**Proiect informatică aplicată**

Tema 33 – Joc Snake cu display grafic



Studenti:

Duta Alexandru-Vlad

**Cuprins:**

I. Introducere

.....................................................................................................

II. Resurse hardware .....................................................................................................

III. Resurse software

......................................................................................................

IV. Implementare hardware

......................................................................................................

V. Implementare software ......................................................................................................

VI. Concluzii

.......................................................................................................

VII. Bibliografie ........................................................................................................

1. **Introducere:**

Prezenta lucrare practica are in vedere realizarea unui joc "snake" în care jucătorul manevrează capătul unei linii în creștere (pe o matrice led de 8 pe 8), adesea tematică ca un șarpe. Jucătorul trebuie să împiedice șarpele să se ciocnească atât de alte obstacole, cât și de el însuși, ceea ce devine mai greu pe măsură ce șarpele se prelungește.

Scopul unei astfel de lucrari este de a oferi utilizatorului oportunitanea de a se intrece cu alti jucatori in stabilirea unui scor cat mai bun.

**II.**   **Resurse hardware:**

In vederea realizarii proiectului, am uitilizat placa de dezvoltare Arduino Uno si componente aditionale, achizitionate separat.

Arduino UNO este o placă de microcontroler bazată pe ATmega328P. Are 14 pini de intrare/ieșire digitale (dintre care 6 pot fi utilizați ca ieșiri PWM), 6 intrări analogice, un rezonator ceramic de 16 MHz, o conexiune USB, o mufă de alimentare, un header ICSP și un buton de resetare. Conține tot ce este necesar pentru a susține microcontrolerul .

Placa Arduino poate fi alimentată folosind cablul USB de la computer, insa aceasta poate fii alimentata si direct la sursa de alimentare CA prin conectarea acesteia la mufa Barrel.

Funcția regulatorului de tensiune este de a controla tensiunea dată plăcii Arduino și de a stabiliza tensiunile DC utilizate de procesor și alte elemente.

Oscilatorul cu cristal îl ajută pe Arduino să facă față problemelor de timp. Cum calculează Arduino timpul? Răspunsul este, folosind oscilatorul cu cristal. Numărul imprimat deasupra cristalului Arduino este 16.000H9H. Ne spune că frecvența este de 16.000.000 de herți sau 16 MHz.

* Pini (3.3, 5, GND, Vin):
* 3,3V (6) − Alimentați 3,3 volți de ieșire
* 5V (7) − Alimentați 5 volți de ieșire
* Majoritatea componentelor utilizate cu placa Arduino funcționează bine cu 3,3 și 5 volți.
* GND (8)(Ground) - Există mai mulți pini GND pe Arduino, dintre care oricare poate fi folosit pentru a vă împământa circuitul.
* Vin (9) - Acest pin poate fi folosit și pentru a alimenta placa Arduino de la o sursă de alimentare externă, cum ar fi sursa de alimentare CA.

