

CUPRINS

Capitolul I. Introducere.	3
1.1. Definirea temei.	3
1.2. Scopul.	3
Capitolul II. Componentele electronice folosite la acest proiect	5
2.1. Descriere.	5
2.2. Driver Motor.	7
2.2. Detectorul de Infrarosi (IR).	7
2.3. Emitator IR.	9
2.4. Oscilator la 38KHz.	11
2.5. Detectorul de lumina.	11
2.7. Unitatea de control	13
2.7.1. Descriere.	16
2.7.2. Modulul CAPCOM.	18
2.7.3. Pulse Width Modulation.	21
2.7.4. CAN.	21
Capitolul III. Schema bloc a sistemului. Principii de functionare.	22
3.1 Descriere.	22
3.2. Circuit detectie obiecte.	23
3.2.1. Ir LED.	23
3.2.2. Oscilator de 38KHz.	24
3.2.3. Comanda emitator IR.	25
3.2.4. Comanda putere senzor IR.	27
3.2.5. Senzor de IR.	27
3.3. Circuit detectie lumina.	29
3.4. Sistemul de comanda si control al motoarelor.	31
3.4.1. Descriere.	32
3.4.2. Unitatea de control.	33
3.4.3. Porile logice SI-NU.	34
3.4.4. Drivelele.	36
3.4.5. Motoarele.	38

Capitol IV. Realizarea mecanica a masinii.	41
4.1. Descriere.	41
4.2. Constructia masinii.	41
Capitolul V. Geometria miscarii propriu-zise.	43
5.1. Descriere.	43
5.2. Ghidarea catre lumina.	43
5.2.1. Deplasarea in fata.	44
5.2.2. Deplasarea in lateral.	44
5.2.3. Miscarea in care se cauta lumina.	45
5.3. Evitarea obstacolelor.	46
5.3.1. Obiectul ce trebuie evitat este detectat in partea stanga.	46
5.3.2. Obiectul ce trebuie evitat este detectat in partea dreapta.	48
5.3.3. Obiectul se afla in zona in care emit cele 2 leduri de IR.	48
Capitolul VI. Realizarea practica a softwarului de comanda si control	49
6.1 Fluxul de informatii.	49
6.2. Prezentarea procedurilor de comanda.	51
Capitolul VII. Concluzii	59
BIBLIOGRAFIE	61

Capitolul I.

Introducere.

1.1. Definirea temei.

În prezenta lucrare descriem modul în care am realizat sistemul automat de orientare și funcționarea acestuia. Sistemul pe care încercăm să îl conducem este format din mai multe module, toate acestea unite creând o mașinuță.

Este vorba de o mașină ce se poate orienta după lumina și tot o dată, în drumul către sursa de lumină să ocolească obiectele ce sunt în calea acesteia. Ideea a plecat de la întâi de la simpla orientare a mașinii după lumină cu ajutorul a doi senzori. Deoarece acest lucru era destul de simplu de realizat sau adăugat și doi senzori de proximitate. Astfel mașina reușește să urmărească o sursă de lumină mobilă. Ea poate lua decizii în orice moment de schimbare a direcției dacă poziția sursei de lumină se modifică. Aceasta mașină comunică doar cu microcontrolerul, o dată ce sistemul este pornit este capabilă să ia singura decizie de urmărire respectiv ocolire de obstacole. Aici avem un sistem automat de reglare.

Deciziile pe care microcontrolerul le ia sunt direct dependente de stimulii mediului. Mașina interacționează cu mediul primind permanent informații prin senzorii atașați pe ea. Toate aceste informații primite prin senzori sunt procesate rezultatul lor fiind mișcarea mașinii cu ajutorul celor două roți.

1.2. Scopul.

Ne dorim ca prin realizarea acestui proiect să punem în evidență ușurința cu care anumite tehnologii destul de simple la prima vedere pot crea un mecanism destul de “inteligent” și atractiv pentru oameni. Pe lângă partea distractivă a acestui proiect acest timp de roboți autonomi au o răspândire destul de mare, ei regăsindu-se în foarte multe domenii.

Cel mai important lucru la acest timp de roboti este ca el nu are oameni la bord. Acest lucru ii confera o autonomie crescuta, deoarece in vehiculul respectiv nu mai trebuie creat un habitacul pentru oameni, acest lucru reducand mult si costurile. In domeniul militar, si-au dovedit eficienta, ei inlocuind trupe in misiuni de recunoastere destul de riscante. Mai nou anumite tipuri de masini autonome au transportat armament si chiar lansat atacuri asupra tintelor inamice. Insa acest aspect lasa loc la speculatii aceste tehnologii fiind departe de a fi 100% sigure. Pe planeta Marte au fost trimisi aceste tipuri de roboti. O data ce au ajuns pe planeta au inceput un program automat de colectare si transmitere de poze sau date provenite din analizele solului sau aerului.

Departa ca acest domeniu sa se termine, el este de abia la inceput. Gradul de inteligenta al acestor masinarii tine foarte mult numarul senzorilor. Cu cat mai multi cu atat mai bine. Daca ei acopera diferite domenii ale fizicii acest lucru sporeste capacitatea robotului de a "intelege" ce se intampla in jurul lui. Bineinteles abundenta de informatie nu acopera lipsa cunostintelor de procesare a acestor informatii. Pentru o buna functionare a sistemului trebuie gasita o balanta intre aceste doua aspecte. Daca ele reusesc sa se completeze reusita sistemului va tine de acum incolo doar de buna functionare a echipamentelor electronice si mecanice care echipeaza sistemul.

La acest proiect s-a cautat simularea comportamentului unei insecte, desi dimensiunile nu sunt chiar aceleasi. Totusi acest sistem imprumuta doua comportamente ce se intalnesc la multe insecte. Masina caută si merge tot timpul catre o sursa de lumina, acest lucru facandu-se cu ocolire de obstacole. Toate aceste lucruri se petrec intr-un plan.

Capitolul II.

Componetele electronice folosite la acest proiect

2.1. Descriere.

În acest capitol se încearcă prezentarea din punct de vedere electronic a modulelor ce fac parte din proiect. Sunt arătate tipul circuitelor integrate și caracteristicile de funcționare ale acestora. Aceste circuite sunt puse apoi în module arătându-se rolul lor, cum funcționează și limitările caracteristice. Toate valorile prezentate sunt luate de la producător.

2.2. Driver Motor.

Acest modul este folosit pentru a putea comanda în putere un motor. La proiectul de față se folosesc două astfel de module, fiecare motor în parte având un astfel de driver. El poate prelua informația logică de la un microcontroller sau alt element logic și pe baza acesteia se poate comanda motorul. Dacă s-ar încerca comanda directă a motorului de la ieșirea logică acest lucru nu ar fi posibil deoarece ieșirile logice de la microcontroller sunt în general ieșiri de mică putere. Motorul este legat la pini **Out₁** și **Out₂**. Intrările logice în acest driver vor fi legate printr-o poartă SI-NU compusă dintr-un PWM și dintr-o ieșire logică a microcontrollerului, P1L.7, P1L.6, P1L.5, P1L. Portile logice folosite în acest modul sunt date de circuitul **CMOS MMC4093**. Deși este făcut după tehnologie CMOS ce presupune valori de până la 15V aceste porți vor fi comandate tot cu 5V, acest lucru oferind o stabilitate și o siguranță mai mare circuitului logic. Deoarece are și tehnologie **Trigger-Schmitt** semnalul de la ieșire este “curatat” și îndreptat având o formă rectangulară. Acest driver a fost construit folosind schema următoare. Pentru că avem două motoare este nevoie pentru comanda lor tot de două drivere.

Piesa cea mai importantă din acest modul este driverul de motor propriu-zis **BA6219B** produs de firma **ROHM**. La intrările **IN₁** și **IN₂** vom avea ieșiri logice din microcontroller. Cele două intrări în driver pot fi comandate cu valori de până la 18V asta

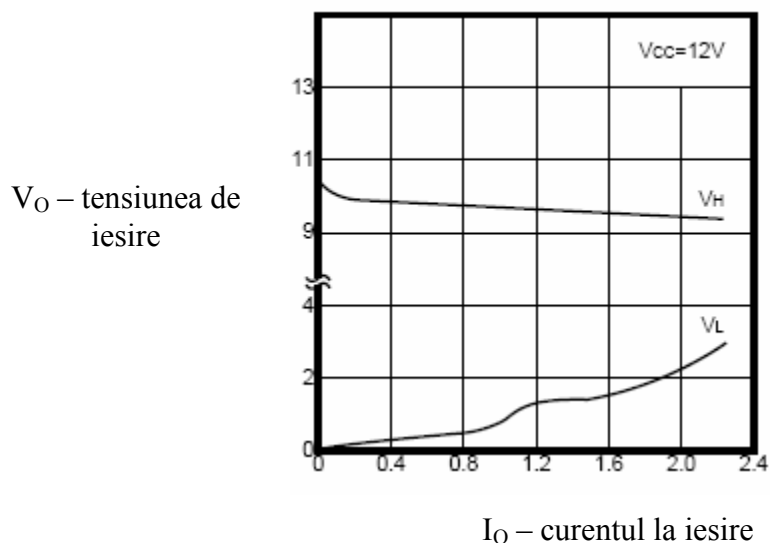
daca si tensiunea de alimentare are aceeasi valoare. Acest driver accepta tensiuni mai mici de comanda fata de tensiunea de alimentare dar invers nu este posibil, lucrul acesta putand periclita buna functionare a integratului. Dupa felul acestor intrari (L,H), motorul va avea mai multe starii. Acestea sunt date in urmtorul tabel.

Intrari		Iesiri		Mod
IN1	IN2	OUT1	OUT1	
L	L	DESCHIS	DESCHIS	Normal
H	L	H	L	In Fata
L	H	L	H	In Spate
H	H	L	L	Franat

* L – consideram low daca tensiune de intrare este sub 1V

H – consideram high daca tensiunea de intrare este peste 3 V

Acesta suporta curenti de iesire de maxim $I_O = 2.2 \text{ A}$ la o alimentare a circuitului ce trebuie sa se faca intre $V_{CC1} = 8\text{V} \sim 18\text{V}$, astfel putem obtine o putere maxima de $P = 39 \text{ W}$. Tensiunea maxima pe care o poate suporta circuitul este de 24V. In cazul de fata noi alimentam circuitul **BA6219B** la 12V(V_{CC1}), pe langa aceasta avem si V_{CC2} , tensiune ce da puterea cu care este alimentat motorul, astfel daca dorim sa crestem viteza motorului crestem aceasta tensiune sau o scadem dupa cum dorim. Capacitatea C_M este folosita pentru a putea atenua efectele parazite ce ar putea aparea atunci cand se da comanda se schimbare de sens la motor. Graficul urmtor prezinta legatura dintre tensiunea de iesire a circuitului si curentul la iesire.

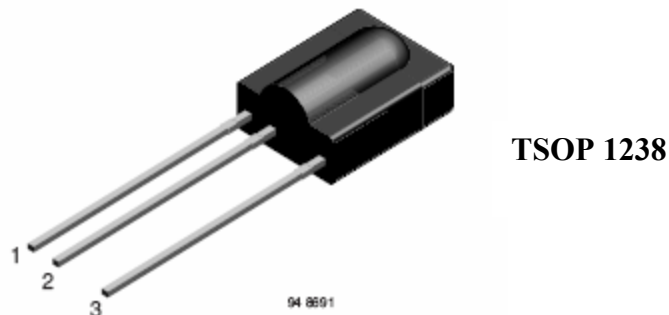


Cum am mai spus si mai devreme viteza de rotatie efectiva a motorului va fi controlata de PWM (Pulse Width Modulation). Deoarece avem doua motoare de comandat vom avea nevoie de doua astfel de PWM. Pentru a putea face roti motoarele atat in fata cat si in spate vom folosi un circuit ce contine porti si-nu acest mecanism de control fiind prezentat in sectiunea **Comanda logica a motoarelor**.

Acest modul prezentat face legatura dintre partea de logica unde folosim curenti de mica putere si partea mecanica – motoare – unde este necesar sa se foloseasca curenti de mai mare putere.

2.2. Detectorul de Intrarosi (IR).

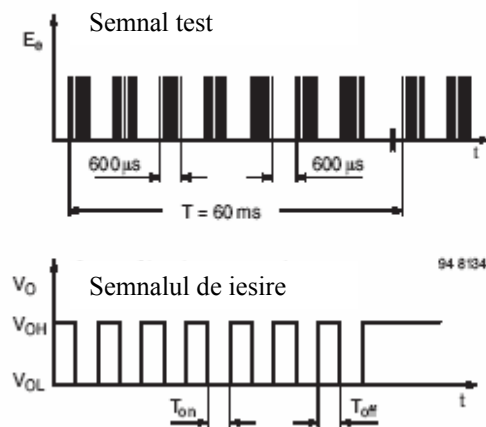
Detectorul de IR este construit in jurul integratului **TSOP 1238**. Acesta contine o dioda care poate receptiona razele de IR si alte cateva module de curatire si amplificare a semnalului. Semnalul de iesire de la acest chip poate fi folosit direct de catre microcontroller ca si semnal de intrare. Circuitul este sensibil la radiatiile IR care sunt emise cu frecventa de 38KHz de aici venind si denumirea de **1238**. In aceeaasi gama mai exista alte integrate ele variind intre **30KHz** si **56KHz**. Acest circuit are si o protectie imbunatatita la radiatiile mediului ambiant astfel senzorul sa nu poata fi influentat de semnale parazite. S-a ales frecventa de 38KHz deoarece semnalele naturale de IR nu se emit cu aceasta frecventa. Acesta este prezentat in figura A urmatoare :



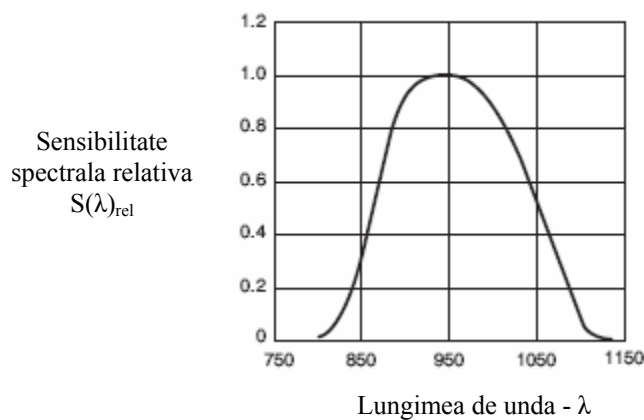
In tabelul urmator sunt prezentate cateva valori de catalog ; tensiunea de alimentare si iesire, curentul :

Parametri	Pini	Simbol	Valoare	Unitate
Alimentare Volti	(Pin 2)	V_S	- 0.3 to + 6.0	V
Alimentare Curent	(Pin 2)	I_S	5	mA
Iesire Volti	(Pin 3)	V_O	- 0.3 to + 6.0	V
Iesire Curent	(Pin 3)	I_O	5	mA
Puterea consumata	($T_{amb} \leq 85\text{ }^{\circ}\text{C}$)	P_{tot}	50	mW

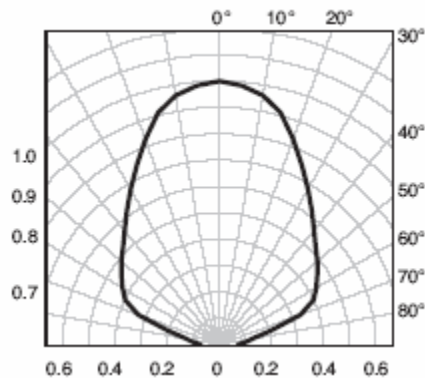
Atunci cand circuitul este alimentat si nu detecteaza IR pe iesire avem 5 V. Aceasta situatie se schimba atunci cand senzorul TSOP este stimulat cu IR, tensiunea de iesire tinzand catre 0V. Pentru a intelege mai bine ne folosim de imaginea urmatoare:



Fiind un senzor de infrarosu acesta funtioneaza cel mai bine cand este produsa radiatia IR de 960nm. El reactioneaza cel mai bine la aceasta lungime de unda. Ne folosim de un tabel petru a exemplifica acest lucru:



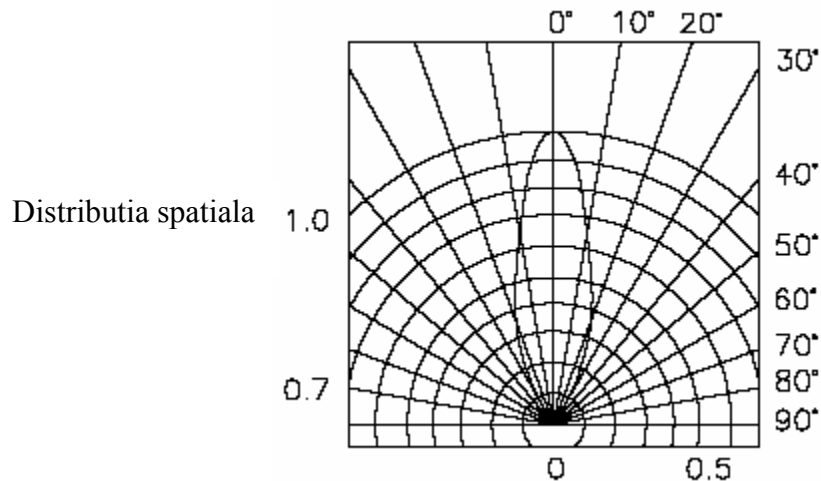
Sensibilitatea senzorului depinde atat de lungimea de unda receptionata cat si de locatia emitatorului de IR. TSOP receptioneaza IR conform graficului dat de figura.



Microcontrollerul va primi date de la senzor prin pinul 3. Am folosit in aceasta schema LED pentru a putea vedea daca senzorul receptioneaza semnal de IR. Atunci cand TSOP capteaza raze IR ledul respectiv se va aprinde semnaland acest lucru.

