

- Sa se realizeze un stand de laborator pentru studiul angrenajelor ordinare si diferentiale.
- Proiectul va fi compus din trei parti: partea mecanica, partea electronica si partea de control/programare.

### Cuprins:

- 1. Componenta mecanica. Introducere
- 1.2. Memoriu justificativ de calcul
- 1.3. Realizarea modelului 3D
- 2. Componenta electronica
- 2.1. Selectarea componentelor
- 2.2. Realizarea schemei electrice
- 3. Componenta de control
- 3.1. Implementarea algoritmului de control
- 3.2. Realizarea interfetei grafice
- 4. Concluzii

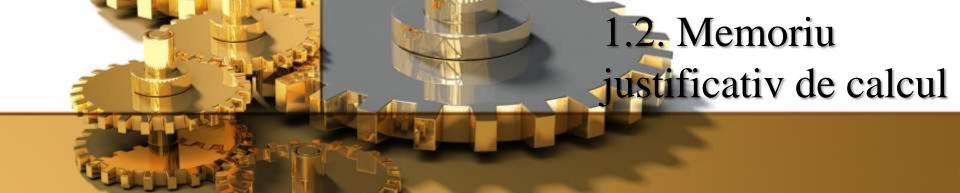




Angrenajul este mecanismul format din doua sau mai multe roti dintate care, prin intermediul dintilor aflati in contact transmite miscarea de rotatie si momentul de torsiune dintre arborii pe care sunt fixate.

Angrenajele au un domeniul larg de utilizare in transmisiile mecanice datorita avantajelor pe care le prezinta: raport de transmitere constant; siguranță în exploatare; durabilitate ridicată; randament ridicat; gabarit redus; posibilitatea utilizarii pentru un domeniu larg de puteri, viteze si rapoarte de transmitere.

Angrenajele diferentiale e sunt utilizate în principal în transmisia automobilului. Principala lor funcție este de a facilita directia si de a îmbunatati manevrabilitatea unui vehicul. El transmite si imparte puterea motorului si cuplul în două fluxuri între rotile de pe acelasi ax.

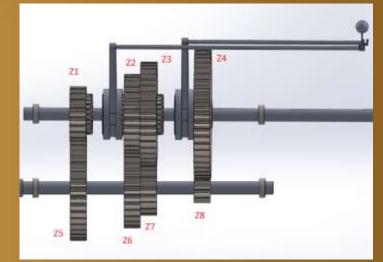


Pentru tema de proiect aleasa se studiaza o cutie de viteze cu 4 viteze si un angrenaj diferential. Rapoartele de transmisie se calculeaza in functie de numerele de dinti ale rotilor, dupa cum urmeaza:

$$\frac{Z1}{Z5} = \frac{20}{40} = 0.5 \qquad \frac{Z3}{Z7} = \frac{40}{20} = 2$$

$$\frac{Z2}{Z6} = \frac{30}{30} = 1$$
  $\frac{Z4}{Z8} = \frac{50}{10} = 1$ 

Toate rotile angrenajului au modulul 2. Distanta dintre axe este de 50 mm.





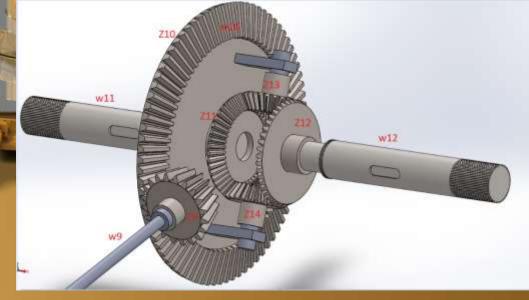
Raportul de transmitere al angrenajului diferential se calculeaza astfel:

$$\frac{\omega 11 - \omega s}{\omega 12 - \omega s} = \frac{Z12}{Z11} = -1$$

$$\omega s = \omega 10$$

$$i_{9 10} = \frac{\omega 9}{\omega 10} = \frac{Z9}{Z10}$$

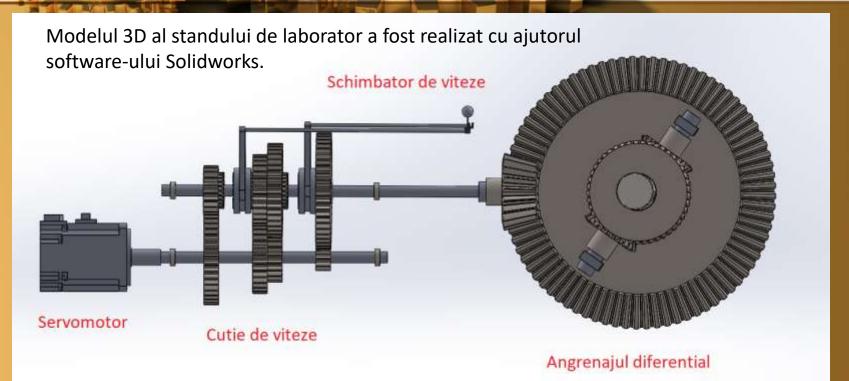
 $i_{9\,10} = \frac{\omega 9}{\omega 10} = \frac{Z9}{Z10} \qquad \Rightarrow i_{9\,10} = \frac{\omega 9}{\omega 10} \Rightarrow \omega s = \frac{\omega 9}{i_{9\,10}}$ 

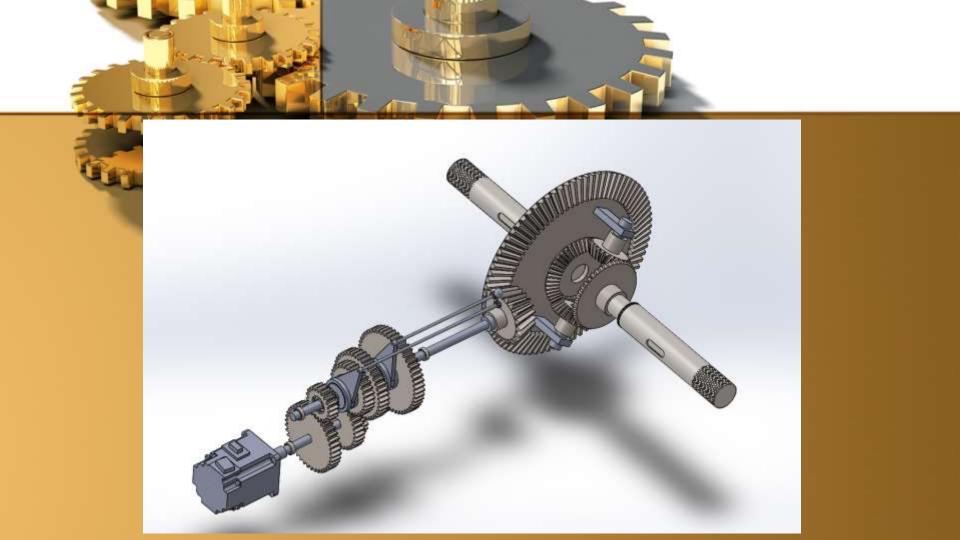


$$\frac{\omega 11 - \frac{\omega}{i_{9 10}}}{\omega 12 - \frac{\omega}{i_{9 10}}} = -1$$

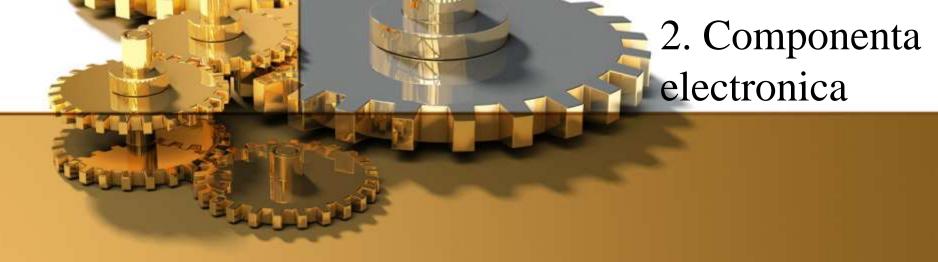
$$\implies \omega 11 + \omega 12 = 2 \omega 9$$

# 1.3. Realizarea modelului 3D



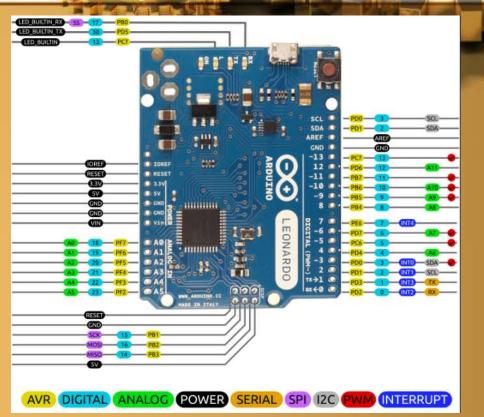






Componenta electronica reprezinta legatura electrica dintre componenta mecanica si componenta de control. Aceasta legatura se realizeaza cu ajutorul unui microcontroller programabil care va primi informatii de la interfata grafica si le va transmite elementelor care trebuie controlate, in cazul de fata, un motor DC 12V. De asemenea, acesta primeste informatii de la sistemul mecanic prin intermediul encoderelor si le transmite calculatorului pentru a fi afisate.

