

**Ege Elektromobil**

**BrakeAndDirection**

**Komponent**

**Gereksinim Spesifikasyon**

**(SRS) Dökümanı**

Atakan Ertekin

*http://www.egeelektromobiltakimi.com/*  
2025

|  |
| --- |
| **Revizyonlar** |

| Versiyon | Yazar(lar) | Versiyon Açıklaması | Tamamlanma  Tarihi |
| --- | --- | --- | --- |
| 1.0.0-0 | Atakan Ertekin | BrakeAndDirectionManager komponentinin gereksinim dökümanı oluşturulmuştur. Temel gereksinimler eklenmiştir. | 10 Şubat 2025  10.02.2025 |
| 1.0.1-0 | Ahmet Vedat  Özkılıç | Düzenleme yapılmıştı. Basit hatalar düzeltilmiştir. | 11 Şubat 2025  11.02.2025 |

|  |
| --- |
| **Gözden geçiren & Onaylayan** |

Gereksinim dökümanları onaylanan geçmiş versiyonlar

| Onaylayan(lar) | Onaylanan  Versiyon | İmza | Tarih |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |

Gereksinim dökümanlarının gözdengeçirenler

| Gözden geçiren(ler) | Onaylanan  Gözden geçirme | İmza | Tarih |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

|  |
| --- |
| **İÇİNDEKİLER** |

[1. Giriş 3](#_Toc244519333)

[2. Genel Açıklamalar 3](#_Toc244519334)

[3. Fonksiyonel Gereksinimler 3](#_Toc244519335)

[4. Arayüz Sınıf Tanımlamaları](#_Toc244519336) 5

[5. Arayüz Tanımlamaları](#_Toc244519342) 5

[6. Test Senaryoları 5](#_Toc244519354)

[7. Görsel Modeller ve Diyagramlar 6](#_Toc244519355)

[7.1 Temel Flowchart Diyagramları 6](#_Toc244519356)

[7.2 Sıralama Diyagramları 6](#_Toc244519357)

[8. Güncelleme Kayıt Defteri 7](#_Toc244519358)

[9. Güncellenen Bütçe Kaydı 7](#_Toc244519359)

[10. Ekler 8](#_Toc244519360)

1. Giriş

* 1. **Giriş**

Bu doküman, **BrakeAndDirectionManager** komponentinin fonksiyonlarını, gereksinimlerini, kullanılan arayüzleri ve test senaryolarını açıklamaktadır. Komponent, fren ve yön bilgilerini okuyarak yönetir ve ilgili arayüzler aracılığıyla sistemin diğer bileşenlerine iletir.

* 1. **Dökümanın kapsamı**

Bu döküman, BrakeAndDirectionManager’ın işleyişini tanımlayan işlevsel gereksinimleri, kullanılan arayüzleri ve doğrulama için test senaryolarını içermektedir. Sistem içindeki konumu ve diğer bileşenlerle olan etkileşimleri açıklanmaktadır.

* 1. **Genel Bakış**

BrakeAndDirectionManager, aracın fren ve yön kontrolünü sağlayan bir yazılım bileşenidir. Bu bileşen, **MotorDriverControlManager** üzerinden gelen fren ve yön bilgilerini işler, sistemde uygun güncellemeleri yapar ve **IBrakeStatus** ve **IDirectionStatus** arayüzleri aracılığıyla güncel bilgileri yayınlar.

## 2. Genel Açıklamalar

**2.1 Temel Fonksiyon tanımlaması**

BrakeAndDirectionManager’ın temel fonksiyonları:

* **Fren durumu okuma ve yönetme:** **MotorDriverControlManager**’dan gelen fren durumunu değerlendirerek sistemde güncellemeler yapar.
* **Yön durumu okuma ve yönetme:** **MotorDriverControlManager**’dan gelen yön bilgisini değerlendirerek güncel yön bilgisini belirler.
* **Fren ve yön durumlarını yayınlama:** **IBrakeStatus** ve **IDirectionStatus** arayüzleri üzerinden güncellenmiş durum bilgilerini paylaşır.
* **Hız yönetimi ile entegrasyon:** **SpeedControlManager\_ruBrakeUpdate()** çağrılarak fren durumundaki hız yönetimi sağlanır.