2.3. Emitator IR.

In acest modul se produce efectiv raza de IR de oscilatie 38KHz. Piesa care emite radiatia este o dioda (LED) **L-53F3BT**. Aceasta dioda poate fi alimentata cu pana la $I_F = 50\text{mA}$. Puterea de emisie a diodei depinde in mare masura de acest curent. Cu cat curentul pe dioda este mai mare cu atat raza de actiune a diodei este mai mare. Deoarece noua ne trebuie ca aceasta raza sa poata fi detectata maxim pana la 30cm – 40cm curentul care trece prin Led este limitat la $I_D = 22\text{mA}$. La fel ca si senzorul pentru receptia IR acest Led emite infrarosu conform graficului urmator ; ledul folosit are o arie destul de restransa de emitere a razei IR.



Am folosit doua circuite care emit raze IR. Cele 2 emitatoare vor fi puse in partea din fata a masinii in partea stanga respectiv dreapta. El sunt comandate de iesirile logice din microcontroller P1L3,P1L2. Aceasta comanda este necesara deoarece folosim doar un singur receptor de IR. Astfel din microcontroller comandam pe rand cele doua emitatoare sa functioneze. In decursul a **10ms**, primele **5ms** sunt folosite pentru comanda **Emitator Dreapta**, urmatoarele **5ms** pentru comanda **Emitator Stanga**. Prin aceasta metoda nu mai folosim doua receptoare de IR. Daca receptia de IR se face in primele 5ms atunci vom stii ca in partea stanga exista un obiect ce reflecta razele infrarosii produse de Emitator Stanga. Acest obiect poate fi depista daca se afla la o distanta mai mica de **15 cm**. Pe baza acestui rationament masina va incerca sa ocoleasca obiectul dand comanda la motoare sa mearga in spate pe o distanta de 10cm si apoi sa faca o rotatie pe loc a masinii in partea dreapta cu aproximativ 45°. Acelasi lucru se va intampla cu masina daca in restul milisecundelor de la 5ms la 10ms detectorul de IR va fi excitat de raze IR. Masina va ocoli de aceasta data prin partea dreapta.

Dupa cum se vede pe schema LED-ul de IR nu emite continuu ci cu o frecventa de 38KHz. Aceasta este data de un oscilator creat cu o poarta NAND. Aceasta frecventa este validata de pini enumerati mai sus (**nici o data cele doua Leduri nu vor emite concomitent**). Deoarece avem curenti mici de comanda nu vom putea folosi direct pentru a emite IR. De aceea vom folosi un tranzistor pe post de comutator. Acesta este **KSP2222**, el este comandat in baza de curentul dat de poarta Nand cu o frecventa data.

Datorita acestui fapt jonctiunea dintre Colector si Emitor se deschide permitand unui curent mai mare sa alimenteze Ledul de IR.

Distanța la care pot fi depistate obiectele depinde atât de curentul care îi trecut prin LED-ul IR cât și de tipul acestuia. Aici rezistența R_1 este calculată pentru a oferi o distanță optimă ce poate permite ocolirea în bune condiții a obiectului.

2.4. Oscilator la 38KHz.

Acest oscilator are ca piesă principală o poartă Nand dată de **MMC4093**. După cum se vede în schema este vorba de o buclă cu reacție. Această buclă se realizează prin rezistența de **1.350 Ω** . Oscilațiile de care aveam nevoie le puteam realiza și print-un cuarț, dar acolo reglarea oscilației nu ar fi fost atât de simplă. Aici doar prin mărirea sau micșorarea rezistenței reglabile de 250 K Ω putem modifica oscilația între valori de aproximativ 27 KHz – 40 KHz.

Frecvența pe care o avem la ieșire se poate calcula după următoarea formulă:

$$f_o = \frac{1}{R \cdot C \cdot \ln 2} ,$$

Astfel dacă dorim o frecvență mai mare putem scădea rezistența sau capacitatea. Pentru o mai bună reglare a frecvenței am folosit pe lângă o rezistență propriuzisă de 2,5 M Ω și o rezistență reglabilă de 250 k Ω .

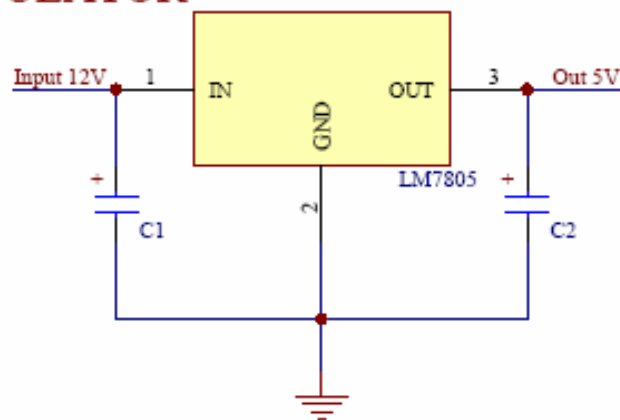
2.5. Regulator de tensiune la 5 V.

Pentru a avea curent de 5V stabilizat folosim un circuit specializat în acest fel **LM7805**. Acest regulator poate transforma tensiuni de până la 24V, având curenți de ieșire de maxim 1A.

Parametri	Simbol	Conditii	LM7805			Unitati
			Min.	Ob.	Max.	
Tensiune de iesire	V_O	$5.0\text{mA} < I_O < 1\text{A}$, $P_O < 15\text{W}$ $V_I = 7\text{V}$ la 20V $V_I = 8\text{V}$ la 20V	4.75	5.0	5.25	V

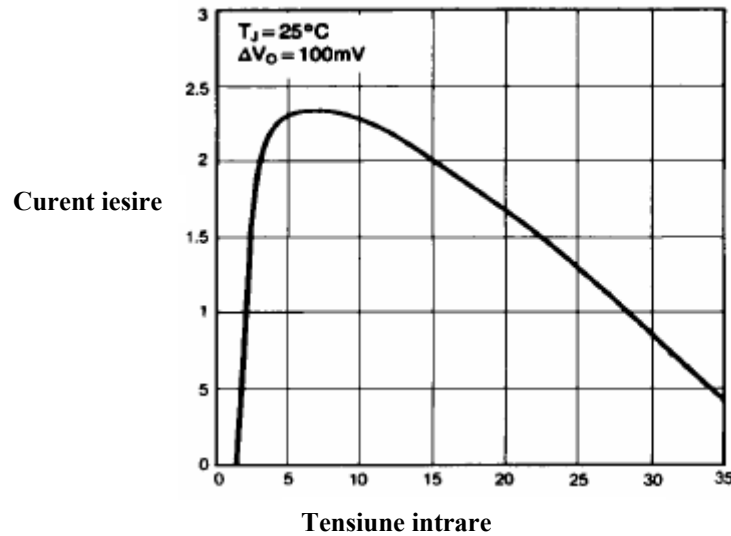
Regulatorul este construit si cu ajutorul a doua condensatoare polarizare la fel ca in schema :

REGULATOR



$$C_1 = 0.47 \mu\text{F} ; C_2 = 0.47 \mu\text{F}.$$

La acest regulator noi facem alimentarea cu 12V de la o baterie, Tensiunea rezultata de 5V este folosit pentru a alimenta circuitele logice din schema. In realitate ea este de 4,94V. Cele doua motoare si driverul deoarece sunt de putere mai mare ele vor fi alimentate la 12V. El este folosit atat la alimentarea circuitelor logice cat si la intrarile unor circuite logice ce trebuie sa aiba anumiti pini doar pe valoare high. Dependenta dintre tensiunea de intrare si curentul de la iesire este data in graficul urmator.



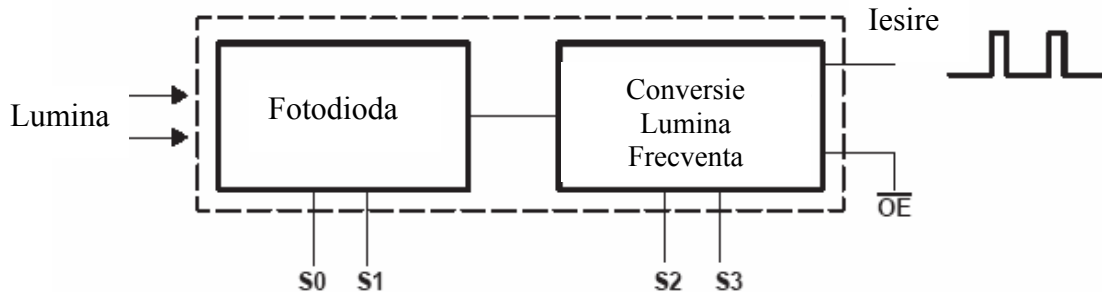
Se observa din acest grafic ca, valoarea cea mai mare a curentului pe care il poate produce acest regulator este de 2,2 A la 5V. Regulatorul este prevazut cu un radiator pentru a putea disipa caldura ce se degaja atunci cand acest regulator este pus in sarcina.

2.6. Detectorul de lumina.

Masinuta ce este construita trebuie sa se orienteze dupa lumina si sa mearga catre acel loc unde intensitatea ei trece un anumit prag. De aceea avem nevoie de detectoare de lumina. Acestea sunt doua la numar, la fel ca si ledurile IR dispuse pe botul masinii. Cei doi senzori detecteaza o valoare a intensitatii si printr-o comparare intre cele doua valori luam decizia in ce parte masina sa se deplaseze. Schema de principiu este una simpla acest integrat fiind unul destul de performant.

Acesta este chipul de transforma intensitatea luminoasa in frecventa. El poate fi programat adica putem sa ii setam sensibilitatea, de asemenea frecventa ce se genereaza poate si divizata pentru a avea o valoare mai mica. Acest chip este construit din aproximativ 100 de diode, aceste diode generand un curent ce este apoi transformat in frecventa de un convertor CMOS. Frecventa ce este generata are un factor de umplere de 50%, semnalul fiind unul rectangular de 5V. Frecventa este proportionala cu lumina. Toate intrarile si iesirile sunt TTL permitand astfel comunicarea direct cu

microcontrolerul. Avem pinul \overline{EO} de validare a circuitului, acesta putand fi folosit si de microcontroler. Circuitul pe care il folosim are o toleranta de masurare de 5%, el fiind sensibil la valori ale lungimii de 300nm – 700nm. In spectrul de lungimi de unda la care el este sensibil intra putin si radiatiile infrarosii, dar in proiect aceasta caracteristica nu este folosita. Spre deosebire de senzorul de Ir folosit acesta este putin sensibil la radiatiile **naturale** si nu cele **artificiale** produse de noi cu ajutorul emitatoarelor.



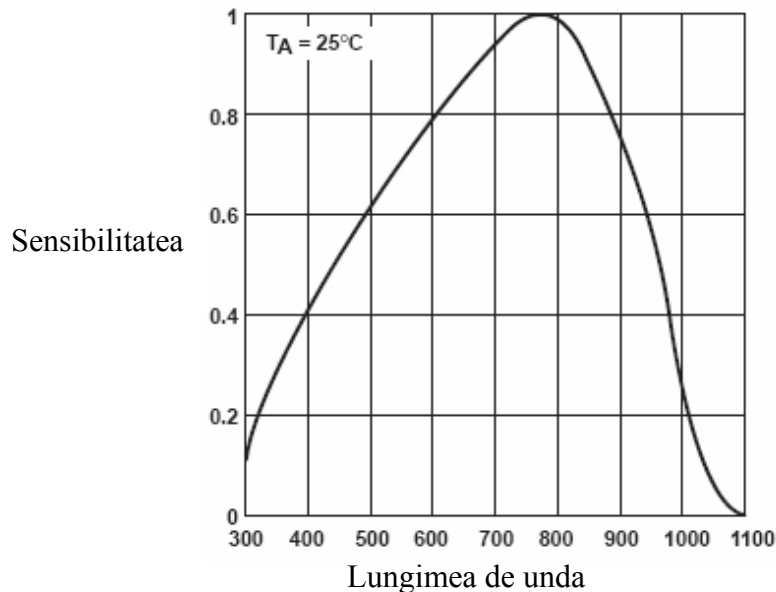
S ₁	S ₀	Sensibilitate
L	L	Nealimentat
L	H	1X
H	L	10X
H	H	100X

Se observa ca din acest grafic gradele de sensibilitate pe care le are TSL230. Daca sunt pe high ambele intrari atunci vor fi alimentate toate diodele crescand astfel frecventa de la iesire. In cazul S₁ = H, S₂ = L doar o poarte din aceste diode sunt active.

S ₃	S ₄	f ₀ (divizata cu)
L	L	1
L	H	2
H	L	10
H	H	100

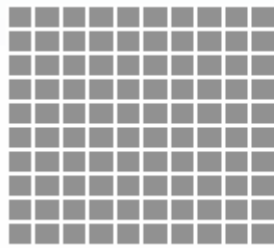
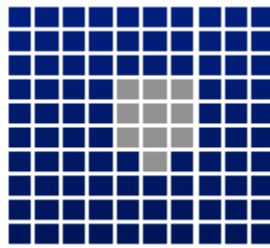
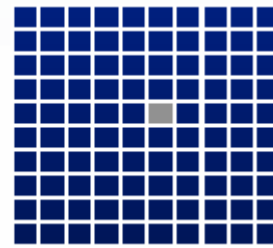
Acesti pini sunt folositi atunci cand vrem sa ca frecventa masurata sa fie cu o toleranta mai mica la eroare. Asa noi putem pune sensibilitatea pe maxim si apoi sa divizam cu 100, divizarea aceasta fiind folosita in general la microcontrolere care nu au asa multe resurse si nu pot reactiona in timp real la frecvente prea mari.

In continuare prezentam un grafic care ne arata la ce lungime de unda diodele sunt cele mai receptive. Aceasta valoare este undeva in jurul a 780nm.



Acest integrat se alimeneaza cu 5V generand un puls cu amplitudinea tot de 5 V. In intuneric « absolut » circuitul la 25° scoate 1 Hz. Daca intensitatea depaseste 1050 lx circuitul se satureaza frecventa maxima pe care o scoate fiind de 1.09 MHz, aceste valori fiind obtinute cu $S_0 = H$, $S_1 = H$, $S_2 = L$, $S_3 = L$.

Prezentam in continuare cum sunt dispuse aceste diode (deschise) in functie de starile S_n . Patrutul mare reprezinta suprafata totala a senzorului. Patratele mici sunt diodele separate. Se observa ca avem 100 de diode. Diodele cu gri sunt deschise, deci sensibile la lumina, iar cele albastre sunt inchise.

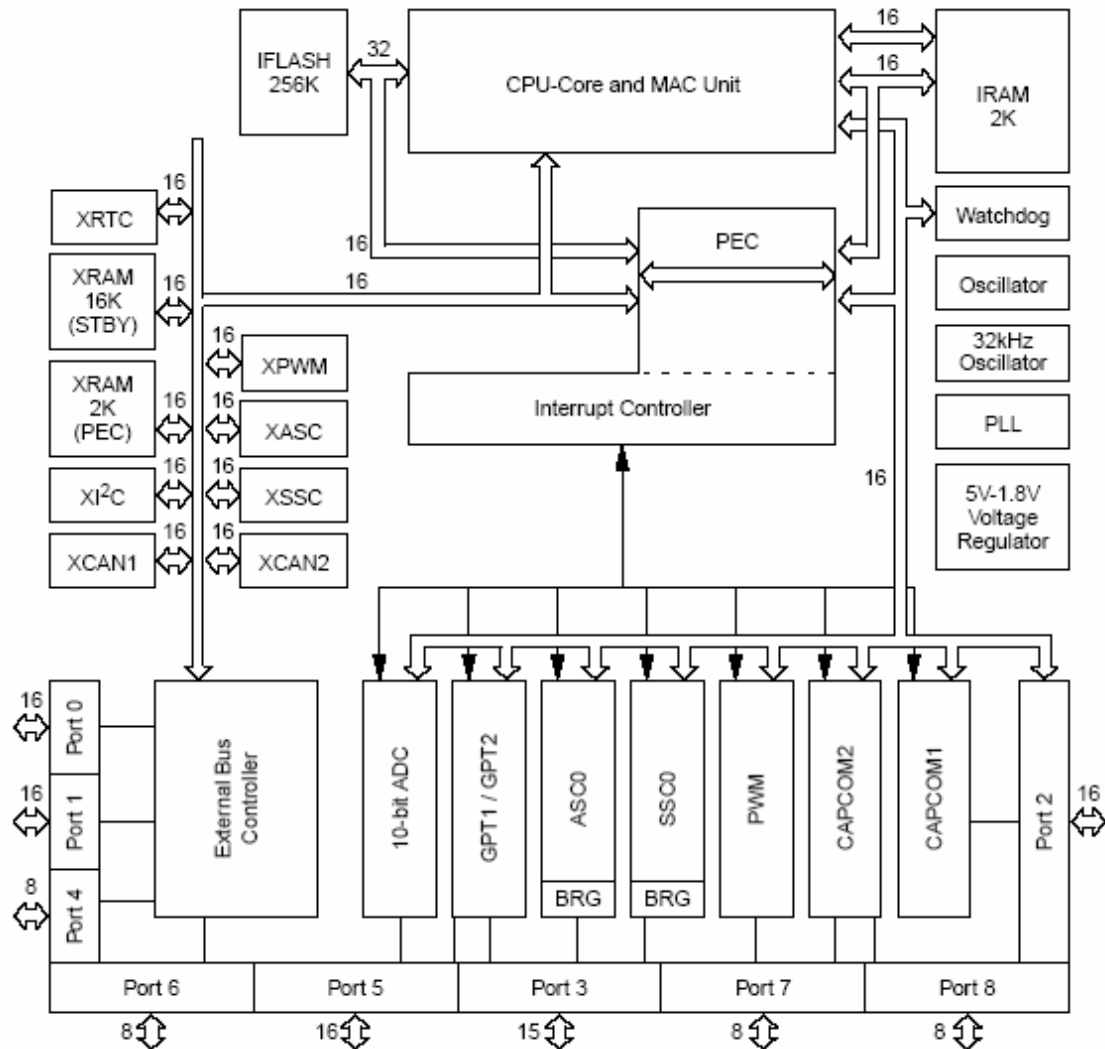
**100X** $S_0 = 1, S_1 = 1$ **10X** $S_0 = 0, S_1 = 1$ **1X** $S_0 = 1, S_1 = 1$

2.7. Unitatea de control

2.7.1. Descriere.

Pentru a putea face ca totul sa functioneze unitar avem nevoie de un sistem de comanda. Acest lucru este realizat cu microcontrolerul pe 16 biti **ST10F272**. In continuare prezentam cateva date tehnice ale acestui chip.

