

UNIVERSITATEA „POLITEHNICA” DIN BUCUREȘTI
FACULTATEA TRANSPORTURI

Departamentul Telecomenzi și Electronică în Transporturi

Sisteme de dirijare a traficului feroviar-Proiect

Coordonator științific
Conf.Dr.Ing. Andrei Razvan
GHEORGHIU

Student
Radu-Marinel BUCȘE

București
2021-2022

Cuprins

CAPITOLUL 1. TEMA PROIECTULUI.....	1
CAPITOLUL 2. CONȚINUTUL PROIECTULUI	2
CAPITOLUL 3. DESCRIEREA PROIECTULUI	3
CAPITOLUL 4. PREZENTAREA SCHEMEI BLOC PROPUSE CU EXPLICAREA BLOCURILOR COMPONENTE ȘI DEFINIREA FUNCȚIILOR ACESTORA	4
4.1 SCHEMA BLOC	4
4.1.1 Bilet.....	4
4.1.2 Bloc de interacțiune cu călătorii	4
4.1.3 Bloc date pentru comparare	4
4.1.4 Bloc de stocare și trimitere date	4
4.1.5 Bloc de primire date (Dispecer)	4
CAPITOLUL 5. PREZENTAREA SCHEMEI DE AMPLASAMENT A ECHIPAMENTELOR UTILIZATE.....	5
CAPITOLUL 6. DESCRIEREA COMPONENTELOR	6
6.1 BILET.....	6
6.2 VALIDATOR.....	7
6.2.1 Modulul incorporat Maikrt QR	7
6.3 LEDURI	8
6.4 PLACA DE DEZVOLTARE ARDUINO MEGA 2560.....	9
6.5 LAPTOP	11
6.6 DISPECER	12
6.7 ALIMENTAREA COMPONENTELOR.....	12
CAPITOLUL 7. SCHEMA ELECTRICĂ.....	14
7.1 SCHEMA ELECTRICĂ A VALIDATORULUI ȘI SISTEMULUI DE AVERTIZARE LUMINOASĂ	14
7.2 SCHEMA ELECTRICĂ LA NIVELUL SLAVE/MASTER.....	15
CAPITOLUL 8. PROGRAM SOFTWARE	16
8.1 COD VALIDATOR ȘI SISTEM SEMNAL LUMINOS	16
8.2 COD PENTRU MEGAMASTER	17
8.3 COD PENTRU MEGAMASTERFINAL	18

8.4	COD PENTRU LAPTOP.....	18
CAPITOLUL 9.	CONCLUZII	19
CAPITOLUL 10.	BIBLIOGRAFIE.....	19

Capitolul 1. Tema proiectului

Să se proiecteze un sistem de control automat al corectitudinii validării biletelor în tren, cu verificare în timp real într-un dispecer. Sunt obligatorii următoarele:

a. Precizarea surselor de date și amplasamentele acestora, precum și modalitatea de conectare (securizată) la acestea. În cazul în care se utilizează baze de date, este necesară precizarea structurii bazei de date necesare.

b. Pentru comunicații se va defini conținutul mesajului care se va transmite și se vor propune modalități de securizare a acestuia.

c. Componentele vor fi identificate în funcție de parametrii care au importanță pentru proiect. Se vor detalia soluțiile tehnice, inclusiv conectarea și, dacă este cazul, programarea modulelor utilizate. În acest sens, nu este admisibilă includerea unor elemente luate „de-a gata”, fără a explica în mod detaliat modalitatea în care pot fi integrate hardware și software în sistemul propus.

d. Justificarea soluțiilor alese: pentru fiecare se vor prezenta avantaje și dezavantaje.

e. Analiza fiabilității sistemului rezultat (evaluarea elementelor/componentelor care se pot defecta și impactul acestor defecțiuni asupra funcționării sistemului).

Pe baza notațiilor:

- nn = numărul de litere din nume (primul, dacă sunt mai multe)= 5;

- np = numărul de litere din primul prenume=4

Se consideră următoarele:

- Viteza de circulație între stații este cuprinsă între $\min(nn, np) \cdot 10$ km/h și $(nn + np) \cdot 10$ km/h. Pentru comunicații se va ține cont de specificul circulației feroviare, incluzând tuneluri, sau zone fără acoperire GSM .

$(nn, np) \cdot 10$ km/h și $(nn + np) \cdot 10$ km/h = 40km/h și 90km/h

- Tensiunea de alimentare pentru toate echipamentele noi introduse în sistem este de $\min(nn, np) \cdot 20$ volți. Dacă nn este impar alimentarea disponibilă este de c.c., dacă nn este par, alimentarea disponibilă este de c.a.

$\min(nn, np) \cdot 20 = \min(5, 4) \cdot 20 = 4 \cdot 20 = 80$ volți, curent continuu

Capitolul 2. Conținutul proiectului

- a. Prezentarea schemei bloc propuse cu explicarea blocurilor componente și definirea funcțiilor acestora.
- b. Prezentarea schemei de amplasament a echipamentelor utilizate.
- c. Descrierea soluției alese pentru realizarea fizică a fiecărui bloc din cele prezentate anterior: justificarea alegerii, schemă electrică, calcule pentru dimensionarea elementelor componente, prezentarea datelor de intrare și ieșire necesare, elemente de fiabilitate etc.
- d. Proiectarea interfețelor dintre blocurile funcționale – dacă este cazul.
- e. Prezentarea programului software (limbaj de asamblare, C/ μ C, pseudocod, schemă logică etc.) – dacă se utilizează microcontroler/microprocesor/placă de dezvoltare.
- f. Prezentarea schemei complete rezultate pentru realizarea temei.

