



TÜBİTAK EFFICIENCY CHALLENGE

ELECTRIC VEHICLE

TECHNICAL DESIGN REPORT

Deadline: July 20th, 2017

VEHICLE AND TEAM NAME:GOKSIGIN/ ETÜ GO

UNIVERSITY:TOBB University of Economics and Technology

ACADEMIC ADVISOR: Assist. Prof. Dr. Aydın ÖZTOPRAK

TEAM CAPTAIN:Tarık Can TÜRKOĞLU

**VEHICLE
TYPE:**

ELECTROMOBILE HYDROMOBILE

CONTENTS

VEHICLE AND TEAM NAME:	1
UNIVERSITY:	1
ACADEMIC ADVISOR:	1
TEAM CAPTAIN:	1
1. Vehicle specifications table	4
2. Domestic Sub-Components	5
3. Motor	6
a. Design calculations.....	6
b. Magnetic analysis studies	6
c. Mechanical analysis studies	6
d. Thermal analysis studies	6
e. Production studies	6
f. Motor efficiency.....	6
4. Motor driver	7
a. Circuit design.....	7
b. Control algorithm.....	7
c. Simulation studies.....	7
d. Printed circuit studies.....	7
e. Production studies	7
f. Motor driver efficiency	7
5. Battery management system (BMS).....	8
a. Circuit design.....	8
b. Balancing method.....	8
c. Control algorithm	8
d. Simulation studies.....	8
e. Printed circuit studies	8
f. Production studies	8

6.	Embedded recharging unit	9
a.	Circuit design	9
b.	Simulation studies.....	9
c.	Printed circuit studies	9
d.	Production studies	9
7.	Energy management system (EMS).....	10
a.	Control algorithm.....	10
b.	Simulation studies.....	10
c.	Application studies	10
8.	Battery Packaging	11
9.	Telemetry.....	12
a.	Circuit design	12
b.	Communication protocols	12
c.	Printed circuit studies	12
d.	Production studies	12
11.	Electronic differential application.....	13
12.	Vehicle Control Unit (VCU)	14
13.	Mechanical details	15
a.	Technical drawings	15
b.	Strength analysis.....	16
c.	Outer shell production.....	16
d.	Energy consumption and maximum inclination account	16
e.	Cost calculation.....	16
14.	Fuel cell	Error! Bookmark not defined.
15.	Fuel cell control system	Error! Bookmark not defined.
16.	Vehicle electric scheme(<i>mandatory</i>)	19
17.	Unique design by team (<i>optional</i>)	194

Table 1: Table of summary score.....	195
--------------------------------------	-----

1. Vehicle specifications table

University name	TOBB University of Economics and Technology	
Team name	ETÜ GO	
Team captain	Tarıkcan TÜRKOĞLU	
Feature	Unit	Value
Length	mm	1500
Width	mm	3000
Height	mm	1400
Number of wheels	#	4
Chassis	material	Aluminium and steel
Shell	material	fiber
The brake system	hydraulic disc, front, rear, hand brake	hydraulic disc, front, rear, hand brake
Diameter of front tyres	mm	580
Width of front tyres	mm	110
Diameter of rear tyres	mm	580
Width of rear tyres	mm	110
Telemetry	yes/no; transmitted information (speed, battery status, temperature, etc.)	Yes speed, battery status, temperature,current
Motor	type	Brushless of hub
Motor driver	yes/no; self-designed, ready-made product	Yes self-designed
Motor power	kW	10
Motor efficiency	%	90
Engine weight	kg	20
Battery	type	LiFePO4
Battery nominal voltage	V	115.2
Maximum voltage of the battery	V	131.4
Battery nominal capacity	Wh	3456
Fuel cell power	kW	-
Number of hydrogen tubes	#	-
Hydrogen tube pressure	bar	-
Super capacitor	yes/no	No
You must fill in the fields related to your category.		

2. Domestic Sub-Components

In this section, teams are asked to check off the domestic sub-components they are planning to design themselves. Note that, teams are required to design all four mandatory sub-components in their category.

1. Motor	Mandatory for Electromobile/Hydromobile	X
2. Motor driver	Mandatory for Electromobile/Hydromobile	X
3. Battery management system (BMS)	Mandatory for Electromobile/Hydromobile	X
4. Embedded recharging unit	Mandatory for Electromobile	X
5. Energy management system (EMS)*	Mandatory for Hydromobile	<input type="checkbox"/>
6. Battery packaging	Optional	X
7. Telemetry	Optional	X
8. Electronic differential application	Optional	X
9. Vehicle control unit (VCU)	Optional	X
10. Fuel cell*	Optional	<input type="checkbox"/>
11. Fuel cell control system (circuit)*	Optional	<input type="checkbox"/>

* *Hydromobile category only*

3. Motor(if designed by team, details mandatory; if ready-made product, please explain briefly)

*All the teams are required to prepare their motors themselves. The teams that make this product domestically will get a maximum of **400 points** depending on the design.*

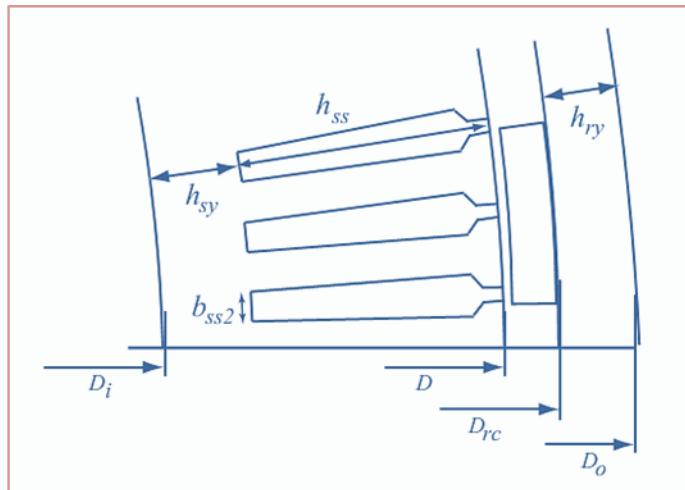
The teams that perform domestic motor design and production are responsible for the following items during the technical inspections:

- 1) Awareness regarding electrical and mechanical features.
- 2) Submitting information on magnetic and thermal analysis.
- 3) Submitting visual information such as photos and videos about production stages.
- 4) Submitting information related to test methods and results (You might be requested to simulate raw documents)

If designed by the team, detailed information should be given on the following topics. Additionally, the raw documents (original files) to show the simulation/design process and calculation results must be sent with the report in a compressed file (i.e..zip, .rar). Otherwise it will not be considered as domestic even if a domestic product motor is presented at technical inspections. All the designs and simulations must be the teams' own work, i.e., magnetic and/or thermal analyses provided by the manufacturer of any part of the system will not be accepted. The teams might be asked to run their source files related to the magnetic and thermal analyses and to show the sketches of their design.

a. Design calculations

HUB Motor Tasarım Denklemleri



Şekil 1. Dış rotorluBLDC motorları için bazı geometrik parametrelerin tanımlanması [1]

Boyutlandırma çalışmalarında dikkat edilmesi gereken en önemli parametreler çıkış eşitliklerinin ve performans denklemlerinin gerçekleştirilmesinde sıkılıkla karşılaşılan stator ve rotor çap boyutları ile motor uzunluğudur. Bunun dışında kalan boyutlar sargı çalışmaları kapsamında performans göz önünde bulundurularak parametrik çözümler ışığında elde edilebilmektedir.

Dış rotorlu BLDC motorlarının stator dış çapı aşağıdaki eşitlik yardımıyla hesaplanabilir;

$$D = D_{rc} - 2l_m - 2\delta$$

Eşitlikte verilen D stator dış çapını, D_{rc} rotor iç çapını, l_m mıknatıs kalınlığını ve δ hava aralığının uzunluğunu vermektedir. Motor tasarımında oluk tasarımı sargı konfigürasyonu ve yapısına bağlı olarak çok önemlidir. Şekil 1'de gösterilen b_{ss1} statorluğunun üst oluk genişlik parametresi olup aşağıdaki eşitlik yardımıyla hesaplanır.

$$b_{ss1} = \pi \frac{D - 2h_{sw}}{Q_s} - b_{ts}$$

h_{sw} stator nüvesinin dış kalınlığını, b_{ts} nüvenin dış genişliğini ve Q_s stator oluk sayısını belirtmektedir. b_{ss2} ise statorluğunun alt oluk genişlik parametresi olup aşağıdaki eşitlik yardımıyla hesaplanır.

$$b_{ss2} = \pi \frac{D - 2h_{ss}}{Q_s} - b_{ts}$$

Eşitlikte ifade edilen h_{ss} oluk boyunu ifade etmektedir. h_{sy} etkin nüve kalınlığını ifade etmektedir ve aşağıdaki eşitlik yardımıyla hesaplanabilir.

$$h_{sy} = \frac{1}{2}(D - D_i - 2h_{ss})$$

Statorun iç çapını belirten D_i sadece dış rotorlu BLDC motorlarına ait bir parametredir. Benzer şekilde rotorda manyetik akının akışını sağlayacak olan ve ST-37 çeliğinden seçilecek olan nüvenin kalınlığı aşağıdaki formülizasyon ile elde edilebilir[4].

$$h_{ry} = \frac{1}{2}(D_0 - D_{rc})$$

Toplam oluk alanı olarak ifade edilen A_{sl} ise;

$$A_{sl} = \frac{1}{2}(b_{ss1} + b_{ss2}) * (h_{ss} - h_{sw})$$

eşitliği ile gösterilebilir. Bu arada stator oluk açıklığının oluk genişliğine oranı da göz önünde bulundurulması gereken bir parametredir ve aşağıdaki eşitlik ile ifade edilmektedir.

$$k_{open} = \frac{b_{so}}{b_{ss1}}$$

Eşitlikte verilen b_{so} stator olukları arasındaki mesafeyi göstermektedir.

4.4.1. Kutup sayısının seçimi

Sürekli mıknatıslı makinalarda kutup sayısının seçimi tasarımda belirleyici faktör olan rotor hızına göre şekillenmektedir. Aynı hız değeri için kutup sayısındaki artış; komütasyon frekansında artıa dolayısıyle anahtarlama elemanlarında frekansa bağlı oluşan anahtarlama kayıplarında artıa ve stator nüvesinde meydana gelen ve bir

önceki bölümde açıklanmış olan demir kayıplarında artışa neden olacaktır [2]. Kutup sayısı temel olarak aşağıdaki formül ile ifade edilmektedir.

$$2p = \frac{120 \cdot f}{n}$$

Eşitlikte gösterilen 2p kutup sayısını, f çalışma frekansını ve n rotor devir sayısını vermektedir. Bu değerin seçimi daha kolaylaştırmak için sargı faktörü göz önünde bulundurularak kutup-oluk sayısı kombinasyonlarına ve bu konu ile ilgili daha önce gerçekleştirilmiş olan çalışmalara göz atmak çok faydalı olacaktır. Konu daha sonraki bölümlerde detaylı olarak açıklanacaktır.

4.4.2. Hava aralığı akısının hesaplanması

Tasarım işlemlerinin düzgün bir şekilde ilerleyebilmesi için hava aralığı akısının analitik olarak düzgün bir şekilde hesaplanması gerekmektedir. Hava aralığının hesaplanması farklı tasarım şemaları için farklı niteliktedir. Gerçekleştirilecek uygulama yüzey mıknatıslı bir tasarıma dikkat çektiği için literatürde bu konuda gerçekleştirilmiş olan çalışmalar göz önünde bulundurularak hesaplamalar gerçekleştirilecektir.

Manyetik akının maksimum değeri olan B_m şu şekilde hesaplanmaktadır;

$$B_m = \frac{B_r k_{leak}}{1 + \frac{\mu_r \delta k_c}{l_m}}$$

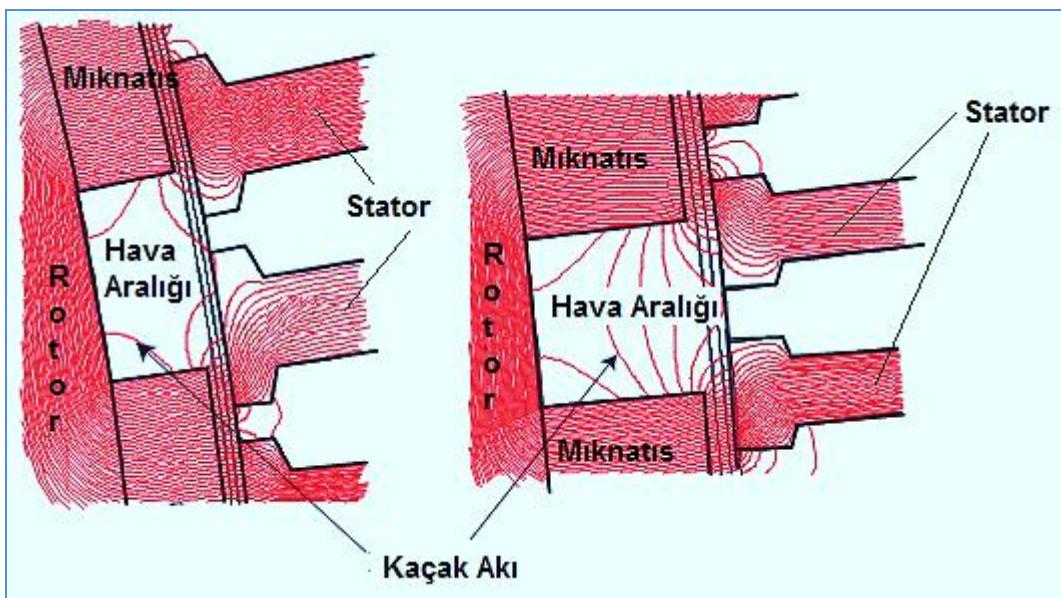
B_r mıknatısın kalıcı akı yoğunluğunu, μ_r relatif mıknatıs geçirgenliğini ve k_c Carter faktörünü belirtmektedir ve aşağıdaki eşitlikle hesaplanır [3].

$$k_c = \frac{T_s}{T_s - \frac{(k_{open} b_{ss1})^2}{b_{ss1} k_{open} + 5\delta}}$$

Eşitlikte ifade edilen ve iç rotorlu ile dış rotorlu makinelerde ortak olan statot dış açıklığı (δ_r) aşağıdaki eşitlikle ifade edilmektedir.

$$T_s = \pi \frac{D}{Q_s}$$

Manyetik akı eşitliğinde ifade edilen k_{leak} iki komşu sabit mıknatıs arasındaki manyetik kaçak akımı ifade eden bir büyüklüktür.



Şekil 2 Farklıkutup sayısına sahip makinelerde kaçak akının durumu (2p=40 ve 2p=70)

Bu ifadenin farklı makine tipleri arasında nasıl değiştiğini kavrayabilmek için Şekil 2'ye bakmak yeterli olacaktır. Kutup sayısı arttıkça sürekli mıknatıslar arasındaki mesafe doğal olarak azalacak ve buna bağlı olarak hava aralığı relüktansında azalma gözlenecektir. Sürekli mıknatıslar arasındaki kaçak akının değeri tabii ki mıknatısın kalınlığına ve hava aralığının uzunluğuna da bağlıdır. Fakat bu durum şartlar aynı olduğu için ihmal edilmiştir. Lineer olarak kutup sayısındaki artışın kaçak akı üzerindeki değişimi Şekil 2'de açıkça görülmektedir. Kaçak akı faktörü hava

aralığından geçen manyetik akı çizgilerinin yoğunluğuna bağlı olarak tanımlanmıştır.
Buna göre;

$$k_{leak} = \frac{100 - \text{number of leakage flux lines}}{100}$$

eşitliği ile tanımlanmaktadır. Ayrıca kaçak akı faktörü sabit mıknatıslı senkron makineler için;

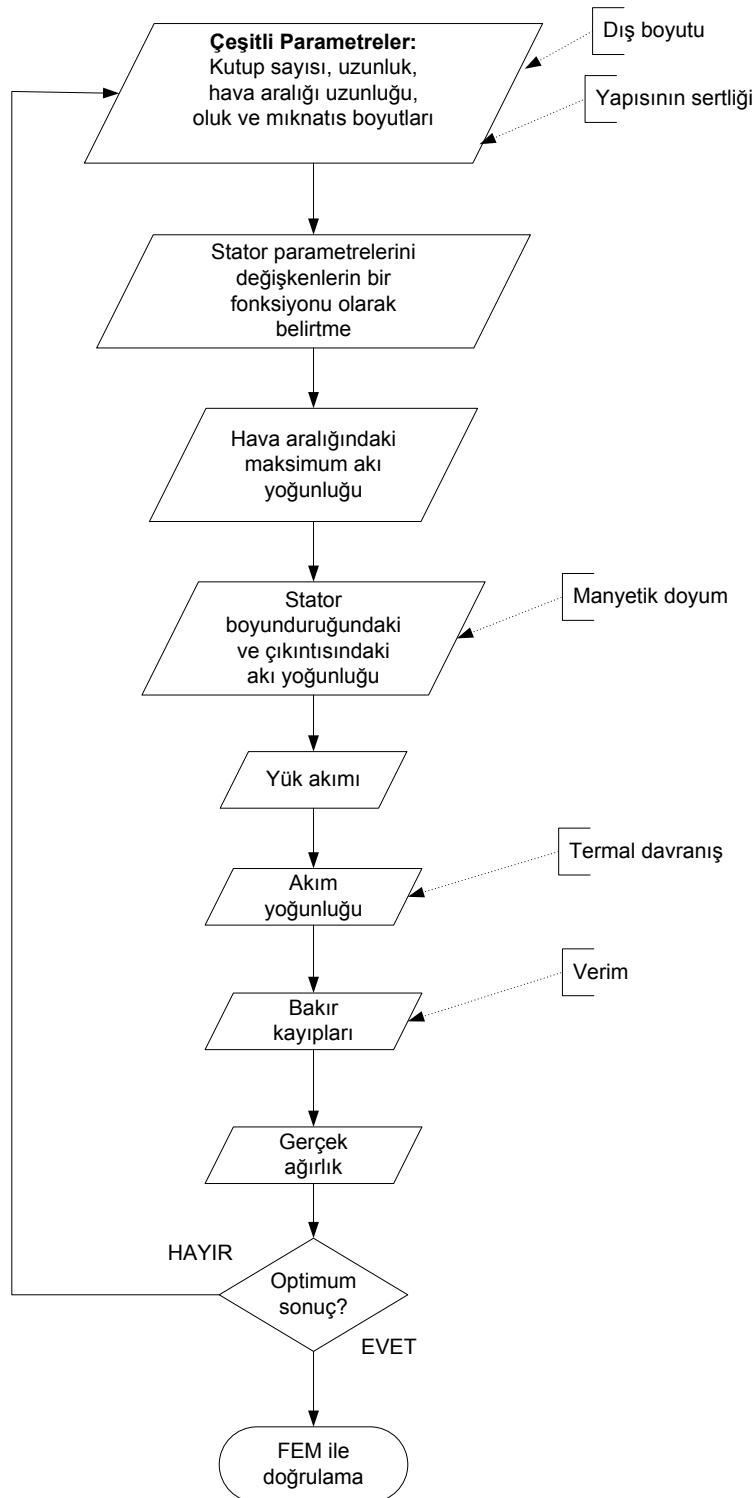
$$k_{leak} = \frac{100 - (7p / 60 - 0.5)}{100}$$

dış rotorlu sabit mıknatıslı doğru akım makineleri için;

$$k_{leak} = \frac{100 - (7p / 60 - 3)}{100}$$

b. Magnetic analysis studies

MANYETİK ANALİZ SONUÇLARI



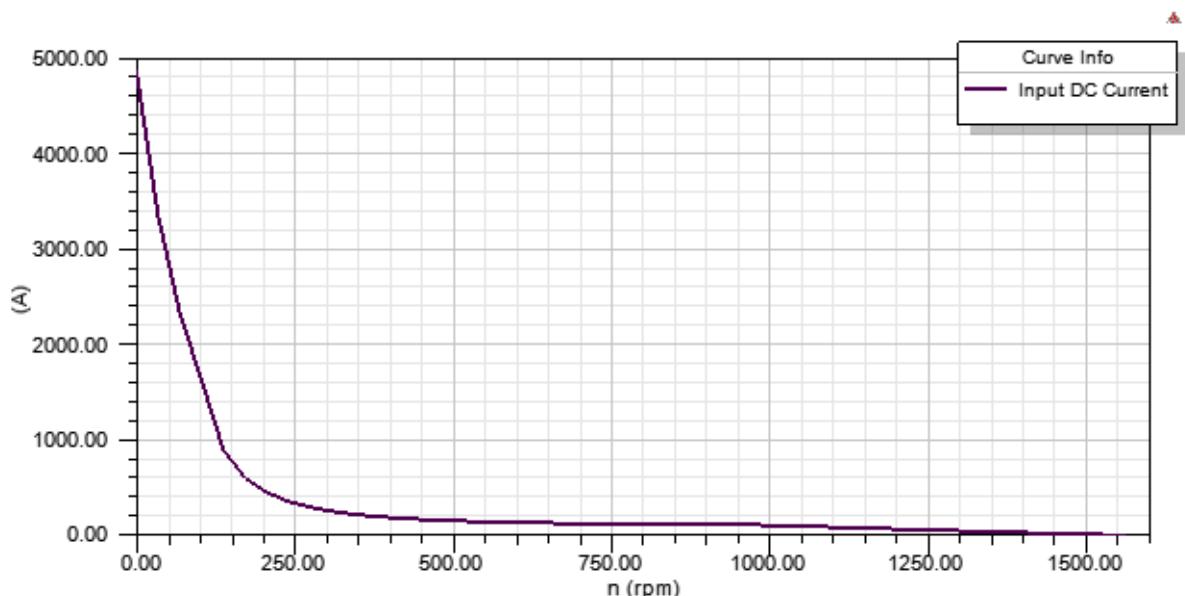
Şekil 4.8. Analitik hesaplama yönteminde kullanılan akış şeması [4]

Akış şemasında da görüldüğü üzere BLDC motorlarının tasarım süreci çok uzun ve karmaşık denklemlerden meydana gelmektedir. 1200 d/d başlangıç devir değeri için hesaplanan sabit mıknatıslı dış rotorlu motorun temel boyutlandırma parametreleri aşağıdaki tabloda gösterildiği gibi hesaplanmıştır.

