Söreg Andra-Marina

A.C., C.T.I.-RO An 4

**Proiect - Sisteme Incorporate**

**Sistem de alarmă bazat pe Arduino microcontroler, tastatură, ecran LCD, buzzer, senzori de prezență de tip PIR și servomotor**

Add Headings (Format > Paragraph styles) and they will appear in your table of contents.

# Cerința Proiectului

* Sistemul va conține 2 senzori de tip PIR pentru detecția prezenței persoanelor;
* Armarea și dezarmarea sistemului de alarmă se va face cu ajutorul unui cod special introdus de la tastatură;
* Armarea alarmei va acționa servomotorul, presupunând că în realitate acesta acționează un sistem de închidere a unei uși;
* Odată armat, sistemul va detecta prezența unei persoane neautorizate în două încăperi (un senzor în fiecare încăpere) și va afișa câte un mesaj corespunzător pe un afișaj LCD, ca de exemplu: „Alerta camera 1”, „Alerta camera 2”, „Alerta ambele camere”, „Zone libere”, etc.
* Fiecare alertă afișată pe LCD va fi însoțită și de un semnal sonor caracteristic, cu ajutorul unui buzzer;
* În cazul prezenței celor două alerte simultan, se vor succeda în buclă cele două semnale sonore diferite, cu repetitivitate de 2 – 3 secunde;
* Dacă sistemul este dezarmat, se va afișa un mesaj corespunzător pe LCD, iar buzzer-ul va fi oprit chiar dacă senzorii PIR detectează ceva.

# Explicarea funcționalității sistemului creat

Cadrul în care se află sistemul este descris de 3 camere, 2 în care se realizează detecția de mișcare, iar a 3-a - camera principală, ce este folosită ca centru de comandă, găzduind și placa de dezvoltare Arduino UNO.

Sistemul de alarmă creat are 2 stări principale:

* Sistem Armat - Ce va pune în funcțiune componentele ce avertizează și protejează utilizatorul;
* Sistem Dezarmat - Ce va scoate din funcțiune, sau va dezactiva, componentele ce avertizează și protejează utilizatorul.

Componentele mentionate mai sus, sunt:

* 2 senzori PIR ce sunt plasați în câte o cameră și vor detecta mișcarea;
* 2 led-uri, plasate într-una din cele 2 camere, ce se vor lumina intermitent atunci când sistemul este armat și senzorul din aceeași cameră detectează mișcare;
* Un display LCD, localizat în camera de comandă, pe care se vor afișa mesaje legate de detecții, armarea și dezarmarea sistemului;
* Un touchpad, ce va acționa ca input pentru a arma sau dezarma sistemul prin introducerea unui cod PIN de 4 cifre, diferit în funcție de acțiunea pe care o dorim(pornire/oprire sistem). O dată citit, PIN-ul este encodat în baza 64, și verificat;
* Un servo-motor, aflat în camera de comandă, ce va acționa pentru inchiderea sau deschiderea unei uși, o dată cu armarea sau dezarmarea sistemului;
* Un buzzer, poziționat tot în camera de comandă, ce va începe să emită câte un mesaj sonor diferit atunci când sistemul este armat și se va detecta mișcare într-una dintre cele 2 camere care se află sub supraveghere.

Inițial, sistemul va începe mereu o sesiune de funcționare în stare dezarmată (-1). Aceasta va presupune următoarele interacțiuni:

* Senzorii din camerele supravegheate vor detecta mișcare, însă semnalele luminoase și acustice nu sunt activate;
* Servo-motorul nu va acționa, rămânând și el în starea sa inițială - lăsând ușa deschisă;
* Ecranul LCD nu va afișa mesaje legate detecțiile făcute de către senzorii de mișcare;
* Buzzer-ul nu va emite semnale sonore.

Pentru a începe armarea sistemului, se va apăsa butonul **\***, iar pe ecran va apărea mesajul **Enter PIN:**.

Pentru a arma sistemul, se va introduce următorul PIN de la tastatură: **1234**. Ecranul va afișa un mesaj despre armarea reușită a sistemului, sau, în caz contrar, **Wrong PIN**. În cazul introducerii unui cod greșit, sistemul rămâne dezarmat și se poate încerca din nou armarea.

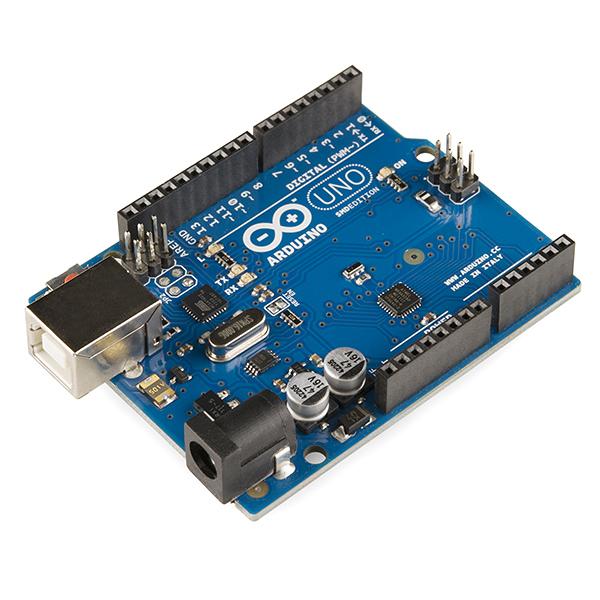
Odată armat(ajuns în starea 1), sistemul va pune în funcțiune următoarele componente:

* Senzorii detectează mișcare, câte un led luminând intermitent când se va detecta mișcare în mișcarea în care acesta de află;
* Servo-motorul va acționa, închizând ușa din camera de comandă;
* Pe afișajul LCD se vor afișa camerele în care s-au făcut detecții de mișcare:
  + Alertă camera 1;
  + Alertă camera 2;
  + Alertă ambele camere;
  + Zone libere.
* Buzzer-ul va emite câte un semnal sonor caracteristic pentru fiecare cameră, atunci cand se va detecta mișcare în ea.

Dezarmarea sistemului este similară cu armarea, însă tasta ce o acționează și PIN-ul corect sunt diferite - tasta **#**, PIN **4321**. De asemenea, în timpul dezarmării, mesajele legate de detecții de pe ecranul LCD sunt oprite, pentru a facilita introducerea pinului de dezarmare și a mesajelor ulterioare atunci când PIN-ul este corect sau greșit. Dezarmarea va duce sistemul în starea inițială, menționată anterior(-1).

Adițional, codurile PIN introduse de la tastatură vor fi encodate în baza 64, ca, mai apoi, să le fie verificate validitatea.