Capitolul 3. Descrierea proiectului

Proiectul va avea în vedere un sistem de control automat al corectitudinii validării biletelor în tren, cu verificare în timp real într-un dispecer. Considerăm că trenul este de tip automotor ce are în componență 80 de locuri, astfel încât va fi montat câte un cititor la fiecare 4 scaune (un grup). Fiecare loc în parte va avea un număr unic și câte două leduri, acestea făcând parte dintr-un sistem de acționare luminoasă. Acel cititor va trimite datele către un microprocesor care va compara datele citite cu datele predefinite; spre exemplu datele predefinite prevăzute unui grup 1,2,3,4 dacă se citește 5, toate ledurile prevăzute fiecărui scaun devin roșii, în cazul citirii unei valori din grup, scaunul respectiv va fi luminat cu verde. Datele locurilor libere pentru fiecare vagon în parte vor fi transmise către dispecer, dispecer care este considerat în fiecare dintre gări astfel încât la momentul pierderii semnalului de la un dispecer, urmatorul să-l preia. În fiecare vagon vor fi 20 de cititoare, fiecare dintre ele transmițând datele către aceeași unitate de stocare (adică blocul de stocare și trimitere date), care la rândul ei va trimite datele dispecerilor. Considerăm că dispecerii vor avea implementat un sistem de comunicație tip internet.

Capitolul 4. Prezentarea schemei bloc propuse cu explicarea blocurilor componente și definirea funcțiilor acestora

4.1 Schema Bloc

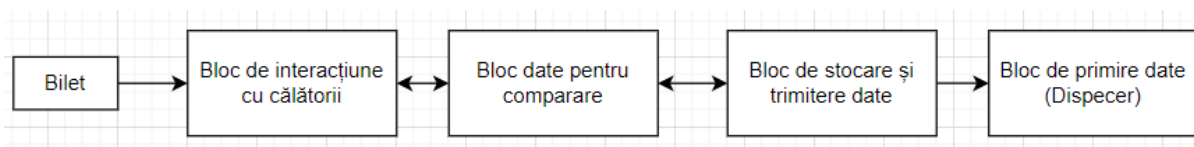


Figura 1. Schema Bloc

4.1.1 Bilet

Acest bloc este reprezentat de biletul de călătorie.

4.1.2 Bloc de interacțiune cu călătorii

Blocul de interacțiune cu călătorii este format atât dintr-un sistem de validare al biletelor, cât și dintr-un sistem de avertizare luminoasă a corectitudinii validării acestora. Acest bloc are rolul de a verifica corectitudinea validării biletelor cu ajutorul unui validator și totodată de a atenționa în momentul scanării biletului printr-un semnal luminos provenit de la un led. În momentul scanării biletului, dacă numărul locului corespunde cu cel de pe bilet, led-ul se va aprinde “verde”, în caz contrar se va aprinde “roșu” validarea fiind incorectă, călătorul urmând să-și caute locul în altă parte a vagonului.

4.1.3 Bloc date pentru comparare

Blocul datelor de comparare are ca funcție compararea datelor citite cu datele predefinite.

4.1.4 Bloc de stocare și trimitere date

Blocul de stocare și trimitere date are ca rol stocarea și trimiterea datelor privind locurile libere din fiecare vagon.

4.1.5 Bloc de primire date (Dispecer)

Blocul de primire date (Dispecer) are rolul de a primi în timp real toate validările care se realizează la nivelul fiecărui tren.

Capitolul 5. Prezentarea schemei de amplasament a echipamentelor utilizate



Figura 2. Locomotivă



Figura 3. Vagon cu capacitate 80 de locuri

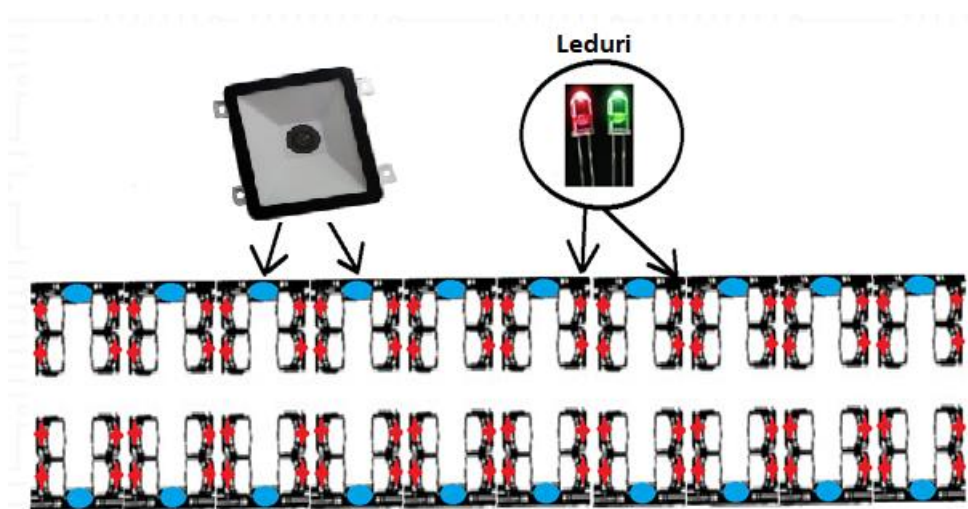


Figura 4. Secțiune vagon formată din locuri, cititoare și leduri ale sistemului de avertizare
Conform figurii 4, fiecare grup de scaune are în componența lui câte un led aflat pe spătar și un cititor/validator de bilete aflat pe partea dinspre geam a fiecărui vagon. Călătorul după ce o să

se urce în tren, va trebui să își valideze biletul, îl va apropia de cititor/validator și va afla în urma semnalului luminos al ledurilor dacă este în regulă unde se va așeza, în caz contrar acesta își va căuta locul mai departe în vagon.

