

Hidsharp

HidSharp, USB HID aygıtları için çok platformlu bir C # kitaplığıdır. Sürüm 2.0, herhangi bir USB HID giriş cihazından raporları okumanıza ve ayrıştırmanıza olanak tanır. USB oyun kollarını, türevlerini ve ihtiyacınız olan her şeyi okuyabilirsiniz. Bunu, tam rapor tanımlayıcı ve rapor ayrıştırma yeteneği sağlayarak yapar. HIDSharp, bunu yapabilen ilk sürücüsüz çapraz platform (cross-platform) kitaplığıdır! Tabii ki, ham okuma ve yazma hala tam olarak destekleniyor. Sürüm 1.5 ve altı, VB6 ve MS Access programlarının kullanımına izin vermek için COM etkindir (.NET Framework elbette yüklenmelidir).

HIDSharp, Windows, MacOS ve Linux'u (hidraw) destekler. HIDSharp, Apache açık kaynak lisansını kullanır.

İlk olarak James F. Bellinger tarafından yazılmıştır. İnternet adresi (<https://www.zer7.com/software/hidsharp>) ve Nuget paketi olarak mevcuttur.

Sürüm Tarihi

2.1.0 (4 Mayıs 2019):

Windows' a Bluetooth Düşük Enerji desteği eklendi. Kütüphanenin bu bölümü ileride değişebilir.

Ayrıcalık katmanındaki (exclusivity layer) bir yarış durumu (race condition) düzeltildi.

Linux'ta .NET Core'da ortaya çıkan bir hata (bug) düzeltildi.

Windows'un buggy usbser.sys sürücüsünü kullanan seri aygıtlar artık bağlantı ve bağlantı kesilmesinin algılanması da dahil olmak üzere HIDSharp ile uyumludur.

2.0.8 (21 Ocak 2019):

SerialStream'de yedi veri biti ve çift / tek eşlik (parity bit) için destek eklendi.

MacOS'ta, seri numarası olmayan HID cihazlarının görünmemesine neden olan bir hata düzeltildi.

2.0.6 (27 Aralık 2018):

SerialStream'de iki durdurma biti için destek eklendi.

2.0.5 (15 Ekim 2018):

HIDSharp artık Linux'ta .NET Core ile uyumludur.

MacOS'ta çöp toplayıcıyla(garbage-collector) ilgili bir kilitlenme düzeltildi.

2.0.2 (5 Mayıs 2018):

Linux'ta, Rapor Kimliği (Report ID) kullanmayan cihazlarda çıktı raporlarının yanlış gönderilmesine neden olan bir hata düzeltildi.

DeviceList GetSerialDeviceOrNull () artık dosya sistemi adıyla (COM1, / dev / ttyUSB, vb.) Ve DevicePath ile eşleşecektir.

2.0.1 (11 Nisan 2018):

Windows'ta, bazı durumlarda, bir cihazın rapor tanımlayıcısının yeniden oluşturulamaması, o cihazın açılmasını tamamen engelleyebileceği bir hata düzeltildi.

HIDSharp artık hem .NET Framework 3.5+ hem de .NET Standard 2.0+ için derlemeler içermektedir.

2.0 (9 Nisan 2018):

Rapor tanımlayıcı ayrıştırma işlevi büyük ölçüde geliştirildi.

Artık tüm platformlarda herhangi bir HID cihazının raporlarının kodunu çözebilirsiniz (neredeyse)!

MacOS sürüm 10.8 ve üzeri için GetReportDescriptor desteği eklendi.

Windows'a GetReportDescriptor desteği eklendi. Windows bu bilgilere doğrudan erişmenin bir yolunu sağlamadığından, bu tanımlayıcı bir yeniden yapılanmadır.

DeviceList'e AreDriversBeingInstalled özelliği eklendi. (Windows bazen sürücülerini yüklenirken açılan bileşik HID aygıtlarının akışlarını kapatır.)

Cihazın bağlanıp bağlantısını kesmeyi algılamak için DeviceList'ne bir Değiştirilmiş olay (event) eklendi.

Cihaza bir NativeImplementation özelliği eklendi.