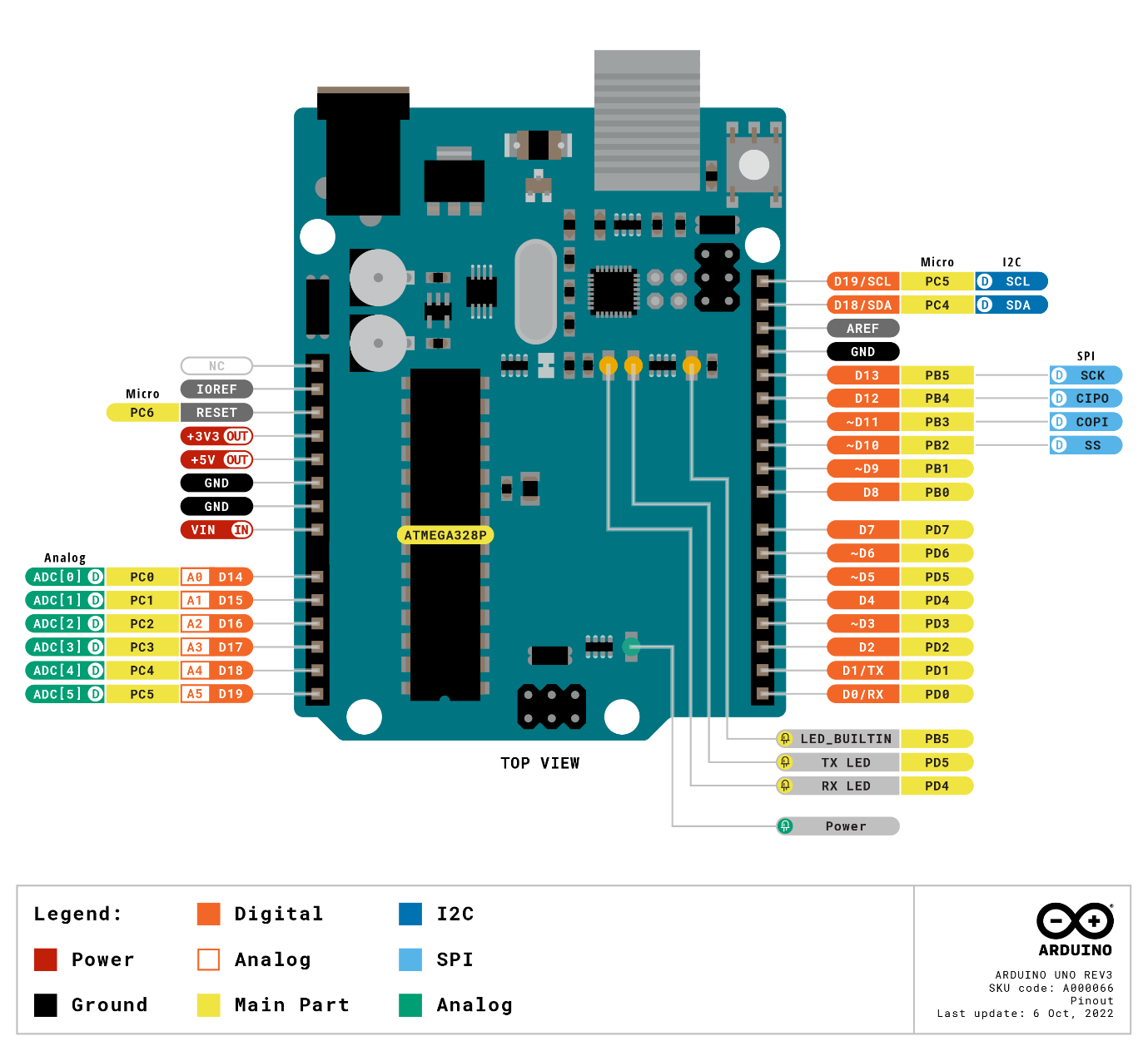
Placa Arduino UNO are șase pini de intrare analogice de la A0 la A5. Acești pini pot citi semnalul de la un senzor analog, cum ar fi senzorul de umiditate sau senzorul de temperatură și îl pot transforma într-o valoare digitală care poate fi citită de microprocesor.

Microcontroler principal

Fiecare placă Arduino are propriul microcontroler . Îl poți presupune ca fiind creierul plăcii t. IC-ul principal (circuit integrat) de pe Arduino este ușor diferit de la o placă la alta. Microcontrolerele sunt de obicei ale companiei ATMEL.

