**Universitatea Poșitehnică Timișoara**

**Facultatea de automatică și Calculatoare**

**Ușa automată cu interfon**

Echipa:

*Mihăilescu Flavius 4.1*

Profesor:

*Korodi Adrian*

1. **Documentare.**

**1.1.Utilitate.**

Cand vorbim de usi automate, ne gandim la siguranta si confort deplin.

Usile automate reprezinta imbinarea armonioasa a pieselor mecanice cu inteligenta componentelor electronice. Mecanismele de acest tip ofera atat confortul prin faptul ca nu mai este necesara actionarea fizica pentru deschiderea, respective inchiderea usii. ne putem da seama ce avantaj se poate oferii persoanelor cu dizabilitati, care sunt astfel scutite de un efort considerabil.

Daca ne refreim la cea de-a doua parte a proiectului si anume interfonul, acesta confera o metoda exrem de efecienta pentru a permitelor persoanelor ce nu au fara acces, in principiu celor care nu sunt locatari (vizitatorilor), sa intre in bloc.

**1.2.Securitate**

In acest domeniu al securitatii lucrurile sunt la fel de clare, deoarece un sistem care nu permite accesul persoanelor din exterior previne invatia persoanelor straine sau nadorite.

De asemenea sistemul automat al usii este mult mai usor de accesat de persoanele cu acces permanent(locatarii), iar dupa inchiderea usii aceasta se blocheaza automat, fara sa ne facem griji daca am inchis sau nu usa cu cheia sau alte metode invechite.

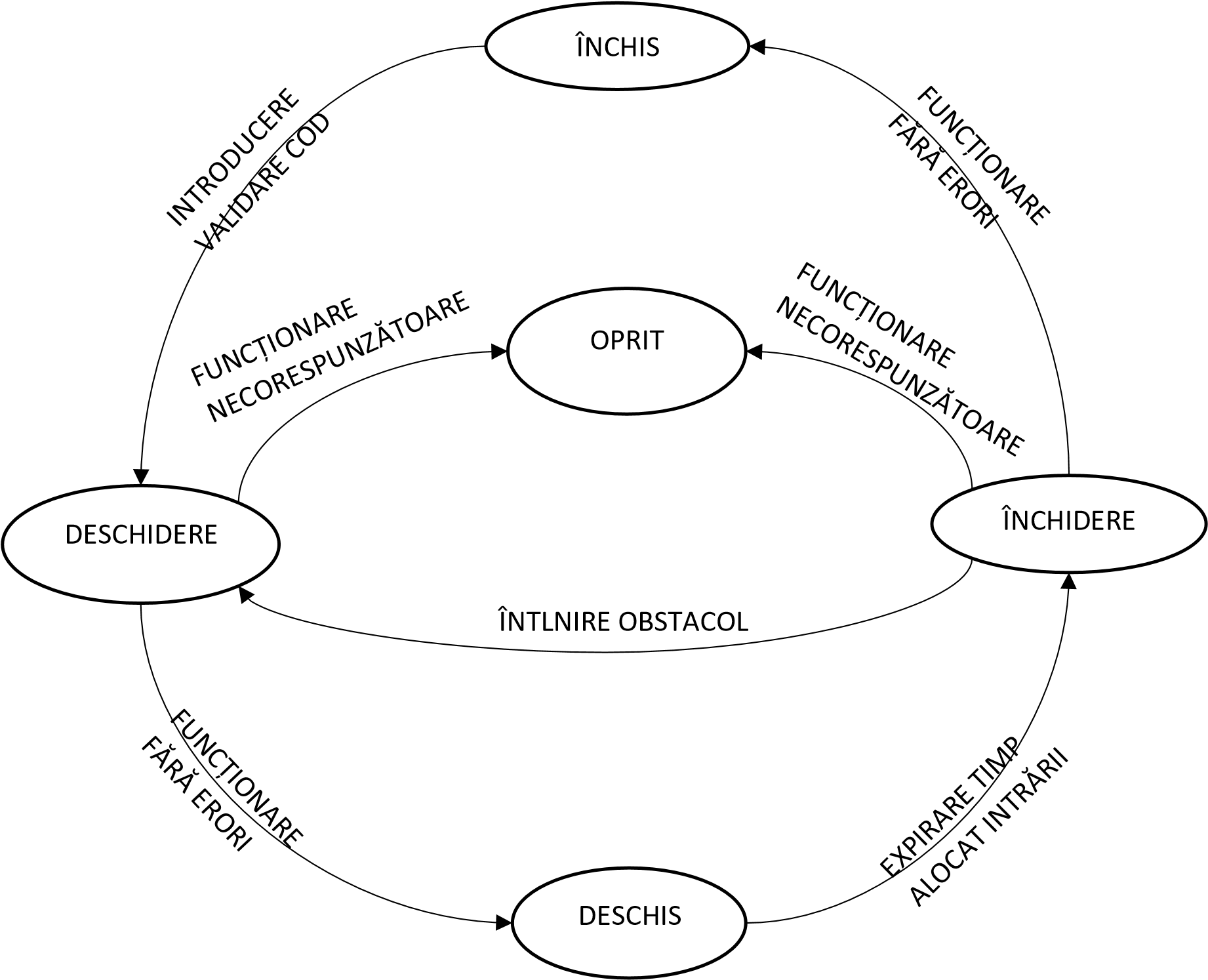
**1.3.Implementari similare.**

Proiectul nostru este gandit pentru un complex de apartamente, insa acest concept automatic se poate implementa cu adaptari in functie de nevoi in diferite domenii.

Un prim domeniu implementabil pentru acest sistem ar fi in cazul usilor de intrare in camere, interfonul fiind inlocuit cu o cartela de acces, amprenta sau orice alta metoda moderna de acces. Utilitatiile si securitatea fiind de semenea sporite. Bancomatele inchise pot beneficia si ele de un astfel de sistem pentru a asigura clientiilor securitate sporita. In acest caz m-am gandit la cardurile bancare, care ar putea avea integrate un sistem de tipul cartelelor de acces, astfel sa nu poti intra la bancomat fara cardul bancar.

Un alt sistem care ar fi excelent de implementat este accesul animalelor de companie. Cu o mini-usa automata si un senzor de prezenta putem oferii o confortul uneii casute izolate in care frigul nu o sa mai patrunda pentru cainele nostru.

