Carla-Gabriela FLORICEL (78501) - Intersectie semaforizata

Autorul poate fi contactat la adresa: **Login pentru adresa**

Introducere

Proiectul simuleaza functionarea unei intersectii semaforizate in cruce pentru pietoni si masini.

Descriere generală

Exista, astfel, in intersectie 4 semafoare pentru masini cu 3 culori (rosu, galben, verde) si 8 semafoare pentru pietoni cu 2 culori (rosu, verde). Se folosesc 2 butoane pentru schimbarea functionarii normale a semafoarelor in modul intermitent si invers. De asemenea se vor monta 4 butoane, cate unul pentru fiecare semafor, ce odata apasate vor afisa culoarea si secundele pana la schimbarea culorii unui semafor pe un LCD.

Motivatie

Am dorit sa programez functionarea semafoarelor unei intersectii cu ajutorul unei matrice de leduri simple si mi s-a parut interesanta aceasta idee de proiect.

Hardware Design

Lista piese:

Număr piese	Nume piesă	Observații
1	Placă de bază cu ATMega324A-PU	Alimentare la 5V
28	led-uri	12 verzi, 12 rosii, 4 galbene
4	Butoane	Pentru afisare pe LCD
8	Distantieri	Stabilitate suport intersectie
2	Placuta test mari	Folosite pentru suport
1	Afisaj LCD	Afiseaza secundele de la un semafor
4	Fotorezistente	Anunta cand e masina in intersectie
4	Rezistente 1k	Pentru legarea ledurilor la placa de baza
4	Rezistente 10k	Pentru legarea fotorezistentelor la placa de baza
1	Fir de alimentare	

	Fire de legatura	

Schema bloc:

Schema electrica :

Poze

Filmulet

https://drive.google.com/file/d/13j7Onf0O-JUSTXn2kzfnLxVuQ0VYCeHA/view

Software Design

Structura proiectului

Pentru a face sa se aprinda semafoarele concomitent cum doream, am aprins si stins in loop-uri ledurile foarte rapid fara a se vedea cu ochiul liber.

Restul software-ului (LCD,joc intermitent pt leduri etc) va fi gata pana la PM fair