Arhitectura lui ST10F272 combina avantajele procesoarelor bazate pe RISC (Reduced Instruction Set Computing) CISC (Complex Instruction Set Computing) cu un subsistem periferic avansat. Nucleul pricipal cuprinde 4 stagii de instructiuni pe pipeline si o unitate aritmetica si logica (**ALU**) pe 16 biti. Anumite caracteristici merita sa fie evidentiata: banda mai mare pentru instructiuni, executie mai rapida, control marit al perifericelor, instructiuni mai bune si mai optimizate si nu in ultimul rand programarea pe prioritati a intreruperilor. Acest microcontroler functioneaza la o frecventa de 16MHz. Diagrama blocului functional.



Cele mai multe instructiuni sunt executate intr-un singur ciclu de instructiune. Altele mai complexe au fost optimizate, astfel o inmultire pe biti 16×16 se face in 5 cicluri de instructiuni, iar o impartire pe biti $32 / 16$ in 10 cicluri de instructiuni. Spatiul de memorie este unificat ceea ce inseamna ca memoria de cod, de date, a registrilor si a porturilor de I/O poate fi adresata liniar, valoarea ei fiind de pana la 16Mbytes. Intreaga memorie poate fi accesata la nivel de byte sau de word. O portiune din chip a fost facuta special pentru a putea fi accesibila bit cu bit. In plus fata de aceasta memorie aplicatia poate accesa inca 16Mbytes prin magistrala externa.

In familia ST10 de microcontrolere **perifericele** sunt separate de nucleu. Acest lucru ofera un mare avantaj microcontrolerului, deoarece permite anumitor periferice sa

fie introduse sau scoase fara a modifica nucleul. Fiecare bloc proceseaza si comunica independent pe magistrala comuna. Perifericele sunt controlate prin datele ce se scriu intr-un registru special SFRs (Special Function Registers). Aceste periferice sunt folosite de CPU, pentru a comunica cu exteriorul sau pentru a implementa diferite functii.

Aceste periferice sunt:

- **9 porturi I/O cu un total de 111 I/O linii.**
- 2 interfete seriale (ASC0 si SSC)
- **2 Timere (GPT1 si GPT2)**
- un watchdog timer
- **2 unitati Capture/Compare cu 16-canale (CAPCOM1 si CAPCOM2)**
- **4 canale PWM (Pulse Width Modulation)**
- 10 biti pentru Convertorul Analog/Numeric

Perifericele ce sunt bolduite au fost folosite la proiectul acesta sau cel putin o parte dintre porturi. Pentru a marii performantele chipului au mai fost introduse inca cateva caracteristici.

- **2 interfete CAN**
- 2 interfete seriale in plus (XASC si XSSC)
- o interfata seriala I²C
- inca 4 canale de PWM

Cum am mai spus comunicarea se face cu ajutorul **SFRs** cand este nevoie a controla starea, datele unui periferic si **intreruperile** emise de periferic atunci cand apare un eveniment in timpul functionarii. Ex. sfarsitul procesului, un nou eveniment, o eroare.

2.7.2. Modulul CAPCOM.

Modulul ce este folosit intens in acest proiect este **CAPCOM** (**Capture/Compare**). Sunt doua astfel de module dar noi il folosim doar unu CAPCOM2. Aceasta unitate este foosita pentru intrari si iesiri de mare viteza, cum ar fi generare de puls si semnal, generare de PWM, conversie Analog/Digitala sau inregistrarea timpului pentru evenimente exterioare.

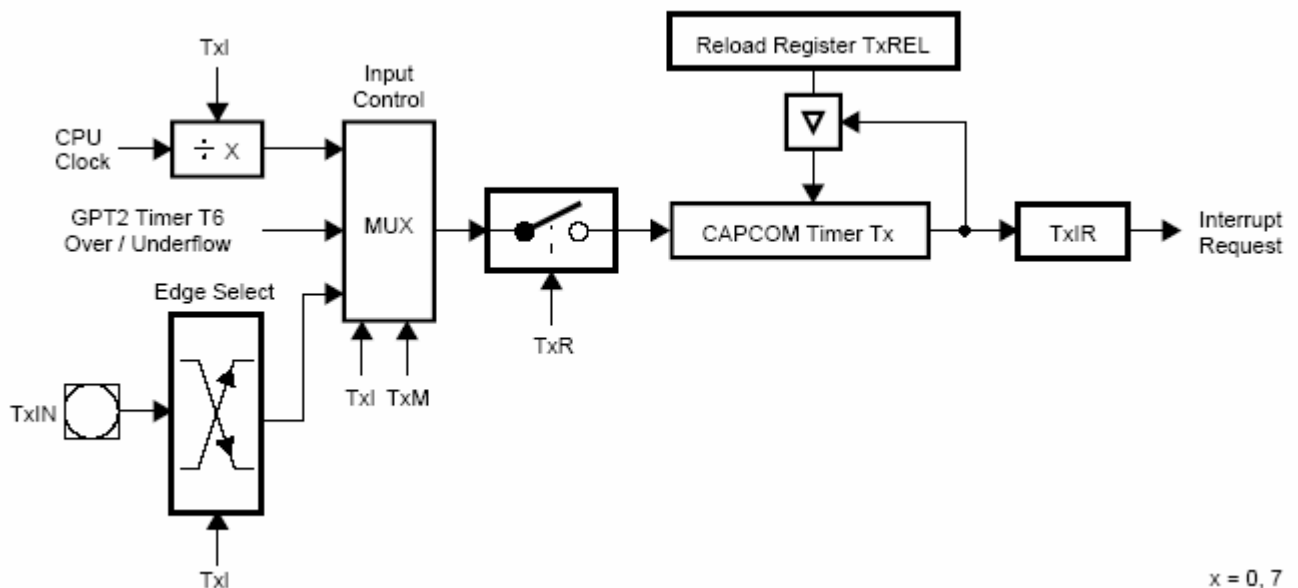
In aceasta unitate avem 4 timere (**T0/T1**, **T7/T8**) de cate 16 biti cu registre de incarcare. Clock-ul ce tine de aceste timere este programabil cu ajutorul a catorva pre-scalere ce schimba frecventa de numarare a clock-ului intern. Aceste timere pot numara atat crescator cat si descrescator. Toate aceste caracteristici ne dau o flexibilitate sporita a perioadei si rezolutiei timer-ului putand face ajustari destul de fine pentru aplicatii care cer aceste lucruri.

Timer-ul pe care noi in folosim in aplicatie este **T7**, el putand capta/compara evenimente externe. Cand registrul capture/compare a fost selectat pentru capturare, continutul curent al timer-ului alocat va fi incarcat in registrul capture/compare. In plus pentru aceasta procedura este emisa si o intrerupere care semnaleaza evenimentul ce a avut loc.

In modul **compare** continutul tuturor registrelor ce au fost selectate sunt comparate cu timerele alocate. Atunci cand avem o egalitate intre timer si registrul de comparat, se va desfasura o anumita procedura insotita si ea de o intrerupere.

Aceste unitati vin cu **32 de canale** si 4 timere. Pentru a intelege mai bine modul de functionare prezentam diagrama ce este in catalogul acestui microcontroler.

Aceasta schema este pentru timerul 0 si 7.



Modul de Compare sau Capture se selecteaza dintr-un registru **T01CON**. In bitul desemnat cu T0M se poate face aceasta selectare. Activarea timerului este data de bitul T0R. Tot aici avem si prescalerul cel care divide frecventa clockului astfel incat sa fie optim pentru aplicatia noastra.

Frecventa de iesire se calculeaza dupa formula:

$$f_{Tx} = \frac{f_{CPU}}{2^{[(TxI)+3]}}$$

T01CON (FF50h / A8h)					SFR								Reset Value: 0000h			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
-	T1R	-	-	T1M	T1I		-	T0R	-	-	T0M	T0I				
RW				RW		RW		RW				RW		RW		

Pentru fiecare registru in parte al unitatii CAPCOM noi putem seta modul in care se lucreaza si timerul cu care comparam. In acest proiect s-a folosit **modul 0** mod in care doar se emit intreruperi. Noi lucram CC24 si CC25. Aici incarcam valoarea ce trebuie comparata. Lucrul cu aceste registre pot fi configurate cu ajutorul registrelor de mod in cazul nostru acest registru.

CCM6 (FF26h / 93h)								SFR				Reset Value: 0000h			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ACC27		CCMOD27		ACC26		CCMOD26		ACC25		CCMOD25		ACC24		CCMOD24	
RW		RW		RW		RW		RW		RW		RW		RW	

Aici in **CCMOD24** punem modul dorit. Modul 0 este dat de valoarea 100, **ACCX** fiind pentru setarea timerului: 0 pentru T0/T7 si 1 pentru T1/T8. Valoarea cu care se compara timerul este pus in **CC24** respectiv **CC25**. Cu ajutorul acestor functii noi putem compara diferite valori si emite intreruperi pe care apoi sa le tratam in mod convenabil. Astfel putem emula si un PWM, emitand o intrerupere la egalitate si inca una la overflow la timer. Fiecare intrerupere scoate pe acelasi pin ori 1 ori 0. Din aceasta alternanta se simuleaza un PWM. Factorul de incarcare depinde de valoarea care o dam in CC24 respectiv in CC25.

2.7.3. Pulse Width Modulation.

Acest modul poate genera pana la patru semnale de PWM. Fiecare canal este controlat de un numarator ce are asociat un registru de incarcare si un registru de comparare. Noi prin acest modul putem regla atat factorul de incarcare cat si frecventa de lucru a acestui modul. PWM este foarte folosit in aplicatiile industriale el fiind si la baza comenzii acestor motoare

2.7.4. CAN.

Este un modul autonom ce se poate ocupa singur de transmisia si receptia datelor conform variantei V2.0 de specificatii. Chipul modulului de CAN poate trimite si primi atat in mod standard date cu 11 biti de identificare cat si in mod extins date cu 29 biti de identificare. Modulul CAN este pe deplin functional cu pana la 32 de obiecte mesaj, fiecare obiect avand un identificator. Aceste obiecte pot si concatenate in modul **FIFO**. Fiecare mesaj obiect poate fi updatat independent de la alt obiect si pot avea pana la maximum 8 bytes. Doi pini folosim pentru a interfata cu magistrala de transmisie. Acesti pini trebuie sa fie legati intre ei printr-o rezistenta de 150Ω deoarece fiecare fir are o tensiune diferita, rezultand de aici o cadere de tensiune intre cele doua fire.

Noi folosim acest modul pentru a comunica intre calculator si microcontroler. Programul care este scris in C este link-editat si apoi compilat. Dupa compilare obtinem un fisier hexa. Acest fisier il incarcam intr-un program care folosindu-se de transmisia prin CAN flashueste microcontrolerul cu programul dorit.

Capitolul III.

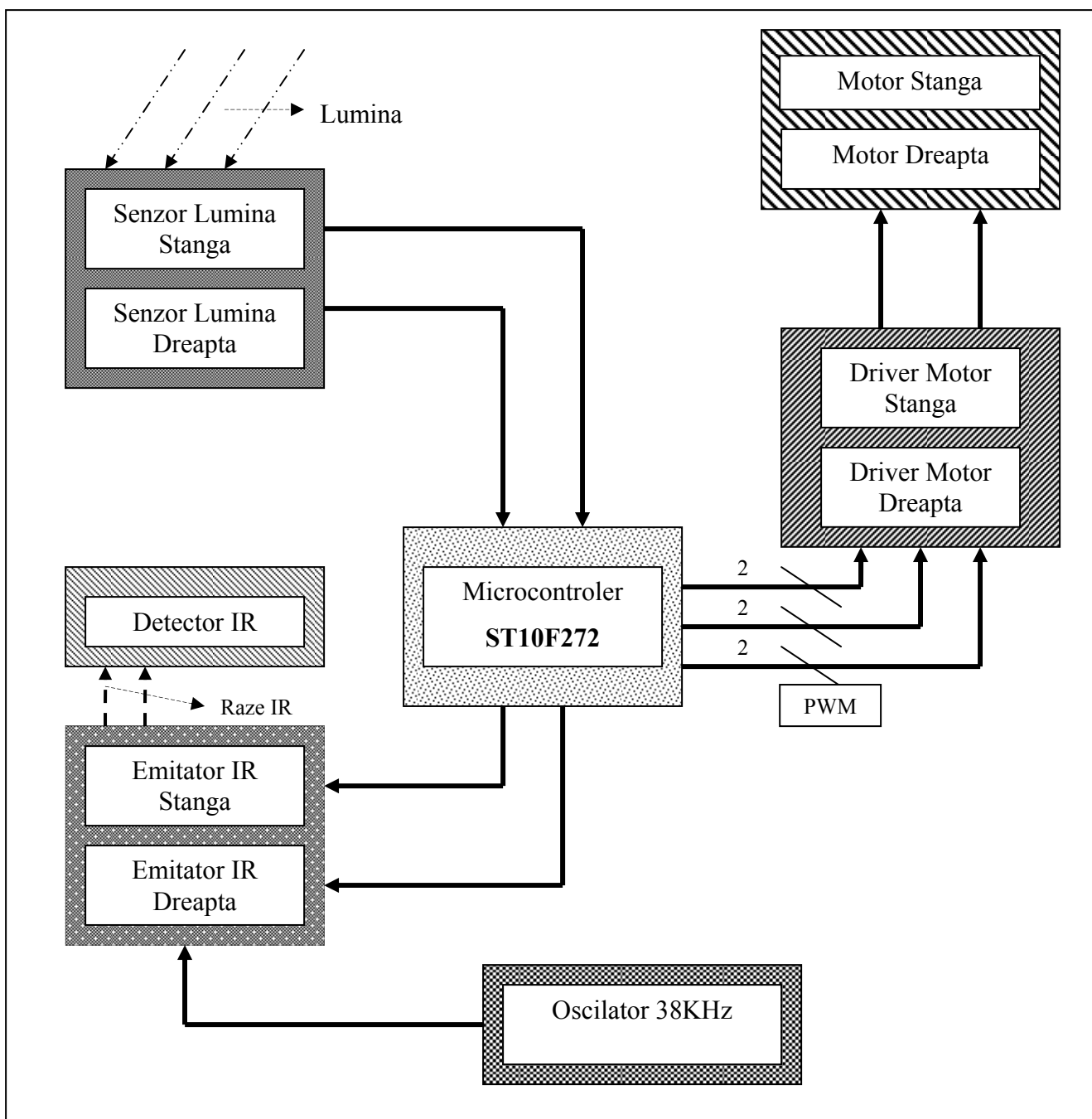
Schema bloc a sistemului. Principii de functionare.

3.1 Descriere.

Pentru o mai buna intelegere a functionarii pieselor electronice de pe masina prezentam in continuare schema bloc a masinii. Aici sunt prezentate modulele si dependintele intre ele.

Schema presupune mai multi senzori de proximitate, de lumina pe baza carora trebuie sa se ia niste decizii pentru buna functionare a mobilului. In explicarea functionarii in totalitate a schemei s-a pornit de la descrierea senzorilor de proximitate (apropiere).

Explicam pe scurt schema bloc realizata. La pornirea masinutei printr-un intrerupator, se initializeaza microcontrolerul. Dupa aceasta in microcontroler porneste procedura de cautare a luminii. Atata timp cat nu exista obiecte in jurul masinii microcontrolerul primeste informatii pe cele doua fire de la senzorii de lumina. Acestia informeaza microcontrolerul despre intensitatea luminii. Pe baza acestora el da comanda la drivere prin cei 4 pini alocati urmand apoi sa dea si puterea cu care sa mearga rotile, aceasta facandu-se cu cele doua PWM-uri. Daca in drumul spre sursa de lumina apare un obstacol acest lucru este semnalizat de detectorul de IR. El va informa microcontrolerul de existenta unui obiect. Unitatea de calcul analizeaza in ce parte este obiectul si initializeaza procedura de ocolire. In timpul acestei proceduri microcontrolerul nu va mai accepta nici un fel de informatie. Citirea senzotilor se va face imediat dupa incheierea acestei proceduri. Deoarece avem doar un singur senzor de Ir, pentru a putea determina in ce parte este obiectul cele doua emitatoare de Ir sunt controlate de unitatea de calcul cu ajutorul a 2 iesiri logice. Driverele pe langa schimbarea directiei motoarelor asigura si comanda acestora in putere, microcontrolerul neputand face acest lucru. Prezentam schema bloc a carui comentariu l-am realizat in paragraful anterior.



Pentru o mai buna intelegere a mecanismului de functionare a sistemului prezentam in continuare fiecare modul si rolul ce il are in mecanism.