1. **Diagrama de stare.**



**3.Schema logica.**

ÎNCHIDERE UȘA



DA



DESCHIDERE UȘA



ASTEPTARE INTRARE



ÎNCHIDERE FĂRĂ ERORI



BLOCARE UȘA



STOP



DA



NU

AFIȘARE MESAJ

“NU RĂSPUNDE”

PE LCD

NU

AȘTEPTARE

REAPELARE

RE

APELARE

LA TIMP

START

INTRODUCERE COD CU AFIȘARE ÎN TIMP REAL PE LCD

APĂSARE BUTON

VALIDARE

ȘI

TRANSMITERE

VALIDAR

E

COD

TRANSMITERE SEMNAL SPRE

APARTAMENT

AFIȘARE MESAJ EROARE

NU

DA

A

Ș

TEPTARE RĂSPUNS

PRIMIRE RĂSPUNS

DA

PERMITERE

ACCES

DA

AFI

Ș

ARE MESAJ

“FĂRĂ ACCES”

NU

NU

**4.Selectie componente. Mod de lucru componente.**

Selectie componente

Componente montaj fizic:

* Material lemnos
* Motor DC
* Roti dintate
* Roata de tip scripete
* Mini-curea de transmisie

Componente automatizare:

* Arduino MEGA 2560 R3 (ATmega2560 + ATmega16u2)
* Punte H l298
* Senzor PIR
* LCD 1602 cu Interfata I2C si iluminare de fundal Albastru
* Modul Tastatură
* Butoane Push cu revenire
* Conectori de tip mama-tata/mama-mama/tata-tata

Componenete ansamblare:

* Pistol de lipit
* Letcon
* Fierastrau circular

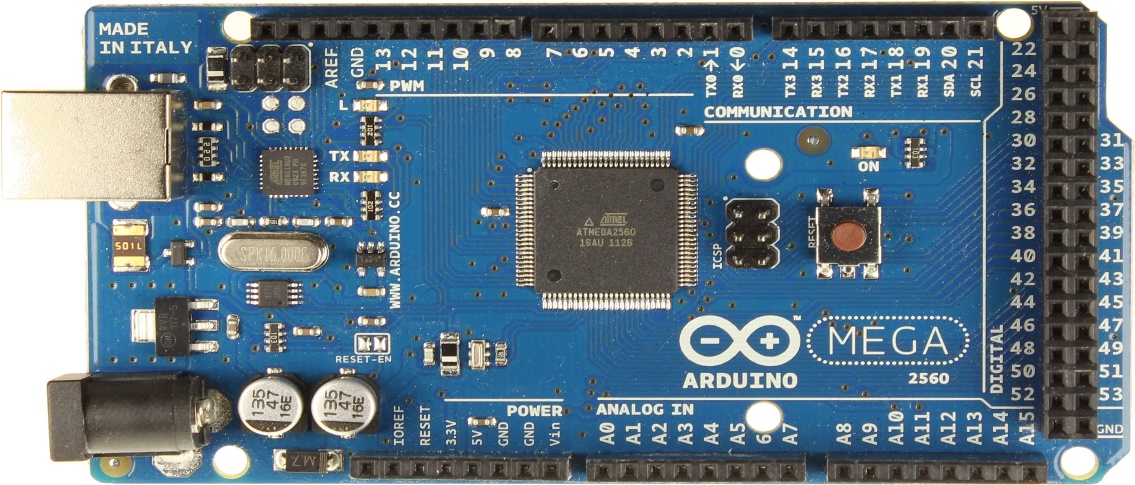
Plus alte componente marunte…

Mod de lucru componente

* Arduino MEGA 2560 R3 Port Mapping: Arduino Mega2560

Caracteristici tehnice:

* Arduino MEGA 2560 R3 (ATmega2560 + ATmega16u2)



[This Photo](https://www.htxt.co.za/2013/10/09/how-to-arduino-class-kicking-off-in-cape-town/) by Unknown Author is licensed under [CC BY-SA-NC](https://creativecommons.org/licenses/by-nc-sa/3.0/)

* Tensiune de functionare: 5V;
* Tensiune de alimentare Jack: 7V - 12V;
* Pini I/O: 54;
* Pini PWM: 15 (din cei de I/O);
* Pini analogici: 16;
* 4 x UART;
* Memorie flash: 256KB, din care 8KB ocupati de bootloader;
* Frecventa de functionare: 16MHz.
* Dimensiuni: 5.3cm x 10cm.