# Descrierea Plăcii



Placa folosită pentru dezvoltarea sistemului în cauză este [**Arduino UNO R3 ATMega328P**](https://docs.arduino.cc/static/dc64f459c8ceddad86aa8f84bca41d6a/A000066-datasheet.pdf). Această platformă de dezvoltare este de mici dimensiuni 6.8/5.3 cm și este construită în jurul unui procesor de semnal, fiind capabilă de a prelua date din mediul înconjurător pe baza a diferiți senzori, dar și de a efectua diverse acțiuni asupra mediului înconjurător prin intermediul mai multor tipuri de dispozitive mecanice.

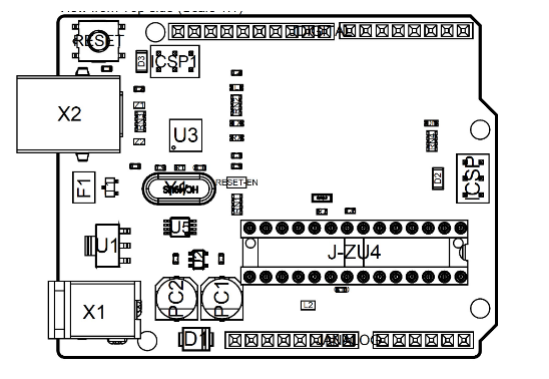
Acest procesor este capabil să ruleze cod scris într-un limbaj de programare similar cu C++.

Pentru a facilita dezvoltarea pe această placă, ea este conectată la portul USB al unui calculator printr-un cablu de tip USB A-B.

**Specificatii:**

* Tensiune de funcționare: 2.7-5.5V;
* Tensiune de alimentare Jack: 7V - 12V;
* Pini de I/O: 14;
* Pini PWM: 6 (din cei 14 de I/O);
* Pini ADC: 8;
* Memorie flash: 32kB (8 ocupati de bootloader);
* Comunicatie TWI, SPI si UART;
* Frecvența de funcționare: 16 MHz.

## **Topologia plăcii**



| **Ref.** | **Description** | **Ref.** | **Description** |
| --- | --- | --- | --- |
| X1 | Power jack 2.1x5.5mm | U1 | SPX1117M3-L-5 Regulator |
| X2 | USB A-B Connector | U3 | ATMEGA16U2 Module |
| PC1 | EEE-1EA470WP 25V SMD Capacitor | U5 | LMV358LIST-A.9 IC |
| PC2 | EEE-1EA470WP 25V SMD Capacitor | F1 | Chip Capacitor, High Density |
| D1 | CGRA4007-G Rectifier | ICSP | Pin header connector (through hole 6) |
| J-ZU4 | ATMEGA328P Module | ICSP1 | Pin header connector (through hole 6) |
| Y1 | ECS-160-20-4X-DU Oscillator |  |  |

Detalii legate de topologie:

* Power jack 2.1x5.5mm - Mufa de alimentare mama;
* USB A-B Connector - Conexiune pentru port USB tip A-B;
* EEE-1EA470WP 25V SMD Capacitor - Condensator;
* CGRA4007-G Rectifier - Redresor;
* ATMEGA328P Module - Microcontroller;
* ECS-160-20-4X-DU Oscillator - Oscilator;
* SPX1117M3-L-5 Regulator - Regulator;
* LMV358LIST-A.9 IC - Amplificator;
* Chip Capacitor, High Density - Condestarul chip-ului;
* Pin header connector - Baretă pini.

### **Pinii de conectare a plăcii**

### 

### **JAnalog**

| **PIN** | **Function** | **Type** | **Description** |
| --- | --- | --- | --- |
| 1 | NC | NC | Not connected |
| 2 | IOREF | IOREF | Reference for digital logic V - connected to 5V |
| 3 | Reset | Reset | Reset |
| 4 | +3V3 | Power | +3V3 Power Rail |
| 5 | +5V | Power | +5V Power Rail |
| 6 | GND | Power | Ground |
| 7 | GND | Power | Ground |
| 8 | VIN | Power | Voltage Input |
| 9 | A0 | Analog/GPIO | Analog input 0 /GPIO |
| 10 | A1 | Analog/GPIO | Analog input 1 /GPIO |
| 11 | A2 | Analog/GPIO | Analog input 2 /GPIO |
| 12 | A3 | Analog/GPIO | Analog input 3 /GPIO |
| 13 | A4/SDA | Analog Input/I2C | Analog input 4/I2C Data line |
| 14 | A5/SCL | Analog Input/I2C | Analog input 5/I2C Data line |

**Detalii legate de pinii analog:**

* IOREF Pin - Pin ce furnizează referința de tensiune pentru a opera placa Arduino. În cazul de față, aceasta este de 5V.
* Reset Button - La apăsare, butonul are același efect ca deconectarea și reconectarea sursei de curent - se așteaptă pentru scurt timp ca o nouă configurație să fie încărcată pe placă și se începe execuția instrucțiunilor de la început.
* Power Pins - Pinii de alimentare (3.3/5V) pot fi folosiți pentru a oferi curent altor module conectate la Arduino Board sau împământare.
* Analog/GPIO Pin:
  + Analog - Pin ce permit citirea sau scrierea unor valori între 0-1023.
  + GPIO(General Purpose Input/Output) - Permite setarea ca input sau output a pin-ului respectiv.
* Analog Input/I2C:
  + Analog input - Pin ce permite doar citirea valorilor primite între 0-1023.
  + I2C(Inter-Integrated Circuit) - este un protocol de comunicare pentru periferice cu viteză redusă.
  + I2C Bus - I2C conectat la 2 linii bidirecționale (SDA-Serial Data Line și SCL- Serial Clock). Aceste linii sunt conectate la cei doi pini (A4, A5) și prin ei se pot conecta alte componente I2C precum Ecranul LCD în cazul nostru.