Observație: Acest sistem prezintă un dezavantaj care trebuie menționat. Există o probabilitate ca unii călători să urce în vagoane și să călătorească fără achiziționarea unui bilet. În cadrul acestui proiect se vor considera doar călătorii cinstiți și educați, iar cealaltă categorie de călători nu o să fie luată în calcul. Având în vedere că acest proces nu se poate automatiza 100%, se consideră un eventual gardian ce supraveghează și raportează în eventualitatea unor evenimente neplăcute.

Capitolul 6. Descrierea componentelor

6.1 Bilet

Biletul va consta dintr-o hârtie cartonată cu dimensiunea de 90x120mm ce va fi folosit pentru validarea locurilor în tren. Pe fiecare bilet vor fi înscrispionate următoarele:

- Locul de plecare/Data plecării;
- Locul sosirii/Data sosirii;
- Data emiterii biletului;
- Numărul trenului;
- Vagon;
- Loc;
- Cod QR.

Plecare:	Data plecării:...
Sosire:.....	Data sosirii:...
Data emiterii biletului: ZZ/LL/AAAA	
Numărul trenului:.....	
Vagon:.....	
Loc:.....	
	

Figura 5. Bilet călătorie

6.2 Validator

În urma eliberării biletelor, fiecare călător are datoria de a valida biletul în momentul în care se urcă în tren și ajunge în vagon pentru a verifica dacă locul corespunde cu cel de pe bilet. Astfel, dispozitivul de recunoaștere a biletului este reprezentat de un cititor tip QR, aflat pe partea dispre geam a fiecărui grup de scaune pentru a nu ocupa foarte mult loc și de a economisi spațiu între scaune. Tot acest proces nu poate însă funcționa independent ci acompaniat de un sistem de avertizare luminoasă pentru validarea/invalidarea biletului. Călătorul va apropia biletul de modulul QR, iar dacă tot acest proces decurge corect, călătorul se va putea așeza pe scaun.

6.2.1 Modulul incorporat Maikrt QR

Modulul Maikrt QR este unul incorporat de dimensiuni mici 1D/2D, perfect pentru validarea de care avem nevoie în interiorul unui vagon. . Poate fi încorporat în orice dispozitiv pentru scanarea codului de pe bilet. Utilizează algoritmul inteligent de recunoaștere a imaginii pentru citirea datelor QR.



Figura 6. Cititor Maikrt QR

Modulul are atât interfețe USB, cât și TTL pentru interfața directă cu un microcontroler. Este echipat cu un port cu 9 pini și cablu USB, ceea ce ne este convenabil să convertim între portul serial UART TTL și interfața USB.

Modulul are 9 pini dintre care folosim 4 pentru a ne lega de microcontroler. Firul roșu este pinul VCC (+5V), negru este GND. Firele verzi și albe sunt pinii Tx și Rx.

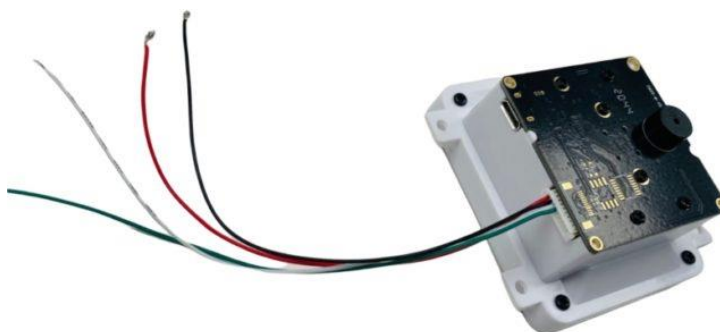


Figura 7. Reprezentarea celor 4 pini

Caracteristici:

-Alimentare: 5V

-Interfata de comunicare: UART TTL\UART RS232;

-Temperatura de operare: -25°C~55°C

6.3 Leduri

Validarea sau invalidarea unui bilet trebuie pusă în vederea călătorului folosind sistemul de avertizare luminoasă. Acesta constă în cele 2 leduri, unul verde pentru validarea corectă a biletului și unul roșu pentru validarea incorectă sau o eventuală eroare, acestea conectându-se la placa Arduino Mega2560.

Așadar, am ales să utilizez leduri cu dimensiunea de 10mm pentru a putea fi vizibile indiferent de situațiile la care sunt puse. Tensiunea de alimentare a acestora este cuprinsă între 1.5V și 2.6V, curentul de alimentare de 20mA, aceste valori putând fi furnizate de placa arduino.



Figura 8. Led roșu de 10 mm



Figura 9. Led verde de 10 mm

Ledurile nu pot fi conectate direct la placa de dezvoltare Arduino, prin urmare ele necesită rezistențe de limitare a curentului. Astfel am ales utilizarea a câte o rezistență de tip CF1/4W-120R de 120Ω; 0.25W; ±5% pentru fiecare led în parte.



Figura 10. Rezistența CF1/4W-120R

Legat de validare, trebuie menționat faptul că plăcile de dezvoltare utilizate necesită programate în vederea comunicării datelor și introducerea acestora corespunzător fiecărui bilet.

În vederea acestui proiect a fost ales un tren de tip automotor cu un număr limitat de vagoane, cu 80 de locuri disponibile călătorilor.