Pin Port Mapping

0 PORTE 0 PORTE=0x01

1 PORTE 1 PORTE=0x02

2 PORTE 4 PORTE=0x04

3 PORTE 5 PORTE=0X08

4 PORTG 5 PORTG=0X20

5 PORTE 3 PORTE=0X08

6 PORTH 3 PORTH=0X08

7 PORTH 4 PORTH=0X10

8 PORTH 5 PORTH=0X20

9 PORTH 6 PORTH=0X40

10 PORTB 4 PORTB=0X10

11 PORTB 5 PORTB=0X20

12 PORTB 6 PORTB=0X40

13 PORTB 7 PORTB=0X80

14 PORTJ 1 PORTJ=0X02

15 PORTJ 0 PORTJ=0X01

16 PORTH 1 PORTH=0X02

17 PORTH 0 PORTH=0X01

18 PORTD 3 PORTD=0X08

19 PORTD 2 PORTD=0X04

20 PORTD 1 PORTD=0X02

21 PORTD 0 PORTD=0X01

22 PORTA 0 PORTA=0X01

23 PORTA 1 PORTA=0X02

24 PORTA 2 PORTA=0X04

25 PORTA 3 PORTA=0X08

26 PORTA 4 PORTA=0X10

27 PORTA 5 PORTA=0X20

28 PORTA 6 PORTA=0X40

29 PORTA 7 PORTA=0X80

30 PORTC 7 PORTC=0X80

31 PORTC 6 PORTC=0X40

32 PORTC 5 PORTC=0X20

33 PORTC 4 PORTC=0X10

34 PORTC 3 PORTC=0X08

35 PORTC 2 PORTC=0X04

36 PORTC 1 PORTC=0X02

37 PORTC 0 PORTC=0X01

38 PORTD 7 PORTD=0X80

39 PORTG 2 PORTG=0X04

40 PORTG 1 PORTG=0X02

41 PORTG 0 PORTG=0X01

42 PORTL 7 PORTL=0X80

43 PORTL 6 PORTL=0X40

44 PORTL 5 PORTL=0X20

45 PORTL 4 PORTL=0X10

46 PORTL 3 PORTL=0X08

47 PORTL 2 PORTL=0X04

48 PORTL 1 PORTL=0X02

49 PORTL 0 PORTL=0X01

50 PORTB 3 PORTB=0X08

51 PORTB 2 PORTB=0X04

52 PORTB 1 PORTB=0X02

53 PORTB 0 PORTB=0X01

A0 PORTF 0 PORTF=0X01

A1 PORTF 1 PORTF=0X02

A2 PORTF 2 PORTF=0X03

A3 PORTF 3 PORTF=0X04

A4 PORTF 4 PORTF=0X10

A5 PORTF 5 PORTF=0X20

A6 PORTF 6 PORTF=0X30

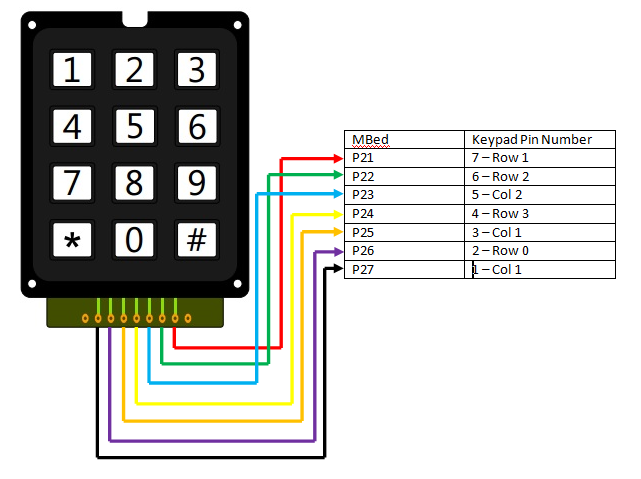
A7 PORTF 7 PORTF=0X40

* Tastatura

Caracteristici tehnice:

* + Contact rating:20mA,24VDC
  + Contact resistance:200 ohm max
  + Life:1,000,000 cycles per key
  + Operating Temperature: -20­ to +60­
  + Storage Temperature: -40­ to +65­

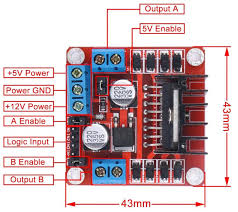
Conectare cu microcontroler:



* Motor DC

Caracteristici tehnice:

* Tensiune de functionare:
  + In functie de tensiunea de alimentare se poate controla viteza motorului
* In functie de modul de alimentare se poate controla sensul de rotatie al motorului
* Punte H L298n



* LCD

LCD-ul poate sa afiseza 32  caractere pe 2 randuri .