Kesinti talep etme özelliğine sahip bir ayrıcalık katmanı eklendi. Bu süreçler arası işbirliği için kullanışlıdır, ancak manuel olarak etkinleştirilmesi gerekir.

Seri bağlantı noktaları için destek eklendi. Bu hala Linux'ta biraz hatalı, ancak MacOS'ta Mono yerel uygulamadan daha iyi çalışıyor.

Linux libudev.so.1 için destek eklendi.

Açıklanan GetFileSystemName (). Bu, Linux'ta hidraw izinlerini kontrol etmek için kullanılabilir.

Bir Rapor Kimliği (Report ID) i kullanmayan cihazlar için MacOS'ta sabit maksimum rapor uzunlukları düzeltildi.

Aygıtın bağlantısı kesildiğinde Linux ve MacOS'un HID Read () üzerinde IOException oluşturmaması düzeltildi.

Son Mono sürümlerinde Linux desteği düzeltildi.

32 bit Linux desteği (Raspberry Pi) düzeltildi.

GetFeature () ve SetFeature () artık Linux'ta çalışıyor.

HIDSharp artık Apache açık kaynak lisansını kullanıyor.

1.5 (18 Mayıs 2013):

HidDevice'e DevicePath özelliği eklendi. (Bu, aynı VID, PID vb.'ye sahip cihazları ayırt etmek için kullanışlıdır.)

1.4.0.1 (21 Nisan 2013):

Eksik XML dokümantasyon dosyası arşive eklendi.

1.4 (20 Nisan 2013):

En önemli sınıfları belgelendi.

64 bit Windows'ta çıktı raporlarıyla ilgili bir sorun düzeltildi.

Windows'ta bir zaman aşımı hatası düzeltildi.

Paralleştirilmiş cihaz numaralandırması eklendi.

Windows'ta ikili sistem yerleştirmeyi önlemek için yeniden sıralı platform algılama eklendi.

1.3 (23 Şubat 2013):

Linux libudev referansının geliştirici paketi sembolik bağlantılarını gerektirmemesi düzeltildi.

1.2 (16 Ekim 2012):

MacOS desteği düzeltildi.

1.1 (28 Temmuz 2012):

MacOS ve Linux desteği eklendi.

Rapor tanımlayıcı ayrıştırma eklendi.

1.0 (28 Ağustos 2010):

İlk sürüm.

HidSharp Namespace

HidSharp, Windows, MacOS ve Linux için HID API'lerini tamamlar. HidDevice nesnelerini numaralandırmak için DeviceList'i kullanmak gerekir ve ardından bir HidStream almak için HidDevice'i açmalıyız.

- **HidDevice**: Bir USB HID sınıfı cihazı temsil eder.
- **HidDeviceLoader**: Sisteme bağlı USB HID sınıfı cihazlarını algılar. [Obsolete]
- **HidStream**: USB HID sınıfı bir cihazla iletişim kurar.
- **DeviceList**: Sisteme bağlı USB HID sınıfı cihazlarını algılar.

HidDevice Members

Bir USB HID sınıfı cihazı temsil eder. Kalıtım Hiyerarşisi, Device class'ından türetilmiştir.

Methods:

- **GetReportDescriptor()**: USB aygıtının ham rapor tanımlayıcısını döndürür. Şu anda bu yalnızca Linux'ta desteklenmektedir.
- **Open()**: USB HID sınıfı cihazla bağlantı kurar veya bağlantı kurulamazsa bir istisna (exception) atar.
- **ToString()**: Geçerli Object 'i temsil eden bir String döndürür. (Object ToString'i geçersiz kılar.)
- **TryOpen()**: USB HID sınıfı cihaza bağlantı kurmaya çalışır. Kuramaz ise herhangi bir exception fırlatmaz.
- **GetManufacturer()**: Üretici adını döndürür.
- **GetMaxFeatureReportLength()**: Rapor Kimliği baytı dahil maksimum özellik raporu uzunluğunu döndürür. Cihaz Rapor Kimliklerini kullanmıyorsa, ilk bayt için 0 kullanın.
- **GetMaxInputReportLength()**: Rapor Kimliği baytı dahil maksimum giriş raporu uzunluğunu döndürür. Cihaz Rapor Kimliklerini kullanmıyorsa, ilk bayt her zaman 0 olacaktır.
- **GetMaxOutputReportLength()**: Rapor Kimliği baytı dahil maksimum çıktı raporu uzunluğunu döndürür. Cihaz Rapor Kimliklerini kullanmıyorsa, ilk bayt için 0 kullanın.
- **GetProductName()**: Üretici adını döndürür.
- **GetRawReportDescriptor()**: USB aygıtının ham rapor tanımlayıcısını döndürür.
- **GetReportDescriptor()**: USB cihazının rapor tanımlayıcısını alır ve ayrıştırır.
- **GetSerialNumber()**: Cihazın seri numarasını verir.
- **GetSerialPorts()**: Bileşik USB aygıtının seri bağlantı noktalarını döndürür. Şu anda bu yalnızca Windows'ta desteklenmektedir.