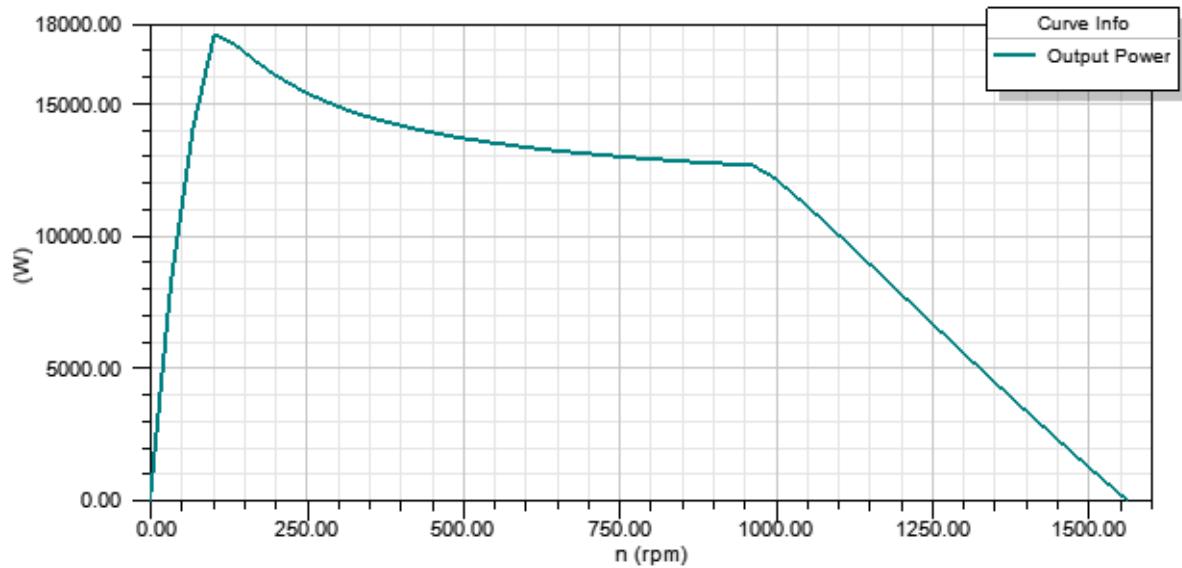
4.1. Temel boyutlandırma parametreleri

TASARIM PARAMETRESİ	DEĞERİ
Stator dış çapı (D_o)	246 mm
Stator iç çapı (D_i)	180mm
Rotor dış çapı	270 mm
Rotor iç çapı	249 mm
Mıknatıs kalınlığı	3 mm
Motor uzunluğu (L)	40 mm
Motor anma hızı	1200 d/d
Besleme gerilimi	120 V DC
Stator oluk sayısı	51
Kutup sayısı	46
Faz sayısı (m)	3

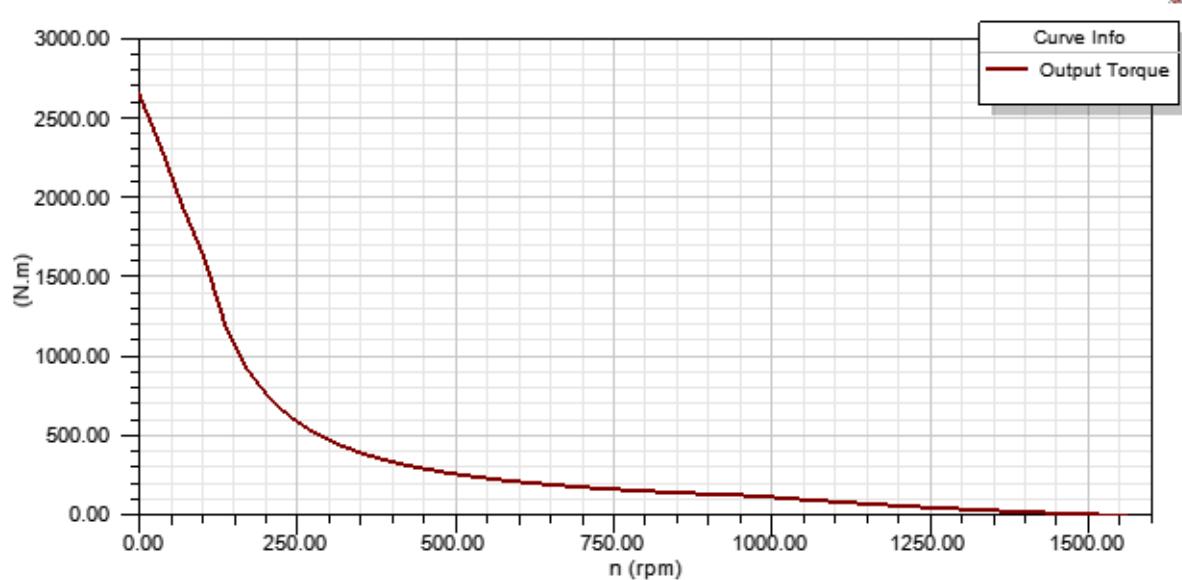
Tablo 1. Temel boyutlandırma parametreleri



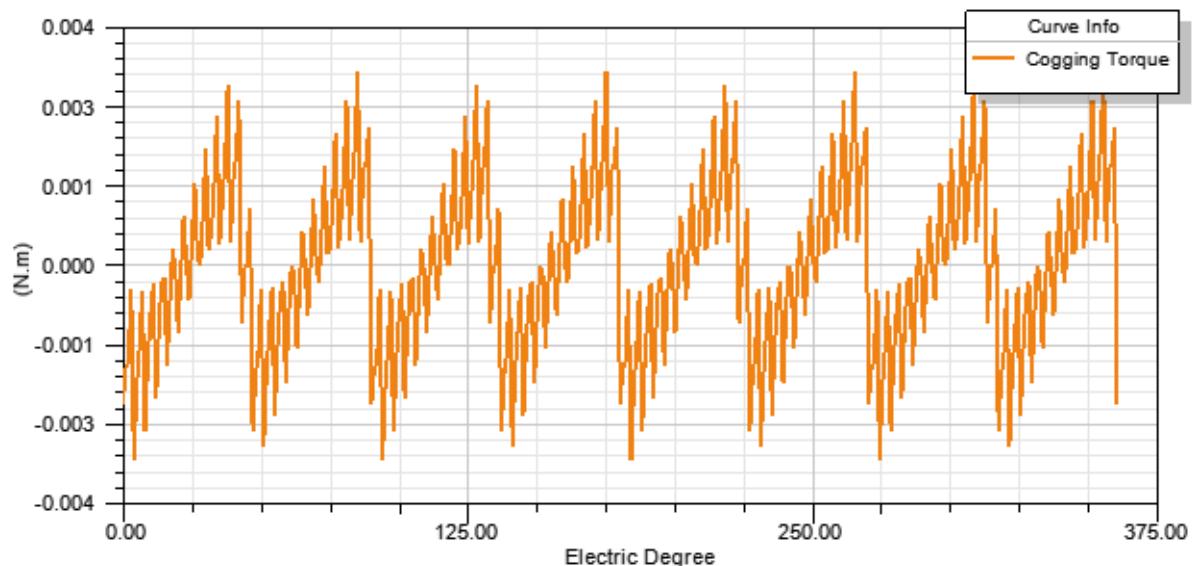
Şekil 3 Motorun Hızına Karşılık Akımdaki Değişim



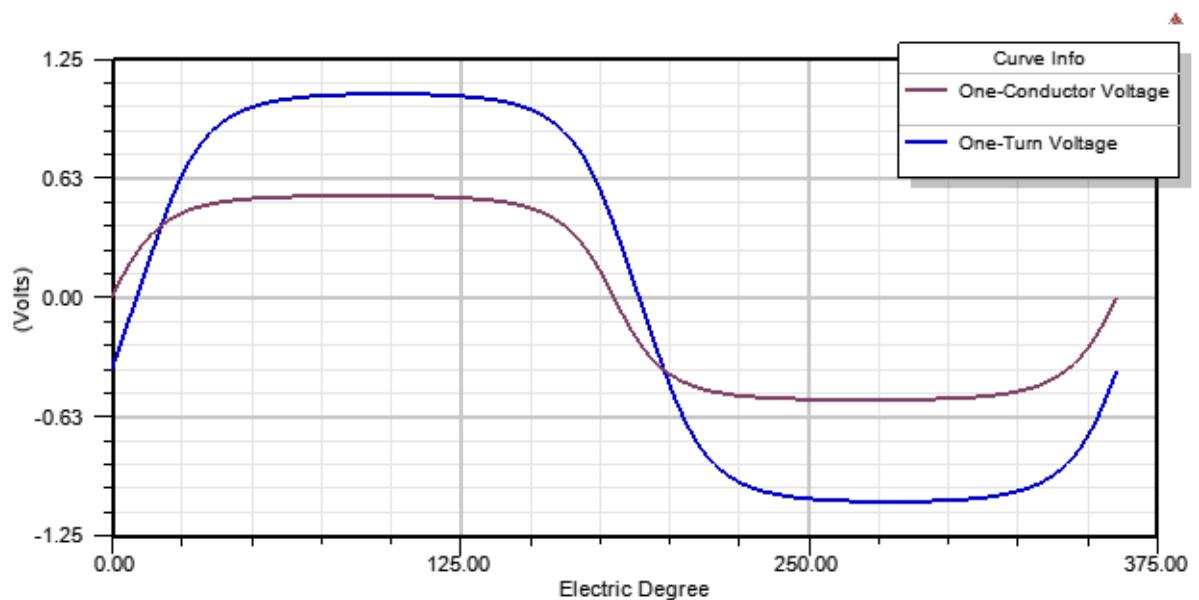
Şekil 4 Motor hızına karşılık motor gücündeki değişim



Şekil 5 Motor hızına karşılık motor milinde oluşan moment

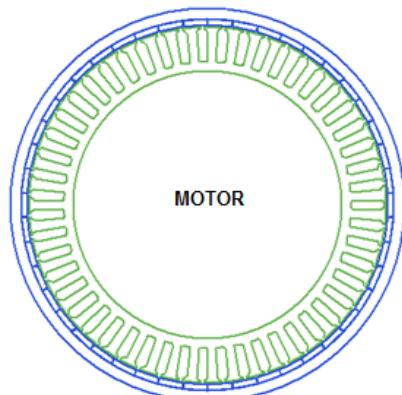
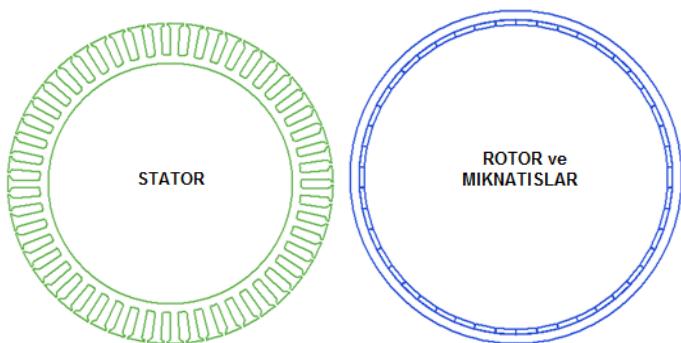


Şekil 6 Elektriksel açıya bağlı olarak mıknatılardan dolayı oluşan tutma momenti



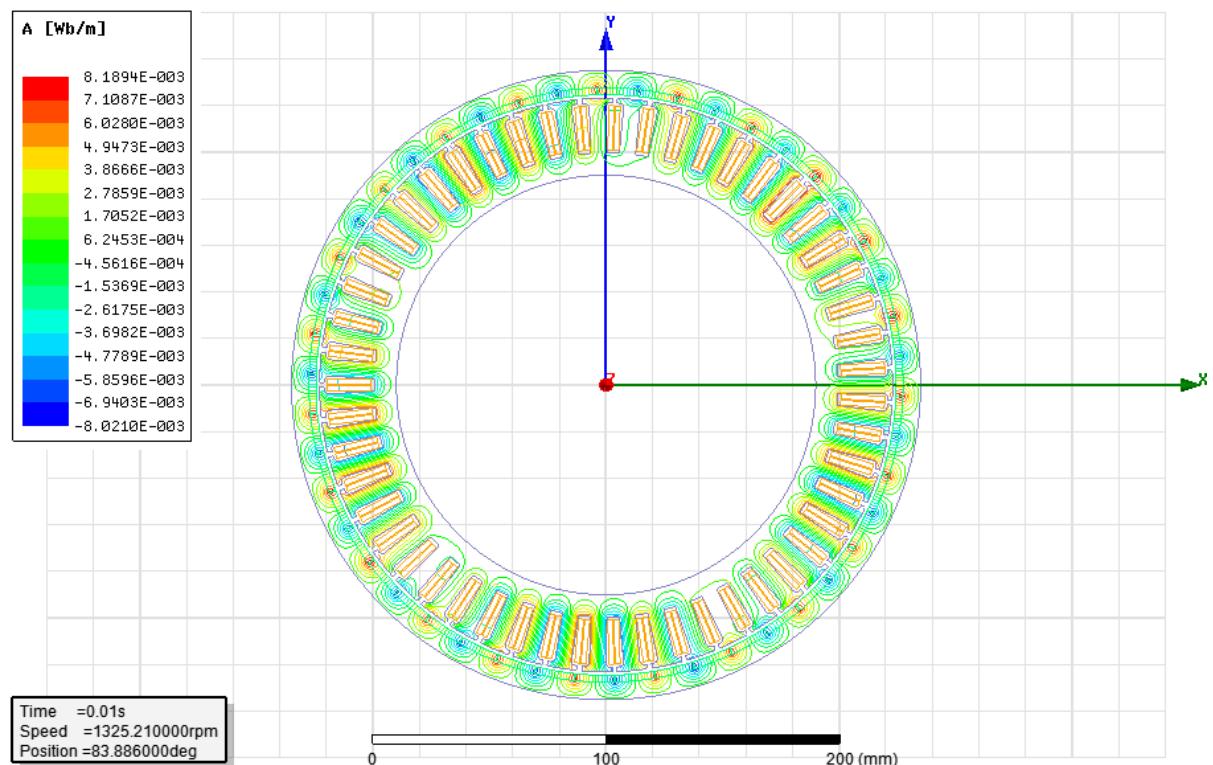
Şekil 7 Elektriksel açıya bağlı olarak motor anma hızındayken sargılarda indüklenen gerilimin değişimi

Parametre	Değer	Motor Parçaları
Anma Çıkış Gücü	5000 Watt	
Anma Gerilim	120 Volt	
Kutup Sayısı	46	
Stator Oluk Sayısı	51	
Stator İç Çapı	180 mm	
Stator Paket Uzunluğu	40 mm	
Stator Sacı	M270-35	
Oluk Başına Sarım Sayısı	6	
Hava Aralığı	1.5 mm	
Hava Aralığı	1.5 mm	
Rotor Çeliği	ST-37	
Mıknatıs Kalınlığı	3mm	
Mıknatıs Malzemesi	NdFeB-N40SH (150°C)	
Toplam Ağırlık	8.25 Kg	
Hava Aralığı Akı Yoğunluğu	0.729 T	
Vuruntu Torku	0.0021 Nm	
Ortalama Giriş Akımı	45 A	
Spesifik Elektrik Yüklemesi	15.849 A/mm	
Stator Sargısı Akım Yoğunluğu	4.34 A/mm ²	
Demir Kaybı	352.971 W	
Bakır Kaybı	59.09 W	
Toplam Kayıplar	412.062 W	
Verim %	92.38 – (Sürtünme ve rüzgâr)	

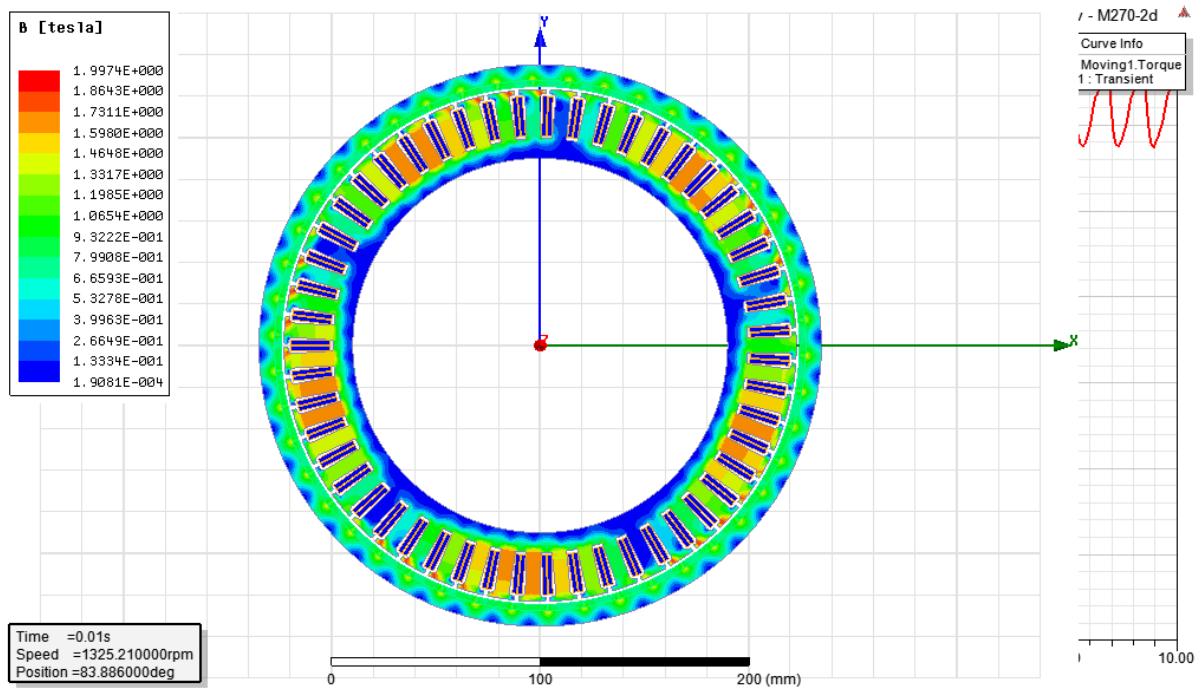


	kayıpları hariç)
Anma Hızı	1325 d/d
Anma Momenti	36.022Nm
Sargı Düzenlemesi	3 Faz, 51 oluk tam kalıp sarım

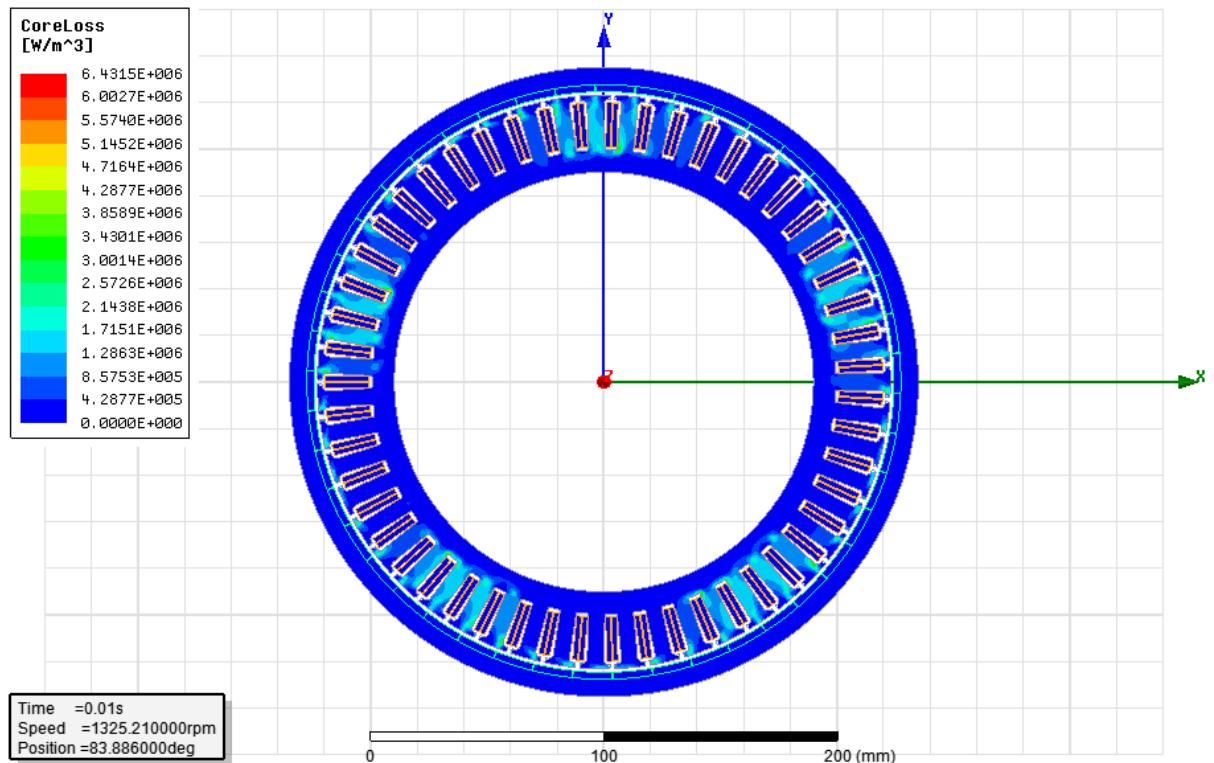
Analitik analizler sonucunda tasarıımı doğrulanan dış rotorlu fırçasız sürekli mıknatıslı doğru akım motorunun sonlu eleman analizleri gerçekleştirılmıştır.



