

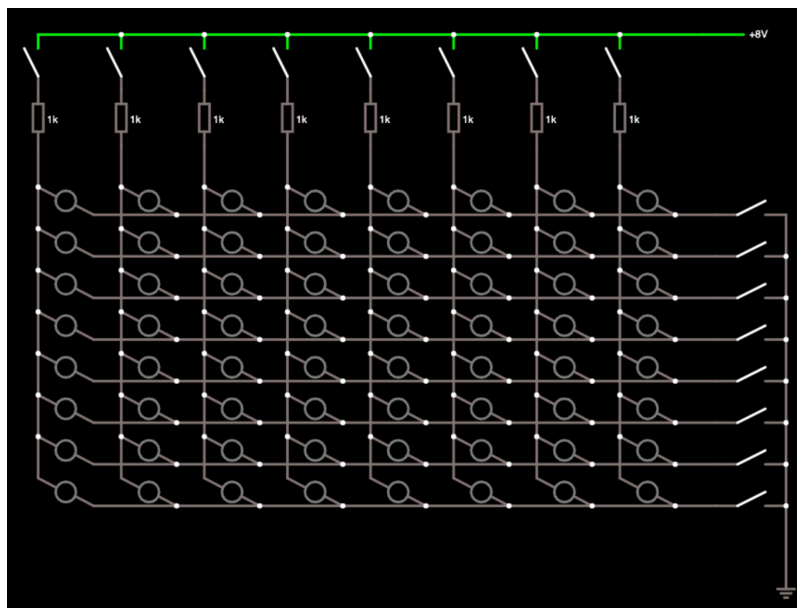
## Electronica Digitala - Proiect

### **8x8 LED Matrix controlled with Shift Register + Arduino**

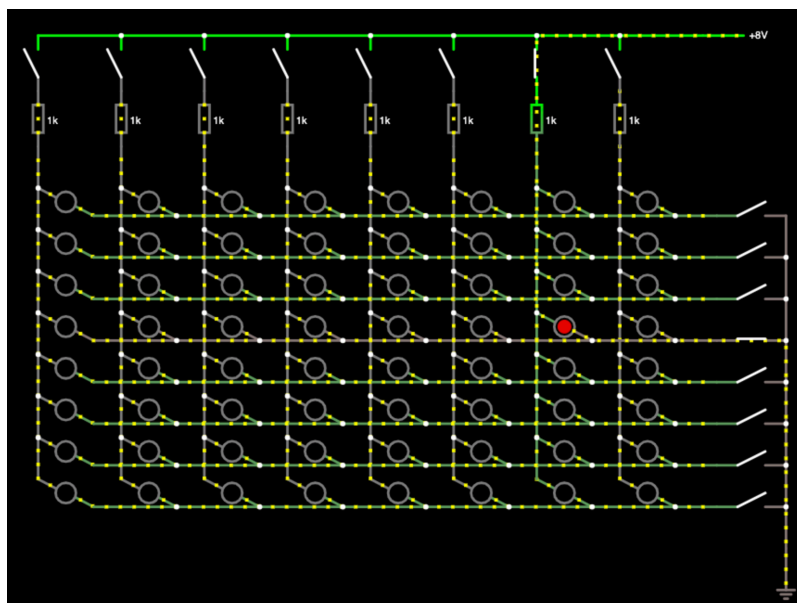
#### Descrierea proiectului:

Acest proiect presupune crearea unei matrici de LED-uri, care sa fie controlate pe randuri si coloane cu ajutorul unui microcontroller Arduino. O componenta intermediara, care ajuta controlarea tuturor LED-urilor, daca este cazul, la aceiasi parametri, sunt doua shift registre. Acestea au rolul de a alege randul si coloana unui LED, atunci cand se doreste iluminarea acestuia.