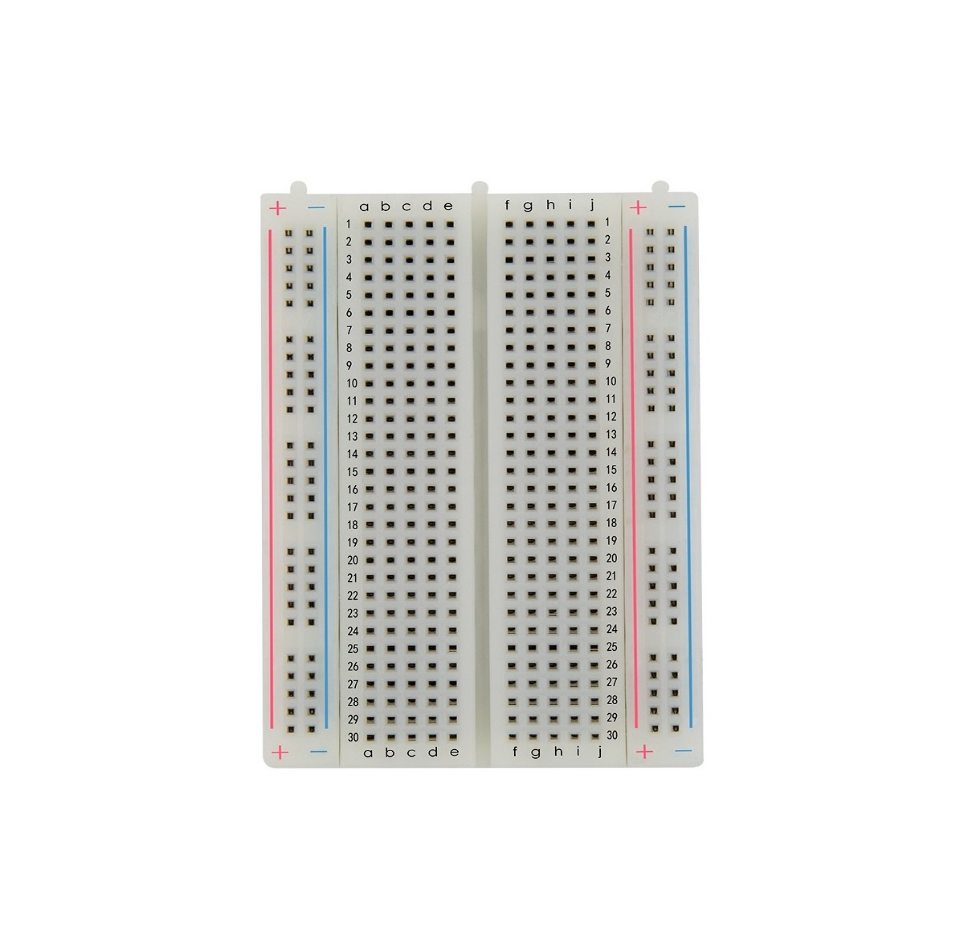
Indicator LED de alimentare

Acest LED ar trebui să se aprindă atunci când conectați Arduino la o sursă de alimentare pentru a indica faptul că placa este alimentată corect

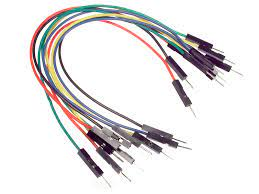


Placa Arduino UNO are 14 pini I/O digitali (15) (dintre care 6 furnizează ieșire PWM (Pulse Width Modulation). Acești pini pot fi configurați să funcționeze ca pini digitali de intrare pentru a citi valori logice (0 sau 1) sau ca digitale. pini de ieșire pentru a conduce diferite module precum LED-uri, relee etc. Pinii etichetați „~” pot fi utilizați pentru a genera PWM.

O altă componentă hardware importantă în realizarea proiectelor este breadboard-ul. Un breadboard reprezintă o placă de prototipare care facilitează realizarea rapidă și ușoară a circuitelor. Acesta conține plăcuțe metalice (contacte) dispuse într-un sistem de grilă. Pe aceste plăcuțe se pot testa funcționalitatea circuitelor înainte de a crea circuitul propriu-zis pe o placă imprimată.



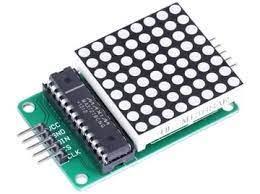
# Pentru a realiza conexiunile între anumite componente, am utilizat cabluri cu conectori tip tată-tată (male-male) si mama-tata. Pentru a realiza conexiunea între placa de dezvoltare și computer, am folosit un cablu USB de tip A la B(Cablu USB A-B 1m Arduino Mega. USB (Universal Serial Bus) reprezintă o interfață de tip plug-and-play care facilitează comunicarea între dispozitive și periferice.