### **JDigital**

| **PIN** | **Function** | **Type** | **Description** |
| --- | --- | --- | --- |
| 1 | D0 | Digital/GPIO | Digital pin 0/GPIO |
| 2 | D1 | Digital/GPIO | Digital pin 1/GPIO |
| 3 | D2 | Digital/GPIO | Digital pin 2/GPIO |
| 4 | D3 | Digital/GPIO | Digital pin 3/GPIO |
| 5 | D4 | Digital/GPIO | Digital pin 4/GPIO |
| 6 | D5 | Digital/GPIO | Digital pin 5/GPIO |
| 7 | D6 | Digital/GPIO | Digital pin 6/GPIO |
| 8 | D7 | Digital/GPIO | Digital pin 7/GPIO |
| 9 | D8 | Digital/GPIO | Digital pin 8/GPIO |
| 10 | D9 | Digital/GPIO | Digital pin 9/GPIO |
| 11 | SS | Digital | SPI Chip Select |
| 12 | MOSI | Digital | SPI1 Main Out Secondary In |
| 13 | MISO | Digital | SPI Main In Secondary Ou |
| 14 | SCK | Digital | SPI serial clock output |
| 15 | GND | Power | Ground |
| 16 | AREF | Digital | Analog reference voltage |
| 17 | A4/SD4 | Digital | Analog input 4/I2C Data line (duplicated) |
| 18 | A5/SD5 | Digital | Analog input 5/I2C Clock line (duplicated) |

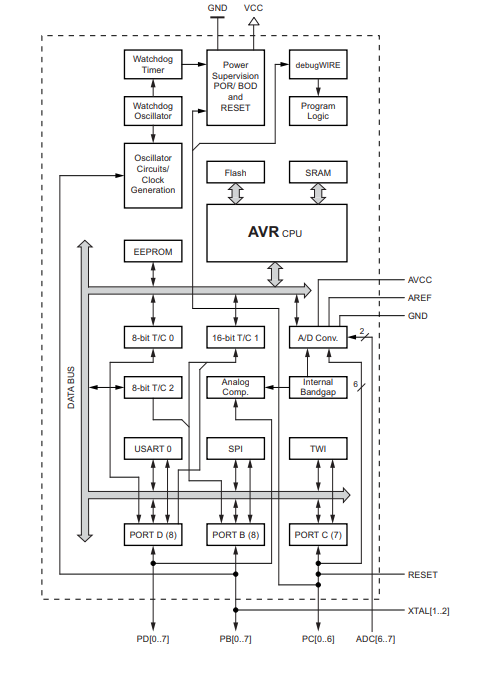
**Detalii legate de pinii analog:**

* Digital/GPIO Pin:
  + Digital - Permit citirea sau scrierea de valori LOW(0) și HIGH(1);
  + GPIO(General Purpose Input/Output) - Permite setarea ca input sau output a pin-ului respectiv.

## **Atmel ATMega328P**

[Microcontroller](https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf)-ul folosit pe aceasta placă de dezvoltare este **Atmel ATMega328P.** Acesta este un microcontroller cu consum redus, bazat pe arhitectura AVR RISC. Este capabil să execute instrucțiuni într-un singur clock cycle, reușind astfel să ajungă la un randament ce a permis optimizarea consumului de putere comparativ cu viteza de procesare.

### Diagrama bloc a microcontroller-ului:

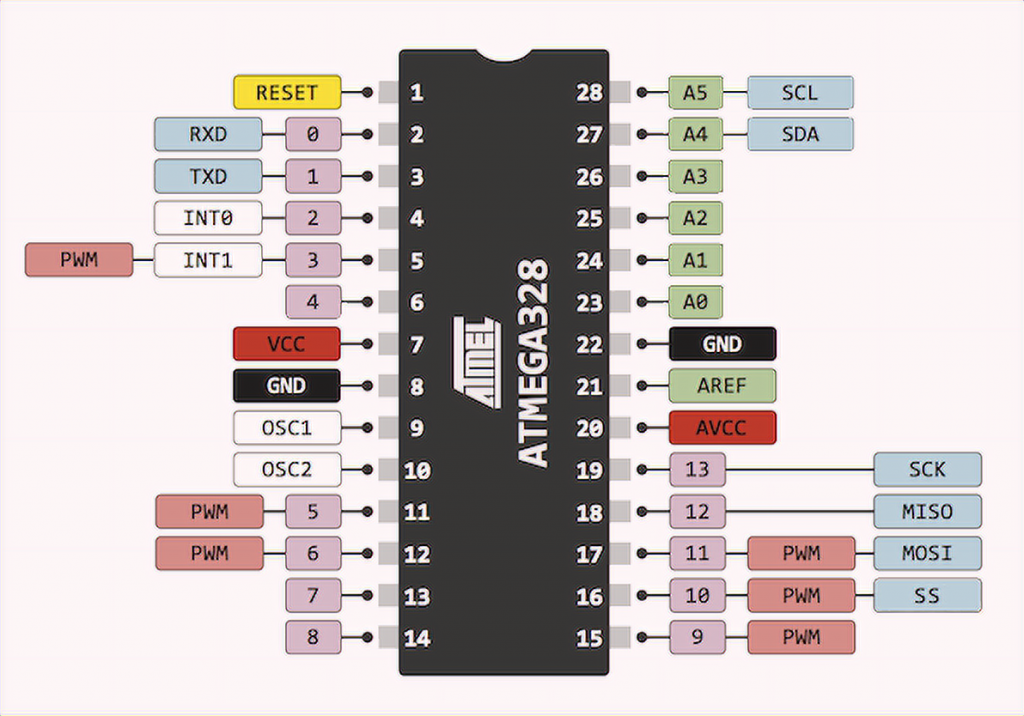


Microcontroller-ul combină o serie de 32 de regiștrii de uz general, toți conectați direct la ALU(Arithmetic Logical Unit), și 2 regiștrii independenți accesați printr-o singură instrucțiune într-un singur clock cycle.

**Specificatii:**

* Tensiune de funcționare: 3.3-5.5V;
* 32K bytes de flash programabil ce pot citi în timpul scrierii;
* 1K bytes EEPROM;
* 2K bytes SRAM;
* 23 de linii de input/output de uz general;
* 32 de regiștrii de uz general;
* 3 timere/countere flexibile cu moduri de comparare;
* Întreruperi interne și externe;
* USART serial programabil;
* O interfață serială cu pe 2 fire;
* Un port serial SPI;
* 10-bit ADC cu 6 canale;
* Watchdog programabil cu oscilator intern;
* 5 moduri de power saving.