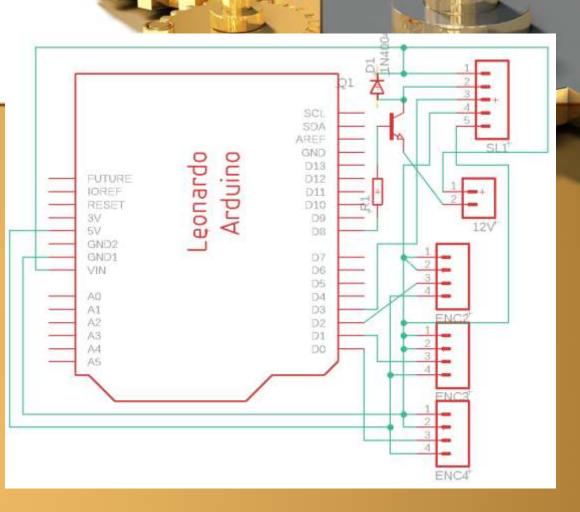
## 2.1. Selectarea componentelor



Am decis sa utilizam o placuta Arduino Leonardo bazata pe un ATmega 32u4 MCU. Datorita faptului ca vom utiliza 4 encodere, am ales o placuta cu suficienti pini de intrerupere externa.

Pentru comanda motorului am conectat pinul 8 la o rezinstenta, apoi la baza unui transistor npn. Pentru siguranta, am adaugat o dioda in paralel cu motorul. Alimentarea atat a motorului, cat si a placutei se realizeaza de la o sursa de 12V.

Informatia de la encodere va fi preluata de pe pinii 0,1,2 si 3, pini de intrerupere externa.

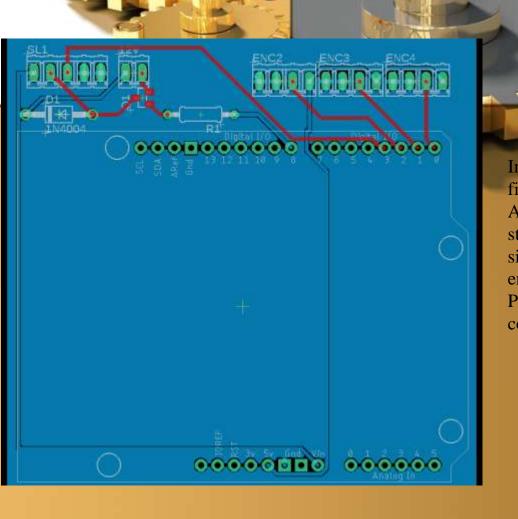


### 2.2. Realizarea schemei electrice

Am realizat schema electrica si proiectarea placutei cu ajutorul Autodesk Eagle.
Pentru simplificarea circuitului, am realizat un shield la care se vor conecta prin fire componentele care nu se pot afla in proximitatea placutei.

Am folosit 5 conectori pentru motor+encoder,

sursa, respective celelalte 3 encodere.



In continuare, am proiectat circuitele de cupru fizice care conecteaza elementele.

Am avut nevoie de 2 straturi :Top si Bottom. Pe stratul Top am realizat conexiunile tranzistorului si ale pinilor de intrerupere externa cu encoderele.

Pe stratul Bottom am realizat pe langa celelalte conexiuni, un ground layer.



Componenta de control a fost realizata cu ajutor programului Matlab Guide. Am decis sa folosesc acest program deoarece este usor de utilizat si ofera o mare varietate de optiuni.

Matlab Guide este un program ce genereaza automat un fisier de program care contine functii Matlab, acesta controland modul in care se comporta o interfata de utilizare. Acest fisier de cod ofera codul pentru a initializa interfata de utilizare si contine un cadru pentru callback-urile interfetei de utilizare.

Callback-urile reprezinta functii care se executa atunci cand utilizatorul interactioneaza cu o componenta a interfetei de utilizare.

Folosind GUIDE Layout Editor, se poate completa o interfata de utilizare facand clic si glisand componente UI cum ar fi axe, panouri, butoane, campuri de text si asa mai departe - în zona de aspect. De asemenea, se pot crea meniuri contextuale pentru interfata de utilizare. Din Editorul de aspect, se poate dimensiona interfata de utilizare, modifica aspectul componentelor, alinia componentele, setati ordinea filelor, vizualiza o lista ierarhica a obiectelor componente si setati optiunile interfetei.