O imagine care conține conector, Cablu de transfer de date, cablu

Descriere generată automat

Dispozitivul de output pentru jocul "Snake" in proiectul nostru il reprezinta o matrice led de 8x8. O matrice de 8×8 LED-uri are 64 de LED-uri (diode emițătoare de lumină) care sunt aranjate sub forma unei matrice pe 8 rânduri și 8 coloane. Prin urmare, este numit ca o matrice LED.



O alta componenta importanta in realizarea proiectului o constituie Joystick-ul . Joystick-ul analogic este similar cu două potențiometre conectate împreună, unul pentru mișcarea verticală (axa Y) și altul pentru mișcarea orizontală (axa X). Joystick-ul vine și cu un comutator Select. Poate fi foarte util pentru jocuri retro, control robot sau mașini RC.

Din joystick vom controla caraterul jocului (sarpele) in directia dorita pentru a ne indeplini scopul, cel de a „manca” cat mai multe mere si a stabili un scor cat mai mare.



**III**. **Resurse software:**

Componenta software reprezintă elementul cheie în asigurarea funcționalității fiecărei componente hardware și constă într-un ansamblu de programe și rutine. În cazul plăcii de dezvoltare Arduin Uno, aceasta se conectează la computer prin intermediul unui cablu USB de tip A, stabilind astfel legătura între placa de dezvoltare și mediul de programare Arduino IDE.

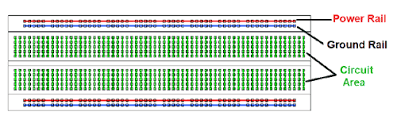
Arduino Software (IDE) reprezintă un mediu de programare open-source care facilitează scrierea și încărcarea codului pe microcontroler. Acest mediu de programare permite programatorilor să dezvolte aplicații și să își încarce codul pe placa Arduino Uno într-un mod simplu și eficient.

În cadrul mediului de programare poate fi accesată interfața Serială, pe care se pot afișa mesaje sau date de input/output.

**IV. Implementare hardware:**

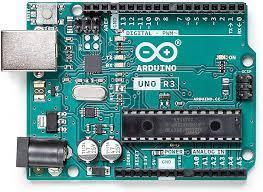
1. **Breadboard-ul:**

Breadboard-ul (placa de prototipare) reprezintă o soluție utilizată pentru a realiza circuite electrice într-un mod provizoriu, rapid și ușor. Toate componentele utilizate în proiect sunt conectate la acesta, iar pinii acestor componente fac contact pe plăcuțele metalice dispuse în sistemul de grilă al breadboard-ului. În figura de mai jos este prezentată o diagramă care ilustrează modul de interconectare a componentelor pe breadboard.



2. **Placa Arduino Uno:**

Componenta centrală a întregului nostru proiect, modulul Arduino Uno, este conectată prin înfingerea pinilor săi pe breadboard. După aceasta, modulul Arduino Uno este conectat la computer prin intermediul cablului USB, pentru a prelua codul din mediul de programare. Odată ce codul este încărcat cu succes, placa Arduino Uno va iniția funcționarea circuitului realizat pe breadboard. Astfel, Modulul Arduino Uno joacă un rol crucial în controlul și coordonarea funcționării întregului sistem.



3.**Joystick-ul:**

Joystick-ul reprezinta o modalitate simpla de controla caracterul din jocul snake. Din acesta vom misca caraterul in 4 directii (sus, jos, dreapta, stanga). Tastatura se conecteaza la placa Arduino Uno cu un pin la GND, cu un pic la 5 V(volti) si cu alti 2 pini la Arduino Uno.

4**.Matricea led de 8x8**:

Matricea led de 8x8 reprezinta o variata simpla de a reda imaginea jocului snake printr-un ecran monocrom, toate ledurile find albe. Aceasta componenta electronica este folosita pentru output-ul jocului. Cu un pin se conectează la borna rece (GND), cu alti trei se conectează la pini de ieșire ai plăcuței Arduino Uno iar cu alt pin se conecteaza la 5V (volti).