Așadar vom avea nevoie de 20 de cititoare, datele colectate de către acestea fiind stocate și gestionate de către un sistem de tip laptop situat în locomotivă (am considerat un laptop deoarece acesta este portabil și nu ocupă foarte mult spațiu în cabină).

Mai departe am considerat un sistem de tip master-slave, unde toate plăcuțele arduino din vagon colectează date și transmit mai departe către un Arduino Mega 2560 folosind comunicația de tip RS485.

Pentru a utiliza o conexiune de tip RS485 cu arduino avem nevoie de un modul convertor TTL la RS485. Am considerat un modul de tip MAX485.

Alimentarea modului MAX485 se face la 5V.

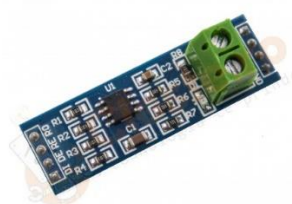


Figura 11. Modul MAX485

6.4 Placa de dezvoltare Arduino Mega 2560

De la acel Arduino Mega 2560 slave se va transmite mai departe către un alt arduino master de tip Arduino Mega 2560 ce procesează și direcționează datele către un ArduinoMega2560 master final.

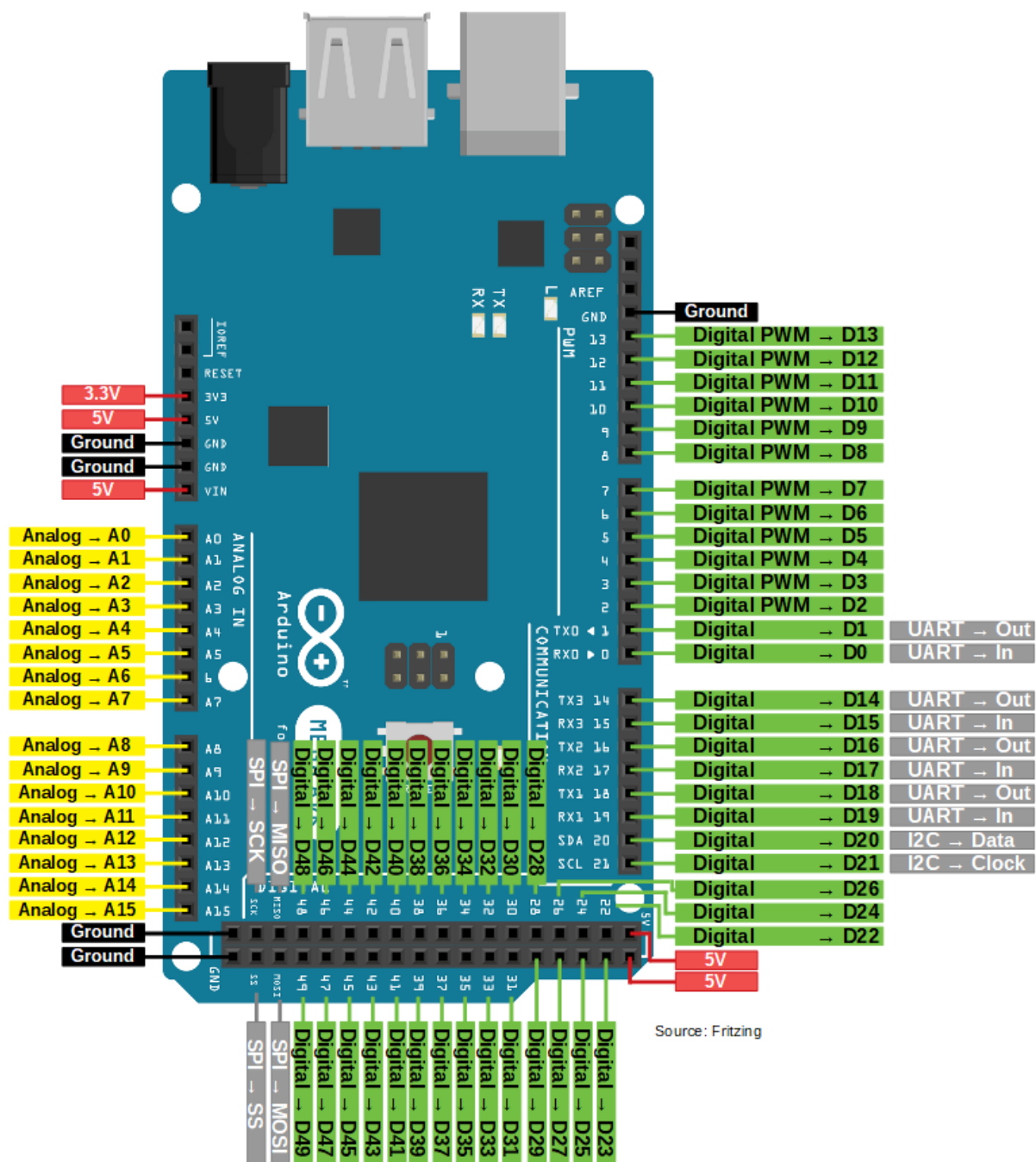


Figura 12. Arduino Mega 2560

Acest Arduino Master 2560 master final se va conecta la sistemul de stocare și gestionare date (reprezentat de un laptop) printr-o interfață USB. Astfel toate datele primite de la blocurile arduino vor ajunge la placa Arduino Master Final transmise în final într-un laptop ce vor putea fi trimise mai departe către dispecer.

