-SHA256, activated `2016-03-17 16:40:46 UTC', expires `2021-03-17 16:40:46 UTC', pin-sha256="YLh1dUR9y6Kja30RrAn7JKnb0G/uEtLMkBgFF2Fuihg="
  - Status: The certificate is trusted.
  - Description: (TLS1.2)-(ECDHE-RSA-SECP256R1)-(AES-256-GCM)
  - Session ID: DF:0F:E3:6E:FB:4D:17:47:3F:CF:D6:B9:72:62:E5:82:79:49:67:36:34:FA:C0:FA:52:22:C0:CC:D4:AD:15:76
  - Ephemeral EC Diffie-Hellman parameters
    - Using curve: SECP256R1
    - Curve size: 256 bits
  - Version: TLS1.2
  - Key Exchange: ECDHE-RSA
  - Server Signature: RSA-SHA256
  - Cipher: AES-256-GCM
  - MAC: AEAD
  - Compression: NULL
  - Options: safe renegotiation, OCSP status request,
  - Handshake was completed

- Simple Client Mode:

- Peer has closed the GnuTLS connection

```

Fig 2. Analiză Certificat TLS

**Access Baza de date** – Acesta nu este chiar o vulnerabilitate ca SQL Injection , e mai mult o greseala facuta de administratorul de sistem in care s-a creat un link din greseală care te direcționează direct în baza de date.

**XSS - Cross Site Scripting** – Atacurile sunt de tip injectare, în care scripturile malware sunt injectate în alte site-uri web. Atacurile XSS apar atunci când un atacator utilizează o aplicație web pentru a trimite un cod rău intenționat, în general sub forma unui script al unui browser, unui alt utilizator final. Defectele care permit acestor atacuri să reușească sunt destul de răspândite și apar oriunde o aplicație web utilizează intrări de la un utilizator în cadrul producției pe care o generează fără validarea sau codarea acesteia. Un atacator poate folosi XSS pentru a trimite un script malware unui utilizator. Browserul utilizatorului final nu are cum să știe că scriptul este infectat și îl va executa . Deoarece crede că scenariul a provenit dintr-o sursă de încredere, scriptul malware poate accesa orice cookie-uri, jetoane de sesiune sau alte informații sensibile păstrate de browser și de site-ul respectiv.

**Buffer Overflow** – Pe scurt atacatorii pot exploata o eroare de depășire a buffer-ului prin injectarea unui cod care este adaptat în mod special pentru a provoca depășirea buffer-ului cu partea inițială a unui set de date, apoi scrie restul datelor la adresa de memorie adiacentă buffer-ului de descărcare. Datele de depășire ar putea conține cod executabil care permite atacatorilor să ruleze programe mai mari și mai sofisticate sau să li se acorde acces la sistem.

**XST - Cross Site Tracing** – Atacul presupune utilizarea de Scripting Cross-site (XSS) și metodele TRACE sau TRACK HTTP. "TRACE REQUEST" permite clientului să vadă ce este primit la celălalt capăt al lanțului de solicitare și să utilizeze aceste date pentru teste sau informații de diagnosticare". Metoda TRACK funcționează în același mod. XST ar putea fi folosit ca metodă de a fura cookie-urile utilizatorilor prin Scripting Cross-site (XSS), chiar dacă cookie-ul are setul de pavilion "HttpOnly" și / sau expune antetul de autorizare al utilizatorului.

**Hidden Folder** – Directoarele și fișierele ascunse lăsate accidental pe serverul web ar putea fi o sursă foarte valoroasă de informații sensibile. Pot exista o mulțime de informații ascunse în directorul rădăcină de la aplicațiile web: fișiere de sisteme (\*.git, \*.gitignore, \*.svn), fișiere de configurare (\*.npmrc, package.json, \*.htaccess), fișiere de configurare personalizate cu extensii comune cum ar fi config.json, config.yml, config.xml, config.php , config.xmlrpc, etc.

**Shellshock** – Este un bug de securitate care determină Bash să execute neintenționat comenzi din variabilele de mediu. Cu alte cuvinte, exploatarea vulnerabilității permite atacatorului să emită de la distanță comenzi pe server, cunoscut și sub numele de executare de cod la distanță. Chiar dacă Bash nu este un serviciu orientat spre internet, multe servicii de internet și de rețea, cum ar fi serverele web, utilizează variabile de mediu pentru a comunica cu sistemul de operare al serverului.

## Rezultatele cercetării

Din cele 10 site-uri analizate, 2 au certificatul SSL/TLS implementat corect, 4 site-uri sunt vulnerabile la XSS , iar într-un site am remarcat peste 50 de vulnerabilitati.

Trei dintre site-urile analizate au vulnerabilitati de tipul Shellshock, 2 site-uri cu Buffer Overflow, 3 site-uri cu vulnerabilitati de tipul XST, 9 site-uri cu fisiere ascunse dintre care la un site i se poate citi bash.history , 1 site care permite accesul la baza de date fără cont și parolă, 7 site-uri predispuse la atacul MITM.

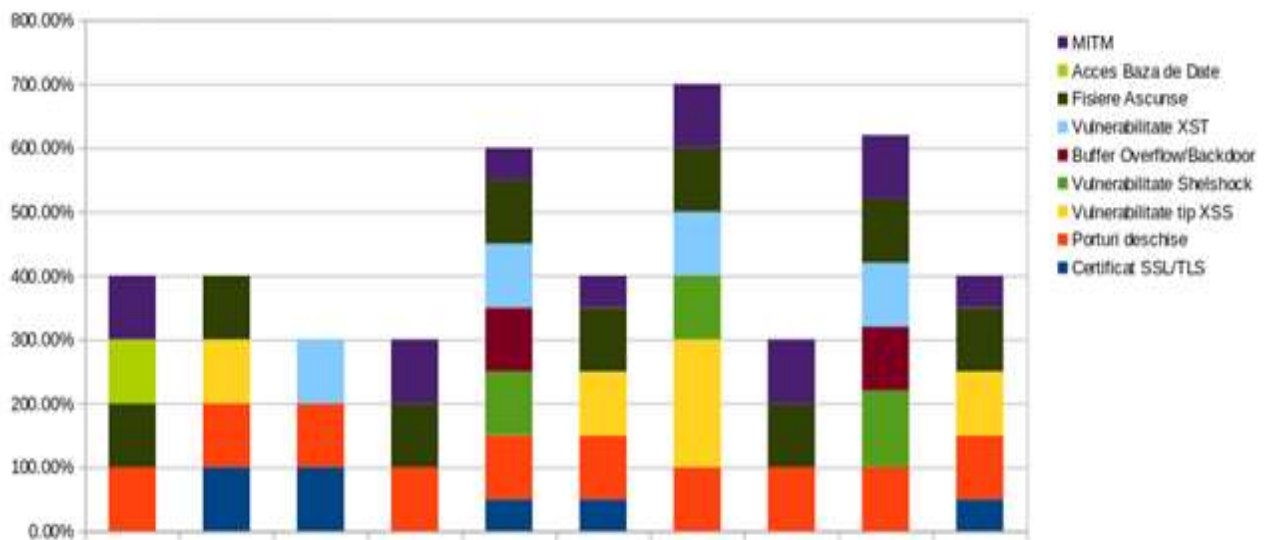


Fig. 3. Rezultatele cercetării

## Concluzii

Facând o scurtă analiză asupra site-urile câtorva instituții din cadrul învățământului preuniversitar, am observat că multe din aceste site-uri nu au acest certificat de tip HTTPS. În plus, au câteva vulnerabilități destul de serioase, ceea ce ar trebui să ne dea un pic de gândit. Existența acestor vulnerabilități poate părea o problemă ne semnificativă, dar lucrurile nu stau tocmai așa, deoarece orice fel de tipuri de date, indiferent că aparțin persoanelor fizice sau instituțiilor publice trebuie să aibă cel puțin un grad minim de protecție, de aceea, această situație se poate și trebuie îmbunătățită. Orice mică scăpare de informație precum și eroările de protecția datelor, pot constitui pericole din punct de vedere al securității informațiilor, de aceea, aceste implementări necesare tuturor site-urilor publice ar trebui considerate absolut necesare și prin urmare luate în serios.

## Bibliografie

- 1) Corbu C., Riurean S., Noțiuni introductive în internet și comerț electronic, Editura Universitas, Petrosani 2009
- 2) Riurean S., Corbu C. E - business și internet, Editura Universitas, Petrosani, 2010
- 3) <https://tutorials.technology/tutorials/Test-SSL-TLS-with-GnuTLS-from-the-Command-Line.html>
- 4) <https://www.digicert.com/ssl-support/openssl-quick-reference-guide.htm>
- 5) <https://nmap.org/>
- 6) [https://gnutls.org/manual/html\\_node/gnutls\\_002dcli-Invocation.html](https://gnutls.org/manual/html_node/gnutls_002dcli-Invocation.html)
- 7) [https://en.wikipedia.org/wiki/Transport\\_Layer\\_Security](https://en.wikipedia.org/wiki/Transport_Layer_Security)
- 8) <https://tools.ietf.org/html/rfc6101>

# STUDIUL OSCILATOARELOR LC UTILIZÂND PLATFORMELE MULTISIM ȘI NI ELVIS

**Autori:** OGORODNIC Evelina<sup>1</sup>, GHETIVU Maxim<sup>2</sup>  
[eva-syte@mail.ru](mailto:eva-syte@mail.ru) / [max\\_ghetivu@mail.ru](mailto:max_ghetivu@mail.ru)

**Coordonator:** Conf.univ.dr.ing. Nicolae Pătrășcoiu ,  
Asist.cercet.drd.ing. Cosmin Rus

<sup>1</sup> Universitatea din Petroșani, Facultatea I.M.E., specializarea: Calculatoare , anul I

<sup>2</sup> Universitatea din Petroșani, Facultatea I.M.E., specializarea: Calculatoare , anul I

## Rezumat

O categorie importantă de oscilatoare sinusoidale – adică oscilatoarele care generează de forma sinusoidală - o formează oscilatoarele cu inductanță și capacitate, numite și oscilatoare LC. Principiul de funcționare a acestui tip de oscilatoare duce la obținerea la ieșirea acestora a unor oscilații de amplitudine constantă, intretinute forțat. Această lucrare oferă o prezentare a funcționării oscilatoarelor LC utilizând platformele MULTISIM și NI ELVIS. Voi compara simularea funcționării circuitului cu Multisim, cu funcționarea circuitului în timp real cu instrumental ELVIS. Pe baza analizelor și a evaluării se poate evidenția funcționarea în regim dinamic a acestor tipuri de circuite.

### Cuvinte cheie

Oscilatoarele LC, MULTISIM, NI ELVIS, condensator, bobina.

## 1. Introducere

Procesele oscilante sunt prezente în orice sistem tehnic în a cărui structură se regăsesc elemente capabile să transforme energia dintr-o formă în alta, cel mai des energia cinetică în energie potențială și invers. În radiotehnică procesele oscilante sunt fundamentale, sistemele care le produc fiind denumite circuite oscilante sau circuite acordate. Într-un astfel de circuit este necesară prezența a două elemente capabile să stocheze energie. În cazul fenomenelor electromagnetice, după cum se știe, aceste două elemente sunt:

- **Condensatorul**, caracterizat de o capacitate electrică  $C$ , capabil să acumuleze **energie electrică** (în câmpul electrostatic ce se creează între armăturile sale), care este o energie potențială (motiv pentru care se mai numește și energie electropotențială);
- **Bobina**, caracterizată de o inductanță  $L$ , capabilă să acumuleze **energie magnetică** (sub forma câmpului magnetic pe care-l generează în jurul ei), care este o energie cinetică (motiv pentru care mai este denumită și energie electrocinetică).

