

İTÜ Rover Takımı

Hero ve Talon ile Motor sürme



Kemal Gür

27 Aralık 2025

# İçindekiler

<b>1 Amaç ve Kapsam</b>	<b>3</b>
<b>2 Gereksinimler</b>	<b>3</b>
2.1 Gerekli yazılımlar . . . . .	3
2.2 Gerekli Donanımlar . . . . .	3
<b>3 Donanım Kurulumu</b>	<b>3</b>
<b>4 Yazılım Kurulumu</b>	<b>4</b>
4.1 CAN ID Tespiti ve Konfigürasyon . . . . .	4
4.2 Visual Studio Proje Kurulumu ve Konfigürasyon . . . . .	4
<b>5 Örnek Senaryolar ve Kod İmplementasyonu</b>	<b>5</b>
5.1 Basit Motor Başlatma: . . . . .	5
5.2 Joystick Veri Okuma: . . . . .	6
5.3 Tek Motorlu Sürüş . . . . .	7
5.4 Çift Motorlu Sürüş: . . . . .	8

# 1 Amaç ve Kapsam

Bu proje, **CTRE HERO Development Board** kullanarak **Talon SRX** motor sürücüsünü CAN Bus üzerinden kontrol etmeyi ve bir **USB Joystick/Gamepad** (örn. Logitech F710) aracılığıyla motorun hızını ve yönünü anlık olarak değiştirmeyi amaçlar.

## 2 Gereksinimler

### 2.1 Gerekli yazılımlar

[Visual Studio 2019 Community](#) (Datasheette 16.2 önermiş ama sitede ona en yakın 16.4 sürümünü indirebilirsiniz sorunsuz çalışacaktır.)

[Phoenix Framework for HERO](#) (İndirirken Hero yazan kutucuklar görürseniz hepsine tıklayın indirsin.)

### 2.2 Gerekli Donanımlar

- **Denetleyici:** CTRE HERO Development Board
- **Motor Sürücü:** Talon SRX
- **Motor:** DC Motor
- **Giriş Cihazı:** USB Gamepad (Logitech F710 önerilir - XInput modunda)
- **Güç Kaynağı:** maksimum 24V Akü veya Güç Kaynağı
- **Bağlantı:** CAN Bus kabloları (Sarı/Yeşil), USB kablosu (HERO → Gamepad)
- **Kablo:** A to A USB kablosu ve mikro USB kablo

## 3 Donanım Kurulumu

### 1. CAN Bağlantısı:

- HERO Board üzerindeki CAN portunu (Üstünde yazıyor CANL(GREEN), CANL(YELLOW), Talon SRX'in CAN kablolarına bağlayın (Sarı → yellow, Yeşil → green)).
- Talon SRX üzerindeki boşta kalan yeşil ve sarı kablolar, CAN Bus hattı üzerinden zincirleme bağlantı yapmak için ayrılmıştır. Sisteme birden fazla motor sürücü ekleneceği durumlarda, bu kabloların bir sonraki Talon SRX'in karşılık gelen (aynı renkteki) kabloları ile birleştirilmesi gerekmektedir.

### 2. Güç Bağlantısı:

- Talon SRX(ler)'i 24Vout yazan yere (+ Kırmızı, - Siyah) bağlayın.
- Motor(ların) uçlarını Talon SRX(lerin)'in diğer çıkışına (M+ / M-) bağlayın.

### 3. Joystick Bağlantısı:

- USB Gamepad'i HERO Board üzerindeki USB Host portuna takın.

- Gamepadı Xinput modunda kullanın (Joystickin arka tarafından değiştirebilirsiniz)
- Joystick verilerinin okunamaması durumunda giriş protokolünü etkileyen Mode tuşunu kontrol ediniz; sistem genellikle Mode ışığının sönük olduğu konfigürasyonda stabil çalışmaktadır.