### function pushbuttonSTART Callback(hObject, eventdate, handles) handle to pushbuttonSTART (see GCBO) reserved - to be defined in a future version of MATLAB structure with handles and user data (see SUIDATA) % global at % a.analogWrite(0,); turatie = strldouble(get(handles.editTuratiaMotorului, 'String')); vitesa = str2double(get(handles.editVitesa, 'String')); if turatie > 530 turatie=530; outputString = sprintf("%.2f", turatie); set (handles.editTuratiaMotorului , 'String', outputString); elseif turatie <0 turatie=0; outputString = sprintf('%.3f', turatie); set Thandles.editTuratiaMotorului , 'String', outputString); if witers >= 4 vitez=5: sutputString = sprintf('%', 4); set (handles.editViteza , 'String', outputString); elseif viterass3 viteza=2; outputString = sprintf('%d', 3); set (handles.editViteza , "String", outputString); elseif viterawe2 vitera\*1 outputString = sprintf('%d', 2)/ set (handles.editViteza , "String", outputString) / elseif viteza <=1 viteza=1/2; outputString = sprintf('%d', 1); set (handles.editViteza , "String", outputString); output-viteza\*turatie; outputString = eprintf('\nk.If', output); set (handles.textRexCV , 'String', outputString); set (handles.textRexDD , 'String', outputString); set (handles.textRexDS , 'String', outputString);

## 3.1. Implementarea algoritmului de control

Algoritmul de control scris a fost structurat si creat dupa specificatiile motorului utilizat, a cutiei de viteze si a diferentialului.

Motorul utilizat este unul de 12V avand performanta de 530rpm.

Cutia de viteze prezinta 4 viteze, iar rapoartele de transmisie au rezultatele astfel:

- Viteza 1 = 0.5
- Viteza 2 = 1
- Viteza 3 = 2
- Viteza 4 = 5

Algoritmul tine cont de viteza selectata, performanta maxima a motorului de 530 rpm.



	4		5
125 - set (handles, textRerCV , 'String	, outputString);	161	
126 - set (handles.textRexDD , 'String		162	function editViteza Callback(hObject, eventdata, handles)
127 - set (handles.textRerDS , 'String	, outputString);	250000	h hObject handle to editViteza (see GCBO)
THE RESERVE THE PROPERTY OF TH	EAUCED SERVICE	9000	
129		164	4 eventdata reserved - to be defined in a future version of MATLAS
135 A Executes on button press		2 ( 2 ( 2 ( ) ) )   1 ( )   1	A handles structure with handles and user data (see SUIDATA)
131 function pushbuttonSTOF_Callback		166	
182 D& hObject handle to pushbuots		1€7	% Hints: get(hObject, 'String') returns contents of editVitesa as text
	efined in a future version of MATLAB	191	a str2double(get(hObject,'String')) returns contents of editViteza as a double
134 % handles structure with han 132 % girbal at	iles and user data (see GGIIATA)	169	
136 % sanalogWrite(8,0);		170	
117		171	4 Executes during object greation, after setting all properties.
138		172	function editViteza CreateFon(hObject, -, -)
	allback(hObject, eventdate, handles)	20000	% hObject handle to editViteza (see GCBC)
148   A hObject   bandle to editTure		174	% eventdata reserved - to be defined in a future version of MATLAS
141 % eventdate reserved - to be do	sfined in a funure version of NATLAN	175	
142 - & handles structure with hand	dles and user data (see GUIDATA)	1000	N handles empty - handles not created until after all CreateFons called
143	AND AND ASSESSMENT OF THE PROPERTY OF THE PROP	17€	
	returns contents of editTurstiaNotorului as test	177	% Hint: edit controls usually have a white background on Windows.
	'String')) returns contents of editTurstiaMotorului as a double	178	See ISFC and COMPUTER.
106	CONSISTENCE OF THE CONTROL OF THE CO	179 -	if impc 44 imequal(get(hObject, "BackgroundColor"), get(0, "defaultUicontrolBackgroundColor"))
347		180 -	set(hObject, 'BackgroundCulor', 'white');
	estion, after setting all properties.	181 -	end
	reateFon(hObject, eventdata, handles)	182	and the second s
	stined in a future version of MATLAS	183	
	corested until after all CresteFuns called	184	* Executes on key press with focus on pushbuttonSTART and none of its controls.
353 4 nonutes sakes - namotes no	A ANDREAS SHARE STATE AND ADDRESS COLLEGE	185	function pushbuttonSTART KeyPressFom(-, -, -)
	ave a white background on Windows.	19€	remotive permutations and new restriction (-, -, -)
155 % See ISPC and COMPUTER.			
	BackgroundColor'), get(0, 'defaultUncomrolBackgroundColor'))	187	
157 - set (hObject, 'BankgraundColor		188	
156 end		189	
159		190	A hObject handle to gushbuttonSTART (see GCBO)
160		191	% eventdata structure with the following fields (see MATLAS,UI,CONTROL,UICONTROL)
161		192	8 Key: name of the key that was pressed, in lower case
362 function editViteza_Callback(hO		193	b Character: character interpretation of the key(s) that was pressed
161 De nObject handle to editVite		194	& Modifier: name(s) of the modifier key(s) (i.e., obotrol, shift) pressed
	efited in a future version of NATLAD	10000	A handles structure with handles and user data (see GUIDATA)
165 - a nandles structure with han	TITES SITE PACT THEM [SEE BRIDGES]	777	- The state of the