Oscilatoarele ce folosesc proprietățile selective ale circuitelor care conțin bobine și condensatoare sunt oscilatoarele de tip LC.

## 2. Funcționarea oscilatoarelor LC

Se știe că la un circuit format dintr-o bobina și un condensator pot avea loc oscilații libere, care sunt sinusoidale și au o frecvență determinată de inductanța și de capacitatea din circuit. Aceste oscilații sunt amortizate, adică amplitudinea lor scade în timp, devenind imperceptibilă după un anumit număr de perioade. Oscilatoarele LC se folosesc, de obicei, la frecvențe mari (oscilatoare de radiofrecvență) din cauza caracteristicilor lor de zgomot de fază bună și ușurința lor de punere în aplicare.

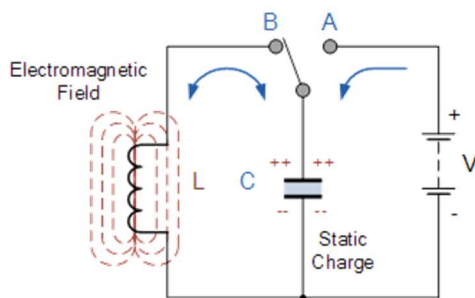


Fig.1.Circuitul de bază al unui oscilator LC



Circuitul constă dintr-o bobină inductivă, L și un condensator, C. Condensatorul stochează energia sub forma unui câmp electrostatic care produce un potențial (tensiune statică) pe plăcile sale, în timp ce bobina inductivă își stochează energia sub forma unui câmp electromagnetic. Condensator este încărcat până la tensiunea de alimentare, V prin punerea comutatorului în poziția A. Când condensatorul este încărcat complet, comutatorul se modifică în poziția B. Condensator este conectat în paralel cu bobina astfel încât condensator începe să se descarce prin bobina. Tensiunea în C începe să scadă ca curentul prin bobina să crească. Acest curent în creștere stabilește un câmp electromagnetic în jurul bobinei care rezistă curentului. Când condensatorul, C este complet evacuat de energie care a fost inițial stocat în condensator, C ca un câmp electrostatic este acum stocat în bobină, L ca un câmp electromagnetic în jurul bobinelor. Deoarece nu există acum o tensiune externă în circuit pentru a menține curentul în bobină, aceasta începe să scadă pe măsură ce câmpul electromagnetic începe să se prăbușească. EMF este indusă în bobină ( $\epsilon = -L \frac{dI}{dt}$ ) păstrând direcția inițială a curentului. Acest curent încarcă condensatorul, C cu polaritate opusă la încărcarea inițială. Continuând să se încarce până când curentul scade la zero, iar câmpul electromagnetic al bobinei s-a prăbușit complet. Energia introdusă inițial în circuit prin comutator, a fost returnată la condensator care are din nou un potențial de tensiune electrostatică peste el, deși este acum de polaritate opusă. Condensator acum începe să se descarce din nou înapoi prin bobină și întregul proces se repetă. Polaritatea de tensiune se schimbă în dependență de energia întoarsă și mai departe între condensator și inductor producând o tensiune de tip AC sinusoidale și forma de undă curent. Acest proces formează apoi baza unui circuit de rezervor de Oscilatoare LC și teoretic acest ciclism înainte și înapoi va continua pe termen nelimitat. Cu toate acestea, lucrurile nu sunt perfecte și de fiecare dată când energia este transferată de la condensator, C la inductor, L și înapoi de la L aceasta se va repeta până la apariția pierderilor de energie care apar ce descompun oscilațiile la zero în timp. Această acțiune oscilatorie de a trece de energie înainte și înapoi între condensator, C la inductor, s-ar continua pe termen nelimitat, dacă nu au fost pierderile de energie în cadrul circuitului. Energia electrică se pierde în DC sau rezistența reală a bobinei inductoare, în dielectric a condensatorului, și în radiații din circuit astfel încât oscilația scade în mod constant până când mor complet și procesul se oprește. Apoi, într-un circuit LC practic amplitudinea tensiunii oscilatorii scade la fiecare jumătate de ciclu de oscilație și în cele din urmă va muri la zero. Oscilațiile sunt apoi declarate a fi "atenuate" cu cantitatea de amortizare fiind determinată de calitate sau Q-factor al circuitului.

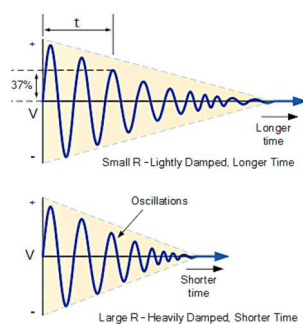


Fig. 2. Oscilații atenuate

Frecvența tensiunii oscilatorii depinde de valoarea inductanței și capacității în circuitul de rezervor LC. Știm acum că pentru ca rezonanța să apară în circuitul rezervorului, trebuie să existe un punct de frecvență, valoarea lui  $X_C$ , reactanța capacitivă este aceeași cu valoarea  $X_L$ , reactanța inductivă ( $X_L = X_C$ ) și care, prin urmare, va anula reciproc lăsând doar rezistența DC în circuit pentru a se opune fluxului de curent.

Dacă acum plasăm curba pentru reactanța inductivă a inductorului pe partea de sus a curbei pentru reactanța capacitivă a condensatorului, astfel încât ambele curbe să fie pe aceleași axe de frecvență, punctul de intersecție ne va da punctul de frecvență de rezonanță, ( $f_r$  sau  $\omega_r$ ) după cum este prezentat în fig.3.

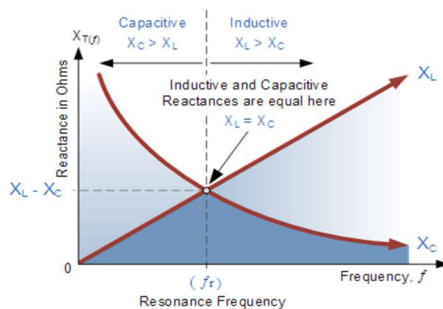


Fig.3. Frecvența de rezonanță

Unde:  $f_r$  este frecvența în Hertz (Hz), L este inductanța în Henries (H) și C este capacitatea în Farads (F)

Apoi, frecvența la care se va întâmpla acest lucru se obține prin calcule succesiv prezentate în relațiile (1)...(6) de mai jos:

$$X_L = 2\pi fL \text{ și } X_C = \frac{1}{2\pi fC} \quad (1)$$

$$X_L = X_C \quad (2)$$

$$2\pi fL = \frac{1}{2\pi fC} \quad (3)$$

$$2\pi f^2 L = \frac{1}{2\pi C} \quad (4)$$

$$f^2 = \frac{1}{(2\pi)^2 LC} \quad (5)$$

$$f = \frac{\sqrt{1}}{\sqrt{(2\pi)^2 LC}} \quad (6)$$

Prin simplificarea ecuației de mai sus vom obține ecuația finală pentru frecvența de rezonanță,  $f_r$  într-un circuit LC ca:

$$f_r = \frac{1}{2\pi\sqrt{LC}} \quad (7)$$

Unde:

- L este inductanța în Henries
- C este Capacitatea în farads
- $f_r$  este frecvența de ieșire în Hertz

Această ecuație arată că, fie L sau C sunt scăzute, frecvența crește. Această frecvență de ieșire este, de obicei, prescurtata ( $f_r$ ) pentru a identifica ca "frecvența de rezonanță". Pentru a menține oscilațiile mergând într-un circuit de rezervor LC, trebuie să înlocuim toată energia pierdută în fiecare oscilație și, de asemenea, să menținem amplitudinea acestor oscilații la un nivel constant. Cantitatea de energie înlocuită trebuie, prin urmare, să fie egală cu energia pierdută în timpul fiecărui ciclu. În cazul în care energia înlocuită este prea mare amplitudinea ar crește. Alternativ, dacă cantitatea de energie înlocuită este prea mică, amplitudinea ar scădea în cele din urmă la zero în timp s-ar opri și oscilațiile.

Cel mai simplu mod de a înlocui această energie pierdută este de a lua o parte de capacitate din circuitul de rezervor LC, amplific și apoi alimentez înapoi în circuitul LC. Acest proces poate fi realizat folosind un amplificator de tensiune folosind un amplificator operațional, transistor FET sau tranzistor bipolar ca dispozitiv activ. Cu toate acestea, în cazul în care câștigul bucla de amplificator de feedback-ul este prea mic, oscilația dorită se descompune la zero și dacă este prea mare, forma de undă devine distorsionată.

Pentru a produce o oscilație constantă, nivelul de energie alimentat înapoi la rețeaua LC trebuie să fie controlat cu exactitate. Apoi, trebuie să existe o formă de amplitudine automată sau de a câștiga controlul atunci când amplitudinea încearcă să varieze de la o tensiune de referință, fie în sus sau în jos. Pentru a menține o oscilație stabilă, câștigul global al circuitului trebuie să fie egal cu unul. Oricare mai puțin și oscilațiile nu vor începe sau vor muri departe de zero, orice mai multe oscilații vor avea loc, dar amplitudinea va deveni tăiată de șinele alimentare care cauzează distorsiune.

Oscilatorul LC este, prin urmare, un "oscilator sinusoidal" sau un "oscilator armonic". Oscilatoarele LC pot genera undele sinusoidale de înaltă frecvență pentru utilizarea în aplicațiile de tip frecvență radio (RF) cu amplificatorul Tranzistor fiind de un tranzistor bipolar sau FET. Oscilatoarele armonice vin în forme diferite, deoarece există mai multe moduri diferite de a construi o rețea de filtrare LC și amplificator cu cele mai comune fiind de Hartley LC oscilator, Colpitts LC oscilator, Armstrong oscilator și Clapp oscilator.

### Exemplu de oscilator LC

O inductanță de 200mH și un condensator de 10 pF sunt conectate împreună în paralel pentru a crea un circuit oscilant LC.

Calculul frecvenței de oscilației se realizează prin formula (7), rezultând:

$$f = \frac{1}{2\pi\sqrt{LC}} = \frac{1}{2\pi\sqrt{200mH \times 10pF}} = 112,5 \text{ kHz} \quad (8)$$

Apoi, cum putem vedea din exemplul de mai sus că, scăderea valorii condensatorului C, sau a inductanței L va avea efectul de a crește frecvența de oscilație a circuitului oscilant LC.