**2.2 Fonksiyonel Hedefler**

BrakeAndDirectionManager, aracın fren ve yön durumlarını periyodik olarak okuyup güncelleyerek, fren değişikliklerini algılar ve uygun aksiyonları gerçekleştirir. Yön değişikliklerini yalnızca durma veya geri vites senaryolarında uygular, hız kontrol yönetimiyle entegre çalışarak fren durumunu günceller ve hata senaryolarında uygun hata yönetimi mekanizmalarını sağlar.

## 3. Fonksiyonel Gereksinimler

Bu bölümde, **BrakeAndDirectionManager** komponentinin işlevlerini yerine getirebilmesi için gerekli olan gereksinimler detaylandırılmıştır. Gereksinimler, komponentin temel yapı taşlarını oluşturan **base class** tanımına dayanarak belirlenmiştir.

1. BrakeAndDirectionManager, aracın **fren ve yön yönetimini sağlayan** temel bir bileşendir. Bu bileşen, gelen **fren ve yön bilgilerini işler**, gerekli güncellemeleri yapar ve **IBrakeStatus** ile **IDirectionStatus** arayüzleri aracılığıyla sistemin diğer bileşenlerine bilgi sağlar.  
   1. Komponent, **başlangıç değerleriyle başlatılmalı** ve çalışma sürecinde bu değerleri yönetmelidir.
   2. Tüm **getter/setter fonksiyonları** ve **arayüz bağlantıları** eksiksiz çalışmalıdır.
   3. Güncellenen yön ve fren bilgileri **ilgili arayüzlere düzenli olarak yazılmalıdır**.
   4. **Hata yönetimi mekanizması** aktif olmalıdır.
2. BrakeAndDirectionManager, sistem başlatıldığında **varsayılan olarak ileri yön (DIRECTION\_FORWARD) ve fren devre dışı (BRAKE\_DISABLE) durumunda başlatılmalıdır**.
   1. Başlangıç değerleri aşağıdaki gibi olmalıdır:
      1. direction = DIRECTION\_FORWARD
      2. brake = BRAKE\_DISABLE
   2. Yeni değerler validasyon kurallarına göre güncellenmelidir.
   3. Başlangıçta, bu değerler **IBrakeStatus** ve **IDirectionStatus** arayüzlerine yazılmalıdır.
3. BrakeAndDirectionManager, **MotorDriverControlManager** bileşeninden **IBrakeAndDirection** arayüzü üzerinden **fren ve yön durumlarını belirli periyotlarla okuma** işlemi gerçekleştirmelidir.
   1. **readBrake()** fonksiyonu çağrılarak **fren durumu**
   2. **readDirection()** fonksiyonu çağrılarak **yön durumu** okunmalıdır.
   3. **Periyodik bir döngü içinde** bu kontroller yapılmalı ve **değişiklikler tespit edilmelidir**.
   4. Eğer bir değişiklik tespit edilirse, ilgili setter fonksiyonları kullanılarak güncelleme yapılmalıdır.
4. MotorDriverControlManager, fren durumu **BRAKE\_UNKNOWN** olarak raporladığında, bu durum bir hata olarak değerlendirilmelidir.
   1. Eğer **readBrake()** fonksiyonu **BRAKE\_UNKNOWN** döndürürse:
      1. Hata durumu **retVal** değişkenine kaydedilmelidir.
      2. **Error handling mekanizması devreye girmeli** ve hata kodu saklanmalıdır.
      3. **Sistemsel hata bildirimi** sağlanmalıdır.
5. MotorDriverControlManager, yön durumu **DIRECTION\_UNKNOWN** olarak raporladığında, bu durum bir hata olarak değerlendirilmelidir.
   1. Eğer **readDirection()** fonksiyonu **DIRECTION\_UNKNOWN** döndürürse:
      1. Hata durumu **retVal** değişkenine kaydedilmelidir.
      2. **Sistemsel hata bildirimi** sağlanmalıdır.
6. Aracın yön değişikliği **yalnızca fren aktifken (BRAKE\_ENABLE) yapılabilir**.
   1. Eğer **brake == BRAKE\_DISABLE** ise, yön değişikliğine **izin verilmemelidir**.
   2. Eğer **brake == BRAKE\_ENABLE** ise, yön değişikliğine izin verilmelidir.
   3. Yön değişikliği yapıldığında, yeni değer **IDirectionStatus** arayüzüne yazılmalıdır.
7. MotorDriverControlManager’dan alınan ve güncellenen fren ve yön bilgileri, **IBrakeStatus** ve **IDirectionStatus** arayüzlerine yazılmalıdır.
   1. Güncellenen **fren durumu**, **IBrakeStatus** arayüzüne yazılmalıdır.
   2. Güncellenen **yön durumu**, **IDirectionStatus** arayüzüne yazılmalıdır.
8. Fren durumundaki herhangi bir değişiklik, **SpeedControlManager\_ruBrakeUpdate()** fonksiyonuna bildirilmelidir. Frenleme durumu, hız kontrolünü doğrudan etkileyen bir faktördür. Fren etkinleştirildiğinde veya devre dışı bırakıldığında, hız yönetimi bunu dikkate alarak uygun değişiklikleri yapmalıdır.
   1. **SpeedControlManager\_ruBrakeUpdate()** çağrılmalıdır.
   2. Güncellenmiş hız durumu ilgili bileşenlere iletilmelidir.