Properties:

- **DevicePath**: Aygıt için işletim sisteminin adı. Aynı Satıcı Kimliğine, Ürün Kimliğine, Seri Numarasına. vb., sahip birden fazla cihazınız varsa bu onları ayırt etmek için faydalı olabilir.
- **Manufacturer**: Üretici adı.
- **MaxFeatureReportLength**: Rapor Kimliği (Report ID) baytı dahil maksimum özellik raporu uzunluğu. Cihaz Rapor Kimliklerini kullanmıyorsa, ilk bayt için 0 kullanın.
- **MaxInputReportLength**: Rapor Kimliği (Report ID) baytı dahil maksimum giriş raporu uzunluğu. Cihaz Rapor Kimliklerini kullanmıyorsa, ilk bayt her zaman 0 olacaktır.
- **MaxOutputReportLength**: Rapor Kimliği baytı dahil maksimum çıktı raporu uzunluğu. Cihaz Rapor Kimliklerini kullanmıyorsa, ilk bayt için 0 kullanın.
- **ProductID**: USB ürün kimliği. Bunlar şu adreste listelenmiştir: <http://usb-ids.gowdy.us>
- **ProductName**: Ürün adı.
- **ProductVersion**: Ürün versiyonu. Bu, sırasıyla üst ve alt 8 bitteki ana ve küçük sürümleri kodlayan 16 bitlik bir sayıdır.
- **SerialNumber**: Cihazın seri numarası.
- **ReleaseNumber**: Cihaz sürüm numarası.
- **ReleaseNumberBcd**: Cihaz sürüm numarası decimal tipinde.
- **VendorID**: USB satıcı kimliği. Bunlar şu adreste listelenmiştir: <http://usb-ids.gowdy.us>

HidDeviceLoader Members [Obsolete]

HidDeviceLoader, HidDeviceLoader sınıfının yeni bir örneğini başlatır. [Obsolete]

HidDeviceLoader class'ının yerine DeviceList class'ı kullanılıyor.

Methods:

- **GetDeviceOrDefault()**: Belirlenen kriterlere uyan ilk bağlanan USB cihazını alır.
- **GetDevices()**: Bağlı USB cihazlarının bir listesini alır.
- **GetDevices (Nullable Int32, Nullable Int32, Nullable Int32, String)**: Bazı kriterlere göre filtrelenmiş bağlı USB cihazlarının bir listesini alır.
- **GetDevicesVB()**: Bağlı USB cihazlarının bir listesini alır. Bu aşırı yükleme, Visual Basic 6 ve COM istemcileri içindir.

HidStream Members

HidStream class'ı DeviceStream class'ndan miras (inheritance) almıştır.

Methods:

- **BeginRead():** (Overrides Stream BeginRead (Byte, Int32, Int32, AsyncCallback, Object).)
- **BeginWrite():** (Overrides Stream BeginWrite (Byte, Int32, Int32, AsyncCallback, Object).)
- **EndRead():** Bekleyen asenkron okumanın tamamlanmasını bekler. (Overrides Stream EndWrite(IAsyncResult).)
- **EndWrite():** Asenkron bir yazma işlemini sonlandırır.
- **GetFeature(Byte):** Bir Get Özelliği kurulum isteği gönderir.
- **GetFeature(Byte, Int32, Int32):** Bir Get Özelliği kurulum isteği gönderir.
- **Read():** HID Giriş Raporlarını okur.
- **Read(Byte):** HID Giriş Raporlarını okur.
- **SetFeature(Byte):** Bir Özellik Ayarlama kurulum isteği gönderir.
- **SetFeature(Byte, Int32, Int32):** Bir Özellik Ayarlama kurulum isteği gönderir.
- **Write():** Cihaza bir HID Çıktı Raporu yazar.
- **Write(Byte):** Cihaza bir HID Çıktı Raporu yazar.