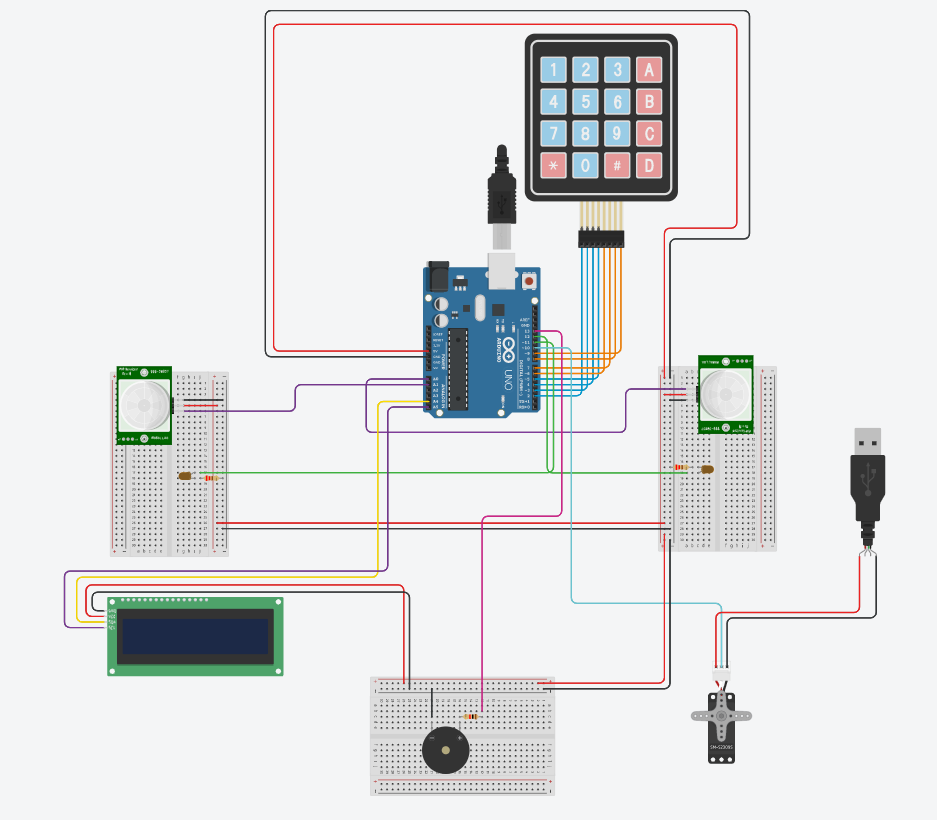
**PINOUT:**

****

# Arhitectura Sistemului

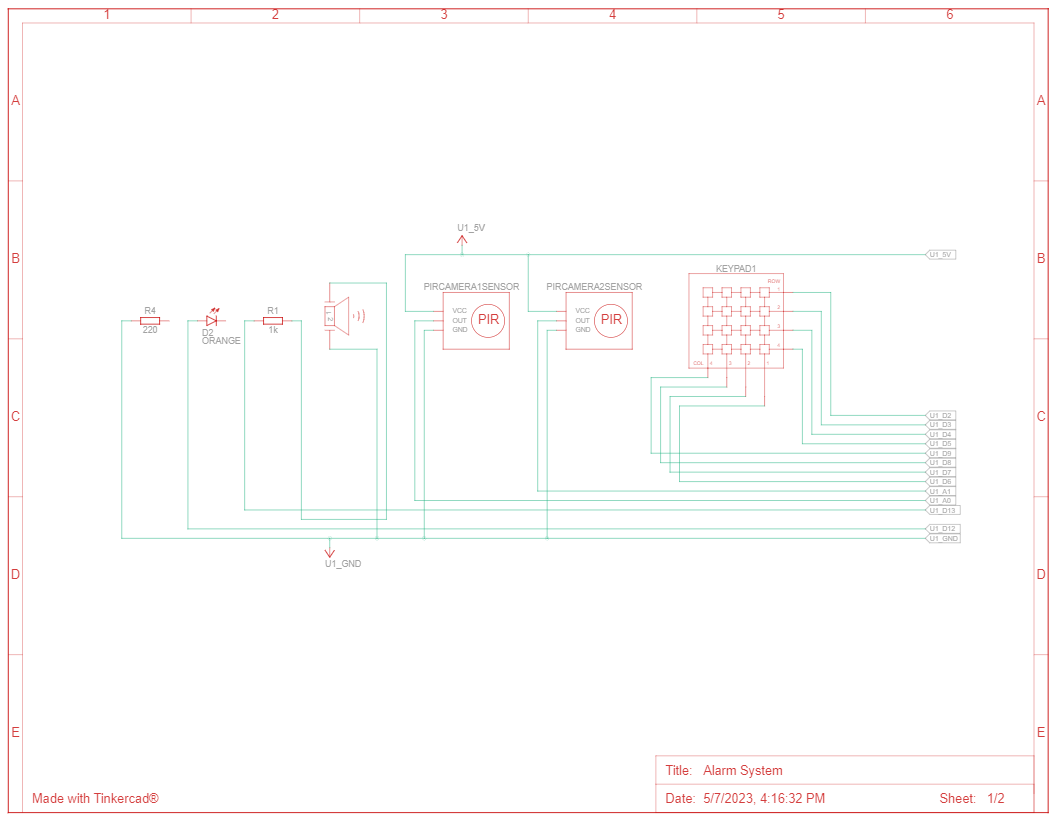
### Diagramele prezentate mai jos au fost făcute prin simularea sistemului în [Tinkercad](https://www.tinkercad.com/things/kAhFS8zgDKC).