În diagrama de mai jos este ilustrată schema electrică a circuitului realizat pentru acest proiect:

O imagine care conține text, captură de ecran, afișaj, Software multimedia

Descriere generată automat

În diagrama de mai jos este ilustrată schema electrică a circuitului realizat pentru acest proiect:

O imagine care conține Cabluri electrice, electronice, cablu, Inginerie electronică

Descriere generată automat

**V. Implementare software:**

Diagramele urmatoare reprezinta schemele logice ale codului folosit:

O imagine care conține text, captură de ecran, proiectare

Descriere generată automat

O imagine care conține schiță, captură de ecran, desen, linie

Descriere generată automat

O imagine care conține captură de ecran, proiectare, artă

Descriere generată automat

Codul sursă utilizat în proiectul nostru se regăsește mai jos:

#include <LedControl.h>

//Define The Snake as a Struct

typedef struct Snake Snake;

struct Snake{

int head[2]; // the (row, column) of the snake head

int body[40][2]; //An array that contains the (row, column) coordinates

int len; //The length of the snake

int dir[2]; //A direction to move the snake along

};

//Define The Apple as a Struct

typedef struct Apple Apple;

struct Apple{

int rPos; //The row index of the apple

int cPos; //The column index of the apple

};

//MAX72XX led Matrix

const int DIN =12;

const int CS =11;

const int CLK = 10;

LedControl lc = LedControl(DIN, CLK, CS,1);

const int varXPin = A3;//X Value from Joystick

const int varYPin = A4;//Y Value from Joystick

byte pic[8] = {0,0,0,0,0,0,0,0};//The 8 rows of the LED Matrix

Snake snake = {{1,5},{{0,5}, {1,5}}, 2, {1,0}};//Initialize a snake object

Apple apple = {(int)random(0,8),(int)random(0,8)};//Initialize an apple object

//Variables To Handle The Game Time

float oldTime = 0;

float timer = 0;

float updateRate = 3;

int i,j;//Counters

void setup() {

// put your setup code here, to run once:

/\*

The MAX72XX is in power-saving mode on startup,

we have to do a wakeup call

\*/

lc.shutdown(0,false);

/\* Set the brightness to a medium values \*/

lc.setIntensity(0,8);

/\* and clear the display \*/

lc.clearDisplay(0);

//Set Joystick Pins as INPUTs

pinMode(varXPin, INPUT);

pinMode(varYPin, INPUT);

}

void loop() {

// put your main code here, to run repeatedly:

float deltaTime = calculateDeltaTime();

timer += deltaTime;

//Check For Inputs

int xVal = analogRead(varXPin);

int yVal = analogRead(varYPin);

if(xVal<100 && snake.dir[1]==0){

snake.dir[0] = 0;

snake.dir[1] = -1;

}else if(xVal >920 && snake.dir[1]==0){

snake.dir[0] = 0;

snake.dir[1] = 1;

}else if(yVal<100 && snake.dir[0]==0){

snake.dir[0] = -1;

snake.dir[1] = 0;

}else if(yVal >920 && snake.dir[0]==0){

snake.dir[0] = 1;

snake.dir[1] = 0;

}

//Update

if(timer > 1000/updateRate){

timer = 0;

Update();

}

//Render

Render();

}

float calculateDeltaTime(){

float currentTime = millis();

float dt = currentTime - oldTime;

oldTime = currentTime;

return dt;

}

void reset(){

for(int j=0;j<8;j++){

pic[j] = 0;

}

}

void Update(){

reset();//Reset (Clear) the 8x8 LED matrix

int newHead[2] = {snake.head[0]+snake.dir[0], snake.head[1]+snake.dir[1]};

//Handle Borders

if(newHead[0]==8){

newHead[0]=0;

}else if(newHead[0]==-1){

newHead[0] = 7;

}else if(newHead[1]==8){

newHead[1]=0;

}else if(newHead[1]==-1){

newHead[1]=7;