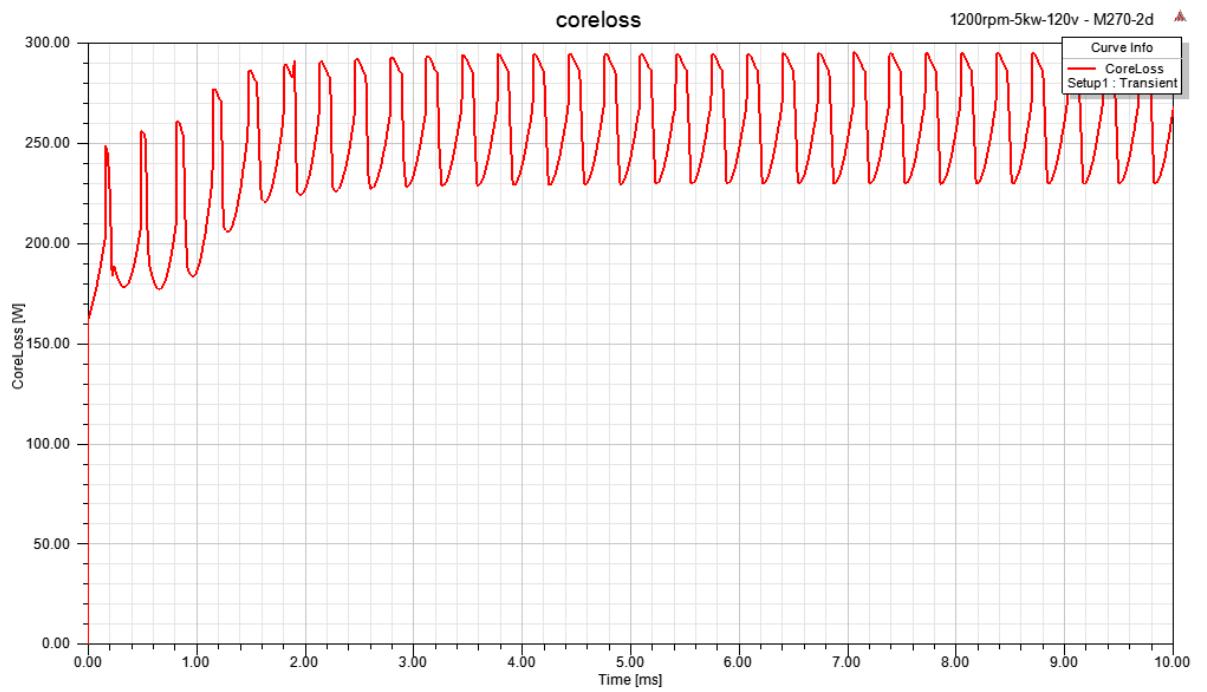
Şekil 80.01 Saniyedeki manyetik akı dağılımı



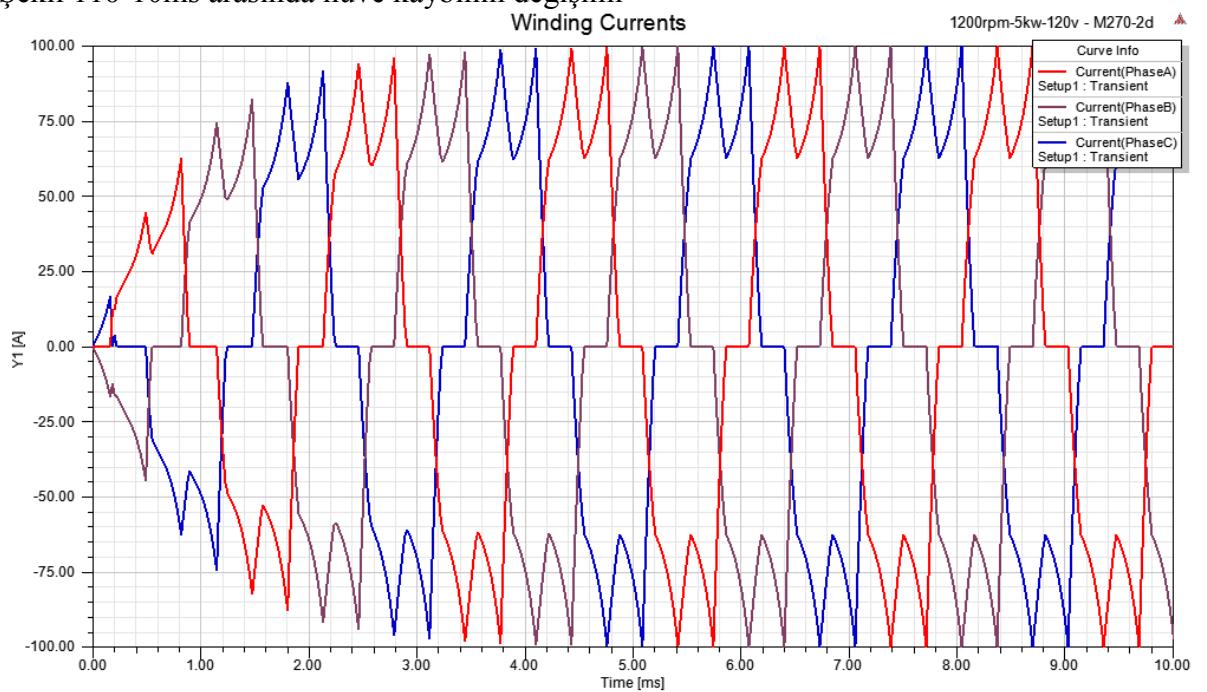
Şekil 90.01 Saniyedeki manyetik akı yoğunluğu dağılımı



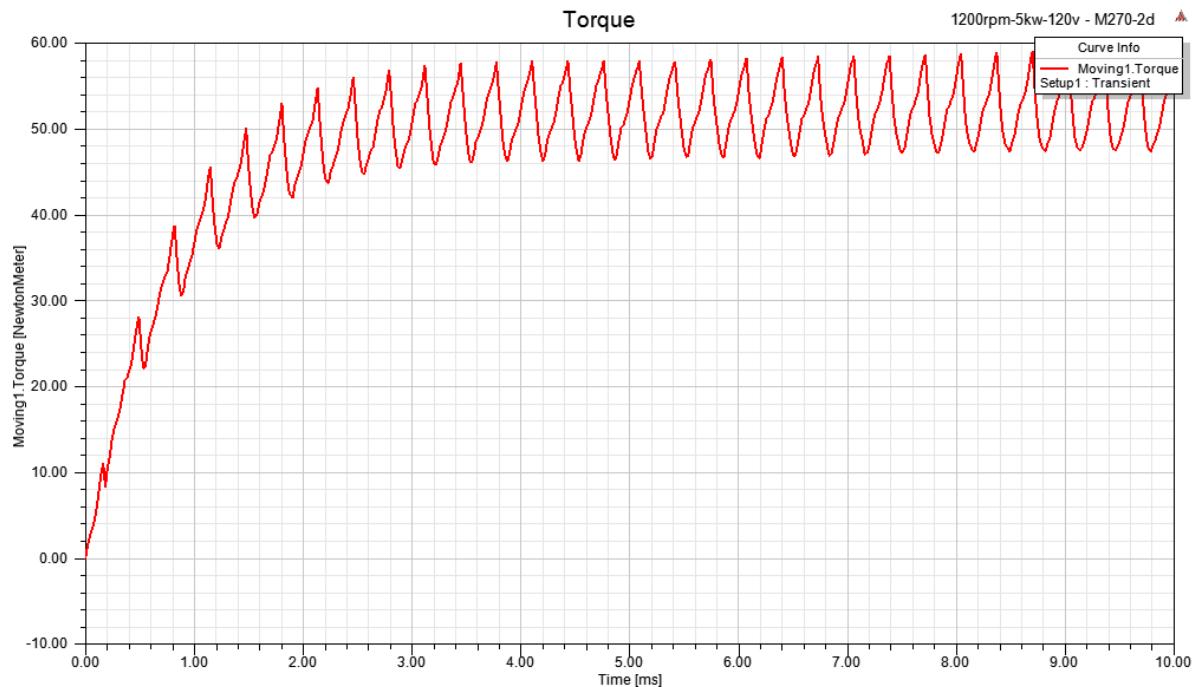
Şekil 100.01 Saniyedeki nüve kayıplarının dağılımı



Şekil 110-10ms arasında nüve kaybının değişimi

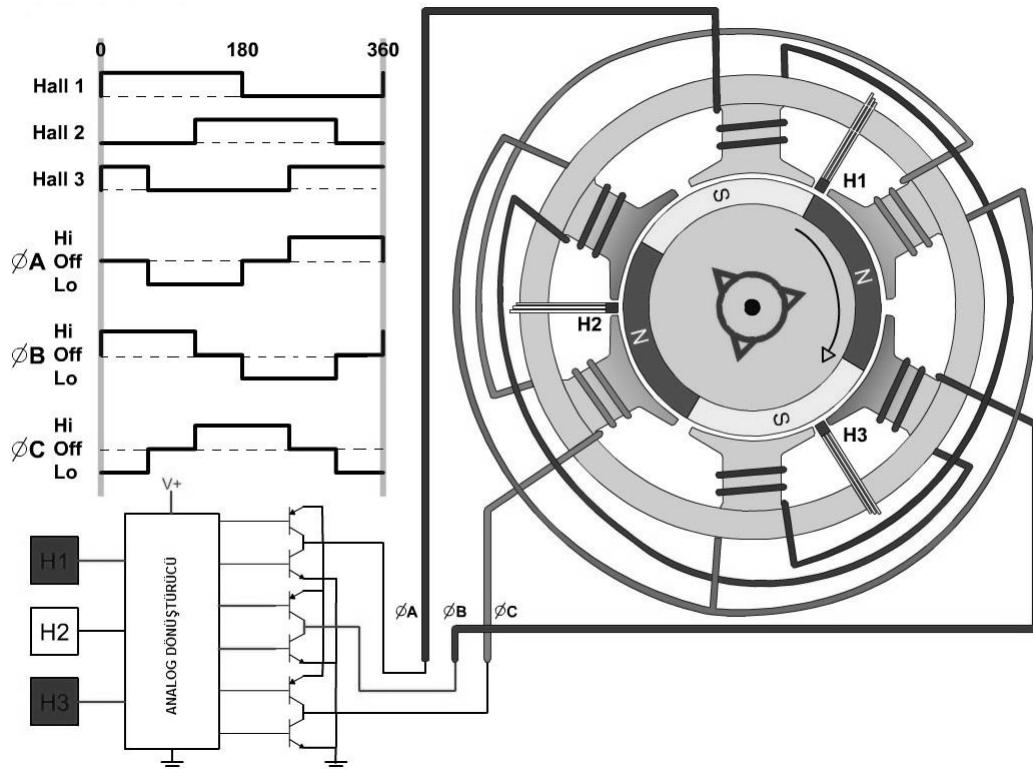


Şekil 110-10ms faz akımlarının değişimi



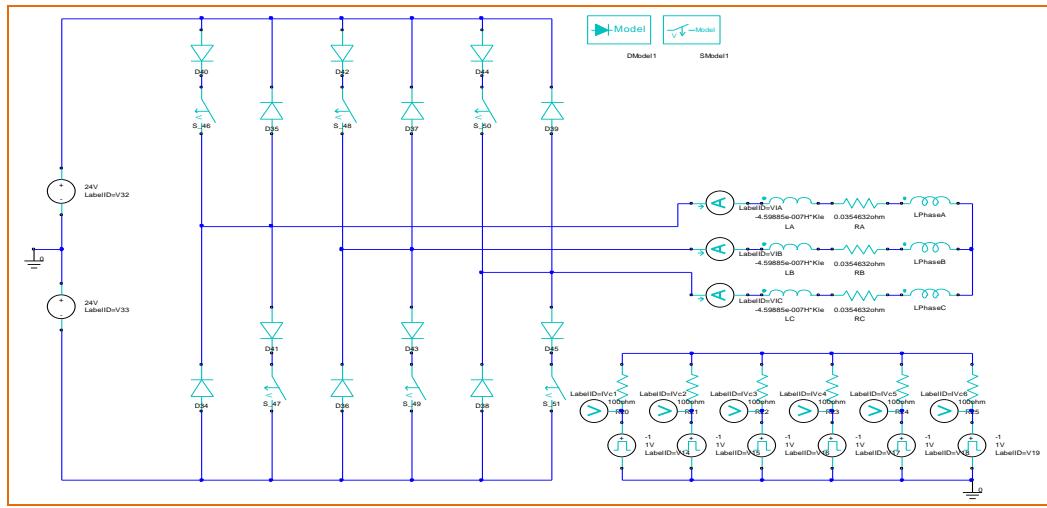
Şekil 110-10ms arasında torkun değişimi

RMxprt ile analitik hesaplamaları doğrulanan ve parametrik analizleri gerçekleştirilen iki boyutlu motor modeli elde edilen parametrik değerlerin sonlu elemanlar metodu ile doğrulanması ve ortaya konulması için Maxwell 2D yardımı ile elektromanyetik analiz döngülerine tabi tutulmuştur. Elektromanyetik analiz gerçekleştirilirken Şekil 12'de verilen devre şemasındaki uyartım şeklinin motor uçlarına anlık uygulanmış olduğu unutulmamalıdır. Sürücü devre bölümünde detaylı olarak açıklanacak olan “halleffect (alan etkili manyetik sensör)” geri besleme sinyallerine bağlı kalınarak simülasyon programında fazlar ikişer A-B, A-C ya da B-C olacak şekilde uyartılmaktadır[4].



Şekil 12 BLDC motorunun simülasyonlardaki uyartım şéklinin basit gösterimi

Sonlu elemanlar metodunun doğruluğunu kanıtlamada en önemli yere sahip olan analiz yöntemi “geçici durum” analizleridir. Geçici durum analizleri belirlenen zaman aralıklarında (örneğin 0. Sn. ile 2. Sn. aralığı) motorun gerçek zamanlı durumda olduğu gibi analiz edilerek tüm operasyonel parametrelerinin gözlemlenebildiği analizlerdir. Bu çalışmada tasarlanan ve manyetostatik analizleri gerçekleştirilen SMFDA motoru Şekil 13’de tasarlanan simülasyon devresi ile 0 ile 100ms aralığında çalıştırılarak motorun manyetik akı yoğunluğu, akı yolları, manyetik alan enerjisi, sargı akım yoğunluğu gibi parametrelerinin görsel olarak izlenmesi ve çıkış momenti, faz akımları, faz gerilimleri, kayıpları gibi birçok parametrenin grafiksel olarak gösterimi mümkün hale gelmiştir.



Şekil 13 Simülasyonda oluşturulan sürme devresi

KAYNAKLAR

1. Andrada, P., Torrent, M., Perat, J.I. and Blanqué, B., "Power Losses in Outside-Spin Brushless D.C. Motors", *Universitat Politecnica de Catalunya*, Tech. Rep. (2004).
2. Libert, F., "Design, Optimization and Comparison of Permanent Magnet Motors for a Low-Speed Direct-Driven Mixer", Doktora Tezi, *Royal Institute*, Stockholm, Sweden. (2004).
3. Rajagopal, K.R. and Sathaiah, C., "Computer Aided Design and FE Analysis of a PM BLDG Hub Motor", *International Conference on Power Electronics, Drives and Energy System PEDES '06*, 1-6 (2006).
4. Ocak, Cemil., ELEKTRİKLİ ARAÇLAR İÇİN ÜÇ KADEMELİ YENİ BİR FIRÇASIZ DA MOTORU TASARIMI, ANALİZİ VE UYGULAMASI, Doktora Tezi., GAZİ ÜNİVERSİTESİ, FEN BİLİMLERİ ENSTİTÜSÜ, EKİM 2013

c. Mechanical analysis studies



Simülasyon - shaft

Tarih: 19 Temmuz 2017

Etüt adı: Static 1

Analiz tipi: Static

İçindekiler

Tanım

Model Bilgisi

Malzeme Özellikleri

Yükler ve Fikstürler

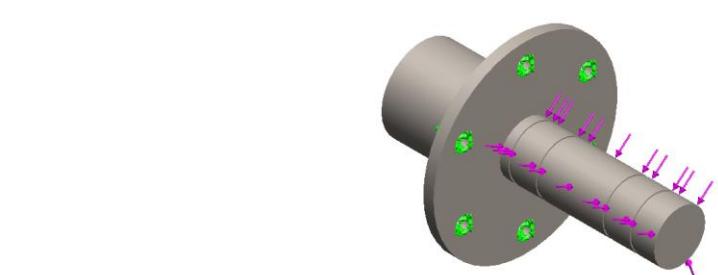
Etüt Sonuçları

Sonuç

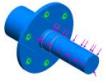
Tanım

Hub Motor, 1 Ton YükAltında, ŞaftAnalizi

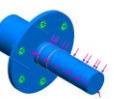
Model Bilgisi



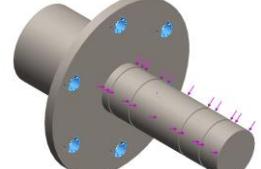
Model adı: shaft2
Geçerli Konfigürasyon: Varsayılan

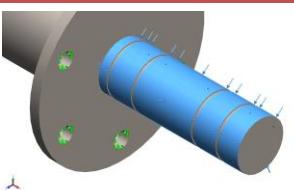
Katı Gövdeler			
Belge Adı ve Referansı	Şöyle Davran	Hacimsel Özellikler	Belge Yolu/Değiştirilme Tarihi
Kes-Ekstrüzyon12 	Katı Gövde	Kütle:0.975904 kg Hacim:0.000124319 m ³ Yoğunluk:7850 kg/m ³ Ağırlık:9.56386 N	

MalzemeÖzellikleri

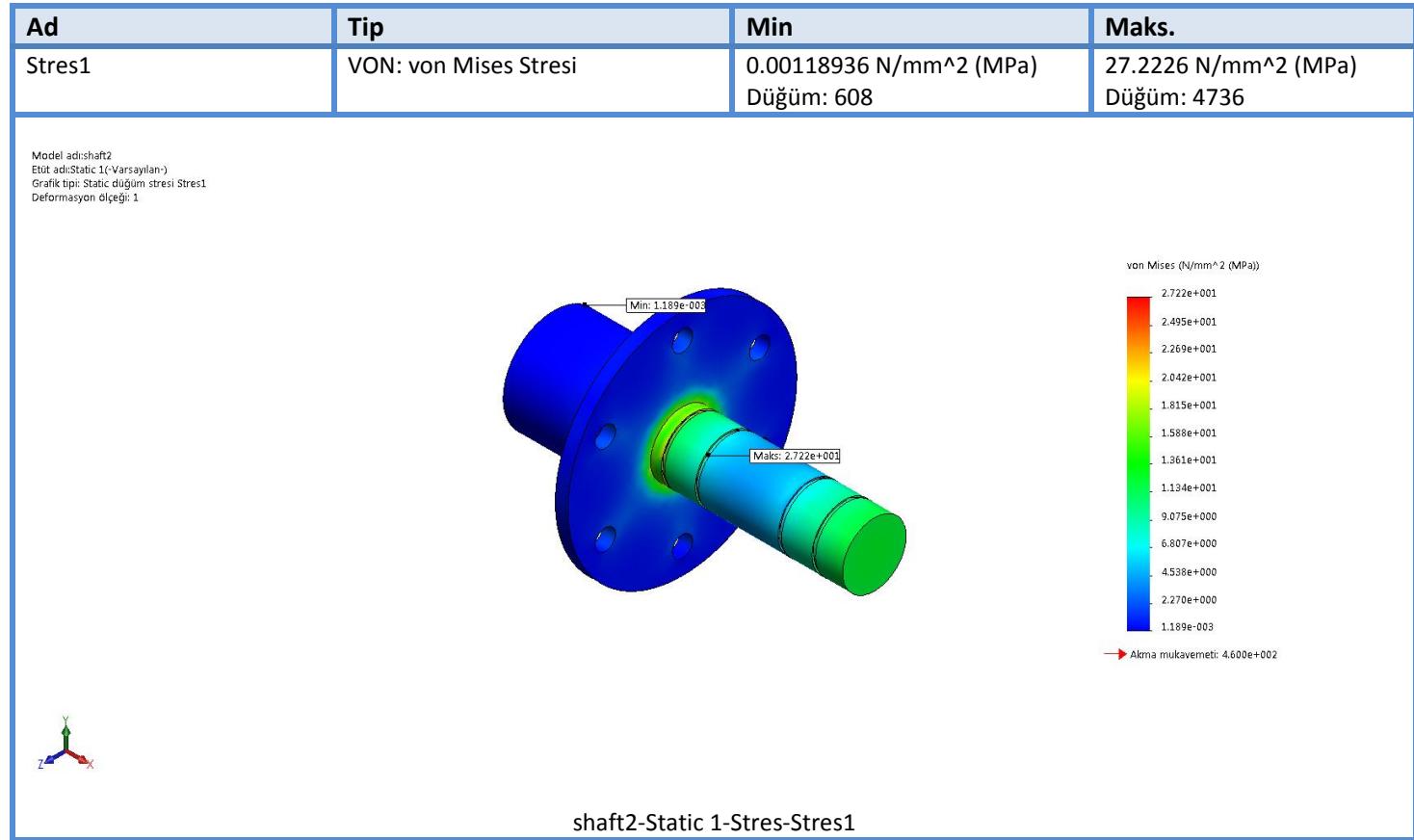
Model Referansı	Özellikler	Bileşenler
	<p>Ad: AISI 4140 Çelik, 870C'de normalize edilmiş</p> <p>Model tipi: İzotropikDoğrusalElastik Analizi</p> <p>Varsayılanhatakriteri: Bilinmeyen</p> <p>Akmamukavemeti: 4.6e+008 N/m²</p> <p>Gerilmemukavemeti: 7.31e+008 N/m²</p> <p>Elastikmodül: 2.05e+011 N/m²</p> <p>Poisson oranı: 0.285</p> <p>Kütleyoğunluğu: 7850 kg/m³</p> <p>Yırtılmamodülü: 8e+010 N/m²</p>	SolidBody 1(Kes-Ekstrüzyon12)(shaft2)
EğriVerisi:N/A		

YüklerveFikstürler

Fikstüradı	FikstürResmi	FikstürDetayları															
Sabitlenmiş-1		Objeler: 6 yüzler Tip: SabitGeometri															
SonuçKuvvetleri																	
<table border="1"> <thead> <tr> <th>Bileşenler</th> <th>X</th> <th>Y</th> <th>Z</th> <th>Sonuç</th> </tr> </thead> <tbody> <tr> <td>Tepkikuvveti(N)</td> <td>-0.00272548</td> <td>0.0461204</td> <td>-0.0289028</td> <td>0.0544967</td> </tr> <tr> <td>TepkiMomenti(N.m)</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> </tr> </tbody> </table>			Bileşenler	X	Y	Z	Sonuç	Tepkikuvveti(N)	-0.00272548	0.0461204	-0.0289028	0.0544967	TepkiMomenti(N.m)	0	0	0	0
Bileşenler	X	Y	Z	Sonuç													
Tepkikuvveti(N)	-0.00272548	0.0461204	-0.0289028	0.0544967													
TepkiMomenti(N.m)	0	0	0	0													

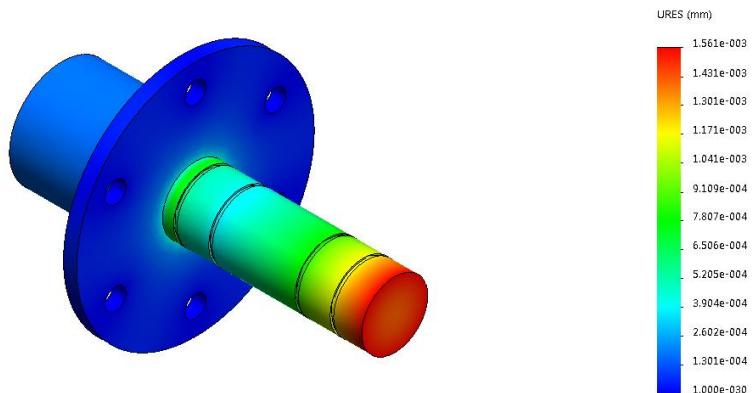
Yükadı	ResimYükle	YükDetayları
Kuvvet-1		Objeler: 5 yüzler Tip: Normal kuvvetugula Değer: 10000 N

Etüt Sonuçları



Ad	Tip	Min	Maks.
Yer değiştirme1	URES: SonuçYerDeğiştirmesi	0 mm Düğüm: 1	0.00156147 mm Düğüm: 253

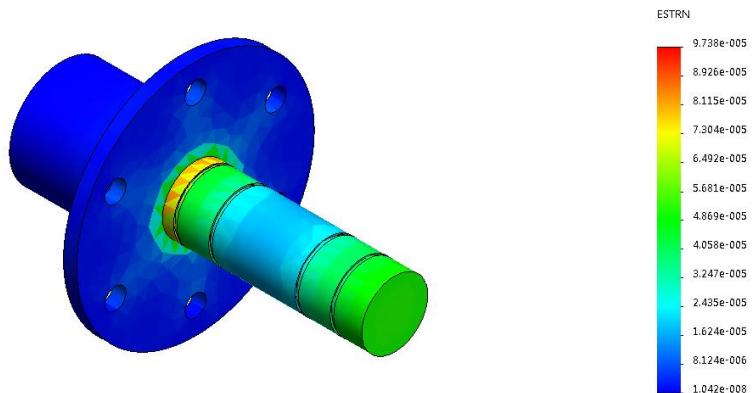
Model adı: shaft2
 Etüt adı: Static 1 (Varsayılan)
 Grafik tipi: Statik yer değiştirme-Yer değiştirme
 Deformasyon ölçüği: 1



shaft2-Static 1-Yer değiştirme-Yer değiştirme1

Ad	Tip	Min	Maks.
Gerinim1	ESTRN: Eşdeğer Gerilme	1.04234e-008 Eleman: 8599	9.73765e-005 Eleman: 5478

Model adı: shaft2
 Etüt adı: Static 1 (Varsayılan)
 Grafik tipi: Static gerinim-Gerinim1
 Deformasyon ölçüği: 1

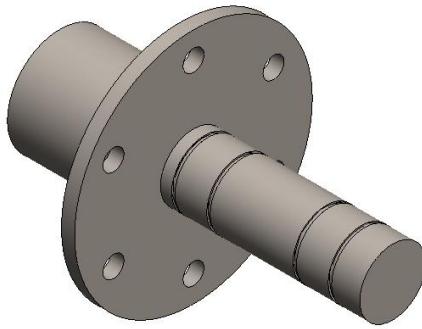


shaft2-Static 1-Gerinim-Gerinim1

Ad	Tip

Yer değiştirme1{1}**Deformeşekil**

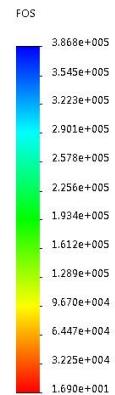
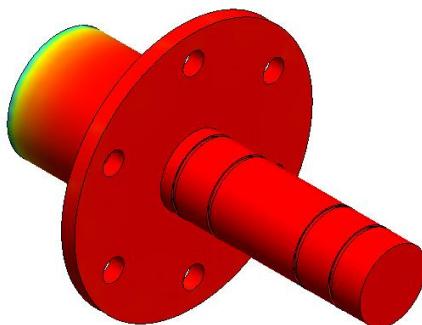
Model adı: shaft2
Etüt adı: Static 1(-Varsayılan-)
Grafik tipi: Deforme şekil Yer değiştirme1{1}
Deformasyon ölçüği: 1



shaft2-Static 1-Yer değiştirme-Yer değiştirme1{1}

Ad	Tip	Min	Maks.
Güvenlik Faktörü1	Otomatik	16.8977 Düğüm: 4736	386764 Düğüm: 608

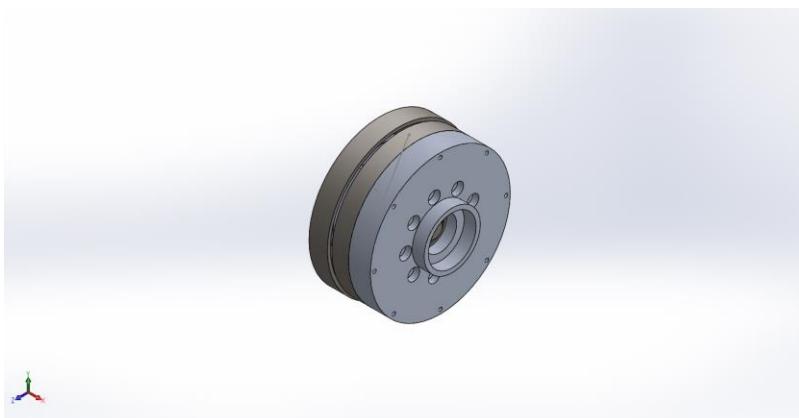
Model adı: shaft2
Etüt adı: Static 1(-Varsayılan-)
Grafik tipi: Güvenlik Faktörü Güvenlik Faktörü1
Kriter: Otomatik
Güvenlik faktörü dağılımı: Min FOS = 17



shaft2-Static 1-Güvenlik Faktörü-Güvenlik Faktörü1

Sonuç

Şaftımız, 1 tona yükümaruz bırakıldığı takdirde 17 emkgibibir sonuç vermiştiirkibudaoldukçaiyi birdeğerdir.