### 3. Modelare si simulare MULTISIM

Multisim este un mediu de simulare grafică a circuitelor electronice în care componentele sunt reprezentate prin modele SPICE. Pentru circuitul LC cu caracteristicile prezentate mai sus schema de simulare Multisim este următoarea:

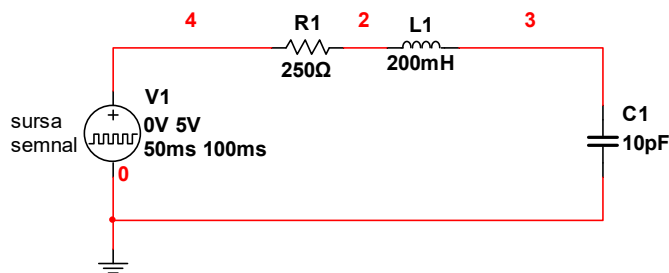


Fig.4. Schema de simulare Multisim

În care s-au utilizat următoarele elemente de circuit: bobina cu inductanța  $L=200\text{mH}$ , condensatorul cu capacitatea  $C=10\text{pF}$ , rezistorul cu rezistența  $R=250\Omega$ . Circuitul este alimentat de la o sursă de semnal de tip dreptunghiular cu perioada  $T=100\text{ms}$ , durata pulsului  $d=50\text{ms}$  și amplitudinea de  $5\text{V}$ .

Pentru vizualizarea formei semnalului în circuitul oscilant se conectează în paralel cu oricare element al schemei, în acest caz cu condensatorul un osciloscop, conform fig.5.. Se obține diagrama de semnal prezentat în fig.6.

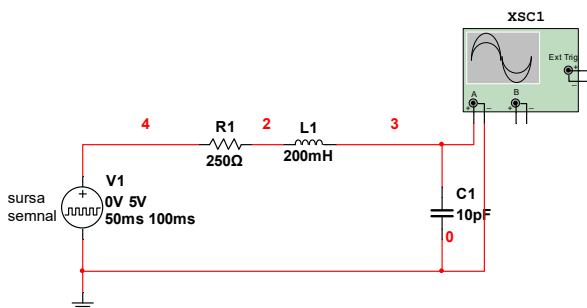


Fig.5. Conectarea osciloscopului

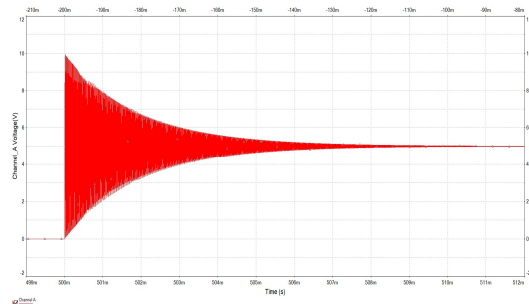


Fig.6. Diagrama de semnal

Se observă oscilațiile care apar în circuit, ca urmare a prezenței celor două elemente de circuit reactiv și anume inductanța  $L$  și capacitatea  $C$ . Aceste oscilații sunt atenuate prin prezența în circuit a rezistenței  $R$  în funcție de valoarea căreia atenuarea are loc într-un timp mai mic sau mai mare după cum valoarea rezistenței este mai mare sau mai mică.

Pentru a determina frecvența de rezonanță în vederea comparației cu frecvența calculată prin intermediul relației (8) se utilizează un instrument numit Bode Plotter conectat conform fig.7.

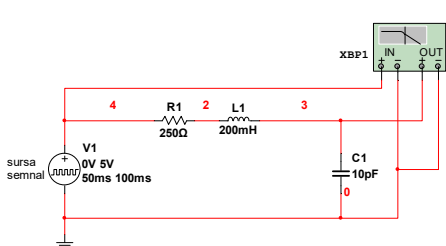


Fig.7. Conectarea Bode Plotter

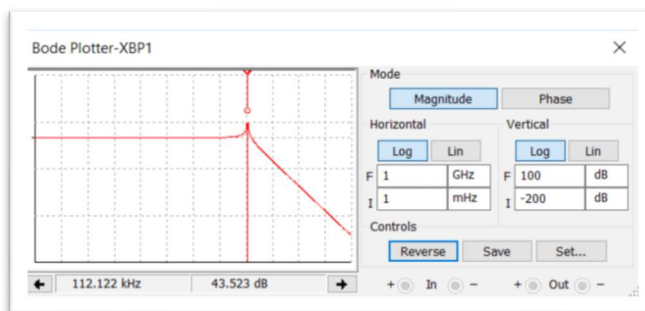


Fig.8. Determinarea frecvenței de rezonanță

Din fig.8. se observă că maximum curentului prin circuit se obține pentru o frecvență de  $112,122\text{kHz}$ , aceasta fiind valoarea frecvenței de rezonanță. Comparând rezultatul obținut prin simulare cu rezultatul obținut conform relației (8) se observă egalitatea celor două valori obținute pentru frecvența de rezonanță.

### 4. Utilizarea NI ELVIS pentru analiza circuitului

NI ELVIS este un modul cu un echipament utilizat pentru proiectarea asistată a circuitelor electronice. Este compus din partea hardware pentru realizarea fizică a circuitelor și conectarea acestora cu un sistem de achiziție de date care este conectat la calculator și partea software ce conține instrumente virtuale pentru generarea de semnale și măsurarea valorilor mărimilor electrice.

Transpunerea schemei electrice pe modulul ELVIS se face conform schemei din fig.9.

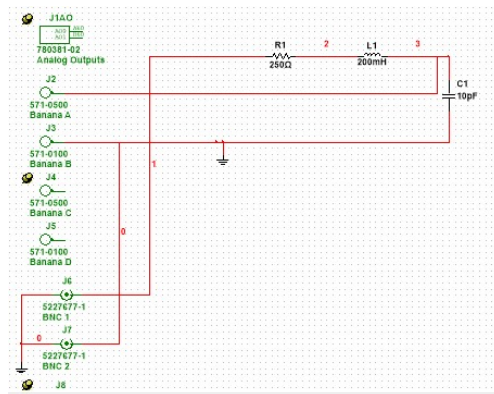


Fig.9. Plasarea elementelor de circuit pe schemă

Realizarea fizica a circuitului este prezentata in fig.10.

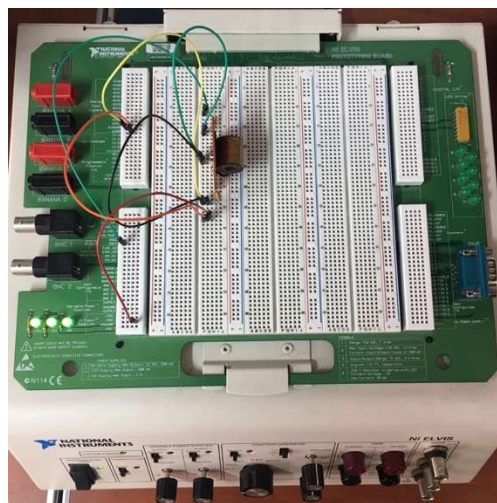


Fig.10. Realizarea fizica a circuitului pe breadboard

Dintre instrumentele virtuale disponibile sunt utilizate generatorul de semnal (Function Generator) și osciloscopul (Oscilloscope) care în urma rulării va genera fig.11.

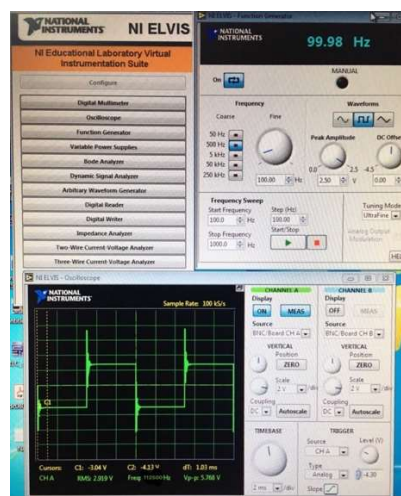


Fig.11. Răspunsul circuitului

Din fig.12. se observa că se obțin oscilații amortizate, ca și în cazul analizei circuitului prin simulare în Multisim.

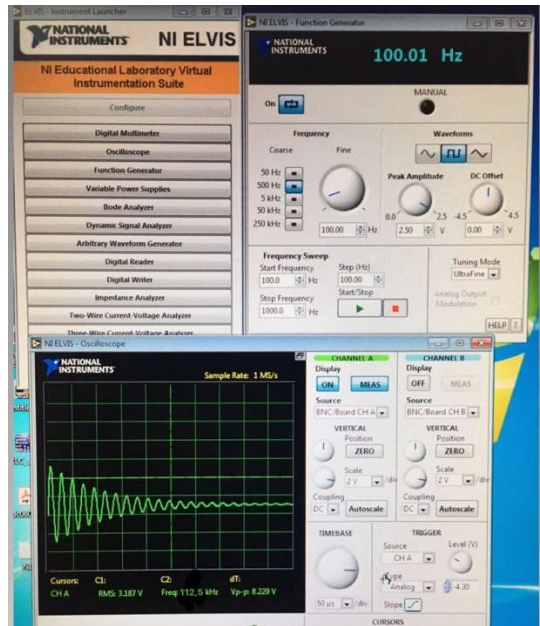


Fig.12. Diagrama semnalelor amortizate

Dacă se modifica inductanța bobinei, prin intermediul unui miez mobil al acesteia, deoarece se modifică și reactanța bobinei fenomenul de amortizare al oscilațiilor este mai pronunțată, după cum este prezentat în fig. 13.

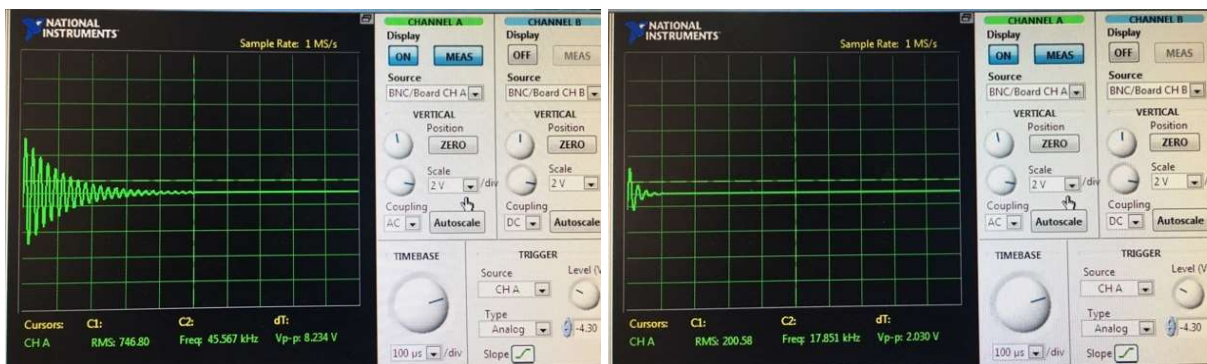


Fig.13. Reprezentarea grafică a inductanței pentru o bobină fără miez și cu miez.

## 5. Concluzii

- Pentru ca oscilațiile să existe un circuit oscilator trebuie să conțină o componentă reactivă (dependentă de frecvență) fie un "inductor", (L), fie un "condensator", (C), precum și o sursă de semnal.
- Într-un circuit LC, oscilații devin atenuate în timp din cauza pierderilor pe componente și pe conexiunile circuitului.
- Studiul unui circuit poate fi realizat prin simulare utilizând aplicații cum ar fi Multisim și modele ale componentelor a circuitului sau pe un circuit real prin generarea la intrare a semnalelor corespunzătoare și citirea semnalelor de ieșire.