Properties:

- **Device:** Bu akışla ilişkili HidDevice'i alır.
- **ReadTimeout:** Bir HID raporu almak için maksimum beklenecek süre (Milisaniye). Varsayılan, 3000 milisaniyedir. Zaman aşımını devre dışı bırakmak için bunu Infinite olarak ayarlayın.
- **WriteTimeout:** Bir HID raporu onaylaması için maksimum beklenecek süre (Milisaniye). Varsayılan, 3000 milisaniyedir. Zaman aşımını devre dışı bırakmak için bunu Infinite olarak ayarlayın.
- **Tag:** Akış (Stream) hakkında kullanıcı tanımlı verileri depolayan bir nesne.
- **CanTimeout:** Geçerli akışın zaman aşımını uğrayıp uğramayacağına göre bool bir değer döndürür.
- **CanWrite:** Akış yazmayı destekliyorsa true, desteklemiyorsa false döndürür.
- **CanSeek:** Akış aramayı destekliyorsa true, desteklemiyorsa false döndürür.
- **CanRead:** Akış okumayı destekliyorsa true, desteklemiyorsa false döndürür.

Events:

- **EventHandler InterruptRequested:** HidSharp.OpenOption.Interruptible true olduğunda ve daha yüksek önceliğe sahip başka bir işlem veya iş parçacığı akışı açmak istediğinde oluşur.
- **EventHandler Closed:** Akış kapatıldığında ortaya çıkar.

DeviceList Members

Methods:

- [GetAllDevices\(\)](#): Bağlı tüm HID, BLE ve seri cihazların bir listesini alır.
- [GetAllDevices\(DeviceFilter\)](#): Bazı kriterlere göre filtrelenmiş bağlı cihazların bir listesini alır.
- [GetBleDevices\(\)](#): Bağlı tüm BLE cihazlarının bir listesini alır. (Bluetooth Low Energy)
- [GetDevices\(\)](#): Bağlı USB cihazlarının bir listesini alır.
- [GetDevices\(DeviceTypes, DeviceFilter\)](#): Bazı kriterlere göre filtrelenmiş bağlı USB cihazlarının bir listesini alır.
- [GetHidDeviceOrNull\(Int32, Int32, Int32, String\)](#): Belirtilen kriterlere uyan ilk bağlanan HID cihazını alır. Cihaz yok ise null döndürür.
- [GetHidDevices\(Int32, Int32, Int32, String\)](#): Bazı kriterlere göre filtrelenmiş bağlı HID cihazlarının bir listesini alır.
- [GetHidDevices\(\)](#): Bağlı tüm HID cihazlarının bir listesini alır.
- [GetSerialDeviceOrNull\(String\)](#): Bağlı seri cihazı, belirtilen cihaz yolu veya dosya sistemi adı ile alır. Aygıtı veya hiçbir şey bulamadıysa null döndürür.
- [GetSerialDevices\(\)](#): Bağlı tüm seri cihazların bir listesini alır. Cihaz listesini döndürür.
- [RaiseChanged\(\)](#): HidSharp.DeviceList.Changed olayını oluşturur.

Properties:

- [Local](#): Bu bilgisayardaki cihazların listesi.
- [AreDriversBeingInstalled](#): Sürücüler doğru şekilde yüklendiyse true, değil ise false döndürür.
- [EventHandler<DeviceListChangedEventArgs> Changed](#): Bir cihaz bağlandığında veya bağlantısı kesildiğinde oluşur.