Aprindere leduri si functionarea semafoarelor concomitent

main.c

```
int main(void)
{
                //setare pini pt semafoare ca output
                DDRB |= (1 << PB0);
                PORTB &= ~(1 << PB0);
                DDRB |= (1 << PB1);
                PORTB &= ~(1 << PB1);
                DDRB |= (1 << PB2);
                PORTB &= ~(1 << PB2);
                DDRB |= (1 << PB3);
                PORTB &= ~(1 << PB3);
                DDRB |= (1 << PB4);
                PORTB &= ~(1 << PB4);
                DDRB |= (1 << PB5);
                PORTB &= ~(1 << PB5);
```

```
                DDRB |= (1 << PB6);
                PORTB &= ~(1 << PB6);
                DDRB |= (1 << PB7);
                PORTB &= ~(1 << PB7);

        while(1) {

                int i=0;

                        //rosu la semafor1 pt masini
                        for(i=0; i<700;i++){

                                PORTB |= (1 << PB0);
                                PORTB &= ~(1 << PB1);
                                PORTB &= ~(1 << PB2);
                                PORTB &= ~(1 << PB3);
                                PORTB &= ~(1 << PB4);
                                PORTB |= (1 << PB5);
                                PORTB |= (1 << PB6);
                                PORTB |= (1 << PB7);

                                _delay_ms(70);

                                PORTB |= (1 << PB0);
                                PORTB &= ~(1 << PB1);
                                PORTB &= ~(1 << PB2);
                                PORTB &= ~(1 << PB3);
                                PORTB |= (1 << PB4);
                                PORTB |= (1 << PB5);
                                PORTB &= ~(1 << PB6);
                                PORTB |= (1 << PB7);

                                _delay_ms(70);

                                PORTB &= ~(1 << PB0);
                                PORTB |= (1 << PB1);
                                PORTB &= ~(1 << PB2);
                                PORTB &= ~(1 << PB3);
                                PORTB |= (1 << PB4);
                                PORTB &= ~(1 << PB5);
                                PORTB |= (1 << PB6);
                                PORTB |= (1 << PB7);

                                _delay_ms(70);

                                PORTB &= ~(1 << PB0);
                                PORTB |= (1 << PB1);
                                PORTB &= ~(1 << PB2);
                                PORTB &= ~(1 << PB3);
                                PORTB |= (1 << PB4);
                                PORTB |= (1 << PB5);
```

```
                        PORTB |= (1 << PB6);
                        PORTB &= ~(1 << PB7);

                        _delay_ms(70);


                        PORTB &= ~(1 << PB0);
                        PORTB &= ~(1 << PB1);
                        PORTB |= (1 << PB2);
                        PORTB &= ~(1 << PB3);
                        PORTB |= (1 << PB4);
                        PORTB |= (1 << PB5);
                        PORTB &= ~(1 << PB6);
                        PORTB |= (1 << PB7);

                        _delay_ms(70);

                        PORTB &= ~(1 << PB0);
                        PORTB &= ~(1 << PB1);
                        PORTB |= (1 << PB2);
                        PORTB &= ~(1 << PB3);
                        PORTB |= (1 << PB4);
                        PORTB |= (1 << PB5);
                        PORTB |= (1 << PB6);
                        PORTB &= ~(1 << PB7);

                        _delay_ms(70);
                }




                //galben la semafor1 pt masini
                        for(i=0; i<300;i++){

                        PORTB |= (1 << PB0);
                        PORTB &= ~(1 << PB1);
                        PORTB &= ~(1 << PB2);
                        PORTB &= ~(1 << PB3);
                        PORTB &= ~(1 << PB4);
                        PORTB |= (1 << PB5);
                        PORTB |= (1 << PB6);
                        PORTB |= (1 << PB7);

                        _delay_ms(70);

                        PORTB |= (1 << PB0);
                        PORTB &= ~(1 << PB1);
                        PORTB &= ~(1 << PB2);
                        PORTB &= ~(1 << PB3);
```

```
                                PORTB |= (1 << PB4);
                                PORTB |= (1 << PB5);
                                PORTB &= ~(1 << PB6);
                                PORTB |= (1 << PB7);

                                _delay_ms(70);

                                PORTB |= (1 << PB0);
                                PORTB &= ~(1 << PB1);
                                PORTB &= ~(1 << PB2);
                                PORTB &= ~(1 << PB3);
                                PORTB |= (1 << PB4);
                                PORTB &= ~(1 << PB5);
                                PORTB |= (1 << PB6);
                                PORTB |= (1 << PB7);

                                _delay_ms(70);

                                PORTB |= (1 << PB0);
                                PORTB &= ~(1 << PB1);
                                PORTB &= ~(1 << PB2);
                                PORTB &= ~(1 << PB3);
                                PORTB |= (1 << PB4);
                                PORTB |= (1 << PB5);
                                PORTB |= (1 << PB6);
                                PORTB &= ~(1 << PB7);

                                _delay_ms(70);

                                PORTB &= ~(1 << PB0);
                                PORTB &= ~(1 << PB1);
                                PORTB &= ~(1 << PB2);
                                PORTB |= (1 << PB3);
                                PORTB &= ~(1 << PB4);
                                PORTB |= (1 << PB5);
                                PORTB |= (1 << PB6);
                                PORTB |= (1 << PB7);

                                _delay_ms(70);

                                PORTB &= ~(1 << PB0);
                                PORTB &= ~(1 << PB1);
                                PORTB &= ~(1 << PB2);
                                PORTB |= (1 << PB3);
                                PORTB |= (1 << PB4);
                                PORTB |= (1 << PB5);
                                PORTB |= (1 << PB6);
                                PORTB &= ~(1 << PB7);

                                        _delay_ms(70);
                                }
```

```
                //verde la semfor1 pt masini
                for(i=0; i<700;i++){

                        PORTB |= (1 << PB0);
                        PORTB &= ~(1 << PB1);
                        PORTB &=~(1 << PB2);
                        PORTB &= ~(1 << PB3);
                        PORTB |= (1 << PB4);
                        PORTB &= ~(1 << PB5);
                        PORTB |= (1 << PB6);
                        PORTB |= (1 << PB7);

                        _delay_ms(70);

                        PORTB |= (1 << PB0);
                        PORTB &= ~(1 << PB1);
                        PORTB &= ~(1 << PB2);
                        PORTB &= ~(1 << PB3);
                        PORTB |= (1 << PB4);
                        PORTB |= (1 << PB5);
                        PORTB |= (1 << PB6);
                        PORTB &= ~(1 << PB7);

                        _delay_ms(70);

                        PORTB &= ~(1 << PB0);
                        PORTB |= (1 << PB1);
                        PORTB &= ~(1 << PB2);
                        PORTB &= ~(1 << PB3);
                        PORTB &= ~(1 << PB4);
                        PORTB |= (1 << PB5);
                        PORTB |= (1 << PB6);
                        PORTB |= (1 << PB7);

                        _delay_ms(70);

                        PORTB &= ~(1 << PB0);
                        PORTB |= (1 << PB1);
                        PORTB &= ~(1 << PB2);
                        PORTB &= ~(1 << PB3);
                        PORTB |= (1 << PB4);
                        PORTB |= (1 << PB5);
                        PORTB &= ~(1 << PB6);
                        PORTB |= (1 << PB7);

                        _delay_ms(70);

                        PORTB &= ~(1 << PB0);
                        PORTB &= ~(1 << PB1);
```

```
                                    PORTB |= (1 << PB2);
                                    PORTB &=~(1 << PB3);
                                    PORTB &= ~(1 << PB4);
                                    PORTB |= (1 << PB5);
                                    PORTB |= (1 << PB6);
                                    PORTB |= (1 << PB7);

                                    _delay_ms(70);

                                    PORTB &= ~(1 << PB0);
                                    PORTB &= ~(1 << PB1);
                                    PORTB |= (1 << PB2);
                                    PORTB &= ~(1 << PB3);
                                    PORTB |= (1 << PB4);
                                    PORTB &= ~(1 << PB5);
                                    PORTB |= (1 << PB6);
                                    PORTB |= (1 << PB7);

                                    _delay_ms(70);
                                    }

            }
}
```

Rezultate Obţinute

O mini intersectie in cruce cu un joc distractiv de leduri si posibilitatea vizualizarii unui contor de timp pe LCD.

Download

Resurse software:

resurse_pm.zip

Concluzii

A fost destul de imbarligata si de lunga durata partea Hardware, nu recomand inceperea ei fara un plan bine facut dinainte pentru ca altfel va fi foarte dificil de realizat legaturile intre toate firele.

Jurnal

Saptamana 1: Realizare schema bloc si inceput placa de baza.

Saptamana 2: Schema electrica si terminare placa de baza.

Saptamana 3: Achizitionare piese necesare.

Saptamana 4: Terminare Hardware.

Saptamana 5: Terminare Software.

Bibliografie/Resurse

-Laboratoarele au fost principalele resurse software

-https://learn.sparkfun.com/tutorials/tags/eagle

-http://www.instructables.com/id/Traffic-Light-Project-1/

* Documentația în format PDF