## 4. Arayüz Sınıf Tanımlamaları

|  |  |
| --- | --- |
| **Abstract Sınıf** | **Tanım** |
| Abstract\_MotorDriver | Motorun hız, fren ve yön yönetimi için gerekli olan arayüzleri içerir. |

## 5. Arayüz Tanımlamaları

### Abstract\_MotorDriver

|  |  |
| --- | --- |
| **Arayüz** | **Tanım** |
| IBrakeAndDirection | Fren ve yön sinyallerinin kontrolünü sağlar. |
| IBrakeStatus | Fren durumunun yönetimini tanımlar. |
| IDirectionStatus | Motor yön bilgilerini okuma ve kontrol işlevlerini sağlar. |
| IMotorDriveError | Motor sürücü hatalarının tanımlanması ve raporlanması için kullanılır. |

## 6. Test Senaryoları

**Senaryo A: Direction SIL TEST**

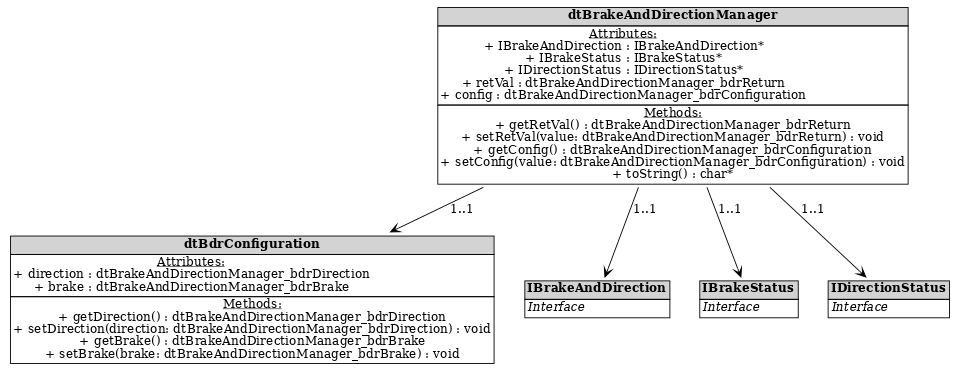
<Gerçekleşebilecek test senaryo açıklaması>

**Senaryo B: Brake SIL TEST**

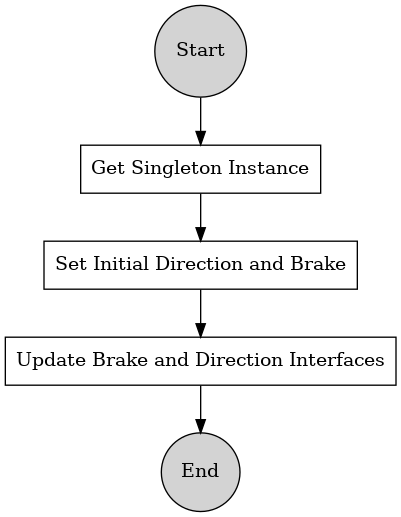
<Gerçekleşebilecek test senaryo açıklaması>