*Schema matricii de LED-uri controlate, in acest desen, de niste switch-uri*



*Cum se realizeaza aprinderea unui LED folosind switch-uri*



## Descrierea circuitului:

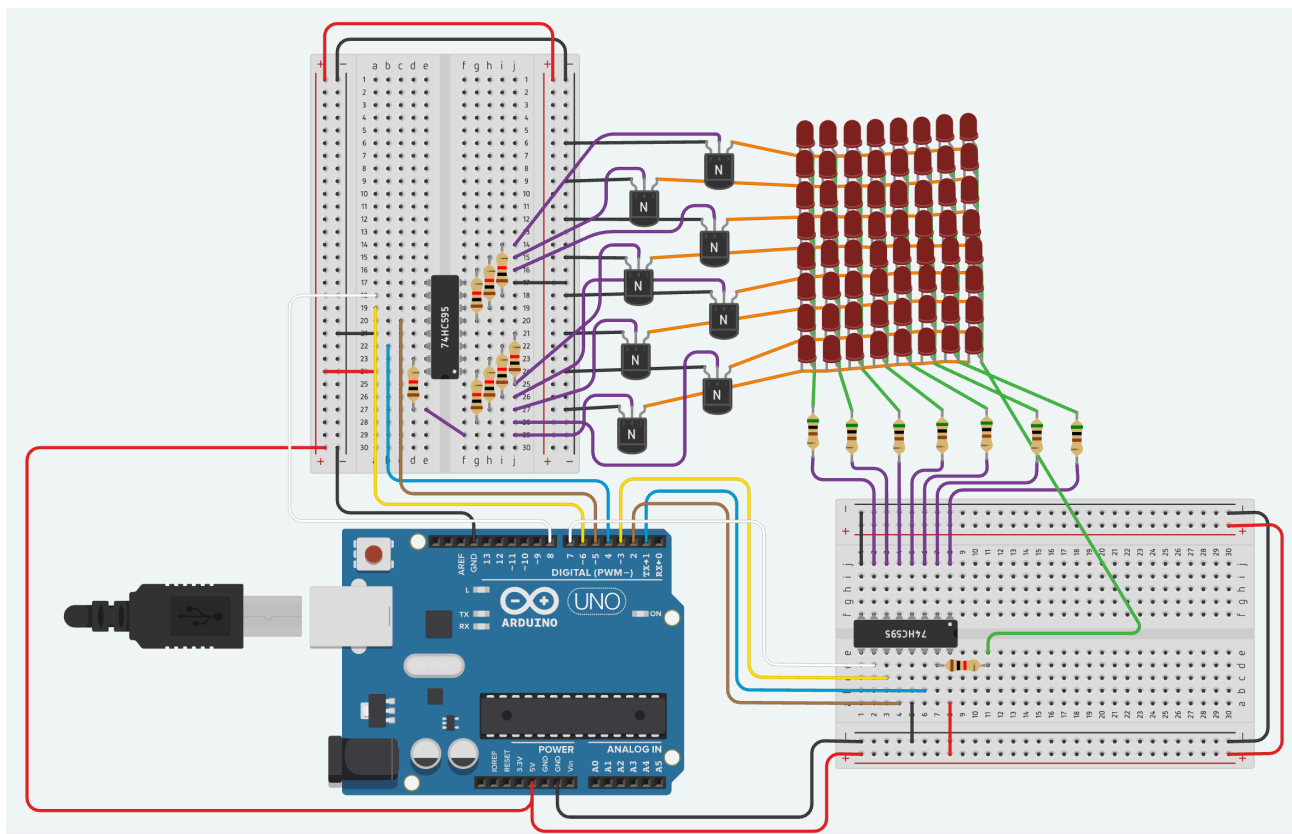
Microcontroller-ul Arduino este legat la cele doua shift registre prin 4 legaturi, care controleaza pentru fiecare din ei:

- *datele de intrare in registru* (SER sau DS) = pe aici sunt introduse datele in registru
- *shift register clock* (SRCLK) = semnalul de ceas care corespunde memorarii datelor de intrare; memorarea (si shiftarea datelor) la o intrare se fac pe frontul crescator al acestui ceas; valoarea nou adaugata poate fi LOW sau HIGH, in functie de intrarea DS
- *shift register clear* (SRCLR) = sterge datele din registru; trebuie tinut pe HIGH (este negat in interior) in timpul memorarii datelor sau pe LOW daca se doreste golirea registrului
- *output register clock* (RCLK) = semnalul de ceas corespunzator flush-ului de date la outputuri; trebuie tinut pe LOW in timpul scrierii datelor in registru; cand se doreste afisarea datelor la iesiri, acesta este trecut pe HIGH

Fiecare *registru* are 8 iesiri (Q1 - Q8), care sunt legate la anozii sau catozii LED-urilor. Matricea de LED-uri are 8 coloane a cate 8 LED-uri. In reprezentarea abordata, LED-urile sunt conectate pe coloane la anozii si pe linii la catodzi. Astfel, un registru este legat la anozii si celalalt la catodzi. Cel care este legat la borna pozitiva are doar cate o rezistenta de sarcina pusa in serie cu coloana de LED-uri pentru fiecare din cele 8 iesiri. Cel de al doilea registru, care este conectat la borna negativa, nu este legat direct la diode, ci are iesirile legate la bazele a 8 tranzistori, care sunt conectati astfel:

- *colector* - conectat la coloana din matrice corespunzatoare
- *emitor* - conectat la masa
- *baza* - conectat la o iesire a registrului, in serie cu o rezistenta

Circuitul realizat in TinkerCad



### Cum se aprinde un LED?

Aprinderea unui LED se realizeaza daca coloana si randul corespunzatoare lui sunt *activate* de registri. Aici, a *activa* presupune ca output-urile care corespund liniei si coloanei sa fie HIGH (1 logic) in momentul in care se face flush la datele memorate de registru. Daca oricare dintre aceste doua este pe LOW, LED-ul nu se aprinde.