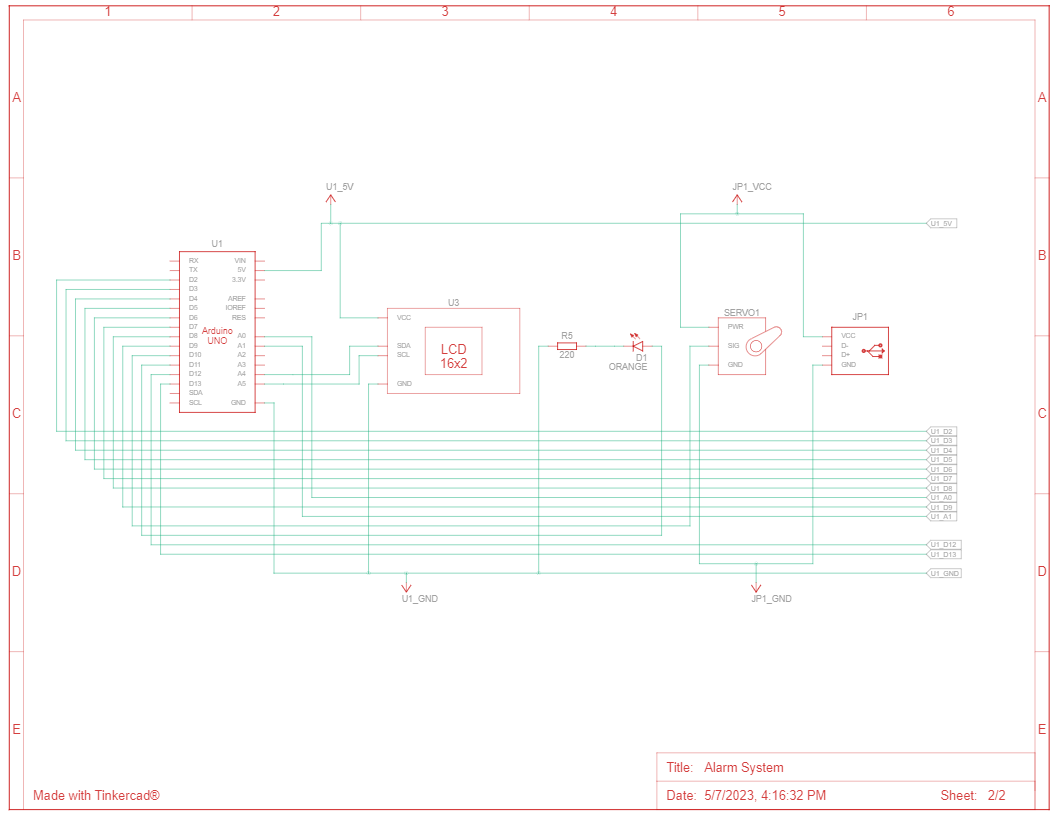
### Diagrama Circuitului



### 

### Schema Arhitecturii



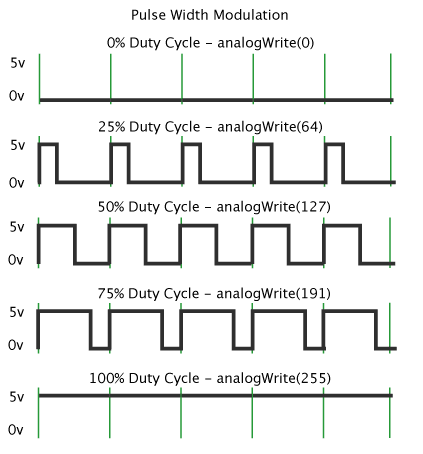


### Modulele folosite ale microcontrolerului

#### [**PWM**](https://docs.arduino.cc/learn/microcontrollers/analog-output) **- Pulse Width Modulation**

Aceasta este o metodă de a controla puterea medie trimisă de un semnal electric, pentru a primii rezultate analog(valori intre 0-1023) prin input digital(0-LOW, 1-HIGH).

Prin baleierea între HIGH și LOW sunt stimulate tensiuni între tensiunea plăcii Arduino(5V) și modul OFF(0V), schimbând durata de timp petrecută în fiecare stare. Durata de timp în starea HIGH este numită durata impulsului(pulse width).

Pentru a determina diverse valori analog, este schimbată această durată a impulsului. Pentru o mai buna exemplificare, graficul de mai jos prezintă mai multe cazuri de comportament la scrierea analog a PWM-ului.

Doar anumiți pini dispun de PWM, aceștia fiind adnotați cu ~ în fața numărului de PIN.

#### [**I2C**](https://docs.arduino.cc/learn/communication/wire) **- Inter-Integrated Circuit**

I2C este un bus serial de comunicare sincron, multi-master/multi-slave, ce folosește 2 linii pentru a trimite și primi date:

* Serial Data Pin (SDA) - prin care se trimite date între 2 componente;
* Serial Clock Pin (SCL) - folosit de controlerul Arduino pentru a trimite impulsuri la intervale regulate.

Pe măsură ce linia de timp(SCL) se modifică din LOW în HIGH, se va trimite câte un bit de informație. Această informație va crea adresa unei componente anume și o comandă sau date.

Informația este mai apoi transferată de la placă la I2C prin linia SDA. Pe măsura transmiterii fiecărui bit de informație, componenta cu care se comunică execută cerința dată și, după caz, trimite date înapoi, pe aceeași linie SDA, dar folosind SCL pentru stabilirea cadenței.

Protocolul I2C permite fiecărui dispozitiv conectat să aibă o adresă unică, deci comunicarea prin cele 2 linii cu multiple dispozitive este posibilă.

### Lista de componente

| **Nume** | **Cantitate** | **Componenta** |
| --- | --- | --- |
| U1 | 1 | Arduino UNO R3 |
| KEYPAD1 | 1 | Keypad 4x4 |
| PIEZZO1 | 1 | Piezo |
| PIRCamera1Sensor | 1 | PIR Sensor |
| PIRCamera2Sensor | 1 | PIR Sensor |
| SERVO1 | 1 | Micro Servo |
| R1 | 1 | 1k Resistor |
| R4 | 1 | 220 Ohm Resistor |
| R5 | 1 | 220 Ohm Resistor |
| D1 | 1 | LED |
| D2 | 1 | LED |
| U3 | 1 | 32 LCD 16x2 (I2C) |
| JP1 | 1 | USB Standard A\* |

Notă: USB-ul este însoțit de un modul de alimentare adăugat pentru a servi consumul servomotorului.

### Descrierea componentelor utilizate

#### [Keypad](https://cdn.sparkfun.com/assets/f/f/a/5/0/DS-16038.pdf)

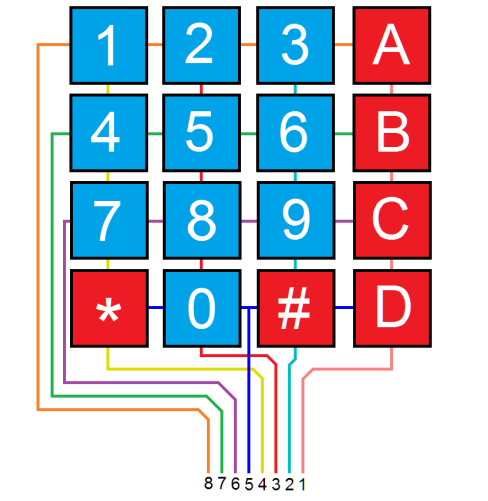
#### 

**Principiul de funcționare:**

Tastatura cu membrană 4x4 folosește o combinație de 4 rânduri și 4 coloane pentru a furniza stările fiecărui buton către dispozitivul de care este conectat. Sub fiecare cheie există un buton de tip push, conectat la 1 rând și la o coloană anume, după cum se poate vedea în imagine.

**Tipul de interfață:**

Interfața furnizează 8 Digital Pins de acces către matricea 4x4.

****

**Modul de interpretare a datelor venite de la componentă:**

Pentru ca microcontroller-ul să poată determina care buton este apăsat, va prelua stările de la fiecare din cele 4 coloane (pinii 1-4 din figura anterioară) ce pot fi HIGH(1) sau LOW(0), iar apoi stările celor 5 rânduri(pinii 5-8 din figura anterioară).

Pentru ca un buton să fie identificat ca apăsat, stările de pe o linie și o coloană ce se intersectează trebuie să apară HIGH.

De exemplu, butonul B va fi citit ca apăsat atunci când a 4-a coloană și al 3-lea rând vor trimite HIGH.