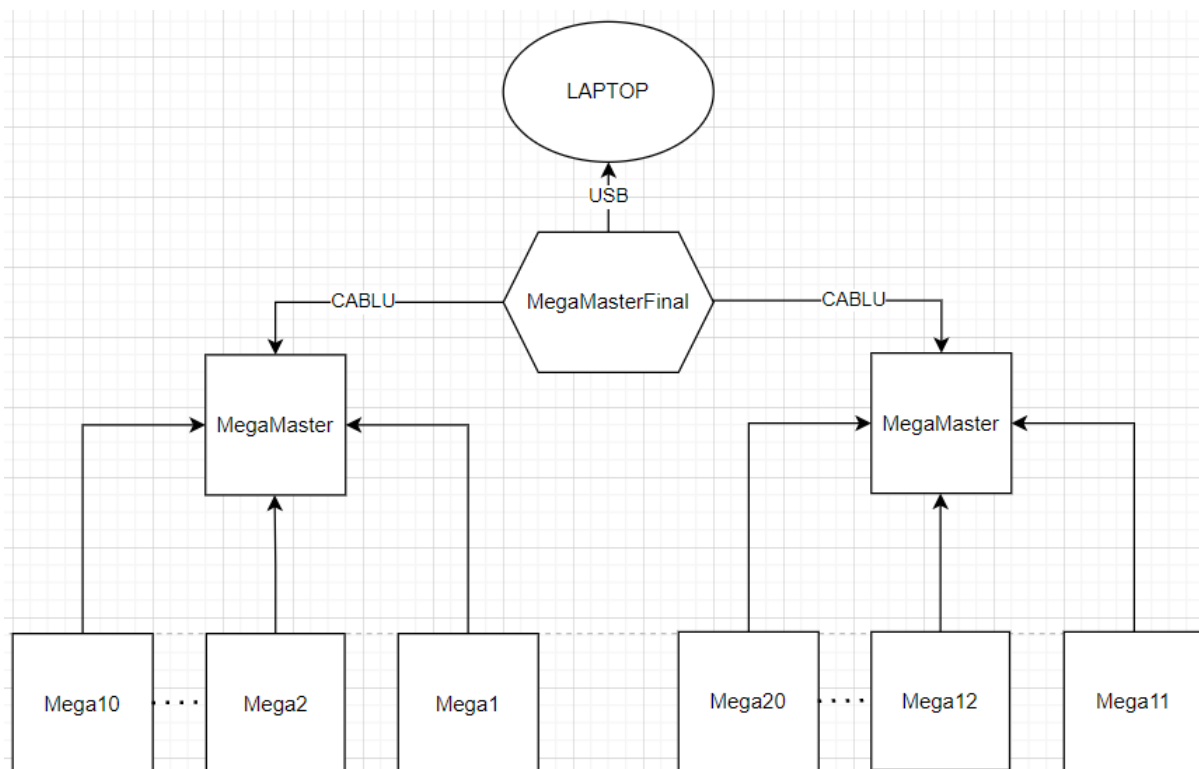


Figura 13. Schema bloc Master/Slave

6.5 Laptop

Atât pentru locomotivă cât și pentru dispecer am ales un laptop de tip Laptop ASUS F542UN-DM015 și un modem de tip Kruger&Matz 4G LTE pentru cartela SIM 4G. Acest modem o să fie conectat atât la laptop-ul din locomotivă cât și la cel de la dispecer pentru procesul de comunicație.

- Procesor: Intel® Core™ i5-8250U;
- Capacitate memorie/Tip memorie: 8GB DDR4;
- Capacitate: 1TB.



Figura 14. Laptop ASUS F542UN-DM015

Caracteristici:

- Alimentare 5V;
- Compatibilitate 3G/4G : 4G.



Figura 15. Modem Krüger&Matz 4G LTE

6.6 Dispecer

Atât laptop-ul din locomotivă, cât și cel de la dispecer vor avea pachetul Microsoft Office și Arduino IDE. De asemenea o să avem nevoie de add-in-ul DATA STREAMER, ce va transmite datele în timp real de la o placă de dezvoltare la baza de date.

Ca bază de date se va folosi programul Excel, împreună cu add-in-ul DATA STREAMER activat.

Laptopul din tren comunica cu laptopul dispecerului prin intermediul unui document Excel partajat în Google Drive.

Pentru a trimite datele către dispecer se va folosi funcția “PARTAJARE ÎN TIMP REAL”, dispecerul având acces la toate datele primite de la fiecare validator.

6.7 Alimentarea componentelor

Placa arduino are nevoie de o alimentare de 5V c.c. Alimentarea mea fiind de 80V c.c. pentru a ajunge la valoarea de 5V se vor folosi divizori de tensiune. Astfel se vor alege 2 rezistențe în următoarea configurație:

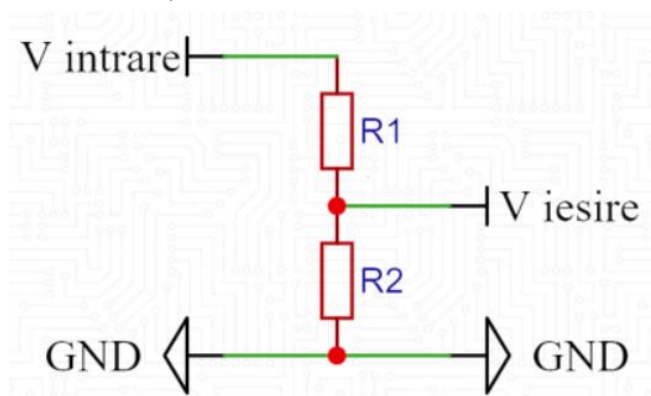


Figura 16. Configurație rezistențe

Prin urmare am ales 2 rezistențe, prima de 12 kOhm, iar cea de a doua de 806 Ohm. Prima valoare a rezistenței a fost aleasă standard, iar pentru cea de-a doua rezistența s-a calculat după formula:

$$V_{iesire} = \frac{V_{intrare} * R2}{R1 + R2}$$



Figura 17. R_1 MOR01SJ0123A10



Figura 18. R_2 MF1/4DCT52R8060F

Capitolul 7. Schema electrică

7.1 Schema electrică a validatorului și sistemului de avertizare luminoasă

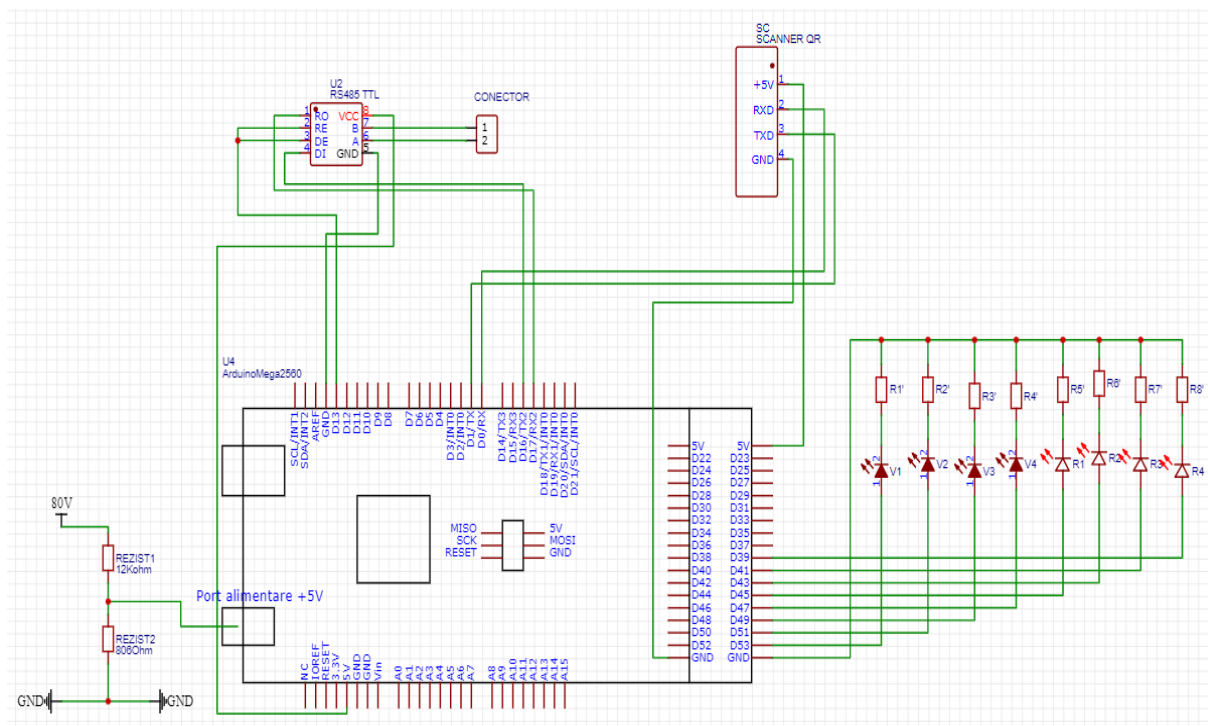


Figura 19. Schema electrică a validatorului și sistemului de avertizare luminoasă

Această schemă electrică cuprinde placa de dezvoltare Arduino Mega 2560 de tip slave, modulul Maikrt (scanner qr), 10 rezistențe, 8 dintre ele folosite pentru leduri iar celelalte 2 pentru sursa de alimentare și modulul de comunicație RS485-TTL Max485.

Toate componentele se vor conecta la placa de dezvoltare și vom utiliza 18 pini (13 pini digitali, 3 pini GND și 2 pini de alimentare) și o mufă pentru alimentarea de la sursă.

Modulul Maikrt se va conecta astfel: pinul TX la D1/TX, pinul RX la D0/TX, pinul GND la un pin GND și pinul +5V la un pin de alimentare 5V.

Ledurile nu se pot conecta direct la placa de dezvoltare, ci numai în serie cu rezistențe. Astfel vom avea catodurile ledurilor verzi V1,V2,V3,V4 conectate în serie cu rezistențele R1', R2', R3', R4' anodurile ledurilor fiind conectate în aceeași ordine la următorii pini D53, D51, D49, D47, iar ceilalți pini ai rezistențelor fiind conectate la GND. Același lucru și pentru catodurile ledurilor roșii R1, R2, R3, R4, conectate în serie cu rezistențele R5', R6', R7', R8', anodurile ledurilor fiind conectate în aceeași ordine la pinii D45, D43, D41, D39, iar ceilalți pini ai rezistenței fiind conectate la GND.

Modulul Max485 se va conecta astfel: Pinul RO la pinul D17/RX2, pinul DI la pinul D16/TX2 pinii RE și DE la pinul D13, pinul GND la GND, pinul VCC la un pin 5V, iar pinii A și B la un conector care face legătura cu următorul modul de comunicație.