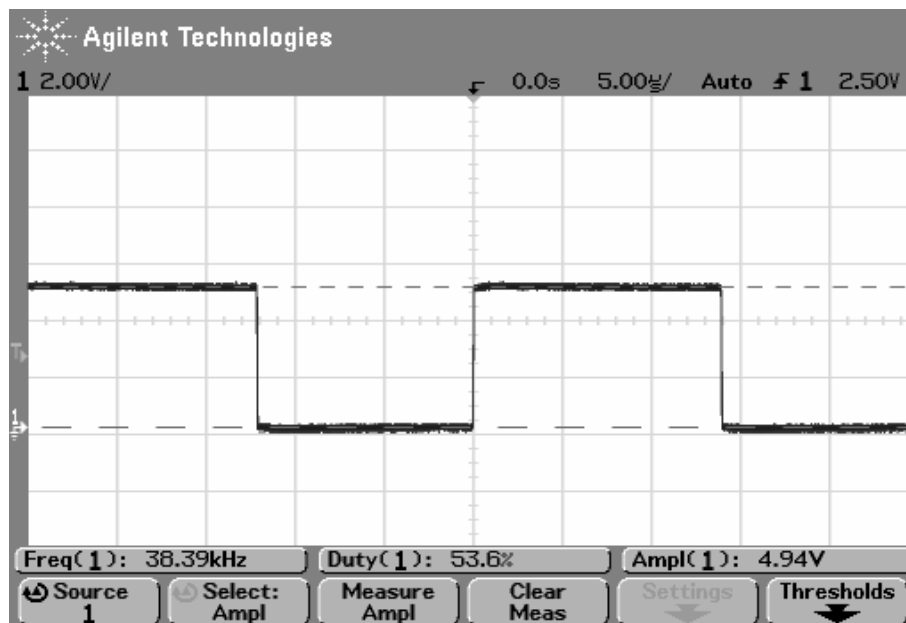
3.2. Circuit detectie obiecte.

3.2.1. Ir LED.

Deoarece toate deciziile ce trebuie luate de microcontroler depind de senzori vom incepe cu ei, si anume **senzorii de proximitate**. Acestia sunt de fapt doua diode care emit in spectrul razelor infrarosi. Aceste leduri trebuie sa se deschida si inchida cu o frecventa de 38KHz. Ledurile ar putea emite fara aceasta frecventa dar in acest caz detectorul de Ir **TSOP 1238** nu ar mai recunoaste aceasta radiatie. S-a ales aceasta frecventa pentru a putea distinge radiatia **artificiala** – creata de noi prin diferite emitatoare – si radiatia **naturala** – creata de diferite obiecte din natura (soare, surse de caldura, oameni) -.

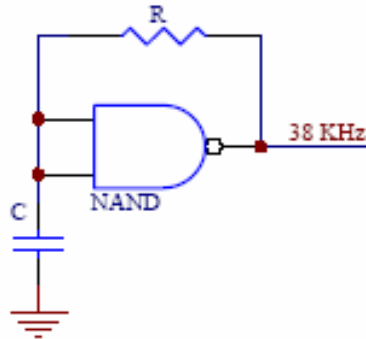
3.2.2. Oscilator de 38KHz.

Aceasta frecventa am creat-o cu un oscilator facut dintr-o poarta SI-NU ce are implementata tehnologie **Schmitt trigger**. Datorita rectiei negative ce am creat-o cu o rezistenta de ordinul Mega-ohmilor, semnalul de la iesire se intoarce prin rezistor la intrarile portii ce sunt legate impreuna. Circuitul logic are 8 pini si 4 porti fiecare cu 2 intrari. Deoarece beneficiaza de tehnologie de intraptare a fronturilor semnalul de la iesire arata in felul urmator:



Frecventa pe care o generam trebuie sa fie in jurul valorii $38\text{KHz} \pm 2\%$. Rezistenta din schema este creata de fapt din doua: una fixa de $1,25\text{ M}\Omega$ si una reglabila de $0\text{-}250\text{K}\Omega$, astfel putem regla mai bine frecventa modificand aceasta rezistenta.

Am realizat acest oscilator dupa schema urmatoare:

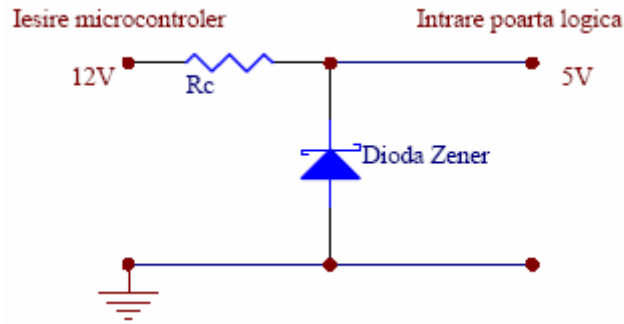


$$C = 20\text{pF} ; R = 1.350\text{ }\Omega\text{M}$$

3.2.3. Comanda emitor IR.

Deoarece am luat decizia de a folosi doar un singur detector de IR, pentru a putea determina in ce parte – stanga sau dreapta – se afla un obiect cei doi senzori ce sunt dispusi in fata masinii vor trebui sa emita pe rand. Dupa fiecare emisie de IR se citeste senzorul **TSOP 1238**. Oscilatia merge direct in cele doua porti, fiecare poarta corespunde unui emitor. Aici apare **comanda senzorilor**, astfel am programat microcontrolerul sa emita un front pozitiv timp de 500ms pentru poarta stanga si 0 pentru poarta din dreapta, restul de 500ms situatie inversandu-se poarta stanga 0, poarta dreapta front pozitiv. Rezultatul este alternanta emiterii intre cei doi senzori, perioada fiind de 1 secunda. Astfel vom stii in fata carui emitor de IR se afla un obiect, luand masurile ce se inipun in acea situatie.

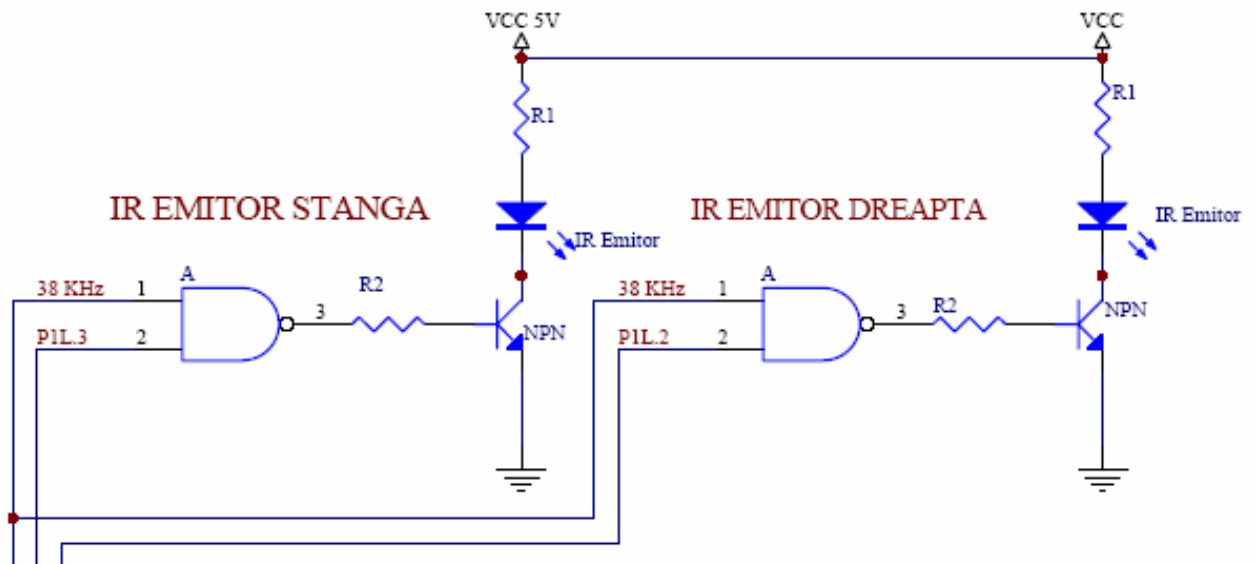
Microcontrolerul ce il folosim are la pinii de la intrare tensiunea de 12V. Din aceasta cauza trebuie sa adaptam toate iesirile logice de la microcontrolerul **ST10F272** astfel incat sa putem comanda circuite logice ce pot avea maxim 5V. Pentru aceasta la fiecare iesire logica a microcontrolerului avem circuitul de fata.



$$R_c = 5 \text{ K}\Omega; D_z = 5 \text{ V (Zener)};$$

Se observa aici ca noi luam caderea de tensiune de pe dioda care in cazul de fata este 5V. Tensiunea de iesire nu depinde de tensiunea de intrare – atata timp cat este mai mare de 5V – pastrandu-se constant la 5V. Daca am fi folosit o rezistenta in loc de dioda, tensiunea de la iesire ar fi depins de tensiunea de intrare.

Aceste doua emitatoare sunt prezentate in figura urmatoare:



$$R_1 = 50 \text{ }\Omega; R_2 = 10 \text{ K}\Omega; \text{ Poarta NAND} = \text{MMC4093}; T_r = \text{KSP2222};$$

$$\text{LED} = \text{L-53F3BT}$$

In proiectul care am realizat comanda masinii au fost definite niste functii pentru cu ajutorul carora putem scrie pe anumiti pini valori logice. In cazul de fata noi trebuie sa

comandam logic cei doi emittori. In acest caz folosim functiile **DOP** (Digital Outputs Pins):

```
/* comanda emittor Ir dreapta */
```

```
J_BIOS_DOP_LEVEL0_LC_SHIFT_SOL() /* fct. pt. inchis emisie emit_dr*/
```

```
J_BIOS_DOP_LEVEL1_LC_SHIFT_SOL() /* fct. pt. deschis emisie emit_dr*/
```

```
/* comanda emittor Ir stanga */
```

```
J_BIOS_DOP_LEVEL0_L_RB_SHIFT_SOL() /* fct. pt. inchis emisie emit_st */
```

```
J_BIOS_DOP_LEVEL1_L_RB_SHIFT_SOL() /* fct. pt. deschis emisie emit_st */
```

Efectul folosirii lor este alternanta de emisie a radiatii.

3.2.4. Comanda putere senzor IR.

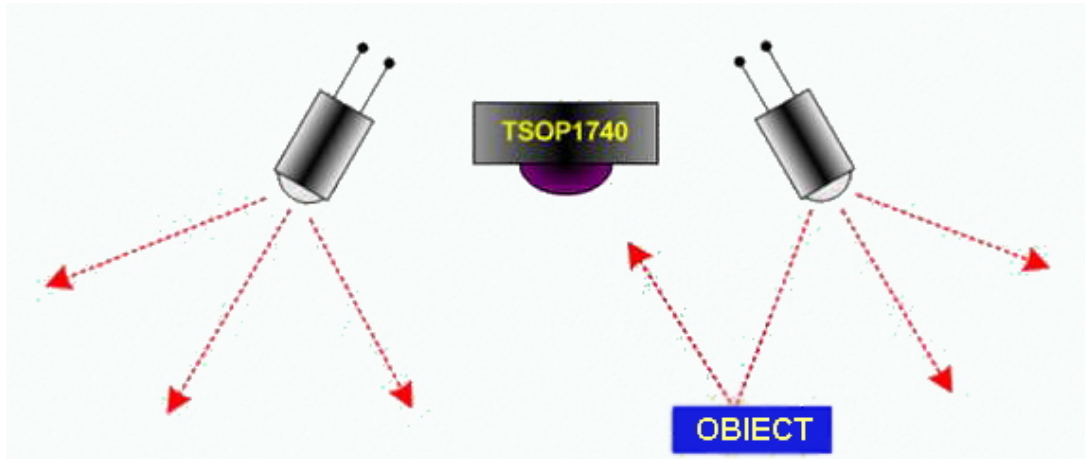
Se observa ca am folosit porti logice, de aceea acest circuit nu este unul de putere, neavand destui amperi. Pentru a putea emite cu cei doi senzori ne trebuie un curent prin ei de 20mA-30mA. Se impune folosirea unui tranzistor care este comandat in baza de catre acest curent. Prin aceasta metoda curentul de la portile logice inchid si deschid tranzistorul cu frecventa de 38KHz. Dupa un calcul putem determina rezistenta care se pune in serie cu Ledul pentru a avea un curent mai mare si implicit o putere de emisie sporita. Pe noi ne intereseaza ca obiectul care se apropie sa fie detectat pe la o distanta de 10cm-15cm – nici mai mare si nici mai mica -.

3.2.5. Senzor de IR.

Emisia de IR trebuie cumva captata. Acest lucru se realizeaza cu circuitul TSOP 1238. Ca si orice integrat este alimentat la 5V. Modul de functionare este destul de simplu. Atunci cand este in prezenta unei radiatii de 38KHz tensiunea de la pinul de iesire tinde catre 0V. Daca nu avem emisie de radiatie la iesire tensiunea este de 5V iar curentul de aproximativ 5mA. Datorita acestui curent destul de mic putem spune ca integratul poate comanda direct un microcontroler. In schema prezentata iesirea este legata direct la un bit de intrare a microcontrolerului. Daca avem 0V pe acest pin

inseamna ca unul dintre cele doua emitatoare are un obiect in fata. Urmeaza sa deducem care dintre ele dupa timpul in care a fost emisa radiatia.

Pentru a intelege mai bine dispunerea in spatiu a acestor senzori prezentam urmatoarea figura:

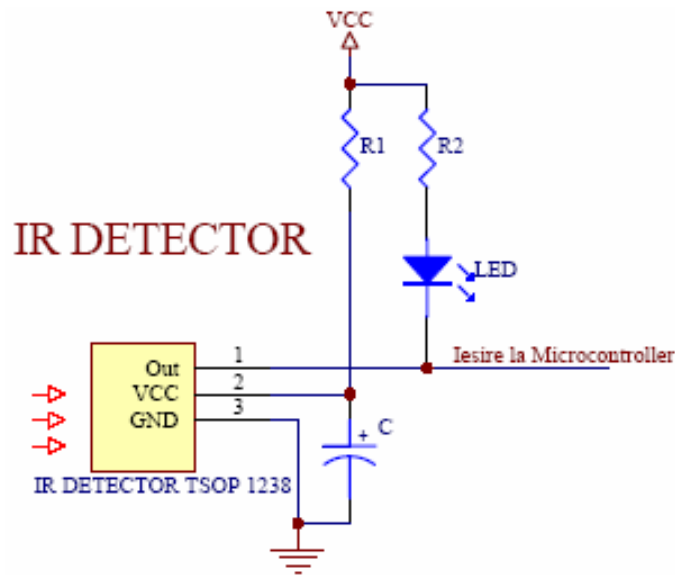


La acest modul am mai pus si un Led rosu in serie cu o rezistenta de $1K\Omega$. Acesta are rolul de a se aprinde atunci cand senzorul detecteaza ceva radiatie. Efectul acestor piese este doar numai unul vizual si de verificare.

Microcontrolerul pe care il folosim sesizeaza fronturi pozitive doar de la 3V in sus. Ideal ar fi pentru el ca acesta sa aiba o valoare de 12V. Desi senzorul de Ir scoate un curent perceptibil pentru un circuit logic **ST10F272** nu sesizeaza fronturile pozitive (aici tensiunea de iesire este de 4,74V pentru high si 0,14V pentru 0). De aceea semnalul de la TSOP 1738 este trecut printr-o poarta logica SI-NU. Astfel daca exista emisii de radiatii avem un bip de 1 logic, daca nu 0 logic. La fel ca si ledurile de infrarosii el este postat in fata masinii putin in spatele emitatoarelor.

Dupa cum se stie aceste radiatii sunt cele mai bine respinse de care obiectele albe. Cu cat culoarea lor se apropie de negru cu atat ele absorb mai multa radiatie. Acest lucru se reflecta in distanta la care pot fi reperate obiectele. Obiectele de culoare alba sunt cele mai bine vizibile in spectrul razelor Ir, fiind identificate de la distanta cea mai mare.

Acest integrat este construit si alimentat conform figurei :



$R_1 = 50 \, \Omega$; $R_2 = 1K \, \Omega$; $C = 10 \, \mu F$ (polarizat).

Citirea starii acestui senzor se face printr-o intrare **DIP** la microcontroler. Functia de citire a portului respectiv este **J_BIOS_DIP_PINSTATE_INH_P0**. Cu aceasta comanda noi putem afla daca pe senzorul de Ir detecteaza sau nu lumina.

3.3. Circuit detectie lumina.

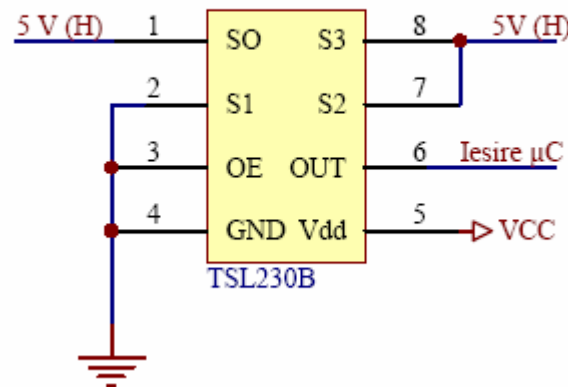
Constructia acestei masini a pornit de la un lucru simplu pe care trebuia sa il indeplineasca. Folosindu-se de senzorii de lumina trebuia ca caute locul cel mai luminos si apoi sa se indrepte catre el. Pentru a putea determina in care parte - stanga sau dreapta - trebuie sa se indrepte masina avem nevoie de doi astfel de senzorii, montati si ei pe botul masinii. A fost adaugat mai apoi senzorul de Ir, care ofera posibilitatea de a ocoli obiectele in timp ce se indreapta catre o sursa de lumina.

Desi cautarea luminii este elementul de baza al acestei masini, masurarea intensitatii luminii se face foarte simplu. Pentru aceasta functie de masurare avem un senzor **TSL230B**. Acest circuit poate fi setat pe diferite grade de sensibilitate. De aceea se poate spune ca este programabil. Iesirea lui este in frecvanta. Semnalul este unul rectangular asemanator PWM-ului, numai ca in acest caz avem tot timpul duty-ul de 50%.