## Bibliografie:

1. Pătrășcoiu Nicolae, *Electronica analogică*. Note de curs. Universitatea din Petroșani, 2015
2. Poanta Aron, *Dispozitive și circuite electronice Vol I*. Editura Universitatea Petrosani, 1997
3. Poanta Aron, *Dispozitive și circuite electronice Vol II*. Editura Universitatea Petrosani, 1997
4. <file:///Users/admin/Downloads/230013010-129905513-Oscilatoare-LC.pdf>
5. <https://www.academia.edu/10164248/Oscilatoare>
6. <http://andrei.clubcisco.ro/cursuri/f/f-sym/2eea/manual/EEA-38.pdf>

# CONCEPT DE REDUCERE A OSCILAȚIILOR CU AJUTORUL ARDUINO LA ELEMENTUL DE RIDICARE A UNEI MACARALE

**Autori:** Marian-Alin VÎLCEA <sup>1</sup>, Ioana-Nicoleta NEGRU <sup>2</sup>  
cosminrus@upet.ro

**Coordonatori:** Conf.univ.dr.ing. Nicolae Pătrășcoiu <sup>3</sup>, Asist.cercet.drd.ing. Cosmin Rus <sup>3</sup>

<sup>1</sup> Universitatea din Petroșani, Facultatea I.M.E., specializarea: STI, anul I

<sup>2</sup> Universitatea din Petroșani, Școala Doctorală., domeniul: Ingineria Sistemelor, anul I

<sup>3</sup> Universitatea din Petroșani, Facultatea I.M.E., Departamentul: A.C.I.E.E.

## Rezumat

În această lucrare se prezintă o metodă de reducere a oscilațiilor elementelor de ridicare și plasare a sarcinii unei macarale pentru a permite manipularea acesteia în spații cu un grad redus de posibilitate de manevrare în cadrul activităților specifice diverselor domenii de activitate unde este utilizat preponderent un sistem de gestionare a încărcăturilor mobile sau imobile.

Cuvinte cheie: regulator, macara, oscilații, arduino, senzor unghi

## 1. INTRODUCERE

Mașina de ridicat (macara) poate fi considerată ca fiind unul dintre cele mai importante utilaje folosite în industrie pentru a transfera sarcini dintr-un punct în altul. Mișcările efectuate de o macara constă în general dintr-o ridicare, urmată de o coborâre a sarcinii. Foarte frecvent mișcarea de deplasare a sarcinii pe verticală se combină cu mișcarea de deplasare pe orizontală, astfel că sarcina fixată în dispozitivul de prindere și ridicată într-un anumit punct poate fi coborâtă și eliberată în alt punct aflat în deschiderea sau raza de acțiune a macaralei. Conform Prescripției Tehnice ISCIR PT R1/2003 ce specifică condițiile tehnice ce trebuie să fie întrunite pentru montarea, punerea în funcțiune, utilizarea, repararea și verificarea tehnică a macaralelor în raport cu gradul de complexitate a deservirii și gradul de pericol în exploatare, macaralele se împart în patru mari grupe. Prima grupă include macaralele deplasabile pe o cale de transport fără șină de rulare (automacara, macara pe șenile, macara autopropulsată pe pneuri), o doua grupă cuprinde macaralele deplasabile pe cale cu șină de rulare având braț și platformă rotitoare (macara turn, port, Derrick), a treia categorie este reprezentată de macaralele speciale deplasabile doar pe cale de rulare prevăzută cu șină (poduri rulante, macara portal, grizi rulante suspendate, macara de turnare), ultima mare categorie este reprezentată de macaralele speciale care nu se mai regăsesc în nici o altă categorie (macara pe cablu, macara cu graifăr, macara foarfece).

## 2. PROBLEME GENERALE

Problema esențială întâlnită în aproape toate domeniile de activitate unde este folosită forța de manipulare a sarcinilor este reprezentată de posibilitatea manevrării cât mai rapidă, sigură și exactă a diverselor materii prime sau a produselor finite obținute în urma unui proces de fabricație prin utilizarea unor mașini de ridicat care să se poată modula exact pe necesitatea absolut legitimă a spectrului de activitate. S-au realizat demersuri pentru obținerea unor tehnici de control pentru a aduce sarcina preluată de brațul unei macarale aproape instantaneu în repaus la punctul final al unui profil de mișcare predefinit. S-a observat că în condiții normale de utilizare se produceau în general peste zece oscilații ale încărcăturii. S-a încercat construirea unor controllere, filtre de zgomot, care eliminau componentele frecvenței naturale a pendulului în sarcină utilă. Chiar dacă rezultatele au arătat o reducere semnificativă a pendulației sarcinii, s-a produs o întârziere de până la 2,5 secunde între momentul declanșării comenzii de către operator și execuția acesteia. Deasemenea filtrul era dependent de lungimea cablului de ridicare și parametrii acestuia trebuiau schimbați de fiecare dată când se modifica lungimea cablului. Mai mult modelarea intrării și controlării filtrului de zgomot era făcută de controllere în buclă deschisă, care le face ineficiente în prezența perturbațiilor externe. Un controllerul pretabil spre a utilizat într-un astfel de domeniu este cel de tip pornit-oprit, cunoscut ca și controller cu histerezis care comută brusc între cele două stări (0 și 1). Operatorul întregului ansamblu al mașinii de ridicat va încerca să păstreze o traiectorie cât mai lină a întregii încărcături manipulând căruciorul de transport cu mișcări fine asupra modulelor de comandă (manete, joystick-uri) obținând astfel un cadru de mișcare și poziționare sustenabil pentru mutarea și plasarea unei sarcini dar totul realizându-se cu un tip de control realizat empiric și care are un înalt grad de subiectivitate depinzând de abilitățile fiecărui manipulant [1, 2].

### 3. DESCRIEREA STRUCTURALĂ

Obiectivul acestei lucrări este de a găsi o strategie de control adecvată pentru a transfera sarcini folosind macarale portal. Strategia de control ar trebui să ia în considerare doi factori principali: timpul de execuției al întregului proces de mutare a sarcinii cât și reducerea la minim a oscilației sarcinii pentru a preveni eventualele riscuri pentru oameni și pentru echipamentele de lucru. Sarcina pendulară se caracterizează prin două unghiuri,  $\delta$  și  $\phi$ .  $\delta$  reprezentând unghiul pe care-l face cablul cu axa z în planul xz iar  $\phi$  unghiul din afara planului, condițiile absolut esențiale pentru obținerea unui comportament predictibil, sigur și eficient al întregului ansamblu fiind ca cele două unghiuri să aibă valori cât mai mici.

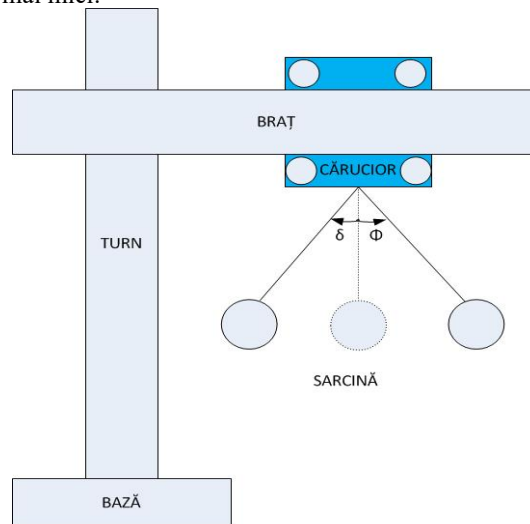


Fig. 1 Structura generală a unei mașini de ridicat

Lucrarea reprezintă un proiect de machetă la scară a unei macarale de tip portal folosită în porturi pentru descărcarea containerelor cu marfă de pe vasele comerciale.

Macheta are în componența sa un gripper acționat de un servomotor, două traductoare optice în cuadratură, unul pentru măsurarea distanței pe orizontală, iar al doilea pentru măsurarea distanței de urcare/ coborâre a gripperului, patru motoare, două pentru deplasarea pe șine, unul pentru deplasarea pe orizontală a containerului simulat și unul pentru urcarea/coborârea containerului acționat printr-un reductor 38:1, unghiul obținut în urma balansului containerului simulat în mișcare este detectat cu un senzor de înclinație.

#### ARDUINO MEGA

Modulul de comandă este compus din placa Arduino Mega 2560 bazată pe microcontrollerul Atmega2560. Placa de comandă și control acționează servomotorul gripper-ului, motorul de urcare/coborâre, motorul căruciorului de transport și motoarele folosite pentru deplasarea pe șine utilizând pini în comandă digitală, și preia valorile de la senzorul de înclinație și de la traductorul optic utilizând pini în comandă analogică [5].

#### SENZORUL STT280

Senzorul STT280 este un senzor de înclinare sigilat, conceput pentru a oferi un grad de încredere ridicat în utilizare. Senzorul măsoară înclinarea într-o carcasă extrem de compactă, fiind adaptat pentru utilizarea într-un mediu industrial. STT280 utilizează tehnologia în stare solidă 3D-MEMS (micro-electro-mecanic-sistem) pentru a măsura înclinarea senzorului în raport cu gravitația Pământului. Prin folosirea acestei tehnologii, STT280 oferă avantaje distincte de siguranță și stabilitate.

STT280 are un interval de măsurare de  $\pm 60^\circ$  și oferă la ieșire o tensiune continuă între 0,5V și 4,5V. Poate fi comandat cu două game de tensiuni de alimentare, care sunt setate din fabrică:

-opțiunea de 5V va opera de la o sursă reglementată în intervalul 4,75V – 5,25V tensiune continuă, are ieșire rațională și asigură 80% din tensiunea de alimentare la înclinații de  $\pm 60^\circ$ .

-opțiune de 8V – 30V tensiune continuă. Această versiune are un regulator intern de tensiune și furnizează la ieșire o tensiune cuprinsă între 0,5V și 4,5V la înclinații de  $\pm 60^\circ$  [6].

La conectarea senzorului, trebuie să se prevadă o protecție la polaritate inversă, la scurtcircuit între ieșire și masă, cazul în care ieșirea este conectată la polul pozitiv al sursei de alimentare, duce inevitabil la defectarea dispozitivului. Zgomotul de ieșire al traductorului este foarte redus, mai mic de 1mV RMS.

STT280 este în fapt un accelerometru liniar în raport cu componenta orizontală a gravitației terestre, „g”. Atunci când este utilizat ca înclinometru, acesta produce la ieșire o formă sinusoidală proporțională cu valoarea  $\alpha = 1g * \sin\delta$ , unde  $\delta$  este unghiul de înclinare în raport cu poziția 0g. Ieșirea urmează o lege de ieșire specifică, în funcție de tensiunea de alimentare:

La o tensiune cuprinsă între 8-30 V avem:

$V_{dc} = 5 * k * \sin\delta + 2.5$ , unde k reprezintă o valoare constantă astfel:

-0,4619 pentru înclinare de  $\pm 60^\circ$ ;

-0,8000 pentru înclinare de  $\pm 30^\circ$ ;

-1,1695 pentru înclinare de  $\pm 20^\circ$ ;

-2,3035 pentru înclinare de  $\pm 10^\circ$ .