7.2 Schema electrică la nivelul slave/master

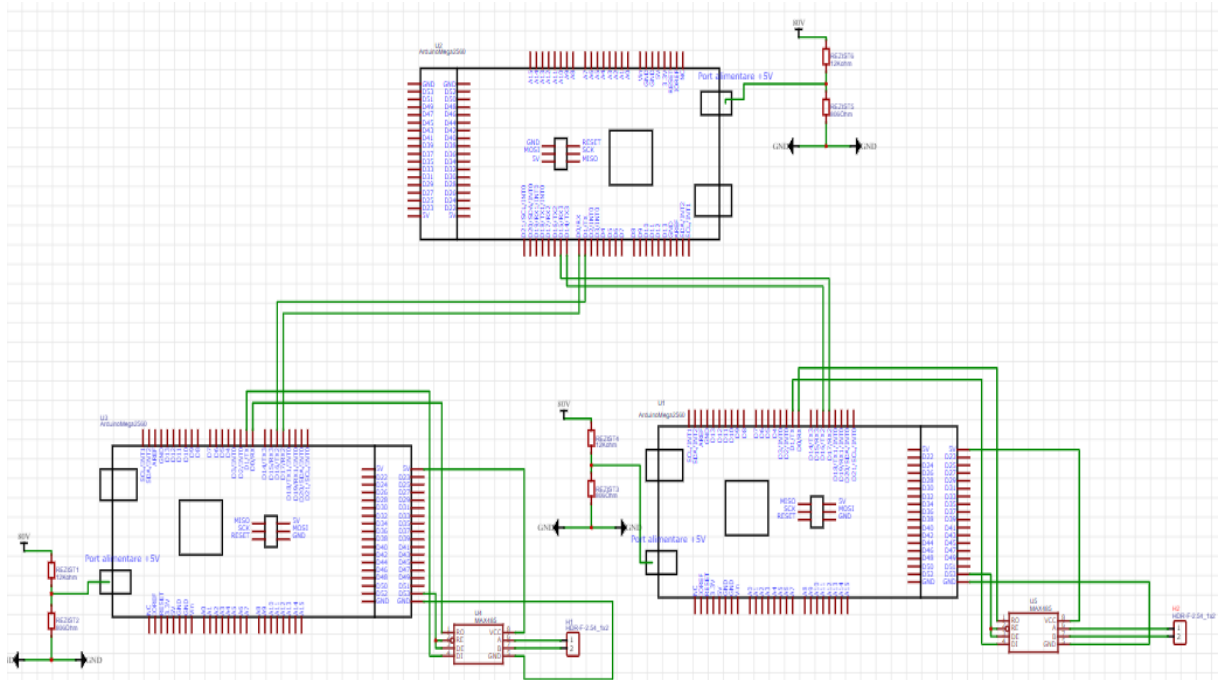


Figura 20. Schemă electrică slave/master

Această schemă cuprinde 3 plăci de dezvoltare de tip Arduino Mega 2560, 2 dintre ele reprezentând mastersele celor 20 de slave-uri, iar unul fiind master final ce are rolul de a primi/trimitte informațiile mai departe.

Mastersele slave-urilor sunt conectate la un modul de comunicație MAX485 astfel: pinul RO la pinul D0/RX, DE și RE la pinul D53, DI la pinul D1/TX, VCC la 5V, GND la GND și pinii A și B la un conector care face legătura cu modulul slave-ului.

Cele 2 masterse ale slave-urilor sunt conectate la rândul lor la masterul final astfel: Pentru masterul din stânga, pinul D16/TX2 la pinul D1/TX și pinul D17/RX2 la pinul D0/RX. Pentru cel din dreapta, pinul D16/TX2 la pinul D14/TX3 și pinul D17/RX2 la pinul D15/RX3.

Având în vedere că fiecare arduino mega master are conectat prin aceleași fire 10 slave-uri, ne vom crea propriul protocol. Acest protocol constă în interogarea a fiecărui slave de către master dacă are vreo informație de transmis, în acest fel se va evita coliziunile dintre semnale.

Capitolul 8. Program software

8.1 Cod validator și sistem semnal luminos

nr_scaun_initial_al_grupului; // variabila globala setata atunci cand arduino se porneste
numar_scaune_per_grup = 4; // numarul de scaune dirijate de fiecare arduino slave
lista_nr_scaun_ocupate = [0, 0, 0, 0]; // lista ce specifica daca un loc este ocupat sau nu (ex. lista[1] = 0 => locul cu indexul 1 este liber; lista[1] = 1 => locul cu indexul 1 este ocupat)

functie manevrare_semnal_scanner(date_scanner) // functie in care se intra atunci cand se primesc datele de la scanner

```
{  
    nr_scaun = date_scanner.nr_scaun; // numarul scaunului de pe biletul scanat  
    id_calatorie = date_scanner.id_calatorie; // identificator unic al calatoriei de pe biletul scanat  
  
    if( (not validare(id_calatorie)) or // bilet invalid(biletul este pentru alta ruta)  
        (nr_scaun < nr_scaun_initial or nr_scaun >= nr_scaun_initial +  
numar_scaune_per_grup ) ) // nr biletului nu corespunde celor 4 scaune  
    {  
        aprinde_led_rosu(0);  
        aprinde_led_rosu(1);  
        aprinde_led_rosu(2);  
        aprinde_led_rosu(3);  
        return; // se iese din executia programului(manevrarea semnalului)  
    }  
  
    numar_led_verde_aprins = nr_scaun - nr_scaun_initial_al_grupului; // al catelea led verde ar trebui aprins(ex. pentru bilet cu nr 15 si un grup cu scaunele 12-15(inclusiv), trebuie aprins ledul  $15 - 12 = 3$ )  
    aprinde_led_verde(numar_led_verde_aprins);  
  
    lista_nr_scaun_ocupate[numar_led_verde_aprins] = 1; // locul curent s-a ocupat  
}
```