#### [Piezo](https://okystar.com/wp-content/uploads/2017/08/OKY0151.pdf)/[Buzzer](https://robocraze.com/blogs/post/how-to-use-buzzer-with-arduino#:~:text=An%20Arduino%20Buzzer%20is%20basically,electric%20pulses%20to%20the%20buzzer.)



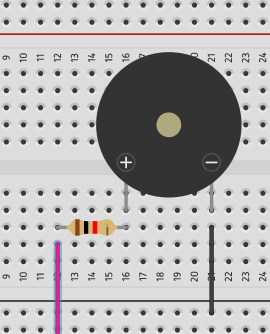
**Principiul de funcționare:**

Buzzer-ul este o componentă ce produce sunet atunci când un curent electric trece prin ea.

Buzzer-ul folosit este OKY0151 și are următoarele specificații:

* Tensiune: 3.5-5.5V;
* Curent: 25mA;
* Frecvență: 2300 +/- 200Hz;
* Sunet minim la 10 cm: 90dB;
* Temperatură de operare: 30-85C;

**Tipul de interfață:**

****

Modul prin care buzzer-ul poate fi interfațat cu placa Arduino este de a lega pin-ul positiv al acestuia la o rezistență, iar mai apoi rezistența la un pin Digital de pe Arduino. Pin-ul negativ va fi legat la Ground.

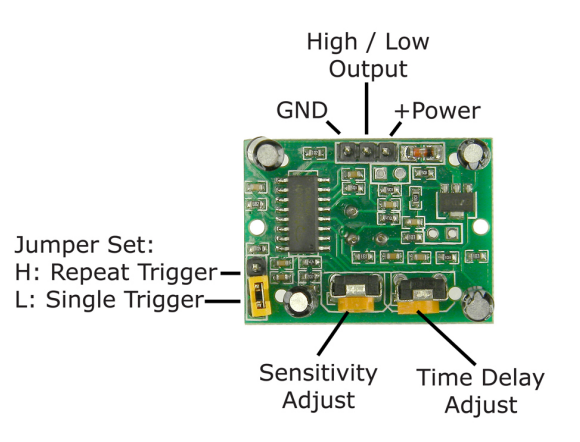
**Modul de interpretare a datelor trimise către componentă:**

Pentru a genera sunet, se va folosi comanda de generare de ton:

**tone(PIN, Frecvență, Timp)**

Prin aceasta, se setează frecvența și durata de timp la care sunetul va acționa.

#### [Pir Sensor](https://www.mpja.com/download/31227sc.pdf)



**Principiul de funcționare:**

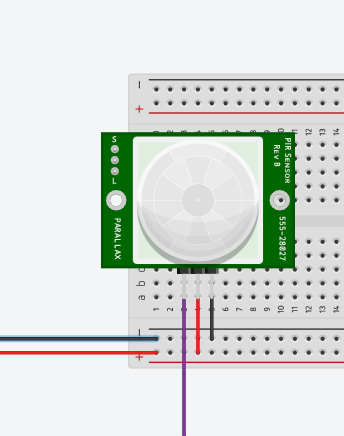
Senzorul PIR(Passive Infrared Sensor), este un modul de control automat cu senzitivitate ridicată, bazat pe tehnologia infraroșu și folosit pentru diferite echipamente de detecție.

Senzorul PIR folosit este HC-SR501, cu următoarele specificații:

* Tensiune: 5-20V;
* Curent: 65 mA;
* Timp de întârziere: 0.3-5 min;
* Timp de închidere: 0.2s;
* Mod trigger: L- dezactivează trigger repetat, H - activează trigger repetat;
* Rază de acoperire: mai puțin de 120 grade pe 7m;
* Temperatura: -15-70 Grade.

Pentru a calibra senzorul, în cazul actual jumperul este setat pe Activare repetată, iar întârzierea de timp este minimă.

**Tipul de interfață:**

****

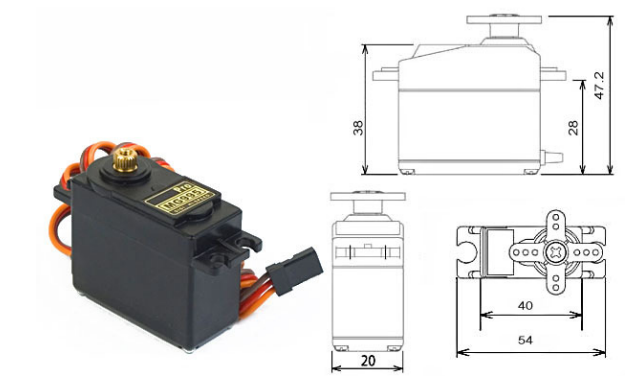
Pe baza figurii anterioare, se poate observa că interfațarea cu senzorul și placa se va face prin conectarea a 3 pini, GND, Digital Output și VCC(+Power).

**Modul de interpretare a datelor venite de la senzor:**

Valorile posibile ale citirii digitale a senzorului sunt:

* LOW (0) - Nu s-a detectat mișcare;
* HIGH (1) - Mișcare detectată.

#### [Servomotor](https://www.electronicoscaldas.com/datasheet/MG995_Tower-Pro.pdf)



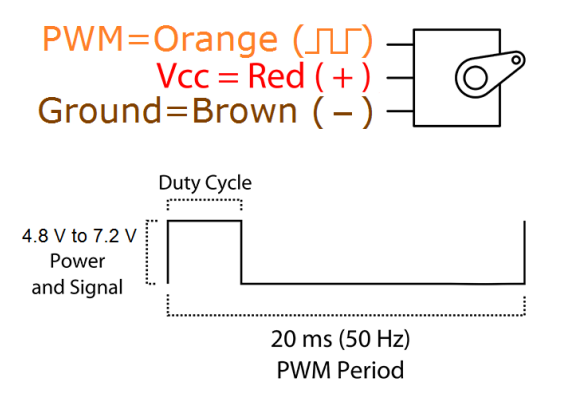
**Principiul de funcționare:**

Servomotorul folosit, MG-995, este un servo standard ce se poate roti aproximativ 120 grade în orice direcție.

Specificațiile acestuia sunt:

* Greutate: 55 g;
* Dimensiuni: 40.7 x 19.7 x 42.9 mm approx;
* Operating speed: 0.2 s/60º (4.8 V), 0.16 s/60º (6 V);
* Operating voltage: 4.8 V a 7.2 V;
* Dead band width: 5 µs;
* Stable and shock proof double ball bearing design;
* Temperature range: 0 ºC – 55 ºC.

**Tipul de interfață:**



Servomotorul este conectat, PWM PIN, Ground și VCC.