## 4 Yazılım Kurulumu

### 4.1 CAN ID Tespiti ve Konfigürasyon

Sistemi kodlamaya başlamadan önce, bağlı olan motor sürücülerin (Talon) CAN ID'lerinin tespit edilmesi gerekmektedir. Bunun için aşağıdaki adımları izleyin:

1. **Yazılım Hazırlığı:** Bilgisayarınızda Phoenix LifeBoat uygulamasını çalıştırın ve sol üst köşedeki “CAN Bus Devices” sekmesine gidin.
2. **Donanım Bağlantısı:** Bilgisayarınız ile HERO Board üzerindeki USB Host portunu birbirine bağlayın.
3. **Önemli:** Bu bağlantı için mutlaka **USB Tip A'dan Tip A'ya (Male-to-Male)** kablo kullanılmalıdır. Standart MicroUSB bağlantısı bu işlemede cihazları görüntülemek için yeterli değildir.
4. **Güç ve Tespit:** Sisteme güç verin. Bağlantılar doğruysa, ekranda aktif olan tüm Talon SRX cihazları ve mevcut ID'leri listelenecaktır.
5. **Konfigürasyon Notu:** Listelenen ID numaralarını mutlaka not edin. İlerleyen aşamalarda USB Host portu Joystick bağlantısı için kullanılacağından, bu ekrana tekrar erişiminiz olmayacağından emin olunmalıdır.

### 4.2 Visual Studio Proje Kurulumu ve Konfigürasyon

Kodlama aşamasına geçmeden önce, HERO Board ile haberleşme ayarlarının doğru yapıldığından emin olunmalıdır.

#### 1. Proje Oluşturma:

- Visual Studio'yu başlatın ve “Create a new project” seçenekine tıklayın.
- Arama çubuğu veya şablon listesine bakarak “HERO Simple Application” şablonunu seçin ve Next diyerek projeyi oluşturun.

#### 2. Deployment (Yükleme) Ayarları:

- Sağ taraftaki Solution Explorer panelinde “Properties” öğesine çift tıklayın.
- Açılan pencerede soldaki menüden .NET Micro Framework sekmesine gelin.
- Burada şu ayarların seçili olduğunu doğrulayın:
  - **Transport:** USB
  - **Device:** Cross Link HERO\_NETMF\_Mini-USB

#### 3. Geliştirme Ortamı:

- Ayarları kaydettikten sonra **Program.cs** dosyasını açarak kodlama ekranına dönün.

## 5 Örnek Senaryolar ve Kod İmplementasyonu

Projenin gereksinimlerine göre (Joystick entegrasyonu, otonom sürüş vb.) farklı kod blokları kullanılabılır. Aşağıda, temel işlevleri test etmek ve sistemi ayağa kaldırmak için kullanabileceğiniz referans kod yapıları verilmiştir:

### 5.1 Basit Motor Başlatma:

Program başladığında motoru belirli bir hızda döndüren başlangıç kodu.

```
1  using System;
2  using System.Threading;
3  using Microsoft.SPOT;
4  using CTRE.Phoenix;
5  using CTRE.Phoenix.MotorControl;
6  using CTRE.Phoenix.MotorControl.CAN;
7
8  namespace MotorTest
9  {
10     public class Program
11     {
12         public static void Main()
13         {
14             // --- 1. Motoru Tanımlama ---
15             // Parantez içindeki '10', LifeBoat'ta gordugun Device ID dir.
16             // Eger motorunun ID'si farklıysa 10 yerine o sayiyi yaz.
17             TalonSRX motorum = new TalonSRX(10);
18
19             // --- 2. Güvenlik Ayarı (Fabrika Ayarları) ---
20             motorum.ConfigFactoryDefault();
21
22             Debug.Print("Motor Testi Başlıyor...");
23
24             // --- 3. Sonsuz Dongu (Motor Surekli Çalışın Diye) ---
25             while (true)
26             {
27                 // Motoru %100 günde ileri dondur.
28                 motorum.Set(ControlMode.PercentOutput, 1.0);
29
30                 // Watchdog'u besle (HERO kartı, sistemin donmadığını anlamak için bunu
31                 // ister)
32                 CTRE.Phoenix.Watchdog.Feed();
33
34                 // 20 milisaniye bekle (İşlemciyi yormamak için)
35                 Thread.Sleep(20);
36             }
37         }
38     }
```