**Specificatii:**

Controller: HD44780

Dimensiuni: 80mm x 36mm

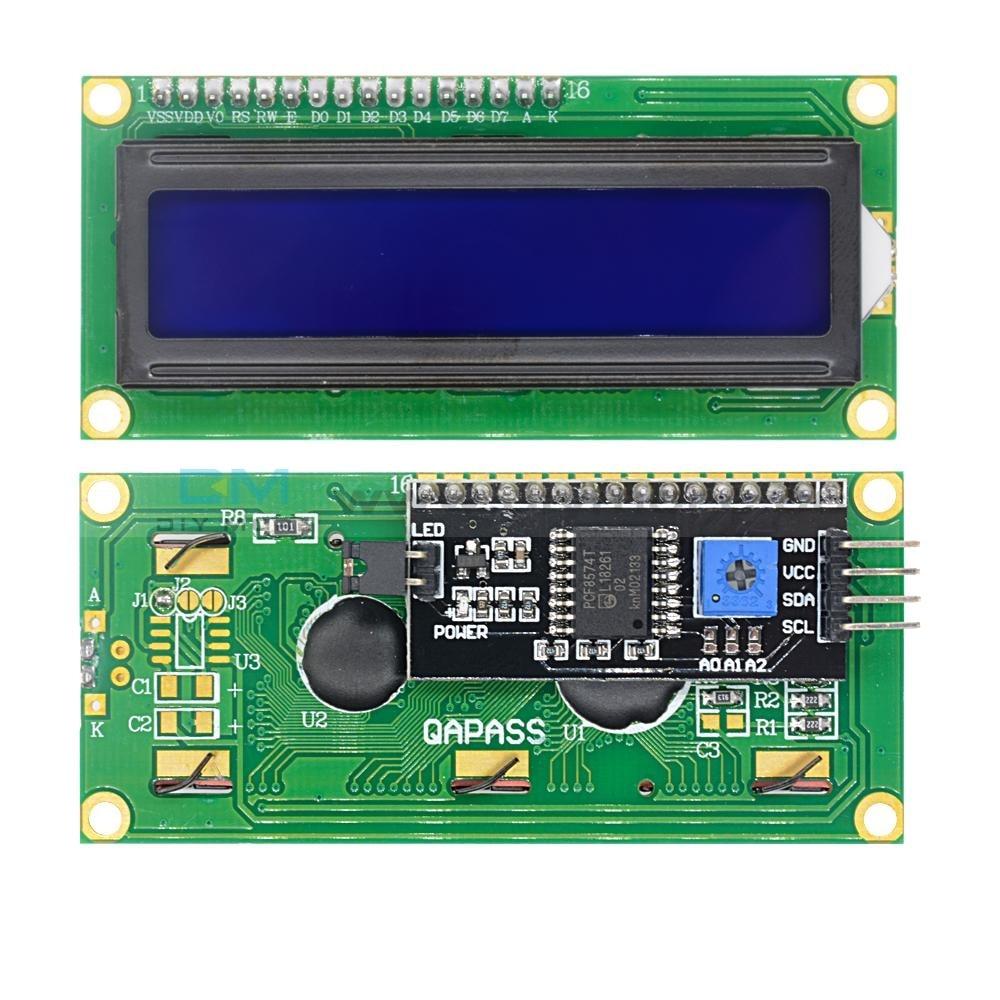
Culoare caractere: galben

Culoare background: albastru

DataSheet: http://www.sparkfun.com/datasheets/LCD/GDM1602K-Extended.pdf

<https://www.sparkfun.com/datasheets/LCD/GDM1602K.pdf>

* LCD 1602 cu Interfata I2C

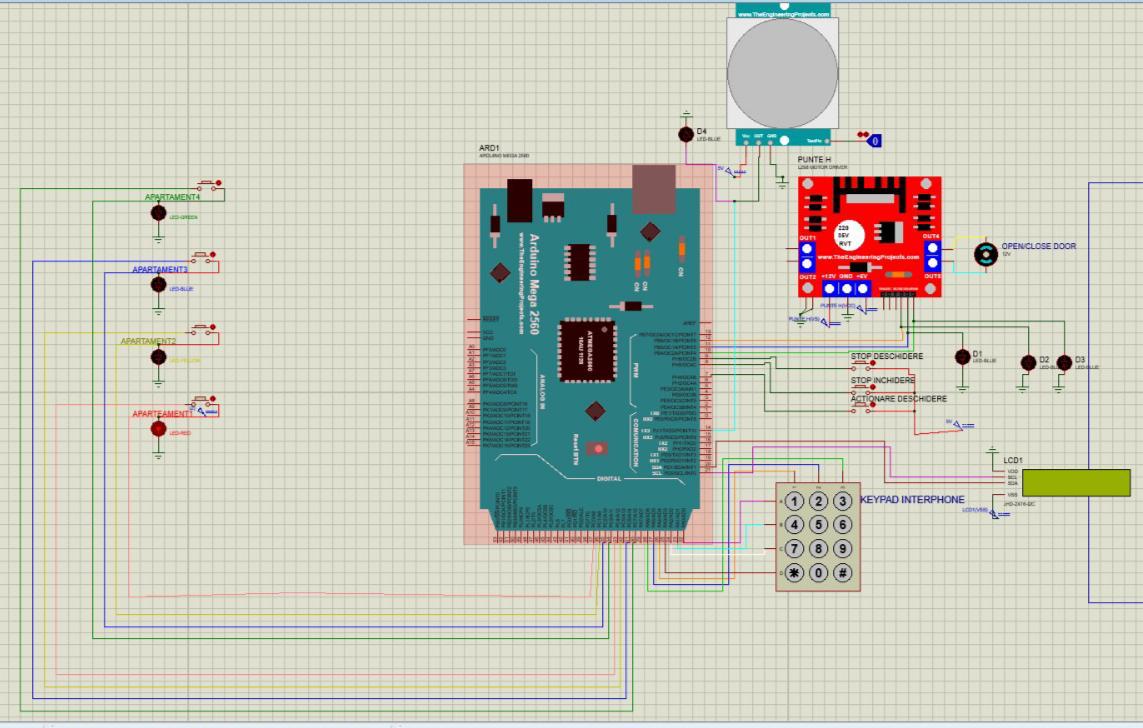


[This Photo](https://www.electronics-lab.com/project/using-16x2-i2c-lcd-display-esp32/) by Unknown Author is licensed under [CC BY-SA](https://creativecommons.org/licenses/by-sa/3.0/)

Caracteristici tehinice:

* 5x8 dots
* Built-in controller (S6A0069 or equivalent)
* +5V power supply
* 1/16 duty cycle
* LED Backlight

**5.Arhitectura HardWare**

****

**6.Implementare SoftWare**

**6.1.Cod Arduino.**

#include <Keypad.h>

#include <LiquidCrystal\_I2C.h>

LiquidCrystal\_I2C lcd(0x27, 16, 2); //initializare LCD

#define pass\_lenght 6 //dimensiunea parolei(ultimul caracter este null)

#define nr\_ap 4 //numarul total al apartamentelor

#define timp\_raspuns 2250 // echivalentul a 9 secunde

int PWM = 60;

char password[pass\_lenght]; //sirul ce va fi format de la tastatura

char\* pass\_ap[nr\_ap] = { "12345", "54321", "13579", "02468"}; //parolele pentru apartamente

int contor\_pass; // folosit pentru numararea apasarii tastelor

int apelari\_totale;

int apelari\_gresite;

int apelari\_incomplete;

int apelari\_corecte[4];

int raspunsuri[4];

int fara\_raspuns[4];

void setup() {

DDRH = 0x60; //butoane oprire deschidere & inchidere

DDRB |= 0x70; //punte H

DDRJ &= ~0X02; //PIR

DDRC |= 0x0F; //LED apartament 1

DDRC |= 0xF0; //Buton apartament 1

init\_timer1();

init\_PWM\_timer2();

Serial.begin(9600);

lcd.begin();

lcd.setCursor(3, 0);

lcd.print("Buna Ziua!");

lcd.setCursor(4, 1);

lcd.print("Parola:");

}

const byte ROWS = 4;

const byte COLS = 3;

char hexaKeys[ROWS][COLS] = {

{'1', '2', '3'},

{'4', '5', '6'},

{'7', '8', '9'},

{'\*', '0', '#'}

};

byte rowPins[ROWS] = {22, 23, 24, 25};//connect to the row pinouts of the keypad

byte colPins[COLS] = {26, 27, 28};//connect to the column pinouts of the keypad

Keypad customKeypad = Keypad(makeKeymap(hexaKeys), rowPins, colPins, ROWS, COLS);//initialize an instance of class NewKeypad

void init\_timer1()

{

cli();

TCCR1A = 0;

TCCR1B = 0;

TCCR1B = (1 << CS10);

sei();

}

void init\_PWM\_timer2()

{

cli();

TCCR2A = 0;

TCCR2B = 0;

TCCR2A = \_BV(COM2A0) | \_BV(COM2B1) | \_BV(WGM20);

TCCR2B = \_BV(WGM22) | \_BV(CS22);

sei();

}

void Asteptare() { // timpul cat usa v-a sta deschisa

int over = 0;

TCNT1 = 0;

while (over <= 2250) {

if (TCNT1 >= 64000) {

TCNT1 = 0;

over++;

}

}

}

int Asteptare\_Raspuns(bool stare\_btn, int contor\_ap) { // timpul in care se poate primi rspuns de la apartament

int over = 0;

TCNT1 = 0;

Serial.print("b=");

Serial.println(stare\_btn);

while ((over <= timp\_raspuns) and stare\_btn == 0) {

if (TCNT1 >= 64000) {

TCNT1 = 0;

over++;

}

stare\_btn = PINC & 0x10;

if (contor\_ap + 1 == 1)

stare\_btn = PINC & 0x10;

if (contor\_ap + 1 == 2)

stare\_btn = PINC & 0x20;

if (contor\_ap + 1 == 3)

stare\_btn = PINC & 0x40;

if (contor\_ap + 1 == 4)

stare\_btn = PINC & 0x80;

}

return over;

}

int i = 0;

void Deschidere(int PWM) {

PORTB = 0X00;

PORTH = 0X00;

while (!(PINH & 0x40)) {

OCR2A = PWM;

PORTB |= 0X40;

}

PORTB = 0X00;

PORTH = 0X00;

}

void Inchidere(int PWM) {

while (!(PINH & 0x20)) {

if (PINJ & 0X02) {

PORTH |= 0X10;

Deschidere(PWM);

Asteptare();

Inchidere(PWM);

}

else {

OCR2A = PWM;

PORTB |= 0X20;

}

}

PORTB = 0X00;

PORTH = 0X00;

}

void Actionare\_Deschidere() {

lcd.clear();

lcd.setCursor(4, 0);

lcd.print("Deschis");

lcd.setCursor(2, 1);

lcd.print("Puteti intra");