Simülasyon - Montaj

Tarih: 19 Temmuz 2017

Etütadi: Dinamik 1

Analiz tipi: Doğrusaldinamikanalız (Modal Zaman Geçmişli)

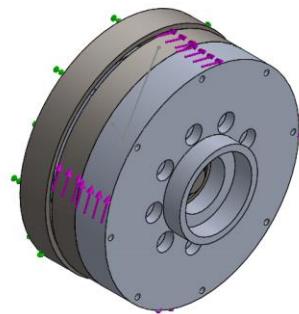
İçindekiler

- [Tanım](#)
- [Model Bilgisi.....](#)
- [Malzeme Özellikleri](#)
- [Yükler ve Fikstürler](#)
- [Etüt Sonuçları.....](#)

Tanım

40 N.m Torkaltında Dinamik Yükleme

Model Bilgisi



Model adı: montaj

Geçerli Konfigürasyon: Varsayılan

Katı Gövdeler

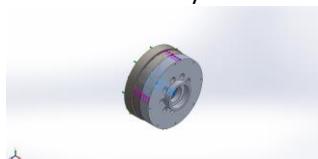
Belge Adı ve Referansı

Şöyle Davran

Hacimsel Özellikler

Belge Yolu/Değiştirilme
Tarihi

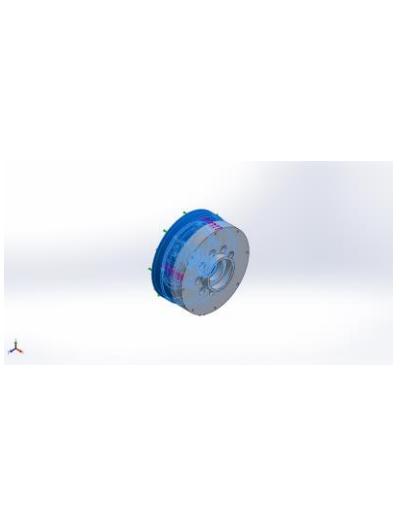
Kes-Ekstrüzyon20 	Katı Gövde	Kütle: 2.32731 kg Hacim: 0.000861967 m ³ Yoğunluk: 2700 kg/m ³ Ağırlık: 22.8076 N	
Kes-Ekstrüzyon5 	Katı Gövde	Kütle: 2.16083 kg Hacim: 0.000275266 m ³ Yoğunluk: 7850 kg/m ³ Ağırlık: 21.1762 N	
Kes-Ekstrüzyon14 	Katı Gövde	Kütle: 0.812241 kg Hacim: 0.00010347 m ³ Yoğunluk: 7850 kg/m ³ Ağırlık: 7.95996 N	
Yükseklik-Ekstrüzyon1 	Katı Gövde	Kütle: 0.117191 kg Hacim: 1.49288e-005 m ³ Yoğunluk: 7850 kg/m ³ Ağırlık: 1.14848 N	
Yükseklik-Ekstrüzyon1 	Katı Gövde	Kütle: 0.117191 kg Hacim: 1.49288e-005 m ³ Yoğunluk: 7850 kg/m ³ Ağırlık: 1.14848 N	
Yükseklik-Ekstrüzyon1 	Katı Gövde	Kütle: 0.110977 kg Hacim: 1.41372e-005 m ³ Yoğunluk: 7850 kg/m ³ Ağırlık: 1.08757 N	
Yükseklik-Ekstrüzyon4 	Katı Gövde	Kütle: 1.71091 kg Hacim: 0.00021795 m ³ Yoğunluk: 7850 kg/m ³ Ağırlık: 16.7669 N	
Kes-Ekstrüzyon8 	Katı Gövde	Kütle: 1.61771 kg Hacim: 0.000206077 m ³ Yoğunluk: 7850 kg/m ³ Ağırlık: 15.8535 N	

Kes-Ekstrüzyon12 	Katı Gövde	Kütle: 0.975904 kg Hacim: 0.000124319 m^3 Yoğunluk: 7850 kg/m^3 Ağırlık: 9.56386 N
---	------------	---

Malzeme Özellikleri

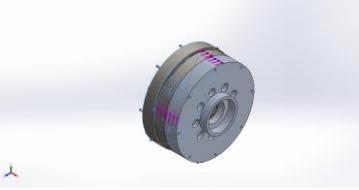
Model Referansı	Özellikler	Bileşenler																				
	<table border="1"> <tr> <td>Ad:</td><td>6061 Alaşım</td></tr> <tr> <td>Model tipi:</td><td>İzotropikDoğrusalElastik Analizi</td></tr> <tr> <td>Varsayılanhatakriteri:</td><td>Bilinmeyen</td></tr> <tr> <td>Akmamukavemeti:</td><td>5.51485e+007 N/m^2</td></tr> <tr> <td>Gerilmemukavemeti:</td><td>1.24084e+008 N/m^2</td></tr> <tr> <td>Elastikmodül:</td><td>6.9e+010 N/m^2</td></tr> <tr> <td>Poisson oranı:</td><td>0.33</td></tr> <tr> <td>Kütleyoğunluğu:</td><td>2700 kg/m^3</td></tr> <tr> <td>Yırtılmamodülü:</td><td>2.6e+010 N/m^2</td></tr> <tr> <td>Termalgenleşmekatsayı si:</td><td>2.4e-005 /Kelvin</td></tr> </table>	Ad:	6061 Alaşım	Model tipi:	İzotropikDoğrusalElastik Analizi	Varsayılanhatakriteri:	Bilinmeyen	Akmamukavemeti:	5.51485e+007 N/m^2	Gerilmemukavemeti:	1.24084e+008 N/m^2	Elastikmodül:	6.9e+010 N/m^2	Poisson oranı:	0.33	Kütleyoğunluğu:	2700 kg/m^3	Yırtılmamodülü:	2.6e+010 N/m^2	Termalgenleşmekatsayı si:	2.4e-005 /Kelvin	SolidBody 1(Kes-Ekstrüzyon20)(Rotor1-1)
Ad:	6061 Alaşım																					
Model tipi:	İzotropikDoğrusalElastik Analizi																					
Varsayılanhatakriteri:	Bilinmeyen																					
Akmamukavemeti:	5.51485e+007 N/m^2																					
Gerilmemukavemeti:	1.24084e+008 N/m^2																					
Elastikmodül:	6.9e+010 N/m^2																					
Poisson oranı:	0.33																					
Kütleyoğunluğu:	2700 kg/m^3																					
Yırtılmamodülü:	2.6e+010 N/m^2																					
Termalgenleşmekatsayı si:	2.4e-005 /Kelvin																					

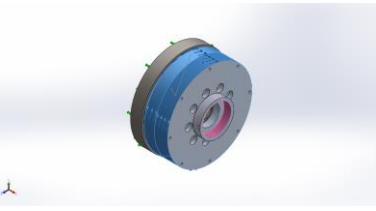
EğriVerisi:N/A

	<table border="1"> <tr> <td>Ad:</td><td>St37, AISI 4130 Çelik, normalize edilmiş</td></tr> <tr> <td>Model tipi:</td><td>İzotropikDoğrusalElastik Analizi</td></tr> <tr> <td>Varsayılanhatakriteri:</td><td>Bilinmeyen</td></tr> <tr> <td>Akmamukavemeti:</td><td>4.6e+008 N/m^2</td></tr> <tr> <td>Gerilmemukavemeti:</td><td>7.31e+008 N/m^2</td></tr> <tr> <td>Elastikmodül:</td><td>2.05e+011 N/m^2</td></tr> <tr> <td>Poisson oranı:</td><td>0.285</td></tr> <tr> <td>Kütleyoğunluğu:</td><td>7850 kg/m^3</td></tr> <tr> <td>Yırtılmamodülü:</td><td>8e+010 N/m^2</td></tr> </table>	Ad:	St37, AISI 4130 Çelik, normalize edilmiş	Model tipi:	İzotropikDoğrusalElastik Analizi	Varsayılanhatakriteri:	Bilinmeyen	Akmamukavemeti:	4.6e+008 N/m^2	Gerilmemukavemeti:	7.31e+008 N/m^2	Elastikmodül:	2.05e+011 N/m^2	Poisson oranı:	0.285	Kütleyoğunluğu:	7850 kg/m^3	Yırtılmamodülü:	8e+010 N/m^2	SolidBody 1(Kes-Ekstrüzyon5)(Rotor3-1), SolidBody 1(Kes-Ekstrüzyon14)(Rulman Yatağı-1), SolidBody 1(Yükseklik-Ekstrüzyon1)(W 6005-2Z-1), SolidBody 1(Yükseklik-Ekstrüzyon1)(W 6005-2Z-2), SolidBody 1(Yükseklik-Ekstrüzyon1)(bağlantı bilezik-1), SolidBody 1(Yükseklik-Ekstrüzyon4)(bilezik-1), SolidBody 1(Kes-Ekstrüzyon8)(kapak1-1), SolidBody 1(Kes-Ekstrüzyon12)(shaft2-1)
Ad:	St37, AISI 4130 Çelik, normalize edilmiş																			
Model tipi:	İzotropikDoğrusalElastik Analizi																			
Varsayılanhatakriteri:	Bilinmeyen																			
Akmamukavemeti:	4.6e+008 N/m^2																			
Gerilmemukavemeti:	7.31e+008 N/m^2																			
Elastikmodül:	2.05e+011 N/m^2																			
Poisson oranı:	0.285																			
Kütleyoğunluğu:	7850 kg/m^3																			
Yırtılmamodülü:	8e+010 N/m^2																			

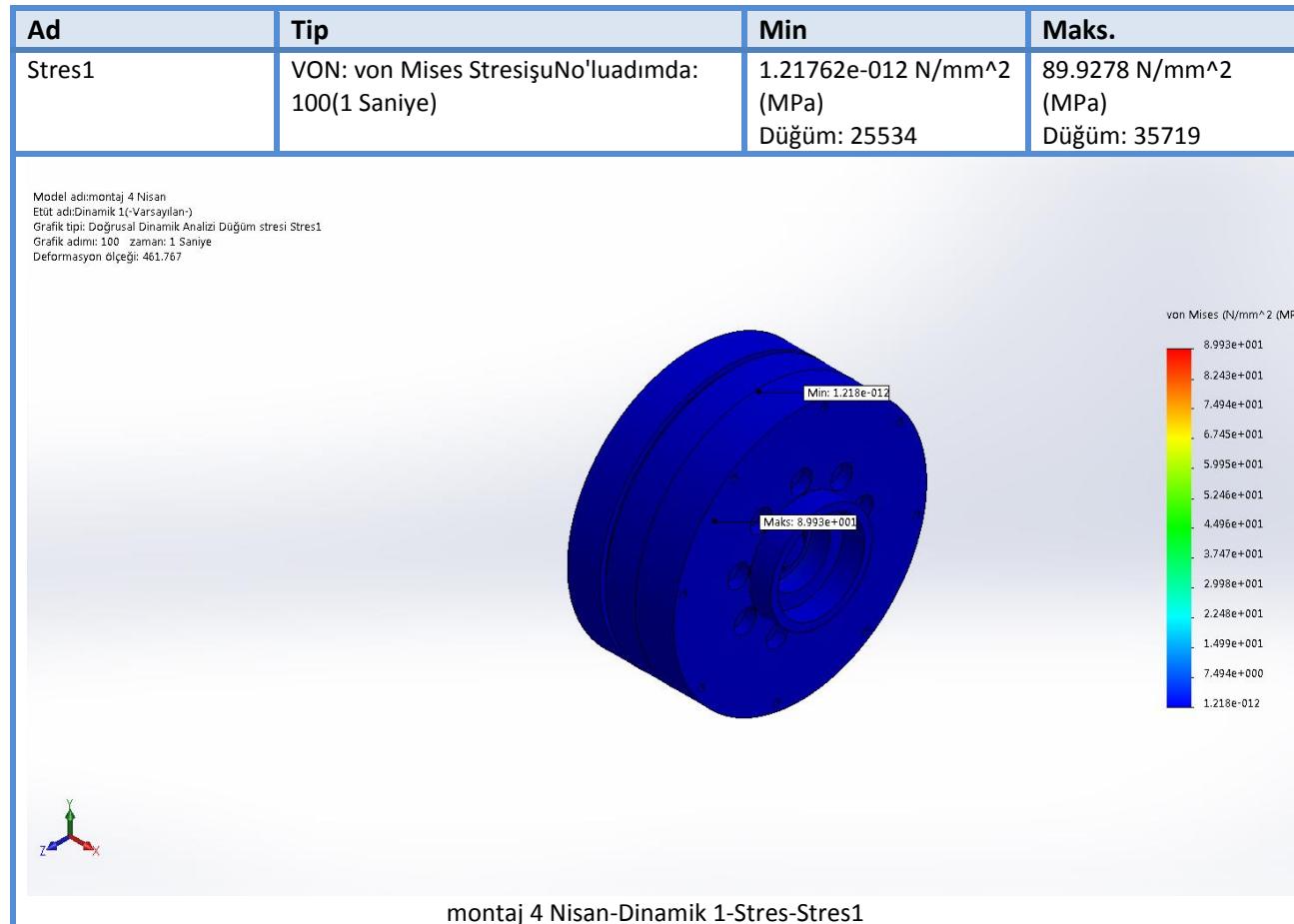
EğriVerisi:N/A

YüklerveFikstürler

Fikstüradi	FikstürResmi	FikstürDetayları		
Sabitlenmiş-1		Objeler: 4 yüzler Tip: SabitGeometri		
SonuçKuvvetleri				
Bileşenler	X	Y	Z	Sonuç
Tepkikuvveti(N)	0.000386645	5.53341e-005	-0.000100186	0.000403229
TepkiMomenti(N.m)	0	0	0	0

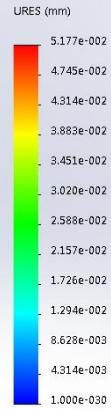
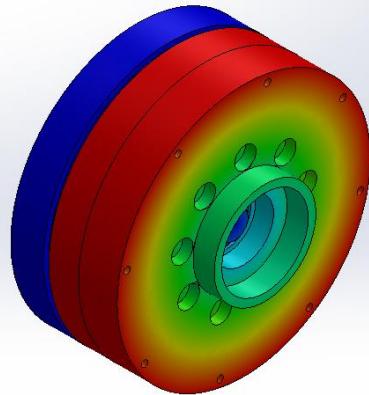
Yükadi	ResimYükle	YükDetayları
Tork-1		Objeler: 2 yüzler Referans: Yüz< 1 > Tip: Torkuygula Değer: 40 N.m

EtütSonuçları



Ad	Tip	Min	Maks.
Yer değiştirme1	URES: SonuçYerDeğiştirmesi No'luadımda: 100(1 Saniye)	0 mm Düğüm: 21173	0.0517685 mm Düğüm: 13114

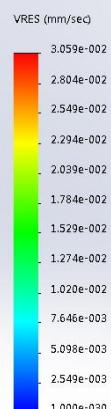
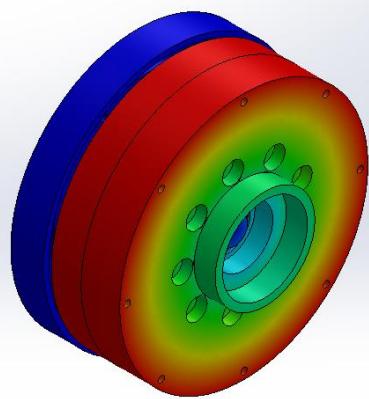
Model adı:montaj 4 Nisan
 Etüt adı:Dinamik 1(-Varsayılan)
 Grafik tipi: Doğrusal Dinamik Analizi Yer değiştirme Yer değiştirme1
 Grafik adımı: 100 zaman: 1 Saniye
 Deformasyon ölçüği: 461.767



montaj 4 Nisan-Dinamik 1-Yer değiştirme-Yer değiştirme1

Ad	Tip	Min	Maks.
Hız1	VRES: SonuçHızıŞuNo'luadımda: 100(1 Saniye)	0 mm/sec Düğüm: 21173	0.0305859 mm/sec Düğüm: 8464

Model adı:montaj 4 Nisan
 Etüt adı:Dinamik 1(-Varsayılan)
 Grafik tipi: Doğrusal Dinamik Analizi Hız Hız1
 Grafik adımı: 100 zaman: 1 Saniye
 Deformasyon ölçüği: 1

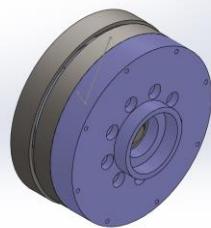


montaj 4 Nisan-Dinamik 1-Hız-Hız1

KütleKatılımı (Normalize)

Mod Sayısı	Frekans(Hertz)	X yönü	Y yönü	Z yönü
1	317.28	4.8981e-011	1.018e-006	1.5433e-006
2	337.38	2.9767e-008	0.14048	9.6292e-005
3	337.8	1.8675e-009	9.1839e-005	0.14082
4	956.44	4.257e-006	0.36369	0.042481
5	958.03	2.3545e-006	0.04247	0.36324
6	1253.6	0.45174	9.9279e-007	3.1622e-006
7	1403.6	2.245e-007	1.311e-007	3.3689e-009
8	1403.9	1.6599e-007	2.3532e-008	1.8863e-008
9	3059.2	3.2319e-009	1.8892e-009	1.035e-010
10	3059.9	3.7082e-008	4.7577e-009	1.0812e-009
11	4612	5.0291e-007	5.3035e-007	4.3519e-007
12	4613.3	1.2937e-006	3.163e-008	6.3411e-008
13	4947.1	0.00011753	0.012435	0.00048928
14	4954.8	0.00016833	0.00049698	0.012473
15	4992.7	4.1832e-007	3.8225e-007	2.9407e-008
		Toplam X = 0.45204	Toplam Y = 0.55967	Toplam Z = 0.5596

Simülasyon - Montaj



Tarih: 19 Temmuz 2017

Etüdatı:Static 1

Analiz tipi:Static

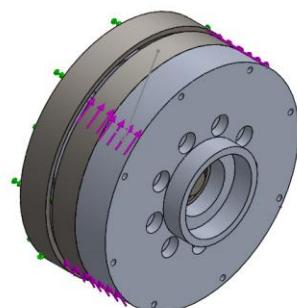
İçindekiler

- Tanım
- Model Bilgisi.....
- Malzeme Özellikleri
- Yükler ve Fikstürler
- Etüt Sonuçları.....
- Sonuç

Tanım

Montaj; 80 N.m Torkaltında, Mukavemet, Deformasyon, Dayanım ve emksonuçları.