Valoarea care se modifica este doar frecventa ce poate ajunge pana la 1,1MHz, duty-ul ramanand neschimbat. Acest semnal poate comanda microcontrolerul. La lumina de camera frecventa care este generata este de ordinul a 300-700 Hz, la lumina naturala valoarea depasind 4002 Hz pe gradul minim de sensibilitate.

Realizarea circuitului acestui senzor este destul de simplu :

DETECTOR DE LUMINA



In microcontroler avem o rutina care citeste periodic valorile acestor marimi. Citirea acestor marimi se face cu ajutorul a doua functii denumite **PIM (Pulse Input Measurement)**. Cele doua functii de citire au forma:

J_BIOS_PIM_PULSESTAMP_OSS() pentru senzorul stanga,

J_BIOS_PIM_PULSESTAMP_TSS() pentru senzorul dreapta.

Aceste functii returneaza o valoare pe 32 de biti. Valoarea obtinuta este impartita in doua marimi ΔContor , ΔTime . Primi 16 biti cei mai semnificativi dau numarul de fronturi crescatoare in intervalul de 10ms. Astfel daca vrem sa citim frecventa de 100Hz, acesti biti ne vor da valoarea 1, pentru 1 KHz avem A si asa mai departe.

Pentru ΔTime avem alta metoda de calcul. Aici este vorba de un tic. Acest tic are valoarea de 0.8us. ΔTime rezultat ne da de cate ori acest tic poate incapa in frecventa masurata. Deci noi daca avem de masurat o frecventa de 20Hz cu o perioada de 0.05ms daca impartim aceasta valoare la durata unui tic de 0.8us ne va da valoarea 62500 rezultand in hexa F424. In proiectul nostru ne folosim de aceasta marime.

Daca la primi 16 biti valoarea de masurare este direct proportionala cu frecventa la masurarea ΔT_{time} aceasta marime este inversproportionala.

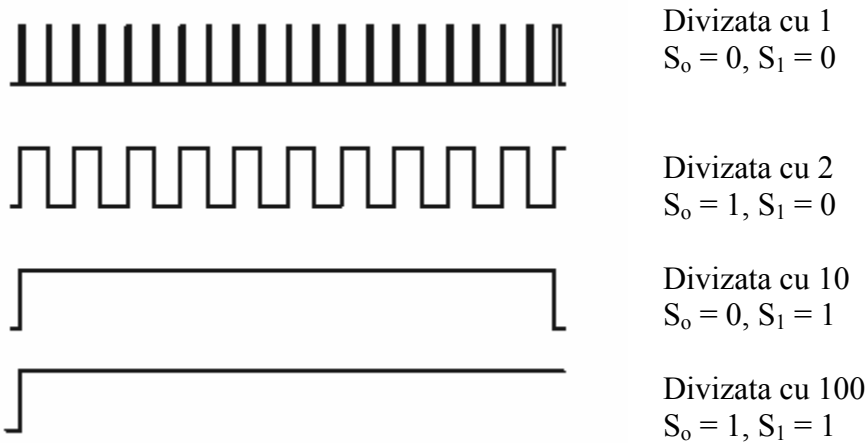
Dupa ce sunt citite valorile ele se compara intre ele. Cele doua marimi se scad dand un rezultat de forma $\Delta f_L = freq_senzor_st - freq_senzor_dr$, aceasta marime este comparata mai apoi cu o valoare stabilita dinainte δ . Daca $\Delta f_L \leq \delta$ atunci masina i-si continua traiectoria care a fost stabilita anterior. In cazul in care inecuatia nu se respecta se verifica care dintre cele doua frecvente este mai mare. Dupa ce determinam acest lucru nu ne ramane de facut decat sa dam comanda la motoare sa vireze in partea in care senzorul are o valoare mai mare. Aceste viraje pot fi luate cu diferite grade, in cazul de fata acest unghi este de 45° . Putem insa imbunatatii acest lucru printr-o analiza directa a frecventelor generate. Tabelul urmator prezinta cateva valori pentru diferite valori ale lui Δf_L .

Valoare Δf_L (Hz)	Unghiul virajului. (grade)
$\leq \delta$	Masina pastreaza traiectoria dinainte stabilita
≤ 20	25°
≤ 40	45°
≤ 60	55°
≥ 60	65°

Deoarece comanda puterii motoarelor se face cu ajutorul PWM-ului, putem crea un raport care sa il inmultim cu acest PWM, si astfel daca el creste la fel va creste si puterea motoarelor luand o curba mai strans. Acest raport poate fi chiar Δf_L . Prin metoda prezentata viteza de reactie a masinii sporeste ajungand mai repede la rezultat. Timpul pe care il va face masina din pozitia A in pozitia B unde se presupune ca exista o sursa de lumina va devenii mai scurt.

Frecventa de iesire la acest senzor arata conform graficului urmator. Se observa ca in primul caz avem un semnal care nu are un factor de umplere de 50%, acesta desi i-si pastreaza perioada timpul cat este pe 1 logic este destul de mic in comparatie cu perioada. El se aseamana mai mult cu un impuls. Acest impuls nu a putut fi receptionat foarte clar

de **PIM** (Pulse Input Measurements). Frecventa de la iesire este divizata conform marimilor prezentate.



S-a incercat folosirea celui mai mic grad de sensibilitate pentru a putea avea o mai mica eroare de masurare dar acest lucru nu a fost posibil. Deoarece microcontrolerul are 1 logic doar pentru 12 V, datele care le genera senzorul erau citite intermitent semnalul neavand destula putere. Astfel se facea o citire la 1 sec, 2 sau chiar 4,5. O data ce am crescut sensibilitatea setand si bitul S_1 pe 1 logic fluxul de date a inceput sa fie continuu, microcontrolerul primind fiecare schimbare de frecventa. Acest chip se poate mandri si cu o viteza de reactie sporita, aceasta are valoarea de **1 puls din noua frecventa plus 1µs**, la fel de rapid este si la schimbarea de programare – modificarea lui S_1, S_2, S_3, S_4 – valoarea in timp fiind de **2 pulsuri din frecventa principala plus 1µs** – frecventa principala este frecventa interna de oscilatie, echivalenta cu impartirea cu 1 a semnalului de iesire.

3.4. Sistemul de comanda si control al motoarelor.

3.4.1. Descriere.

Acesta parte a masinii este creata din mai multe parti. Toate deciziile pe care le luam cu privire la conducerea masinii trebuie sa se aplice in aceasta parte. De aceea functionarea in bune conditii a acestui modul este neaparat necesara dar nu suficienta. El este alcatuit din **unitatea de control** – microprocesorul – unde se iau deciziile in ce priveste directia si sensul de deplasare al masinii, **portile logice SI-NU** unde aceasta

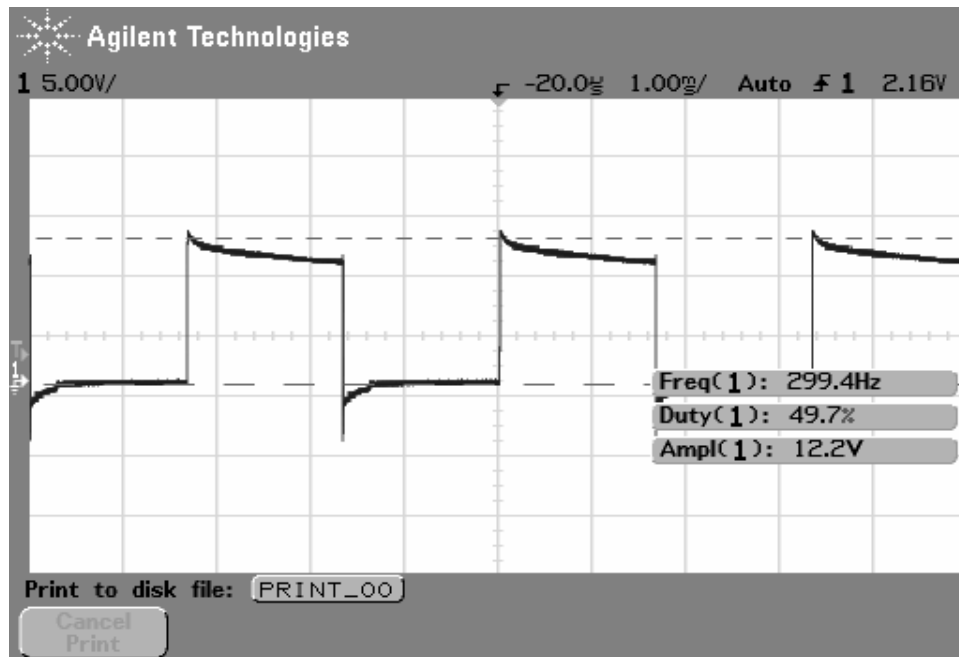
informatie este adunata si directionata catre motorul corespunzator, **driverule** unde informatia logica de mica putere este transformata in informatie de putere mai mare, **motoarele** locul unde se vad toate informatiile si deciziile pe care noi le-am luat bazandu-ne pe senzorii si calcule.

3.4.2. Unitatea de control.

Ca si in cazul senzoriilor este vorba de **ST10F272**, la fel ca la comanda portilor de la emitatoarele de Ir iesirele logice ale unitatii de control sunt niste iesiri de putere la 12V. Si in acest caz am folosit divizoare de tensiune pentru a putea aduce tensiunea de 12V la 5V.

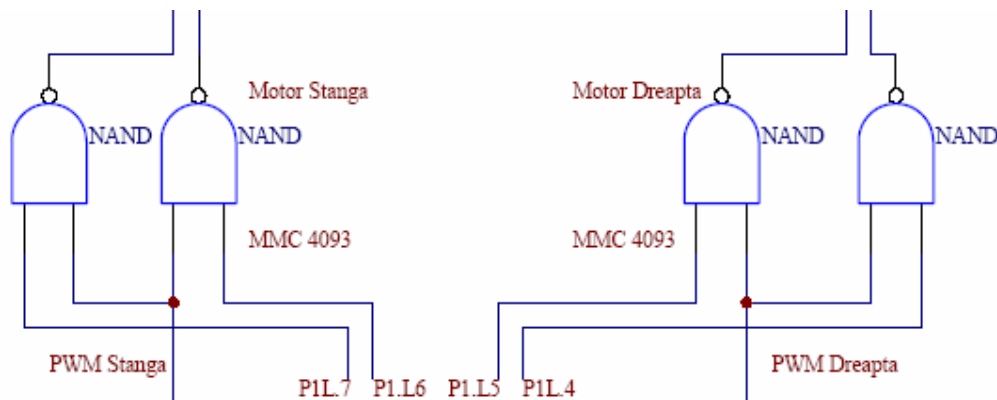
Pentru a comanda motoarele avem nevoie de doua **PWM**-uri, acestea sunt la 12V fiind tot iesiri de putere. Teoretic am fi putut folosi direct aceste iesiri pe motoare scotand din schema driverule (**_parameters**) folosite pentru comanda stanga respectiv dreapta a turatiei motorului. Parametrul **c_test_pwm_parameters** accepta valori intre **0x0000** - **0x3FFF** ce corespund unui factor de incarcare de **0.01 %** - **99.99%**. Frecventa la care este setat PWM-ul este de 300Hz. Cu aceasta frecventa prin experimente am observat ca cele doua motoare incep sa se invarta pe la valoarea **0x1A00**. Valoarea corespunde unui factor de incarcare de **28%**. Pentru a putea misca masina avem nevoie de **0x2000** - **32%**., dar acest lucru ar fi facut imposibila deplasarea masinii cu spatele. Am fi avut nevoie in acest caz de 4 PWM-uri, lucru care nu se putea obtine de la iesirile microcontrolerului, interfata prin care pini acestui chip ieseau nu aveau scose toate PWM-urile din microcontroler. Puterea cu care se invarte fiecare motor este comandat cu ajutorul unor functii ce accepta ca parametru o valoare hexa pe 16 biti. Aceste functii au forma **J_BIOS_PWM_DUCY_LOW_C_LIN_SOL(c_test_pwm_parameters)** si **J_BIOS_PWM_DUCY_LU_LRB_LIN_SOL(c_test_pwm_parameters)**. Aceste PWM-uri ajung in portile logice SI-NU.

Aceste PWM au amplitudinea de 12V, dar dupa cum se observa pe graficul osciloscopului, el are si o tensiune negativa de -5V.



3.4.3. Porile logice SI-NU.

Aici are loc comanda logica a motorului. Folosim 4 astfel de porti, doua pentru motor stanga si doua pentru motor dreapta. Astfel fiecare PWM intra in doua porti logice si in fiecare poarta logica pe langa PWM intra si bitul de comanda dat de microcontroler. Pentru o mai buna intelegere prezentam schema din figura.



Noi putem seta directia de miscare a rotii prin cei 4 pini de comanda P1L.7, P1L.6, P1L.5, P1L.4. Prezentam un table cu starile acestor pini si deirectia de miscare a masinii. Bineinteles gradul cu care maisna ia virajul depinde de PWM dat la poarta.

P1L.7	P1L.6	P1L.5	P1L.4	Directie mers
1	0	1	0	Fata
0	1	0	1	Spate
1	0	0	1	Viraj dreapta
0	1	1	0	Viraj stanga

Portile SI-NU sunt realizate cu circuitul **MMC4093** ce are tehnologie **CMOS**, suportand tensiuni de intrare de pana la 15V, daca este alimentat corespunzator. La fel ca si in cazul comenzii emitatoarelor de Ir am folosit divizor de tensiune pentru a obtine o valoare de 5V. Acesta tensiune de comanda este transmisa mai departe la drivere motoarelor. Nu am folosit direct cei 12V pusi la dispozitie de microcontroler deoarece am fi avut atunci la intrarea driverelor aceasta valoare. Am incercat sa comandam asa driverele deoarece driverul suporta tensiune de pana la 18 V, dar acest lucru a dus la distrugerea unui driver, el neputand curentii dati.

Aceste iesiri logice din microcontroler sunt generate la fel ca si celelate cu ajutorul unor functii. Aceste functii sunt de tipul DIP, si au forma urmatoare :

- pentru comanda drivere stanga:

J_BIOS_DOP_STATE_N_SHIFT_SOL ()

J_BIOS_DOP_STATE_P_SHIFT_SOL ()

- pentru comanda drivere dreapta:

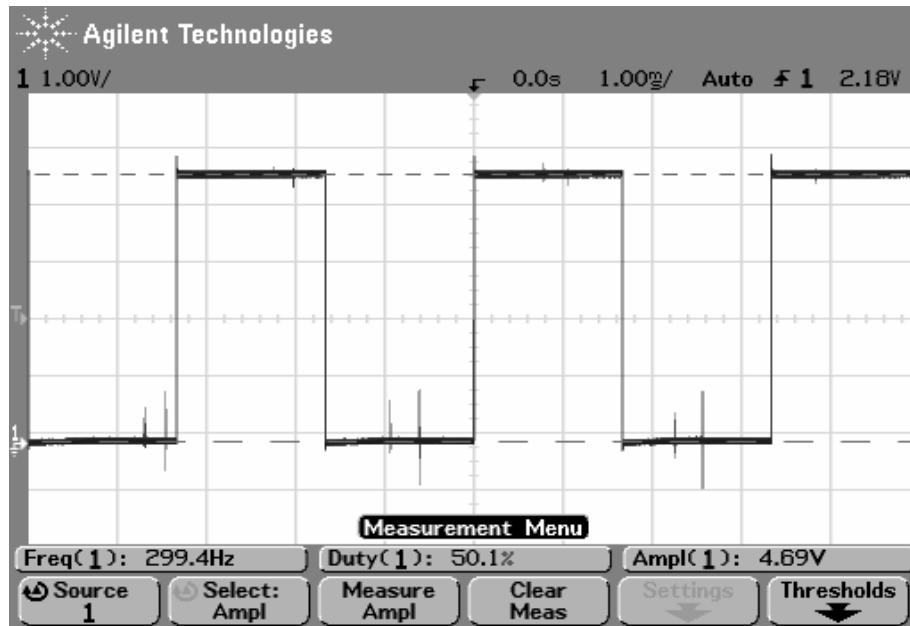
J_BIOS_DOP_STATE_KEY_SOL ()

J_BIOS_DOP_STATE_EEP_TXD ()

Ca si functiile pentru selectarea emitatorului de Ir, aceste functii se pun pe level 0 sau 1 cu ajutorul unor functii derivate din acestea.

Folosind aceste functii se usureaza foarte mult partea de programare a microcontrolerului, astfel avem o functie de setare pe 1 a iesiri si o functie de setare pe 0 a aceleiasi iesiri.

Deoarece au tehnologie Trigger-Schmitt, ele netezesc semnalul prezentat anterior ca fiind iesirea PWM din microcontroler. Acest semnal curatat are forma:



Se observa ca are valoarea de 5V. Chiar daca aveam PWM de 12V, datorita divizorului de tensiune, acestui semnal de comanda i s-a redus tensiunea la una cu care se poate mai sigur.