Deschidere(PWM);

Asteptare();

Inchidere(PWM);

}

void Nu\_Raspunde() {

lcd.clear();

lcd.setCursor(2, 0);

lcd.print("Nu Raspunde");

lcd.setCursor(2, 1);

lcd.print("Reincercati");

}

int aux = 0; // folosit pentru a verificarea corectitudinii parolei

int contor\_ap;

int ok = 0; // folosit pentru verificarea apasarii tastaturii

void Rulare() { // timpul cat usa v-a sta deschisa

bool stare\_btn;

int over = 0;

TCNT1 = 0;

while (over <= 2250) {

if (TCNT1 >= 64000) {

TCNT1 = 0;

over++;

char customKey = customKeypad.getKey();

if (customKey) {

ok=1;

password[contor\_pass] = customKey;

lcd.clear();

lcd.setCursor(1, 0);

lcd.print("Parola:");

lcd.setCursor(contor\_pass, 1);

lcd.print(password[contor\_pass]);

contor\_pass++;

if (contor\_pass == pass\_lenght - 1) {

ok=0;

for (contor\_ap = 0; contor\_ap < nr\_ap; contor\_ap++) {

if (!(strcmp(password, pass\_ap[contor\_ap]))) {

aux = 1;

break;

}

}

if (aux == 0) {

Inchidere(PWM);

lcd.clear();

lcd.setCursor(1, 0);

lcd.print("Parola gresita");

lcd.setCursor(2, 1);

lcd.print("Reincercati");

Serial.println("Parola gresita");

over = 0; //resetam contorul pentru ca timpul alocat tastarii sa se reseteze

apelari\_gresite++;

}

else {

lcd.clear();

lcd.print("Parola corecta:");

lcd.println(contor\_ap + 1);

lcd.setCursor(3, 1);

lcd.print("Asteptati");

Serial.print("Parola corecta pentru apartamentul: ");

Serial.println(contor\_ap + 1);

apelari\_corecte[contor\_ap]++;

if (contor\_ap + 1 == 1) {

stare\_btn = PINC & 0x10;

PORTC |= 0x01;

if (Asteptare\_Raspuns(stare\_btn, contor\_ap) < timp\_raspuns) {

raspunsuri[contor\_ap]++;

Actionare\_Deschidere();

PORTC &= ~ 0x01;

break;

}

else {

fara\_raspuns[contor\_ap]++;

Nu\_Raspunde();

PORTC &= ~ 0x01;

over = 0; //resetam contorul pentru ca timpul alocat tastarii sa se reseteze

}

}

if (contor\_ap + 1 == 2) {

stare\_btn = PINC & 0x20;

PORTC |= 0x02;

if (Asteptare\_Raspuns(stare\_btn, contor\_ap) < timp\_raspuns) {

raspunsuri[contor\_ap]++;

Actionare\_Deschidere();

PORTC &= ~ 0x02;

break;

}

else {

fara\_raspuns[contor\_ap]++;

Nu\_Raspunde();

over = 0; //resetam contorul pentru ca timpul alocat tastarii sa se reseteze

PORTC &= ~ 0x02;

}

}

if (contor\_ap + 1 == 3) {

stare\_btn = PINC & 0x40;

PORTC |= 0x04;

if (Asteptare\_Raspuns(stare\_btn, contor\_ap) < timp\_raspuns) {

raspunsuri[contor\_ap]++;

Actionare\_Deschidere();

PORTC &= ~ 0x04;

break;

}

else {

fara\_raspuns[contor\_ap]++;

Nu\_Raspunde();

over = 0; //resetam contorul pentru ca timpul alocat tastarii sa se reseteze

PORTC &= ~ 0x04;

}

}

if (contor\_ap + 1 == 4) {

stare\_btn = PINC & 0x80;

PORTC |= 0x08;

if (Asteptare\_Raspuns(stare\_btn, contor\_ap) < timp\_raspuns) {

raspunsuri[contor\_ap]++;

Actionare\_Deschidere();

PORTC &= ~ 0x08;

break;

}

else {

fara\_raspuns[contor\_ap]++;

Nu\_Raspunde();

over = 0; //resetam contorul pentru ca timpul alocat tastarii sa se reseteze

PORTC &= ~ 0x08;

}

}

}

contor\_pass = 0;

aux = 0;

}

}

}

}

aux = 0;

contor\_pass = 0;

lcd.begin();

lcd.setCursor(3, 0);

lcd.print("Buna Ziua!");

lcd.setCursor(4, 1);

lcd.print("Parola:");

/\*in cazul in care se doreste afisarea in Serial monitor a evidentei intrarilor

if(ok==1){

apelari\_incomplete++;

}

ok=0;

Serial.print("Apelari gresite: ");

Serial.println(apelari\_gresite);

Serial.println("");

Serial.print("Apelari incomplete: ");

Serial.println(apelari\_incomplete);

Serial.println("");

for(int j=0;j<nr\_ap;j++){

Serial.print("Apelari corecte pentru apartamentul ");

Serial.print(j+1);

Serial.println(" : ");

Serial.println(apelari\_corecte[j]);

Serial.println();

Serial.print("Raspunsuri pentru apartamentul ");

Serial.print(j+1);

Serial.println(" : ");

Serial.println(raspunsuri[j]);

Serial.println();

Serial.print("Fara raspuns pentru apartamentul ");

Serial.print(j+1);

Serial.println(" : ");

Serial.println(fara\_raspuns[j]);

Serial.println();

}

\*/

}

void loop() {

Rulare();

}

void loop() {

Rulare();