## 7. Görsel Modeller ve Diyagramlar

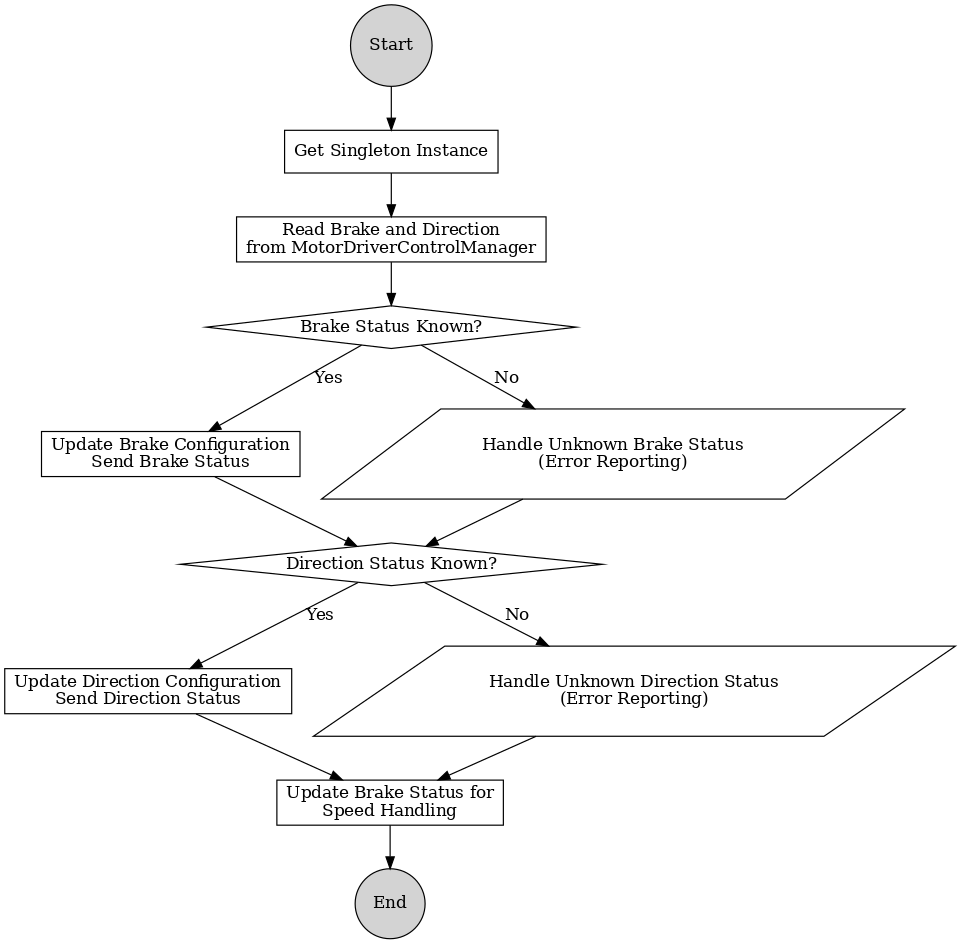
7.1 UML Diagram



7.2 BrakeAndDirectionManager\_ruInitialisation Activity Diagram



7.2 BrakeAndDirectionManager\_ruRefresh Activity Diagram



## 8. Güncelleme Kayıt Defteri

|  |
| --- |
|  |

Versiyon güncellemeleri sonunda hangi versiyon ne amaçla değiştiği buraya maddde madde yazılır.

V1.0.1-0 Belgede düzeltmeler yapılmıştır

## 9. Güncellenen Bütçe Kaydı

|  |
| --- |
|  |

Güncellenen versiyon bütçe gerektiren bir güncelleme ise burada kayıt tutulmaktadır.

|  |
| --- |
|  |

## 10. Ekler

|  |
| --- |
|  |