### SENZORUL AEDS-964X

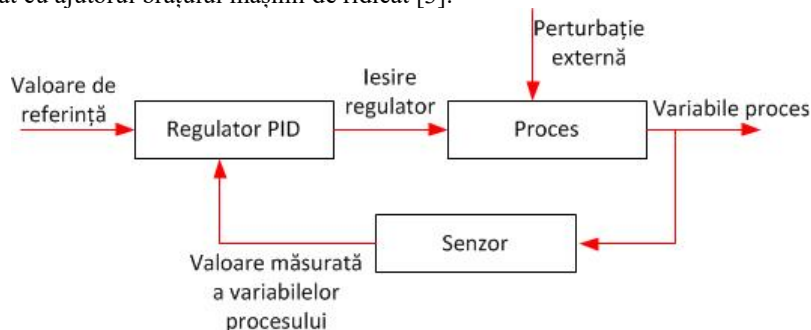
Senzorul optic AEDS-964X este un encoder optic incremental de înaltă performanță, care poate funcționa la tensiuni continue de alimentare între 3,3 – 5V. Când este folosit cu un codestrip acest modul detectează poziția liniar. Codificatorul constă dintr-un led și un detector integrat închise într-un mic pachet de plastic sub forma literei „C”. Cele două ieșiri digitale și alimentarea cu tensiune sunt accesate prin intermediul a patru conductoare situate central. LED-ul utilizat are un consumă de circa 16mA.

AEDS-964X este un modul emițător/detector cuplat cu o roată cod relativă sau în acest caz cu o riglă de codare, acesta traducând mișcarea de rotație sau translație într-o ieșire digitală. Rigla de codare incrementală se mișcă între emițător și detector, provocând întreruperea fascicolului de lumină în funcție de modelul cu spații și bare. Fotodiodele care detectează aceste întreruperi sunt aranjate într-un model care corespunde cu spațiile și barele de pe rigla de codare. Aceste detectoare sunt distanțate astfel încât pe o pereche de detectoare corespunde o bandă întunecată, iar pe detectoarele adiacente un spațiu gol care permit trecerea luminii. Ieșirile sunt alimentate prin circuite de procesare a semnalului. Două comparatoare primesc semnalele de la circuitul de procesare semnal și produc rezultate finale pentru canalele A și B. Având în vedere această tehnică integrată de ieșire canalul A este în cuadratură cu canalul B (defazat cu 90 grade) [6].

### 4. ALGORITM DE FUNCȚIONARE

Pentru automatizarea întregului proces de conducere a sistemului este necesar să se dispună de dispozitive care pe baza informațiilor primite de la proces să elaboreze mărimea de conducere. Astfel se utilizează un regulator de tip PID implementat prin intermediul software-ului utilizând o bibliotecă special create pentru mediul de programare Arduino.

Un sistem de control proporțional-integral-derivativ (PID) este un mecanism generic de control în buclă de feedback, utilizat pe scară largă în sistemele de control industrial. Un controler PID calculează o "eroare generică" ca diferență între o variabilă de proces măsurată și o valoare de referință dorită. Regulatorul încearcă să minimizeze eroarea prin ajustarea intrărilor procesului, în cazul de față prin mișcarea spre stânga și spre dreapta a căruciorului de transport al sarcinii încercându-se anularea perturbației exercitată asupra sistemului de greutatea efectivă a obiectului manipulat cu ajutorul brațului mașinii de ridicat [3].

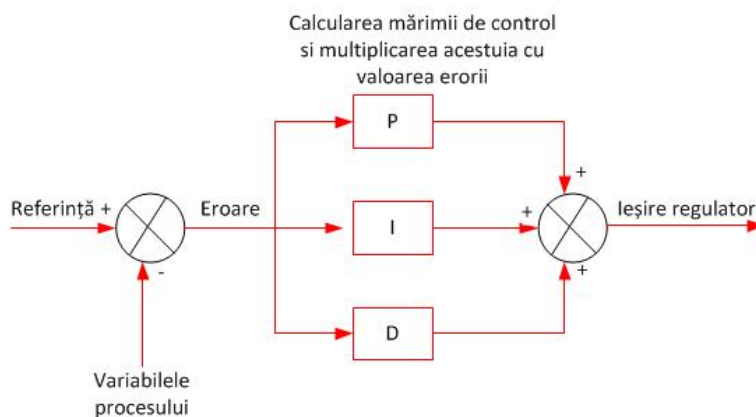


**Fig. 2 Circuit de autoreglare (regulator PID)**

Algoritmul după care funcționează un astfel de regulator implică trei parametri diferiți, pentru cazul liniar este o combinație de efecte dinamice elementare P(proporțional), I(integrator), D(derivativ). Euristic, aceste valori pot fi interpretate în funcție de timp astfel : P depinde de eroare prezentă, I de acumularea erorii din trecut, iar D



este o predicție a erorilor viitoare, în funcție de rata actuală de schimbare. Suma ponderată a acestor trei acțiuni este utilizată pentru a regla procesul printr-un element de comandă.



**Fig. 3 Principiul de funcționare al regulatorului PID**

Prin alinierea celor trei parametri în algoritmul regulatorului PID, regulatorul poate furniza schema de control proiectată pentru cerințele specifice ale unui proces. Platforma Arduino utilizând o implementare software a unui dispozitiv de reglare de tip PID preia de la senzorii utilizați, mărimile de intrare necesare pentru comanda elementelor mobile obținând astfel un caracter predictibil al întregului proces [5].

## SIMULAREA PROCESULUI

La conectarea alimentării placa Arduino comandă gripperul pentru a încărca și ridica containerul simulat. După ridicarea containerului la înălțimea stabilită, motorul de acționare pe orizontală (căruciorul de transport) intră în funcțiune deplasând containerul cu viteză spre destinație. Poziția căruciorului de transport pe orizontală este incrementată de senzorul optic. Ajuns la destinație motorul pe orizontală frânează, momentul de inerție imprimat containerului îl determină să oscileze. Intră în acțiune senzorul de înclinație și este citit unghiul de oscilație al containerului, această valoare servind ca și mărime de intrare pentru regulatorul PID, care începe să miște căruciorul de transport pe o traiectorie stâng-dreapta atenuând rapid oscilația elementului de prindere.

## 5. CONCLUZII

Utilizarea mașinilor de ridicat a permis un ritm mult mai accelerat al inovațiilor tehnologice mai ales în industria grea, obținându-se rezultate favorabile progresului general. Un astfel de utilaj extrem de necesar pentru manipularea anumitor sarcini nu ar putea fi folosit însă dacă nu ar avea încorporate anumite dispozitive care să permită un minim sistem de protecție pentru factorul uman dar și pentru restul construcțiilor sau utilajelor dintr-o incintă industrială.

Prin utilizarea unui sistem de reducere a oscilațiilor sistemului de prindere, controlat automat, al unei mașini de ridicat într-o activitate care presupune mutarea rapidă a sarcinilor pe distanțe scurte într-un spațiu limitat ca dimensiuni ar putea elimina posibilele acțiuni accidentale de natură subiectivă din partea unor operatori umani.

## BIBLIOGRAFIE

- [1] Pătrășcoiu N., Rus C., Barbu C., *Virtual Instrumentation for Data Acquisition and Remote Control*, ICCC 2017
- [2] Leba M., *Microcontrolere și automate programabile – Note de curs*, Universitatea din Petroșani, 2018
- [3] Pop E., Bubatu R., *Teoria sistemelor educație prin e-learning*, Editura Universitas, Petroșani, 2012
- [4] Sochirca B., Poanta A., *Proiectarea și dezvoltarea aplicațiilor cu microcontroler*, Editura Universitas, Petroșani, 2012
- [5]\*\*\*<http://www.arduino.cc/>
- [6]\*\*\*<http://www.robofun.ro>

# Proiectarea și implementarea unei aplicații web de tip forum

**Autori:** Radu-Alexandru PÎRCIU <sup>1</sup>, Andrei Gabriel BERCEANU <sup>2</sup>  
[alexandru37@gmail.com](mailto:alexandru37@gmail.com)/ [berceanu.andreigabriel@yahoo.com](mailto:berceanu.andreigabriel@yahoo.com)

**Coordonatori:** Conf.univ.dr.ing. Egri Angela <sup>3</sup>, Asist.cercet.drd.ing. Cosmin Rus <sup>3</sup>

<sup>1</sup> Universitatea din Petroșani, Facultatea I.M.E., specializarea: Calculatoare , anul III

<sup>2</sup> Universitatea din Petroșani, Facultatea I.M.E., specializarea: Automatică , anul III

<sup>3</sup> Universitatea din Petroșani, Facultatea I.M.E., Departamentul: A.C.I.E.E.

## Rezumat

Există o mulțime de aplicații web pe internet, aplicații pe care le folosim zilnic, fie pe desktop, fie pe mobil. De asemenea există o varietate de tehnologii ce ne permit contruirea unor astfel de aplicații, de aceea trebuie să alegem ceva care este bun, care răspunde nevoilor noastre și permite să întreținem cu ușurință aplicația de-a lungul procesului de dezvoltare, dar și pe termen lung, când aplicația ajunge în faza de producție. Luând toate acestea în considerare, avem nevoie de o tehnologie web ce ne permite dezvoltarea durabilă de aplicații, ce pot fi întreținute cu ușurință. Pe lângă aceasta, este de datoria programatorului să își ia măsuri de siguranță și să aibe grijă de modul în care construiește aplicația. În cele din urmă, atenția acordată fazei de proiectare va fi foarte utilă în viitor, când vom economisi resurse încercând să reparăm greșeli ce ar fi putut fi evitate în această etapă. Așadar, lucrul inteligent este să utilizăm o așa-numită „arhitectură curată” pentru aplicația noastră. În această lucrare, vom discuta despre modul în care putem să facem asta și totodată despre tehnologiile folosite și modul în care le vom aplica pentru a construi aplicația noastră web de tip forum.

## Cuvinte cheie

Aplicație web, .NET Core, MVC, comunicație fără fir, gateway LoRa

### 1. Introducere

.NET este un framework dezvoltat de Microsoft, care are ca principală țintă Windows, dar nu numai. Dezvoltarea aplicațiilor pe .NET are numeroase avantaje printre care se pot aminti varietatea de limbaje de programare suportate (C++, VB, C#, F#), existența unui runtime ce permite executarea oricărui program scris într-un limbaj compatibil .NET, integrarea limbajelor și multe altele. Pe lângă numeroasele limbaje de programare utilizate, .NET permite și dezvoltarea unei game foarte largi de tipuri de aplicații printre care aplicații de desktop, aplicații pe mobil, web și altele. Fiecare tip de aplicație are un framework specializat ce poate fi utilizat. Pentru aplicațiile de tip web se utilizează ASP.NET.

ASP.NET este framework-ul Microsoft pentru crearea de aplicații și servicii web. Lansat în 2002, acesta a evoluat de-a lungul timpului, odată cu versiunile .NET și permite crearea aplicațiilor utilizând diferite modele precum Web Forms (construirea paginilor a căror evenimente sunt gestionate pe server), MVC (acesta utilizează design-ul model-view-controller; despre acesta vom discuta mai pe larg, în cele ce urmează) sau Web Pages (un mod de a adăuga cod dinamic direct în markup). În prezent există două direcții de proiectare a aplicațiilor ASP, una pe platforma .NET standard, iar cealaltă pe .NET Core, cel mai nou framework Microsoft. Inovația pe care o aduce .NET Core este legată de modularitate, astfel că programatorilor le este mai ușor să scrie aplicații utilizând o serie de module, dar totodată sunt aduse îmbunătățiri legate de dimensiune și gestionarea pe server. De asemenea .NET Core este open source, comunitatea de dezvoltatori fiind motivată să participe activ la îmbunătățirea platformei prin contribuția lor adusă proiectului.