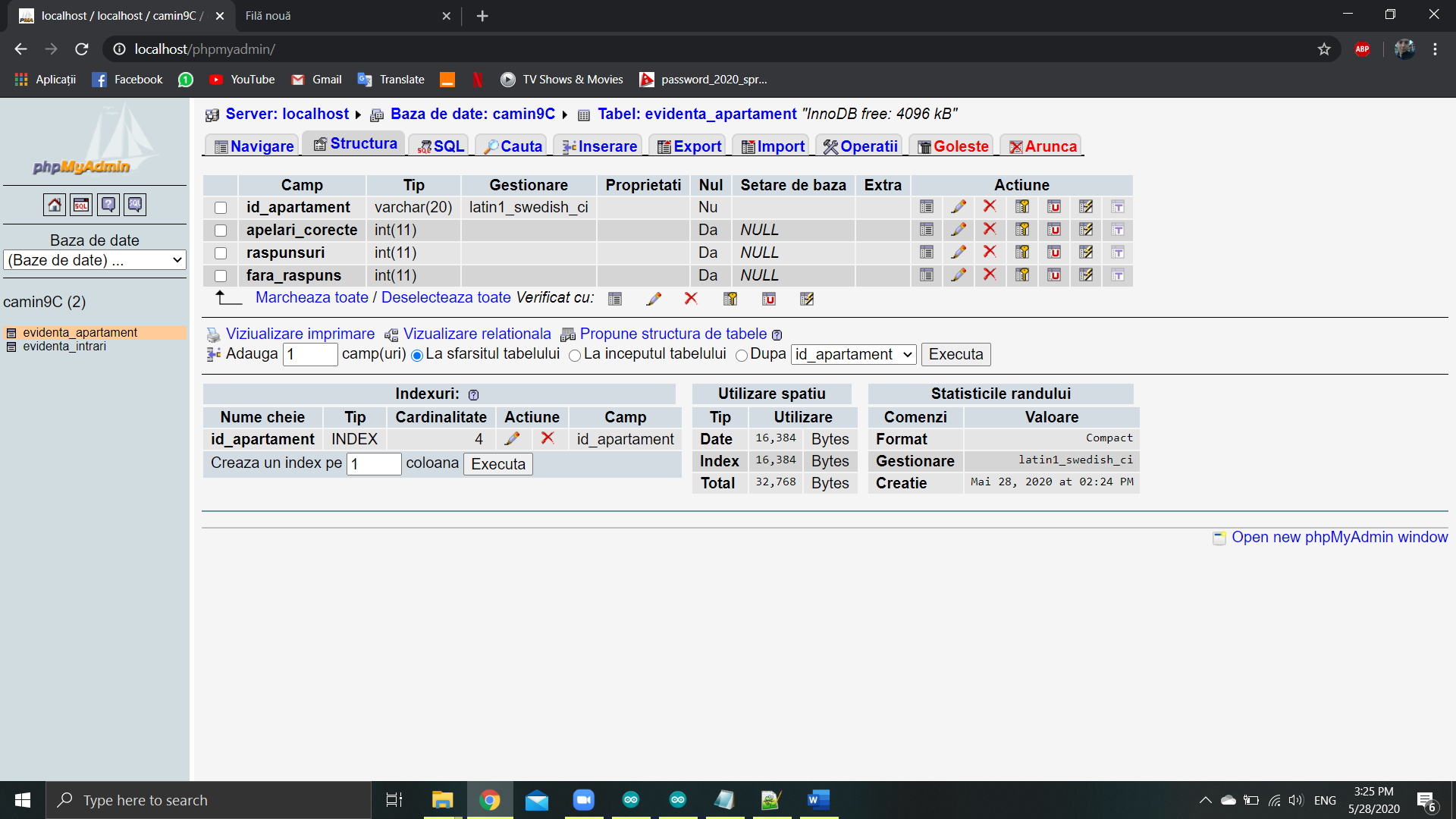
}

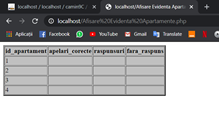
**6.2.Baza de date.** (Task nefinalizat)

Din pacate echipa noastra si-a dat seama prea tarziu ca avem nevoie de un shield ethernet pentru a transmite datele din arduino in baza de date. Astfel nu am reusit sa finalizam taskul. Ce am realizat se poate vedea in continuarea documentatie.

**6.2.1. Tabela “evidenta\_apartament”.**

In aceasta tabela se stocheaza, dupa cum se vede, numarul apartamentului(id\_apartament), Numarul apelarilor corecte(apelari\_corecte), numarul raspunsurilor(raspunsuri) si numarul apelarilor fara raspuns(fara\_raspuns)



****

**6.2.1.1. Codul php pentru afisarea tabelei.**

<html>

<body>

<?php

mysql\_connect("localhost","root","") or die ("Eroare conectare server");

mysql\_select\_db("camin9c") or die ("Eroare BD");

$query=mysql\_query("select \* from evidenta\_apartament");

$nr=mysql\_num\_rows($query);

if($nr>0){

echo "<table border='2' bgcolor='silver'>";

echo "<tr>";

$coln=mysql\_num\_fields($query);

for($i=0;$i<$coln;$i++){

$var=mysql\_field\_name($query,$i);

echo "<th> $var </th>";

}

echo "</tr>";

$nr\_val=0;

while($row=mysql\_fetch\_row($query)){

echo"<tr>";

foreach($row as $value){

echo"<td> $value </td>";

}

echo"</tr>";

}

echo"</table>";

}

else{

die ("nu exista inregistrari in tabela");