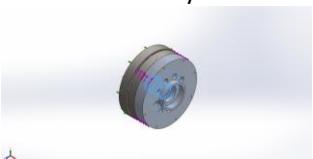
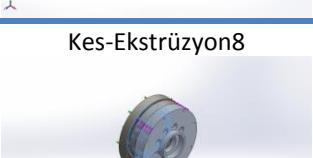
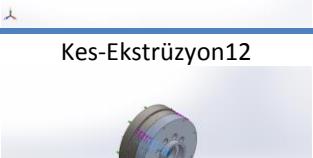
Model Bilgisi



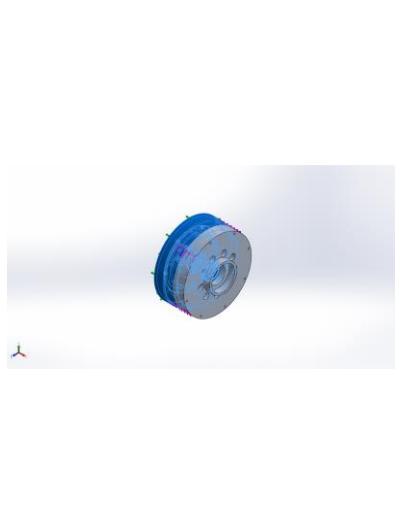
Model adı: montaj
Geçerli Konfigürasyon: Varsayılan

Katı Gövdeler

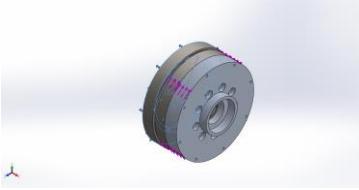
Belge Adı ve Referansı	Şöyle Davran	Hacimsel Özellikler	Belge Yolu/Değişтирilme Tarihi
Kes-Ekstrüzyon20 	Katı Gövde	Kütle: 2.32731 kg Hacim: 0.000861967 m ³ Yoğunluk: 2700 kg/m ³ Ağırlık: 22.8076 N	
Kes-Ekstrüzyon5 	Katı Gövde	Kütle: 2.16083 kg Hacim: 0.000275266 m ³ Yoğunluk: 7850 kg/m ³ Ağırlık: 21.1762 N	

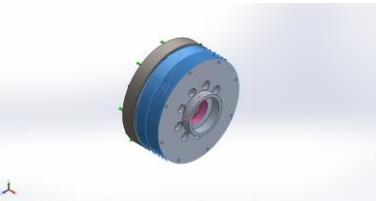
Kes-Ekstrüzyon14 	Katı Gövde	Kütle: 0.812241 kg Hacim: 0.00010347 m ³ Yoğunluk: 7850 kg/m ³ Ağırlık: 7.95996 N	
Yükseklik-Ekstrüzyon1 	Katı Gövde	Kütle: 0.117191 kg Hacim: 1.49288e-005 m ³ Yoğunluk: 7850 kg/m ³ Ağırlık: 1.14848 N	
Yükseklik-Ekstrüzyon1 	Katı Gövde	Kütle: 0.117191 kg Hacim: 1.49288e-005 m ³ Yoğunluk: 7850 kg/m ³ Ağırlık: 1.14848 N	
Yükseklik-Ekstrüzyon1 	Katı Gövde	Kütle: 0.110977 kg Hacim: 1.41372e-005 m ³ Yoğunluk: 7850 kg/m ³ Ağırlık: 1.08757 N	
Yükseklik-Ekstrüzyon4 	Katı Gövde	Kütle: 1.71091 kg Hacim: 0.00021795 m ³ Yoğunluk: 7850 kg/m ³ Ağırlık: 16.7669 N	
Kes-Ekstrüzyon8 	Katı Gövde	Kütle: 1.61771 kg Hacim: 0.000206077 m ³ Yoğunluk: 7850 kg/m ³ Ağırlık: 15.8535 N	
Kes-Ekstrüzyon12 	Katı Gövde	Kütle: 0.975904 kg Hacim: 0.000124319 m ³ Yoğunluk: 7850 kg/m ³ Ağırlık: 9.56386 N	

Malzeme Özellikleri

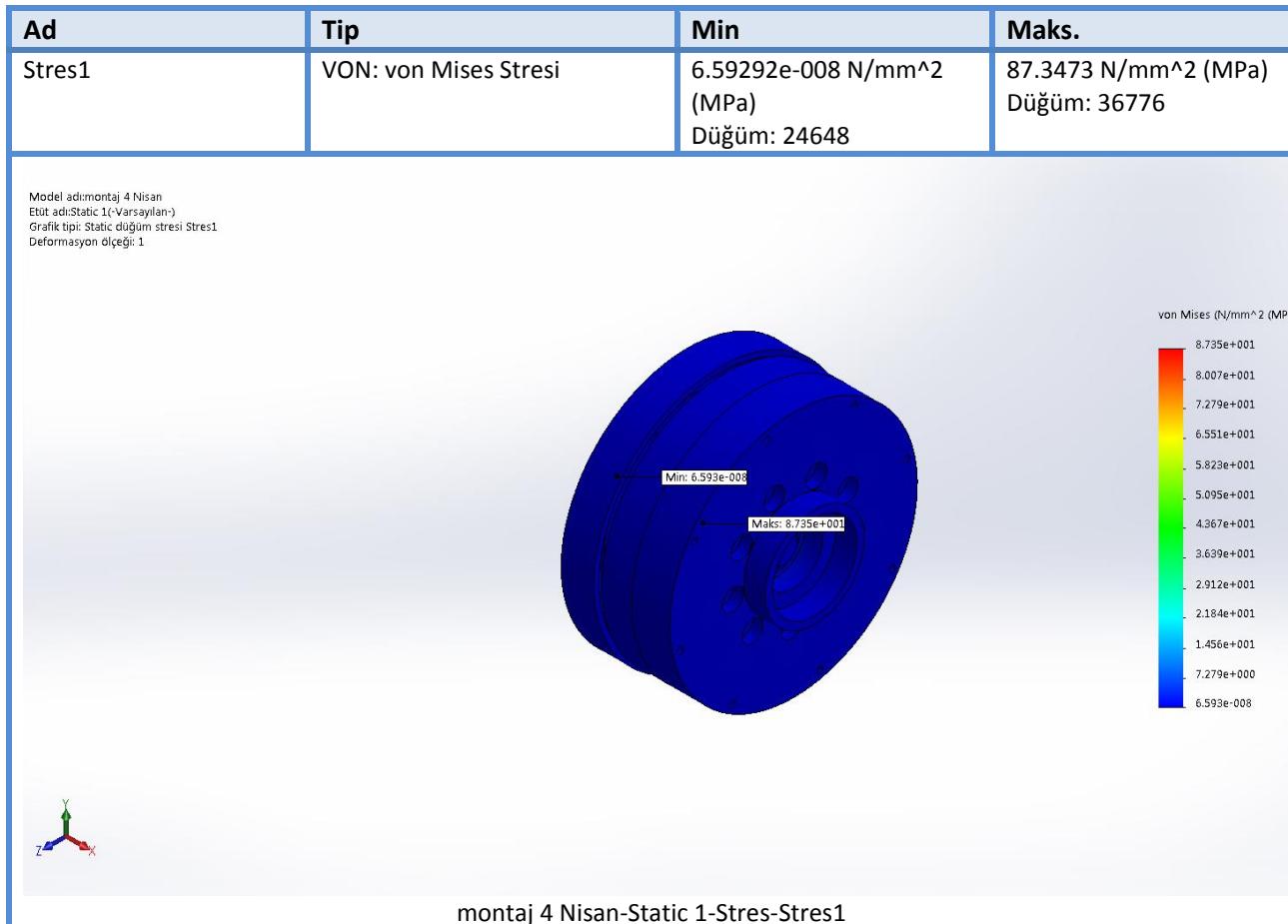
Model Referansı	Özellikler	Bileşenler																				
	<table border="1"> <tr> <td>Ad:</td><td>6061 Alaşım</td></tr> <tr> <td>Model tipi:</td><td>İzotropikDoğrusalElastik Analizi</td></tr> <tr> <td>Varsayılanhatakriteri:</td><td>Bilinmeyen</td></tr> <tr> <td>Akmamukavemeti:</td><td>5.51485e+007 N/m^2</td></tr> <tr> <td>Gerilmemukavemeti:</td><td>1.24084e+008 N/m^2</td></tr> <tr> <td>Elastikmodül:</td><td>6.9e+010 N/m^2</td></tr> <tr> <td>Poisson oranı:</td><td>0.33</td></tr> <tr> <td>Kütleyoğunluğu:</td><td>2700 kg/m^3</td></tr> <tr> <td>Yırtılmamodülü:</td><td>2.6e+010 N/m^2</td></tr> <tr> <td>Termalgenleşmekatsayı si:</td><td>2.4e-005 /Kelvin</td></tr> </table>	Ad:	6061 Alaşım	Model tipi:	İzotropikDoğrusalElastik Analizi	Varsayılanhatakriteri:	Bilinmeyen	Akmamukavemeti:	5.51485e+007 N/m^2	Gerilmemukavemeti:	1.24084e+008 N/m^2	Elastikmodül:	6.9e+010 N/m^2	Poisson oranı:	0.33	Kütleyoğunluğu:	2700 kg/m^3	Yırtılmamodülü:	2.6e+010 N/m^2	Termalgenleşmekatsayı si:	2.4e-005 /Kelvin	SolidBody 1(Kes-Ekstrüzyon20)(Rotor1-1)
Ad:	6061 Alaşım																					
Model tipi:	İzotropikDoğrusalElastik Analizi																					
Varsayılanhatakriteri:	Bilinmeyen																					
Akmamukavemeti:	5.51485e+007 N/m^2																					
Gerilmemukavemeti:	1.24084e+008 N/m^2																					
Elastikmodül:	6.9e+010 N/m^2																					
Poisson oranı:	0.33																					
Kütleyoğunluğu:	2700 kg/m^3																					
Yırtılmamodülü:	2.6e+010 N/m^2																					
Termalgenleşmekatsayı si:	2.4e-005 /Kelvin																					
EğriVerisi:N/A		<table border="1"> <tr> <td>Ad:</td><td>St 37 Çelik, normalize edilmiş</td></tr> <tr> <td>Model tipi:</td><td>İzotropikDoğrusalElastik Analizi</td></tr> <tr> <td>Varsayılanhatakriteri:</td><td>Bilinmeyen</td></tr> <tr> <td>Akmamukavemeti:</td><td>4.6e+008 N/m^2</td></tr> <tr> <td>Gerilmemukavemeti:</td><td>7.31e+008 N/m^2</td></tr> <tr> <td>Elastikmodül:</td><td>2.05e+011 N/m^2</td></tr> <tr> <td>Poisson oranı:</td><td>0.285</td></tr> <tr> <td>Kütleyoğunluğu:</td><td>7850 kg/m^3</td></tr> <tr> <td>Yırtılmamodülü:</td><td>8e+010 N/m^2</td></tr> </table>	Ad:	St 37 Çelik, normalize edilmiş	Model tipi:	İzotropikDoğrusalElastik Analizi	Varsayılanhatakriteri:	Bilinmeyen	Akmamukavemeti:	4.6e+008 N/m^2	Gerilmemukavemeti:	7.31e+008 N/m^2	Elastikmodül:	2.05e+011 N/m^2	Poisson oranı:	0.285	Kütleyoğunluğu:	7850 kg/m^3	Yırtılmamodülü:	8e+010 N/m^2	SolidBody 1(Kes-Ekstrüzyon5)(Rotor3-1), SolidBody 1(Kes-Ekstrüzyon14)(Rulman Yatağı-1), SolidBody 1(Yükseklik-Ekstrüzyon1)(W 6005-2Z-1), SolidBody 1(Yükseklik-Ekstrüzyon1)(W 6005-2Z-2), SolidBody 1(Yükseklik-Ekstrüzyon1)(bağlantı bilezik-1), SolidBody 1(Yükseklik-Ekstrüzyon4)(bilezik-1), SolidBody 1(Kes-Ekstrüzyon8)(kapak1-1), SolidBody 1(Kes-Ekstrüzyon12)(shaft2-1)	
Ad:	St 37 Çelik, normalize edilmiş																					
Model tipi:	İzotropikDoğrusalElastik Analizi																					
Varsayılanhatakriteri:	Bilinmeyen																					
Akmamukavemeti:	4.6e+008 N/m^2																					
Gerilmemukavemeti:	7.31e+008 N/m^2																					
Elastikmodül:	2.05e+011 N/m^2																					
Poisson oranı:	0.285																					
Kütleyoğunluğu:	7850 kg/m^3																					
Yırtılmamodülü:	8e+010 N/m^2																					

YüklerveFikstürler

Fikstüradi	FikstürResmi	FikstürDetayları		
Sabitlenmiş-1		Objeler: 3 yüzler Tip: SabitGeometri		
SonuçKuvvetleri				
Bileşenler	X	Y	Z	Sonuç
Tepkikuvveti(N)	0.0145943	0.0279753	-0.0716984	0.0783344
TepkiMomenti(N.m)	0	0	0	0

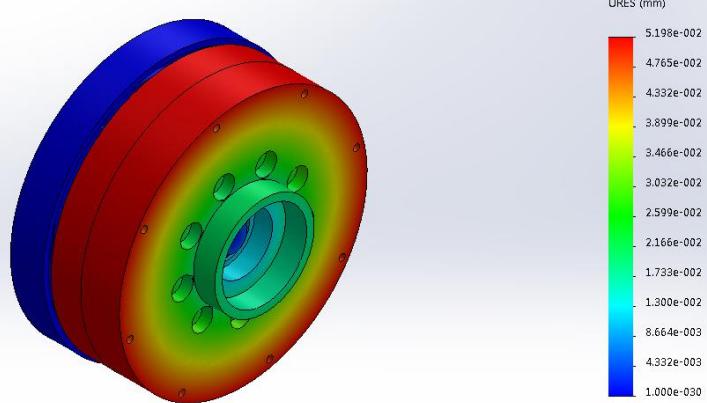
Yükadi	ResimYükle	YükDetayları
Tork-1		Objeler: 2 yüzler Referans: Yüz< 1 > Tip: Torkuygula Değer: 80 N.m

EtütSonuçları



Ad	Tip	Min	Maks.
Yer değiştirme1	URES: SonuçYerDeğiştirmesi	0 mm Düğüm: 21211	0.0519844 mm Düğüm: 14509

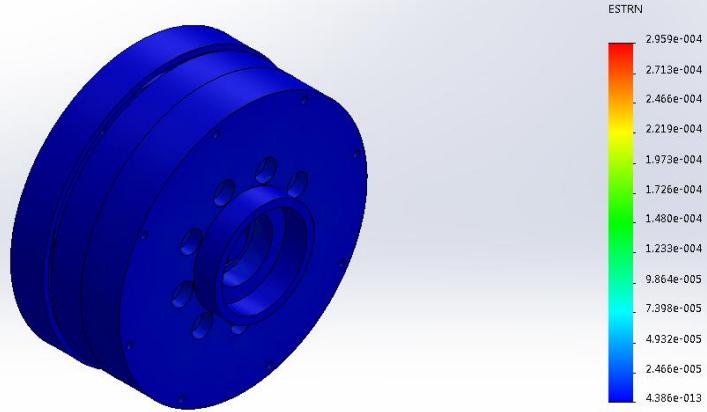
Model adı:montaj 4 Nisan
 Etüt adı:Static 1<Varsayılan>
 Grafik tipi:Statik yer değiştirme Yer değiştirme1
 Deformasyon ölçüği: 1



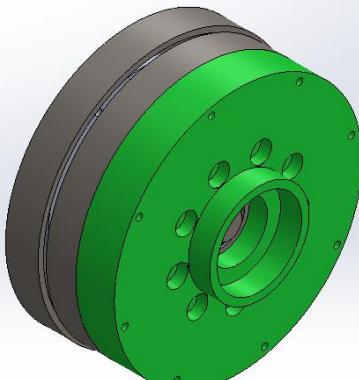
montaj 4 Nisan-Static 1-Yer değiştirme-Yer değiştirme1

Ad	Tip	Min	Maks.
Gerinim1	ESTRN: Eşdeğer Gerilme	4.38603e-013 Eleman: 12379	0.00029592 Eleman: 19687

Model adı:montaj 4 Nisan
 Etüt adı:Static 1<Varsayılan>
 Grafik tipi:Statik gerinim Gerinim1
 Deformasyon ölçüği: 1

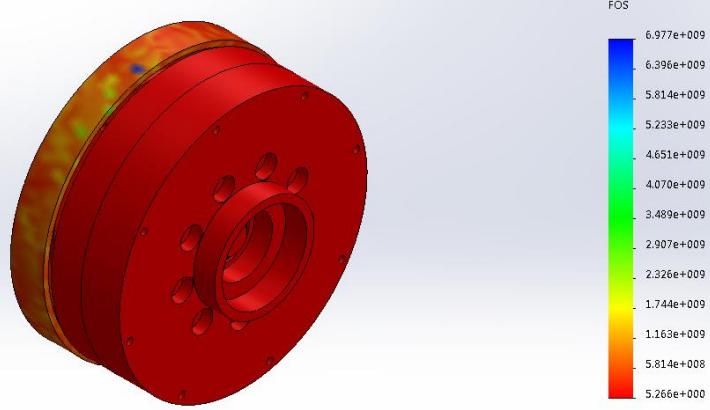


montaj 4 Nisan-Static 1-Gerinim-Gerinim1

Ad	Tip
Yer değiştirme1{1}	Deformeşekil
<p>Model adı:montaj 4 Nisan Etüt adı:Static 1(-Varsayılan-) Grafik tipi: Deforme şekil Yer değiştirme1[1] Deformasyon ölçüsü: 1</p> 	
montaj 4 Nisan-Static 1-Yer değiştirme-Yer değiştirme1{1}	

Ad	Tip	Min	Maks.
Güvenlik Faktörü1	Otomatik	5.26633 Düğüm: 36776	6.97718e+009 Düğüm: 24648

Model adı:montaj 4 Nisan
Eltüt adı:Static 1(Varsayılan)
Grafik tipi: Güvenlik Faktörü Güvenlik Faktörü
Kriter: Otomatik
Güvenlik faktörü dağılımı: Min FOS = 5.3



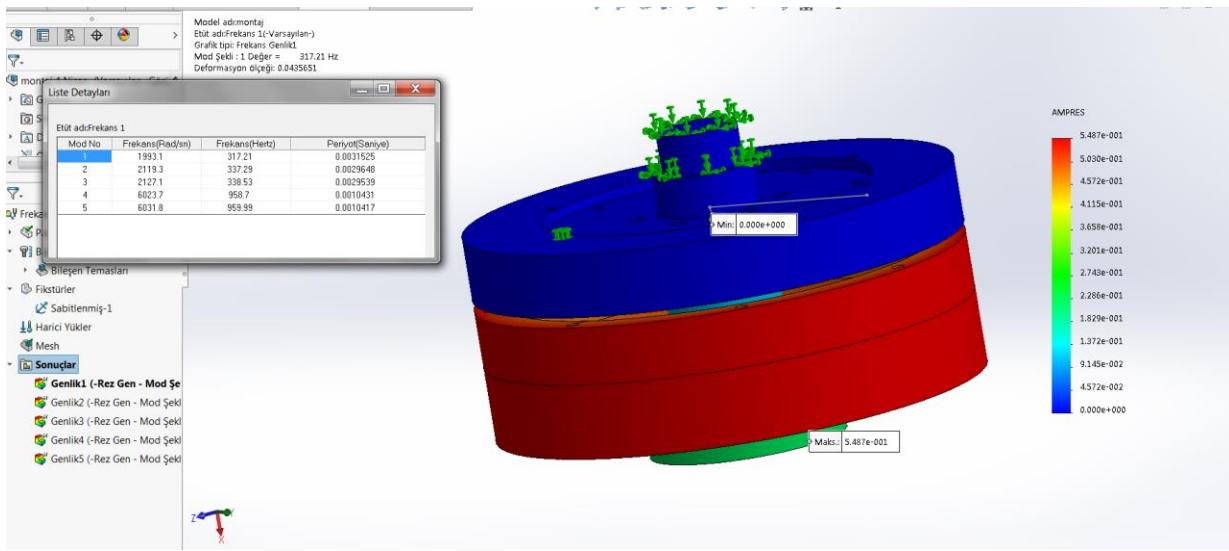
montaj 4 Nisan-Static 1-Güvenlik Faktörü-Güvenlik Faktörü

Sonuç

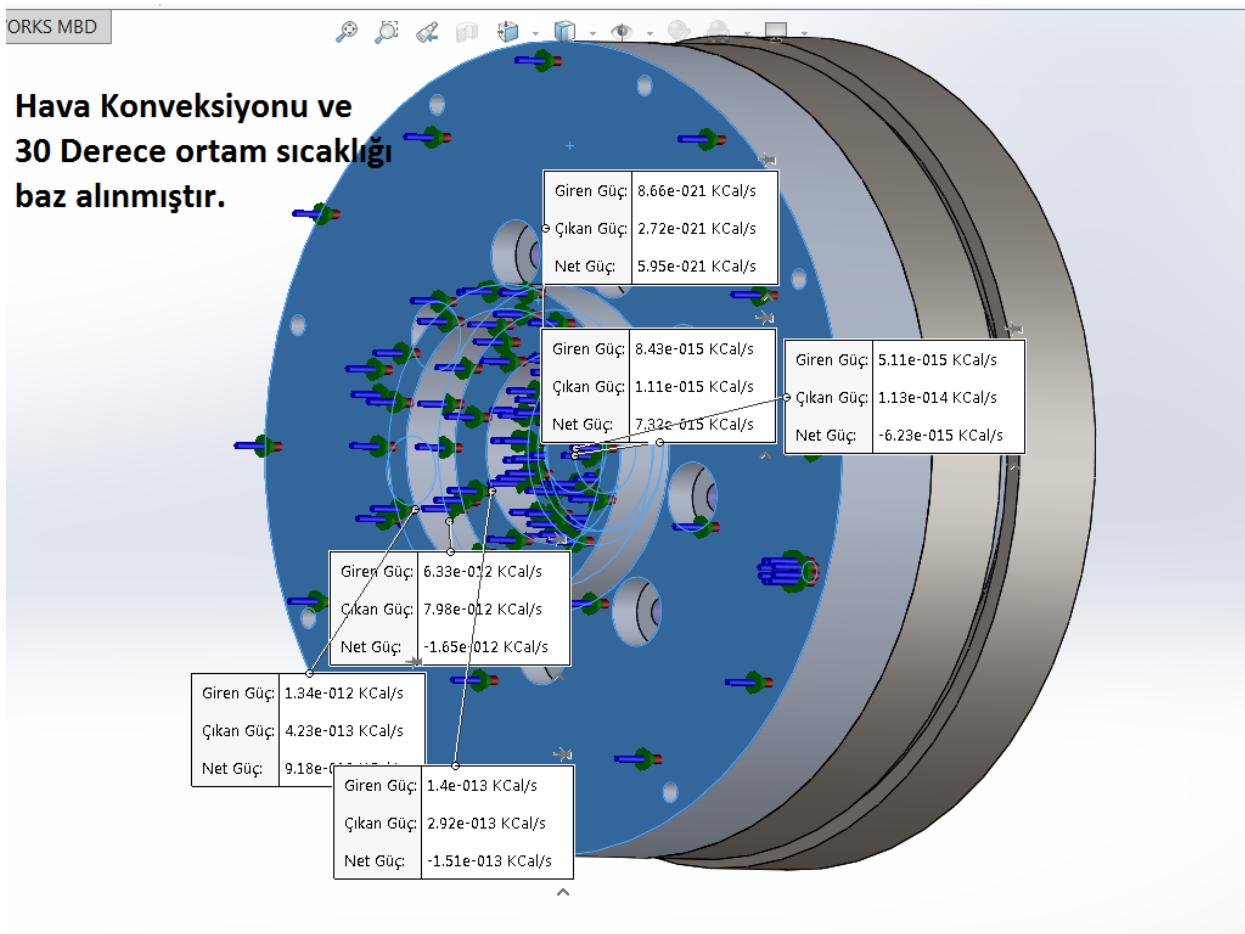
Hub Motoru, 80 N.mlikbirtorkamaruz bırakılmış ve 5.3 gibi iyibirem ksonuc ueldeedilmişdir.

Daha riyit ve daha hassas çalışma amaçlı, radyal ve eksenel yüklerle

karşı dayanıklı, 7005 serisi omuzlu rulman (Angular contact) tercih edilmiştir.