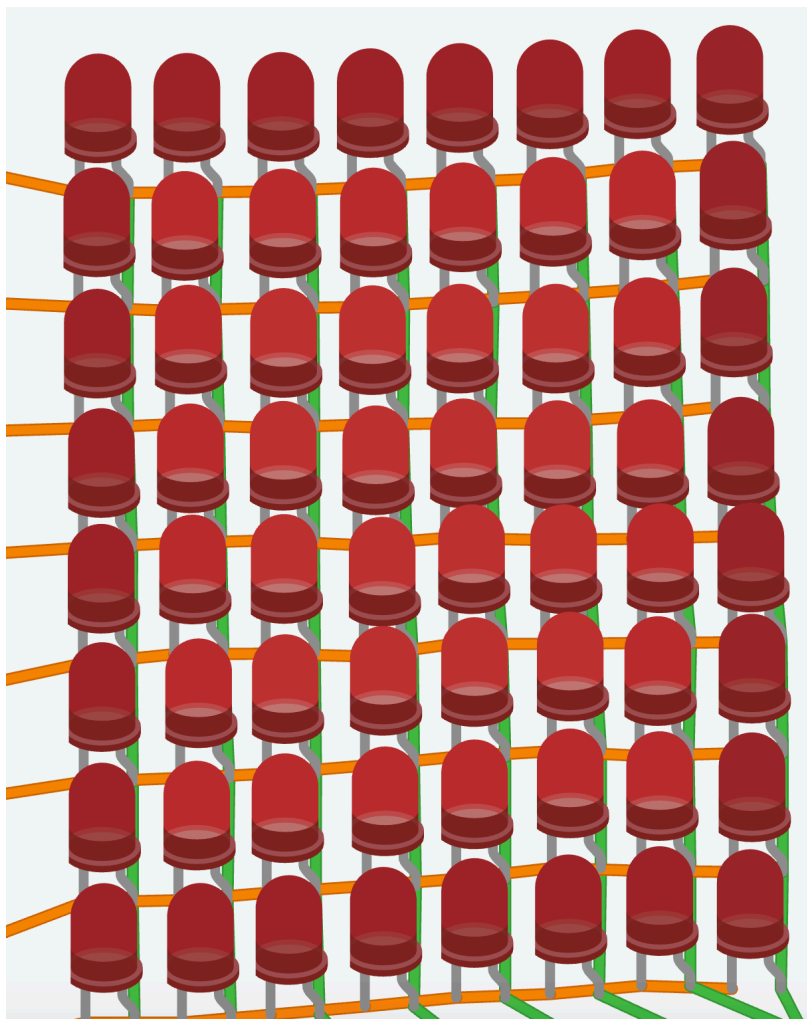
### Controlarea cu Arduino

Microcontroller-ul are rolul de a inmagazina in registre valorile de HIGH si LOW dorite pentru “desenarea” oricarei imagini cu ajutorul LED-urilor din matrice. Pentru registrul de la anazi, coloanele sunt legate la output-uri de la dreapta la stanga, iar pentru celalalt registru de jos in sus.

Pentru orice fel de desen, trebuie respectati urmasorii pasi:

1. Setarea SRCLR pe HIGH pentru a preveni “curatarea” registrului
2. Setarea RCLK pe LOW (output clock)
3. Scrierea in fiecare dintre registre a 8 valori de HIGH sau LOW in combinatia dorita (valoarea trebuie incadrata de setarea pe LOW a SRCLK inainte si pe HIGH dupa)
4. Trecerea RCLK pe HIGH pentru a da flush la date si a vedea imaginea
5. *Optional*: Setarea SRCLR pe LOW pentru curatarea registrului, daca se doreste repetarea procesului

*Aprinderea LED-urilor din interior, lasand neaprinse pe cele de la margine*



**Observatii, dificultati intampinate, lucruri invatate:**

- O dificultate la inceput a fost, desi probabil s-a intamplat tuturor, intelegerea in totalitate a circuitului si a relationarii dintre elementele sale. Am urmarit clipurile si am citit materialele sugerate de mai multe ori, pana am fost in stare sa zic ca am inteles circuitul. Chiar si asa, am gresit desenele de cateva ori.
- Ce am invatat? Probabil cele mai de baza concepte si instructiuni pentru Arduino. Chiar si asa, controller-ul nu a fost deloc greu de inteles, iar faptul ca pentru registre am avut datasheet-ul mereu deschis, in care se explica functiile acestuia si cum se executa fiecare, in functie de intrarile sale, a usurat mult scrierea codului. Mai mult mi-a luat sa imi dau seama ca am totusi nevoie de golirea registrului (SRCLR) daca vreau sa desenez mai multe lucruri unele dupa altele si sa nu am probleme la afisare.
- Codul Arduino scris afiseaza intr-o bucla infinita niste patrate concentrice care se micsoreaza treptat si niste modele simple de LED-uri aprinse si stinse. Initial am vrut sa imi scriu numele, insa nu am reusit sa fac tocmai cum mi-am dorit, fiind si prima data cand incercam sa scriu instructiunile pentru registri. De aceea, in cod exista vreo 300 de linii comentate cu ce reusisem sa scriu din numele meu pana sa renunt si sa realizez ceva mai usor.
- Ce am mai invatat? Sa citesc datasheet-uri \*smiley face\* si sa pot asocia desenele, schemele din datasheet-uri la toate utilizarile in realitate a elementelor de circuit (de pe YouTube). Dupa ce stiam ce pini au registrii si ce rol are fiecare, parca si tutorialele aveau un sens mult mai clar. Am observat ca registrul 74hc595 utilizat in TinkerCad se potriveste de minune documentatiei de la Texas Instruments, desi am utilizat si documentatie de la NXP Semiconductors in paralel cu prima.
- Alaturi de acest document, arhiva va contine si codul in C pentru Arduino si circuitul in TinkerCad, dar si Falstad.