3.4.4. Driverele.

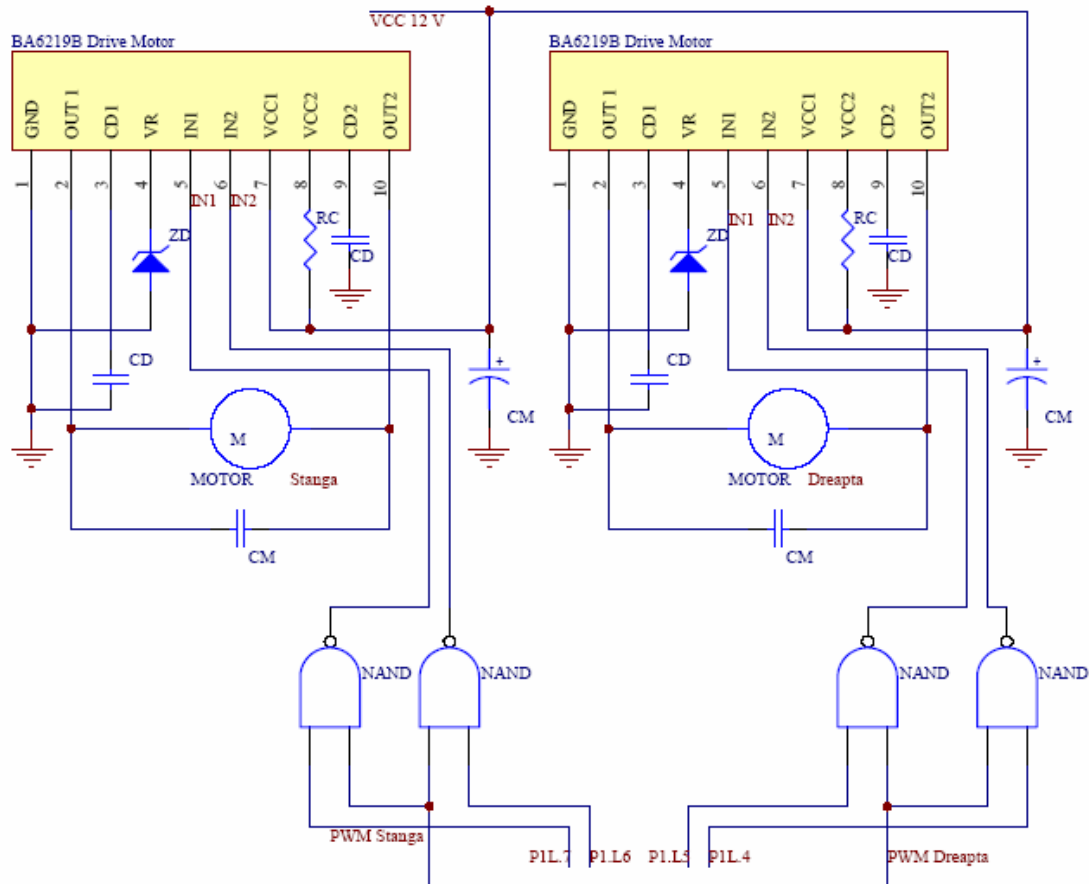
Este locul unde se transforma comanda logica in putere. El este alimentat in acest caz la 12V, dar poate fi alimenta intre 8V si 18V. Are doua tensiuni de alimentare, una pentru functionarea circuitului V_{cc1} si una pentru alimentarea motorului. Tensiunea de intrare de comanda poate merge pana la maxim V_{cc1} , in cazul nostru avem 5V. Pinii ce ne intereseaza pe noi sunt cei ce duc la motor si cei ce primesc comanda. Motorul se leaga pur si simplu la acest driver prin doi pini, comanda este si ea la fel de simpla ea facandu-se tot prin doi pini. Tabelul de comportare al motorului l-am detaliat in alta sectiune.

La folosirea acestui circuit trebuie avut grija la anumite aspecte pentru buna functionare a lui :

- la pini de intrare nu trebuie sa aplicam nici o data tensiune daca circuitul nu este alimentat – riscam sa il stricam -.

- in paralel cu motorul trebuie pus un condensator ce are rolul de a limita **curentii paraziti** atunci cand se schimba directia de rotatie a motorului.
- pinul la care se face alimentarea motoarelor trebuie prevazut cu o rezistenta intre $3\ \Omega$ - $10\ \Omega$ pentru a preveni un scurtcircuit.
- capacitatea C_M (polarizat) este folosita pentru filtrarea tensiunii de alimentare
- dioda Zener de 6,5V are ca rol limitarea tensiunii de alimentare a motoarelor. De aceea aceasta trebuie calculata astfel incat tensiunea de iesire sa nu arda motoarele. In cazul de fata motoarele se alimenteaza pe la 5,5V.

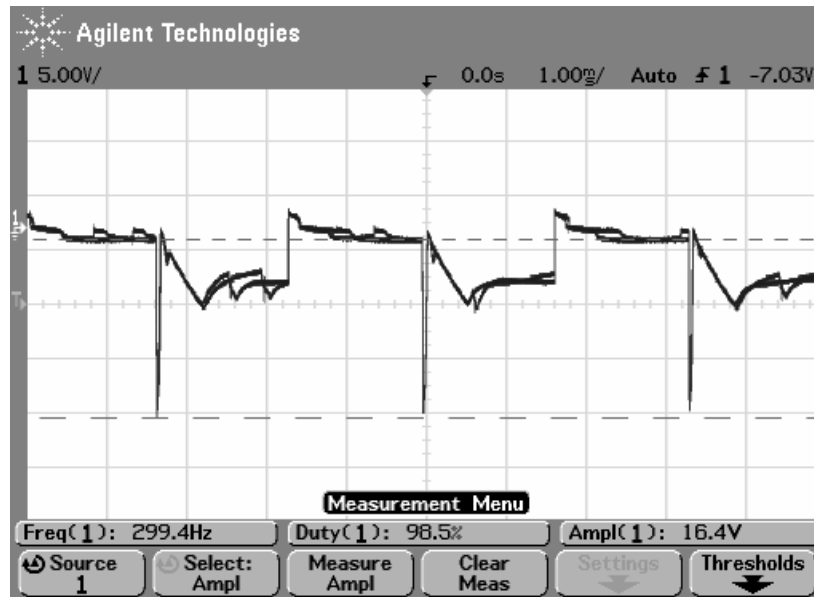
O problema ce trebuie rezolvata la fiecare aplicatie ce contine acest driver este calcularea puterii rezistentei R_c . La proiectul prezentat puterea care trece prin aceasta rezistenta este de 0.7W. De aceea daca aceasta rezistenta este mai mica exista riscul incalzirii si distrugerii piezei. Prezentam in continuare schema acestui modul :



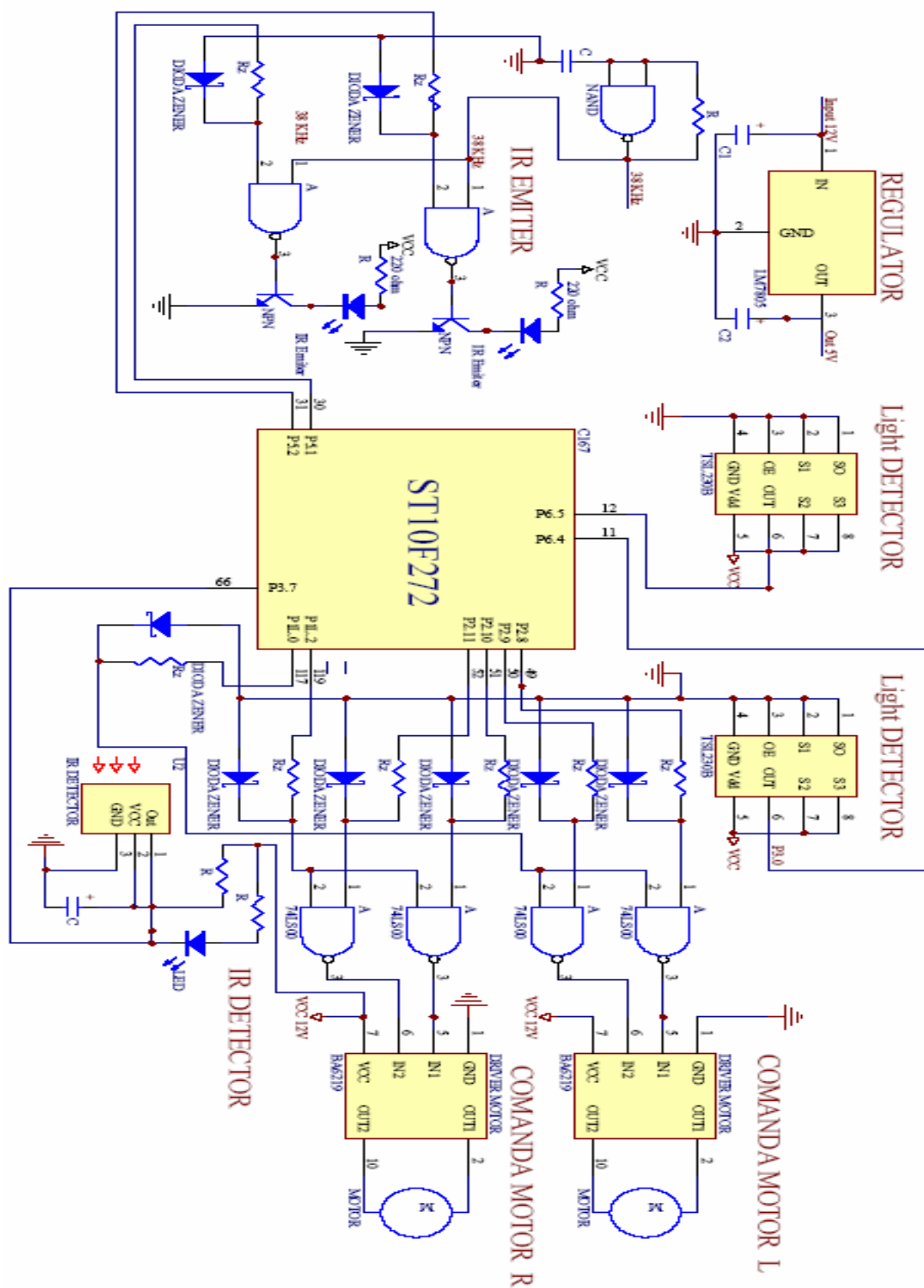
$Z_D = 6.7V$; $C_D = 0.1 \mu F$; $R_C = 10 \Omega$; $C = 100 \mu F$ (polarizat); $V_{CC} = 12 V$; $C_M = 0.1 \mu F$.

3.4.5. Motoarele.

Pentru a putea vira masina s-au folosit 2 motoare. Acestea sunt motoare de curent continuu. Viteza cu care ele se misca depinde de PWM-ul pe care noi il dam din microcontroler. Chiar daca driverul este alimentat la 12V cu ajutorul diodei Zener limitam curentul la 5,5V pentru a proteja motoarele. Deoarece directia masinii se schimba datorita diferentei de viteza dintre cele doua motoare, acestea vor fi pozitionate in partea din fata. Consumul depinde mult si de valoarea PWM-ului. In cazul in care acesta are un factor de incarcare de 32% puterea de motoarelor nu depaseste 1W. Pentru a avea o imagine mai clara asupra tensiunilor ce apar la comanda motoarelor am luat de pe osciloscop caderea de tensiune de pe un motor. Gaficul arata astfel :



Legaturile electronice dintre aceste module sunt prezentate in schema electronica a intregului circuit. Circuitul de divizare apare destul de des deoarece microcontrolerul ne da la iesire PWM, sau date 12V. De aceea trebuie sa facem aceasta reducere de tensiune la fiecare intrare in parte.



Capitol IV.

Realizarea mecanica a masinii.

4.1. Descriere.

Acest scurt capitol evidentiaza problemele ce pot sa apara din punct de vedere mecanic, si tot o data incearca sa ofere o imagine completa asupra realizarii masinii din acest punct de vedere. Deoarece nu avem prea multe parti mobile, exceptie facand doar motoarele proprii-zise si reductoarele lor, probleme mecanice se reduc mai degraba la treburi ce tin de interactiunea masinii cu diferite medii de rulare. In cazul acestei masini aderenta rotilor are un rol major in deplasarea corecta a masinii.

4.2. Constructia masinii.

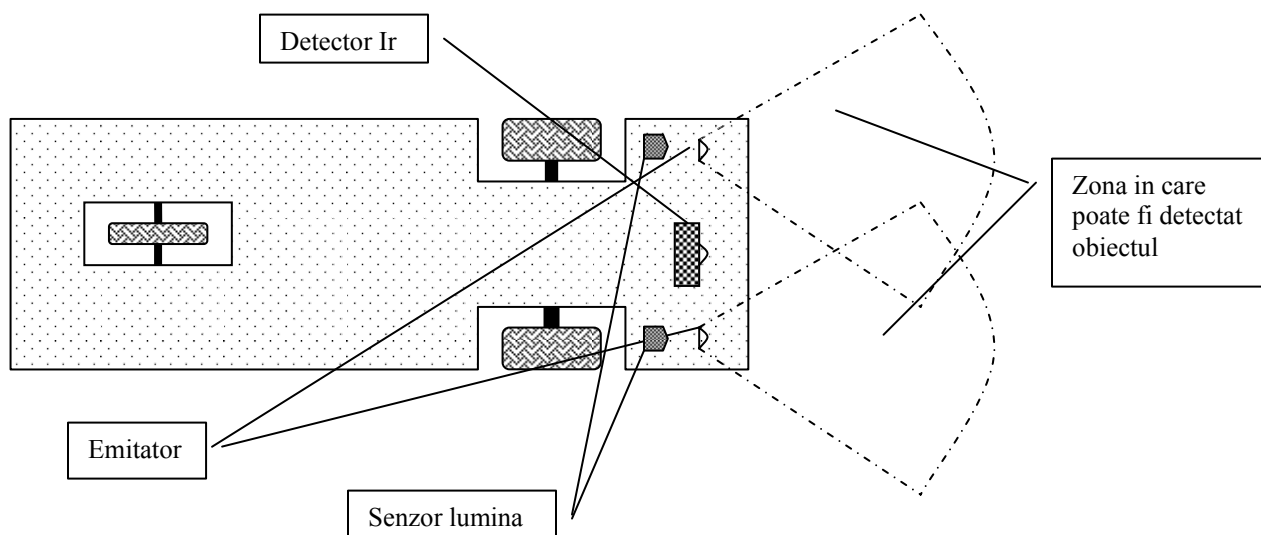
Procesul pe care noi vrem sa il comandam are si cateva parti mobile ce trebuiesc analizate. In primul rand masina are trei roti, doua in fata pe care sunt puse cele doua motoare si o roata in spate care este lasata libera asupra ei nu actioneaza nici un fel de mecanism. Acest tip cu **3 roti** a fost folosit deoarece oferea cateva avantaje mecanice. Deoarece masina a fost construita din mai multe piese gradul de fiabilitate a angrenajelor a scazut proportiona cu numarul pieselor.

Cele 3 roti ofera o aderenta sporita a rotilor din fata si implicit o miscare mai usor de prevazut si calculat. Acest lucru ar fi fost mai greu de atins folosind 4 roti. Deoarece avem tractiunea pe fata, traiectoria pe care masina o face este mai sigura, nu mai exista un cuplu de forte asa de mare ca si in cazul tractiunii pe spate. Un punct in minus la tipul de tractiune folosit este ca masina nu poate vira destul de strans. De asemenea daca masina va da cu spatele directia ei va depinde foarte mult de echilibrul dat de roata din spate. Folosirea doar a unei roti mediane pe spate usureaza si ea forta cu care masina se opune la schimbarea de directie. Aceasta forta ar mai putea scadea si mai mult daca s-ar folosi in loc de roata o bila.

Motoarele ce le-am folosit au si un **reductor**. Acestea este folosit de obicei la motoarele in curent continuu. Aceste motoare au o anumita **viteza minima de rotatie**. De

aceea ele se invart cu o viteza destul de mare. Rolul reductorului este de a transforma aceasta viteza a motorului in putere. Angrenajul este alcatuit din mai multe rotite ce fac aceasta transformare. Piesele electrice sunt asezate pe sasiul masinii. Pentru a ne face o idee de cum arata acest mecanism prezentam un asemenea angrenaj.

Dispunerea motoarelor si emitatoarelor de Ir este urmatoarea.



Capitolul V.

Geometria miscarii propriu-zise.

5.1. Descriere.

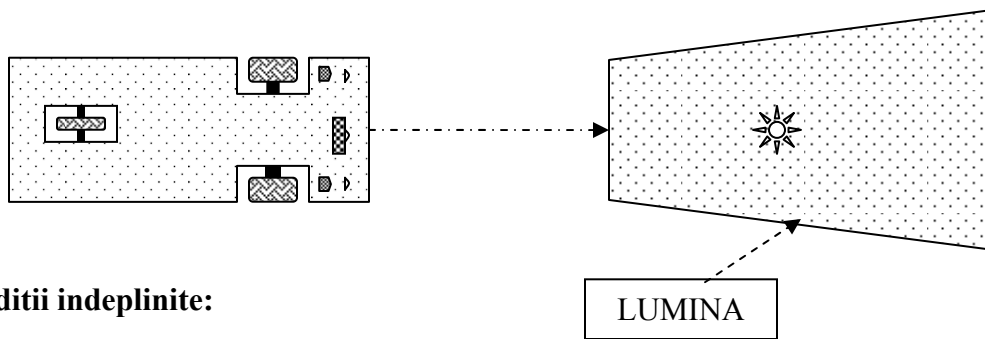
Masina va trebui sa respecte anumite reguli pentru a putea fi siguri ca actiunea care o incepe nu o duce intr-o situatie fara iesire. Vom studia si prezenta in continuare anumite trasee pe care masina le va face in situatii particulare. Aceste situatii putem sa le catalogam ca fiind doua. Una in care masina trebuie sa se indrepte catre o sursa de lumina si a doua ar fi ocolirea obstacolelor ce le va intalni catre aceasta sursa de lumina.

5.2. Ghidarea catre lumina.

Aceasta miscare presupune orientarea si deplasarea masinii catre sursa de lumina. Pentru acest lucru noi avem nevoie de 2 senzori de lumina pentru a face diferenta intre ei si a ne da seama in ce parte este sursa de lumina.