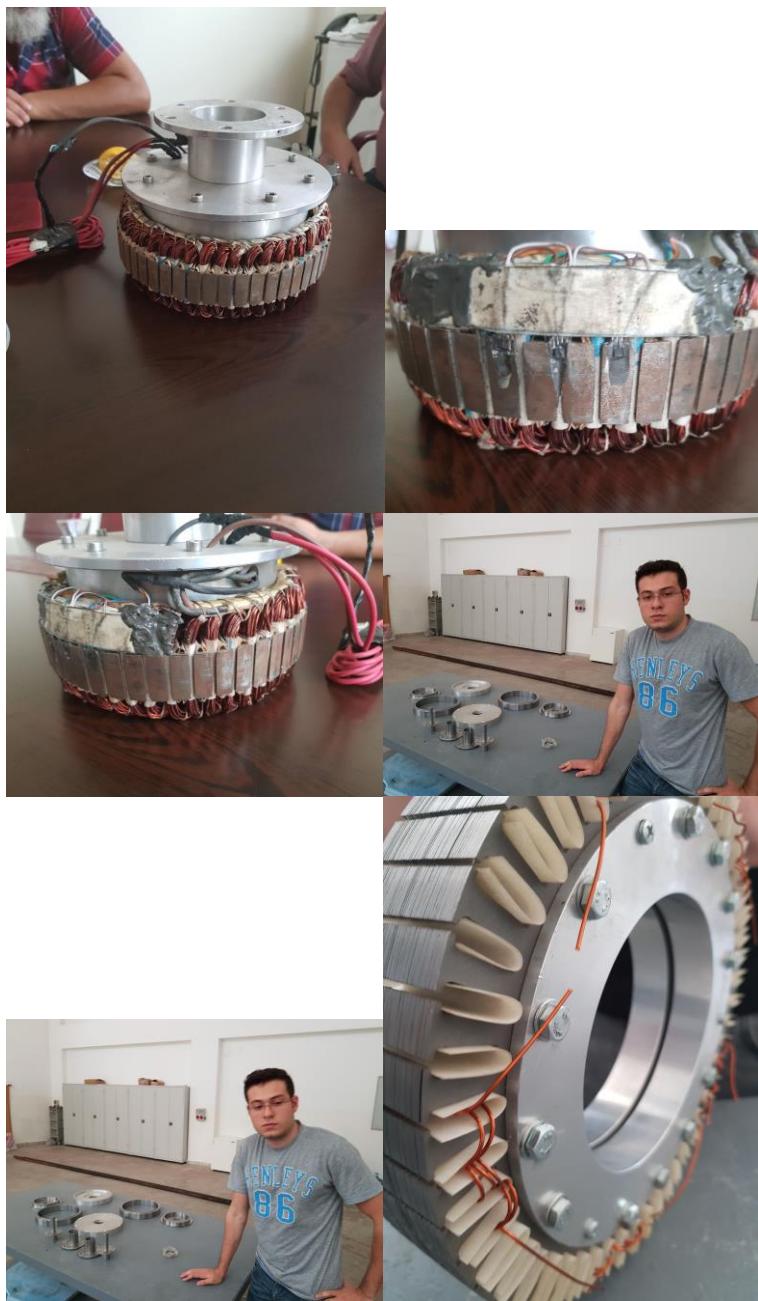
d. Thermal analysis studies

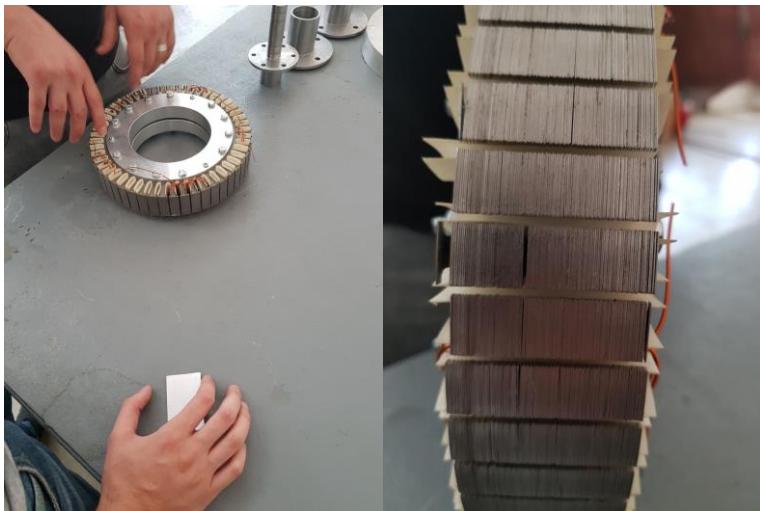


e. Production studies

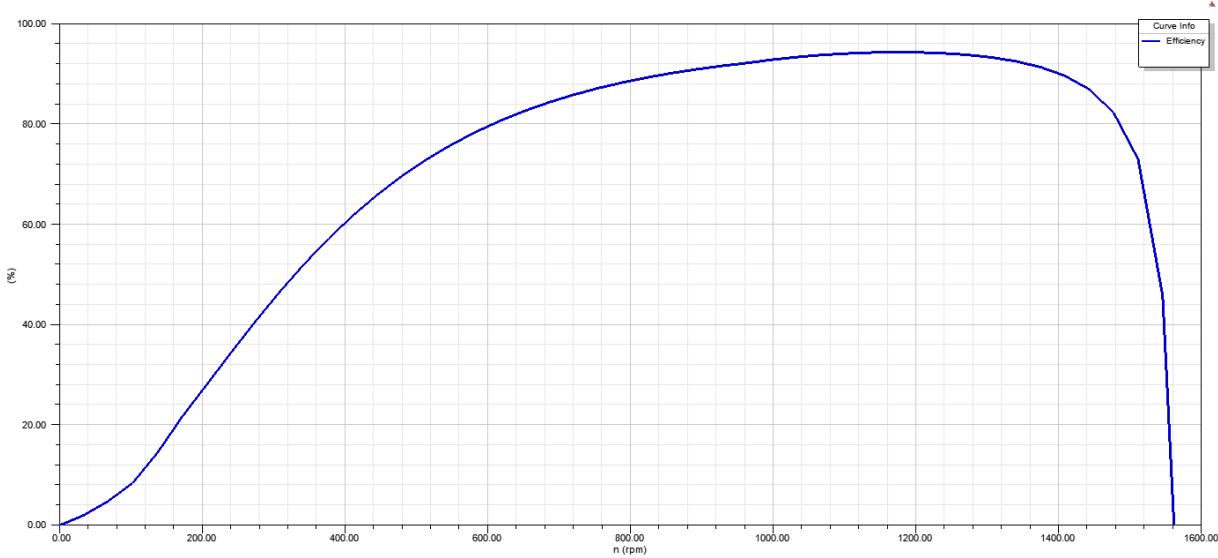








f. Motor efficiency



Performans Çıktıları	
Verim	92,00%
Anma Hızı	1346 Rpm
Anma Torku	35Nm
Maksimum Çıkış Gücü	19kW
Ortalama Giriş Akımı	45Amper
	Dc Kaynaktan Çekilen Akım
Motora Uygulanan RMS Akım	39.4 Amper

Note: Production process continues . Calculations are predominantly theoretical.

4. Motor driver(if designed by team, details mandatory; if ready-made product, please explain briefly)

All the teams are required to prepare their motor drivers themselves. Teams that have produced this product domestically will get a maximum of **300 points** depending on the design.

The teams that perform domestic motor driver design and production are responsible for the following items during the technical inspections:

- 1) Physically illustrating the motor driver.
- 2) Providing information regarding simulation studies and control algorithms.
- 3) Providing circuit and printed circuit drawings.
- 4) Submitting information related to test methods and results(You might be requested to simulate raw documents)

If designed by the team, detailed information should be given on the following topics. **Additionally, the raw documents (original files) to show the simulation and design process and results must be sent with the report in a compressed file (i.e. .zip, .rar). Otherwise it will not be considered as domestic even if a domestic motor driver is presented at technical inspections. All the designs and simulations must be the teams' own work, i.e., simulation results, pcb schematics or embedded code provided by the manufacturer of any part of the system will not be accepted. The teams might be asked to run their source files related to the simulation results, pcb schematics or embedded code and to show the sketches of their design.**

a. Circuit design

Design is in rar file which is attached to mail.

b. Control algorithm

Motor Driver Codes:

Motor driver is a module that controls motor speed according to the value which comes from ECU via UART.

unsigned short hall;

unsigned char MoveTableF[8] = {0, 33, 6, 36, 24, 9, 18, 0};

unsigned char MoveTableR[8] = {0, 18, 9, 24, 36, 6, 33, 0};

charreceivedText[5]={'1','0','0','0','/'};

intdir=1;

```

char amp[3]={0,0,0};

intamp_int=0;

//Uart data receiving function

voidserialDataRead()

{

inti=0;

while(i<5){

if(UART1_Data_Ready())

{



receivedText[i]=UART1_Read();

i++;

}

}

}

//Data converting function

voidconvertReceivedData()

{

dir=receivedText[0]-'0';

amp_int+=(receivedText[1]-'0')*100;

amp_int+=(receivedText[2]-'0')*10;

amp_int+=receivedText[3]-'0';

}

//Stop Button Function

```

```

void Interrupt(){

if(INTF_bit){

INTCON = 0;

PORTB.F2 = 0;

PORTD = 0;

PWM1_Stop();

}

if (RBIF_bit){

RBIF_bit = 0;

hall = PORTB;

hall = hall & 112;

hall = hall >> 4;

if(dir==1)

PORTD = MoveTableF[hall];

else if(dir==2)

PORTD = MoveTableR[hall];

}

}

voidInitInterrupts(){

OPTION_REG = 0;

TMR1IE_bit = 0;

INTCON = 0xD8;

}

void main() {

ADCON1 = 14;

```

```

CMCON = 7;
PORTB = 0;
TRISB = 243;
PORTD = 0;
TRISD = 0;
InitInterrupts();
PWM1_Init(10000);
UART1_Init(9600);

while(1){

PORTB.F2 = 1;
serialDataRead();
convertReceivedData();

PORTB.F2 = 0;

hall = PORTB;
hall = hall & 112;
hall = hall >> 4;

if(dir==1)

PORTD = MoveTableF[hall];

else if(dir==2){

PORTD = MoveTableR[hall];

}

PORTB.F2 = 0;

PWM1_Set_Duty(amp_int);

amp_int=0;
}

```

```
PWM1_Start();
```

```
}
```

```
}
```

Note: Motor Driver Codes had written in MikroC.

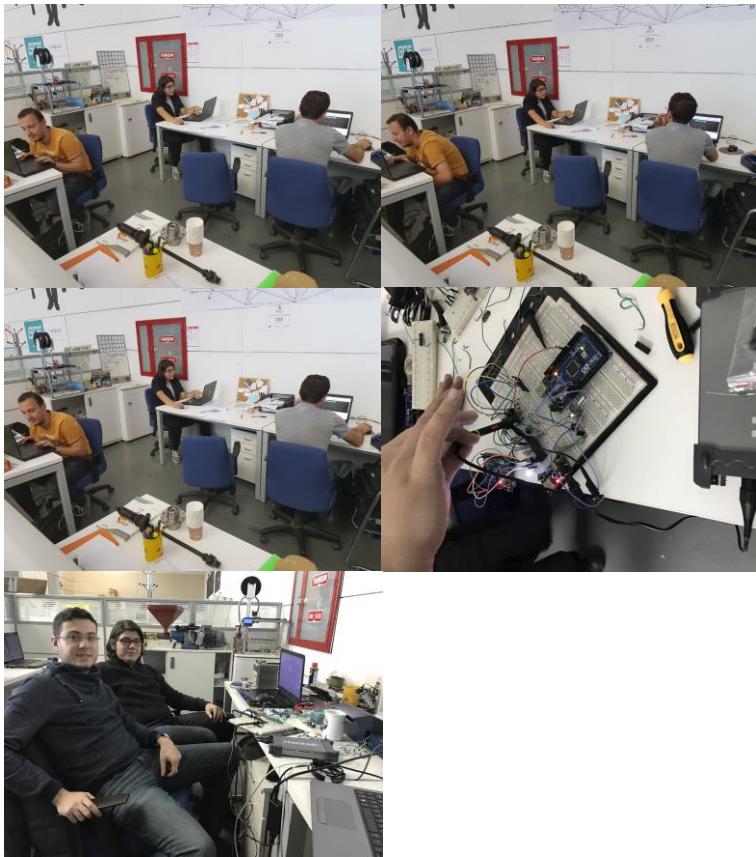
c. Simulation studies

There are no simulation study but we have constructed the circuit on breadboard and tested, circuit works well. Photos at f part.

d. Printed circuit studies

Still in progress. You can find PCB file in rar file.

e. Production studies



f. Motor driver efficiency

We havent calculated yet. Because we don't have our driver psychically yet.

5. Battery management system (BMS)(if designed by team, details mandatory; if ready-made product, please explain briefly)

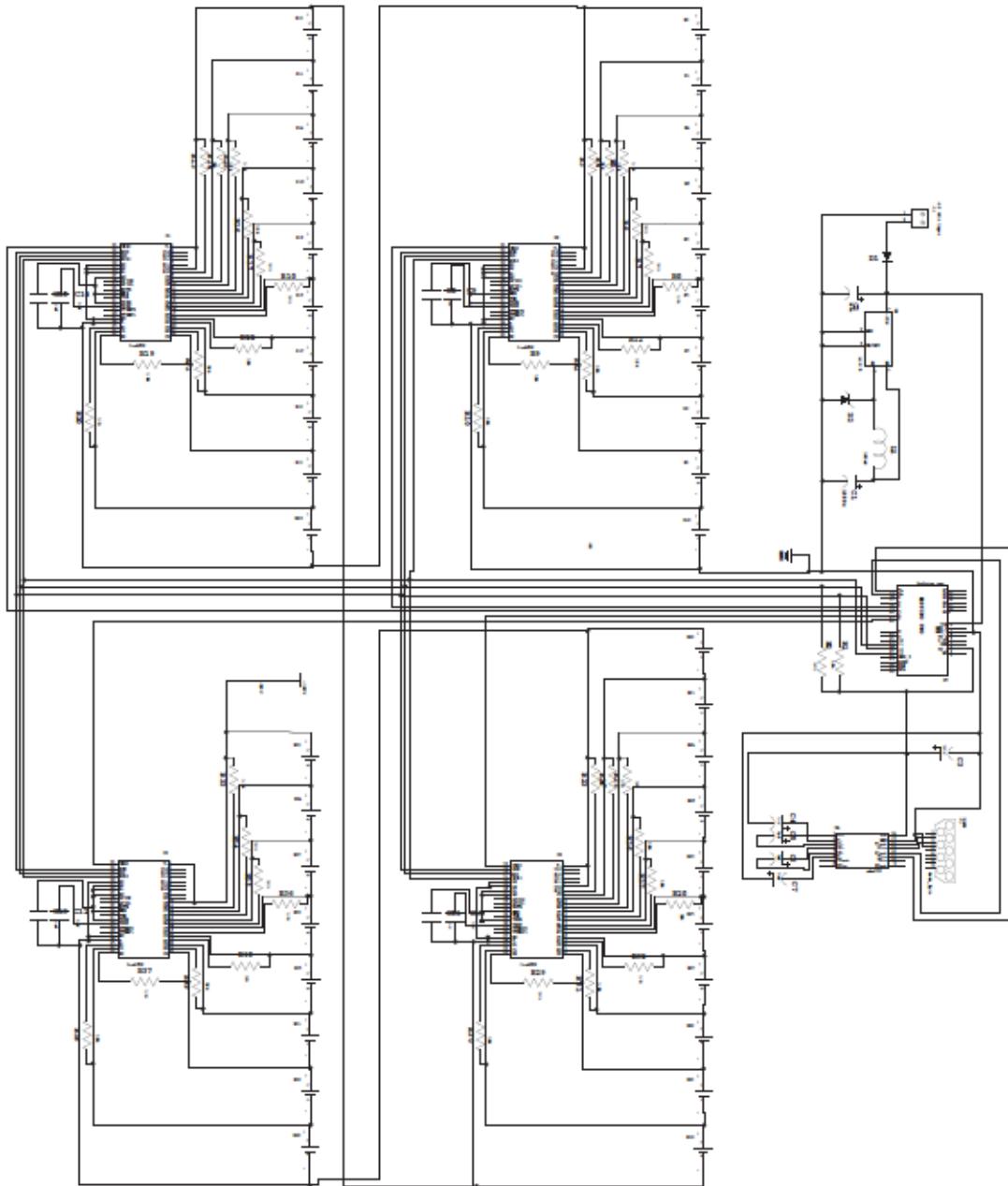
*All the teams are required to prepare their BMS themselves. Teams that produce this product domestically will get a maximum of **200 points** depending on the design.*

The teams that develop the BMS domestically are responsible for the following items during the technical inspections:

- 1- Physically illustrating the BMS.*
- 2- Submitting overall information related to BMS design and working principles.*
- 3- Submitting overall information related to the balancing method and how it is performed.*

If designed by the team, detailed information should be given on the following topics:

a. Circuit design



b. Balancing method

Passive Balancing

c. Control algorithm

```
#include "SPI.h"
#include <LiquidCrystal.h>
// SDO - PIN 12
// SDI - PIN 11
#define WRCFG 0x01 //Write Configuration Registers
#define RDCFG 0x02 // Read config
#define RDCV 0x04 // Read cells

#define STCVAD 0x10 // Start all A/D's - poll status
#define RDFLG 0x06 //Read Flags
#define RDTMP 0x08 //Read Temperatures

#define STCDC 0x60 //A/D converter and poll Status
#define STOWAD 0x20 //Start Test - poll status
#define STTMPAD 0x30// Temperature Reading - ALL

#define address 0x80
//Functions
#define TOTAL 4
#define LIMIT 0.01
byte byteTemp;
byte CFGR1 = 0x00;
byte CFGR2 = 0x00;
float cellVolt[TOTAL];

LiquidCrystal lcd(7,6,5,4,3,2);

void setup()
{
pinMode(10,OUTPUT);
pinMode(11,OUTPUT);
pinMode(12,INPUT);
pinMode(13,OUTPUT);
digitalWrite(10, HIGH);
SPI.setBitOrder(MSBFIRST);
SPI.setDataMode(SPI_MODE3);
SPI.setClockDivider(SPI_CLOCK_DIV16);
SPI.begin();
Serial.begin(9600);
//LCD Setup
lcd.begin(16,2);
lcd.setCursor(0,0);
lcd.print("C1: ");
lcd.setCursor(8,0);
lcd.print("C2: ");
lcd.setCursor(0,1);
lcd.print("C3: ");
lcd.setCursor(8,1);
lcd.print("C4: ");



}
```

```

//lcd.print("Hello world!");
writeReg(); //Initial setup
}

void loop()
{
readV();
//Serial.print("Lowest cell: "); Serial.print(lowCell()); Serial.print(" ,
Value: "); Serial.println(cellVolt[lowCell()-1]);
//Serial.print("Highest cell: "); Serial.print(highCell()); Serial.print(" ,
Value: "); Serial.println(cellVolt[highCell()-1]);
startBalance();
//delay(2000);
Serial.println(CFGR1,BIN);
lcd.setCursor(3,0);
lcd.print(cellVolt[0]);
lcd.setCursor(11,0);
lcd.print(cellVolt[1]);
lcd.setCursor(3,1);
lcd.print(cellVolt[2]);
lcd.setCursor(11,1);
lcd.print(cellVolt[3]);
delay(1000);
}

void writeReg()
{
Serial.println("Writing config...");
digitalWrite(10, LOW);
SPI.transfer(address);
SPI.transfer(WRCFG);
SPI.transfer(0x01); //0
SPI.transfer(CFGR1); //1
SPI.transfer(CFGR2); //2
SPI.transfer(0x00); //3
SPI.transfer(0x71); //4
SPI.transfer(0xAB); //5
digitalWrite(10, HIGH);
}
void readReg()
{
Serial.println("Reading config...");
digitalWrite(10, LOW);
SPI.transfer(address);
SPI.transfer(RDCFG);
for(int i = 0; i < 6; i++)
{
byteTemp = SPI.transfer(RDCFG);
Serial.println(byteTemp, HEX);
}
digitalWrite(10, HIGH);
}
void readV()
{
digitalWrite(10,LOW);
SPI.transfer(STCVAD);
delay(20); // wait at least 12ms as per data sheet, p.24
}

```

```

digitalWrite(10,HIGH);
byte volt[18];
digitalWrite(10,LOW);
SPI.transfer(0x80);
SPI.transfer(RDCV);
for(int j = 0; j<18;j++)
{
volt[j] = SPI.transfer(RDCV);
}
digitalWrite(10,HIGH);
cellVolt[0] = (((volt[0] & 0xFF) | (volt[1] & 0x0F) << 8)*1.5*0.001);
cellVolt[1] = (((volt[1] & 0xF0) >> 4 | (volt[2] & 0xFF) << 4)*1.5*0.001);
cellVolt[2] = (((volt[3] & 0xFF) | (volt[4] & 0x0F) << 8)*1.5*0.001);
cellVolt[3] = (((volt[4] & 0xF0) >> 4 | (volt[5] & 0xFF) << 4)*1.5*0.001);
Serial.println(cellVolt[0]);
Serial.println(cellVolt[1]);
Serial.println(cellVolt[2]);
Serial.println(cellVolt[3]);
Serial.println("-----");
}

int startBalance()
{
CFGReg1 = 0x00;
CFGReg2 = 0x00;
int low = lowCell();
for(int i = 1; i < TOTAL+1; i++)
{
if(i != low)
{
balanceCell(i); //set each cell individually; discharge if not lowest cell
}
}
Serial.println(CFGReg1,BIN);
Serial.println(CFGReg2,BIN);
writeReg();

}

int balanceCell(int x)
{
double diff = cellVolt[x-1] - cellVolt[lowCell()-1]; //check the difference
between current cell and lowestCell
if(x <= 8 && diff > 0.03) //discharge if difference more than 0.03V
{
CFGReg1 += 0x01<<(x-1);
}
if(x <= 8 && diff <= 0.03)
{
CFGReg1 += 0x00<<(x-1);
}
if(x > 8 && diff > 0.03)
{
CFGReg2 += 0x01<<(x-9);
}
if(x > 8 && diff <= 0.03)

```

```

{
CFG2 += 0x00<<(x-9);
}

int highCell() //find highest voltage cell value
{
    int num = 0;
    float cellTemp = 0;
    for(int i = 0; i < TOTAL;)
    {
        if(cellVolt[i] > cellTemp)
        {
            cellTemp = cellVolt[i];
            num = i+1;
        }
        i++;
    }
    return num;
}

int lowCell() //find lowest cell value
{
    int num = 0;
    float cellTemp = 5;
    for(int i = 0; i < TOTAL;)
    {
        if(cellVolt[i] < cellTemp)
        {
            cellTemp = cellVolt[i];
            num = i+1;
        }
        i++;
    }
    return num;
}

```

d. Simulation studies (*not mandatory if designed by the team*)

e. Printed circuit studies

Still in progress.

f. Production studies

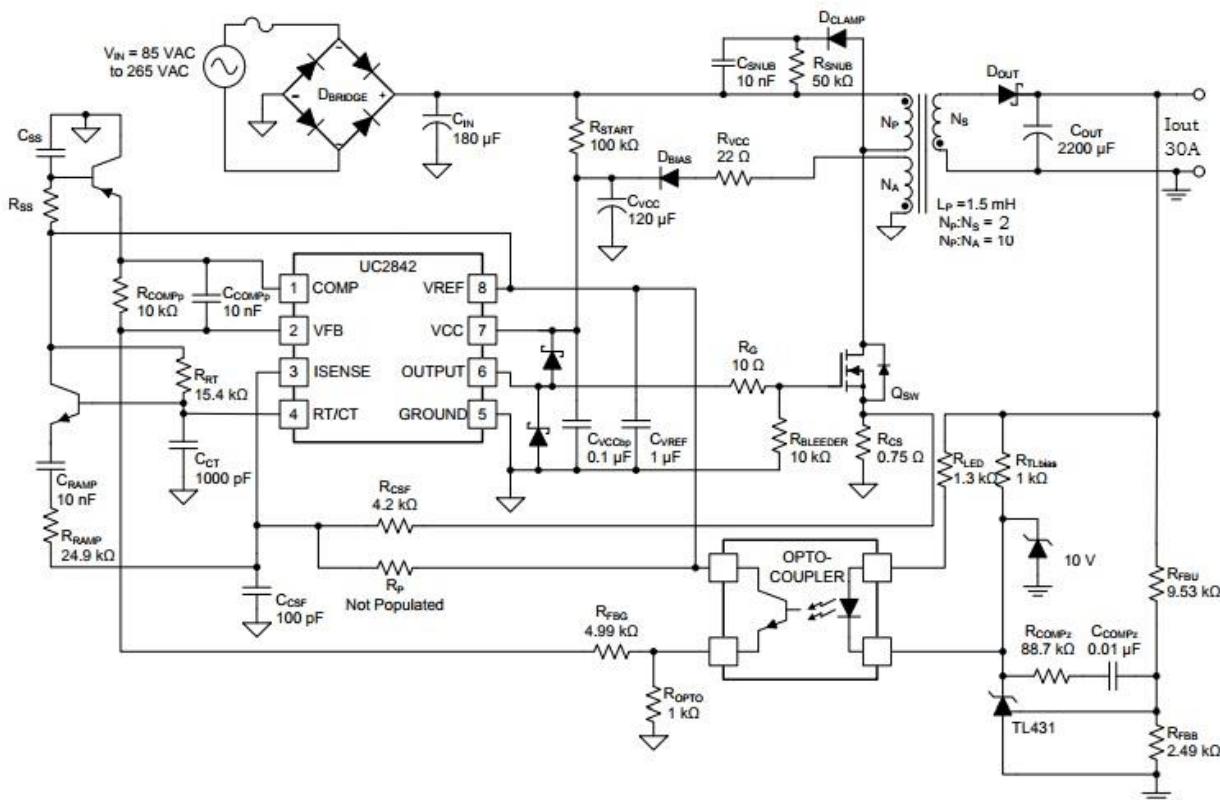
6. Embedded recharging unit (if designed by team, details mandatory; if ready-made product, please explain briefly)

All the Electromobile teams are required to prepare their embedded recharging unit themselves. Teams that make this product domestically will get a maximum of **200 points** depending on the design.