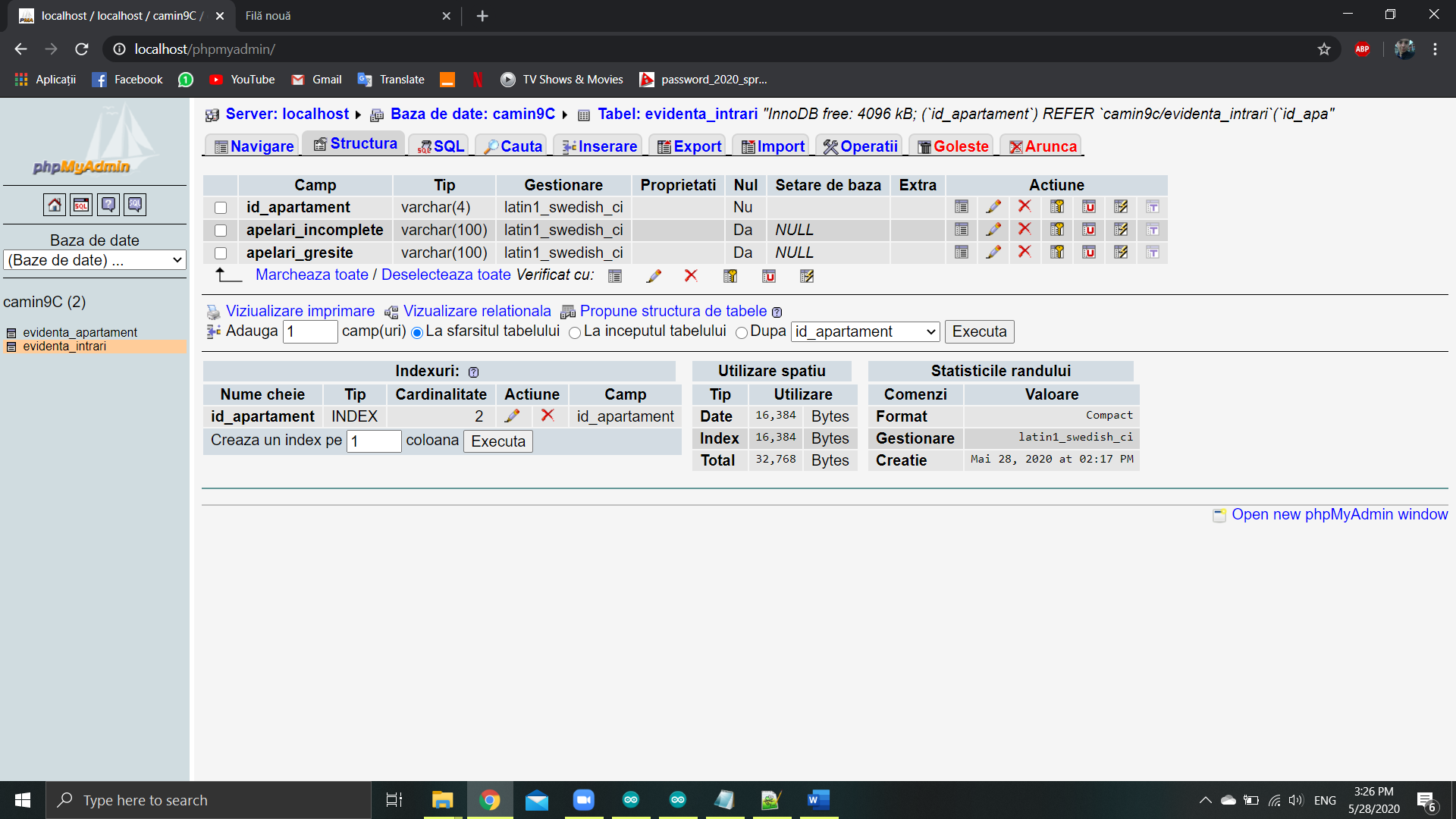
}

?>

</body>

</html>

**6.2.2. Tabela “evidenta\_intrari”.**



**6.2.1.1. Codul php pentru afisarea tabelei.**

<html>

<body>

<?php

mysql\_connect("localhost","root","") or die ("Eroare conectare server");

mysql\_select\_db("camin9c") or die ("Eroare BD");

$query=mysql\_query("select \* from evidenta\_intrari");

$nr=mysql\_num\_rows($query);

if($nr>0){

echo "<table border='2' bgcolor='silver'>";

echo "<tr>";

$coln=mysql\_num\_fields($query);

for($i=0;$i<$coln;$i++){

$var=mysql\_field\_name($query,$i);

echo "<th> $var </th>";

}

echo "</tr>";

$nr\_val=0;

while($row=mysql\_fetch\_row($query)){

echo"<tr>";

foreach($row as $value){

echo"<td> $value </td>";

}

echo"</tr>";

}

echo"</table>";

}

else{

die ("nu exista inregistrari in tabela");