### 3.2. Realizarea interfetei grafice



Interfata este alcatuita din doua butoane, unul pentru pornirea motorului (START) si unul pentru oprirea lui (STOP), si dintr-o serie de casete de introducere a datelor (cea de introducere a turatie si cea de schimbare a vitezei), si o serie de casete de afisare atat a indicatiilor cat si a rezultatelor obtinute.

\* In poza din stanga sunt afisate rezultatele obtinute a turatiei motorului la 530 rpm in viteza 1.

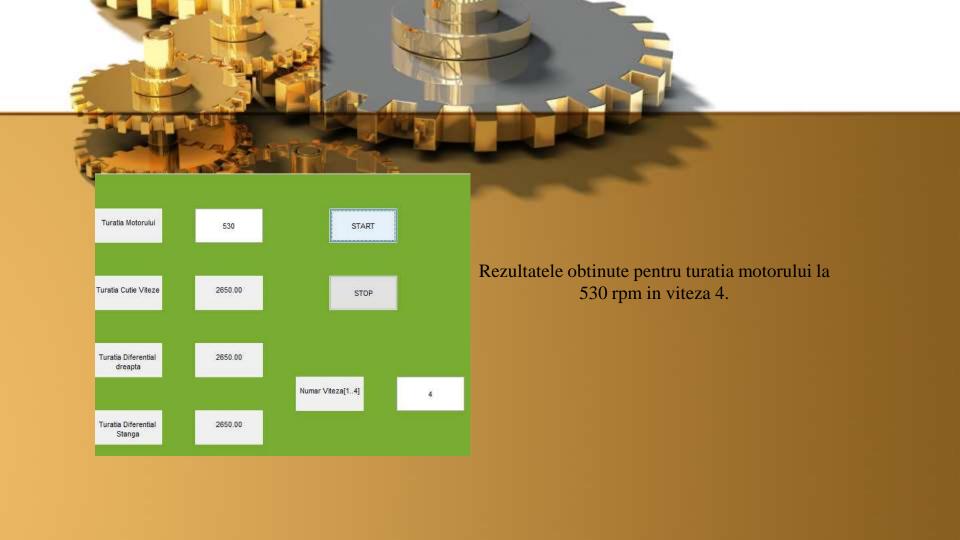
### 3.2. Realizarea interfetei grafice

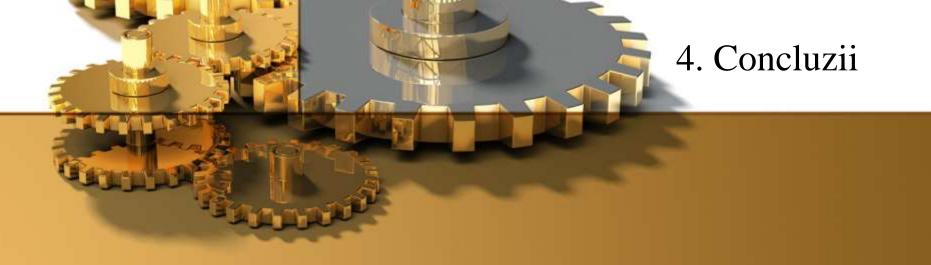


Rezultatele obtinute pentru turatia motorului la 530 rpm in viteza 2.



Rezultatele obtinute pentru turatia motorului la 530 rpm in viteza 3.





Tema propusa prezinta aplicatii practice pentru studiul atat a angrenajelor ordinare si diferentiale, cat si pentru studiul motoarelor de curent continuu si al encoderelor.

De asemenea, am observat importanta utilizarii unor programe specializate in realizarea fiecareia dintre componentele cerintei.