HidSharp.Reports Namespace

Classes:

[DataConvert](#), [DataItem](#), [DescriptorCollectionItem](#), [DescriptorItem](#), [DeviceItem](#), [Indexes](#), [IndexList](#), [IndexRange](#), [Report](#), [ReportDescriptor](#)

Enums:

DataItemFlags, ExpectedUsageType, ReportType, Usage

Delegates:

ReportScanCallback, ReportValueCallback

Struct:

DataValue

I. ReportDescriptor

ReportDescriptor, HidSharp.Reports.ReportDescriptor sınıfının yeni bir örneğini (instance) başlatır ve ham bir HID rapor tanımlayıcısını ayrıştırır (parse).

- **MaxInputReportLength**: Maksimum girdi raporu uzunluğudur. Rapor Kimliği (Report ID) bu uzunluğa dahildir.
- **MaxOutputReportLength**: Maksimum çıktı raporu uzunluğudur. Rapor Kimliği (Report ID) bu uzunluğa dahildir.
- **MaxFeatureReportLength**: Maksimum özellik raporu uzunluğudur. Rapor Kimliği (Report ID) bu uzunluğa dahildir.
- **ReportsUseID**: Cihaz Report ID'yi gönderirse true, göndermezse false değerini alır.
- **DeviceItems**: Her fiziksel HID cihazı, mantıksal cihazlara karşılık gelen bir dizi koleksiyon sunar. Örnek olarak basit bir oyun konsolu, oyun kumandası vb. için tipik olarak bir HidSharp.Reports.DeviceItem olacaktır. Çift gamepad adaptörlerinde iki tane olacaktır. Klavyelerde genellikle tuşları için bir tane ve ses seviyesi, medya kontrolleri için en az bir tane bulunur.
- **CreateHidDeviceInputReceiver()**: Bu cihazdan rapor almaya uygun bir HidSharp.Reports.Input.HidDeviceInputReceiver oluşturur.
- **GetReport(ReportType, Byte)**: Report tipinde aldığı raporu geriye döndürür.
- **TryGetReport(ReportType, Byte, Report)**: Raporu almaya çalışır eğer alırsa true, alamaz ise false döner.

II. Report

Report, HidSharp.Reports.Report sınıfının yeni bir örneğini (instance) başlatır.

- **Length**: Raporun uzunluğu. Rapor Kimliği (Report ID) bu uzunluğa dahildir.
- **ReportID**: Raporun Id'si.
- **ReportType**: Raporun hangi tipte olduğunu söyler (Input = 0, Output = 1, Feature = 2).

- [Read\(Byte, Int32, ReportScanCallback\)](#): Her veri ögesi için sağlanan metodu geri çağıran bir HID raporu okur. Bu geri arama, her veri ögesi için çağrılacaktır. İhtiyacınız olan her değeri okumak için bunu kullanın.
- [Write\(ReportScanCallback\)](#): Her veri ögesi için sağlanan metodu geri çağıran bir HID raporu yazar. Bu geri arama, her bir rapor veri ögesi için çağrılacaktır.
- [Write\(Byte, Int32, ReportScanCallback\)](#): Her veri ögesi için sağlanan metodu geri çağıran bir HID raporu yazar. Bu geri arama, her bir rapor veri ögesi için çağrılacaktır.

HidSharp.Reports.Encoding Namespace

Class:

[EncodedItem](#)

Enums:

[CollectionType](#), [GlobalItemTag](#), [ItemType](#), [LocalItemTag](#), [MainItemTag](#)

HidSharp.Reports.Input Namespace

Classes:

[DeviceItemInputParser](#), [HidDeviceInputReceiver](#)

HidSharp.Reports.Units Namespace

Enums:

[CurrentUnit](#), [LengthUnit](#), [LuminousIntensityUnit](#), [MassUnit](#), [TemperatureUnit](#), [TimeUnit](#), [UnitKind](#), [UnitSystem](#)

Struct:

Unit

HidSharp.Reports.Utility Namespace

Classes:

HidSharpDiagnostics, HResult

HidSharp.Experimental Namespace

Classes:

BleCharacteristic, BleDescriptor, BleDevice, BleService, BleStream

Enums:

BleCccd, BleCharacteristicProperties, BleRequestFlags

Structs:

BleEvent, BleUuid