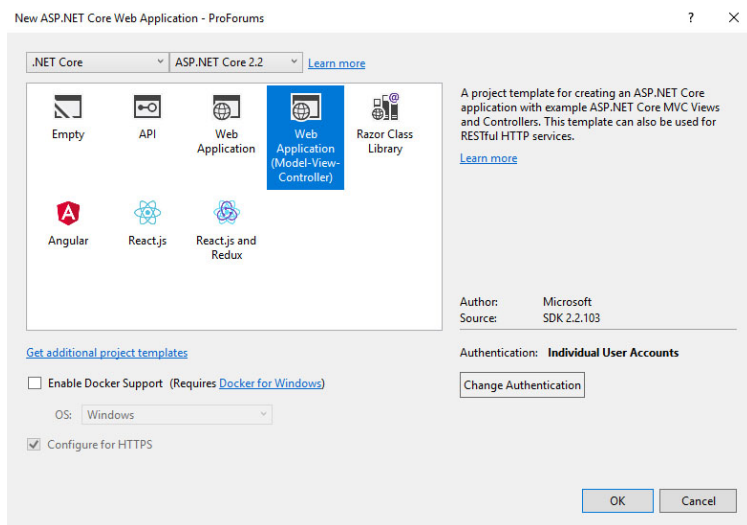
În prezent există foarte multe limbaje de programare și framework-uri ce permit dezvoltarea de back-end. Cu o serie întregă de funcționalități, precum și avantaje și dezavantaje, cel care dorește să dezvolte aplicații web este pus în fața deciziei de a-și alege direcția pe care vrea să o urmeze pentru aceasta. Întrebarea care se ridică este de ce ar alege cineva să utilizeze .NET Core pentru aceasta? În continuare vom prezenta o serie de motive:

- susținerea unei companii de talia Microsoft;
- disponerea de un număr mare de unelte puse la dispoziția de către Microsoft;
- posibilitatea de hostare pe cloud prin Azure;
- mai multe limbaje de programare din care se poate alege (pentru acest proiect vom utiliza C#);
- platforma este open source;
- posibilitatea de a dezvolta pe mai multe sisteme de operare (Windows, Linux, MacOS);
- viteza de procesare.

Acum că am prezentat platforma și am discutat despre avantajele utilizării ei, vom trece la prezentarea structurii aplicației și a arhitecturii utilizate.

## 2. Prezentarea aplicației

### 2.1 Modelul aplicației



**Figura 1** Template-urile pentru crearea aplicației web

sau .NET Core), versiunea acesteia, configurarea autentificării și template-ul utilizat. În cazul de față vom utiliza o aplicație web ce respect șablonul de design MVC.

MVC (Model-View-Controller) este un șablon ce permite crearea de aplicații web într-o manieră eficientă, astfel că fiecare componentă are un rol bine definit. Componenta Model este cea care se ocupă de abstractizarea datelor. În acest strat vom crea clasele ce sunt de fapt reprezentări ale unor concepte concrete pe care dorim să le implementăm în aplicație și cu care dorim să lucrăm (spre exemplu clasa Forum conține elementele necesare procesării unui astfel de obiect, precum titlu, descriere etc.). Componenta View este cea care se ocupă de interacțiunea cu utilizatorul. Aici vor fi toate componentele ce țin de markup, stil și dinamica paginilor web. Componenta Controller este cea care descrie logica aplicației. Prin aceasta sunt preluate cererile care vin de la utilizator, sunt atribuite unor funcții care fac legătura între informația de care utilizatorul are nevoie și servește la randul ei răspunsul, procesând modelele și view-urile în funcție de logica descrisă.

### 2.2 Structura aplicației

Odată ce ne-am decis ce fel de aplicație vom construi, putem trece la proiectarea efectivă a acesteia. Pentru început, vom analiza modul în care .NET Core tratează aplicațiile web.

Orice aplicație web este de fapt o aplicație de consolă. La crearea acesteia, observăm mai multe fișiere și directoare, figura 2, despre care vom discuta în continuare.

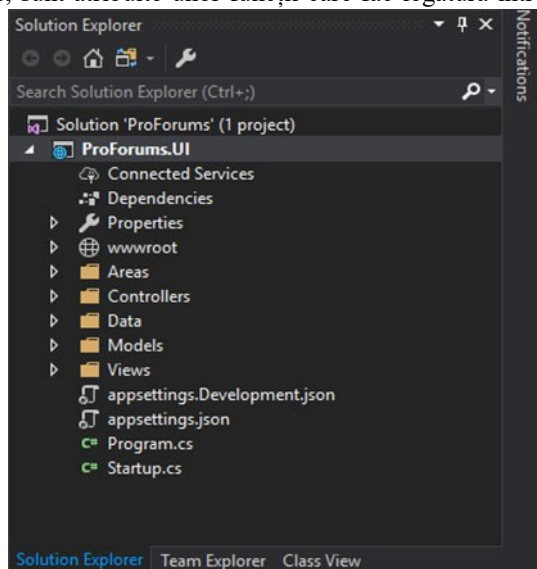
Directorul wwwroot conține fișierele statice care vor fi servite pe paginile web ale aplicației, precum fișiere CSS sau fișiere JavaScript. Se pot vedea directoarele Models, Views și Controllers care conțin fișiere ce respectă structura compozițională despre care am discutat înainte. Fișierele appsettings, de tip json, conțin datele de configurare ale aplicației, precum nivelurile de logare, stringurile pentru conectarea la baza de date și altele.

Ultimele două fișiere sunt unele importante. Fișierul program conține punctul de intrare în aplicație, după cum putem vedea în figura 3.

Așa cum spuneam, Microsoft pune la dispoziția programatorului o serie de unelte ce pot fi folosite pentru a construi o aplicație într-un mod cât mai eficient, una dintre acestea fiind IDE-ul (Integrated Development Environment) Visual Studio. Acesta oferă foarte multe facilități printre care un editor ce te ajută să scrii cod într-o manieră rapidă, oferind ajutor prin IntelliSense (completare inteligentă de cod), posibilități de testare și depanare eficiente și multe altele.

În momentul în care dorim să construim o aplicație web, trebuie să ne gândim la tipul acesteia. .NET Core pune la dispoziție o serie de template-uri pe care le putem vedea în figura 1.

În figură vedem că putem selecta platforma pe care construim (.NET Standard



**Figura 2** Structura aplicației create

```

Program.cs
ProForums.UI
1 using System;
2 using System.Collections.Generic;
3 using System.IO;
4 using System.Linq;
5 using System.Threading.Tasks;
6 using Microsoft.AspNetCore;
7 using Microsoft.AspNetCore.Hosting;
8 using Microsoft.Extensions.Configuration;
9 using Microsoft.Extensions.Logging;
10
11 namespace ProForums.UI
12 {
13     public class Program
14     {
15         public static void Main(string[] args)
16         {
17             CreateWebHostBuilder(args).Build().Run();
18         }
19
20         public static IWebHostBuilder CreateWebHostBuilder(string[] args) =>
21             WebHost.CreateDefaultBuilder(args)
22                 .UseStartup<Startup>();
23     }
24 }

```

Fig. 3 Punctul de intrare în aplicație

Deși poate părea ciudat faptul că funcția main nu face altceva decât să apeleze două funcții ale unei instanțe de IWebHostBuilder, aceasta face tot ceea ce este necesar creării aplicației. Codul din spatele acesteia este proiectat să seteze directorul aplicației, să încarce și să citească fișierele de configurare, să integreze atât IIS (serverul utilizat de Microsoft pentru rularea aplicației), precum și Kestrel (un server cross-platform utilizat pentru aplicație) și setarea fișierului de Startup. Cel din urmă conține o clasă ce are două metode: ConfigureServices și Configure.

Prima metodă este containerul pentru inversiunea dependențelor (în momentul în care dorim să utilizăm o funcționalitate specifică unei clase, nu vom utiliza o instanță a acesteia, deoarece asta ar presupune un șir de dependențe între clase care poate avea rezultate negative pe termen lung, de aceea utilizăm un contract reprezentat de o interfață care are rolul de a asigura folosirea metodelor de care avem nevoie, fără a ne interesa ce implementare a acesteia folosim; acest lucru este gestionat de către containerul de inversiune a dependențelor). Aici vom specifica ce implementări ale interfețelor dorim să folosim.

Cea de-a doua metodă reprezintă pipeline-ul care se ocupă de modul în care vor fi tratate cererile de la server. Aceste cereri vor trece prin pipeline și în funcție de ceea ce trebuie servit, se vor da răspunsurile necesare. Ordinea în care adăugăm cererile este foarte importantă, deoarece dacă nu este întrunită una dintre condiții, trecerea prin pipeline va înceta și va fi trimis răspunsul corespunzător. În figura 4 este prezentată structura clasei.

```

Startup.cs
ProForums.UI.Startup
28 // This method gets called by the runtime. Use this method to add services to the container.
29 public void ConfigureServices(IServiceCollection services)
30 {
31     services.Configure<CookiePolicyOptions>(options =>
32     {
33         // This lambda determines whether user consent for non-essential cookies is needed for a given request.
34         options.CheckConsentNeeded = context => true;
35         options.MinimumSameSitePolicy = SameSiteMode.None;
36     });
37
38     services.AddDbContext<ApplicationDbContext>(options =>
39     {
40         options.UseSqlServer(
41             Configuration.GetConnectionString("DefaultConnection"));
42     });
43     services.AddDefaultIdentity<IdentityUser>()
44         .AddDefaultUI(UIFramework.Bootstrap4)
45         .AddEntityFrameworkStores<ApplicationDbContext>();
46
47     services.AddMvc().SetCompatibilityVersion(CompatibilityVersion.Version_2_2);
48
49 // This method gets called by the runtime. Use this method to configure the HTTP request pipeline.
50 public void Configure(IApplicationBuilder app, IHostingEnvironment env)
51 {
52     if (env.IsDevelopment())
53     {
54         app.UseDeveloperExceptionPage();
55         app.UseDatabaseErrorPage();
56     }
57     else
58     {
59         app.UseExceptionHandler("/Home/Error");
60         // The default HSTS value is 30 days. You may want to change this for production scenarios, see https://aka.ms/aspnetcore-hsts.
61         app.UseHsts();
62     }
63
64     app.UseHttpsRedirection();
65     app.UseStaticFiles();
66     app.UseCookiePolicy();
67
68     app.UseAuthentication();
69
70     app.UseMvc(routes =>
71     {
72         routes.MapRoute(
73             name: "default",
74             template: "{controller=Home}/{action=Index}/{id?}");
75     });
76 }

```

Figura 4 Structura clasei Startup

### 2.3 Arhitectura aplicației

Așa cum am spus deja, un bun programator trebuie să ia în serios faza de proiectare a aplicației, deoarece alegând o structură bună pentru aceasta de la bun început, poate salva mult timp pierdut pentru depanare în faza de producție. Totodată o structură logică va ajuta la dezvoltarea cu ușurință a proiectului. Acesta este momentul în care trebuie aleasă o arhitectură pe care să construim mai departe.