## 5.2 Joystick Veri Okuma:

Gamepad üzerinden Y ekseni (Y-axis) verisinin anlık takibi.

```
1  using System;
2  using System.Threading;
3  using Microsoft.SPOT;
4  using Microsoft.SPOT.Hardware;
5
6  namespace Hero_Simple_Application5
7  {
8      public class Program
9      {
10         public static void Main()
11         {
12             /* bir gamepad (oyun kolu) nesnesi olustur */
13             CTRE.Phoenix.Controller.GameController myGamepad = new CTRE.Phoenix.Controller.
14                 GameController(new CTRE.Phoenix.UsbHostDevice(0));
15
16             /* debugger uzerinden izlemek icin basit bir sayac */
17             int counter = 0;
18
19             /* sonsuz dongu */
20             while (true)
21             {
22                 /* while dongusunun icine eklenen kisim */
23                 if (myGamepad.GetConnectionStatus() == CTRE.Phoenix.UsbDeviceConnection.Connected)
24                 {
25                     /* eksen (axis) degerini yazdir - Hata duzeltildi */
26                     Microsoft.SPOT.Debug.Print("axis: " + myGamepad.GetAxis(1).ToString());
27
28                     /* motor kontrolune izin vermek icin watchdog beslenir */
29                     CTRE.Phoenix.Watchdog.Feed();
30                 }
31
32                 /* sayaci artir */
33                 ++counter;
34
35                 /* buraya breakpoint koymak 'counter' degiskeninin anlik degerini gorebilirsin */
36
37                 /* biraz bekle */
38                 System.Threading.Thread.Sleep(10);
39             }
40         }
41     }
```

### 5.3 Tek Motorlu Sürüş

Bu kod, motorun düşük voltajlarda hareket etmeye (sürtünme) sorununu çözer ve Joystick merkezdeyken motorun titremesini engeller.

```
1  using System;
2  using System.Threading;
3  using Microsoft.SPOT;
4  using CTRE.Phoenix;
5  using CTRE.Phoenix.MotorControl;
6  using CTRE.Phoenix.MotorControl.CAN;
7  using CTRE.Phoenix.Controller;
8
9  namespace Hero_Motor_Gamepad_Control
10 {
11     public class Program
12     {
13         public static void Main()
14         {
15             // ===== AYARLAR =====
16             // Yon ters ise burayı 'true' yap, duz ise 'false' kalsın.
17             bool tersYon = false;
18
19             // Motorun donmeye basladığı en küçük güç.
20             // Redüktörler motorları genelde 0.15 veya 0.20'nin altında donmez.
21             // Eğer motor geç teptik veriyorsa bu sayıyı arttır (orn: 0.25f).
22             float minHareketGucu = 0.20f;
23             // -----
24
25             TalonSRX motorum = new TalonSRX(10);
26             motorum.ConfigFactoryDefault();
27
28             // Kolu bıracıkça motor zınk diye dursun (Coast yerine Brake)
29             motorum.SetNeutralMode(NeutralMode.Brake);
30
31             GameController myGamepad = new GameController(new UsbHostDevice(0));
32             Debug.Print("Hiz ve Yon ayarları kod başladı... ");
33
34             while (true)
35             {
36                 if (myGamepad.GetConnectionStatus() == UsbDeviceConnection.Connected)
37                 {
38                     // 1. Veriyi Oku (-1.0 ile 1.0 arası)
39                     // Logitech F710/F310 kullanıyorsan arkadaki salter 'X' modunda olsun.
40                     float joystickInput = myGamepad.GetAxis(1);
41
42                     // 2. Yon Ayarı (Ters ise çevir)
43                     if (tersYon) joystickInput = -joystickInput; }
44
45                     // 3. Hesaplanacak Motor Gücü
46                     float motorOutput = 0.0f;
47
48                     // Eğer kol hareket ettirildiyse (Kucuk titremeleri yoksay: 0.05)
49                     if (System.Math.Abs(joystickInput) > 0.05f)
50                     {
51                         // --- HATEMATIKSEL DUZELTME ---
52                         // Kol azıcık ittiše bile motora 'minHareketGucu' kadar enerji verelim.
53                         if (joystickInput > 0) // İLERİ GİDİYORSA
54                         {
55                             motorOutput = minHareketGucu + (joystickInput * (1.0f - minHareketGucu));
56                         }
57                         else // GERİ GİDİYORSA
58                         {
59                             motorOutput = -minHareketGucu + (joystickInput * (1.0f - minHareketGucu));
60                         }
61                     }
62                     else
63                     {
64                         motorOutput = 0.0f; // Kol merkezde, dur.
65                     }
66
67                     // 4. Motora Gönder
68                     motorum.Set(ControlMode.PercentOutput, motorOutput);
69
70                     // 5. Kontrol Et
71                     Debug.Print("Input: " + joystickInput + " -> Output: " + motorOutput);
72                     CTRE.Phoenix.Watchdog.Feed();
73
74                 Thread.Sleep(20);
75             }
76         }
77     }
78 }
```