During technical inspections, testing of the embedded recharging unit will be done by measuring the current and voltage values at output terminals.

If designed by the team, detailed information should be given on the following topics:

a. Circuit design



b. Simulation studies

There are no simulation study. We trust our calculations.

c. Printed circuit studies

Still in progress.

d. Production studies

Still in progress.

7. Energy management system (EMS) (if designed by team, details mandatory; if ready-made product, please explain briefly)

*All the Hydromobile teams are required to prepare their EMS themselves. Teams that produce this product domestically will get a maximum of **300 points** depending on the design.*

A software and/or hardware that optimises power flow between the energy sources and the load can be regarded as an EMS.

Teams that develop a domestic EMS will be responsible for the following items during technical inspections:

- 1- *Demonstrating the EMS product physically,*
- 2- *Providing general information about the EMS design and operating principles,*
- 3- *Measuring and showing the current and voltage values at which the DC-DC converters operate.*

If designed by the team, detailed information should be given on the following topics.

- a. **Control algorithm**
- b. **Simulation studies**
- c. **Application studies**

8. Battery Packaging(*if designed by team, details mandatory; if ready-made product, please explain briefly*)

Teams that produce this product domestically will get a maximum of **150 points** depending on the design.

Note: All documents are included in rarfiles.Production process continues .

9. Telemetry (*if designed by team, details mandatory; if ready-made product, please explain briefly*)

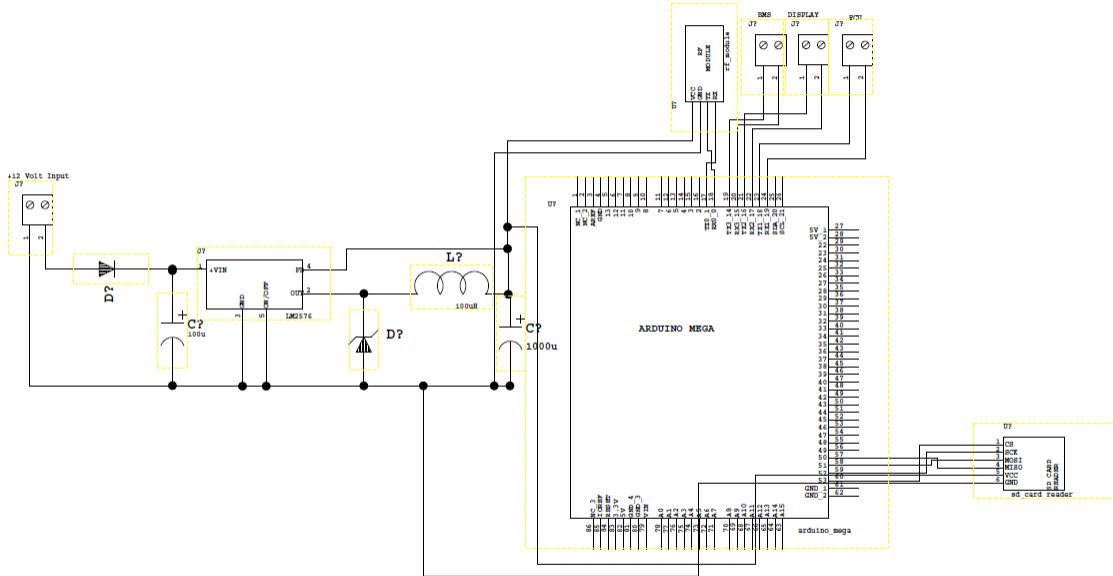
Teams that make this product domestically will get a maximum of **150 points** depending on the design.

Teams that employ a domestic telemetry design are responsible for the following items during the technical inspections:

- 1- Physically illustrating the final telemetry circuit design.
- 2- Providing details about the information transmitted by telemetry.
- 3- Providing overall information related to communication protocols used by telemetry.
- 4- Demonstration of the data transmission by telemetry from the telemetry screen.
- 5- Logging the transferred data by the telemetry system.

If designed by the team, detailed information should be given on the following topics:

a. Circuit design



We have sliced VCU unit into two parts these are SensorKit and ECU. Sensorkit consists Telemetry unit (rf unit and some codes in SensorKit codes.)

b. Communication protocols

UART via RF module.

Code:

Sensor Kit:

SensorKit is a module that collects data such as speed, battery status, battery and motor driver temperture etc. from Engine Control Unit (ECU), Battery Management System (BMS) and sends them to Telemetri module, showsnecessary informations on HMI screen and saves them for drive test.

SensorKit Codes:

```
#include <VirtualWire.h>
```

```
#include <SPI.h>
#include <SD.h>

//Serial definetions
#define ECUSerial Serial1
#define BMSSerial Serial3
#define HMISerial Serial2
#define RFSerial Serial

//Function prototypes
voidreadFromECU(String inData);
voidsaveToSD();
voidsetupSD();
voidsetupRF();
voidsentToRF();
voidupdateHMI();

//Sensor variables
int speed = 100;
String speedString = "0";

int currMotor1 = 0;
String currMotorString1= "0";

int currMotor2 = 0;
```

```
String currMotorString2= "0";
```

```
int tempMotor1 = 0;
```

```
String tempMotorString1 = "0";
```

```
int tempMotor2 = 0;
```

```
String tempMotorString2 = "0";
```

```
inttempBattery = 0;
```

```
String tempBatteryString = "0";
```

```
intbattStatus = 0;
```

```
String battStatusString = "0";
```

```
char gear = 'o';
```

```
int counter = 0;
```

```
//Data logger variables
```

```
Sd2Card sdCard;
```

```
SdVolumessdVolume;
```

```
SdFile root;
```

```
intpinCS = 53;
```

```
File logFile;
```

```
//RF Module Variables
```

```
intpinRF = 12;
```

```

void setup() {
    ECUSerial.begin(9600);
    BMSSerial.begin(9600);
    HMISerial.begin(9600);
    setupSD();
    setupRF();
}

void loop() {
    updateHMI();
    readFromECU();
    readFromBMS();
    saveToSD();
    sendToRF();
}

void setupSD(){
    if (!sdCard.init(SPI_QUARTER_SPEED, pinCS)) {
        return;
    }else {
        if(!SD.begin(pinCS)) {
            return;
    }
}

```

```

}else {

SD.begin(pinCS);

logFile = SD.open("data.csv",FILE_WRITE);

logFile.println("Motor current 1,Motor Current 2,Motor Temperature 1,tMotor Temperature
2,Speed,Gear,Battery Temperature,BatteryStatus,Time");

logFile.close();

}

}

}

```

```

void saveToSD(){

logFile = SD.open("data.csv",FILE_WRITE);

if(logFile){

logFile.print(currMotorString1);

logFile.print(",");
logFile.print(currMotorString1);

logFile.print(",");
logFile.print(tempMotorString1);

logFile.print(",");
logFile.print(tempMotorString2);

logFile.print(",");
logFile.print(speedString);

logFile.print(",");
logFile.print(gear);

logFile.print(",");
logFile.print(tempBatteryString);

```

```

logFile.print(",");
logFile.print(battStatusString);
logFile.print(",");
logFile.print(millis());
logFile.println();
logFile.close();
}

}

void sendToRF(){
    String RFMessage =
    "*"+currMotorString1+"/"+currMotorString2+"/"+tempMotorString1+"/"+tempMotorString2+"/"+speedString+
    "+tempBatteryString+"/"+battStatusString+"?";
    charRFArray[RFMessage.length()+1];
    RFMessage.toCharArray(RFArray,RFMessage.length());
    vw_send((uint8_t *)RFArray, RFMessage.length());
    vw_wait_tx();
}

void setupRF(){
    vw_set_ptt_inverted(true);
    vw_set_tx_pin(pinRF);
    vw_setup(4000);
}

void readFromECU(){

```

```

if(ECUSerial.available()){

    String inData = ECUSerial.readStringUntil('?');

    charindataArray[inData.length() + 1];

    //Serial.println(inData);

    inData.toCharArray(indataArray,inData.length());

    sscanf(indataArray,
    "*%d/%d/%d/%d/%c",&currMotor1,&currMotor2,&speed,&tempMotor1,&tempMotor2,&gear);

    //double to Str convert

    currMotorString1 = String(currMotor1);

    currMotorString2 = String(currMotor2);

    speedString = String(speed);

    tempMotorString1 = String(tempMotor1);

    tempMotorString2 = String(tempMotor2);

}

}

void updateHMI(){

if(counter == 10000){

    HMISerial.print("speedGauge.val=");

    HMISerial.print(speed);

    HMISerial.write(0xff);
}

```

```
HMISerial.write(0xff);

HMISerial.write(0xff);

HMISerial.print("motorTemp1.txt=");

HMISerial.write(0x22);

HMISerial.print(tempMotor1);

HMISerial.write(0x22);

HMISerial.write(0xff);

HMISerial.write(0xff);

HMISerial.write(0xff);

HMISerial.print("motorTemp2.txt=");

HMISerial.write(0x22);

HMISerial.print(tempMotor2);

HMISerial.write(0x22);

HMISerial.write(0xff);

HMISerial.write(0xff);

HMISerial.write(0xff);

HMISerial.print("batteryTemp.txt=");

HMISerial.write(0x22);

HMISerial.print(tempBattery);

HMISerial.write(0x22);

HMISerial.write(0xff);

HMISerial.write(0xff);

HMISerial.write(0xff);
```

```
HMISerial.print("batteryStat.txt=");  
HMISerial.write(0x22);  
HMISerial.print(battStatus);  
HMISerial.write(0x22);  
HMISerial.write(0xff);  
HMISerial.write(0xff);  
HMISerial.write(0xff);
```

```
HMISerial.print("gearText.txt=");  
HMISerial.write(0x22);  
HMISerial.print(gear);  
HMISerial.write(0x22);  
HMISerial.write(0xff);  
HMISerial.write(0xff);  
HMISerial.write(0xff);
```

```
HMISerial.print("yearNum.val=rtc0");  
HMISerial.write(0xff);  
HMISerial.write(0xff);  
HMISerial.write(0xff);
```

```
HMISerial.print("yearNum.val=rtc0");  
HMISerial.write(0xff);  
HMISerial.write(0xff);  
HMISerial.write(0xff);
```

```
HMISerial.print("monthNum.val=rtc1");
```

```
HMISerial.write(0xff);
```

```
HMISerial.write(0xff);
```

```
HMISerial.write(0xff);
```

```
HMISerial.print("dayNum.val=rtc2");
```

```
HMISerial.write(0xff);
```

```
HMISerial.write(0xff);
```

```
HMISerial.write(0xff);
```

```
HMISerial.print("hourNum.val=rtc3");
```

```
HMISerial.write(0xff);
```

```
HMISerial.write(0xff);
```

```
HMISerial.write(0xff);
```

```
HMISerial.print("minuteNum.val=rtc4");
```

```
HMISerial.write(0xff);
```

```
HMISerial.write(0xff);
```

```
HMISerial.write(0xff);
```

```
HMISerial.print("secondNum.val=rtc5");
```

```
HMISerial.write(0xff);
```

```
HMISerial.write(0xff);
```

```
HMISerial.write(0xff);
```

```
counter = 0;
```

```

    }

counter++;

}

voidreadFromBMS(){

String inData = BMSSerial.readStringUntil('?');

charindataArray[inData.length()+1];

inData.toCharArray(indataArray,inData.length());

sscanf(indataArray,"%d",&battStatus);

battStatusString = String(battStatus);

}

```

Note: All codes except Motor Driver had written in Arduino IDE, Motor Driver Codes had written in MikroC.

c. Printed circuit studies

Still in progress.

d. Production studies

Still in progress.

Note: All documents are included in rarfiles.Production process continues . Calculations are predominantly theoretical

10. Electronic differential application (*if designed by team, details mandatory; if ready-made product, please explain briefly*)

*Teams that produce this product domestically will get a maximum of **150 points** depending on the design.*

Our electronic differential unit is working by using an encoder which is connected to steering wheel and a function which is in ECU(a part of VCU) code. Simply according the steering wheel angle we have created a difference between two motors.

11. Vehicle Control Unit (VCU)(*if designed by team, details mandatory; if ready-made product, please explain briefly*)

Teams that produce this component domestically will get a maximum of **300 points** depending on the design.

Teams can get points according to functions in VCU. As described in rules, following items are the main functions that teams can develop for their VCU. Achieving one function will be sufficient to be encountered as domestic part. Teams get more point if they have more properly working functions.

1. Motor Torque Control
2. Regenerative braking optimization
3. Vehicle energy management system
4. Management of vehicle communication system
5. Diagnostic
6. Monitor vehicle condition and warn the user etc.

Our VCU is sliced into two parts. ECU and SensorKit. ECU manages the motor drivers according to steering wheel angle and throttle pedal value. SensorKit collects all the data in modules(tachometers,temp sensors,current sensors..etc.) and sends to us via RF module. Also, SensorKit saves these values into a SD card and displays these values on Dashboard Display.

ECU code:

Engine Control Unit:

Engine Control Unit is a module that consists electronic differential system, PID controller. This module controls the motor drivers according to throttle pedal value and steering wheel angle.

Engine Control Unit (ECU) Codes:

```
#include <PID_v1.h>
#include <SoftwareSerial.h>

//Serial definitions
#define tachometerSerial1 Serial1
#define tachometerSerial2 Serial2
#define motorDriverSerial1 Serial3
#define motorDriverSerial2 Serial
```

```
SoftwareSerial sensorKitSerial(52,53);
```

```
//Pin definitions
#define pedalPin1 0
#define pedalPin2 1
#define gearPin 51
#define encoderPin 3
```

```
//Constants
#define gasPedalTreshold 100
#define tresholdPID 50
#define consKp 1
#define consKi 1
#define consKd 1
```

```
#define aggKp 5
```

```
#define aggKi 5
```

```
#define aggKd 5
```

```
//Variable definitions
```

```
int currMotor1 = 0;
```

```
int currMotor2 = 0;
```

```
int tempMotor1 = 0;
```

```
int tempMotor2 = 0;
```

```
intbattStat = 0;
```

```
int speed = 0;
```

```
char gear = 'D';
```

```
intgasPedal = 0;
```

```
String gasPedalString = "0";
```

```
double rpm1 = 0;
```

```
double rpm2 = 0;
```

```
int rpm1_int = 0;
```

```
int rpm2_int = 0;
```

```

double desRpm1 = 0;

double desRpm2 = 0;

double pwm1 = 0;

double pwm2 = 0;

int angle = 0;

int counter = 0;

//PID cons

PID motorPID1(&rpm1, &pwm1, &desRpm1, consKp, consKi, consKd, DIRECT);

PID motorPID2(&rpm2, &pwm2, &desRpm2, consKp, consKi, consKd, DIRECT);

```

```

void setup() {

sensorKitSerial.begin(9600);

tachometerSerial1.begin(9600);

tachometerSerial2.begin(9600);

motorDriverSerial1.begin(9600);

motorDriverSerial2.begin(9600);

motorPID1.SetMode(AUTOMATIC);

motorPID2.SetMode(AUTOMATIC);

pinMode(gearPin,INPUT);

}

```

```

void loop() {

readGear();

```

```
readGasPedal();  
readTachometer();  
readEncoder();  
calculateDesiredRPM();  
computePID();  
readMotorCurrent();  
sendPWM();  
sendToSensorKit();  
  
}  
  
}
```

```
void readGasPedal(){  
int gas1 = analogRead(pedalPin1);  
int gas2 = analogRead(pedalPin2);  
  
if(abs(gas1-gas2) > gasPedalThreshold){  
gasPedal = min(gas1,gas2);  
}  
else{  
gasPedal= gas1;  
//gasPedal = gasPedal>> 2;  
}  
  
if(gear == 'D')  
gasPedal+=1000;  
else if(gear == 'R')  
gasPedal+=2000;
```

```

gasPedalString = String(gasPedal);

}

voidreadTachometer(){

if(tachometerSerial1.available()){

    String inData1 = tachometerSerial1.readStringUntil('?');

    char indataArray1[inData1.length() + 1];

    inData1.toCharArray(indataArray1,inData1.length());

    sscanf(indataArray1, "*%d",&rpm1_int);

    rpm1 = rpm1_int;

}

if(tachometerSerial2.available()){

    String inData2 = tachometerSerial2.readStringUntil('?');

    char indataArray2[inData2.length() + 1];

    inData2.toCharArray(indataArray2,inData2.length());

    sscanf(indataArray2, "*%d",&rpm2_int);

    rpm2 = rpm2_int;

    speed = (rpm1+rpm2)/2;

}

}

voidreadMotorCurrent(){

if(motorDriverSerial1.available()){

    String inData1 = motorDriverSerial1.readStringUntil('?');

    char indataArray1[inData1.length() + 1];

```

```

inData1.toCharArray(indataArray1,inData1.length());
sscanf(indataArray1, "*%d",&currMotor1);

}

if(motorDriverSerial2.available()){

    String inData2 = motorDriverSerial2.readStringUntil('?');

    char indataArray2[inData2.length()+1];

    inData2.toCharArray(indataArray2,inData2.length());

    sscanf(indataArray2, "*%d",&currMotor2);

}

}

voidsendPWM(){

if(motorDriverSerial1.available()){

Serial.println("1");

String strPWM1 = String((int)pwm1);

motorDriverSerial1.print("*"+strPWM1+"?");

}

if(motorDriverSerial2.available()){

String strPWM2 = String((int)pwm2);

motorDriverSerial1.print("*"+strPWM2+"?");

}

}

voidsendToSensorKit(){

String currMotorString1 = String(currMotor1);

String currMotorString2 = String(currMotor2);

```

```

String speedString = String((int)speed);
Serial.println(speed);

String tempMotorString1 = String(tempMotor1);

String tempMotorString2 = String(tempMotor2);

String outData =
"*"+currMotorString1+"/"+currMotorString2+"/"+speedString+"/"+tempMotorString1+"/"+tempMotorString2+"/"+gear+"?";

sensorKitSerial.print(outData);

Serial.println(outData);

}

voidreadGear(){

if(digitalRead(gearPin) == LOW)

gear = 'R';

else

gear = 'D';

}

voidreadEncoder(){

angle = analogRead(encoderPin);

angle = map(angle , 0, 1024, 0, 360);

}

voidcalculateDesiredRPM(){

//calculations according to formula

}

```

```

voidcomputePID(){

double gap1 = abs(rpm1-desRpm1);

if(gap1 >thresholdPID)

motorPID1.SetTunings(aggKp,aggKi,aggKd);

else

motorPID1.SetTunings(aggKp,aggKi,aggKd);

double gap2 = abs(rpm2-desRpm2);

if(gap2 >thresholdPID)

motorPID2.SetTunings(aggKp,aggKi,aggKd);

else

motorPID2.SetTunings(aggKp,aggKi,aggKd);

motorPID1.Compute();

motorPID2.Compute();

}

```

Note: All codes except Motor Driver had written in Arduino IDE, Motor Driver Codes had written in MikroC.

SensorKit Code:

Sensor Kit:

SensorKit is a module that collects data such as speed, battery status, battery and motor driver temperture etc. from Engine Control Unit (ECU), Battery Management System (BMS) and sends them to Telemetri module, showsnecessary informations on HMI screen and saves them for drive test.