**Modul de interpretare a datelor trimise către componentă:**

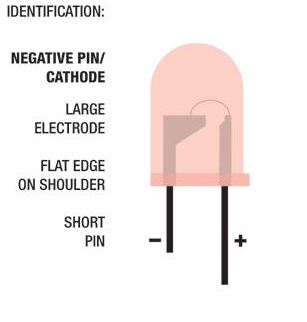
Prin folosirea librariei Arduino Servo.h, se pot trimite valori către servomotor ce vor incrementa sau decrementa progresul rotirii acestuia.

#### [Rezistențe](https://roboticsbackend.com/arduino-led-complete-tutorial/#Create_the_Arduino_LED_circuit)

Componente auxiliare ce ce limitează curentul ce poate trece prin componentele înaintea căruia este legat.

#### [LED](https://learn.adafruit.com/all-about-leds/the-led-datasheet)

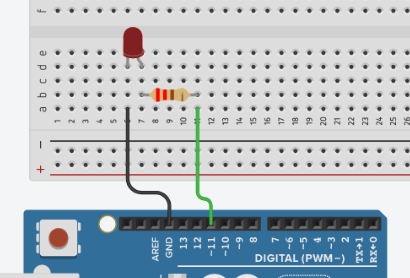
**Principiul de funcționare:**

****

[LED](https://nerdtechy.com/led-anode-vs-cathode-difference)-ul(Light Emitting Diode) este o sursă de lumină semiconductoare, aceasta fiind lumină ce este produsă de un curent ce trece prin componentă.

LED-urile sunt folosite ca semnale pentru iluminare sau indicare. LED-urile folosite în cadrul acestui proiect sunt pentru indicare, însemnând că vor indica sau reacționa la o anumită stare. LED-urile pot fi stinse, aprinse sau aprinse parțial la variante intensități.

**Tipul de interfață:**

****

Pin-ul negativ ai LED-ului va fi legat la o rezistență ce va fi la rândul ei conectată la GND. Pin-ul pozitiv va fi conectat ca si Digital Input la placă.

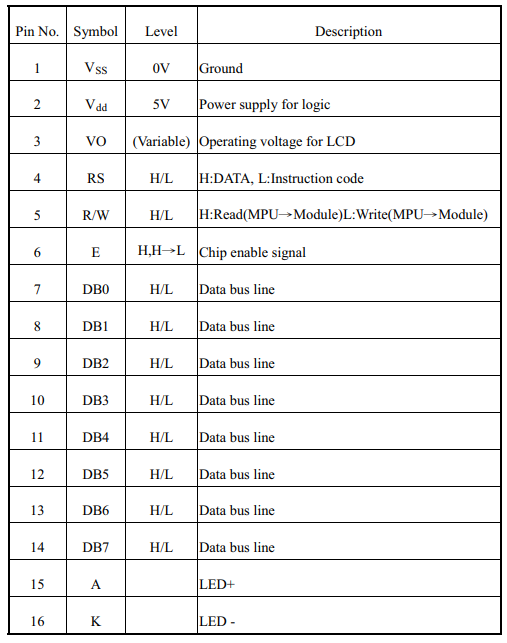
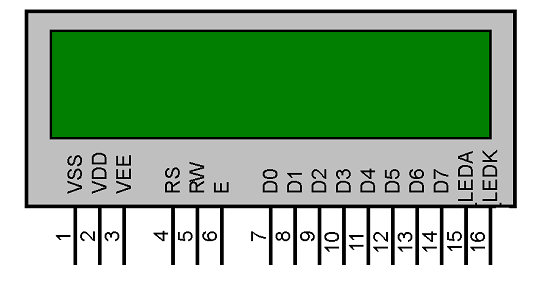
**Modul de interpretare a datelor trimise către componentă:**

LED-ul poate fi setat ca aprins(HIGH) sau stins(LOW).

#### [32 LCD 16x2 Display](https://www.vishay.com/docs/37481/lcd016n002mttiet.pdf) cu Adaptor I2C (Inter-Integrated Circuit)

**Principiul de funcționare:**

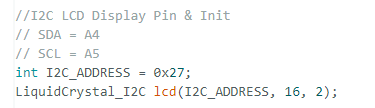
Display-ul folosit pentru afișarea mesajelor este este unul cu 16 coloane, 2 rânduri care se găsesc pe un fundal albastru în cadrul căruia caracterele sunt albe, având următoarea configurație a pinilor:



Se poate realiza afișarea datelor prin I2C cu doar 2 fire, fapt pentru care display-ului LCD i-a fost integrat un Adaptor I2C, care converteste date de la Arduino.

#### **Modul de interpretare a datelor trimise către componentă:**

Modul de conectare A display-ului LCD împreună cu Adaptorul I2C se realizeaza prin folosirea librăriei [LiquidCrystal](https://github.com/fdebrabander/Arduino-LiquidCrystal-I2C-library), pe baza căreia se determină și scrie inițializează interfața de folosire. O dată inițializat, prin folosirea metodei print de pot trimite caractere se vor fi afișate pe ecran.



#### Modul Alimentare Externă



**Principiul de funcționare:**

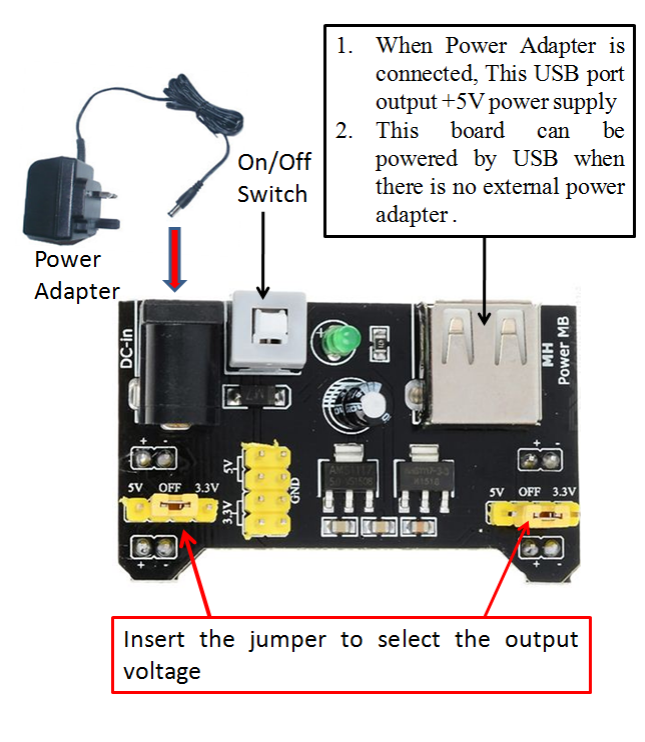
Datorită spike-urilor în consum atunci cand servomotorul este acționat, am ales să folosesc o sursă de alimentare externă pentru acesta - [HW-131](http://www.handsontec.com/dataspecs/mb102-ps.pdf).