Din softul scris pe microcontroler noi am setat o anumita valoare sub care masina nu mai considera acea valoare sursa de lumina. Aceasta este de aproximativ **400Hz**. Am folosit aceasta procedura pentru a putea elimina alte surse de lumina, astfel masina nu va considera fereastra sursa de lumina. Sursa de lumina folosita in acest caz va fi o lanterna. Microcontrolerul va citi cei doi senzori de lumina si pe baza algoritmului va decide in ce parte este sursa de lumina. Aici putem identifica trei situatii.

5.2.1. Deplasarea in fata.



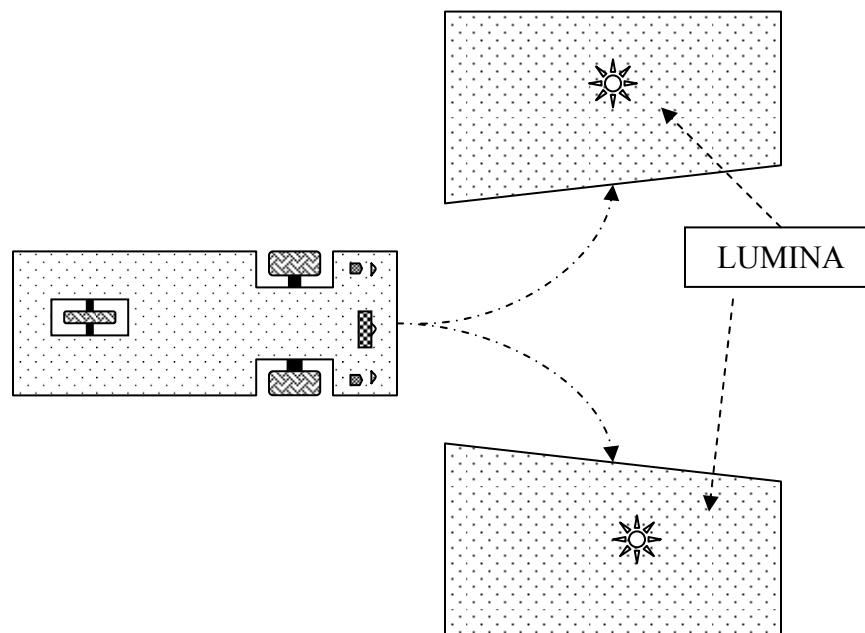
Conditii indeplinite:

- $frec_st > 400 \text{ Hz}$

- $frec_dr > 400 \text{ Hz}$
- $|frec_st - frec_dr| < 50 \text{ Hz}$

Daca sursa de lumina se afla aproximativ in acest trapez masinuta va merge in fata fara a lua nici un viraj. Acest lucru este realizat cu metoda prezentata anterior. Se introduce o diferenta minima – o **eroare** - intre cele doua valori ale luminii sub care microcontrolerul considera ca sursa se afla in fata si da comanda ca atare. Acest lucru previne situatia de oscilatie prea deasa a masinii. Daca nu am fi introdus aceasta valoare masina si la cea mai mica diferenta intre valorile senzorilor ar fi schimbat directia. Masinuta va merge in fata daca diferenta intre cei doi senzori este sub 50Hz.

5.2.2. Deplasarea in lateral.



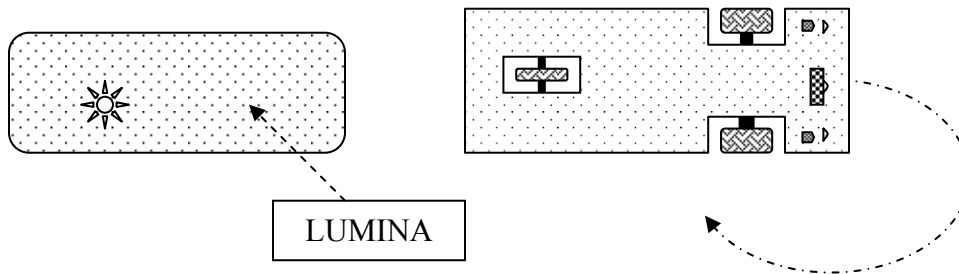
Conditii indeplinite:

- $frec_st > 400 \text{ Hz}$
- $frec_dr > 400 \text{ Hz}$
- $|frec_st - frec_dr| > 50 \text{ Hz}$

Daca diferenta celor doi senzori este mai mare decat eroarea considerata de noi atunci pe baza comenzii data de la microcontroler va vira in partea in care intensitatea

luminii este mai mare. Acest viraj consta de fapt intr-o miscare de spin, o roata se va deplasa in spate si cealalta in fata, din aceasta combinatie rezultand schimbarea directiei masinutei. Aceasta miscare se va realiza pana cand diferenta dintre cei doi senzori va deveni mai mica decat eroarea calculata de noi. Dupa realizarea acestui lucru masina va trece in cazul **a**. caz in care masina se va deplasa in fata pana cand diferenta intre senzori va creste peste aceasta valoare.

5.2.3. Miscarea in care se cauta lumina.



Conditii indeplinite:

- $frec_st < 400 \text{ Hz}$
- $frec_dr < 400 \text{ Hz}$

In acest caz sursa de lumina nu influenteaza direct senzorii nostri. De acest caz era vorba cand vorbeam de eliminarea surselor parazite de lumina. Exista posibilitatea ca miscand sursa de lumina aceasta sa fie undeva unde nu mai poate sa influenteze senzorii asa de mult. Deoarece am considerat ca experimentul nu se desfasoara intr-o camera obscura inseamna ca in acea incapere mai sunt si alte surse care nu ne intereseaza. De aceea inainte de a porni experimentul cu ajutorul senzorilor vom putea masura lumina in camera ce provine din alte surse. Dupa ce am masurat in mai multe locuri vom considera valoarea **maxima + 50 Hz** pragul sub care masinuta va intra in procedura de cautare.

Aceasta procedura presupune efectuarea un viraj la dreapta pana cand un senzor va detecta ceva. In cele din urma masina se va intoarce destul pentru a putea capta lumina de la sursa noastra de tensiune. Dupa aceasta cautare masinuta va intra in procedura de la punctul **a** sau mai sigur de la punctul **b**.

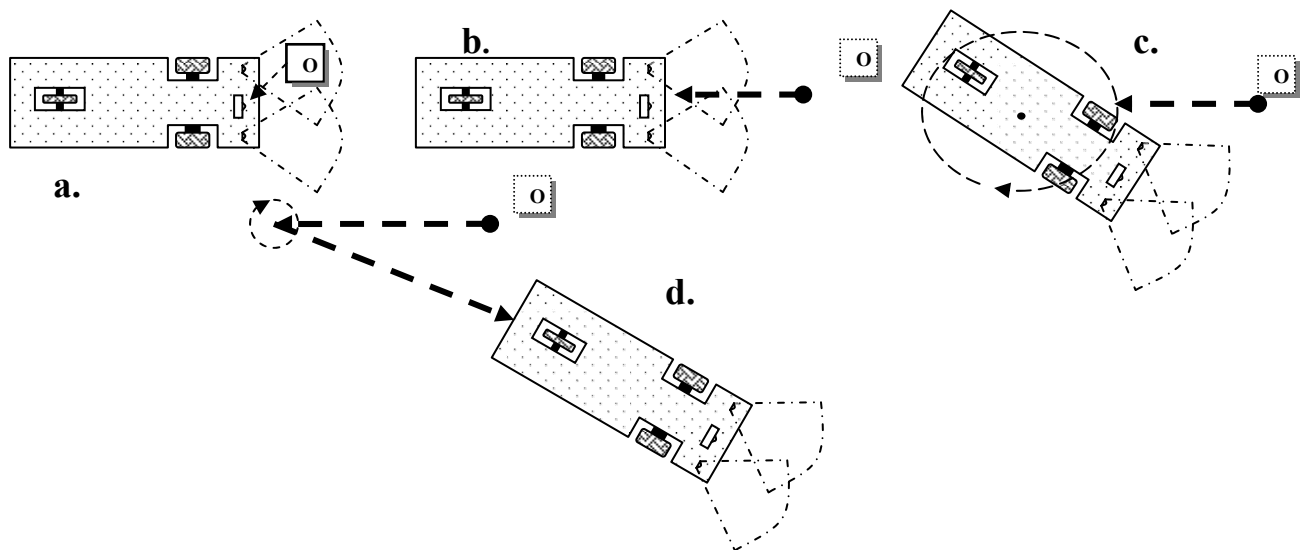
Cele trei cazuri cuprind aproape toate situatiile, singurele necunoscute aici sunt cele legate luminozitatea ambientului de ambient si puterea sursei de lumina. Aceste valori trebuie citite si scrise in program la inceputul experimentului. Masina este capabila

sa urmareasca o sursa de lumina ce nu are pozitia constanta. Deoarece senzorii de lumina sunt foarte rapizi si citirea lor se face si ea in timpi ce tin de ordinul milisecundelor, timpul de reactie al masinii fata de schimbarea pozitiei sursei de lumina este extrem de mic.

5.3. Evitarea obstacolelor.

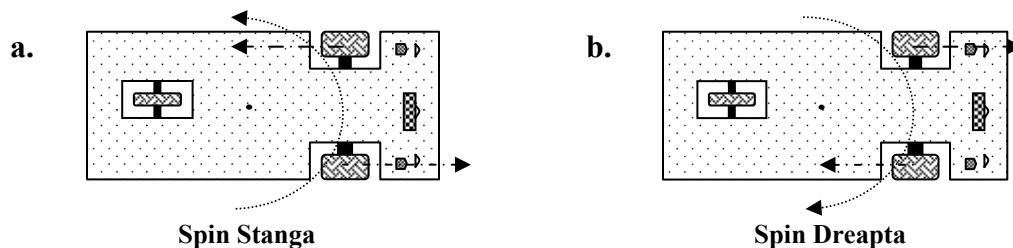
Aceasta caracteristica a masinii vine in completarea celei prezentate anterior. Complexitatea miscarii creste si de aceea analiza va fi putin mai complexa. Gradul de « inteligenta » a sistemului comandat depinde in mare masura de numarul de valori de intrare. Un sistem daca este bine conceput poate procesa multe date si poate lua decizii ce ii confera o stabilitate in timp, dar nimic din puterea de procesare nu se compara cu informatia ce este primita din exterior. De aceea senzorii sunt un lucru vital pentru sistemele ce trebuie sa se deplaseze, sa coordoneze, sa conduca un sistem. Pentru a putea functiona orice sistem are nevoie de un numar minim de senzori. In cazul de fata senzorii de lumina ar fi fost necesari si suficienti, dar cum traim intr-o lume cu « obstacole » am introdus si **senzorii de proximitate** pentru a putea evita obstacolele ce sunt aici sub forma de cuburi albe. Si aici identificam mai multe situatii de conducere a masinii.

5.3.1. Obiectul ce trebuie evitat este detectat in partea stanga.



În acest caz obiectul detectat se afla în partea stângă. După ce senzorul de pe acea parte l-a detectat mașina intră în procedura de ocolire. Microcontrolerul va da comanda ca timp de 3 secunde mașina să dea cu spatele. Acest timp considerăm că este suficient pentru a putea să ne îndepărtăm de obiect. Deoarece avem tracțiunea pe față acest timp de mișcare cu spatele va fi puțin mai nesigur. Cuplul de forțe care se formează pot face ca mașina să meargă într-o parte. Acest lucru poate fi evitat prin scurtarea distanței parcurse și prin micșorarea vitezei de deplasare.

După efectuarea acestei mișcări, mașina va face o mișcare de spin unghiul de rotație fiind de 45° . Această mișcare se realizează prin rotația în sens diferit a motoarelor mașinii.



O dată ce mașina a efectuat mișcarea de spin tot din microcontroler îi dăm comanda ca timp de 1 secundă să se deplaseze în față. Această comandă este necesară pentru a împiedica mașina să facă aceeași mișcare de mai multe ori. Între sursa de lumină și mașina ar putea exista acest obiect. Dacă nu am fi obligat mașina să se deplaseze în față fără să citească senzorii de lumină, după mișcarea de spin ea ar fi detectat sursa de lumină pe direcția obiectului și ar fi încercat iar aproximativ aceeași mișcare, detectând iar obiectul. Această mișcare de detecție și ocolire ar fi putut continua până când ar fi rămas fără curent mașina. Pentru a ocoli această situație am introdus mișcarea în față.

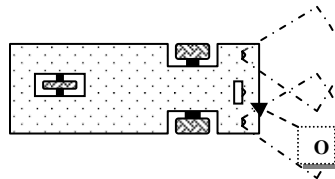
După ce am terminat de executat această procedură de ocolire vom începe iar să citim senzorii de lumină și de Ir, pentru a determina direcția în care trebuie să se îndrepte mașina.

Situația **a.** este folosită dacă obiectul ce trebuie evitat se afla în partea dreaptă, cazul **b.** fiind pentru situația în care obiectul se afla în partea stângă. Trebuie ținut cont în aceste situații ca vitezele celor două roți și aderența lor la sol să fie aceeași pentru a putea imprima mașinii mișcarea dorită. Alimentarea celor două motoare se face de la aceeași

sursa de curent pentru a putea elimina diferentele de alimentare, si implicit diferentele de rotatie al motoarelor.

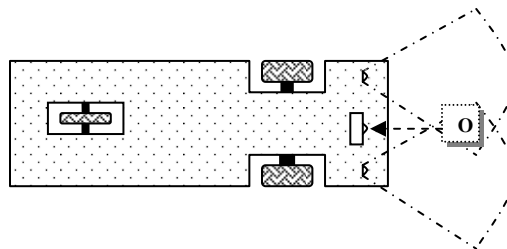
5.3.2. Obiectul ce trebuie evitat este detectat in partea dreapta.

Situatia este prezentata in figura urmatoare :



Emitatorul primește semnal de la senzorul de pe partea dreapta. Datorita alternantei emiterii, ne vom putea da seama ca raza Ir provine de la emitatorul din dreapta. In acest caz intram in procedura de evitare a obiectului ce se afla in partea dreapta. Aceasta este exact ca in cazul **a.** singura diferenta fiind sensul de spin al masinii care va fi acum in partea stanga, parametrii de functionare se pastreaza aceeasi.

5.3.3. Obiectul se afla in zona in care emit cele 2 leduri de IR.



In acest caz obiectul care vrem sa il ocolim este in raza ambilelor emitatoare. Aceasta situatie se poate rezolva destul de simplu. Cei doi senzori emit alternativ, deci chiar daca teoretic obiectul se afla in zona ambelor emitatoare doar unul dintre ei este activ. Verificand care luam decizia ca si cum obiectul s-ar afla in totalitate in partea emitatorului respectiv. De aici procedura se continua dupa cum am deschis-o in **a.** sau **b.**

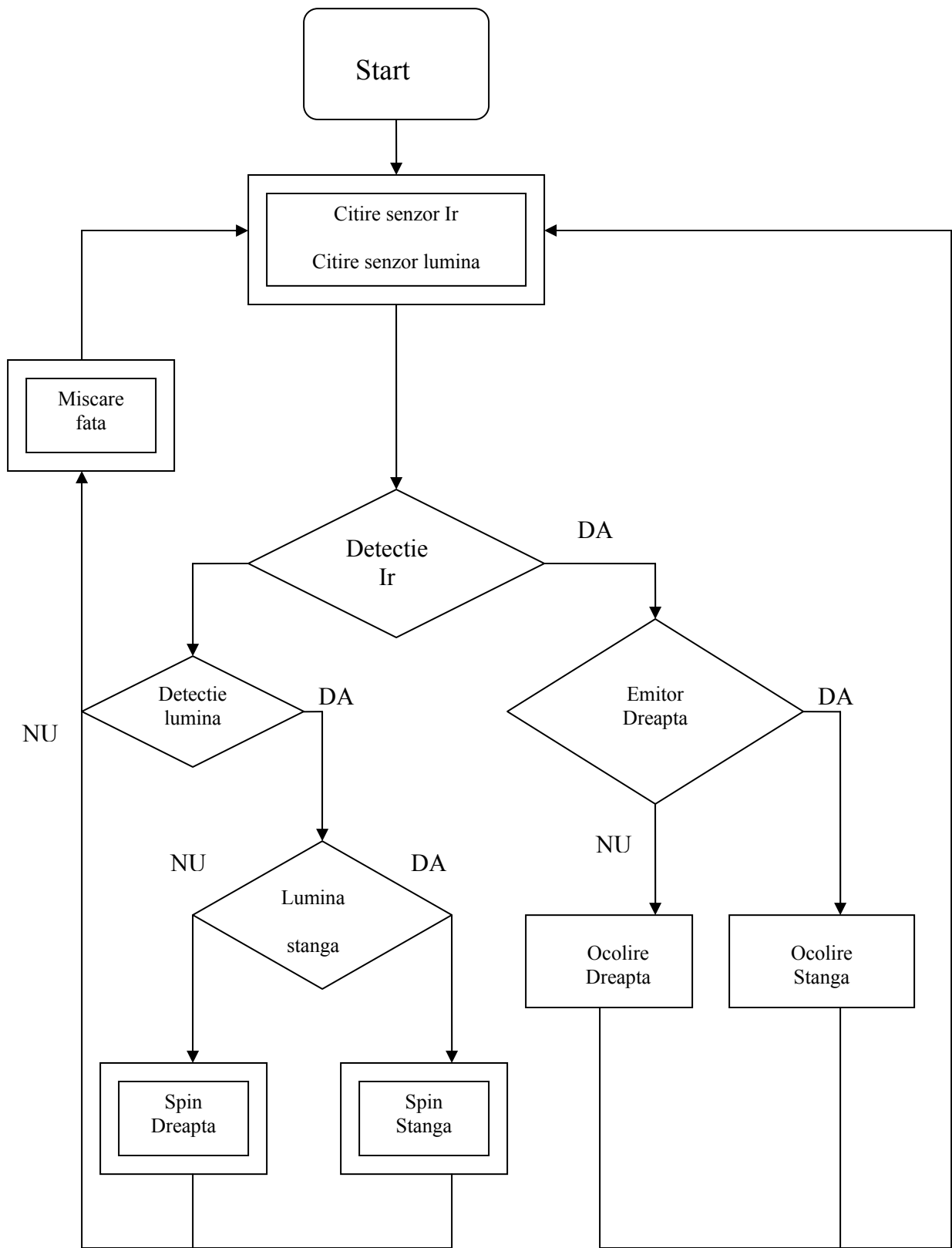
Capitolul VI.