functie manevrare_semnal_master() // functie in care se intra atunci cand masterul interogheaza slave-ul cu privire la locurile ocupate

```
{  
    scaune_ocupate = [] // lista care va contine numerele globale(din tren) ale locurilor ocupate(ce corespund arduino-ului curent)
```

```

j = 0;
for( i = 0; i < 4; i++) {
    if (lista_nr_scaun_ocupate[i] == 1) // daca locul cu indexul i este ocupat
    {
        scaune_ocupate[j] = i + nr_scaun_initial_al_grupului; // construiesc o
lista care o sa contina numerele globale(din tren) ale locurilor ocupate
        j = j + 1;
    }
}
trimite_catre_master(scaune_ocupate ) // trimite masterului situatia curenta a
scaunelor
}
//AICI SE INTRA LA INCEPUT IN PROGRAM
program_principal_arduino_slave(nr_scaun_initial)
{
    nr_scaun_initial_al_grupului = (nr_scaun_initial); // setarea variabilei globale -
numarul primului scaun al grupului

    //se seteaza manevrarea semnalelor pentru pinii de la arduino care comunica cu
scannerul, respectiv cu masterul
    //se va intra in aceste 2 functii(manevrare_semnal_scanner,
manevrare_semnal_master) doar atunci cand va veni un semnal pe pinii respectivi
    seteaza_manevrare_semnal(port_comunicatie_scanner, manevrare_semnal_scanner);
    seteaza_manevrare_semnal(port_comunicatie_master, manevrare_semnal_master);
}

```

8.2 Cod pentru MegaMaster

```

program_principal_mega_master(id) // id poate fi 0(primul megaMaster) sau 1(al doilea
megaMaster)
{
    while (1) { // while ul nu se termina niciodata
        for(i = id * 10 + 1; i <= (id + 1) * 10; i++) //pentru id = 0 -> for(1,10), pentru
id = 1 -> for(11,20),
        {
            scaune_ocupate = interogheaza_slave() // interoghez slaveul si primesc
o lista cu scaunele lui ocupate
            trimite_catre_megamasterfinal(scaune_ocupate) // trimit mai departe
catre MegaMasterFinal lista cu scaunele ocupate ale slave-ului pe care il interoghez la
moment curent

```

```

    }

}

```

8.3 Cod pentru MegaMasterFinal

```

functie manevrare_semnal_mega_master(scaune_ocupate)
{
    trimite_catre_laptop(scaune_ocupate) // trimite catre laptop lista de scaune ocupate
    primita
}

```

```

program_principal_mega_master_final()
{
    //se seteaza manevrarea semnalelor pentru pinii de la arduino care comunica cu cei
    doi megaMasteri
    seteaza_manevrare_semnal(port_comunicatie_mega_master_0,
    manevrare_semnal_mega_master);
    seteaza_manevrare_semnal(port_comunicatie_mega_master_1,
    manevrare_semnal_mega_master);
}

```

8.4 Cod pentru Laptop

```

lista_scaune_ocupate[81] // baza de date locala pentru tot vagonul (ex. Daca
lista_scaune_ocupate[i] == 1 -> locul i din tren este ocupat, altfel locul i este liber)
functie manevrare_semnal_mega_master_final(scaune_ocupate)
{
    // pentru fiecare scaun ocupat primit de la MegaMasterFinal, actualizeaza baza de
    date locala, adica pune 1 in lista_scaune_ocupate fiecarui scaun ocupat
    for (i = 0; i < lungime(scaune_ocupate); i++) {
        nr_scaun_occupat = scaune_ocupate[i];
        lista_scaune_ocupate[nr_scaun_occupat] = 1;
    }
    updateaza_excel() // pune datele din lista_scaune_ocupate in excel-ul de pe drive
}

program_principal_laptop()
{

```

```

for (i = 0; i <= 80; i++) {
    lista_scaune_ocupate[i] = 0; // initializarea listei cu toate scaunele din vagon
}

//se seteaza manevrarea semnalelor pentru portul usb care comunica cu arduino
megaMasterFinal
    seteaza_manevrare_semnal(port_comunicatie_mega_master_final,
manevrare_semnal_mega_master_final);
}

```

Capitolul 9. Concluzii

În concluzie, acest tip de sistem poate fi un eventual înlocuitor pentru controlori, însă nu poate proteja împotriva tuturor tipurilor de fraudă, acesta fiind aplicat într-un mediu în care oamenii sunt cinstiți și corecți, ceva utopic dacă stăm să ne gândim, mai ales pe teritoriul țării noastre. Pentru a se evita o parte din fraude, se poate lua în considerare angajarea unui eventual paznic ce are rolul de a supraveghea întregul tren și de a marca eventualele fapte de fraudă.

Capitolul 10. Bibliografie

- 1) <https://how2electronics.com/barcode-qr-code-reader-using-arduino-qr-scanner-module/>
- 2) <https://docs.arduino.cc/hacking/hardware/PinMapping2560>
- 3) <https://www.tme.eu/ro/details/1w-12k/rezistente-metalizate-tht-1w/royal-ohm/mor01sj0123a10/>
- 4) <https://ro.mouser.com/ProductDetail/KOA-Speer/MF1-4DCT52R8060F?qs=XQnoX2oBLyAEhA7ITzm7nQ%3D%3D>
- 5) https://www.tme.com/ca/en/details/cf1_4w-120r/1-4w-carbon-tht-resistors/sr-passives/
- 6) <https://3dstar.ro/calculator-divizor-tensiune>