Acest modul este compatibil cu tensiuni de ieșire de 5V, respectiv 3.3V și intrări între 6.5-12V DC.

Specificații:

* Tensiune Intrare: DC6.5-12V sau USB;
* Tensiune Ieșire: 3.3V, 5V inversabil;
* Curent maxim de ieșire: <700mA;
* Mărime: 53 \* 32mm;
* Greutate: 11g;

**Mod de folosire:**



În sistemul descris, modulul este folosit ca alimentare externă pentru servomotor.

## Cod și biblioteci folosite

Codul sursă al proiectului se găsește în acest [GitHub](https://github.com/AndraMarina/SI_Proj_Home_Alarm_System) repository, împreună cu lămuriri adiționale legate de implementare.

### Diagrama Cod

### 

### Biblioteci folosite

#### [Base64](https://github.com/Densaugeo/base64_arduino)

Librăria folosită pentru encodarea codului PIN citit de la tastatură și realizarea encodării prealabile a codurile de armare și dezarmare este Base64.

Metoda folosită pentru encodare este:



Această metodă convertește un șir de octeți într-un șir de caractere codate în baza 64, urmate de un terminator de șir și îl returnează.

**Variabilele metodei:**

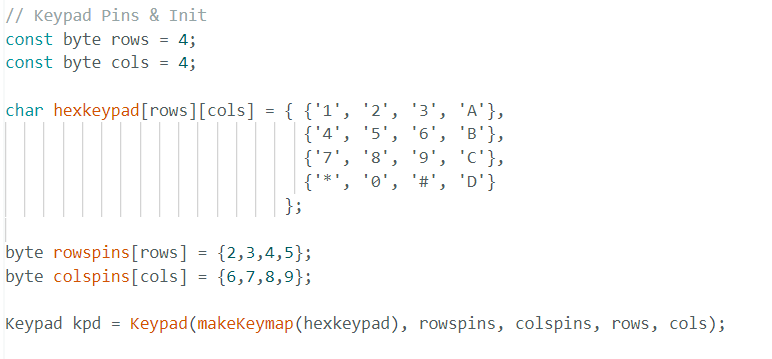
* input - pointer la stringul de input ce va urma a fi convertit;
* input\_length - lungimea șirului de caractere dat ca input;
* output - pointer la stringul ce va conține rezultatul encodat.

**Rezultat:**

Numărul de octeți al șirului de caractere decodat.

#### [Keypad](https://github.com/Chris--A/Keypad)

Biblioteca folosită pentru inițializarea și folosirea tastaturii 4x4 este Keypad.



Pentru inițializare, sunt necesare matricea caracterelor tastaturii, pinii liniilor/coloanelor, numărul de linii și coloane. Aceste inputuri sunt trimise pentru a crea o instanță a obiectului de tip Keypad, prin intermediul căruia se va face setup-ul și se vor realiza operațiile ce țin de folosirea tastaturii.

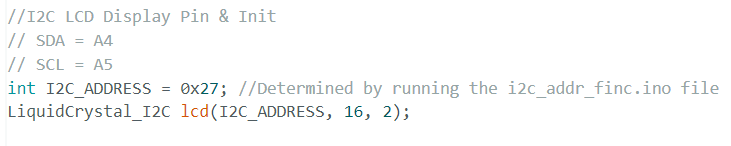
Metoda acestei clase care a fost folosită este:



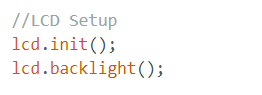
Această metodă returnează caracterul citit de la tastatură, dacă unul din butoane este apăsat și diferit de ultima citire.

#### [LiquidCrystal\_I2C](https://github.com/johnrickman/LiquidCrystal_I2C)

Această bibliotecă este folosită pentru a interfața cu Ecrane LCD cu Adaptor I2C pentru Arduino.



Inițializarea se realizează prin instanțierea unui obiect de tip LiquidCrystal\_I2C, numit lcd și având ca input adresa dispozitivul cu care se realizează comunicarea, numărul de coloane ale display-ului - 16 și linii - 2.



La setup, ecranul este inițializat și aprins.

Metodele următoare sunt folosite în cadrul proiectului:

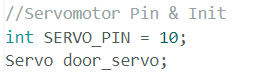


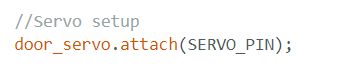
Clear va șterge mesajele ce se află deja pe ecran, rămânând astfel gol, iar print va afișa pe ecran șirul de caractere dat ca parametru metodei.

#### [Servo](https://github.com/arduino-libraries/Servo)

Această librărie este folosită pentru a inițializa și manipula servomotorul.

O instanță a unui obiect de tip servo este creată inițial.





În cadrul setup-ului, obiectului Servo îi este atașat pinul corespunzător, inițializat anterior.

Metoda folosită este:



Ea primește un întreg mai mic de 200 în cazul sistemului în cauză, deci reprezintă unghiul la care se mișcă servomotorul, o data ce aceasta metodă este apelată.

## Bibliografie

1. [Tinkercad Project.](https://www.tinkercad.com/things/kAhFS8zgDKC)
2. [Arduino UNO R3 Datasheet](https://docs.arduino.cc/resources/datasheets/A000066-datasheet.pdf).
3. [Atmel ATMega328P Datasheet](https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf).
4. [PWM Basics](https://docs.arduino.cc/learn/microcontrollers/analog-output).
5. [Guide to I2C](https://docs.arduino.cc/learn/communication/wire).
6. [4x4 KeyPad Datasheet](https://cdn.sparkfun.com/assets/f/f/a/5/0/DS-16038.pdf).
7. [OKY0151 Buzzer Datasheet](https://okystar.com/wp-content/uploads/2017/08/OKY0151.pdf).
8. [HC-SR501 PIR Sensor Datasheet](https://www.mpja.com/download/31227sc.pdf).
9. [MG-995 Servo Motor Datasheet](https://www.electronicoscaldas.com/datasheet/MG995_Tower-Pro.pdf).
10. [LCD 16x2 Display Datasheet](https://www.vishay.com/docs/37481/lcd016n002mttiet.pdf).
11. [HW-131 Power Source Datasheet](http://www.handsontec.com/dataspecs/mb102-ps.pdf).
12. [Librăria Base64](https://github.com/Densaugeo/base64_arduino).
13. [Librăria Keypad](https://github.com/Chris--A/Keypad).
14. [Librăria LiquidCrystal\_I2C](https://github.com/johnrickman/LiquidCrystal_I2C).
15. [Librăria Servo](https://github.com/arduino-libraries/Servo).