SensorKit Codes:

```
#include <VirtualWire.h>
#include <SPI.h>
#include <SD.h>

//Serial definetions
#define ECUSerial Serial1
#define BMSSerial Serial3
#define HMISerial Serial2
#define RFSerial Serial

//Function prototypes
voidreadFromECU(String inData);
voidsaveToSD();
voidsetupSD();
voidsetupRF();
voidsentToRF();
voidupdateHMI();

//Sensor variables
```

```
int speed = 100;  
String speedString = "0";  
  
int currMotor1 = 0;  
String currMotorString1= "0";  
  
int currMotor2 = 0;  
String currMotorString2= "0";  
  
int tempMotor1 = 0;  
String tempMotorString1 = "0";  
  
int tempMotor2 = 0;  
String tempMotorString2 = "0";  
  
inttempBattery = 0;  
String tempBatteryString = "0";  
  
intbattStatus = 0;  
String battStatusString = "0";  
  
char gear = 'o';  
  
int counter = 0;  
//Data logger variables  
Sd2Card sdCard;
```

```
SdVolumessdVolume;
```

```
SdFile root;
```

```
intpinCS = 53;
```

```
File logFile;
```

```
//RF Module Variables
```

```
intpinRF = 12;
```

```
void setup() {
```

```
ECUSerial.begin(9600);
```

```
BMSSerial.begin(9600);
```

```
HMIserial.begin(9600);
```

```
setupSD();
```

```
setupRF();
```

```
}
```

```
void loop() {
```

```
updateHMI();
```

```
readFromECU();
```

```
readFromBMS();
```

```
saveToSD();
```

```
sendToRF();
```

```
}
```

```

voidsetupSD(){

if (!sdCard.init(SPI_QUARTER_SPEED, pinCS)) {

return;

}else {

if(!SD.begin(pinCS)) {

return;

}else {

SD.begin(pinCS);

logFile = SD.open("data.csv",FILE_WRITE);

logFile.println("Motor current 1,Motor Current 2,Motor Temperature 1,tMotor Temperature
2,Speed,Gear,Battery Temperature,BatteryStatus,Time");

logFile.close();

}

}

}

}

```

```

voidsaveToSD(){

logFile = SD.open("data.csv",FILE_WRITE);

if(logFile){

logFile.print(currMotorString1);

logFile.print(",");
logFile.print(currMotorString1);

logFile.print(",");
logFile.print(tempMotorString1);

logFile.print(",");

```

```

logFile.print(tempMotorString2);

logFile.print(",");
logFile.print(speedString);
logFile.print(",");
logFile.print(gear);
logFile.print(",");
logFile.print(tempBatteryString);
logFile.print(",");
logFile.print(battStatusString);
logFile.print(",");
logFile.print(millis());
logFile.println();
logFile.close();

}

}

void sendToRF(){
    String RFMessage =
"**"+currMotorString1+"/"+currMotorString2+"/"+tempMotorString1+"/"+tempMotorString2+"/"+speedString+
"/"+tempBatteryString+"/"+battStatusString+"?";
charRFArray[RFMessage.length()+1];
RFMessage.toCharArray(RFArray,RFMessage.length());
vw_send((uint8_t *)RFArray, RFMessage.length());
vw_wait_tx();
}

void setupRF(){

```

```

vw_set_ptt_inverted(true);
vw_set_tx_pin(pinRF);
vw_setup(4000);
}

voidreadFromECU(){

if(ECUSerial.available()){

    String inData = ECUSerial.readStringUntil('?');

    charindataArray[inData.length()+1];
    //Serial.println(inData);

    inData.toCharArray(indataArray,inData.length());

    sscanf(indataArray,
    "*%d/%d/%d/%d/%c",&currMotor1,&currMotor2,&speed,&tempMotor1,&tempMotor2,&gear);

    //double to Str convert

    currMotorString1 = String(currMotor1);
    currMotorString2 = String(currMotor2);
    speedString = String(speed);
    tempMotorString1 = String(tempMotor1);
    tempMotorString2 = String(tempMotor2);

}

}

```

```
void updateHMI(){  
if(counter == 10000){  
  
HMISerial.print("speedGauge.val=");  
HMISerial.print(speed);  
HMISerial.write(0xff);  
HMISerial.write(0xff);  
HMISerial.write(0xff);  
  
HMISerial.print("motorTemp1.txt=");  
HMISerial.write(0x22);  
HMISerial.print(tempMotor1);  
HMISerial.write(0x22);  
HMISerial.write(0xff);  
HMISerial.write(0xff);  
HMISerial.write(0xff);  
  
HMISerial.print("motorTemp2.txt=");  
HMISerial.write(0x22);  
HMISerial.print(tempMotor2);  
HMISerial.write(0x22);  
HMISerial.write(0xff);  
HMISerial.write(0xff);  
HMISerial.write(0xff);
```

```
HMISerial.print("batteryTemp.txt=");  
HMISerial.write(0x22);  
HMISerial.print(tempBattery);  
HMISerial.write(0x22);  
HMISerial.write(0xff);  
HMISerial.write(0xff);  
HMISerial.write(0xff);
```

```
HMISerial.print("batteryStat.txt=");  
HMISerial.write(0x22);  
HMISerial.print(battStatus);  
HMISerial.write(0x22);  
HMISerial.write(0xff);  
HMISerial.write(0xff);  
HMISerial.write(0xff);
```

```
HMISerial.print("gearText.txt=");  
HMISerial.write(0x22);  
HMISerial.print(gear);  
HMISerial.write(0x22);  
HMISerial.write(0xff);  
HMISerial.write(0xff);  
HMISerial.write(0xff);
```

```
HMISerial.print("yearNum.val=rtc0");  
HMISerial.write(0xff);
```

```
HMISerial.write(0xff);  
HMISerial.write(0xff);
```

```
HMISerial.print("yearNum.val=rtc0");
```

```
HMISerial.write(0xff);  
HMISerial.write(0xff);  
HMISerial.write(0xff);
```

```
HMISerial.print("monthNum.val=rtc1");
```

```
HMISerial.write(0xff);  
HMISerial.write(0xff);  
HMISerial.write(0xff);
```

```
HMISerial.print("dayNum.val=rtc2");
```

```
HMISerial.write(0xff);  
HMISerial.write(0xff);  
HMISerial.write(0xff);
```

```
HMISerial.print("hourNum.val=rtc3");
```

```
HMISerial.write(0xff);  
HMISerial.write(0xff);  
HMISerial.write(0xff);
```

```
HMISerial.print("minuteNum.val=rtc4");
```

```
HMISerial.write(0xff);  
HMISerial.write(0xff);
```

```

HMISerial.write(0xff);

HMISerial.print("secondNum.val=rtc5");

HMISerial.write(0xff);

HMISerial.write(0xff);

HMISerial.write(0xff);

counter = 0;

}

counter++;

}


```

```

voidreadFromBMS(){

String inData = BMSSerial.readStringUntil('?');

charindataArray[inData.length()+1];

inData.toCharArray(indataArray,inData.length());

sscanf(indataArray,"*%d?",&battStatus);

battStatusString = String(battStatus);

}

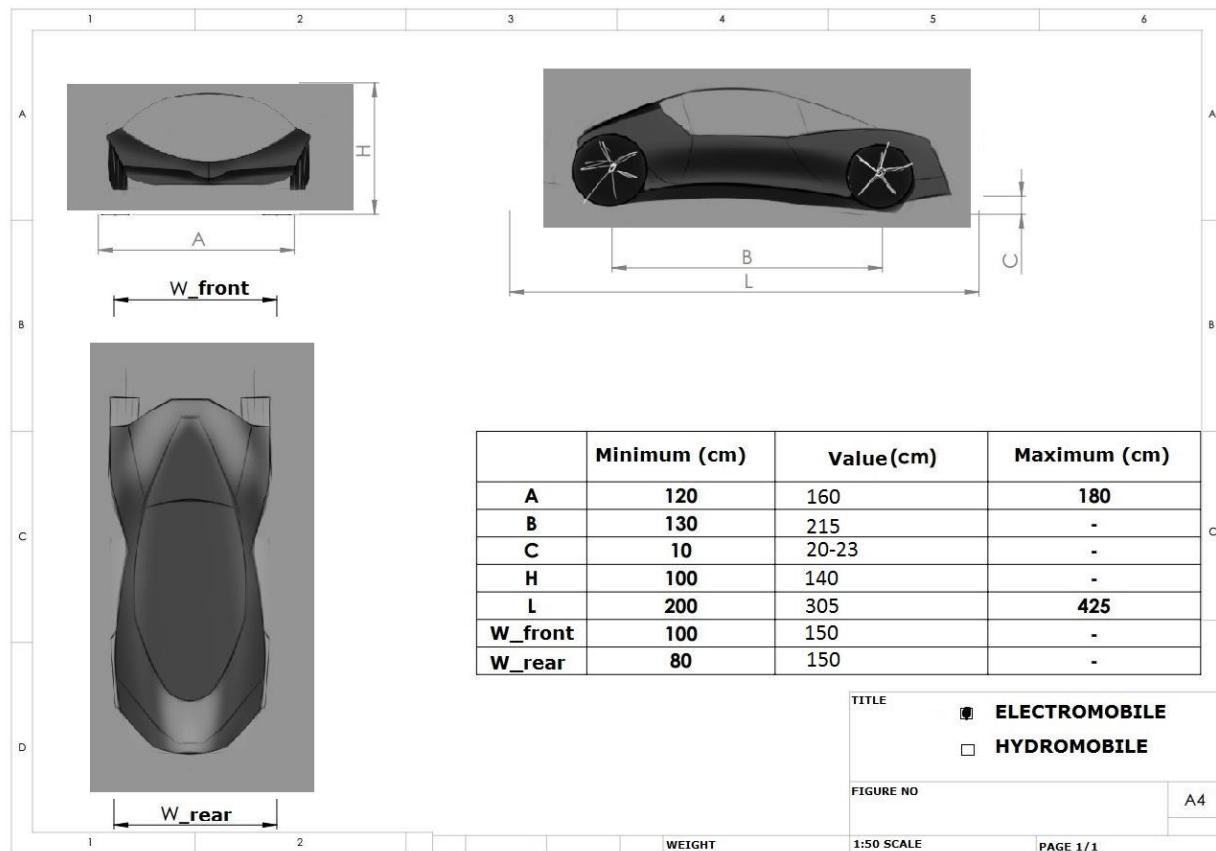

```

Note: All codes except Motor Driver had written in Arduino IDE, Motor Driver Codes had written in MikroC.

12. Mechanical details (mandatory)

Teams that provide sufficient information will get a maximum of **150 points** depending on the design.

a. Technical drawings



Like some other mechanical parts, some other parts of our chassis are being produced while this report has send. As we decided to make some changes on our chassis design, new parts have been produced again. Our team carefully follows the production procedure to prepare more effective design. And this part includes some process we made so far. Chassis has been designed.

Our workings about chassis are still going on, after the production stage we produce more decisive and certain results. In this week our chassis production will be over. Our hand-drawings were turned into programmed-versions.

b. Strength analysis

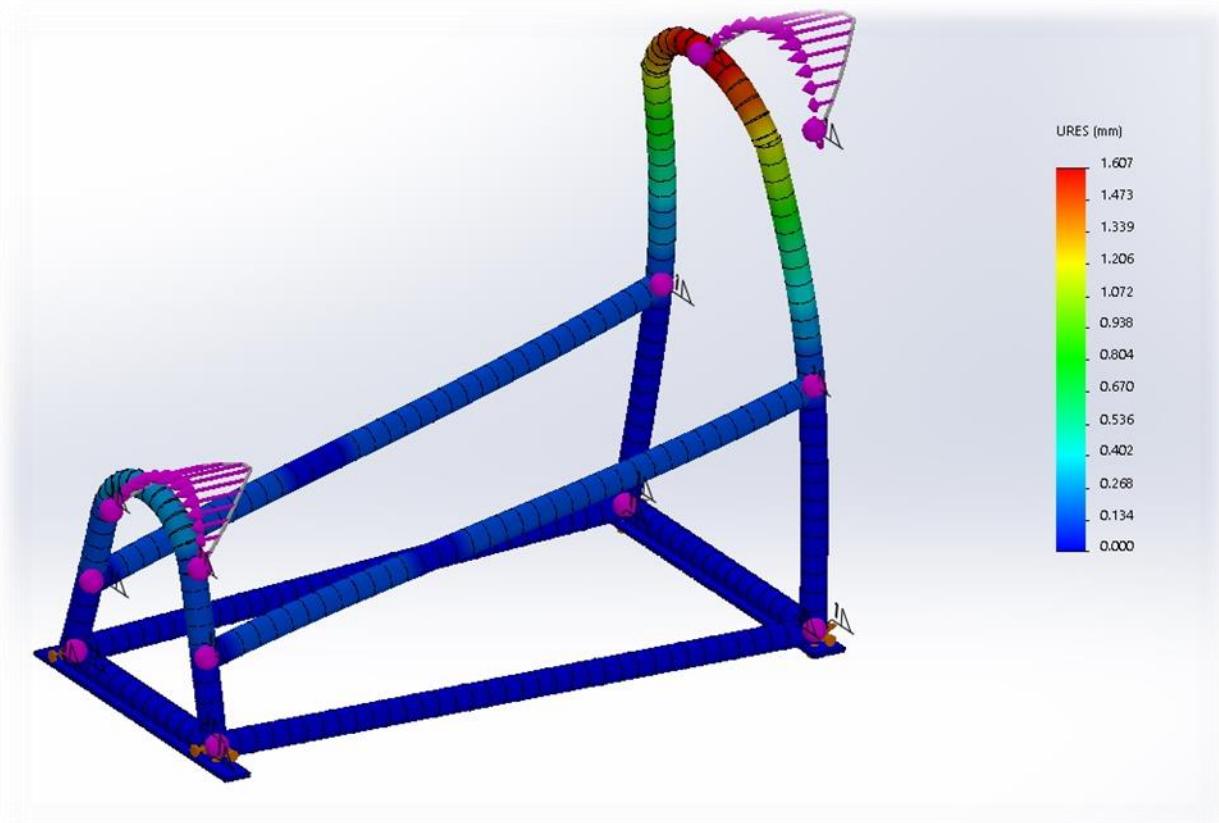


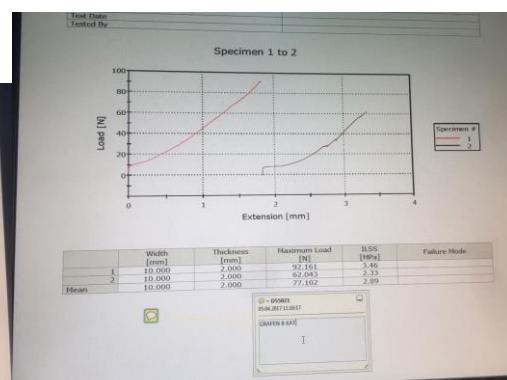
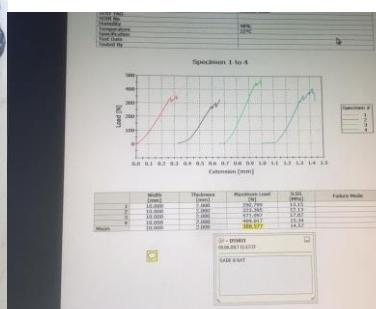
Figure 1: Deflection Analysis of Rollbar

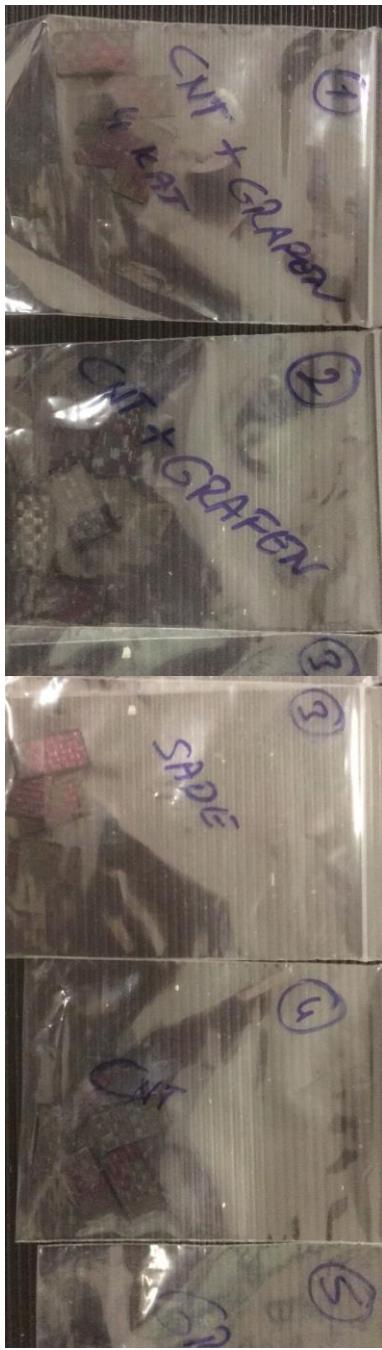
Rollbar has designed by the data of driver's seating position. In case of a crash, by design of the Rollbar, driver can be protected against smashing.

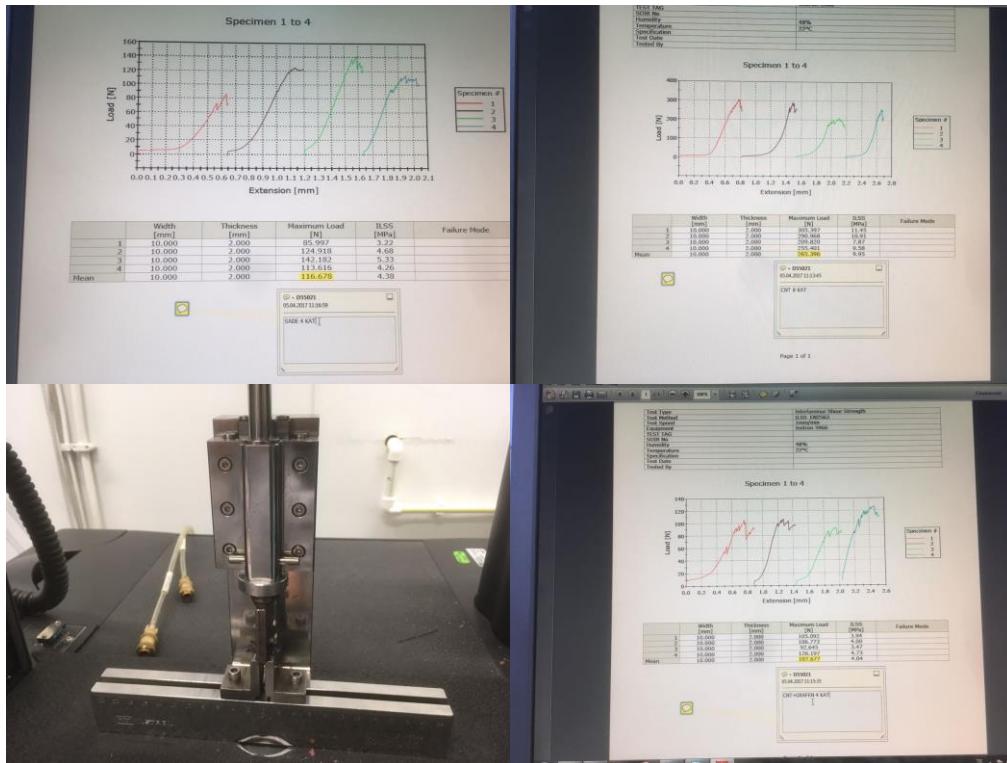
c. Outer shell production

Now our tests about outer shell is continuing. If our test results are successful, we are going to create our own outer shell. For adding details to our model; firstly the wooden model of chassis has been designed. This chassis can be seen from the photos and the preparation of original chassis is in progress. It will be ready in five days.

d. Outer shell production







e. Energy consumption and maximum inclination account

- Taking into account the front area, wheel friction, and internal losses of vehicles, the teams must calculate the energy consumption necessary to complete a 2000-m flat track at 60 km/h.

The designed electric vehicle consumes 3333,3 joule for 2000-m flat track at 60 km/h. When it is calculated for one meter, the car consumes 27,8 watt.

$$F_D = \frac{1}{2} \rho v^2 C_d A$$

$$F_D = \frac{1}{2} 1,2041x(16,67)^2 0,42x0,019 = 1,335078 \text{ N}$$

$$1,335 \times 2000\text{m} = 2680 \text{ J}$$

$2680 \text{ J} \times 100/90$ (motor's efficiency) + (some other inner frictions) = **3300 J = Total Consumption**

T (sec) >> 2000m @60km/h =>> **119 seconds**

$3300 \text{ J} / 119 \text{ s} = \mathbf{28W (@2000m)}$

f. Cost calculation

A bill of materials and the cost of the vehicle must be presented in detail in a table.

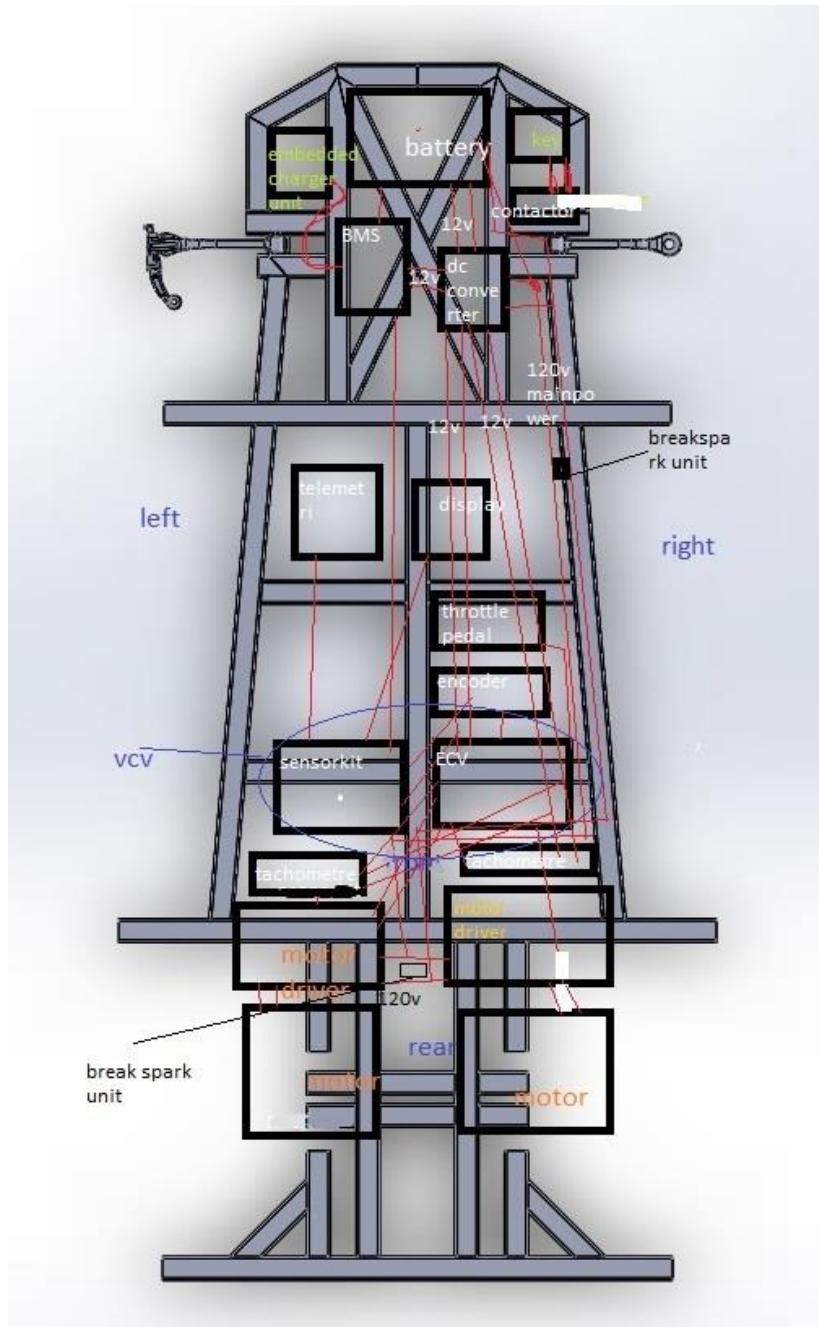
PRODUCT	PRICE	Continue	Finish	TOTAL
coil winding	3000 TL	X		
Cnc	4000 TL		X	
Front system	2000 TL		X	
Rear system	2000 TL	X		
Battery cell	13500 TL	X		
Components of motor	8000 TL		X	
				28900 TL

PRODUCT	PRICE	IN TESTING	TOTAL
OUTER SHELL	7500 TL	X	
CHASSIS	5000 TL	X	
			12500TL
		GENERAL TOTAL	41400 TL

13. Vehicle electric scheme

Teams that provide sufficient information will get a maximum of **50 points** depending on the design.

It is mandatory to provide an A4 drawing (21 × 29.7 cm) illustrating all power circuits of the electrical equipment of the vehicle. The drawing must include the battery, fuses, circuit breakers, power control buttons, capacitors, motor control tools, motor or motors, charge unit, connection cables, and so on. A second drawing must also clearly illustrate the components within the vehicle as viewed from above.



14. Unique design by team (*optional*)

*Team designs regarded as original can receive a maximum of **200 points**.*

Note:

All pcb design are our creation. We divide VCU into two parts one of them is ECU and the other one is SensorKit. The reason of this to decrease the error percentage and speeding up the system.

Scoring

			Electromobile		Hydromobile	
			Domestic	Ready-made	Domestic	Ready-made
1	Vehicle specifications table	Mandatory	100	-	100	-
2	Motor	If designed by team, details mandatory; if ready-made product, please explain briefly	400	-	400	-
3	Motor driver	If designed by team, details mandatory; if ready-made product, please explain briefly	300	-	300	-
4	Battery management system	If designed by team, details mandatory; if ready-made product, please explain briefly	200	50	200	50
5	Embedded recharging unit	If designed by team, details mandatory; if ready-made product, please explain briefly	200	50	200	50
6	Energy management system	If designed by team, details mandatory; if ready-made product, please explain briefly	-	-	300	50
7	Battery packaging	If designed by team, details mandatory; if ready-made product, please explain briefly	150	-	150	-
8	Telemetry	If designed by team, details mandatory; if ready-made product, please explain briefly	150	50	150	50
9	Vehicle control unit	If designed by team, details mandatory; if ready-made product, please explain briefly	300	-	300	-
10	Electronic differential application	If designed by team, details mandatory; if ready-made product, please explain briefly	150	50	150	50
11	Mechanical details	Mandatory	150	150	150	150
12	Fuel cell	If designed by team, details mandatory; if ready-made product, please explain briefly	-	-	500	-
13	Fuel cell control system	If designed by team, details mandatory; if ready-made product, please explain briefly	-	-	300	-
14	Electric scheme	Mandatory	50	-	50	-
15	Original design	Not mandatory; fill in if applicable	200	-	200	-
TOTAL			2350		3450	

Table 1: Table of summary score