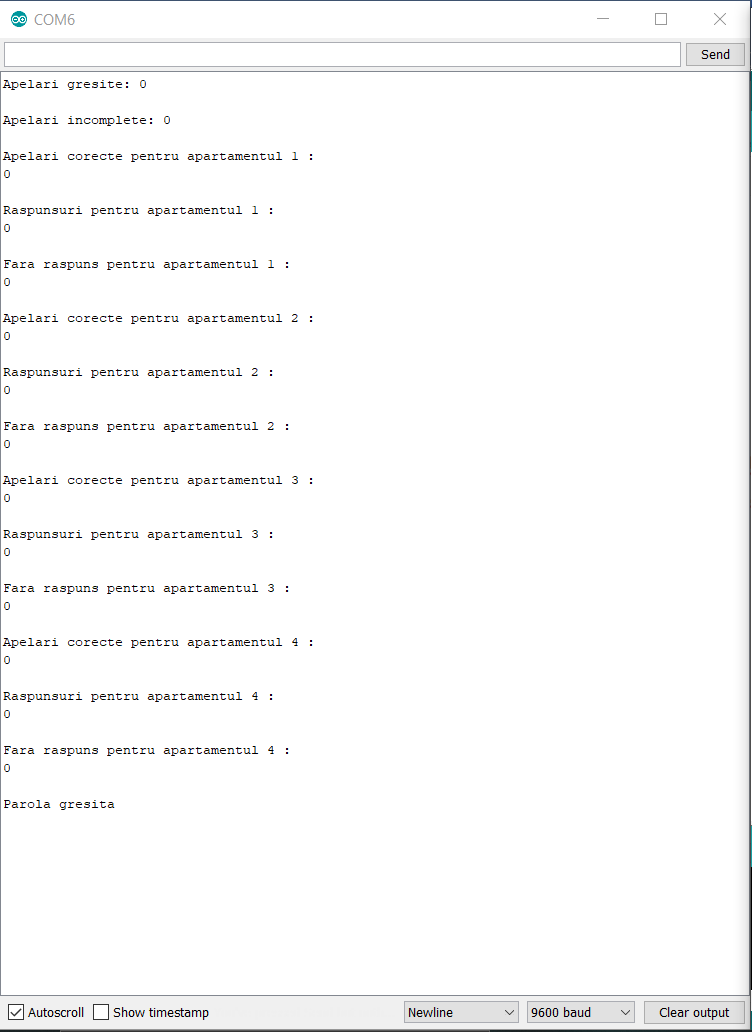
}

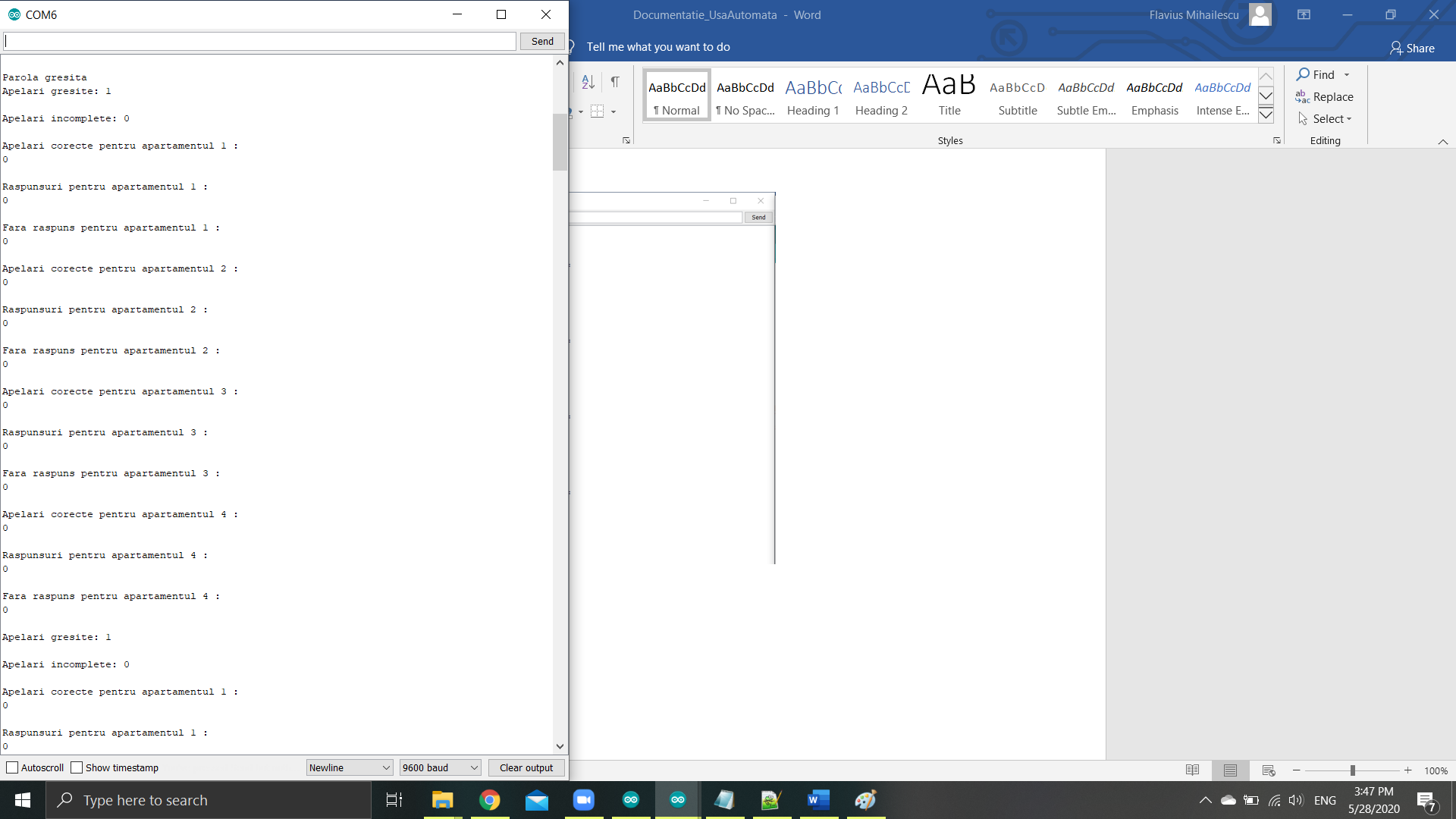
?>

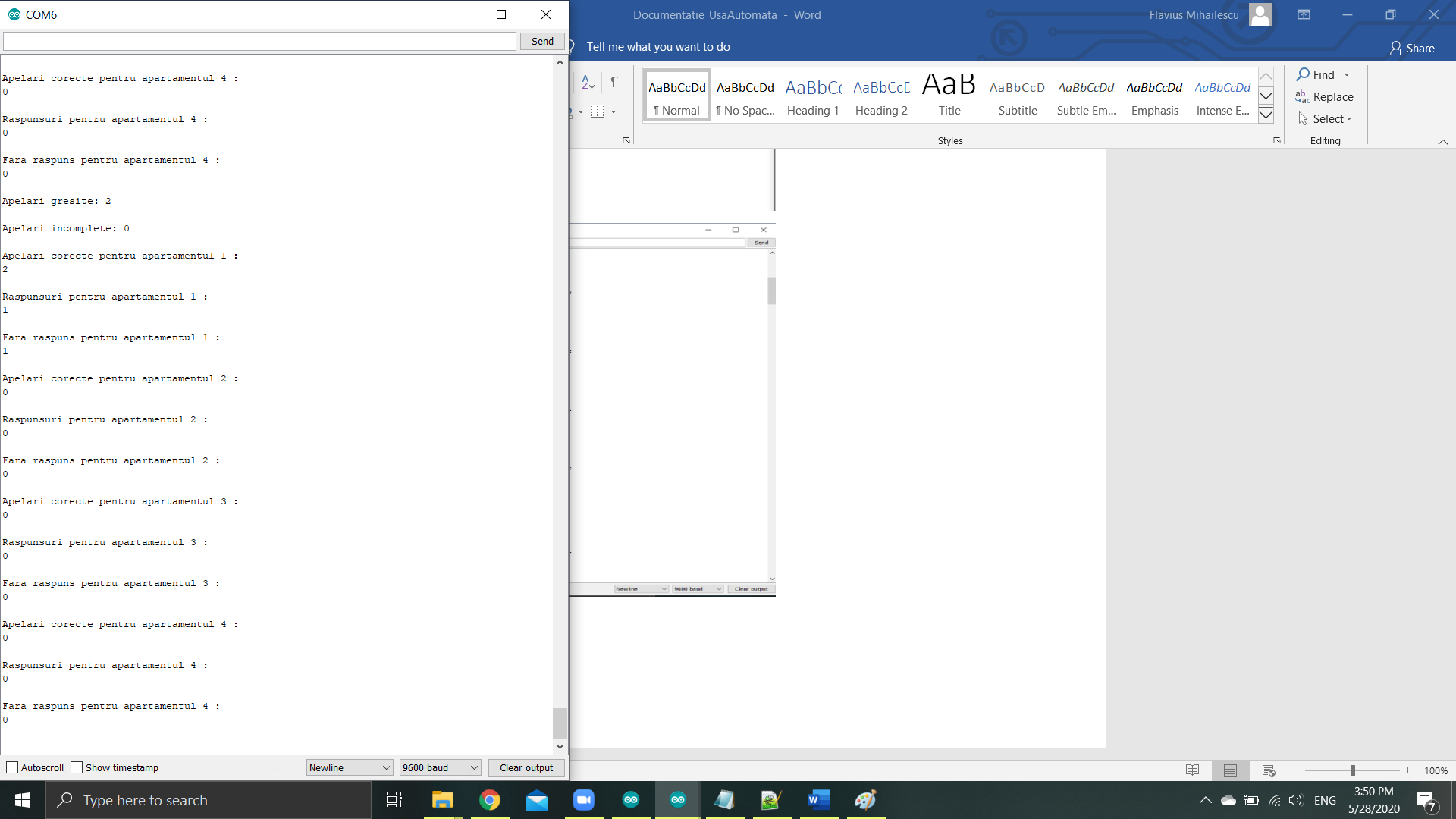
</body>

</html>

**6.3. Prezenetarea in Serial Monitor a informatilor ce trebuie introduse in baza de date .**

****

****

****