## 5.4 Çift Motorlu Sürüş:

İki motorun senkronize veya bağımsız olarak Joystick ile yönetilmesi.

```
1  using System;
2  using System.Threading;
3  using Microsoft.SPOT;
4  using CTRE.Phoenix;
5  using CTRE.Phoenix.MotorControl;
6  using CTRE.Phoenix.MotorControl.CAN;
7  using CTRE.Phoenix.Controller;
8
9  namespace Hero_Motor_Secmeli_Final
10 {
11     public class Program
12     {
13         public static void Main()
14         {
15             // --- GUVENLIK BEKLEMESİ (Cok Onemli) ---
16             // Kart acilir acilmaz USB'yi mesgul etmesin diye 3 saniye susuyoruz.
17             // Bu sayede Device not found hatasından kurtulursun.
18             Debug.Print( $istem aciliyor... Lutfen bekleyiniz... );
19             Thread.Sleep(3000);
20
21             // ===== AYARLAR =====
22             // ID'LERINE GORE AYARLADIM:
23             int motor1_ID = 10; // Talon SRX(1)
24             int motor2_ID = 5; // left_rear
25
26             float minHareketGucu = 0.20f; // Motorun kimildamasi icin min guc
27             // =====
28
29             // Motorlari Tanimla
30             TalonSRX motor1 = new TalonSRX(motor1_ID);
31             TalonSRX motor2 = new TalonSRX(motor2_ID);
32
33             // Ayarlar
34             motor1.ConfigFactoryDefault();
35             motor2.ConfigFactoryDefault();
36
37             // Fren Modu (Kolu bıracınca motor akmasın, dursun)
38             motor1.SetNeutralMode(NeutralMode.Brake);
39             motor2.SetNeutralMode(NeutralMode.Brake);
40
41             GameController myGamepad = new GameController(new UsbHostDevice(0));
42             int seciliMotor = 1;
43
44             Debug.Print( $istem Hazir. A (Yesil) veya B (Kirmizi) tusuna bas. );
45
46             while (true)
47             {
48                 // Baglanti koparsa kod patlamsasin diye kontrol
49                 if (myGamepad.GetConnectionStatus() == UsbDeviceConnection.Connected)
50                 {
51                     // --- TUS ILE SECIM ---
52                     if (myGamepad.GetButton(1)) // A Tusu (Genellikle Yesil)
53                     {
54                         seciliMotor = 1;
55                         Debug.Print( SECILDI: MOTOR ID 10 );
56                     }
57                     if (myGamepad.GetButton(2)) // B Tusu (Genellikle Kirmizi)
58                     {
59                         seciliMotor = 2;
60                         Debug.Print( SECILDI: MOTOR ID 5 );
61                     }
62
63                     // --- JOYSTICK OKUMA ---
64                     float axisValue = myGamepad.GetAxis(1); // Sol analog
65                     float motorGucu = 0.0f;
66
67                     // --- GUC HESAPLAMA ---
68                     if (System.Math.Abs(axisValue) > 0.05f)
69                     {
70                         if (axisValue > 0)
71                             motorGucu = minHareketGucu + (axisValue * (1.0f - minHareketGucu));
72                         else
73                             motorGucu = -minHareketGucu + (axisValue * (1.0f - minHareketGucu));
74                     }
75
76                     // --- SURME ---
77                     if (seciliMotor == 1)
78                     {
79                         motor1.Set(ControlMode.PercentOutput, motorGucu);
80                         motor2.Set(ControlMode.PercentOutput, 0.0f); // Digerini durdur
81                     }
82                     else
83                     {
84                         motor2.Set(ControlMode.PercentOutput, motorGucu);
85                         motor1.Set(ControlMode.PercentOutput, 0.0f); // Digerini durdur
86                     }
87
88                     // Watchdog (Motor güvenligi)
89                     CTRE.Phoenix.Watchdog.Feed();
90                 }
91             }
92         }
93     }
94 }
95 }
```