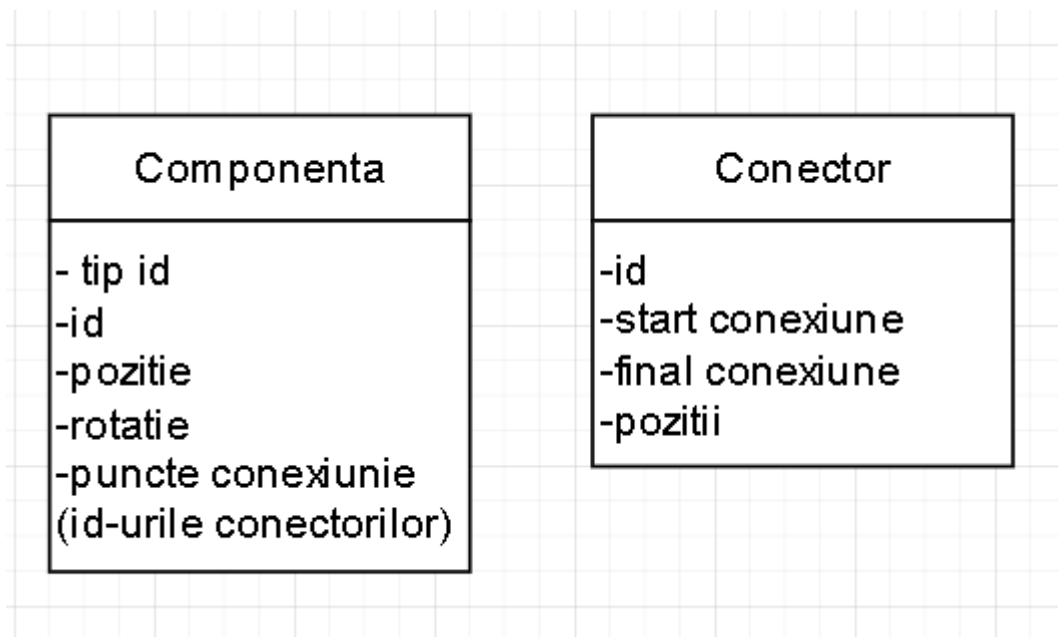


Sistem Save Load Electron
Documentatie tehnica

Circuitul e salvat si incarcat printr-un **fișier text**. Fiecarei salvări îi corespunde **un singur** fișier text.

SALVARE

Datele salvate sunt:



Orice camp care salveaza o conexiune cu un **Conector**, **Componenta**, sau **PunctConexiune** foloseste **id**-ul acelui element pentru a-l gasi.

Structura unui fisier de salvare:

```

1  <id> (inceput citire a ultimului id atribuit)
datelei 21 (ultimul id atribuit)
unei
componente <componente> (inceput citire date componente)
4  3 (tip_id , tipul de componenta, din TipuriComponente.h)
5  13 (id, numar unic atribuit fiecarei Componente create)
6  (9,10) (pozitia)
7  0 (rotatia)
8  19 -1 21 (pentru fiecare punct de conexiune, in ordine,
9          id-ul Conectorului legat)
10 7
11 15
12 (12,11)
13 0
14 21
15 0
16 17
17 (10,8)
18 0
19 19 -1
20 <conectori> (inceput citire date conectori)
21 19 (id conector)
22 (17,0) start_conex (date puncte conexiune sub forma:
23 (13,0) final_conex " (id parinte pct, index pct) ")
24 (9,8) (9,9) pozitiile conectorului
25 (10,10) (11,10) (12,10) (pozitiile prin care trece pe grid)
26 21
27 (13,2)
28 (15,0)

```

Diagram illustrating the data structure for components and connectors. The input is a sequence of values, with annotations explaining their meaning and grouping.

Componente (Lines 4-19):

- Line 4: 3 (tip_id , tipul de componenta, din TipuriComponente.h)
- Line 5: 13 (id, numar **unic** atribuit fiecarei Componente create)
- Line 6: (9,10) (pozitia)
- Line 7: 0 (rotatia)
- Line 8: 19 -1 21 (pentru fiecare punct de conexiune, in ordine, id-ul Conectorului legat)
- Line 9: 7
- Line 10: 15
- Line 11: (12,11)
- Line 12: 0
- Line 13: 21
- Line 14: 0
- Line 15: 17
- Line 16: (10,8)
- Line 17: 0
- Line 18: 19 -1

Conectori (Lines 20-28):

- Line 20: <conectori> (inceput citire date conectori)
- Line 21: 19 (id conector)
- Line 22: (17,0) start_conex (date puncte conexiune sub forma: " (id parinte pct, index pct) ")
- Line 23: (13,0) final_conex
- Line 24: (9,8) (9,9) pozitiile conectorului (pozitiile prin care trece pe grid)
- Line 25: (10,10) (11,10) (12,10)
- Line 26: 21
- Line 27: (13,2)
- Line 28: (15,0)

- in cazul in care un Conector nu are pozitii, in loc de lista de poziti se va salva -1
- in cazul in care o Componenta are un punct de conexiune care NU e conectat la un Conector, se va salva -1 in loc de ID-ul conectorului

LOAD

Incarcarea fisierului se face in cateva etape:

- 1.separa datele din fisier
- 2.construieste componentele si conectorii (fara datele de asociere)
- 3.completeaza datele de asociere, conectand dupa id-uri

1: se parcurge fisierul, si in functie de tagurie de inceput date (<id>, <componente>, etc) se separa textul din fisier in 3 categorii:

- Ultimul ID salvat
- Date despre componente
- Date despre conectori

```
int categorie_date = 0; //1 - date componente, 2 - date conectori, 3 - date contor id

string date_componente = "";
string date_conectori = "";

//obține datele componentelor is a conectorilor separat

int ultim_id = -1;
```

```

while (!input.eof()) {
    getline(input, linie_citita);
    if (linie_citita == "<componente>") {
        categorie_date = 1;
        continue;
    }
    else if (linie_citita == "<conectori>") {
        categorie_date = 2;
        continue;
    }
    else if (linie_citita == "<id>") {
        categorie_date = 3;
        continue;
    }
    if (categorie_date == 1) {
        date_componente += linie_citita + "\n";
    }
    else if (categorie_date == 2) {
        date_conectori += linie_citita + "\n";
    }
    else if (categorie_date == 3) {
        ultim_id = stoi(linie_citita);
        categorie_date = 0;
    }
}

```

2: Se creaza componentele si conectorii din datele citit.