Realizarea practica a softwarului de comanda si control

6.1 Fluxul de informatii.

Observam pe graficul fluxului informational program nu are un sfrasit. Masina citeste in continu date de la senzori si se comporta ca atare. Atunci cand ea detecteaza o raza de Ir intra in procedura de ocolire. In aceasta procedura microcontrolerul nu mai citeste nici o data, el in acest punct da numai comenzi de ocolire a obstacolului detectat. Acesta procedura presupune: mers cu spatele timp de 500ms, invartire pentru 200 ms, si mers cu fata tot 200ms.

Dupa aceasta procedura masina va citi senzorii si daca nu mai exista nici un obstacol, va tine cont de senzorii de lumina. Dupa analiza lor ia decizia in ce parte se va indrepta. Aceasta poate sa fata o miscare de rotatie pana fixarea masinii pe traectorie sau daca este deja pe traectorie ea deja va incepe sa se deplaseze spre acel loc. In orice moment in fata ei poate sa apara un obstacol. Daca acest lucru se intampla masina se va opri controlul ei fiind luat de procedura de ocolire.



6.2. Prezentarea procedurilor de comanda.

Programarea acestui microcontroler este bazat pe taskuri. Acestea sunt module care se ruleaza la intervale diferite de timp. In acest caz am beneficiat de taskuri de 1ms, 10ms, 20ms, 40ms, 320ms. O data initializat microcontrolerul se incepe cu rularea lor. Astfel dupa trecerea a 1ms se executa instructiunile din tasckul de 1 ms. La 2 ms la fel. Asta pana la 10ms. Aici se rula taskul de 1ms si apoi cel de 10ms. La 320 de ms se executa toate tasckurile prezentate.

Putem spune ca tackul cel mai prioritar este cel de 1ms. In acest tack este pus codul critic care are nevoie de timp de respuns foarte mic. Pentru procesele ce trebuiesc condu-se rapid codul scris se va imlementa in acest task. La procesele care nu necesita o viteza de reactie asa de mare ele vor putea fi puse si in taskurile cu timpi de executie mai mari.

Pentru a putea usura programarea am definit niste functii ale nostre care numai schimba numele functilor aratate anterior. Cateva din aceste schimbari sunt prezentate in randurile urmatoare.

```

/* stare senzor lumina */
#define state_senzor_light_left() J_BIOS_DIP_PINSTATE_INH_R ()
#define state_senzor_light_right() J_BIOS_DIP_PINSTATE_INH_N ()

/* comanda motor stanga*/
#define comand_Motor_leftA_level0() J_BIOS_DOP_LEVEL0_LC_SHIFT_SOL()
#define comand_Motor_leftB_level0() J_BIOS_DOP_LEVEL0_L_RB_SHIFT_SOL()
#define comand_Motor_leftA_level1() J_BIOS_DOP_LEVEL1_LC_SHIFT_SOL()
#define comand_Motor_leftB_level1() J_BIOS_DOP_LEVEL1_L_RB_SHIFT_SOL()

/* comanda motor dreapta*/
#define comand_Motor_rightA_level0() J_BIOS_DOP_LEVEL0_P_SHIFT_SOL()
#define comand_Motor_rightB_level0() J_BIOS_DOP_LEVEL0_N_SHIFT_SOL()
#define comand_Motor_rightA_level1() J_BIOS_DOP_LEVEL1_P_SHIFT_SOL()
#define comand_Motor_rightB_level1() J_BIOS_DOP_LEVEL1_N_SHIFT_SOL()

```

```

    /* comanda emitator Ir stanga */
#define comand_Ir_left_level0(low) J_BIOS_PWM_DUCY_PL_LIN_SOL(low)
#define comand_Ir_left_level1(hight) J_BIOS_PWM_DUCY_PL_LIN_SOL(hight)

    /* comanda emitator Ir dreapta */
#define comand_Ir_right_level0(low) J_BIOS_PWM_DUCY_HC_LIN_SOL(low)
#define comand_Ir_right_level1(hight) J_BIOS_PWM_DUCY_HC_LIN_SOL(hight)

```

Aici am redefinit functiile ce le folosim in program. Parametri lor raman neschimbati doar numele lor este apropiat de functiile ce le inceplinesc in program.

Cel mai important lucru al acestui program este detectia obiectelor la timp. Daca un obiect nu a fost detectat in timp util sansele ca el sa pericliteze buna functionare a masinii sunt destul de mari. Deoarece este asta de important ii dam prioritatea cea mai mare. De aceea programul pentru detectia in Ir a obiectelor se va pune in taskul de 1ms.

Programul are structura :

```

/* detectie Ir, comada masina */
{if ((state_senzor_TSOP())&&(!reverse)&&(!spin)&&(!forward))
{
    stop();
    state_TSOP = 1;
    reverse = 1;
    counter_t = 0;
}
if (state_TSOP)
{
    switch (counter_t)
    {
        case 500:
        {
            stop();
            reverse = 0;
            spin = 1;
            break;}
    }
}

```

```
        case 700:
        {
            stop();
            spin = 0;
            forward = 1;
            break;
        }
        case 900:
        {
            stop();
            forward = 0;
            state_TSOP = 0;
            break;
        }
    }
    counter_t++;
    if ( reverse ) move_back();
    if (spin)
    {
        if (led_state) spin_left();
        else spin_right();
    }
    if (forward ) move_forward();
}
```

In aceasta procedura ne-am folosit de repetitia data de task care ruleaza la 1 ms. Pentru a putea masura timpul de comanda ne-am folosit de o variabila `counter_t` care la fiecare rulare se incrementeaza cu 1. Aceasta incrementare porneste doar cand este detectat de functia **state_senzor_TSOP()** un front pozitiv (raza Ir). In acest caz vom pune pe 1 flagul **state_TSOP** care blocheaza celelalte proceduri in executie. De aici incepe incrementarea variabilei **counter_t**. Tot o data folosim si trei flaguri pentru a stii in ce faza a acestei proceduri ne aflam. La inceput avem flagul de **reverse**. Aceasta variabila

este activa timp de 500×1 ms. In acest timp dupa case din procedura se face o verificare a acestor flaguri cu ajutorul lui if. Cele 3 if-uri pot fi active pe rand. Astfel in prima perioada de 500ms doar procedura de dat cu spatele functioneaza adica **move_back()**.

Dupa ce trece aceasta perioada flagul de reverse se reseteaza si se pune pe 1 cel de **spin**. Operatiunea aceasta dureaza 200 ms. In acest timp masina va face o rotatie in partea opusa obiectului detectat. Procedura este **spin_left()** si **spin_right()**.

Ajungem apoi la timpul de 700 de ms. Aici flagul spin se reseteaza si avem **forward** pe 1. Masina va merge in fata timp de 200ms cu ajutorul functiei **move_forward()**.

Terminand toata aceasta miscare resetam si flagul **state_TSOP** permitand si celorlalte proceduri sa functioneze. In tot acest timp in care s-a desfasurat procedura **nici un senzor nu mai era receptionat**.

In continuare prezentam pcedura de emitere a Ir. Deoarece nu este asa de importanta aceasta procedura poate fi folosita in takul de 320ms.

```
/* emitere IR */
```

```
if (counter % 2 == 0)
```

```
{
```

```
if (!state_TSOP)
```

```
{
```

```
    if (led_state)
```

```
    {
```

```
        comand_Ir_left_level0(0x0000);
```

```
        comand_Ir_right_level1(0x3fff);/* comanda emitator dreapta */
```

```
        led_state = 0;
```

```
    }
```

```
else
```

```
{
```

```
    comand_Ir_right_level0(0x0000);
```

```
    comand_Ir_left_level1(0x3fff);/* comanda emitator dreapta */
```

```
    led_state = 1;
```

```
}
```

```
}}counter++;
```

Aici noi emitem alternativ pe cele doua emitatoare raze Ir. Timpul in care emite fiecare led este $320\text{ms} \times 2$. Si aici pentru a putea schimba starile unui flag am folosit un **counter**. El se incrementeaza la fiecare 320ms. Deoarece se afla valoarea de adevar a impartirii lui cu 2 timpul de emisie a unui led de Ir creste la 640ms. Folosim flagul **led_state** pentru a putea determina in taskul de 1ms in ce parte este obiectul. Noi stim ca daca led_state este 0 emisia s-a facut in partea dreapta si sunt sanse foarte mari ca obiectul sa se afle preponderent in acea zona.

Se observa aici flagul state_TSOP. Acesta devine activ cand se intra in procedura de ocolire, de aceea acest task nu se executa daca masina a gasit un obiect in fata.

Pana aici am vorbit de detectia de Ir si ocolirea obstacolelor. In continuare prezentam cum se face orientarea catre lumina. Pentru aceasta incepem cu programul.

```
/* detectare lumina comanda motoare*/
```

```
if( !state_TSOP){
if ((state_senzor_light_left() > limits) || (state_senzor_light_right() > limits))
    if (state_senzor_light_left() + dif > state_senzor_light_right())
        {
            spin_left();
        }
    else if (state_senzor_light_left() < state_senzor_light_right() + dif)
        {
            spin_right();
        }
    else move_forward();
else spin_left();
}
```

Aici avem procedura ce face scopul intregului proiect, detectia luminii si deplasarea catre aceasta. Acesta procedura este realizata in taskul de 10ms. Se observa si aici flagul de validare state_TSOP. Si aceasta procedura nu functioneaza daca masina s-a inregistrat intr-o procedura de ocolire a unui obiect. La inceput se verifica daca senzorii capteaza lumina peste un anumit prag. Daca acest lucru nu se intampla vom roti masina

catre alta directie in cautarea de lumina. Pragul acesta este dinainte ales, el depinzand foarte mult de locul unde se desfasoara experimentul. El este tinut minte in variabila pe 16 biti **limits**.

Presupunem ca senzorii detecteaza lumina. Se face o comparare pentru a vedea care dintre ei are o valoare mai mare determinand astfel zona in care se afla sursa de lumina. In aceasta comparare introducem si o marime de eroare. Aceasta foloseste pentru a determina masina sa aiba o anumita inertie in miscare. Daca nu ar fi introdusa aceasta valoare masina s-ar fi indreptat catre sursa de lumina intr-o miscare sinusoidala. Cu aceasta marime acesti caracteristici sunt atenuate.

Dupa ce se face compararea masina se va invarti pana cand valoarea intre cele doua marimi ale senzorilor va deveni mai mica decat acest **dif**. Daca s-a atins aceasta stare masina va merge in fata pana cand va intalni un obiect sau pana cand diferenta dintre valorile senzorilor de lumina va creste iar peste valoarea dif. Atunci va avea loc alt reglaj automat identic cu cel prezentat.

Procedurile folosite anterior au descrierea in urmatoarele randuri.

```
void stop()
{
    PWM_Motor_left(0x0000);
    PWM_Motor_right(0x0000);
}
```

Acesta procedura este folosita pentru a opri motoarele. Valoarea 0 in hexa are acest rol. Procedura se foloseste de obicei mai mult pentru a ne asigura ca regimul de lucru care va urma nu interfereaza cu cel trecut. Este o procedura de sigurant.

```
void move_back()
{
    comand_Motor_leftA_level0();
    comand_Motor_leftB_level1();
    comand_Motor_rightA_level1();
    comand_Motor_rightB_level0();
    PWM_Motor_left(0x1000);
}
```



```
PWM_Motor_right(0x1000);  
}
```

Aici in primi comandam motoarele sa se invarta in sensul invers de mers. In ultimele doua randuri dam comanda puterii motoarelor.

```
void move_forward()  
{  
    comand_Motor_leftA_level1();  
    comand_Motor_leftB_level0();  
    comand_Motor_rightA_level0();  
    comand_Motor_rightB_level1();  
    PWM_Motor_left(0x1000);  
    PWM_Motor_right(0x1000);  
}
```

La fel ca la procedura anterioara, singura diferenta fiind sensul de rotatie al rotilor. Acum ele merg in fata. Puterea motoarelor ramane aceeasi.

```
void spin_left()  
{  
    comand_Motor_leftA_level1();  
    comand_Motor_leftB_level0();  
    comand_Motor_rightA_level1();  
    comand_Motor_rightB_level0();  
    PWM_Motor_left(0x2000);  
    PWM_Motor_right(0x2000);  
}
```

In aceasta procedura avem rotatia la stanga. Puterea la aceste rotatii trebuie sa fie mai mica. Acest lucru determina un reglaj mai bun pe traiectoria sursei de lumina. Daca viteza ar fi mai mare masina din inertie s-ar deregla, necesitant un reglaj suplimnetar.

```
void spin_right()
```

```
{  
    comand_Motor_leftA_level0();  
    comand_Motor_leftB_level1();  
    comand_Motor_rightA_level0();  
    comand_Motor_rightB_level1();  
    PWM_Motor_left(0x0000);  
    PWM_Motor_right(0x0000);  
}
```

La fel cu paragraful anterior putem spune si de aceasta procedura. Putere mica in roti pentru un reglaj mai bun.

Capitolul VII

Concluzii

În prezenta lucrare s-a încercat prezentarea unui sistem de reglare automată. Acest sistem nu are o aplicare imediată, el fiind doar un proiect de laborator. S-a încercat doar arătarea principiilor de bază la unele tehnologii. Dacă piesele din care este construit sunt bine calibrate sistemul poate funcționa destul de bine.

Fiind o mașinuță o caracteristică principală la ea ar fi autonomia, atât de curent cât și de resurse. Din păcate acest lucru nu a fost îndeplinit. Chiar dacă este o « jucărie » consumul de curent este destul de mare. În perioadele de testare o baterie de 9V nu ajungea unui timp mai mare de câteva ore. Acesta a fost și unul dintre motivele care a dus la alimentarea mașinuței de la o sursă externă de 12V. Dependința de resurse s-a datorat necesității de folosire a unui microcontroler. Acest tip de microcontroler este folosit în industria de autovehicule fiind unul destul de rezistent și sigur. Pentru a putea folosi funcțiile microcontrolerului a fost necesară folosirea unui JATCO, o cutie ce intermediază schimbul de informații al lui ST10F272 cu exteriorul. De la acest lucru a plecat și dependența de mașinuței de resurse. Ea nu funcționează autonom din cauza acestor motive.

Deși echipamentul mașinii este destul de minimal sistemul creat se poate comporta ca un sistem inteligent luând decizii pe baza informațiilor captate de la mediu. Cum am mai spus această mașină simulează comportarea simplă a unui gândac.

Se poate mult îmbunătăți și aici nu este vorba de mecanisme mai performante ci pur și simplu de alte componente. În primul rând se poate pune un sonar care poate astfel să detecteze nu numai obiectele ci și distanța până la ele. Se pot pune senzori de înclinare pentru a determina denivelarea terenului și astfel să creștem sau să scădem puterea motoarelor dacă este nevoie. Pe această mașinuță pot fi puse diferite detectoare de substanțe, ea putând intra în locuri ostile omului. Dacă această mașinuță ar fi fost echipată cu baterii solare autonomia lui ar fi crescut considerabil. Toate aceste echipamente desigur îmbunătățesc considerabil comportamentul mașinuței.

O problema de mecanica care a existat a fost : ce tip de cauciucuri sa se foloseasca. Dupa cum se stie angrenajul este sustinut de 3 roti cu 2 roti motoare. Folosirea senilelor ar fi fost o solutie destul de buna. Ar fi crescut in felul acesta foarte mult si adereanta. Dar si aici avantajele ar fi venit cu un consum ridicat de curent, frecarea in acest caz fiind mai mare.

Senzorii sau comportat destul de bine. O problema a aparut la calibrarea emitatoarelor de Ir. Deoarece sunt doar niste leduri care emit in infrarosu, raza de emisie este destul de restransa, existand riscul ca obiectele sa nu poata fi identificate daca masina se misca prea repede. Raza de actiune se poate creste crescand curentul prin ele.

Un avantaj major al acestor sisteme este ca ele pot lua singure decizii. Daca sunt bine programate ele pot fi folosite unde erorile umane poat sa apara destul de des din cauza oboselii, stresului sau pur si simplu a mediului ostil. Deja in aviatie se construiesc, testeaza si chiar folosesc avioane fara pilor. Din cauza rezistentei slabe a oamenilor aceste sisteme erau inainte limitate la acceleratii de max 10G pe perioade scurte. Fara piloti ele pot ajunge si pana la 15G - 16G, acceleratii unde pilotii de mult ar fi fost inconstienti.

Desi tot timpul la anumite proiecte se cauta sa se foloseaca in tehnologii avansate, si folosirea lor la scara industrială este un beneficiu. Tipuri asemanatoare de roboti sunt folositi la industria constructoare de masini, Aici un intreg preces putand fi condus de foarte putini oameni. Din cauza automatizarii ei nu mai depun o munca efectiv de productie ci mai mult asista ca mecanisme care produc sa aiba parametrii optimi de functionare.

Masinuta creata este functionala ea putand sa se fereasca de obiecte in timp ce se indreapta catre o sursa de lumina.

BIBLIOGRAFIE

- 1. Rusu Anca** Circuite integrate liniare - catalog
- 2. Peter Suber** Microelectronica Data Book Componente optoelectronice
- 3. Iulian Ardelean** Circuite integrate CMOS
- 4. Tiberiu Muresan** Circuite integrate numerice - aplicatii
- 5. Miller Thomas** Automated guided vehicle system kempston,