Pentru aplicația pe care dorim să o proiectăm, am ales o arhitectură pe 4 straturi, după cum se poate vedea în figura 5. În cazul în care vorbim despre un proiect complex, este bine să ne gândim la o structură puțin mai complex, însă în cazul dat nu este nevoie. În continuare vom vorbi mai pe larg despre fiecare strat în parte.

Cea mai complex structură este cea a interfeței pentru utilizator (UI). Aceasta este de fapt aplicația de tip MVC și este pilonul central al proiectului. Celelalte straturi sunt pentru susținerea acestuia.

Stratul Domain este nivelul de abstractizare pentru baza de date. .NET pune la dispoziția utilizatorului un framework (Entity Framework Core) pentru relaționarea cu baza de date. Astfel, nu mai este necesară scrierea de comenzi în SQL, ci putem scrie cod care gestionează lucrul cu baza de date. În cazul de față în Domain vom avea entitățile ce reprezintă clase abstracte (asemănător cu modelele), care vor fi transpuse ulterior în tabele în baza de date.

Stratul Persistence este cel care face trecerea efectivă de la cod la baza de date. Aici se află clasa DbContext care face trecerea de la entități la DBSets, care nu sunt altceva decât tabele în baza de date. Procesul efectiv de mapare se numește migrare. Tot aici se înregistrează un istoric al migrațiilor.

În momentul creării tabelor, iar mai apoi la modificarea acestora, se vor rula migrații ce vor reflecta schimbările din cod (fiecărei proprietăți din entitate îi corespunde o coloană dintr-un tabel, iar în momentul în care adăugăm sau modificăm aceste proprietăți, trebuie să ne asigurăm că acestea sunt reflectate în baza de date).

Stratul Application este cel care se ocupă de logică. Aici sunt definite operațiile de tip CRUD (Create, Read, Update, Delete) pentru fiecare entitate din domeniu. Aici se va scrie logica ce va fi proiectată pe interfețe, iar în controllere, se vor utiliza anumite metode definite aici. Prin containerul de inversiune a dependențelor se vor servi implementările concrete ale contractelor definite aici (acestea sunt clase care se află tot în acest strat).

### 3. Concluzii

Așadar, crearea de aplicații web este un proces pe cât de plăcut, pe atât de incomod dacă nu este abordat cu atitudinea potrivită. În momentul în care dorim să creăm o astfel de aplicație trebuie avut în vedere faptul că faza de proiectare are un rol extrem de important, atât pe termen scurt, deoarece ne oferă o direcție pe care începem să construim, cât și pe termen lung deoarece un proiect de dimensiuni mari va necesita o gestionare inteligentă a componentelor acestuia. .NET Core este o soluție foarte bună, oferind numeroase avantaje, susținând procesul de creare și gestionare de aplicații.

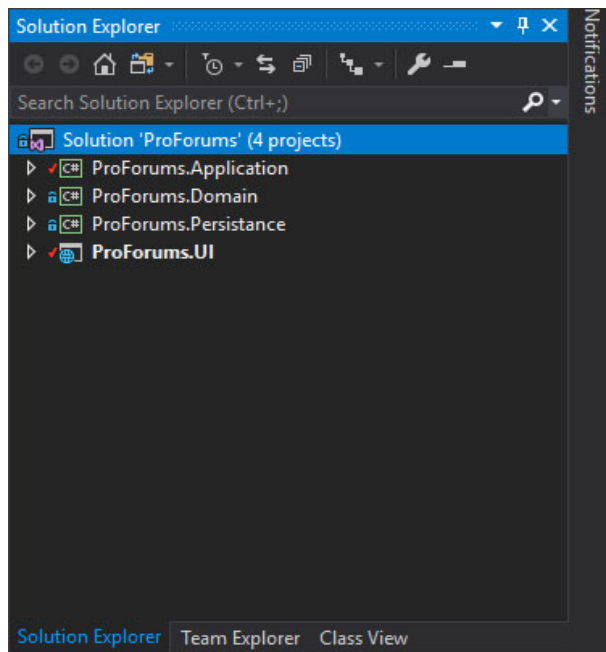


Figura 5 Arhitectura aplicației

### Bibliografie:

1. Freeman A, Pro ASP.NET Core MVC 2, 2011, Apress, London.
2. Martin C. R., Clean Architecture, 2018, Pearson Education
3. [https:// docs.microsoft.com/en-us/aspnet/core /](https://docs.microsoft.com/en-us/aspnet/core/)

# STABILIREA CRITERILOR OBIECTIVE DE EVALUAREA SITE-URILOR WEB CONSTRUIE DE ELEVI

**Autorii: Dan-Dumitru MANU<sup>1</sup> ȘI Darius MOLDOVAN<sup>2</sup>**

[raiku.doku@yahoo.com](mailto:raiku.doku@yahoo.com)

**Coordonator:**

**Sef. lucr. dr.ing. Simona Mirela RIUREAN<sup>3</sup>**

<sup>1</sup>Universitatea din Petroșani, Facultatea de Inginerie Mecanică și Electrică Specializarea Calculatoare anul IV

<sup>2</sup>Universitatea din Petroșani, Facultatea de Inginerie Mecanică și Electrică Specializarea Calculatoare anul II

<sup>3</sup>Universitatea din Petroșani, Facultatea de Inginerie Mecanică și Electrică, Departamentul ACIEE

## Rezumat

To teach someone but even more IMPORTANT, to evaluate a student, are both art and science. In this paper you will find some items for student's evaluation. In the first part we present specific models structures of evaluation according to their level of knowledge used in a test construction followed by a short presentation of a subject web. In the second part, the principles mentioned are applied to evaluate the student's final results at the web presented at college level. There are a lot of examples described in different papers, some of them mostly describe the issue at the theoretical level and a few make a review on web pages created by students.

## Cuvinte cheie

Web , teaching, evaluation.

## 1.Principile fundamentale pentru evaluarea adecvată a unui test

Educația următoarelor generații, învățământul în general, reprezintă cel mai important și sensibil domeniu. Evaluarea elevilor devine o problemă importantă dacă ținem cont de faptul că planurile trebuie să conțină toate elementele necesare, astfel încât, să le dea elevilor "unelte" necesare pentru a fi buni profesioniști și în special practicieni. Atât profesorii cât și elevii sunt interesați în realizarea primelor obiective bazate pe criterii uniforme, corecte și aplicate în practica din licee [1].

În perspectiva unui parcurs profesional ulterior cu alegeri multiple, diverse, elevii trebuie să fie pregătiți să facă față și să treacă cu bine testele ce au ca scop evaluarea în mod corect a cunoștințelor acestora. În procesul de învățare, predare și evaluare aceste elemente importante sunt strâns legate. Orice schimbare produsă la una dintre aceste activități le afectează pe celelalte și se generează o reacție în lanț. Prin urmare, predarea și evaluarea didactică trebuie concepute simultan [2].

Evaluarea, este o activitate importantă în sine, care constă în trei etape principale:

- Măsurarea rezultatelor educaționale prin intermediul unor activități specifice, proceduri, folosind instrumentele adecvate (teste scrise, teste orale, proiecte practice și portofolii);
- Evaluarea rezultatelor pe baza criteriilor uniforme;
- Formularea concluziilor trase din interpretarea rezultatele obținute pentru o decizie educațională adecvată.

Teoria și practica pedagogică operează cu mai multe tipuri de evaluări [3]. Ținând cont de scopul predării, evaluările pot fi de trei tipuri:

- Evaluare inițială: se face pentru a începe un nou ciclu de învățare;
- Evaluarea formală: învățământul însoțește toate etapele, se realizează prin examinarea sistematică a tuturor elevilor la toate disciplinele studiate;
- Evaluarea sumativă: se face, de obicei, la sfârșitul anului sau după o perioadă de formare mai lungă.

În ceea ce privește metodele de evaluare, între teorie și practică există diferențe enorme între metodele de evaluare românești tradiționale și cele noi.

Metodele tradiționale de evaluare sunt cele mai cunoscute, fiind încă cele mai utilizate pe scară largă. Acestea se referă la probe orale, teste scrise și teste practice. Aceste metode tradiționale sunt bine cunoscute și aplicate de toți profesorii. Din păcate, toate se bazează pe evaluarea subiectelor memorate de către elevii, care nu se referă la interpretarea subiectului, capacitatea de înțelegere sau creativitatea elevilor.

Cu toate acestea, noile metode de evaluare, au un mare potențialul formativ care sprijină actul de învățământ prin sprijinirea educației elevilor, dar sunt mai puțin cunoscute. Acestea se referă la elemente specifice de observarea muncii și comportamentului elevului, investigarea, proiect, portofoliul și autoevaluarea [4].

Astăzi, majoritatea testelor de evaluare sunt efectuate de către aplicații specifice, PC-ul fiind instrumentul principal de evaluare în liceele din România sau în timpul unui examen. Testele de evaluare dezvoltate cu sprijinul software-ului multimedia specializat fiind capabil să evalueze cunoștințele elevilor. Acest software trebuie să fie dezvoltat de profesori care au cel puțin cunoștințe minime despre tehnologia informației și a comunicării [5].

Principalele calități ale unui instrument de evaluare sunt: valabilitatea, fiabilitate, obiectivitate și aplicabilitate.

Mai mult, pentru a fi considerat valabil, trebuie să fie testat să acopere întregul conținut al curriculumului de examen într-un mod adecvat, luând în considerare lungimea, dificultatea sau importanța diferitelor aspecte ale conținutului acoperit.

## 2. Curricula la nivel de liceului pentru construirea unui site WEB

Datorită creșterii exponențiale a Internetului, transferul de informații rapid atât în rândul companiilor, cât și între o firmă și clienții acesteia, aplicațiile web se răspândesc destul de mult pe o arie ce acoperă un număr uriaș de persoane. Astăzi, comunicarea online este una dintre activitățile cheie de afaceri oferite prin Internet.

În România programarea web se face abea în clasa a XII-a. Curricula de programare web conține următoarele:

- Principii generale ale proiectării interfețelor Web
  - Etapele procesului de dezvoltare a unei aplicații Web
  - Aspecte generale ale proiectării interfețelor Web
  - Realizarea interfețelor Web utilizând limbajul de marcare HTML (elemente avansate): tabele, formulare, cadre, layer-e
  - Foi de stiluri (CSS)
- Modelul client-server. Protocele de comunicație
- Mediul de lucru (server web - Apache, IIS etc., instrumente de dezvoltare a aplicațiilor - PHPdev etc.)
- Prezentarea unui limbaj de scripting server-side (PHP, ASP etc.)
  - Elemente de bază ale limbajului
  - Instrucțiuni
  - Funcții
  - Structuri de date
- Interacțiunea cu baze de date Web (MySQL, SQL Server etc.)
  - Aplicații pentru definirea și gestionarea unei baze de date
  - Conectare/deconectare la baza de date
  - Transmiterea interogărilor SQL către baza de date. Preluarea și prelucrarea datelor returnate de interogările SQL
- Proiectarea și realizarea unei aplicații Web [6]

## 3. Exemple de teste de evaluare la crearea paginilor WEB

Tema I - total 36 de puncte.