}

//Check If The Snake hits itself

for(j=0;j<snake.len;j++){

if(snake.body[j][0] == newHead[0] && snake.body[j][1] == newHead[1]){

//Pause the game for 1 sec then Reset it

delay(1000);

snake = {{1,5},{{0,5}, {1,5}}, 2, {1,0}};//Reinitialize the snake object

apple = {(int)random(0,8),(int)random(0,8)};//Reinitialize an apple object

return;

}

}

//Check if The snake ate the apple

if(newHead[0] == apple.rPos && newHead[1] ==apple.cPos){

snake.len = snake.len+1;

apple.rPos = (int)random(0,8);

apple.cPos = (int)random(0,8);

}else{

removeFirst();//Shifting the array to the left

}

snake.body[snake.len-1][0]= newHead[0];

snake.body[snake.len-1][1]= newHead[1];

snake.head[0] = newHead[0];

snake.head[1] = newHead[1];

//Update the pic Array to Display(snake and apple)

for(j=0;j<snake.len;j++){

pic[snake.body[j][0]] |= 128 >> snake.body[j][1];

}

pic[apple.rPos] |= 128 >> apple.cPos;

}

void Render(){

for(i=0;i<8;i++){

lc.setRow(0,i,pic[i]);

}

}

void removeFirst(){

for(j=1;j<snake.len;j++){

snake.body[j-1][0] = snake.body[j][0];

snake.body[j-1][1] = snake.body[j][1];

}

**VI. Concluzii:**

Concluziile pentru un proiect Snake pe o matrice LED folosind Arduino Uno pot fi următoarele:

* Proiectul Snake pe matrice LED reprezintă o modalitate distractivă și interactivă de a-ți dezvolta abilitățile de programare și de a te familiariza cu Arduino Uno.
* Utilizarea unei matrice LED pentru a reprezenta jocul Snake oferă o interfață vizuală atractivă și ușor de înțeles pentru jucători.
* Programarea jocului Snake pe Arduino Uno necesită manipularea matricei LED pentru a afișa și actualiza poziția șarpelui, mâncarea și orice obstacole sau margini ale jocului.
* Algoritmul de control al șarpelui trebuie implementat astfel încât să se asigure mișcarea corectă a acestuia pe matrice. Acest algoritm poate implica detectarea tastelor apăsate de utilizator sau implementarea unei logici de mișcare automate.
* Pentru a evita conflictele și coliziunile, este important să se asigure că șarpele nu trece peste propriul său corp sau nu se ciocnește de marginea matricei sau de alte obstacole.
* Este necesară implementarea unui mecanism de creștere a lungimii șarpelui atunci când acesta mănâncă o bucată de hrană și de a genera noi bucăți de hrană în mod aleatoriu pe matrice.
* Implementarea unei logici de scor pentru a urmări performanțele jucătorului poate adăuga un element competitiv jocului.
* Este important să se ia în considerare limitările de memorie și de putere de procesare ale Arduino Uno în timpul dezvoltării și optimizării jocului Snake.
* Proiectul Snake pe matrice LED poate fi adaptat și extins pentru a include caracteristici suplimentare, cum ar fi niveluri de dificultate, sunete sau o interfață utilizator mai avansată.
* În final, acest proiect poate reprezenta o oportunitate excelentă de a explora și de a experimenta cu Arduino Uno și de a-ți îmbunătăți abilitățile de programare într-un mod distractiv și creativ.

**Probleme intampinate**:

* Probleme cu librarile
* Probleme cu codul pentru placuta ESP32
* Conectivitate( cabluri)

**VII. Bibliografie:**

Schemă electrică: <https://wokwi.com/projects/new/esp32>

Mediu de programare: <https://www.arduino.cc/en/software>

Librarie: <https://github.com/Matiasus/SSD1306>

Arduino Uno: <https://www.tutorialspoint.com/arduino/arduino_board_description.htm#>

Joystick: <https://exploreembedded.com/wiki/Analog_JoyStick_with_Arduino>

Matrice led: <https://how2electronics.com/8x8-led-matrix-max7219-arduino/>