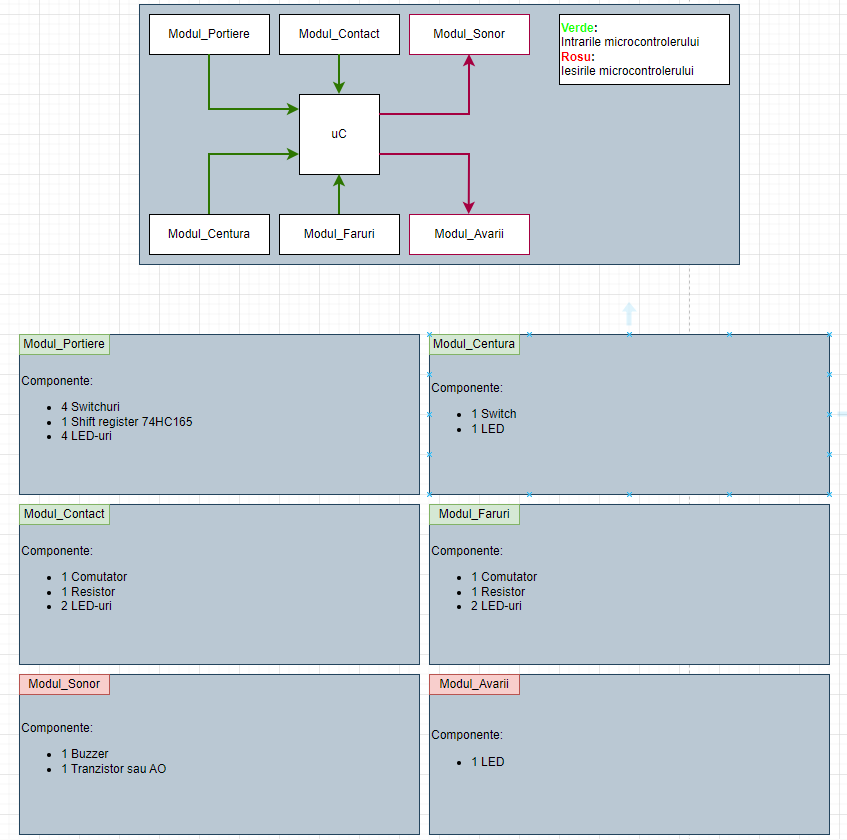
Proiectarea Unui Circuit de Avertizare Sonora

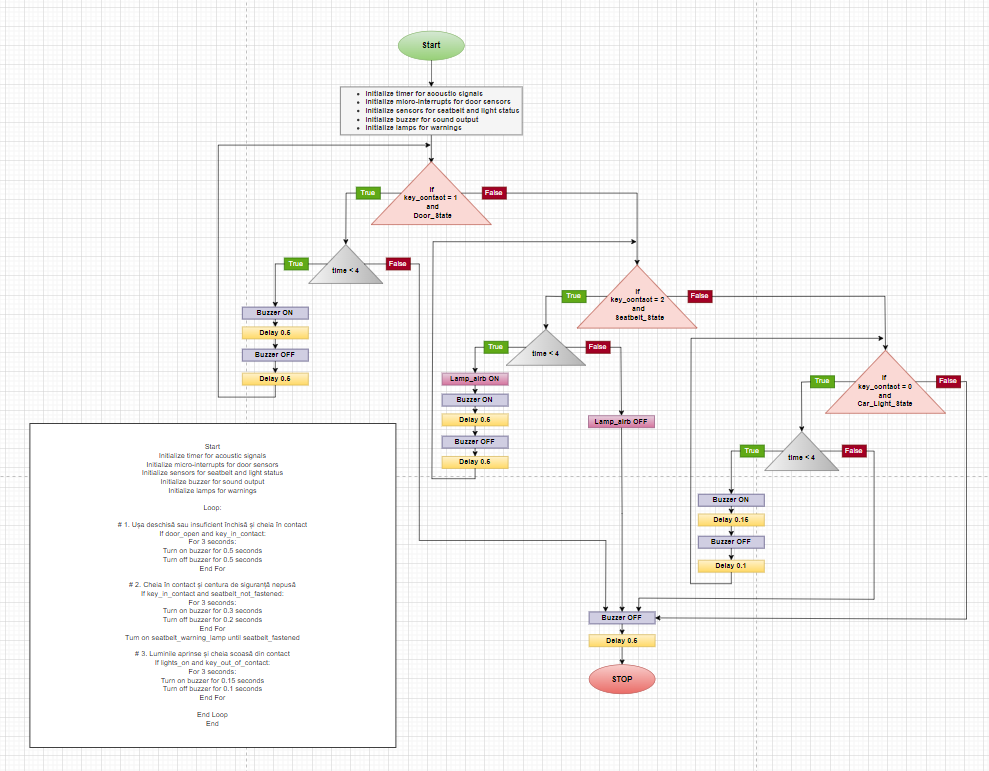
Realizarea circuitului s-a efetuat, atât la nivel de simulare cât şi în format fizic. Pentru a proiecta şi simula circuitul, am folosit Proteus unde am realizat circuitul electric, fisierele codului sursa si proiectarea PCB-ului.

Pentru realizarea circuitului am folosit un microcontroler PIC16F887, 5 butoane, 5 leduri si un buzzer.

Schema bloc a circuitului este afisata in imaginea de mai jos. Acolo se poate vedea interactiunea dintre componentele electrice, prin urmarirea sagetilor se poate obeserva de asemenea modul de comunicatie intre module.



Arhitectura codului sursa a fost realizata urmand schema bloc si pseudocodul din imaginile de mai jos. Urmand ca pentru fiecare functie sa fie realizata o logica interna.



Cod Sursa:

#define \_XTAL\_FREQ 8000000

#define DOOR\_SENSOR RB0

#define KLEM1\_SENSOR RB4

#define KLEM2\_SENSOR RB5

#define BELT\_SENSOR RB6

#define LIGHT\_SENSOR RB7

#define DOOR\_LIGHT RA0

#define BELT\_LIGHT RA1

#define LIGHTS\_LIGHT RA2

#define WARNING\_LIGHT RA3

#define PWM\_DS CCPR1L

char door\_status = 0;

char belt\_status = 0;

char light\_status = 0;

char door\_pressed = 0;

* \_XTAL\_FREQ 🡪 seteaza frecventa uC la 8 kHz
* DOOR\_SENSOR 🡪 reprezinta feedback-ul primit de uC de la portierele masinii
* KLEM1\_SENSOR 🡪 reprezinta prima pozitie a contactului unui auovehicul
* KLEN2\_SENSOR 🡪 reprezinta a doua pozitie pe care o poate avea contactul unui autovehicul
* BELT\_SENSOR 🡪 este pinul de la uC care confirma daca este conectata centura de siguranta
* LIGHT\_SENSOR 🡪 este pinul de la uC care verifica daca farurile au fost sau nu pornite

void init()

{

    TRISC = 0x00;              // Configurare RC2 ca iesire (PWM)

    PR2 = 0xFF;                // Setare perioada PWM

    CCP1CON = 0b00001100;       // Modul PWM activat

    T2CON = 0b00000100;         // Timer2 pornit, prescaler 1:1

    PWM\_DS = 0x00;              // Duty cycle ini?ial

    TRISB = 0b11111111;

    TRISA = 0b00000000;         // RA0 configurat ca ie?ire

    ANSELH = 0x00;              // Dezactiveaza func?iile analogice pentru AN8-AN13

    ANSEL = 0x00;               // Dezactiveaza func?iile analogice pentru AN0-AN7

}