18 întrebări grilă, cu cinci opțiuni de răspunsuri din care există o singură opțiune corectă.

Pentru fiecare articol sunt date două puncte.

Exemplu:

Din ce este alcătuită o pagină web?

- a. două părți: antet(head) și corp(body);
- b. una sau mai multe diviziuni;
- c. imagini, sunete, animații;
- d. 1 frameset sau mai multe frameset-uri;
- e. 1 frameset principal și un număr variabil de frameset-uri care sunt cuprinse în frameset-ul principal.

Tema II - Total 24 de puncte.

16 întrebări și fraze adevărate/false din care:

8 întrebări grilă, cu cinci opțiuni de răspunsuri unde există mai multe (sau toate) opțiunile sunt corecte.

Pentru fiecare articol sunt date două puncte.

Exemplu:

Cum poate fi modificat aspectul unei pagini web?

- a. cu ajutorul unuia sau mai multor fișier CSS;
- b. se poate specifica pentru fiecare diviziune în parte detalii legate de aspect în pagina html;
- c. se poate crea o zonă specială în zona head a pagini html denumită și style care conține detalii legate de aspectul pagini;
- d. se poate folosi toate cele trei variante a, b și c în același timp;
- e. oricare din variantele a, b și c dar nu simultan.

Notă: Întrebarea de mai sus are 4 răspunsuri corecte. Doar dacă este dat răspunsul complet (toate cele patru marcate) vor fi evaluate exercițiul ca fiind corect, având în vedere cele două puncte.

8 fraze duble (Adevărat / Fals), câte 1 punct fiecare.

Exemplu:

Linkurile în paginile html pot să fie puse sub forma de: buton, text, și imagine. (adevarat/fals)

Tema III - În total 30 de puncte.

Elevii trebuie să-și creeze propriul site web.

Evaluarea acestei teme nu este o sarcină ușoară. De aceea, puțini indicatorii importanți trebuie subliniați și marcați ca adecvați pentru evaluare, ținând seama de aspectele comune pentru comparare ca în figura nr. 1

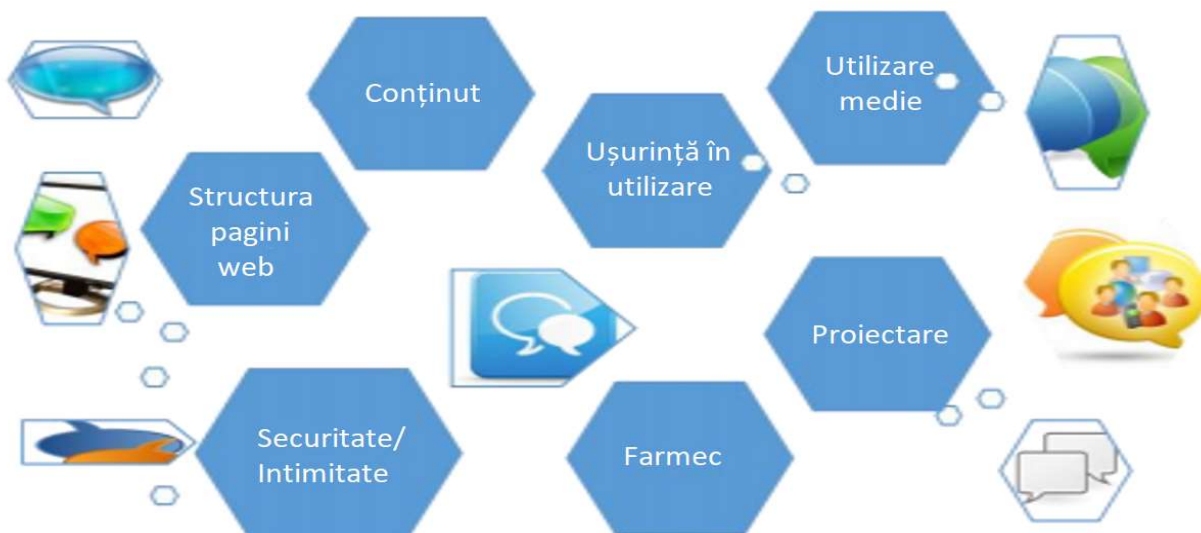


Fig. nr.1 Trăsături comune de design pentru un site web

Funcțiile comune de design constau în câteva marcaje specificate pentru o evaluare adecvată [7].

a) 8 puncte pentru conținutul cu:

- conținut util și suficient;
- complexitate;
- precizie;
- conținut adecvat pentru grupurile țintă vizate.

b) 8 puncte pentru design cu:

- font, stil, aspectul paginii, culoarea, consistența designului.

c) 4 puncte pentru structura sitului

- organizarea de informații, sprijin pentru navigație.

d) 4 puncte pentru securitate / intimitate

- nici o încălcare a dreptului individual;
- protecția împotriva actelor criminale.

e) 3 puncte pentru ușurința în utilizare

- conștientizarea nevoilor populației afectate;
- respectarea versiunilor de browser;
- platforme hardware, viteza rețelei

f) 3 puncte pentru utilizare medie

- utilizarea adecvată a interactivității;
- mișcare multimedia;
- program de aplicație, sunet, video.

g) 2 puncte pentru farmec

- dispoziții pentru a face utilizatorii să se simtă confortabil.

Au fost multe încercări de a sublinia propriu-zis evaluarea conform unor construcții teoretice riguroase și date empirice. De exemplu, Zhang și von Dran au susținut teoria motivării lui Herzberg [8] Jung și Butler au subliniat 15 orientări importante în [9] în timp ce [10] au fost descrise 100 și chiar mai mult (600) au fost descrise în [11].

Cu toate acestea, indiferent cât de general sau cât de adânc este subiectul împărțit în cât mai multe bucăți, este responsabilitatea profesorului pentru a alege întrebarea potrivită pentru cele mai importante subiecte predate la programarea web. Ne referim aici la subiectul semnificativ având aplicații practice. De exemplu, este mai important să



cunoască de inima situurilor, diverse liste de caracteristici sau avantaje decât să fie capabil să explice diferența dintre http și https.

Cu toate acestea, există multe considerații cu privire la un aspect adecvat evaluării unei bune aplicații web proiectate de elevii.

În primul rând, elevii trebuie să fie încurajați să-și lase liberă imaginația și să construiască site-ul web în funcție de propriile lor preferințe. Indiferent dacă sunt alese servicii sau produse pe platformă, criteriile de evaluare a proiectului lor trebuie să fie explicate corespunzător înainte de începerea proiectului. Această concluzie se bazează pe percepția utilității și ușurința utilizării percepute. Percepția utilității se referă la gradul în care un individ crede că utilizarea unui anumit sistem ar îmbunătăți performanța locului de muncă și, pe de altă parte, ușurința utilizării percepute afirmă gradul în care o persoană consideră că folosirea unei metode a sistemul particular ar fi lipsit de efort fizic și mental [12].

Astfel, pilonii de ghidare trebuie marcați corespunzător de la început și taskurile de evaluare devin ușoare pentru profesor și mai aproape de posibilitatea de obiectivitate necesară.

#### 4. Concluzii

Predarea implică o mare responsabilitate luând în considerare că fiecare elev trebuie să descopere cel mai bun "mod profesionist" și să-și atingă scopul profesional după finalizarea educației. Pentru a estima cu exactitate informații și instrumente care au fost oferite viitorilor profesioniști, evaluarea adecvată este un pas foarte important care trebuie luat în considerare. De aceea, această etapă este importantă și profesorii trebuie să acorde importanța corespunzătoare testelor pe care le fac cu scopul de a evalua elevii cu un grad ridicat de cunoștințe. Testele finale trebuie să conțină cât mai multe tipuri posibile de articole în scopul de a avea o bună imagine despre achiziția finală și abilitățile practice dobândite de către absolvenți.

În acest scop, această lucrare prezintă câteva exemple posibile de evaluare obiectivă și câteva întrebări legate de subiect în conformitate cu curricula paginilor web, împreună cu punctele maxime care pot fi acordate pentru răspunsurile corecte. O evaluare completă corectă, alături de întrebări, constă în evaluarea unei aplicații web construite de elevii. Conținutul și designul ar trebui să fie obiectivele cele mai importante pe care elevii să le atingă în proiectul aplicației web, urmat de structura site-ului, securitate, ușurința în utilizare, utilizare medie și nu în ultimul rând, farmec. Toate aceste caracteristici menționate ale site-ului sunt acordate treptat cu puncte în funcție de importanța lor propusă în această lucrare. Astfel, datorită creșterii exponențiale și a evoluției rapide a percepției unui utilizator pe internet precum și progresul tehnologic, criteriile menționate aici, pentru o evaluare obiectivă și adecvată a unei aplicații web concepute de elevii, nu sunt definitive, fiind subiectul îmbunătățirii continue.

#### 5. Bibliografie

- [1]. Ionescu C., "Metodica predării informaticii", Universitatea "Babeș- Bolyai" Cluj 1999
- [2]. Masalagiu C., I. Asiminoaei, "Didactica predării informaticii", Editura Polirom, 2004.
- [3]. Nicola I., "Pedagogie", Editura Didactica și Pedagogica București, 1994
- [4]. Masalagiu C, Asiminoaei, I. Maxim, "Metodica predării informaticii" Editura MatrixRom, București, 2001.
- [5]. Petre C., Popa, D., et al. – Metodica predării Informaticii și Tehnologiei Informației. Editura Arves, Craiova 2002
- [6] \*\*\*<http://oldsite.edu.ro/index.php/articles/12808>
- [11] Auer, N. J. (2003). "Bibliography on evaluating web information". Retrieved October 20, 2003 from <http://www.lib.vt.edu/research/evaluate/evalbiblio.html>.
- [7] Yoshida, T., Ariga, T. (1998). "Chekkurisuto wo tsukatte Web pe-ji no hihanteki hyouka to sono kouka [A critical evaluation checklist for web pages and its effectiveness]" Proceedings of the 56th conference of the Information Processing Society of Japan (4) (pp. 260–261).
- [13] Zhang, P., & von Dran, G. M. (2000). Satisfiers and dissatisfiers: A twofactor model for website design and evaluation. Journal of the American Society for Information Science, 51(14), 1253–1268.
- [8] Zhang, Ping, Gisela von Dran, Ruth Small, Silvia Barcellos (2000), A Two-Factor Theory for Website Design, Proceedings of the Hawaii International Conference on Systems Science (HICSS 33), Hawaii, January, 2000.
- [9] Jung Timothy H, Butler Richard (2000) The measurement of the marketing effectiveness of the Internet in the tourism and hospitality industry Information and Communication Technologies in Tourism, pag. 460-472, Publisher Springer, Vienna
- [10] Ivory Melody, Mankoff Jennifer (2003) Using Automated Tools To Improve Web Site Usage By Users With Diverse Abilities IT&SOCIETY, VOLUME 1, ISSUE 3, PP. TK-TK <http://www.ITandSociety.org>
- [11] Carton I (2002) The 160 best practices of the top eRetail sites. Lessons to be learnt for your own site. eShopability Newsletter.
- [12] Davis, F. D. (1989). Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology, MIS Quarterly, 13,319–340.