Atentie, nu se completeaza datele de asociere (orice tine de pointeri la alte componente, care sunt salvate prin ID-uri) pentru ca acele componente nu exista inca, trebuie create.

De exemplu, daca am un rezistor si un bec conectate, salvate in fisier, in etapa asta creez rezistorul, becul si conectorul (cablul) dintre ele, dar nu le unesc inca, adica vor fi deconectate una de alta dar vor exista.

Conexiunea se face pe baza de id, dar pentru a le conecta trebuie intai sa existe toate componentele din fisier.

Datele marcate cu rosu sunt ignorate (nu sunt atribuite) in prima etapa de creare

Componenta	Conector
- tip id -id -pozitie -rotatie -puncte conexiune (id-urile conectorilor)	-id -start conexiune -final conexiune -pozitii

```

int camp_citit = 0;
//- 0 id_tip
//- 1 id
//- 2 poz
//- 3 rotatie
//- 4 id-uri pct conexiune

if (date_componente != "" && !eroare_citire) {
    istringstream stream_linie(date_componente);
    string linie_comp;

    Componenta* componenta_curenta = NULL;

    getline(stream_linie, linie_comp);

    while (!stream_linie.eof()) {

        istringstream stream_elemente(linie_comp);

        string pozitie;
        switch (camp_citit)
        {
            case 0:
                int tip_id;
                stream_elemente >> tip_id;
                if (tip_id < 0 || tip_id > tipuri_componente.size()) {
                    eroare_citire = true;
                    break;
                }
                componenta_curenta = new Componenta(tipuri_componente[tip_id]);

                break;
            case 1:
                int id;
                stream_elemente >> id;
                if (tip_id < 0) {
                    eroare_citire = true;
                    break;
                }
                id = 0;
                componenta_curenta->id = id;
                break;
            case 2:
                stream_elemente >> pozitie;
                componenta_curenta->SetPozitie(ParseazaPozitie(pozitie));
                break;
            case 3:
                int rotatie;
                stream_elemente >> rotatie;
                componenta_curenta->rotatie = (ORIENTARE)rotatie;
                if (rotatie > 3) {
                    eroare_citire = true;
                    break;
                }
                break;
            case 4:
                //nimic momentan
                conexiuni_componente += linie_comp + '\n';

                //reseteaza campul pentru urmatoarea componenta. adauga componenta
                camp_citit = -1;
                componente_citite.push_back(componenta_curenta);
                componenta_curenta = NULL;
                break;
            default:
                break;
        }

        camp_citit++;
        getline(stream_linie, linie_comp);
    }
}

```

iar codul pentru creat Conectori functioneaza analog, doar ca folosind datele despre conectori.

3. Se parcurg datele de conectare si se fac conexiunile care depind de ID-uri:

```
if (conexiuni_componente != "" && !eroare_citire) {
    istringstream stream_linie(conexiuni_componente);
    istringstream stream_elemente;
    string linie;
    for (auto &comp:componente_citite)
    {
        getline(stream_linie, linie);
        stream_elemente = istringstream(linie);
        for (int i = 0; i < comp->nr_pct_conexiune; i++)
        {
            int id_conector;
            stream_elemente >> id_conector;
            if (id_conector == -1) {
                comp->puncte_conexiune[i].conector = NULL;
            }
            else {
                if (id_conector > ultim_id) {
                    eroare_citire = true;
                    break;
                }
                comp->puncte_conexiune[i].conector = (Conector*)GasesteComponentaDupaID(&conectori_cititi, id_conector);
            }
        }
    }
}
```

conexiuni_componente reprezinta din fisierul de salvare doar liniile care contin datele de conectare ale componentelor.

le-am alcatuit in etapa anterioara cand atunci cand citeam datele de conectare, nu facem nimic, si doar le adaugam la textul cu date de conectare

```
case 4:
    //nimic momentan
    conexiuni_componente += linie_comp + '\n';

    //reseteaza campul pentru urmatoarea componenta. adauga componenta curenta in lista
    camp_citit = -1;
    componente_citite.push_back(componenta_curenta);
    componenta_curenta = NULL;
    break;
```

La final se verifica daca nu exista vreo eroare de formatare in unul dintre pasii anteriori si se creaza lista de componente, care se atribuie listei de componente din UIManager.h

```
if (!eroare_citire) {
    while (!toate_componentele.empty()) {
        editor_componente::StergeComponenta(toate_componentele.front());
    }
    for (auto& comp : componente_citite)
    {
        toate_componentele.push_back(comp);
    }
    for (auto& comp : conectori_cititi)
    {
        toate_componentele.push_back(comp);
    }

    SetIdActualComponente(ultim_id);

    printf("Incarcare circuit\n");
    RefreshUI();
}
else {
    printf("Eroare Load: Fisierul nu este un save file valid sau formatarea a fost corupta.");
}
```